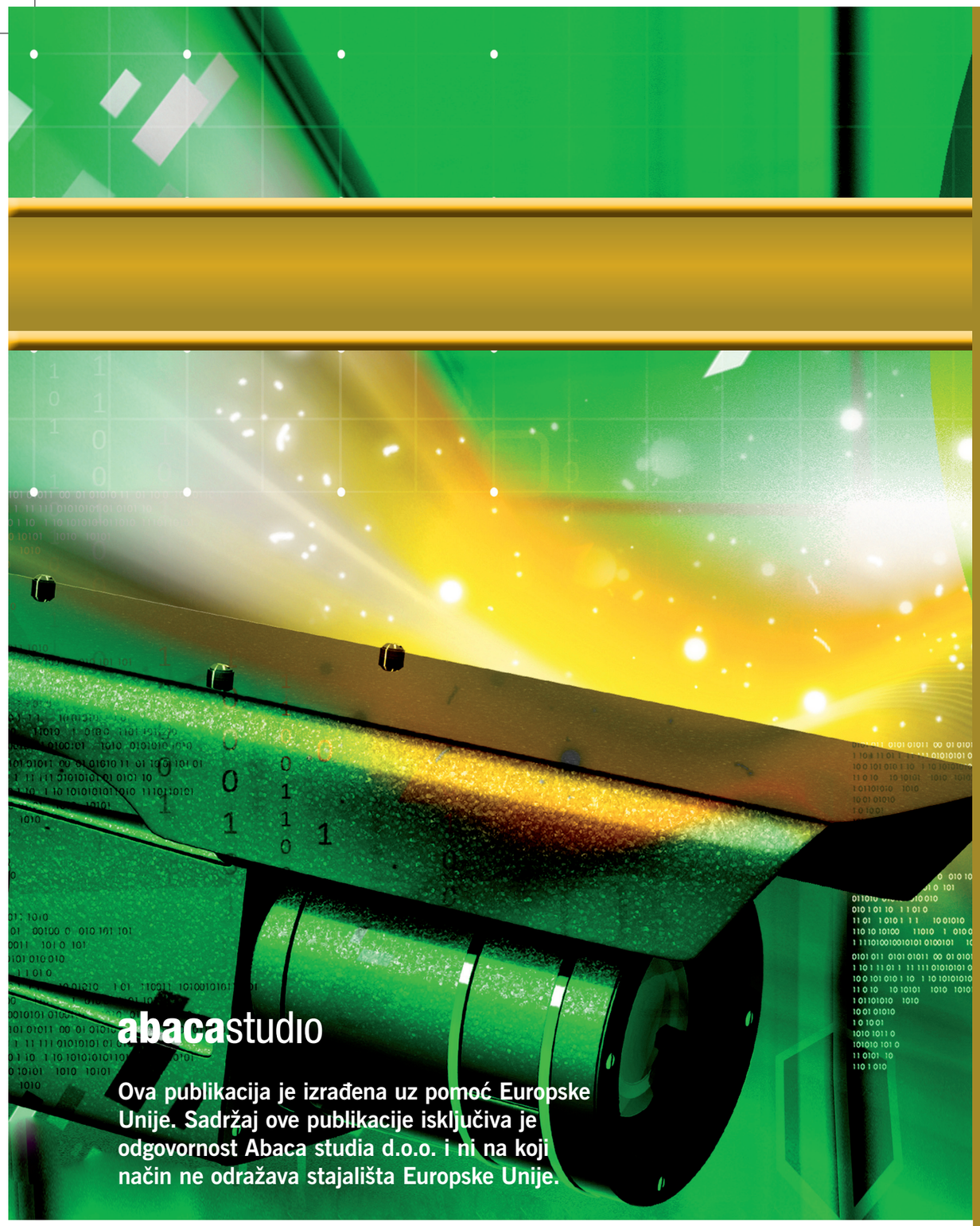


abacastudio

Ova publikacija je izrađena uz pomoć Europske Unije. Sadržaj ove publikacije isključiva je odgovornost Abaca studia d.o.o. i ni na koji način ne odražava stajališta Europske Unije.

Bojan Ždrnja SIGURNOST INFORMACIJSKIH SUSTAVA

priručnik za srednje strukovne škole

Bojan Ždrnja

SIGURNOST INFORMACIJSKIH SUSTAVA

abacastudio

Project is funded by the European Union
HRVATSKA UDRUGA POSLODAVACA
Srednja škola Krapina
TEHNIČKA ŠKOLA RUĐERA BOŠKOVIĆA ZAGREB
ELEKTROTEHNIČKA ŠKOLA SPLIT
FER
RAČUNARSTVO visoka škola
IPA RAZVOJ LJUDSKIH POTENCIJALA

Project is funded by the European Union
HRVATSKA UDRUGA POSLODAVACA
Srednja škola Krapina
TEHNIČKA ŠKOLA RUĐERA BOŠKOVIĆA ZAGREB
ELEKTROTEHNIČKA ŠKOLA SPLIT
FER
RAČUNARSTVO visoka škola
IPA RAZVOJ LJUDSKIH POTENCIJALA

BOJAN ŽDRNJA, dipl. ing.

SIGURNOST INFORMACIJSKIH SUSTAVA

Izdavač
Abacastudio d.o.o.

Za izdavača
Nataša Cesar

Urednik
Zlatan Morić

Lektorica
Dijana Stilinović

Oblikovanje naslovnice
Abaca studio d.o.o.

Tehnički urednik
Silvije Cesar

Ilustracije
Arhiva, Shutterstock i Wikimedia commons

Tisak
Abaca studio, Zagreb

Ova publikacija je izrađena
uz pomoć Europske unije.

Sadržaj ove publikacije isključiva je odgovornost Abaca studia d.o.o.
i ni na koji način ne odražava stajališta Europske unije.

Bojan Ždrnja

SIGURNOST INFORMACIJSKIH SUSTAVA

priručnik za srednje tehničke škole

Zagreb, rujan 2011.
abacastudio

Sadržaj

O IPA projektu izrade novih kurikuluma i nastavnih materijala

Računarstvo je, kao najkonkurentniji hrvatski izvozni sektor, ugroženo manjkom kadrova koji je već godinama prisutan. Takvom stanju nažalost doprinosi još uvijek razmjerno malen broj učenika (oko 1 300) koji se godišnje odlučuju upisati u program obrazovanja za stjecanje strukovne kvalifikacije/zanimanja tehničar za računalstvo, iako je riječ o najtraženijem obrazovnom programu u odjelu elektrotehnike i računarstva. Kako bismo potaknuli što veće zanimanje među učenicima i poboljšali kvalitetu obrazovanja unutar ovog strukovnog programa, okupili smo skupinu vrhunskih stručnjaka u području računarstva iz srednjih strukovnih škola: Tehničke škole Ruđera Boškovića iz Zagreba, Srednje škole Krapina i Elektrotehničke škole Split. Intenzivna suradnja sa stručnjacima s Fakulteta elektrotehnike i računarstva te iz Visoke škole za primijenjeno računarstvo u Zagrebu rezultirala je pripremom IPA projekta „Stvaranje novih mogućnosti za gospodarstvo utemeljeno na znanju u području ICT-a“ unutar grant sheme za izradu novih kurikuluma.

Cilj projekta financiranog sredstvima Europske unije, čiji je nositelj Tehnička škola Ruđera Boškovića iz Zagreba, bila je izrada novog kurikuluma za program obrazovanja Tehničar za računalstvo te izrada nastavnih materijala – udžbenika, zbirki zadataka, prezentacija, ispitnih pitanja…

Projekt je odobren za financiranje unutar programa IPA u rujnu 2010.

Nedostatak kvalitetnih nastavnih materijala unutar ovog i sličnih programa obrazovanja u koje se upisuje razmjerno malen broj učenika (zbog manjka interesa komercijalnih izdavača da pripreme i izdaju materijale) jedan je od znatnijih problema u osiguranju kvalitetne nastave. Situacija je posebno složena u računarstvu gdje privreda, ovisno o stopi rasta gospodarstva, ima tendenciju preuzimanja najkvalitetnijeg kadra iz obrazovnog sustava što još više naglašava potrebu da se uz novi kurikulum osiguraju i svi nastavni materijali koji mogu osigurati kvalitetnije izvođenje nastave čak i uz fluktuaciju nastavnog kadra.

Upravo je zbog toga, uz izradu novog kurikuluma, izrađena i skupina nastavnih materijala u koju spada i ova knjiga. Uvjereni smo da će ona doprinijeti izvedbi i organizaciji nastave unutar novog programa te posredno potaknuti ne samo interes učenika za karijere u području računarstva već i veću mogućnost zapošljavanja.

PROJEKTNI TIM

Predgovor

Poslovanje praktički svake korporacije danas ovisi o informacijama pohranjenim u njihovim informacijskim sustavima. Informacijski su sustavi tako postali integralni dio elektroničkog poslovanja bez kojeg se ono ne može niti zamisliti.

S druge strane, strmovit i dinamičan razvoj informacijske tehnologije postavlja velike zahtjeve pred zaštitu informacija i informacijskih sustava. Nedovoljno ulaganje u zaštitu informacijskih sustava u zadnjem desetljeću kulminiralo je velikim brojem sigurnosnih incidenata koji su obilježili početak 21. stoljeća. Žrtve ovakvih sigurnosnih incidenata bile su ne samo male tvrtke već i velike korporacije, kao i vladine institucije u svim dijelovima i državama svijeta.

Iako je tehnologija u zadnjih desetak godina uvelike napredovala, osnove zaštite informacijskih sustava nisu se mijenjale. Cilj ovog priručnika jest omogućiti upoznavanje sa širokim područjem zaštite informacijskih sustava, uključujući ne samo tehničke metode zaštite već i fizičke i administrativne.

Razumijevanje pojmova, metoda i tehnika zaštite informacijskih sustava učenicima će omogućiti implementiranje osnovne razine zaštite informacijskog sustava. Uz navedeno, cilj priručnika bio je omogućiti daljnji razvoj učenika te probuditi njihov interes za ovim područjem IT-a koje se iznimno brzo razvija i bez kojeg se ne može zamisliti današnje elektroničko poslovanje.

AUTOR

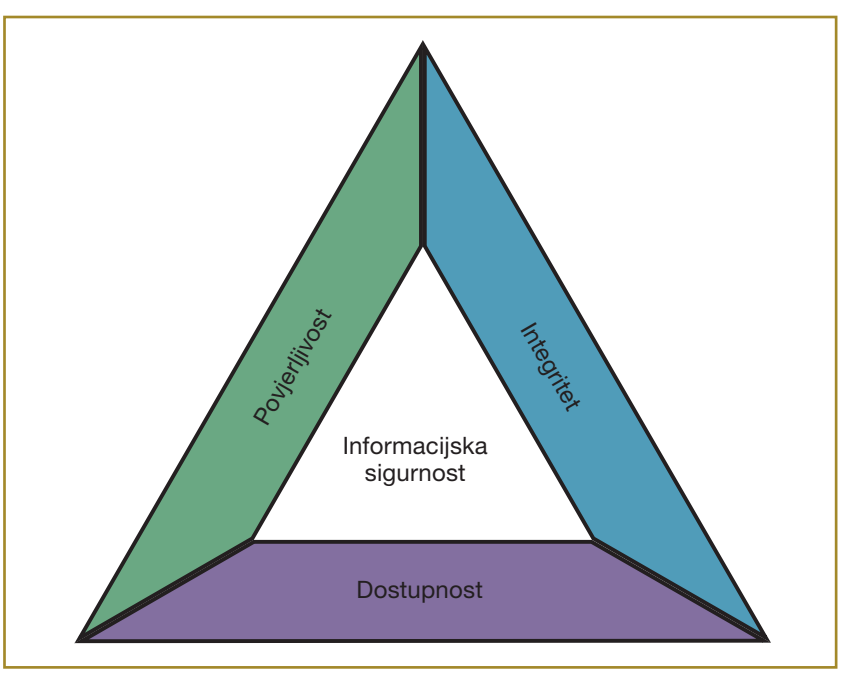
1. Uvod

Još od samih početaka ljudskog roda bilo je jasno da određene informacije treba tretirati s povjerenjem. Tako je Julije Cezar još 50 godina prije Krista izmislio poznati Cezarov kriptografski algoritam kako bi spriječio neovlašteno čitanje poruka koje su njegovi glasnici prenosili.

Informacijska sigurnost često se povezuje s računalima i računalnim mrežama, no potrebno je napomenuti da, kada govorimo o informacijskoj sigurnosti, ne govorimo isključivo o podacima pohranjenim na računalima već o svim informacijama koje je potrebno zaštititi od neovlaštenog pristupa, promjene ili uništavanja, bez obzira na oblik i mjesto gdje su ove informacije pohranjene.

1.1. CIA trokut

Sigurnost informacijskog sustava može se prikazati kao zbroj triju ključnih komponenti: povjerljivosti, integriteta i dostupnosti. Ove tri komponente predstavljaju temelje sigurnog informacijskog sustava te se obično predstavljaju kao osnovni sigurnosni trokut, prikazan na sljedećoj slici.



Slika 1.1. CIA trokut

Povjerljivost (engl. *Confidentiality*), kao što je već spomenuto, predstavlja jedan od najčešćih zahtjeva koji se postavljaju pred informacijske sustave. Jednostavno rečeno, cilj povjerljivosti jest osigurati da samo one osobe koje trebaju imati pristup određenim informacijama taj pristup i imaju. Sve druge osobe ne smiju moći pristupiti informacijama kojima nemaju dozvolu pristupati.

Povjerljivost podataka danas se uglavnom osigurava implementiranjem kontrola pristupa na različitim razinama (lokalne i mrežne) koje omogućuju sprječavanje neovlaštenih osoba u pristupanju navedenim podacima. Povjerljivost je moguće osigurati i pomoću enkripcije (više o enkripciji u poglavlju 3.3), matematičkih algoritama čiji je cilj transformacija informacije u nečitljivu, čime se sprječava neovlašteno čitanje i mijenjanje podataka.

Komponenta **integriteta** (engl. *Integrity*) osigurava da samo ovlaštene osobe/mehanizmi mogu promijeniti pohranjene informacije. Drugim riječima, integritet onemogućuje bilo kakvu neovlaštenu promjenu informacija.

Integritet pohranjenih informacija može biti narušen na različite načine. Primjerice, neovlašteni korisnik može promijeniti pohranjene informacije (npr. stanje računa u banci). No integritet informacija mogu narušiti i ovlašteni korisnici – npr. ovlašteni korisnik može slučajno obrisati dio neke datoteke i na taj način narušiti njezin integritet. Konačno, integritet može biti narušen zbog nekog vanjskog utjecaja kao što je nestanak struje, u kojem se slučaju datoteka može neispravno pohraniti na tvrdi disk.

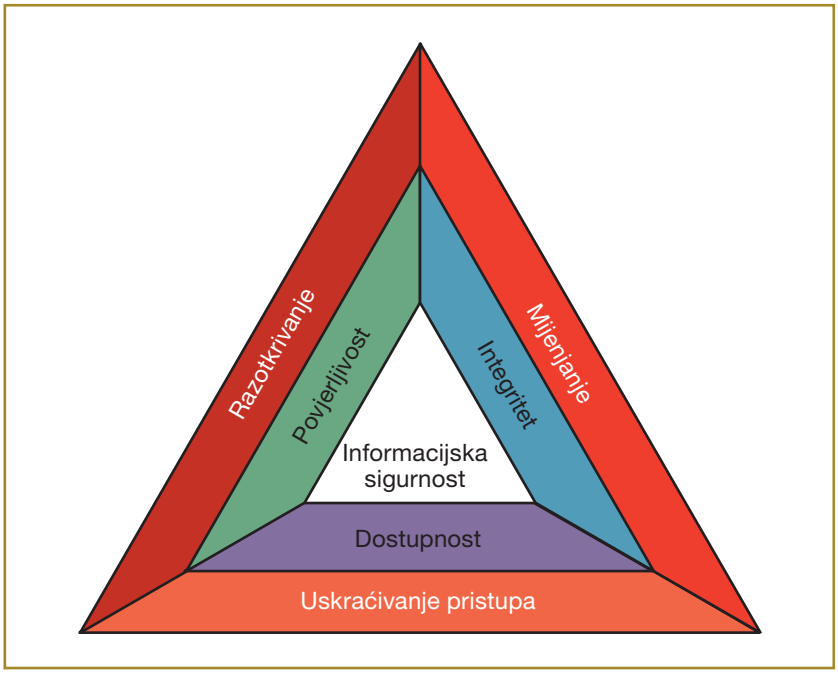
Mnogi su sigurnosni mehanizmi koji čuvaju integritet pohranjenih informacija identični onima koji čuvaju i povjerljivost informacija. Tako postavljanje ispravnih kontrola pristupa koje neovlaštenim korisnicima zabranjuju pristup pohranjenim podacima automatski čuva i njihov integritet budući da neovlašteni korisnici ne mogu niti pristupiti datotekama. Osim sprječavanja pristupa, u svrhu osiguravanja integriteta mogu se upotrebljavati posebni matematički algoritmi koji omogućuju digitalno potpisivanje podataka. Datoteka koja je digitalno potpisana dobiva dodatak koji služi za provjeru integriteta datoteke. Ako je datoteka promijenjena, pomoću digitalnog je potpisa moguće otkriti je li došlo do promjene (ali ne i što je promijenjeno).

Treća ključna komponenta sigurnosti informacijskog sustava je **dostupnost** (engl. *availability*). Cilj dostupnosti vrlo je jednostavan i jasan – legitimni i autorizirani korisnici trebaju uvijek moći pristupiti potrebnim informacijama, servisima i poslužiteljima. Iako se dostupnost sustava često zanemaruje, pa čak i ne smatra sigurnosnim zahtjevom, u današnjem svijetu elektroničkog poslovanja

dostupnost predstavlja iznimno bitan faktor. U slučaju nedostupnosti informacija njihov integritet i povjerljivost mogu biti praktički nebitni. Dostupnost informacija može se narušiti na različite načine – od najjednostavnijeg primjera kada neovlašteni korisnik pobriše datoteke s ključnim podacima te ih tako napravi nedostupnima pa do slučaja sofisticiranih mrežnih napada na poslužitelje (engl. *Denial of Service* – DoS napadi).

1.2. DAD trokut

Prethodno opisan CIA trokut predstavlja model sigurnosti informacijskog sustava. S druge strane, napadači imaju svoj vlastiti model koji predstavlja inverz CIA trokuta te se naziva DAD (engl. *Disclosure, Alteration, Denial*) trokut. DAD trokut predstavlja tri napada na tri osnovne sigurnosne komponente CIA trokuta.



Slika 1.2. DAD trokut

Prva komponenta DAD trokuta je **razotkrivanje** informacija (engl. *Disclosure*). Cilj razotkrivanja informacije je zaobilaženje sigurnosnih kontrola koje osiguravaju povjerljivost informacija. Na primjer, napadač koji ostvaruje pristup informacijskom sustavu te je u stanju pročitati povjerljive informacije s njega (npr.

brojeve kreditnih kartica) razotkrivanjem ovih informacija izravno napada komponentu povjerljivosti informacijskog sustava.

Mijenjanje informacija (engl. *Alteration*) predstavlja drugu komponentu napada na informacijske sustave. U slučaju neautorizirane promjene pohranjenih podataka narušen je njihov integritet – na taj način mijenjanje informacija predstavlja inverz očuvanja integriteta navedenih informacija. Primjer narušavanja integriteta predstavlja napadač koji je ostvario pristup informacijskom sustavu te promijenio pohranjene informacije, npr. brojeve računa u nekom ugovoru.

Konačno, uskraćivanje pristupa (engl. *Denial*), zadnja komponenta modela napada na sigurnost informacijskog sustava legitimnim korisnicima onemogućuje pristup informacijskom sustavu ili informacijama pohranjenim u njemu.

1.3. Zakonska podloga

Značaj sigurnosti informacija i informacijskih sustava prepoznale su različite krovne organizacije, ali i sami zakonodavci. U tu svrhu postoji čitav niz zakonskih i regulatornih zahtjeva koji obvezuju pojedine tvrtke i državna tijela na različite korake koje je potrebno poduzeti u svrhu postizanja određene razine sigurnosti informacija i samih informacijskih sustava.

1.3.1. Zakonski zahtjevi

U Republici Hrvatskoj su u ovom trenutku na snazi tri zakona vezana za sigurnost podataka.

- Zakon o sigurnosno-obavještajnom sustavu. Ovaj zakon donesen je sredinom 2006. i njime su postavljeni temelji za rad Sigurnosno-obavještajne agencije (SOA) i Vojne sigurnosno-obavještajne agencije (VSOA). Osim navedenih tijela, ovaj zakon postavlja i temelje za rad Zavoda za sigurnost informacijskih sustava (ZSIS) koji je središnje državno tijelo za obavljanje poslova u tehničkim područjima informacijske sigurnosti državnih tijela Republike Hrvatske.
- Konačno, ovaj zakon omogućuje i rad Operativno-tehničkog centra za nadzor telekomunikacija (OTC) koji upravlja tajnim nadzorom telekomunikacijskih usluga, djelatnosti i prometa.
- Zakon o informacijskoj sigurnosti usredotočen je na klasificirane i neklasificirane podatke državne uprave, odnosno državnih tijela, tijela jedinica lokalne i područne samouprave i na pravne osobe s javnim ovlastima.

- Potporu ovom zakonu daje Zakon o tajnosti podataka koji definira pojam klasificiranih i neklasificiranih podataka, stupnjeve tajnosti te pristup klasificiranim i neklasificiranim podacima, njihovu zaštitu i nadzor. Klasificirani podaci predstavljaju one podatke koji su dobili određenu klasifikaciju (npr. „Povjerljivo“, „Tajno“ ili „Vrlo tajno“).

Konačno, Zakon o zaštiti osobnih podataka definira što su osobni podaci te postavlja zahtjeve za zaštitu tih podataka. Zakon definira osobni podatak kao svaku informaciju kojom je fizička osoba identificirana ili kojom se fizička osoba može identificirati, izravno ili neizravno. Drugim riječima, bilo koja organizacija koja prikuplja podatke koji spadaju u kategoriju osobnih podataka (npr. tvrtka koja skuplja podatke o zaposlenicima) mora iste prikladno i zaštititi.

1.3.2. Regulatorni zahtjevi

Osim navedenih zakona, pojedine su tvrtke (npr. banke i druge financijske institucije) obvezne poštovati i neke druge regulatorne zahtjeve. Tako je udruženje najkorištenijih tvrtki kreditnih kartica (Master Card, Visa i neke druge) osnovalo PCI Security Standards konzorcij. Cilj ovog konzorcija jest promicanje sigurnosti informacijskih sustava svih tvrtki koje obrađuju kreditne kartice. Ključni standard koji je ovaj konzorcij izdao naziva se PCI DSS (engl. *Payment Card Industry Data Security Standard*), a definira niz zahtjeva koje moraju ispuniti sve tvrtke koje rukuju kreditnim karticama. Riječ je o tehničkim i operacijskim zahtjevima koji omogućuju uspostavljanje snažnog procesa rukovanja kreditnim karticama uključujući sprječavanje, otkrivanje i odgovarajuće korake koje je potrebno poduzeti u slučaju sigurnosnog incidenta.

1.3.3. ISO 27001 standard

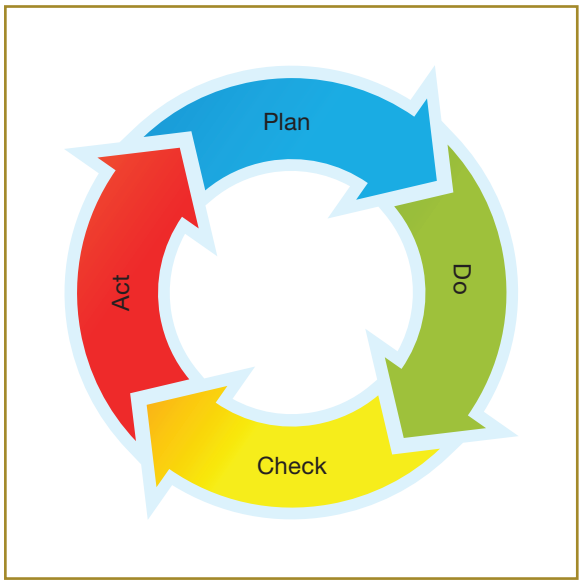
Značaj sigurnosti informacijskog sustava zakonodavci su, ali i velike tvrtke, prepoznali još davno. Tako je 1995. britanska organizacija za standardizaciju (engl. *BSI British Standards*) objavila standard na području informacijske sigurnosti BS7799. BS7799 je prvi standard koji je objedinio dobre prakse na području informacijske sigurnosti te na taj način vladinim tijelima, ali i tvrtkama pokušao dati preporuke za dizajn, implementaciju i održavanje sustava za upravljanje informacijskom sigurnošću.

BS7799 su poslije preuzele ISO (engl. *International Organization for Standardization*) i IEC (engl. *International Electrotechnical Commission*) standardizacijske organizacije te objavile seriju standarda pod oznakom ISO 27000.

Riječ je o šest standarda u ovoj seriji koji se danas smatraju glavnim smjernicama svim tvrtkama koje žele uvesti sustav upravljanja informacijskom sigurnošću. Ovdje je potrebno napomenuti da je riječ o generičkim standardima, čija je namjena osigurati okvir, odnosno smjernice za upravljanje informacijskom sigurnošću te kao takvi ne sadržavaju tehničke već samo generičke informacije. Drugim riječima, u standardima nema uputa za implementaciju pojedinih sigurnosnih kontrola poput npr. vatrozida ili sustava za sprječavanje neovlaštenih aktivnosti, već su dane generičke smjernice koje opisuju kontrole koje je potrebno implementirati poput npr. upotrebe zasebnog korisničkog imena i zaporke za svakog korisnika i slično.

ISO 27001 standard iz ISO 27000 serije standarda predstavlja središnji standard kojim je definirana uspostava sustava za upravljanje informacijskom sigurnošću. Sustav za upravljanje informacijskom sigurnošću jednostavno predstavlja sve tehničke, operacijske i administrativne mjere koje tvrtka treba provesti kako bi informacijsku sigurnost svog sustava dovela na zadovoljavajuću razinu. Generičnost standarda osigurava njegovu primjenjivost na tvrtke svih veličina – od onih malih s nekoliko zaposlenika pa sve do najvećih, multinacionalnih kompanija. Osim navedenih smjernica, ISO 27001 standard omogućuje i certifikaciju tvrtki, odnosno pojedinih poslovnih procesa. Tako je npr. moguće jedan poslovni proces (npr. internetsko bankarstvo ili web-trgovinu) certificirati po ISO 27001 standardu, čime se krajnjim korisnicima pokazuje da je tvrtka uložila u sigurnost svog poslovanja.

ISO 27001 stavlja naglas na kontinuirani, ciklički proces upravljanja informacijskom sigurnošću. Ovo je vjerojatno najbitnija poruka ovog standarda, a to je da informacijska sigurnost nije nešto što se jednom implementira i zaboravi, već još jedan proces koji je potrebno redovito nadzirati, analizirati nove prijetnje koje su se pojavile te na odgovarajući način osvježiti sve sigurnosne kontrole koje su implementirane tijekom jednog od prijašnjih ciklusa. Kod ISO 27001 standarda ovaj se ciklički proces naziva PDCA ciklus: Plan, Do, Check i Act. Plan (Planiranje) znači da je sve aktivnosti potrebno unaprijed planirati: Do (Implementacija) se odnosi na fazu implementacije planiranih aktivnosti, Check (Provjera) uključuje mjerenje i nadzor implementiranih kontrola te Act (Djelovanje) predstavlja osvježavanje procesa u svrhu unaprjeđenja. PDCA ciklus prikazan je na sljedećoj slici.



Slika 1.3. ISO 27001 PDCA ciklus

1.4. Osnovna terminologija

U području informacijske sigurnosti upotrebljava se određena terminologija koja će biti definirana na početku te će se njome koristiti tijekom cijelog ovog priručnika. Terminologija kojom se služi u nastavku općenito je prihvaćena te se upotrebljava i u standardima poput ISO 27001.

Sigurnosna ranjivost predstavlja grešku ili nedostatak u bilo kojem elementu informacijskog sustava. Sigurnosna ranjivost najčešće se realizira u vidu tehničkog nedostatka određene sigurnosne kontrole (npr. sigurnosna ranjivost u nekom mrežnom servisu), no ona može biti i dio sigurnosnih procedura, implementacije pa čak i samog dizajna određenog informacijskog sustava. Iskorištavanje sigurnosne ranjivosti može rezultirati kompromitiranjem informacijskog sustava. Primjer sigurnosne ranjivosti u implementaciji je otvoren mrežni port na vatrozidu koji je trebao biti blokiran dok je primjer sigurnosne ranjivosti u dizajnu prenošenje korisničkog imena i zaporke preko računalne mreže u čistom tekstualnom obliku (a ne enkriptirano), što svakom napadaču koji ima pristup na računalnu mrežu omogućuje saznavanje ovih osjetljivih podataka. Sigurnosne ranjivosti mogu postojati i u hardveru: npr. osjetljivost pojedinog poslužitelja na prašinu predstavlja sigurnosnu ranjivost koja može u konačnici utjecati na dostupnost poslužitelja, ako isti prestane s radom.

Prijetnja je mogućnost **izvora prijetnje** da iskoristi (slučajno ili namjerno) određenu sigurnosnu ranjivost. Razlikujemo dva izvora prijetnje: to su nakana i metoda namjernog iskorištavanja određene sigurnosne ranjivosti te situacija i metoda slučajnog iskorištavanja određene sigurnosne ranjivosti. Klasično se izvori prijetnji dijele u sljedeće tri grupe:

- prirodne prijetnje poput potresa, poplava i drugih elementarnih nepogoda koje izravno utječu na cjelokupni informacijski sustav, i to obično na njegovu dostupnost;
- ljudske prijetnje s kojima se najčešće susrećemo u vidu neovlaštenih korisnika, odnosno napadača čiji je cilj narušavanje jedne ili više komponenti CIA trokuta. Ljudske prijetnje mogu biti i legitimni zaposlenici koji nenamjerno ili slučajno naruše CIA trokut;
- prijetnje okoline koje također utječu uglavnom na dostupnost informacijskog sustava (ali mogu neizravno narušiti i npr. integritet podataka). U prijetnje okoline spadaju ispadi električne energije, zagađenje, kemikalije i slično.

Exploit predstavlja specijalnu kategoriju programa, grupe podataka ili naredbi koje u tom obliku (dakle u obliku *exploita*) iskorištavaju specifičnu sigurnosnu ranjivost. Na primjer, ako je neki mrežni servis ranjiv na određenu kategoriju sigurnosnih ranjivosti, kao što je detaljnije opisano u poglavlju 4.1, napadač koji upotrebljava specijalizirani *exploit* za tu ranjivost može preko računalne mreže preuzeti potpunu kontrolu nad ranjivim servisom, a ovisno o drugim uvjetima čak i nad cijelim poslužiteljem na kojem je ranjivi servis pokrenut.

1.4.1. Sigurnosni rizik

Konačno, **sigurnosni rizik** predstavlja mogućnost ostvarivanja određene prijetnje, odnosno štete koja nastaje od te prijetnje. Upravljanje sigurnosnim rizikom predstavlja proces identificiranja i procjenjivanja sigurnosnog rizika s ciljem smanjivanja sigurnosnog rizika na prihvatljivu razinu implementiranjem određenih sigurnosnih kontrola. Primjerice, ako je rizik od napadača koji ima pristup računalnoj mreži preko koje se prenosi korisničko ime i zaporka u čistom tekstualnom obliku velik, s ciljem smanjivanja ovog sigurnosnog rizika može se implementirati sigurnosna kontrola u vidu enkripcije.

Ovdje je potrebno napomenuti da ne postoji stopostotna informacijska sigurnost. U svakom informacijskom sustavu uvijek će postojati određena količina sigurnosnog rizika koji proizlazi iz sigurnosnih prijetnji i ranjivosti koje će uvijek postojati – nemoguće je napraviti sustav koji neće imati niti jednu sigurnosnu

ranjivost. Pri određivanju sigurnosnih kontrola koje će se implementirati potrebno je uzeti u obzir i njihov utjecaj na sam informacijski sustav te njihov cjelokupni trošak. Naime, čest je slučaj da se u svrhu eliminiranja sigurnosnog rizika implementiraju sigurnosne kontrole koje sam sustav čine neupotrebljivim (npr. korisnik se mora prijaviti na sustav nakon svake provedene akcije) ili su same sigurnosne kontrole toliko skupe da su neisplative. Sigurnosni rizik koji preostaje u informacijskom sustavu nakon što su sve sigurnosne kontrole implementirane naziva se rezidualnim sigurnosnim rizikom.

Sigurnosni rizik računa se za sve parove prijetnji i ranjivosti pri čemu se još u obzir uzimaju i parametri poput vjerojatnosti iskorištavanja pojedine sigurnosne ranjivosti od strane pripadajuće prijetnje za čiji se par računaju sigurnosni rizik, posljedica (odnosno izravno šteta) u slučaju da se ta prijetnja realizira odnosno događaj i dogodi te možebitne sigurnosne kontrole koje su implementirane i koje mogu smanjiti vjerojatnost da će se prijetnja ostvariti ili rezultirajuću štetu.

Prijetnja koja se razmatra primjerice može biti potres na dijelu gdje je izgrađena sistemska sala s poslužiteljima. Za potrese i prirodne nepogode postoje dobri podaci koji prikazuju njihovu vjerojatnost te se mogu iskoristiti i pri računanju sigurnosnog rizika. Posljedica, odnosno šteta u ovom je primjeru također jasna (uništena sistemska sala i svi poslužitelji, zaustavljeni poslovni procesi koji ovise o ovim resursima). Konačno, implementirana sigurnosna kontrola u ovom primjeru može biti druga (redundantna) sistemska sala u nekom drugom gradu čime se također izravno utječe na sigurnosni rizik.

Sigurnosni rizik se, dakle, računa za svaki pojedini resurs (npr. poslužitelj, ali to može biti i softver ili čak i zaposlenik koji je nužan za obavljanje pojedinog poslovnog procesa). Sigurnosni rizik je funkcija sljedećih pet varijabli: vrijednosti resursa (engl. *Asset Value* – AV), ranjivosti resursa (engl. *Vulnerability* – V), prijetnji koje mogu iskoristiti te ranjivosti (engl. *Threat* – T), vjerojatnosti ostvarivanja prijetnji (engl. *Probability* – P) i na kraju posljedice (štete) u slučaju ostvarivanja prijetnji (engl. *Impact* – I):

$$R = f(AV, V, T, P, I)$$

Sama procedura analize sigurnosnog rizika može biti vrlo kompleksna i prelazi okvire ovog priručnika pa će biti navedene samo dvije metode koje se najčešće upotrebljavaju.

- Kvantitativna metoda analize sigurnosnog rizika pokušava pridijeliti stvarne i smislene brojke svim elementima izračuna sigurnosnog rizika poput pojedinih resursa u tvrtki, utjecaja ostvarivanja prijetnje na samo poslovanje, vjerojatnosti ostvarivanja prijetnje i slično.
  Kvantitativne metode danas se rjeđe upotrebljavaju jer je neke parametre vrlo teško kvantificirati. Primjerice, ako se statistički gleda broj sigurnosnih ranjivosti jednog servisa kao što je npr. Apache HTTP servis u odnosu na drugi servis kao što je npr. Microsoft IIS, može se zaključiti da je jedan servis manje rizičan od drugog, što ne mora odgovarati situaciji u stvarnom svijetu.
- Kvalitativna metoda analize sigurnosnog rizika ne koristi se apsolutnim vrijednostima parametara, kao što je to bio slučaj kod prethodne metode, već se utjecaj navedenih parametara na rizik pokušava kvalitativno odrediti različitim metodama procjene. U ovom se slučaju procjena, dakle, obično temelji na iskustvu osobe koja provodi procjenu.

Iako se procjena provodi kvalitativno, u konačnici se i ovi rezultati prikazuju brojčano kako bi se upotrebom jednostavnih matematičkih formula mogao izračunati konačni rizik.

Nakon završene procjene rizika dobiva se čitav niz brojčanih vrijednosti koje odgovaraju svim parovima ranjivost/prijetnja za pojedini resurs. Ovaj se postupak provodi za sve identificirane resurse te u konačnici određuje koji je minimalni rizik u skladu s time prihvatljiv, što znači da se na njemu neće ništa raditi, a koji je rizik neprihvatljiv te se moraju implementirati sigurnosne kontrole kako bi se on umanjio ili eliminirao.

Prihvatljiv rizik može biti teoretska mogućnost da će netko probiti enkripcijski algoritam koji se upotrebljava za zaštitu osjetljivih podataka. Iako ova mogućnost postoji, njezina je vjerojatnost iznimno niska (osim ako nije riječ o tajnim službama države) pa će i rezultirajući sigurnosni rizik biti nizak, odnosno niži od unaprijed postavljenog praga.

S druge strane, činjenica da su zaporke korisnika pohranjene u čistom tekstualnom obliku čini ih vrlo osjetljivim na razne napade te je rezultirajući sigurnosni rizik automatski visok. Ovakav će rizik biti neprihvatljiv te je nužno implementirati neke sigurnosne kontrole kako bi se rizik umanjio, npr. enkripcijom zaporki korisnika.

Na sljedećoj je slici prikazan primjer matrice kojom se može služiti za određivanje praga sigurnosnog rizika. Prema slici, sve procjene sigurnosnog rizika koje rezultiraju vrijednostima označenim u žutim ili višim kvadratima moraju biti tretirane na odgovarajući način, dok se ostali sigurnosni rizici (plavi i zeleni kvadrati) mogu prihvatiti.

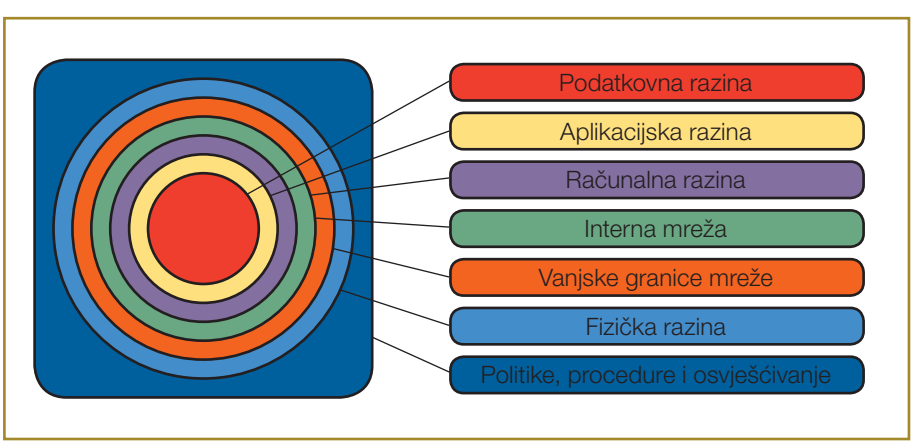
Slika 1.4. Primjer matrice rizika

1.5. Višeslojni strateški model sigurnosti

Višeslojni strateški model sigurnosti (engl. *Defense In Depth*) predstavlja ideju zaštite informacijskog sustava od bilo koje maliciozne aktivnosti pomoću višestrukih razina zaštite. Riječ je o modelu koji zapravo kopira model kojim se koristila američka vojska, a koji je na područje informacijske sigurnosti prva prenijela američka organizacija NSA (*National Security Agency*).

Ideja modela višeslojne strateške sigurnosti zapravo je relativno jednostavna: implementacijom višestrukih razina (metoda) zaštite u konačnici se podiže razina sigurnosti cjelokupnog informacijskog sustava. Ako napadač uspije zaobići jedan sigurnosni mehanizam, druge bi razine (odnosno drugi sigurnosni mehanizmi) trebali i dalje uspješno štititi kompletan informacijski sustav. Na primjer, oslanjanje samo na vatrozid u svrhu zaštite informacijskog sustava nije preporučena zaštita zbog mnogih načina na koje napadač može zaobići vatrozid: od modificiranja napada na mrežnoj razini koje će vatrozid propustiti do mogućnosti provođenja drugih napada koji u potpunosti zaobilaze vatrozid kao npr. socijalni inženjering gdje napadač navodi legitimnog korisnika da na USB memorijskom mediju prenese neki program izravno na računalo koje štiti vatrozid. Jasno je da je u ovakvim slučajevima potrebno implementirati zaštitu na više razina: vatrozid će štititi poslužitelj od nepoželjnog mrežnog prometa, dok je napad s USB memorijskim medijima potrebno onemogućiti na drugim razinama, između ostalog i ispravnim sigurnosnim politikama – dokumentima koji će svim zaposlenicima dati do znanja da ne smiju unositi USB memorijske medije u štićeni prostor ili ih spajati na kritične poslužitelje.

Višeslojni strateški model sigurnosti obično se zasniva na sedam razina sigurnosti koje su prikazane na sljedećoj slici.



Slika 1.5. Višeslojni strateški model sigurnosti

Jasno je da niti jedna razina zaštite, a niti sve u kombinaciji ne mogu pružiti stopostotnu zaštitu (koja ne postoji), no cilj je pojedinih razina zaštite spriječiti provođenje napada na određenim razinama, kao što je objašnjeno za pojedinu razinu u nastavku.

- Podatkovna razina (engl. *Data*) nastoji zaštititi ultimativni cilj napadača, a to su informacije odnosno podaci pohranjeni u informacijskom sustavu bez obzira na metodu njihovog pohranjivanja (je li riječ o podacima pohranjenim u datotekama, bazama podataka ili nečemu trećem). Podatke na ovoj razini štitimo postavljanjem ispravnih prava pristupa te, ovisno o kritičnosti podataka, enkripcijom i upotrebom snažnih zaporki.
- Aplikacijska razina (engl. *Application*) obuhvaća sve aplikacije, odnosno servise koji omogućuju pristup podacima. U velikom broju slučajeva napadači pokušavaju iskoristiti upravo sigurnosne ranjivosti u različitim aplikacijama da bi došli do podataka.
- Računalna razina (engl. *Host*) obuhvaća računala odnosno poslužitelje na kojima su pokrenute aplikacije i pohranjeni podaci. Zaštita računalne razine obično je vrlo kompleksna te se sastoji od zaštite od malicioznih programa (npr. antivirusnim programima), instalacije sigurnosnih zaporki i različitih drugih sigurnosnih mehanizama poput ispravne i minimalne konfiguracije poslužitelja kako bi se što je moguće više smanjile mogućnosti napada.
- Interna mreža (engl. *Internal network*) predstavlja internu računalnu mrežu tvrtke na kojoj se nalaze poslužitelji kojima se pristupa. Ova se razina vrlo često kombinira sa sljedećom razinom budući da je u oba slučaja riječ o kontroli mrežnog pristupa poslužiteljima.
- Vanjska granica računalne mreže (engl. *Perimeter*) predstavlja točku kontrole i razdvajanja interne računalne mreže te javne mreže, odnosno interneta. Vanjska granica računalne mreže obično se implementira u vidu vatrozida koji kontrolira dolazni mrežni promet (dakle, mrežni promet koji dolazi s interneta), ali i odlazni mrežni promet koji generiraju radne stanice i poslužitelji na internoj računalnoj mreži tvrtke.
- Fizička razina (engl. *Physical*) kontrolira fizički pristup računalnoj mreži i poslužiteljima. Poslužitelji su obično locirani u sigurnom fizičkom prostoru (sistemskoj sali) gdje je moguće kontrolirati pristup, budući da sam fizički pristup poslužiteljima napadaču znatno olakšava postupak kompromitacije poslužitelja. Općenito, napade provođene na fizičkoj razini najteže je spriječiti i kontrolirati te je zbog toga ovdje potrebno implementirati de-

taljne sigurnosne kontrole kako bi se fizički pristup poslužiteljima što je moguće više ograničio samo na autorizirane osobe.

- Sigurnosne politike, procedure i sigurnosno osvješćivanje korisnika (engl. *Security policies, procedures and security awareness*). Riječ je o organizacijskoj razini gdje se definiranjem sigurnosnih strategija i njihovim dokumentiranjem jasno definiraju metode zaštite i obveze korisnika informacijskog sustava. Bez ispravno definirane sigurnosne strategije tvrtke te pravilnog dokumentiranja nemoguće je postići zadovoljavajuću razinu zaštite informacijskog sustava.

Osim same zaštite informacijskog sustava, implementiranje višeslojnog strateškog modela sigurnosti tvrtki omogućuje i pravodobnu detekciju neovlaštenih aktivnosti te iniciranje reakcije na iste u svrhu sprječavanja budućih napada.

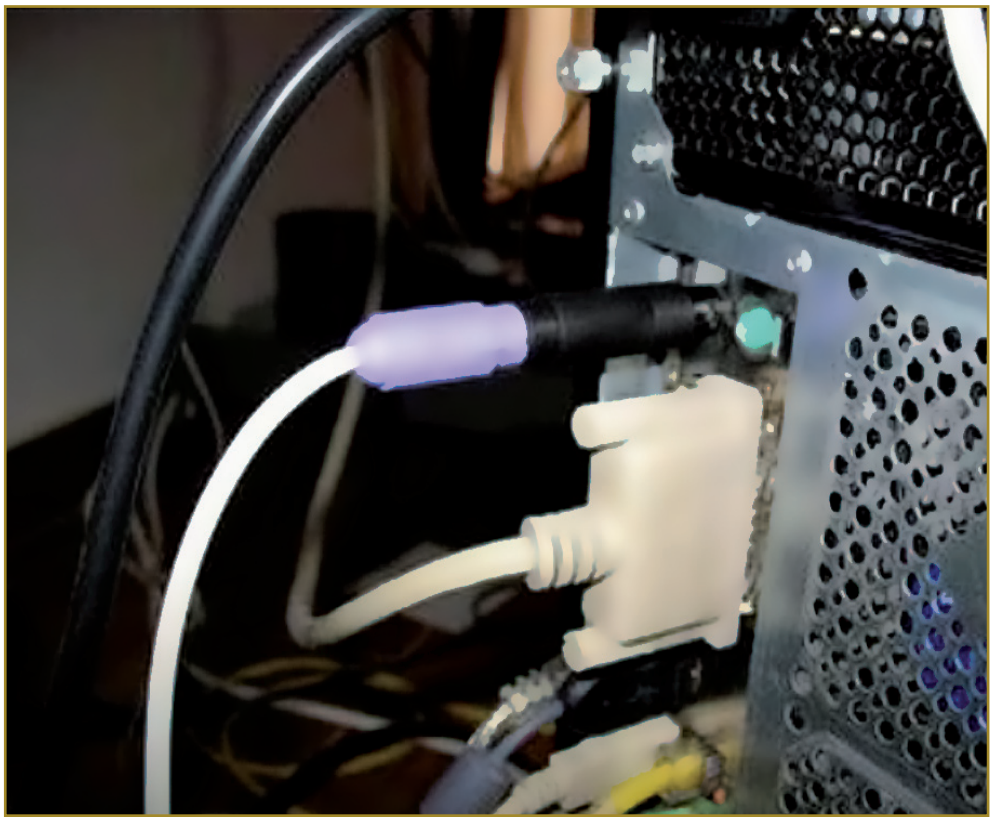
2. Fizički sloj

Kao što je već rečeno, fizička sigurnost poslužitelja predstavlja veliki izazov za svaku tvrtku. Glavni razlog ovom izazovu je činjenica da napadač koji ima fizički pristup poslužitelju može pokrenuti čitav niz napada koje je vrlo teško ili čak nemoguće u potpunosti spriječiti.

Upravo je zbog ovog izazova i Microsoft u svojih 10 zakona informacijske sigurnosti na treće mjesto stavio zakon koji kaže sljedeće: „Ako napadač ima neograničen fizički pristup vašem računalu, to više nije vaše računalo“. Naime, napadač koji ima fizički pristup računalu ima praktički neograničene mogućnosti napada: od onih jednostavnih fizičkih napada na razini uništavanja računalne opreme (koji dovode do uskraćivanja pristupa podacima) pa do pokušaja podizanja računala s alternativnog medija poput CD-ROM-a na kojem napadač može imati vlastiti operacijski sustav koji je u stanju jednostavno zaobići sigurnosne kontrole pristupa izvornog operacijskog sustava.

Povijesno gledajući, 60-ih i 70-ih godina prošlog stoljeća implementacija sigurnosnih kontrola fizičkog pristupa bila je relativno jednostavna budući da je većina računala bila tzv. mainframe-ovi koji su jednostavno morali biti smješteni u specijalnim prostorijama pod stalnim nadzorom. Današnja osobna računala međutim pomiču ovu granicu na drugu krajnost – naime, ako zaposlenik tvrtke pohrani osjetljive informacije na svoje prijenosno računalo koje nakon toga odnese iz tvrtke i izgubi, jasno je da zaštita od fizičkog pristupa ovako pohranjenim informacijama više nije jednostavna.

Čak i u slučajevima kada zaposlenici ne pohranjuju osjetljive informacije na prijenosna računala već se samo koriste računalima u tvrtci, osiguravanje samo autoriziranog fizičkog pristupa nije jednostavno. I u ovakvom je slučaju napadaču dovoljan samo trenutak fizičkog pristupa nekom računalu da bi postavio opremu za napad. Klasičan primjer ovakvog napada je npr. postavljanje hardverskog uređaja između tipkovnice i računala koji bilježi sve pritisnute tipke (engl. *hardware keylogger*). Primjer ovakvog uređaja dan je na sljedećoj slici gdje se može vidjeti da je riječ o praktički neprimjetnim uređajima koje prosječan zaposlenik neće nikada vidjeti, a na računalo ih može postaviti bilo tko, pa čak i čistač u tvrtci kojeg je možda vanjski napadač naveo da postavi ovakav uređaj za njega te ga pokupi nakon određenog vremenskog intervala (npr. tjedan dana), nakon što je uređaj prikupio dovoljno podataka.



Slika 2.1. Hardverski uređaj za bilježenje pritisnutih tipki (dodan na kabel od tipkovnice)

Još jedan primjer fizičkog napada na zaposlenike predstavlja unošenje neautoriziranih medija u tvrtku. Napad koji je još prošlog stoljeća proveo poznati kriminalac pod imenom Kevin Mitnick (više o njemu u poglavlju 7.2.1) sastojao se od postavljanja CD-ROM ili USB memorijskog medija na određena mjesta u tvrtci gdje se zaposlenici često okupljaju (npr. kod uređaja za kavu ili u zahodu). Napadač zatim na medij napiše da sadržava npr. datoteku s revizijom plaća zaposlenika za tekuću godinu te ga ostavi na vidljivom mjestu. Na mediju zapravo nije nikakva datoteka s revizijom plaća već specijalno napravljeni maliciozni program (obično je riječ o trojanskom konju, o čemu je više detalja dano u poglavlju 4.3.3) koji će napadaču omogućiti kontrolu inficiranog računala preko računalne mreže, kada je jednom maliciozni program pokrenut. Napadač sada samo čeka da jedan od naivnih zaposlenika vidi postavljeni medij, odnese ga do svog računala u tvrtci i pokrene maliciozni program.

Kao što je moguće vidjeti iz navedenih primjera, implementiranje sigurnosnih kontrola fizičkog pristupa može biti vrlo kompleksno te nikako (kao niti jedna

druga razina sigurnosnih kontrola) ne štiti informacijski sustav stopostotno, ali može onemogućiti određene kategorije napada, poput izravnog pristupa diskovnom sustavu poslužitelja.

Kao što je navedeno u nastavku poglavlja, fizička se zaštita danas uvelike temelji na samim informacijskim sustavima te je zbog brzog razvoja na ovom području vrlo često moguće vidjeti sustave za fizičku zaštitu koji su zapravo integrirani sa samim informacijskim sustavom. Tako se npr. sustav za videonadzor može koristiti internom računalnom mrežom za prijenos videozapisa u stvarnom vremenu na središnji poslužitelj. Na ovaj se način postižu znatne uštede, ali je istodobno potrebno naglasiti i da sam sustav za fizičku zaštitu u ovakvom slučaju postaje izložen većem sigurnosnom riziku budući da neovlašteni korisnik koji ima pristup računalnoj mreži sada može omesti čak i rad sustava za videonadzor.

Cilj implementacije sigurnosnih kontrola je generiranje kombinacije sigurnosnih mehanizama koje će omogućiti sljedeće:

- odvraćanje napadača (engl. *Deter*). Ispravno postavljene sigurnosne kontrole fizičke zaštite poput npr. visokih ograda s bodljikavom žicom, sustava za videonadzor i slično odvratit će napadača od napada;
- otežavanje napada (engl. *Delay*). Uspostavljanjem višestrukih sigurnosnih mehanizama, kao što je i riječ kod višeslojnog strateškog modela sigurnosti, napadi će biti otežani te će njihovo provođenje dulje trajati;
- omogućavanje detekcije napada (engl. *Detection*). Sigurnosne kontrole poput uspostavljanja sustava videonadzora omogućit će detektiranje potencijalnih napadača i njihovo pravodobno sprječavanje.

2.1. Građevine, okoliš i prostorije

Tijekom uspostavljanja sigurnosnih mehanizama fizičke zaštite posebnu je pažnju potrebno posvetiti samim građevinama u kojima će biti postavljene sistemske sale, ali i prostorijama u kojima će raditi zaposlenici.

Ovisno o tome kolika je kritičnost podataka koji se čuvaju u informacijskom sustavu bit će potrebno na odgovarajući način planirati i raspored te upotrebu prostorija. Na primjer, kod malih je tvrtki uobičajeno da se svi poslužitelji nalaze u odvojenoj prostoriji kojoj je pristup ograničen samo administratorima sustava. Kod velikih tvrtki te onih koje pohranjuju posebno osjetljive podatke obično se implementiraju cjelovite sistemske sale koje, osim pažljivo postavljenih sigurnosnih mehanizama koji kontroliraju pristup sistemskoj sali, sadržavaju i druge

sigurnosne mehanizme poput sustava za detekciju i gašenje požara, UPS sustava koji omogućuju neprekidni izvor napajanja i slično.

Pri odabiru građevine u kojoj će se nalaziti sistemska sala potrebno je obratiti pažnju na okoliš i lokaciju građevine te infrastrukturu koja će biti potrebna za samu implementaciju. Pri procjeni i odabiru lokacije potrebno je obratiti pažnju na sljedeće elemente:

- infrastrukturu u vidu napajanja sustava. Sistemske sale obično udomljavaju veliki broj poslužitelja te potreba za električnom energijom kod ovakvih građevina može biti iznimno velika. Zato je potrebno dobro procijeniti i provjeriti može li infrastruktura podržati potrebe sistemske sale. Jednako tako, potrebno je provjeriti i može li telekomunikacijska infrastruktura zadovoljiti sve zahtjeve;
- mogući opasni materijali u okolini građevine. Lokacije građevine u čijem se susjedstvu nalaze druge tvrtke ili tvornice koje rade s potencijalno opasnim materijalima (npr. kemijske tvornice) obično se smatraju nepovoljnima za izgradnju sistemskih sala;
- drugi vanjski elementi poput fizičke povezanosti (ceste, blizina zrakoplovnih luka i slično) te čak i povijesni podaci poput frekvencije potresa i rizika od poplava područja mogu utjecati na lokaciju izgradnje zgrade u kojoj će se nalaziti sistemska sala.

Prethodno navedeni elementi procjenjuju se prije odabira lokacije na kojoj će se nalaziti zgrada odnosno sistemska sala, ali i uredi sa zaposlenicima. U svrhu implementiranja daljnjih sigurnosnih kontrola koje ograničavaju fizički pristup upotrebljavaju se sljedeće metode sigurnosnih zaštita:

- ograde – osnovna zaštita pristupa štićenom prostoru tvrtke. Vrlo se često preporučuje da ograde budu što je moguće vidljivije kako bi odvratile napadača, no zbog prihvatljivosti okoline većina tvrtki sadi zelenilo oko njih.

  Ograde se općenito dijele na tri kategorije, prema namjeni odvraćanja pojedinih napadača:

  - ograde do 1,2 m visine upotrebljavaju se za označavanje štićenog prostora odnosno prostora na kojem tvrtka želi kontrolirati fizički pristup. Ovakve ograde ne predstavljaju znatno fizičko ograničenje te će zaustaviti samo slučajnog provalnika;
  - ograde do 1,8 m visine koje danas najčešće služe za štićenje fizičkog pristupa prostoru te će odvratiti većinu provalnika;

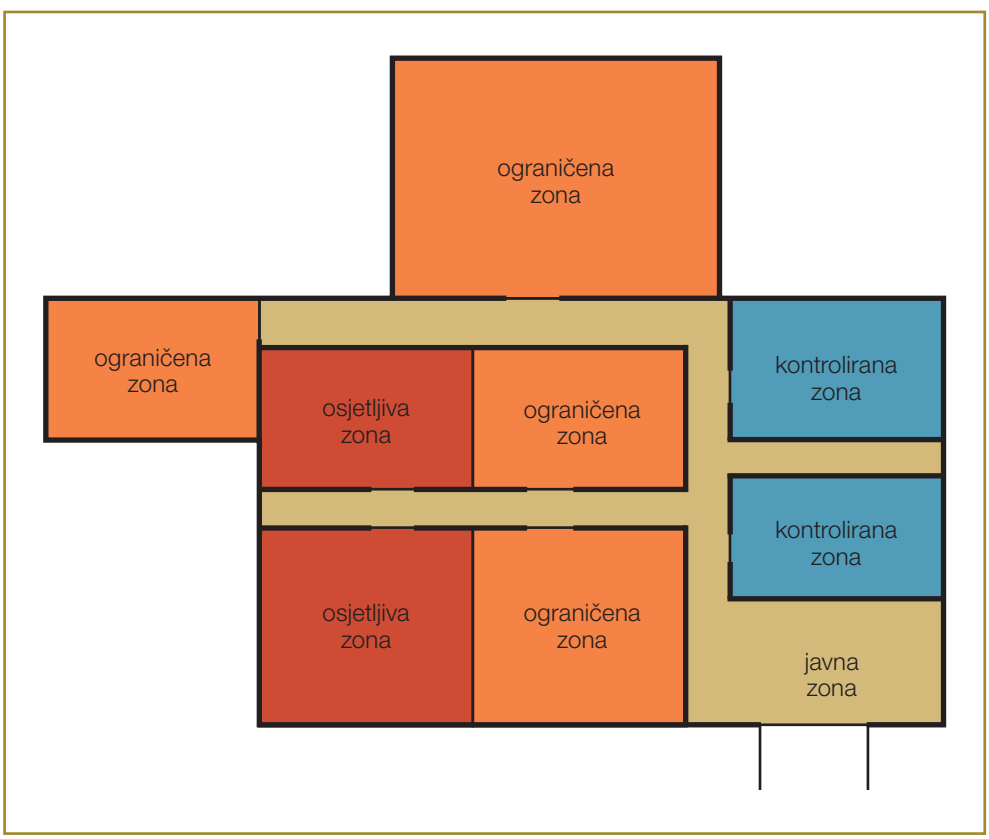
- ograde visine do 2,4 m koje mogu na vrhu imati i bodljikavu žicu upotrebljavaju se kod najkritičnijih okruženja gdje je potrebno spriječiti ili otežati pristup što je moguće većem broju napadača odnosno provalnika.

Slika 2.2. Ograda kod jedne od budućih Googlovih sistemskih sala

Pri samoj konstrukciji zgrada posebnu je pažnju potrebno posvetiti i materijalu od kojeg se grade zidovi i bilo kakve druge pregrade unutar zgrade. Ovdje je posebno potrebno naglasiti zapaljivost odnosno otpornost na vatru te otpornost na fizičke napade materijala koji se upotrebljavaju za izgradnju.

Uz otpornost materijala pažnju je potrebno posvetiti i njihovoj nosivosti. Naime, kod izgradnje sistemskih sala posebno je bitan proračun nosivosti budući da će u te sale biti postavljeni poslužitelji čija kombinirana težina nerijetko može iznositi i nekoliko tona, pogotovo kada je riječ o velikim ormarima u sistemskim salama.

Pri ograničavanju pristupa pažnju je potrebno posvetiti i vratima, odnosno zonama pristupa kojima pojedini zaposlenici smiju pristupiti. Danas se klasično za kontrolu pristupa rabe beskontaktne kartice kojima je moguće vrlo precizno kontrolirati razinu pristup zaposlenika. Tako je zgradu moguće podijeliti na višestruke zone koje će definirati gdje zaposlenici, ali i posjetitelji imaju pravo pristupa. Klasičan model definira nekoliko zona poput kontrolirane, ograničene, javne i osjetljive zone, kao što je prikazano na sljedećoj slici.



Slika 2.3. Ograničavanje fizičkog pristupa prema zonama

Zaposlenici na ovaj način mogu dobiti kartice za pristup odnosno ključeve samo onih prostorija kojima trebaju pristupati zbog svojih poslovnih obaveza dok im je pristup svim drugim prostorijama automatski onemogućen.

Konačno, posebnu je pažnju potrebno posvetiti i prozorima. Prozori moraju biti ispravno postavljeni te trebaju imati okvire odgovarajuće snage i po potrebi odgovarajuću glazuru. Glazura se dodaje pri izradi prozora te može mijenjati fizičke značajke stakla: kaljeno staklo tako ima bolje fizičke značajke te se teže

razbija. Osim kaljenog stakla, moguće je, ovisno o osjetljivosti prostora, rabiti i ožičeno staklo ili laminirane prozore koji se sastoje od dvije staklene površine između kojih se postavlja vrlo tanki plastični film. Ovakvi laminirani prozori iznimno su korisni budući da ograničavaju prodiranje topline u prostorije te onemogućuju vanjske prolaznike u gledanju unutrašnjosti zgrade.

2.2. Alarmni sustavi i videonadzor

Osnovna uloga alarmnih sustava i videonadzora je u detekciji neovlaštenih korisnika odnosno provalnika, budući je riječ o kontroli fizičkog pristupa.

Alarmni sustavi su danas u vrlo širokoj uporabi i nerijetko služe i za zaštitu privatnih prostora. Riječ je o sustavima koji u najvećem broju slučajeva detektiraju pokušaje provale ili kretanja u ograničenim (čuvanim) prostorijama te se upotrebljavaju samo za uzbunjivanje. Današnji sustavi tako vrlo često dojavljuju alarm na centralno mjesto putem telefonske linije ili SMS poruke. Ovakvi sustavi trebaju imati redundantno napajanje u vidu baterije kako bi se osiguralo da pri nestanku napajanja električkom energijom i dalje ispravno funkcioniraju.

Osim alarmnih sustava čiji je zadatak detekcija provalnika, alarmni se sustavi rabe i za detekciju požara te detekciju povišene vlage, odnosno poplava prostora.

2.2.1. Sustavi za detekciju provalnika

Alarmni sustavi za detekciju provalnika koriste se čitavim nizom mehanizama koji su najčešće temeljeni na detekciji prisustva u prostoriji:

- fotoelektrički i alarmni sustavi za detekciju provalnika sastoje se od odašiljača i prijamnika. Odašiljač emitira svjetlosne zrake koje prijamnik cijelo vrijeme prima. Kod većine ovakvih sustava navedene svjetlosne zrake su zapravo golom oku nevidljive. Ako provalnik prekine svjetlosnu zraku, prijamnik to detektira i može pokrenuti alarm;
- toplinski (IR – engl. *Infra-red*) detektori rade na načelu mjerenja topline, odnosno na detekciji promjene topline;

elektromehanički alarmni sustavi najčešće se implementiraju na vrata ili na prozore. Pri otvaranju se prekida strujni krug nakon čega se automatski pokreće alarm.

2.2.2. Sustavi za detekciju požara

Sustavi za detekciju požara funkcioniraju na detekciji dima ili povišene temperature. Detektori dima uglavnom funkcioniraju na načelu optičke detekcije (fotoelektrični detektori) ili putem fizičkih procesa poput ionizacije.

Fotoelektrički detektori dima, slično istoimenim alarmnim sustavima, također funkcioniraju na načelu odašiljača i prijamnika. Odašiljač emitira svjetlosnu zraku koju prijamnik prima. U slučaju požara dim sprječava primanje svjetlosne zrake od strane prijamnika na osnovi čega se može dignuti alarm.

Slika 2.4. Fotoelektrički detektor dima

Ionizacijski detektori dima koriste se iznimno malim količinama radioaktivnih elemenata koji prolaze kroz poseban prostor u detektoru te na taj način čine zatvoreni strujni krug. U slučaju požara dim ulazi u prostor u detektoru te prekida proces ionizacije, a samim time i strujni krug. Nedostatak ionizacijskih detektora dima je ponekad čak i prevelika osjetljivost budući da mogu detektirati iznimno male količine dima.

2.2.3. Videonadzor

Videonadzor se danas vrlo često upotrebljava s drugim sigurnosnim mehanizmima fizičkog nadzora te predstavlja veliku pomoć pri detekciji provalnika, naročito čuvarima fizičkog prostora.

CCTV sustav (engl. *Closed-circuit TV*), odnosno sustav za videonadzor, sastoji se od kamera, odašiljača, prijamnika, sustava za snimanje te samog nadzornog ekrana. U klasičnim se implementacijama snimka sustava za videonadzor pohranjuje i čuva određeno vrijeme na trakama ili drugim medijima. Ovakav način pohranjivanja snimki omogućuje kasnije pregledavanje ako se ustanovi da je došlo do neovlaštenih aktivnosti odnosno provala.

Budući da CCTV sustavi obično imaju višestruke kamere, videozapis koji ove kamere odašilju prima središnji videosustav koji rabi multiplekser u svrhu prikaza slike na jednom ekranu, gdje se istodobno može vidjeti slika sa svih kamera, što čuvaru olakšava nadzor.

Slika se obično prenosi upotrebom koaksijalnih kabela koji predstavljaju odvojenu mrežu (od tuda i naziv engl. *Closed Circuit*). U kritičnim bi CCTV sustavima ova koaksijalna mreža trebala biti otporna na fizičke napade, odnosno pokušaj pristupa trebalo bi detektirati kako bi se napadač onemogućio u mijenjanju videozapisa.

Na sljedećoj je slici prikazan tipični CCTV sustav.

Slika 2.5. Tipičan CCTV sustav

Današnji sustavi za videonadzor često su opremljeni posebnim kamerama koje se mogu upotrebljavati i u apsolutnom mraku te se temelje na detekciji topline. Osim toga, videozapis se u stvarnom vremenu šalje na poslužitelj na kojem je pokrenut poseban softver koji omogućuje automatsku detekciju pokreta na osnovi analize statične slike te odgovarajuće uzbunjivanje, ako je pokret detektiran. Na taj se način znatno smanjuje opterećenje čuvara zgrade koji više ne mora konstantno nadgledati kontrolni ekran videosustava.

Kao što se može vidjeti na slici 2.5, klasični CCTV sustavi koriste se zasebnom, koaksijalnom mrežom za slanje videozapisa. U svrhu smanjivanja troškova postavljanja CCTV sustava danas se vrlo često ide na integraciju s informacijskim sustavom u tvrtkama te se postavljaju moderne kamere koje mogu slati videozapis preko IP mreža. Ovakve kamere često su opremljene megapikselnim senzorima te mogu biti spojene čak i izravno na diskovne sustave što omogućuje pohranjivanje snimljenih videozapisa.

No ovakav pristup ima očigledne nedostatke, a to je da dijeli isti medij za prijenos videozapisa kao i lokalna računalna mreža. Drugim riječima, napadač koji ima pristup na lokalnu računalnu mrežu teoretski može modificirati videozapis ili, što je puno jednostavnije, onemogućiti prijenos videozapisa s kamere na središnji sustav provođenjem napada uskraćivanja računalnih resursa na lokalnu računalnu mrežu. Ovisno o mrežnoj opremi koja se upotrebljava, moguće je dati veći prioritet mrežnom prometu generiranom od IP kamera, no u kritičnim se okruženjima još uvijek preporučuje fizičko odvajanje interne računalne mreže tvrtke i one kojom se koriste IP kamere.

2.3. Biometrijski sustavi

Biometrijski sustavi predstavljaju sigurnosne mehanizme koji ograničavaju pristup na osnovi fizičkih (biometrijskih) karakteristika korisnika. Ovakvi se mehanizmi najčešće upotrebljavaju za ograničavanje pristupa prostorijama, no danas su učestali biometrijski sustavi koji funkcioniraju na provjeri otiska prsta i koji se mogu naći čak i na prijenosnim računalima.

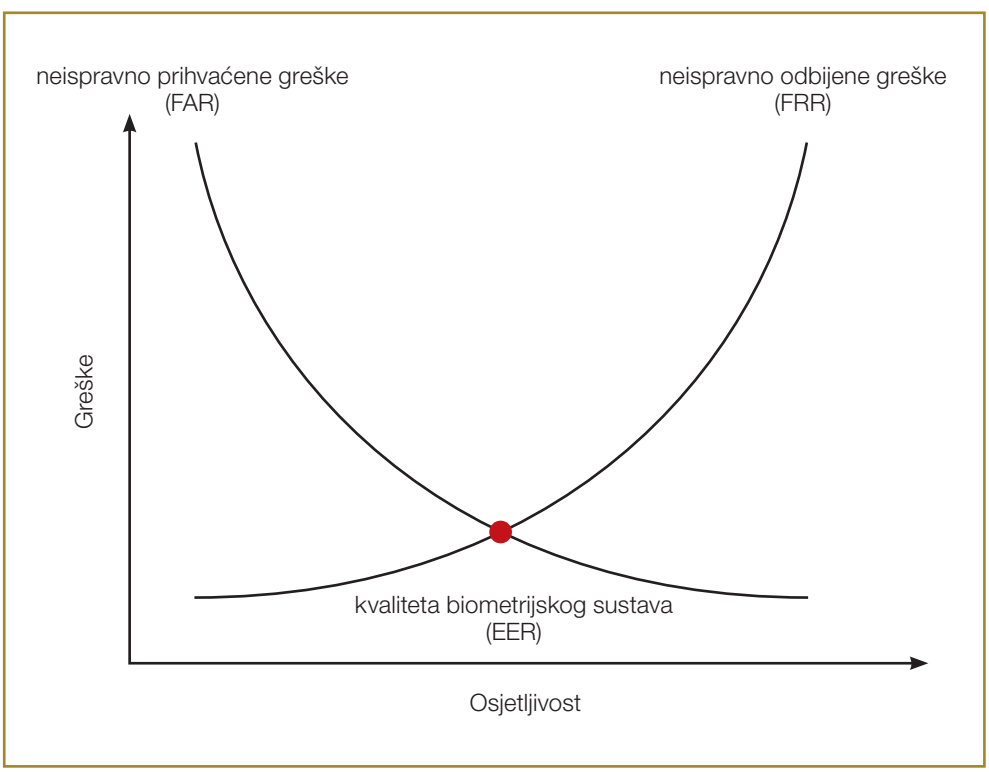
Iako se možda na prvi pogled čini da su ovi sustavi najtočniji budući da se fizičke karakteristike korisnika ne mogu (ili mogu vrlo teško) mijenjati, to nažalost nije činjenica zbog nesavršenosti biometrijskih sustava. Naime, ovi sustavi ovise o velikom broju čimbenika kao što su okolina u kojoj se nalaze, ali i stanje samog korisnika kojeg se pokušava identificirati, a u nekim slučajevima čak i o njegovom zdravstvenom stanju.

Tako npr. ako se za identifikaciju korisnika upotrebljava biometrijska metoda analize glasa, a korisnik je promukao ili zbog drugih razloga izgubio glas, jasno je da dolazi do problema u radu biometrijskog sustava te da se legitimni, autorizirani korisnik ne može autenticirati.

Greške u radu biometrijskih sustava dijele se na dva tipa:

- greške tipa 1 predstavljaju tzv. neispravno odbijene greške (engl. *false rejection rate* – FRR) kada je sustav neispravno ili neuspješno identificirao legitimnog korisnika te mu onemogućio pristup sustavu ili prostoriji;
- greške tipa 2 predstavljaju tzv. neispravno prihvaćene greške (engl. *false acceptance rate* – FAR) kada je sustav prihvatio identitet korisnika koji bi trebao biti odbijen, odnosno identificirao je nelegitimnog ili neautoriziranog korisnika kao nekog drugog korisnika, koji ima pravo pristupa sustavu ili prostoriji.

U svrhu definiranja kvalitete biometrijskog sustava upotrebljava se vrijednost pod nazivom CER (engl. *Crossover Error Rate*) ili EER (engl. *Equal Error Rate*). CER je na sljedećem grafu prikazan kao sredina na kojoj se sijeku FRR i FAR.

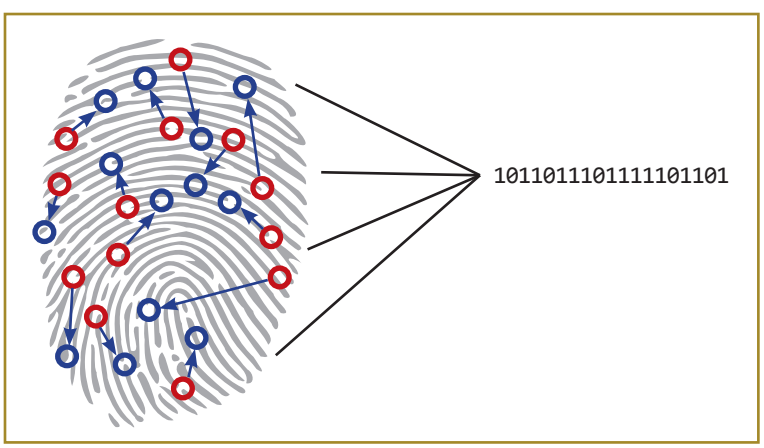


Slika 2.6. Graf koji prikazuje kvalitetu biometrijskog sustava (CER)

Općenito gledano, što je CER (odnosno EER) manji, to je biometrijski sustav kvalitetniji i manje je podložan greškama.

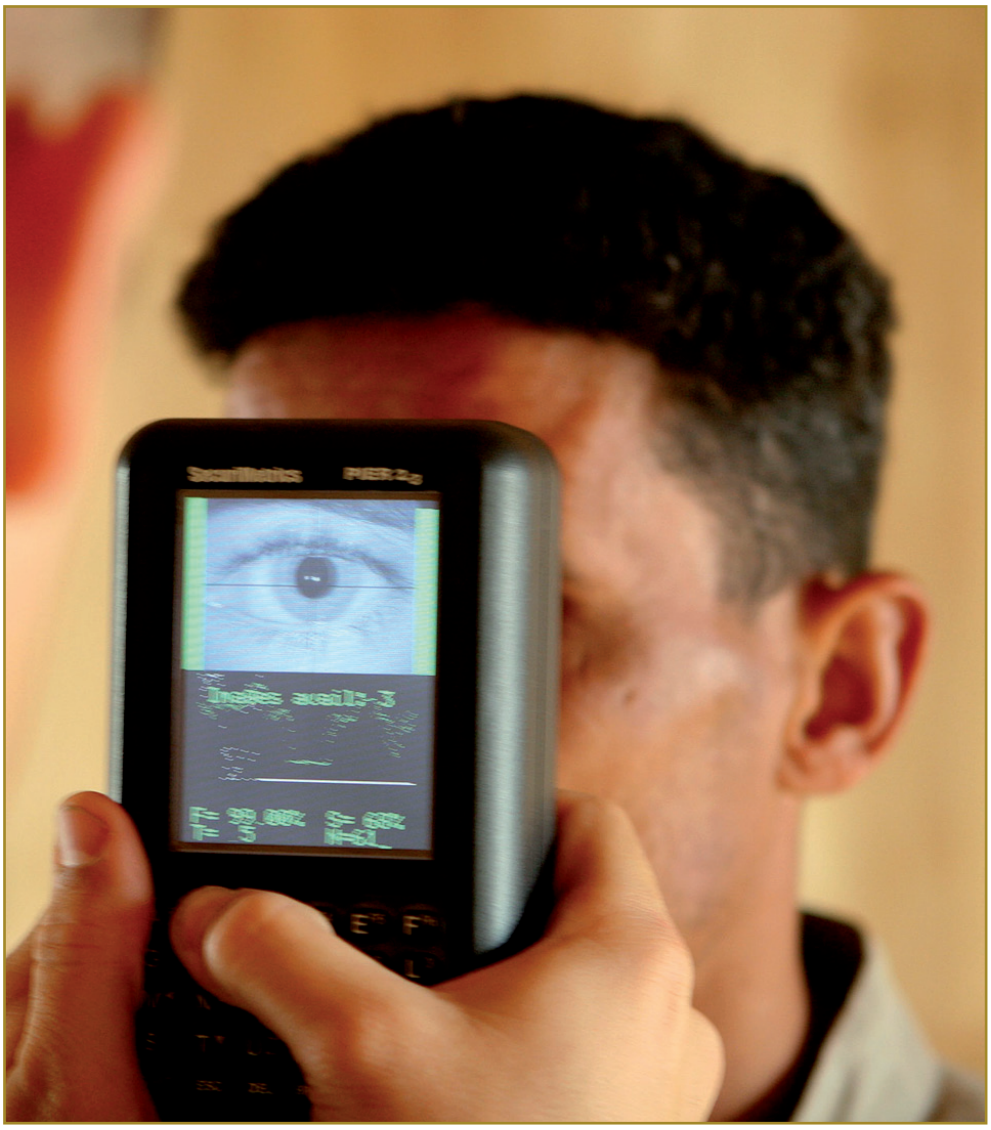
Biometrijski sustavi mogu se koristiti čitavim nizom fizičkih karakteristika korisnika koji se identificiraju. U nastavku su navedene one najčešće korištene, a ujedno i najprihvatljivije od strane korisnika budući da je potrebno uzeti u obzir da biometrijski sustavi ipak narušavaju privatnost korisnika.

- Otisak prsta. Identifikacija na bazi otiska prsta zasnovana je na činjenici da su udubljenja i izbočine na površini prsta unikatni te da se razlikuju za svakog čovjeka.
  Pri inicijalnoj se konfiguraciji ove brazde čitaju i pohranjuju u bazu identiteta. Kada se korisnik opet identificira, očitane se vrijednosti uspoređuju s onima u bazi.



Slika 2.7. Biometrija na načelu otiska prsta

- Šarenica oka (engl. *iris*). Ova se metoda identifikacije zasniva na načelu uzimanja slika šarenice visoke rezolucije te generiranja matematičkih uzoraka šarenice. Navedeni se matematički uzorci pohranjuju u bazu podataka te jednako prethodnom slučaju uspoređuju pri identifikaciji novog korisnika. Problem identifikacije pomoću šarenice oka jest da nije u stanju razlikovati jednojajčane blizance čija je šarenica oka ista.
- Mrežnica oka (engl. *retina*). Identifikacija je zasnovana na uzimanju uzorka žila koje se nalaze u pozadini oka. Ove žile imaju jedinstven raspored za svakog čovjeka, uključujući i jednojajčane blizance te tako omogućuju preciznu identifikaciju. Uzimanje uzorka temelji se na slanju infracrvenog spektra u oko. Krvna zrnca u pozadini oka apsorbiraju više svjetlosti od ostatka te je na taj način moguće uzeti uzorak mrežnice oka.



Slika 2.8. Biometrija na načelu šarenice oka

- Prepoznavanje uzorka glasa zasniva se na unikatnosti boje glasa i načina izgovaranja riječi. Pri korištenju ovom metodom obično se rabe neke standardne rečenice i riječi koje korisnik koji se identificira izgovara. Uzorak se uzima na osnovi izgovorenog teksta.

Osim navedenih metoda postoje još i brojne druge biometrijske metode poput identifikacije na osnovi uzorka crta lica, pregledavanja vena u dlanu, mirisa, geometrije ruke pa čak i identifikacija temeljene na DNA-u.

2.4. UPS sustavi

Stabilno napajanje električnom energijom ključno je za ispravan rad poslužitelja i mrežne opreme. Danas se u svrhu osiguravanja stabilnog napajanja upotrebljavaju UPS sustavi (engl. *Uninterruptible Power Supply*).

Uloga UPS uređaja nije samo u osiguravanju kontinuiranog napajanja električnom energijom već i u ispravljanju određenih problema u napajanju od kojih se razlikuju sljedeće kategorije:

- gubitak napajanja električnom energijom;
- povišeni napon. Tipično napajanje električnom energijom može davati konstantno viši napon (engl. *Surge*) ili trenutno povišeni napon (engl. *Spike*). Oba događaja mogu uzrokovati probleme u radu elektroničke opreme, pogotovo poslužitelja tako da se UPS uređaji rabe za stabilizaciju napona;
- sniženi napon (engl. *Sag*). Slično prethodnom slučaju može doći do konstantnog ili trenutnog snižavanja napona;
- šum. Zbog druge opreme u blizini može doći do oscilacije u naponu;
- frekvencijska nestabilnost.

Kod svih navedenih slučajeva, UPS uređaji trebaju omogućiti potpunu stabilnost napajanja električnom energijom. Idealan će UPS sustav na izlazu davati konstantan napon i frekvenciju pri bilo kojem od navedenih događaja.

Prema načinu rada UPS uređaji dijele se u sljedeće dvije kategorije:

- pričuvni (engl. *Offline / Standy*) UPS sustavi koji omogućuju samo osnovnu zaštitu od gubitka napajanjem električne energije. Ovi su UPS sustavi spojeni na glavno napajanje te u slučaju gubitka napajanja ili padanja napona ispod neke unaprijed definirane vrijednosti mehanički mijenjaju spoj i osiguravaju napajanje iz baterije. Tipično vrijeme potrebno za uključivanje ovog napajanja iznosi oko 25 ms;
- kontinuirani (engl. *Online*) UPS sustavi rabe se u okruženjima gdje je oprema osjetljiva na promjene napona. Povijesno su ovi UPS sustavi bili prisutni samo u velikim okruženjima, no danas se mogu naći i u manjim tvrtkama. Ovi UPS sustavi kontinuirano napajaju baterije iz izvora izmjenične struje, iz kojih inverter ponovno proizvodi izmjeničnu struju i napaja uređaje. U slučaju nestanka električne energije UPS uređaj samo isključuje glavno napajanje i upotrebljava baterije za daljnje napajanje. Kada se napajanje ponovno uspostavi, ono se rabi i za napajanje uređaja i za punjenje baterije.

Slika 2.9. Kontinuirani UPS sustavi

UPS uređaji su zbog svoje cijene namijenjeni uglavnom zaštiti od kratkotrajnih ispada napajanja električnom energijom – baterije ovih uređaja obično omogućuju napajanje od 30-ak minuta pa do nekoliko sati, ovisno o veličini i snazi (ujedno i cijeni) UPS uređaja. U slučajevima kada dolazi do dugotrajnijih ispada električne energije rabe se generatori struje. Riječ je o uređajima koji se obično koriste dizelskim gorivom ili plinskim turbinama (kada je riječ o većim generatorima) i koji mogu proizvoditi električnu energiju kroz dulja razdoblja.

2.5. Kontrola medija

Fizička kontrola medija također predstavlja jedan bitan proces koji je, osim ispravnog provođenja, nužno i na odgovarajući način dokumentirati u različitim sigurnosnim politikama i pravilnicima. Mediji predstavljaju bilo koje sredstvo na kojem se nalaze osjetljive informacije. Danas su to najčešće papir i elektronički mediji poput CD/DVD medija, USB memorijskih medija, magnetskih traka i drugih.

Tijekom radnog života ovih medija s njima je potrebno ispravno rukovati u smislu fizičke pohrane. Jasno je da papirnati mediji koji sadržavaju osjetljive podatke moraju biti odgovarajuće pohranjeni kako bi se spriječile različite prijetnje na ove medije poput krađe, nepažljivog rukovanja odnosno uništenja, neovlaštenog pristupanja i drugih prijetnji.

Jednako tako, navedeni su mediji osjetljivi na utjecaje iz okoline – magnetski mediji poput traka naročito su osjetljivi na izlaganje toplini i svjetlosti. Osim toga,

trake su osjetljive i na prljavštinu i prašinu što zahtijeva njihovo adekvatno pohranjivanje, pogotovo kada je riječ o medijima na kojima su pohranjeni podaci koji moraju dulje biti sačuvani , kao što su npr. pričuvni podaci.

Osjetljivi papirnati mediji, ali i drugi elektronički mediji u svrhu zaštite od uništavanja i neovlaštenog pristupa najčešće se pohranjuju u sefove. Sefovi se razlikuju prema otpornosti na krađu, ali i otpornosti na požar i druge prijetnje iz okoline poput vlage, vode i prašine.

2.5.1. Uništenje medija

Nakon što mediji više nisu potrebni, a ako su sadržavali osjetljive podatke, potrebno ih je primjereno odložiti odnosno, ako to sigurnosne procedure zahtijevaju, uništiti.

Papirnati se mediji obično uništavaju specijaliziranim rezačima papira. Rezači papira dijele se prema veličini i obliku izrezanih komada papira. Postoji čitav niz različitih rezača papira od kojih su najčešći rezači koji režu papir na trake (engl. *Strip-cut*) te rezači koji upotrebljavaju dva okomita noža te na taj način proizvode tzv. konfete.

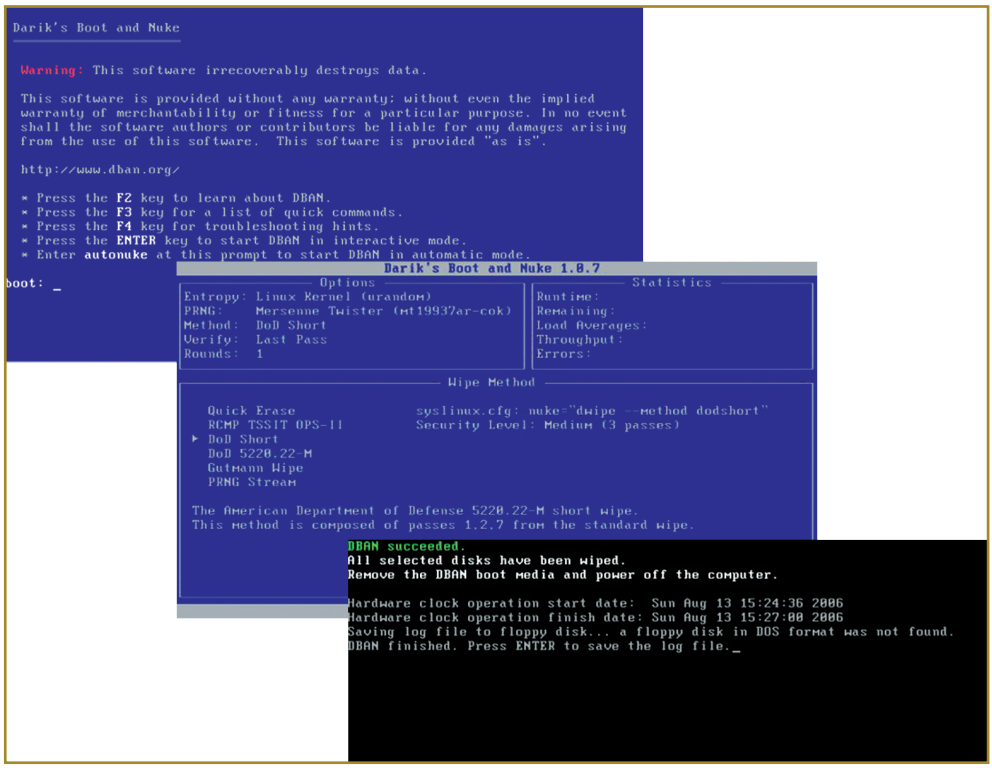
Rezači papira obično se mogu upotrebljavati i za rezanje CD/DVD medija, koji se u ovom slučaju režu na trake.

Slika 2.10. Ručni uništavač dokumenata

Za uništavanje elektroničkih medija također postoje specijalizirani uređaji. Riječ je o uređajima koji omogućuju demagnetiziranje i namijenjeni su uništavanju podataka s tvrdih diskova.

Slika 2.11. Uređaj za uništavanje elektromagnetskih uređaja

Budući da ovakvo uništavanje tvrdih diskova sa sobom povlači i trošak po tvrtku jer su tvrdi diskovi trajno uništeni, ovisno o osjetljivosti podataka koji su bili pohranjeni na tvrdom disku u većini je slučajeva dovoljno prepisivanje tvrdog diska određeni broj puta. DBAN (engl. *Darik's Boot And Nuke*) besplatna distribucija dostupna na http://www.dban.org/ omogućuje brisanje tvrdog diska korištenjem različitih metoda i brojeva brisanja. Metode se sastoje od pisanja slučajnog niza znakova na tvrdi disk određeni broj puta, i to tako da se svi sektori na tvrdom disku prepišu, što znači da ovaj postupak kod velikih tvrdih diskova može dugo trajati. DBAN omogućuje veliki broj metoda brisanja od kojih se najčešće upotrebljava tzv. DoD Short metoda brisanja koja zadovoljava zahtjeve američkog ministarstva obrane i sastoji se od tri prijelaza preko svih sektora tvrdog diska.



Slika 2.12. Brisanje tvrdog diska korištenjem DBAN distribucije

3. Podatkovni sloj

Informacije pohranjene u informacijskom sustavu (CIA trokut) na bilo koji način predstavljaju ultimativni cilj napadača. Kao što je u prethodnim poglavljima već bilo spomenuto, višeslojni strateški model sigurnosti štiti navedene informacije na različitim razinama, a podatkovni je sloj zadužen za definiranje prava pristupa osjetljivim informacijama na onoj zadnjoj, najvišoj razini.

U ovom slučaju korisnik već ima pristup na lokalni sustav koji pohranjuje podatke, bilo da je riječ o pohranjivanju u datotekama, u bazama podataka ili u nekom drugom načinu pohranjivanja.

Pohranjene informacije u ovom slučaju predstavljaju objekte kojima pristupaju subjekti. Subjekti su najčešće sami korisnici informacijskog sustava, ali oni ne moraju nužno predstavljati ljudski faktor – subjekti mogu biti i drugi servisi ili aplikacije kojima se želi ograničiti pristup podacima, odnosno objektima.

Na podatkovnom se sloju dakle koristimo različitim metodama i sigurnosnim mehanizmima kontrole pristupa da bismo ograničili pristup subjekata objektima, odnosno nametnuli određena pravila sustava.

Postoje tri različita modela definiranja kontrola pristupa podacima: diskrecijski, obvezni i grupni.

- **Diskrecijski** model definiranja kontrola (engl. DAC – *Discretionary Access Control*) omogućuje vlasniku pojedinog resursa (npr. datoteke u kojoj su pohranjene osjetljive informacije) definiranje subjekata kojima se dopušta pristup navedenoj datoteci.
  Ovaj se model naziva diskrecijskim zato što su dozvole datoteke temeljene na diskrecijskim odlukama vlasnika datoteke. Diskrecijski model definiranja kontrola prisutan je na većini današnjih operacijskih sustava, uključujući Windows i Linux operacijske sustave.
  Navedeni model također dobro preslikava potrebe tvrtki budući da je obično vlasnik datoteke taj koji treba definirati tko još u tvrtci ima pravo pristupa.
- **Obvezni** model definiranja kontrola (engl. MAC – *Mandatory Access Control*) ne dopušta toliko slobode u definiranju pravila pristupa pojedinim datotekama poput DAC modela. Kod MAC modela operacijski sustav određuje koja će konačna prava pristupa biti dodijeljena pojedinom objektu te, ako je tako definirano, može promijeniti postavke koje je želio postaviti vlasnik datoteke.

MAC model ne rabi se prečesto kod današnjih operacijskih sustava, a njemu je svojstveno to da se informacije klasificiraju (npr. javne, tajne i vrlo tajne) te se onda prema njihovoj klasifikaciji određuju i prava pristupa. Subjekti su klasificirani jednako kao i datoteke što automatski znači da subjekt s klasifikacijom tajno ne može pristupiti datotekama koje su klasificirane kao vrlo tajne. Ove se oznake daju svim resursima na sustavu, bilo da su to datoteke, direktoriji, subjekti ili nešto treće. Jednako tako, subjekt ne može promijeniti klasifikaciju pojedine datoteke (ne može datoteku koja je označena vrlo tajno promijeniti u tajno), osim ako to vlasnik cijelog sustava nije eksplicitno dopustio.

- Grupni model definiranja kontrola (engl. RBAC – *Role-Based Access Control*) često se naziva i nediskrecijski model definiranja kontrola. Glavna osobina ovog modela definiranja kontrola jest da se one administriraju sa središnjeg mjesta. RBAC model omogućuje definiranje prava pristupa prema grupama u kojima se nalazi subjekt. Grupe može definirati samo vlasnik informacijskog sustava (odnosno administrator koji ima prava definiranja takvih grupa), dok sam subjekt ne može mijenjati grupe u kojima se nalazi.

  Današnji moderni operacijski sustavi poput Windowsa i Linuxa podržavaju i RBAC model definiranja kontrola gdje je korisnike moguće staviti u korisničke grupe te na taj način olakšati upravljanje pravima pristupa datotekama i direktorijima. Primjerice, ako je korisnik zaposlen u kadrovskoj službi, moguće je njegov korisnički račun (koji predstavlja njegov subjekt autenticiran sustavu) dodati u posebnu grupu za kadrovsku službu koja će subjektu automatski dopustiti pristup svim datotekama navedene grupe. Na taj je način moguće vrlo jednostavno mapirati ulogu korisnika u tvrtci s njegovim pravima pristupa. Ako navedeni korisnik promijeni poziciju unutar tvrtke te ode u odjel marketinga, dovoljno je ukloniti njegov korisnički račun iz grupe kadrovske službe te ga staviti u grupu odjela marketinga. Subjekt će automatski izgubiti sva prava pristupa koja je imao s inicijalnom korisničkom grupom te dobiti prava pristupa nove grupe.

Kao što se može vidjeti, današnji operacijski sustavi podržavaju mješavinu DAC i RBAC modela definiranja kontrola. Ova dva modela preuzeta su upravo zato što najviše odgovaraju zahtjevima poslovanja pojedine tvrtke (RBAC), a još uvijek omogućuju precizno definiranje prava pristupa (DAC), ako je isto potrebno.

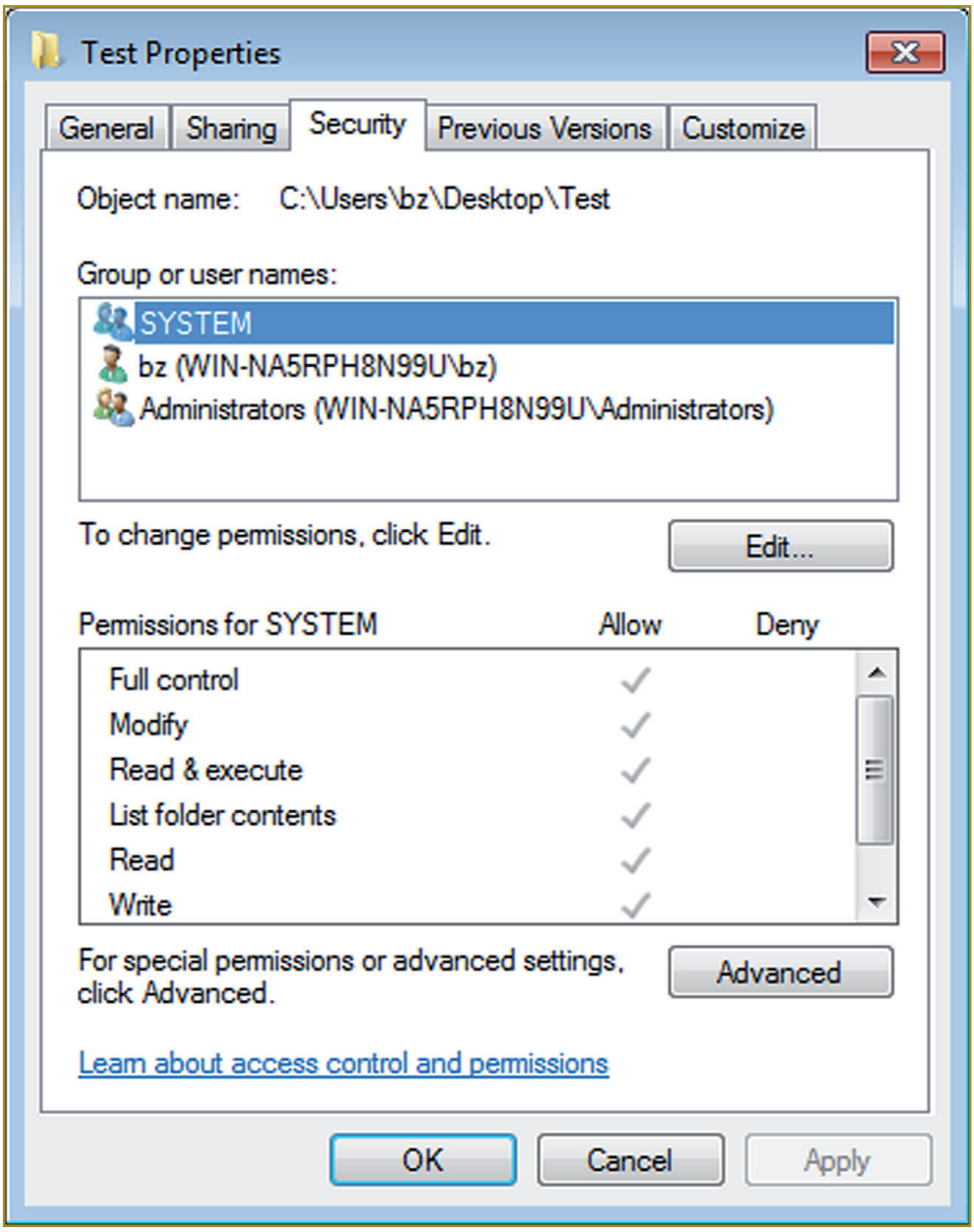
3.1. Prava pristupa na Windows operacijskim sustavima

Windows operacijski sustavi omogućuju vrlo detaljno i precizno određivanje prava pristupa gotovo svim resursima na sustavu. Resurse ovdje najčešće predstavljaju datoteke i direktoriji, no Windowsi omogućuju i definiranje prava pristupa drugim objektima poput Registryja.

Informacije o tome koje su aktivnosti dopuštene kojim subjektima nad svakom objektu Windowsi drže u posebnoj podatkovnoj strukturi koja se naziva pristupna lista, ACL (engl. *Access Control List*). ACL zapravo definira sve subjekte koji imaju pravo pristupa pridijeljenom objektu. Pri zahtjevu bilo kojeg subjekta za pristup određenom objektu, bez obzira na to je li subjekt aktivnost korisnika ili nekog servisa ili aplikacije, Windowsi pregledavaju ACL strukturu pridijeljenu objektu te na temelju podataka u ovoj strukturi određuju ima li objekt pravo pristupa ili ne.

- ACL struktura detaljno definira prava pristupa korisnika koja uključuju sljedeće mogućnosti:
- puna kontrola (engl. *Full control*) subjektu dopušta provođenje bilo koje aktivnosti nad objektom;
- promjena (engl. *Modify*) subjektu dopušta promjenu datoteke;
- čitanje (engl. *Read*) dopušta čitanje sadržaja datoteke;
- čitanje i izvršavanje (engl. *Read & Execute*) dopušta čitanje sadržaja datoteke i njezino izvršavanje, ako je riječ o izvršnoj datoteci;
- pisanje (engl. *Write*) dopušta pisanje po datoteci;
- posebne dozvole (engl. *Special permissions*) omogućuju daljnje detaljno definiranje prava pristupa pojedinom resursu. Posebne dozvole uključuju i neke druge atribute poput npr. prava preuzimanja vlasništva datoteke kojim se može promijeniti vlasnik datoteke. Jednako tako, u slučaju direktorija, posebne dozvole uključuju i dozvolu čitanja sadržaja direktorija kojom je korisnicima moguće dopustiti i zabraniti čitanje sadržaja direktorija. Ovdje je potrebno napomenuti da navedena dozvola nema utjecaja na prava čitanja pojedinih datoteka unutar direktorija.

Na sljedećoj je slici prikazan standardni prozor koji omogućuje definiranje prava pristupa datoteci. Ovaj je prozor moguće otvoriti desnim klikom na datoteku i odabirom kartice Security.

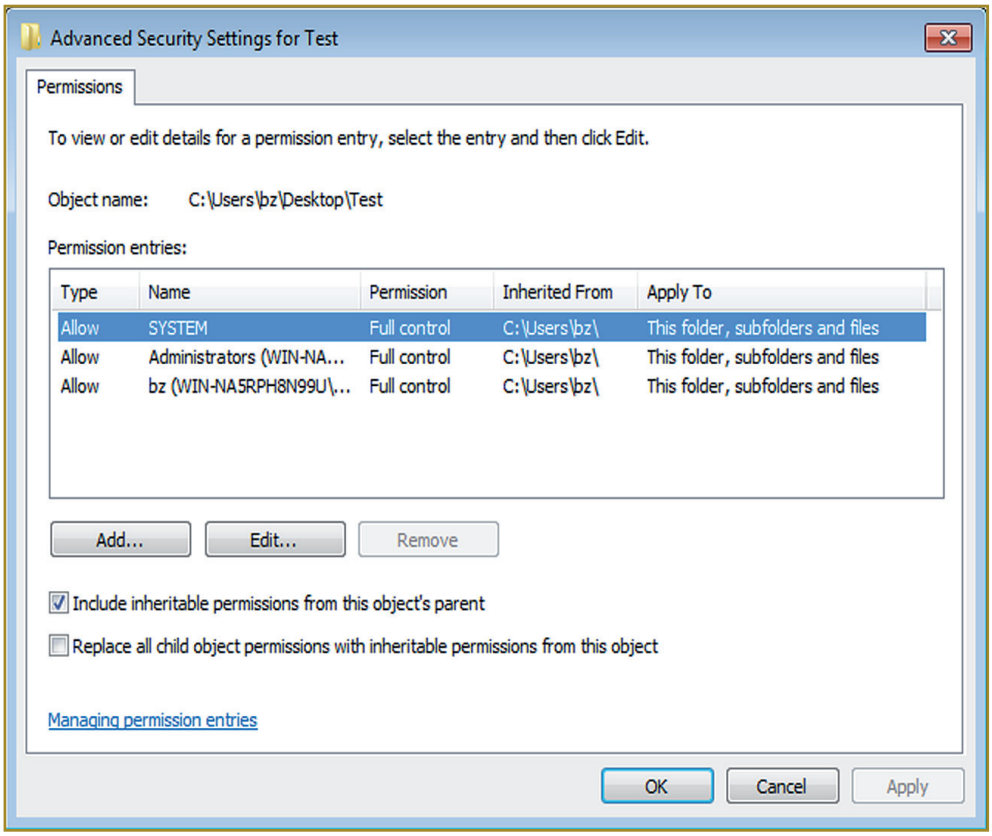


Slika 3.1. Definiranje prava pristupa na Windows operacijskim sustavima

Windows operacijski sustavi također omogućuju dodjeljivanje prava pristupa prema grupama korisnika. Na pojedinoj se datoteci ova prava pristupa dodjeljuju diskrecijski: vlasnik datoteke može npr. određenoj grupi dopustiti čitanje datoteke koje je on vlasnik.

Da bi se donekle omogućio rad sličan RBAC modelu, Windows operacijski sustavi omogućuju definiranje prava pristupa koja se nasljeđuju od hijerarhijski viših (roditeljskih) direktorija. Drugim riječima, vlasnik direktorija može postaviti poseban atribut na direktorij koji će automatski primijeniti određeno pravo pristupa na sve datoteke i direktorije koji su napravljeni unutar tog direktorija. Na ovaj se način može automatski dati pravo pristupa svim datotekama i poddirektorijima određenim grupama te neizravno simulirati RBAC model. Vlasnik resursa koji se nalazi u direktoriju, međutim, može sam odlučiti oko toga želi li naslijediti prava pristupa definirana u hijerarhijski višem direktoriju ili ne.

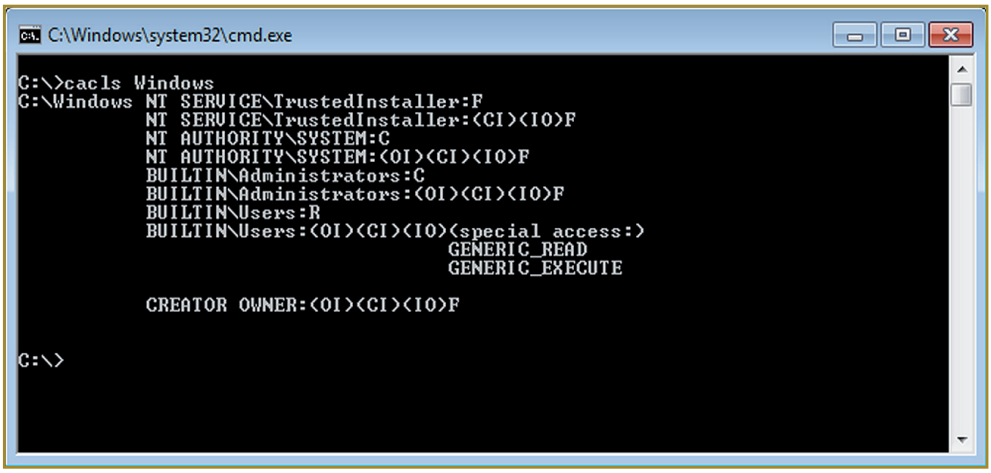
Nasljeđivanje prava pristupa moguće je definirati posebno za svaku datoteku ili direktorij u ekranu za posebne dozvole.



Slika 3.2. Definiranje posebnih dozvola i nasljeđivanja prava pristupa

Kao što je već spomenuto, navedena je prava pristupa moguće je definirati nad gotovo svim resursima na Windows operacijskim sustavima, uključujući i Registry zapise kao i druge strukture kojima se koristi operacijski sustav.

U slučaju potrebe za mijenjanjem prava pristupa na velikom broju datoteka na Windows operacijskim sustavima može se upotrebljavati cacls.exe aplikacija koja omogućuje pregledavanje i promjenu prava pristupa iz naredbenog retka.



Slika 3.3. cacls.exe aplikacija omogućuje pregledavanje i promjenu prava pristupa iz naredbenog retka na Windows operacijskim sustavima

Potrebno je napomenuti da administratorski račun na Windows operacijskim sustavima ne može mijenjati prava pristupa datotekama kojima on sam nije vlasnik. Na taj je način strogo podržan DAC model dodjeljivanja prava pristupa. U slučaju da administrator Windows sustava treba promijeniti prava pristupa datoteci kojoj nije vlasnik, prvo treba preuzeti vlasništvo nad datotekom (što administratorski račun automatski može napraviti, ali jednako će tako biti zabilježeno na sustavu) da bi tek onda bio u stanju modificirati ACL strukturu objekta.

Osim eksplicitnog davanja prava pristupa datotekama, vlasnik datoteke može i eksplicitno zabraniti pristup. Na ovaj se način postiže vrlo precizno i granularno podešavanje prava pristupa gdje vlasnik može npr. jednoj grupi korisnika dopustiti pristup datoteci, a onda pojedinom korisniku koji pripada toj grupi zabraniti pristup, ako je to potrebno.

3.2. Prava pristupa na Linux operacijskim sustavima

Slično Windows operacijskim sustavima, Linux operacijski sustavi također podržavaju DAC model dodjeljivanja prava pristupa pojedinim datotekama i direktorijima. Kao i kod Windows operacijskih sustava, Linux omogućuje dodjeljivanje prava pristupa grupama korisnika.

Tradicionalna Linux prava pristupa dijele pristup prema tri klase:

- korisnik – klasa vlasnika resursa, odnosno datoteke ili direktorija te definira prava pristupa samog korisnika. Korisnik teoretski može maknuti prava pristupa i sebi samom, ali ih uvijek može dodijeliti natrag;
- grupa – klasa grupe kojoj su dodijeljeni datoteka ili direktorij. Prava pristupa dodijeljena ovoj klasi primjenjuju se na sve korisnike koji su članovi navedene grupe. Tradicionalna Linux prava omogućuju definiranje samo jedne grupe, no moderne Linux distribucije omogućuju preciznije definiranje prava pomoću naredbe *chattr*;
- ostali – klasa svih drugih korisnika na sustavu. Za sve druge korisnike koji nisu sam vlasnik datoteke i ne pripadaju primarnoj grupi (ako nisu korištene napredne funkcije naredbe *chattr*) primjenjuju se ova prava pristupa.

Sama prava pristupa na Linux operacijskim sustavima mogu imati samo tri razine:

- čitanje (r) – omogućuje čitanje sadržaja datoteke;
- pisanje (w) – omogućuje pisanje, odnosno modificiranje sadržaja datoteke. Ova dozvola omogućuje i brisanje datoteke;
- izvršavanje (x) – omogućuje izvršavanje datoteke, ako je riječ o izvršnoj datoteci.

Prava pristupa na Linuxu se definiraju prema klasama, i to po tri znaka za svaku klasu (korisnik, grupa, ostali). Ako pojedina klasa ima dozvolu, prikazano je slovo dozvole, u protivnom se rabi znak -. Prava pristupa moguće je pregledati pomoću 'ls –l' naredbe, kao što je prikazano u nastavku:

```
$ ls -l test
-rwxr-xr-- 1 root apache 0 Jun 19 23:49 test
```

Potonji primjer pokazuje prava pristupa datoteci test koja su označena žutom bojom. Može se vidjeti da korisnik, odnosno vlasnik datoteke koji je u ovom slučaju *root* korisnički račun, ima sva prava pristupa (rwx), grupa korisnika (*apache*) ima mogućnost čitanja i izvršavanja datoteke (r-x), bez w dozvole, dok svi ostali imaju mogućnost samo čitanja sadržaja datoteke (r--).

3.3. Kriptografija

Postavljanje ispravnih prava pristupa, kao što je prikazano u prethodnim poglavljima omogućuje definiranje korisničkih računa koji imaju pravo pristupa pojedinim datotekama (i podacima pohranjenim u njima), no još uvijek u potpunosti ne omogućuju zadovoljavanje osnovnog zahtjeva CIA trokuta, a to je povjerljivost. Naime, jednostavan primjer u kojem administrator sustava može pristupiti svim datotekama jasno pokazuje da je potrebna još jedna razina zaštite povjerljivosti pohranjenih informacija.

Slika 3.4. Konfederacijski kriptografski disk koji se koristio u ratu za zaštitu poruka

Kao što je već rečeno u uvodu, potreba za skrivanjem informacija postoji još od samih početaka ljudskog roda. Osiguravanje povjerljivosti sadržaja informacija postiže se skrivanjem samih informacija (dakle skrivanjem njihovog postojanja) ili promjenom njihovog značenja. Znanost koja se bavi proučavanjem metoda

skrivanja informacija naziva se **kriptografija**. Ime ove znanosti dolazi od grčke riječi *kryptos* koja znači sakriti. U terminologiji same kriptografije, proces skrivanja informacije (odnosno onemogućavanja napadača u čitanju izvorne informacije) naziva se **enkripcija**, dok se proces vraćanja skrivene informacije u izvornu naziva **dekripcija**.

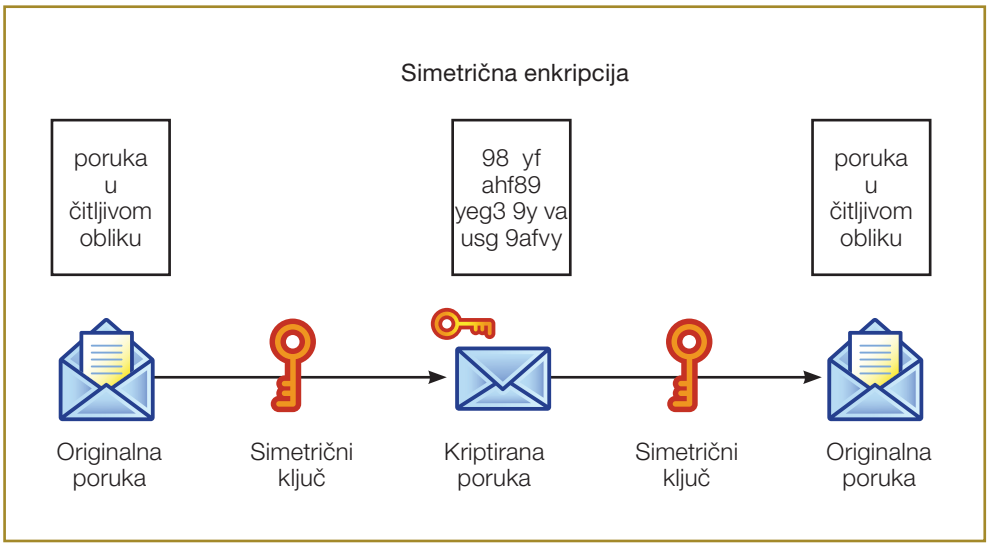
Postupak kojim se informacije enkriptiraju naziva se kriptografskim algoritmom (engl. *cipher*). Informacije mogu biti pohranjene u čistom tekstualnom obliku (engl. *plaintext*) te u enkriptiranom obliku (engl. *ciphertext*). Pri procesu enkripcije ključna su zapravo dva elementa: već navedeni kriptografski algoritam, dakle metoda kojom se informacije prevode iz čistog tekstualnog oblika u enkriptirani oblik te tajni ključ (engl. *secret key*) koji je poznat samo pošiljatelju i primatelju informacija. Snaga enkripcijskog algoritma leži upravo u tom tajnom ključu – čak i ako je napadaču enkripcijski algoritam poznat (dakle, poznata mu je metoda koja se rabi za prevođenje informacija), ako ne zna tajni ključ, dolazak do izvornih informacija trebao bi mu biti nemoguć. Tajni ključ predstavlja varijablu (često se naziva i zaporkom) koja bi idealno trebala biti velika. Duljina ključa mjeri se u bitovima i općenito vrijedi pravilo da što je ključ veći, to je teže probiti navedeni kriptografski algoritam, naravno pod uvjetom da se uspoređuju ekvivalentni kriptografski algoritmi, odnosno da napadač ne može iskoristiti neku drugu ranjivost samog kriptografskog algoritma koja bi omogućila dolazak do izvornih informacija.

Postoje brojni kriptografski algoritmi koji se danas upotrebljavaju u svrhu osiguranja povjerljivosti informacija. Kriptografske algoritme obilježavaju tri glavne osobine:

- sigurnost samog algoritma koja označava koliko je kriptografski algoritam otporan na današnje i buduće napade;
- brzina rada algoritma pokazuje potrebnu procesnu snagu te vrijeme potrebno za enkripciju i dekripciju poruka;
- jednostavnost implementacije omogućuje čak i implementaciju u hardverskim uređajima, što može znatno ubrzati brzinu rada algoritma.
- Prema načinu rada i svrsi kriptografske algoritme dijelimo u sljedeće tri grupe, koje će biti zasebno opisane u nastavku poglavlja:
  - simetrični kriptografski algoritmi (engl. *Symmetric algorithms*),
  - asimetrični kriptografski algoritmi (engl. *Asymmetric algorithms*),
  - kriptografski algoritmi za računanje sažetaka (engl. *Hashing algorithms*).

3.3.1. Simetrični kriptografski algoritmi

Ključna osobina simetričnih kriptografskih algoritama je da za enkripciju i dekripciju podataka rabe jedan tajni ključ, kao što je prikazano na sljedećoj slici.



Slika 3.5. Simetrična enkripcija

Kao što se iz potonje slike može vidjeti, simetrična enkripcija rabi tajni ključ koji mora biti dijeljen između pošiljatelja tajne poruke i njezinog primatelja. Upravo ovo dijeljenje ključa može predstavljati veliki problem budući da je isti potrebno transportirati od pošiljatelja do primatelja na siguran način.

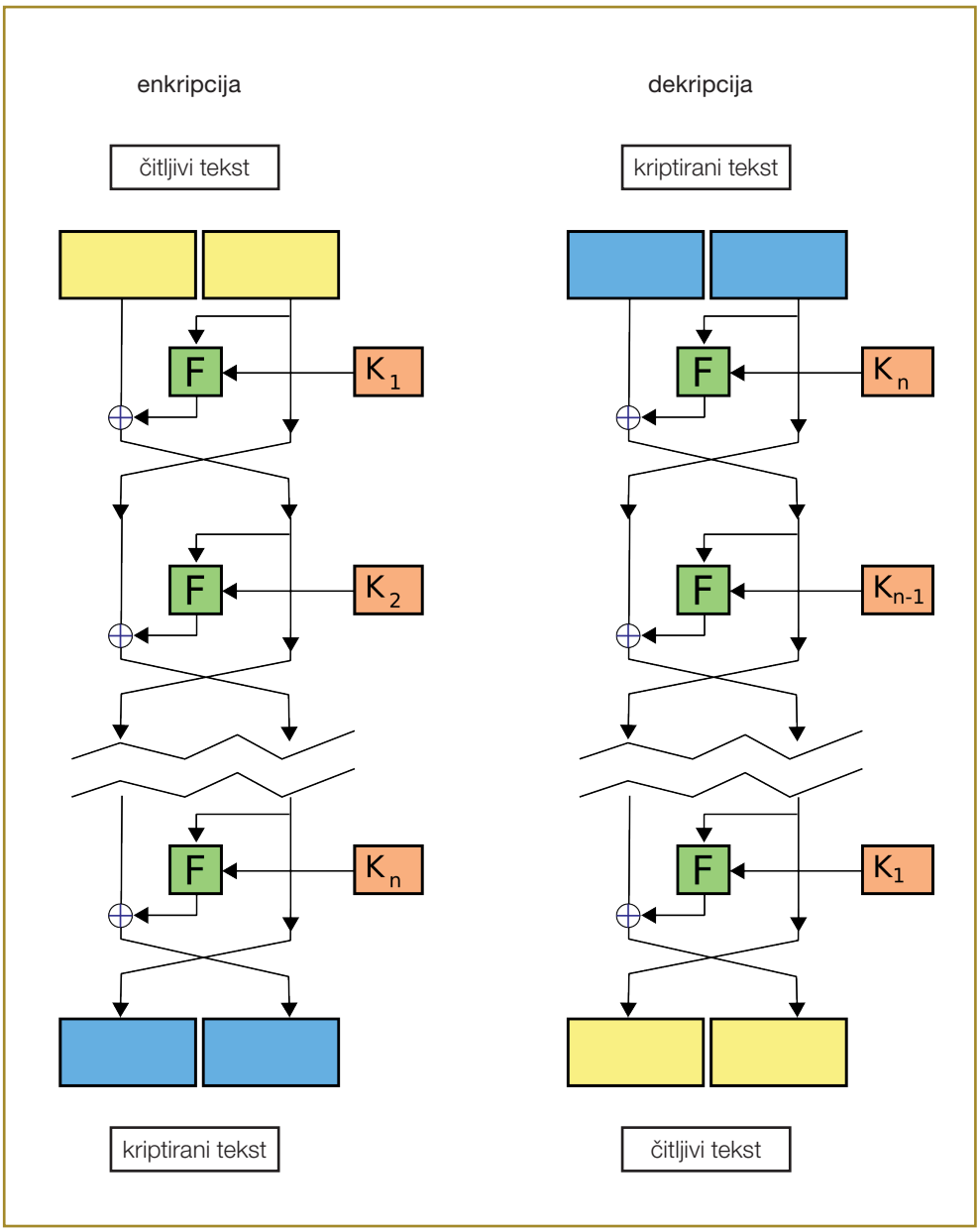
Ključna značajka simetričnih enkripcijskih algoritama je njihova brzina. Zbog toga se ovi enkripcijski algoritmi upotrebljavaju pri enkripciji velike količine podataka.

Blokovni i algoritmi toka podataka

Simetrični enkripcijski algoritmi dijele se u dvije grupe: algoritmi koji rade s blokovima i algoritmi koji rade s tokovima podataka.

- Algoritmi koji rade s blokovima (engl. *block algorithm*) – rade s blokovima podataka koji su obično određene veličine, npr. 64-bitni. Ulazni podaci dijele se na 64-bitne blokove na kojima se provodi niz matematičkih operacija kako bi se dobio enkriptirani tekst. U slučaju da ulazni tekst ne daje cjelobrojni rezultat dijeljenja s veličinom bloka (npr. ulazni tekst je veličine 72 bita), na zadnji se blok obično dodaje prazan sadržaj koji se naziva engl. *padding*.

Na sljedećoj je slici prikazano načelo rada enkripcije i dekripcije upotrebom blokovnog enkripcijskog algoritma koji se naziva Feistel, prema njemačkom kriptografu Horstu Feistelu. Feistelov algoritam korišten je kao baza velikog broja današnjih enkripcijskih algoritama.

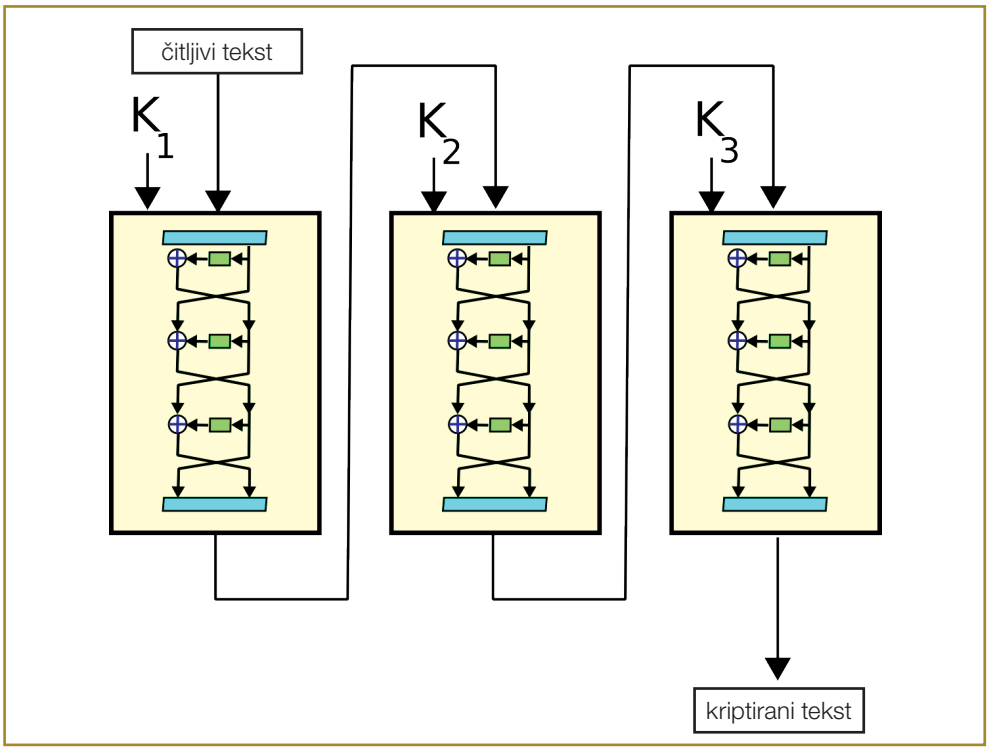


Slika 3.6. Feistelov blokovni enkripcijski algoritam

Neki od danas najčešće korištenih simetričnih blokovnih enkripcijskih algoritama dani su u nastavku.

- DES (engl. *Data Encryption Standard*) predstavlja vjerojatno najpoznatiji simetrični blokovni enkripcijski algoritam. Riječ je o enkripcijskom algoritmu koji je američka vlada odabrala kao standardni algoritam još davne 1976. Ovaj algoritam za zaštitu podataka koristi se 56-bitnim ulaznim ključem, što znači da je ključ veličine 7 okteta. Budući je navedeni ključ relativno mali, još je 1999. pokazano da se upotrebom modernih računala može probiti zaštita podataka enkriptiranih DES algoritmom za manje od 24 sata. DES algoritam uzima ulazne podatke u 64-bitnim blokovima koji se složenim matematičkim operacijama enkriptiraju zajedno s 56-bitnim ključem te dodatnih 8 bitova koji predstavljaju paritet poruke.

  Budući je DES algoritam bio efektivno probijen, preporučena je upotreba 3-DES algoritma u svrhu ojačanja samog algoritma. Kod 3-DES algoritma riječ je jednostavno o proširenju običnog DES algoritma, i to tako da se rabe operacije enkripcije i dekripcije tri puta.



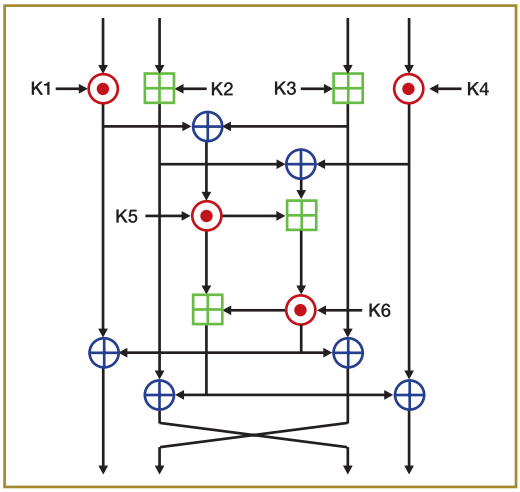
Slika 3.7. 3DES algoritam

3-DES algoritam dopušta duljinu ključa od 56, 112 ili 168 bitova, ovisno o segmentima u kojima se dijelovi ključa upotrebljavaju. Jednostavno opisano, 3-DES algoritam može se opisati sljedećom jednadžbom:

```
enkriptirana poruka = EK3(DK2(EK1(ulazna poruka)))
```

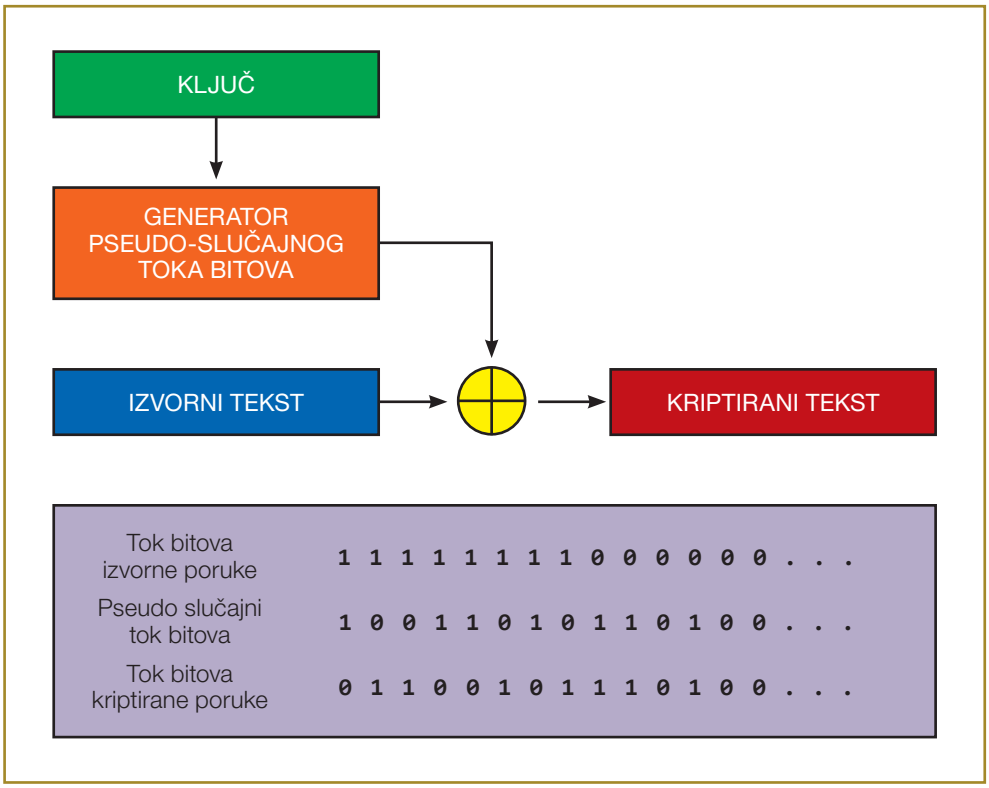
Drugim riječima, ulazna se poruka prvo enkriptira ključem 1, zatim se dekriptira ključem 2 te se opet enkriptira ključem 3. Ovisno o ulaznom ključu upotrebljavaju se različite kombinacije ili samo jedan ključ – npr. ako je ulazni ključ 168-bitni, onda se on razbija na tri 56-bitna dijela.

- AES (engl. *Advanced Encryption Standard*) je nasljednik DES i 3-DES enkripcijskih standarda. Ovaj je algoritam i službeno izabrao američki Institut za standarde i tehnologije. Riječ je o simetričnom blokovnom enkripcijskom algoritmu koji radi na blokovima veličine 128 bitova te podržava ključeve veličine 128, 192 i 256 bitova.
  Pri upotrebi simetričnih blokovnih enkripcijskih algoritama AES danas predstavlja logičan i preporučen izbor budući je riječ o snažnom i brzom algoritmu.
- IDEA (engl. *International Data Encryption Algorithm*) je europski odgovor na DES enkripcijski algoritam. Poput DES algoritma, IDEA radi na blokovima veličine 64 bita sa 128-bitnim ključevima.



Slika 3.8. IDEA algoritam

- Algoritmi koji rade s tokovima podataka (engl. *stream algorithm*) – rade na načelu generiranja pseudo-slučajnog niza bitova. Nad navedenim se nizom zatim obično provode jednostavne logičke operacije kao što je isključivo ili (engl. *Exclusive* OR – XOR), zajedno s nizom ulaznih podataka da bi se dobio enkriptirani niz bitova. Pri dekripciji se rabi zrcalno svojstvo XOR operacije: A XOR B = B XOR A. Drugim riječima, primatelj poruke ponovno generira jednaki pseudo-slučajni niz kao i pošiljatelj te provodi jednaku logičku operaciju na enkriptiranom nizu ne bi li dobio izvorni tekst.
  Primjer rada algoritma s tokovima podataka dan je na sljedećoj slici.



Slika 3.9. Algoritam koji radi s tokovima podataka

Neki od danas najčešće korištenih simetričnih enkripcijskih algoritama koji rade s tokovima podataka navedeni su u nastavku.

- RC4 je najpoznatiji simetrični enkripcijski algoritam koji radi s tokovima podataka. Razvio ga je 1987. Ron Rivest, poznati kriptograf koji je radio i na razvijanju asimetričnih algoritama. RC4 algoritam korišten je u

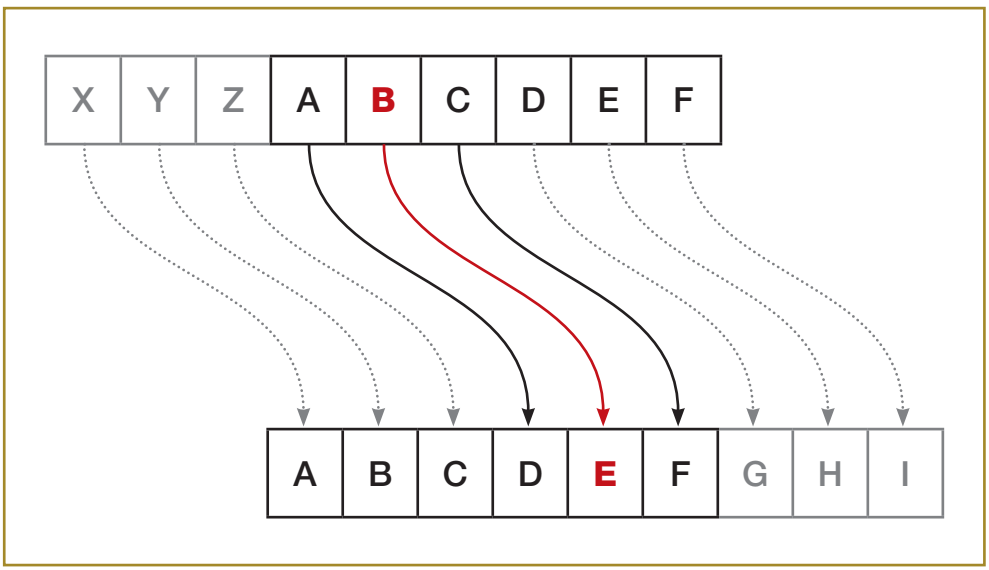
cijelom nizu protokola koji zahtijevaju enkripciju podataka, poput SSL-a te u WEP protokolu za zaštitu bežičnih računalnih mreža.
RC4 podržava veličine ključeva od 40 pa sve do 2048 bitova.

- A5/1 je simetrični enkripcijski algoritam koji radi s tokovima podataka, a koji se danas najčešće upotrebljava za zaštitu GSM mobitela. Riječ je o algoritmu koji je u početku bio držan tajnim, no tijekom godina je otkrivena njegova funkcionalnost. A5/1 rabi efektivni ključ veličine 54 bita.

Supstitucija i transpozicija

Osim blokovnog načina rada ili rada s tokovima podataka simetrični se kriptografski algoritmi dijele i po samom načinu enkriptiranja. Tijekom enkriptiranja, uz ostale matematičke operacije, osnovu predstavljaju supstitucije i transpozicija, operacije koje su same po sebi vrlo jednostavne.

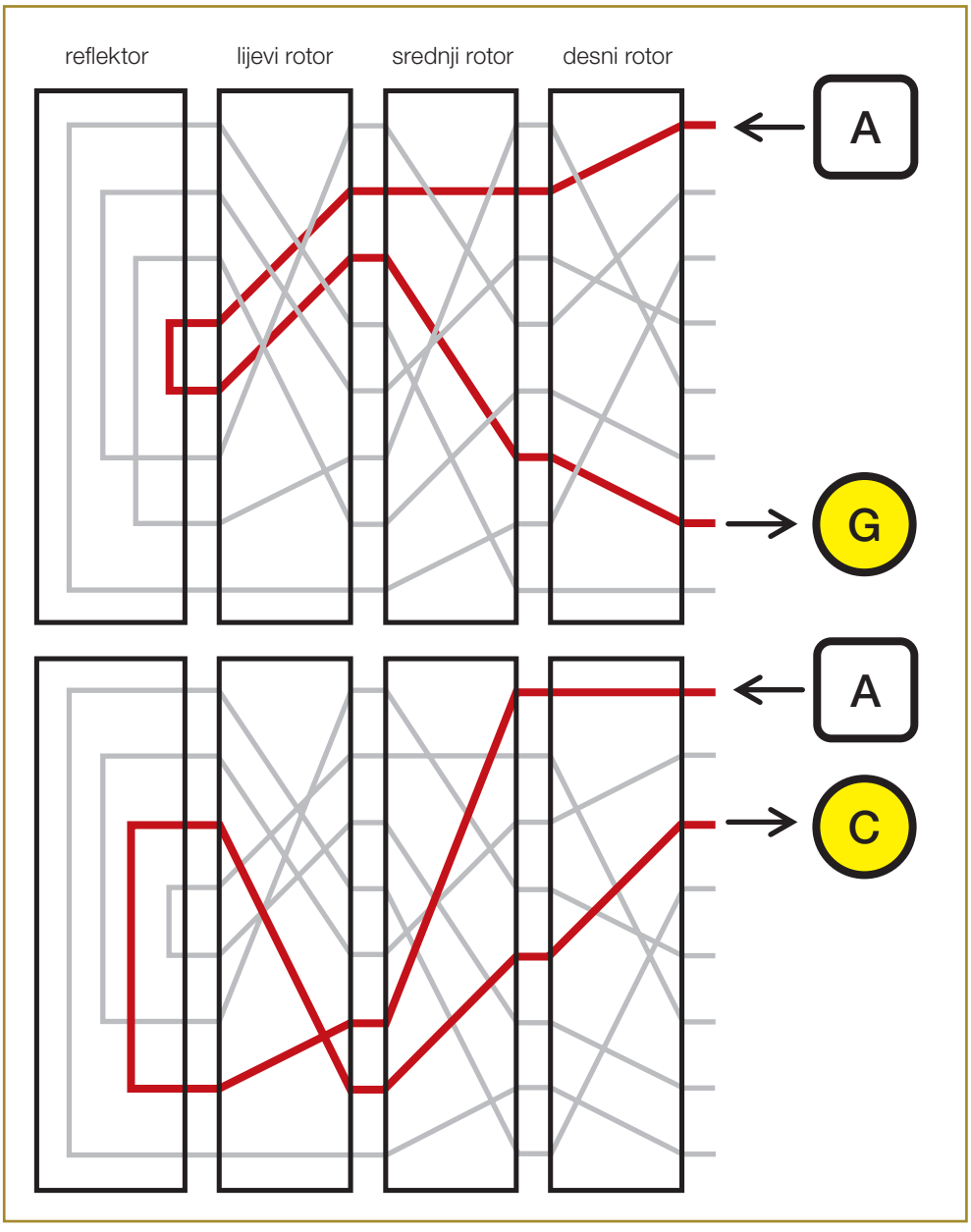
Supstitucijski algoritmi zamjenjuju bitove, oktete ili blokove ulazne poruke koja se enkriptira drugim bitovima, oktetima ili blokovima. Ove se metode zaštite podataka upotrebljavaju već tisućama godina, a najpoznatiji supstitucijski algoritam je Cezarov algoritam koji svako slovo ulazne poruke zamjenjuje s točno određenim slovom abecede koja je pomaknuta za određeni broj polja. Drugim riječima, kod Cezarovog algoritma koji upotrebljava pomak od 3, svako će slovo A iz ulazne poruke biti zamijenjeno slovom D, kao što se može vidjeti na sljedećoj slici.



Slika 3.10. Cezarov algoritam s pomakom od 3 znaka

Puno kompleksnijom supstitucijskom metodom koristio se i Enigma uređaj tijekom drugog svjetskog rata. Enigma uređaj upotrebljavao je višestruke rotore koji su mijenjali položaj ovisno o ulaznom slovu, ali i o trenutnoj poziciji. Slika Enigma uređaja i njegov rad prikazani su na sljedećim slikama.

Slika 3.11. Enigma uređaj



Slika 3.12. Načelo rada Engima uređaja

Transpozicija, druga metoda koja se rabi kod enkripcije podataka, zasniva se na mijenjanju ulaznih podataka prema nekoj unaprijed određenoj metodi. Ova metoda mora podržavati inverznu metodu, koja se u tom slučaju upotrebljava za dekripciju podataka.

Postoji čitav niz transpozicijskih algoritama. Jedan od najjednostavnijih transpozicijskih algoritama je tzv. pružni transpozicijski algoritam (engl. *Rail Fence*). Kod ovog se algoritma ulazna poruka zapisuje prema donjim "prugama" (otud i ime) dok ne stigne do zadnje pruge, nakon čega se zapis nastavlja prema prvoj pruzi, kao što je prikazano na sljedećem primjeru s četiri pruge, za tekst „ULAZNITEKST!!":

```
U . . . . . T . . . . . !
. L . . . I . E . . . ! .
. . A . N . . . K . T . .
. . . Z . . . . . S . . .
```

Enkriptirani niz je sljedeći:

```
U T ! L I E ! A N K T Z S
```

Znajući da je riječ o četiri pruge, izvorni tekst se sada može vrlo jednostavno dobiti natrag iz enkriptiranog.

Potonji primjeri, naravno, pokazuju samo jednostavne mogućnosti transpozicijskih i supstitucijskih operacija. Kod simetričnih se algoritama ove operacije rabe s višestrukim ponavljanjima i s mnogo kompleksnijim postupcima, što osigurava snagu navedenih algoritama.

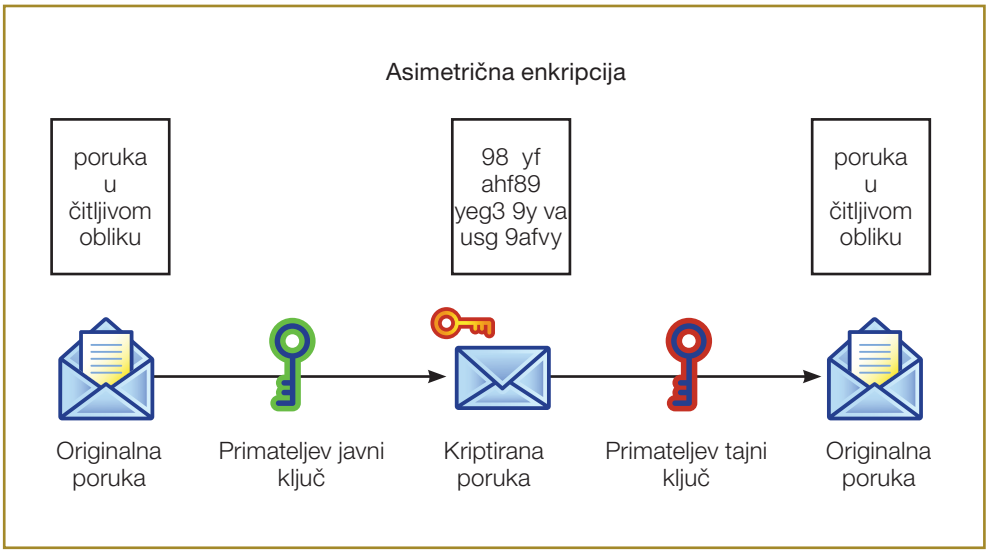
3.3.2. Asimetrični kriptografski algoritmi

Asimetrični kriptografski algoritmi često se nazivaju i algoritmi javnog ključa, budući da je riječ o algoritmima koji sigurnost temelje na paru (dva) ključa. Ovaj par ključeva obično se dijeli na javni i tajni ključ: javni ključ moguće je, kao što mu i ime govori, objaviti dok tajni ključ smije posjedovati samo njegov vlasnik.

Ovaj se par ključeva generira na osnovi matematičkih operacija tako da je jedinstven. No potrebno je napomenuti da je nemoguće iz javnog ključa doći do tajnog ključa i obrnuto – samo osoba koja generira par ključeva može imati oba ključa.

Ovisno o svrsi asimetričnog kriptografskog algoritma, pri postupku enkripcije rabi se jedan ključ iz para dok se za dekripciju upotrebljava drugi ključ iz para. Na primjer, ako se za enkripciju upotrebljavao javni ključ, dekripciju je moguće provesti samo uz posjedovanje tajnog ključa i obrnuto, ako se za enkripciju rabio tajni ključ, dekripciju je moguće provesti samo s javnim ključem.

Asimetrična kriptografija pokazana je na sljedećoj slici.



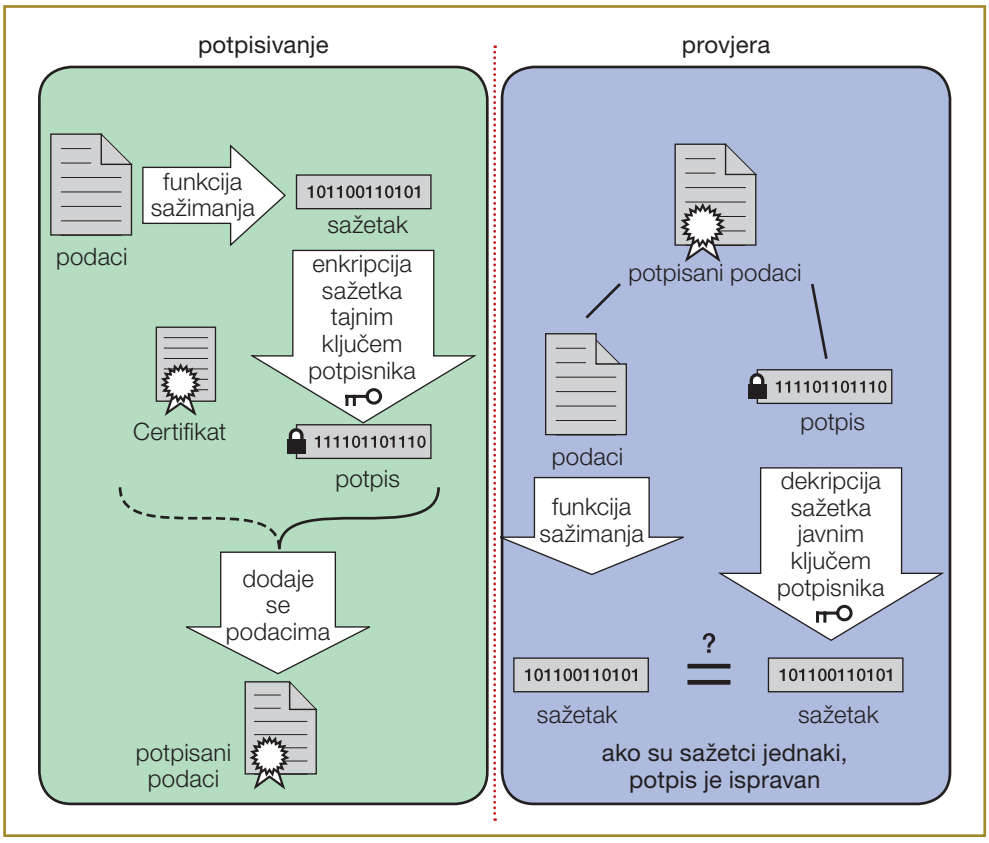
Slika 3.13. Načelo rada asimetrične kriptografije

Za razliku od simetričnih kriptografskih algoritama koji se temelje na operacijama supstitucije i transpozicije, asimetrični kriptografski algoritmi polaze od matematičkih problema vezanih za iznimno velike brojeve. Ideja iza asimetričnih kriptografskih algoritama je da je pomoću velikih brojeva neke operacije jednostavno napraviti dok su druge vrlo teške ili nemoguće. Na primjer, množenje dva broja je vrlo jednostavno, ali je faktorizacija tog produkta natrag na izvorna dva broja vrlo teška, pogotovo ako se uzme u obzir da se pri asimetričnoj kriptografiji upotrebljavaju brojevi koji su općenito veći od 10^{100}.

Zbog navedene se značajke može vidjeti da je sigurnost asimetričnih kriptografskih algoritama zasnovana na kompleksnosti matematičkih operacija na kojima su zasnovani. Budući je kompleksnost ovih matematičkih operacija velika, asimetrični su kriptografski algoritmi znatno sporiji od simetričnih algoritama te se općenito ne upotrebljavaju za enkripciju velikih količina podataka već obično samo za enkripciju nekih ključnih podataka koji mogu biti čak i simetrični ključevi. Naime, čest je postupak da se velika količina podataka enkriptira ko-

rištenjem simetričnog kriptografskog algoritma sa slučajno generiranim tajnim ključem. Ovaj se ključ zatim enkriptira korištenjem asimetričnog kriptografskog algoritma što omogućuje jednostavno slanje primatelju budući da pošiljatelj u ovom slučaju treba imati samo primateljev javni ključ, koji je po definiciji dostupan svima. Primatelj zatim pomoću svog tajnog ključa (koji ima samo on) prvo dekriptira simetrični tajni ključ pomoću kojeg nakon toga dekriptira veliku količinu podataka. Na ovaj je način uspostavljen siguran kanal između pošiljatelja i primatelja te omogućena razmjena podataka.

Na načelu asimetrične kriptografije zasnovan je i digitalni potpis. U ovom se slučaju tajni ključ upotrebljava za potpisivanje poruke. Nakon što je poruka digitalno potpisana, bilo tko može provjeriti potpis pomoću javnog ključa, odnosno dekripcijom te znati da je samo vlasnik tajnog ključa mogao napraviti potpis (odnosno izvornu enkripciju). Digitalni potpis prikazan je na sljedećoj slici.



Slika 3.14. Načelo rada digitalnog potpisa

Neki od najčešće korištenih asimetričnih enkripcijskih algoritama dani su u nastavku:

- Diffie-Hellman predstavlja jedan od prvih asimetričnih enkripcijskih algoritama. Izvorno je ovaj algoritam napravljen kako bi omogućio razmjenu simetričnih tajnih ključeva (korištenih od strane DES enkripcijskog algoritma) preko nesigurnih računalnih mreža.
  Diffie-Hellman algoritam koristi se velikim cijelim brojevima i zasniva se na kompleksnim matematičkim funkcijama, diskretnim logaritmima koje je jednostavno provoditi u jednom smjeru, no iznimno kompleksno u drugom.
- RSA algoritam dobio je ime prema njegovim autorima, Rivestu, Shamiru i Adlemanu. Algoritam je temeljen na matematičkom problemu faktorizacije velikih brojeva i danas predstavlja najčešće korišten asimetrični algoritam. Veličine ključeva kojima se koristi RSA algoritam obično su 1024, 2048 ili 4096 bitova.
- El Gamal predstavlja unaprijeđenu inačicu Diffie-Hellman algoritma te je također temeljen na diskretnim logaritmima.

3.3.3. Kriptografski algoritmi za računanje sažetaka

Izračunavanje sažetaka predstavlja algoritam koji se primjenjuje na ulazne podatke da bi se napravio jedinstveni sažetak (engl. *hash*) koji je fiksne duljine.

Značajka kriptografskih algoritama za računanje sažetaka je, dakle, da uzimaju ulazni tekst proizvoljne duljine te iz njega generiraju jedinstveni sažetak fiksne duljine, obično 128, 160 ili 256 bitova, primjenom neke jednosmjerne funkcije. Budući da ulazni podaci mogu biti praktički bilo koje duljine (od 0 do beskonačno bitova), jasno je da postoji mogućnost kolizije: dva ulazna teksta koji će dati jednak sažetak fiksne duljine. Jedna od značajki dobrih kriptografskih algoritama za računanje sažetaka je upravo ta da je vrlo teško pronaći dva ulazna niza koji će rezultirati jednakim sažetkom.

Kriptografski algoritmi za računanje sažetaka vrlo se često rabe upravo za provjeru integriteta datoteka, budući da jednostavno upućuju na promjenu ulaznih podataka. Naime, čak i promjena samo jednog bita ulaznih podataka uzrokovat će veliku promjenu sažetka, kao što se može vidjeti na sljedećem primjeru koji se služi MD5 algoritmom za računanje sažetaka.

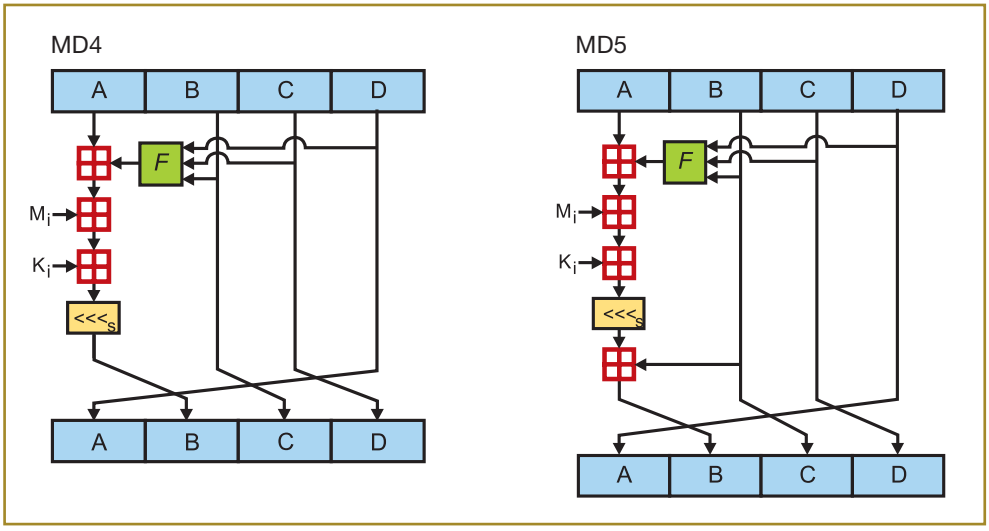
```
MD5(ovo je ulazni tekst.) =
db25762 5b3fd78ba4e46b1d62b06910f

MD5(Ovo je ulazni tekst.) =
969008c5b9f275dde03492a2d94d00d2
```

Iz potonjeg se primjera može vidjeti da je promjena samo prvog slova iz malog u veliko O uzrokovala znatnu promjenu izračunatog sažetka.

Osim za očuvanje integriteta, sažetci se često upotrebljavaju i za pohranjivanje informacija poput zaporki. Naime, pri pohranjivanju zaporki u svrhu zaštite od malicioznog administratora može se pohraniti samo sažetak. Kada se korisnik prijavljuje na sustav i unosi zaporku, sam sustav može opet primijeniti algoritam sažimanja (npr. MD5 kao u prethodnom primjeru) i provjeriti je li dobiveni sažetak jednak onom zapisanom u bazu. Ako je korisnik je unio ispravnu zaporku, dopušta mu se pristup. Ovakav način pohrane zaporki štiti ih od malicioznog administratora koji u bazi zaporki može vidjeti samo sažetak, iz kojeg je nemoguće ili teško doći do ulazne zaporke.

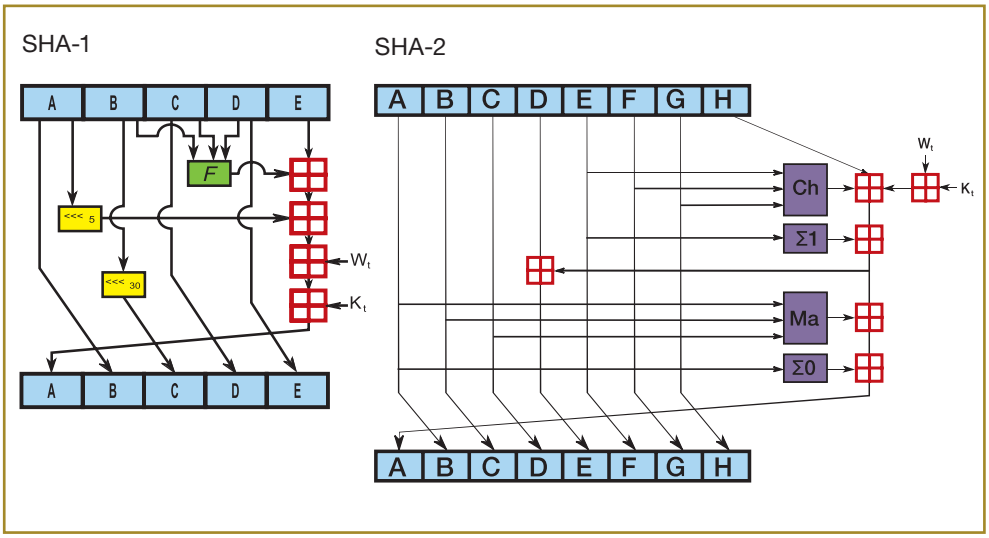
Neki od najčešće korištenih kriptografskih algoritama za računanje sažetaka dani su u nastavku.

- MD4/MD5 (engl. *Message Digest*) klasu algoritama razvio je Ron Rivest (jedan od autora RSA algoritma) u svrhu primjene digitalnih potpisa. Naime, pri računanju digitalnog potpisa, kao što je objašnjeno u poglavlju 3.3.2, vlasnik tajnog ključa ne enkriptira cijelu poruku već samo njezin sažetak. Na taj se način osigurava i njezin integritet, budući da će promjena poruke uzrokovati promjenu sažetka, koji više neće odgovarati onom enkriptiranom (odnosno potpisanom).
  MD4 i MD5 generiraju 128-bitne sažetke, no zbog pronađenih se slabosti u ovim algoritmima pri računanju sažetaka u osjetljivim okolinama njihova upotreba više ne preporučuje.



Slika 3.15. MD4 i MD5

- SHA (engl. *Secure Hash Algorithm*) napravila je američka vlada u svrhu zamjene nesigurne MD klase algoritama sažimanja. Najčešće korišteni algoritam je SHA-1, koji generira 160-bitne sažetke, no moguće su i druge upotrebe koje generiraju veće sažetke te su time i sigurnije.



Slika 3.16. SHA-1 i SHA-2

3.3.4. Napadi na kriptografske algoritme

Pokušaj napada na kriptografske algoritme naziva se kriptoanalizom. Riječ je o disciplini koja pokušava na različite načine doći do izvornog teksta ili tajnog ključa koji je korišten pri enkripciji.

Prva metoda napada zapravo je vrlo jednostavna i logična – napadač koji je u posjedu enkriptiranog teksta te zna koji je enkripcijski algoritam bio korišten može jednostavno isprobavati sve moguće ključeve dok ne pogodi onaj tajni ključ koji je bio korišten pri enkripciji. Ovaj se proces naziva napadanje sirovom snagom (engl. *brute force*) te može uroditi plodom samo kod slabih (jednostavnih) kriptografskih algoritama poput DES-a ili u slučaju kada je korišten jednostavan tajni ključ. U drugim je slučajevima ovakav napad za napadača neisplativ jer proces pronalaska tajnog ključa može trajati predugo – snažni kriptografski algoritmi mogu uzrokovati traženje ključa milijunima godina uz današnja računala.

Drugi se napadi zasnivaju na različitim matematičkim značajkama kriptografskog algoritma koji se napada. Jednako tako, ovisno o podacima kojima napadač raspolaže, napad može biti jednostavniji ili kompliciraniji. Na primjer, ako

je napadaču uz enkriptirani tekst dostupan i dio izvornog teksta, napad je jednostavniji. Ovakav se napad naziva napadom poznatim izvornim tekstom (engl. *known-plain text attack*). Osim navedenog napada, moguć je scenarij u kojem napadač može sam birati izvorni tekst te na taj način pokušati napraviti seriju ulaznih podataka koje mu olakšavaju možebitnu analizu. Ovakvi se napadi nazivaju napadima odabranim poznatim izvornim tekstom (engl. *chosen plain text attack*).

3.4. Zaštita pristupa podacima pomoću enkripcije

Kao što je rečeno u poglavljima 3.1 i 3.2, prava pristupa datotekama mogu ograničiti korisnike koji istima mogu pristupiti. No u svrhu povećanja sigurnosti podataka danas se vrlo često osjetljive informacije enkriptiraju, i to ne samo u vidu datoteka, već i direktorija pa čak i cijelih diskova.

Klasičan scenarij gdje je enkripcija diskova poželjna je na prijenosnim računalima na kojima su pohranjeni osjetljivi podaci. U slučaju gubitka ili krađe prijenosnog računala možebitno postavljena prava pristupa napadaču ne predstavljaju nikakav problem budući da može podignuti računalo s alternativnim operacijskim sustavom (npr. s CD-ROM-a) te tako zaobići prava pristupa. No ako je enkripcija datoteka bila ispravno korištena, napadač u ovom slučaju još uvijek ne može pristupiti datotekama već treba dodatno napasti korišteni kriptografski algoritam što može znatno ili u potpunosti onemogućiti ovakav napad.

3.4.1. Windows EFS

U svrhu zaštite podataka Windows operacijski sustavi omogućuju primjenu enkriptiranog datotečnog sustava (engl. Encrypting File System – EFS). Ovaj tip datotečnog sustava dostupan je od Windowsa 2000 te zahtijeva upotrebu NTFS datotečnog sustava koji ga podržava.

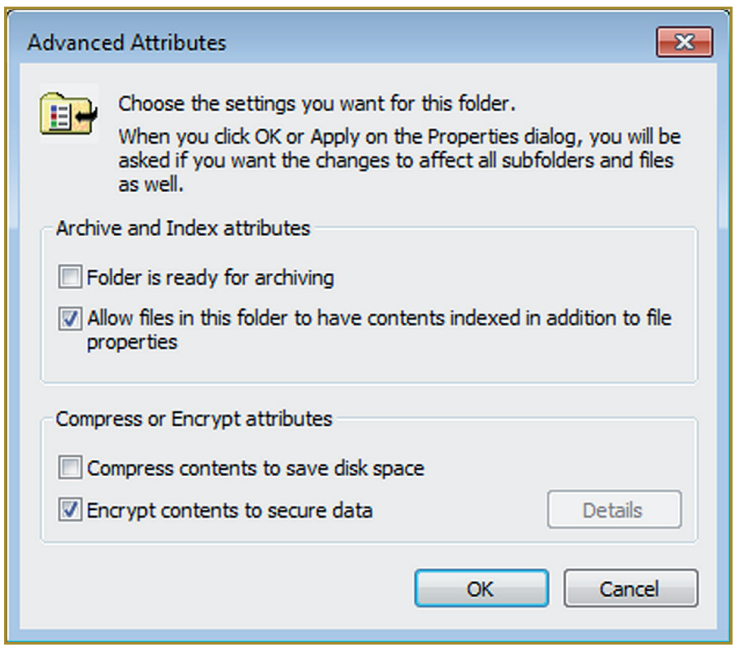
EFS je vrlo koristan dodatak koji omogućuje transparentno enkriptiranje i dekriptiranje podataka u stvarnom vremenu, a svaki ga korisnik Windows operacijskog sustava može upotrebljavati prema volji, bez potrebe za ikakvim dodatnim konfiguriranjem.

EFS se koristi i simetričnom i asimetričnom kriptografijom pri radu. Kada se datoteka želi enkriptirati primjenom EFS-a, Windowsi generiraju slučajan tajni ključ za simetrični enkripcijski algoritam koji će služiti za enkripciju datoteke. Windowsi omogućuju primjenu 3-DES i AES algoritama za enkripciju datoteka.

Navedeni tajni ključ, koji se još naziva i FEK (engl. *File Encryption Key*), zatim se enkriptira pomoću asimetričnog enkripcijskog algoritma i javnim ključem korisnika koji je vlasnik datoteke. U slučaju da korisnik ovu, enkriptiranu, datoteku želi dijeliti s drugim korisnicima (npr. njezinim postavljanjem na dijeljeni disk), potrebno je ručno odabrati sve korisnike kojima se želi omogućiti pristup (obratiti pažnju da ove dozvole nemaju veze s pravima pristupa). Windowsi će automatski iskoristiti javne ključeve svih odabranih korisnika te pomoću njih također enkriptirati FEK.

Kada korisnik želi pristupiti enkriptiranoj datoteci, Windowsi će prvo provjeriti ima li korisnik uopće pravo pristupa prema ACL-ovima. Ako korisnik ima pravo pristupa, u posebnoj strukturi pohranjenoj s datotekom potražit će se enkriptirani FEK, koji je bio enkriptiran s javnim ključem tog korisnika. Ako je ovaj enkriptirani FEK pronađen, Windowsi će automatski primjenom tajnog ključa korisnika koji je trenutno prijavljen dekriptirati FEK te pomoću dekriptiranog FEK-a dekriptirati i samu datoteku upotrebom simetričnog kriptografskog algoritma. Sav ovaj postupak odvija se u pozadini, u stvarnom vremenu i u potpunosti transparentno za korisnika.

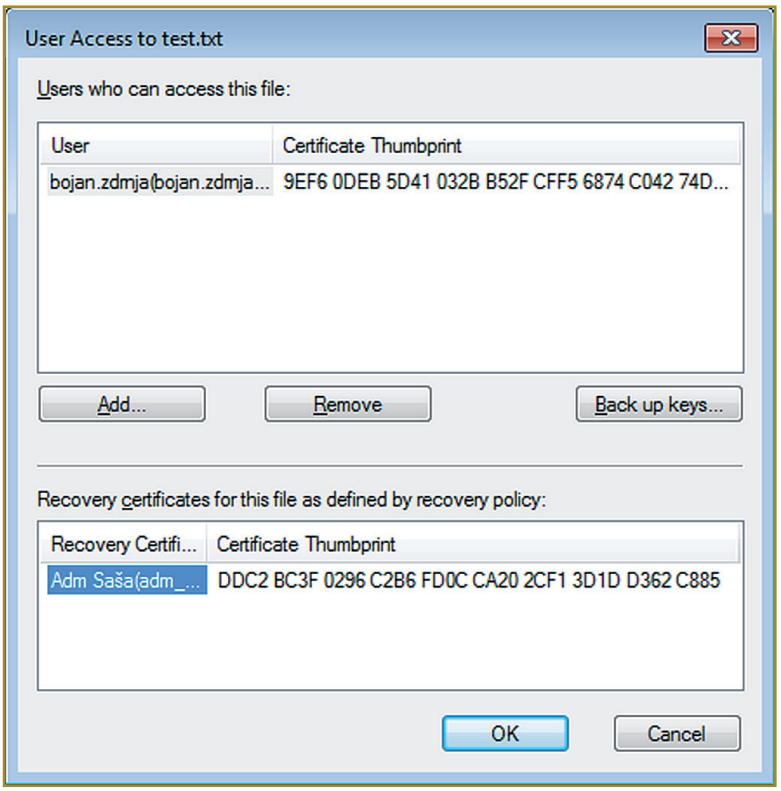
Korisnici mogu uključiti enkripciju na pojedinim datotekama ili direktorijima jednostavnim odabirom Properties izbornika datoteka i otvaranjem naprednih atributa, kao što je prikazano na sljedećoj slici.



Slika 3.17. Izbornik za enkriptiranje datoteka i direktorija

U slučaju da je računalo u Windows domeni, administrator ima mogućnost definiranja korisničkog računa koji će se upotrebljavati za povrat datoteka (engl. *Recovery Agent*). Naime, u slučaju da korisnik koji je vlasnik npr. zaboravi svoju zaporku, datoteke će automatski postati nedostupne. U slučaju da je bio definiran korisnički račun za povrat datoteka, pri inicijalnoj će enkripciji Windowsi automatski enkriptirati FEK ne samo s javnim ključem korisnika već i s javnim ključem korisničkog računa za povrat datoteka. To će omogućiti administratoru vraćanje sadržaja datoteka pomoću ovog posebnog računa čak i u slučaju da je korisnik zaboravio svoju zaporku.

Na sljedećoj su slici prikazani detalji enkriptirane datoteke na kojima se može vidjeti definirani korisnički račun za vraćanje podataka u Windows domeni.



Slika 3.18. Definiranje Recovery Agenta

Na kraju, potrebno je napomenuti da EFS štiti samo datoteke pohranjene na tvrdom disku. Ako se ove datoteke prenose računalnom mrežom (npr. kopiranjem na dijeljeni tvrdi disk), one se prenose u čistom tekstualnom obliku pa ih je potrebno dodatno zaštititi, ako za tim postoji potreba.

3.4.2. Enkripcija cijelog diska

Ako je upotreba EFS-a nedovoljna, moguće je enkriptirati cijeli disk. Iako EFS pruža zadovoljavajuću zaštitu pojedinih datoteka, potrebno je napomenuti da pri rukovanju istima operacijski sustav često sam kreira kopije datoteka koje nisu enkriptirane. Na primjer, ako nije enkriptiran cijeli direktorij, pri otvaranju datoteke u aplikaciji poput Microsoft Worda bit će napravljene privremene datoteke u čistom tekstualnom obliku. Ako napadač dođe do tvrdog diska računala (npr. krađom prijenosnog računala), forenzičkim tehnologijama analize tvrdog diska moguće je vratiti obrisane datoteke uključujući i ove privremene datoteke što bi napadaču omogućilo dolazak do izvorne datoteke bez potrebe za napadom na enkripciju. Jednako tako, pri samoj enkripciji EFS zapravo prvo stvara datoteku u čistom tekstualnom obliku koju zatim enkriptira u novu ciljnu (enkriptiranu) datoteku, a izvornu datoteku briše. Ovdje se također može vidjeti da se vraćanjem obrisanih datoteka može doći do izvornog teksta.

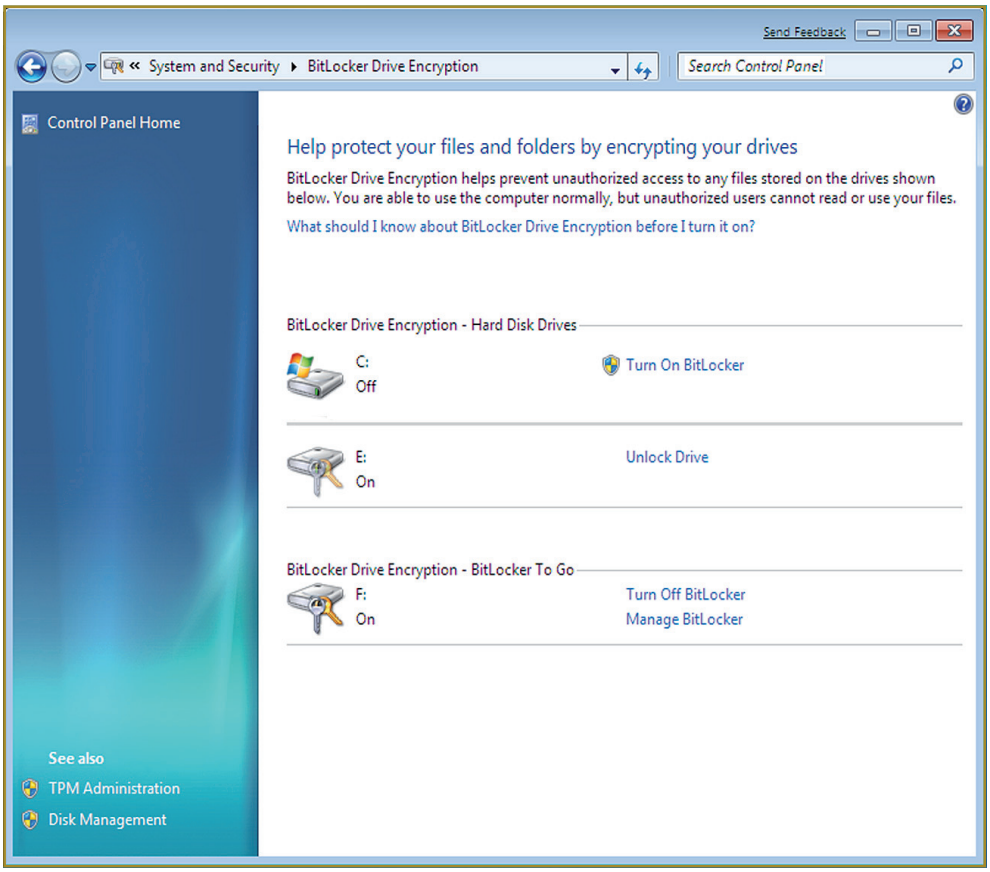
Slika 3.19. Zaštita na nivou cijelog diska

BitLocker

Počevši s Windows Vista operacijskim sustavima, Microsoft nudi BitLocker Drive Encryption mogućnost enkriptiranja pojedinih particija na tvrdom disku. Za enkripciju se BitLocker koristi AES simetričnim algoritmom sa 128-bitnim ključem. Ovisno o hardveru u računalu, za pohranu tajnog ključa moguće je upotrebljavati čak i TPM čip (engl. *Trusted Platform Module*) koji omogućuje

transparentnu upotrebu – korisnik može samo uključiti računalo i Windowsi će se normalno podići. Osim navedenog načina rada ključ može biti pohranjen i na USB memorijski medij čime se dodatno podiže razina sigurnosti.

Postavljanje BitLockera vrlo je jednostavno i može se napraviti nakon instalacije Windowsa.

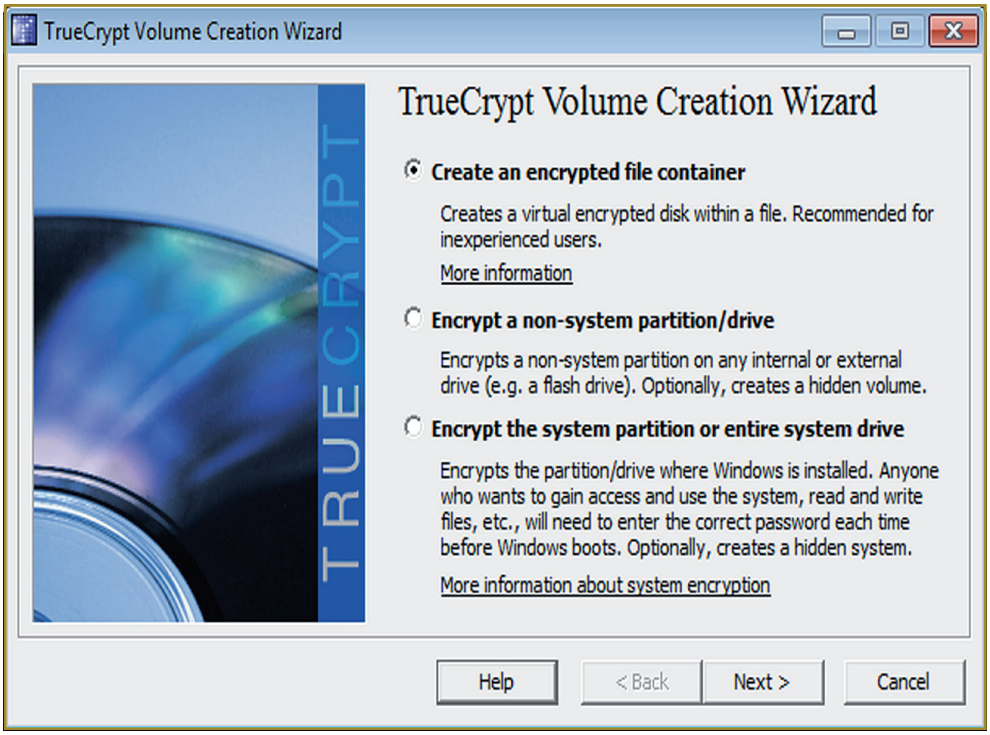


Slika 3.20. Uključivanje BitLockera

3.4.2.2 TrueCrypt

TrueCrypt predstavlja još jedan paket koji omogućuje enkripciju cijelog diska. Riječ je o besplatnom, *open source* paketu koji je podržan na čitavom nizu operacijskih sustava, uključujući Windowse i Linux. Osim enkripcije tvrdog diska, TrueCrypt omogućuje i stvaranje enkriptiranih particija koje se mogu prenositi s jednog računala na drugo – particije su pohranjene u vidu običnih datoteka koje se zatim mapiraju na računalo, naravno nakon unošenja ispravnog tajnog ključa koji omogućuje dekripciju.

Kod enkripcije cijelog diska TrueCrypt od korisnika traži unošenje zaporke pri pokretanju. TrueCrypt omogućuje upotrebu različitih enkripcijskih algoritama, uključujući AES, Twofish i Serpent. Veličina ključa koji se rabi je 256 bitova.



Slika 3.21. Definiranje particija koje će se enkriptirati TrueCryptom

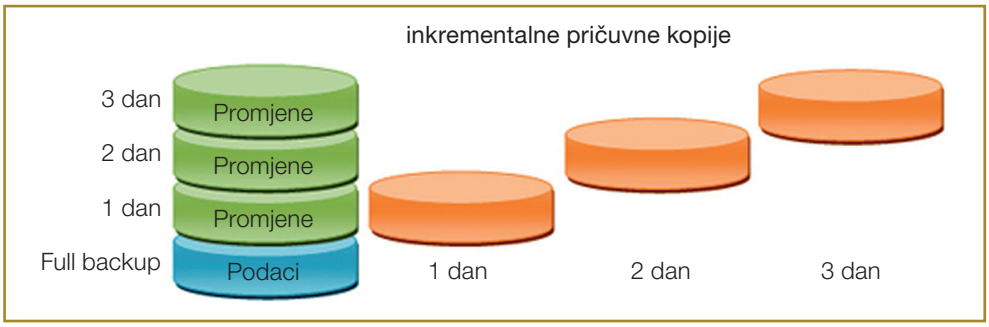
3.5. Pričuvne kopije podataka

Izrada pričuvnih kopija podataka predstavlja osnovni zahtjev za čuvanjem podataka. U slučaju katastrofalnog događaja, pričuvne su kopije vrlo često jedini izvor iz kojeg se mogu vratiti podaci. Kada se radi pričuvna kopija, izvorni se podaci pohranjuju na medije koji mogu biti spremljeni na fizički odvojenu lokaciju od glavne lokacije. Na ovaj se način pokušava reducirati rizik od mogućeg katastrofalnog događaja na glavnoj lokaciji. Vrlo često tvrtke u ovu svrhu iznajmljuju poseban čuvani prostor (ili recimo sef) u bankama ili drugim tvrtkama koje pružaju specijalizirane usluge čuvanja ovakvih medija.

Vrsta medija na koji se pohranjuju pričuvne kopije ovisi o količini podataka koji se pohranjuju. Danas se najčešće upotrebljavaju digitalne audiotrake (DAT), digitalne linearne trake (DLT) te CD/DVD mediji.

Pri izradi pričuvnih kopija podataka administrator treba odlučiti koji će podaci biti kopirani na pričuvne medije, uključujući i metodu izrade pričuvne kopije podataka. Metode izrade pričuvne kopije uključuju sljedeće tehnike:

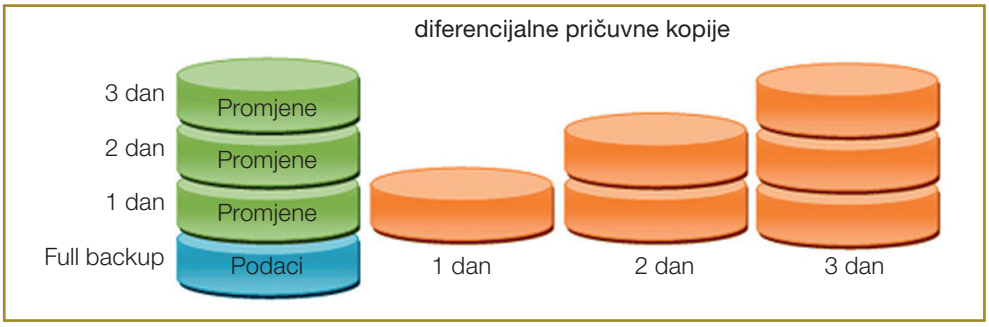
- puna pričuvna kopija (engl. *Full backup*) – kopira cijeli sustav (sve datoteke) na pričuvni medij odjednom. Pune pričuvne kopije obično uključuju sve korisničke datoteke, datoteke sustava i bilo kojih drugih aplikacija na samom poslužitelju. Pri izradi pričuvne kopije svake datoteke softver za izradu pričuvne kopije postavlja arhivnu zastavicu datoteke kako bi se znalo da je izrađena pričuvna kopija iste. Neki sustavi za izradu pričuvnih kopija ne postavljaju arhivnu zastavicu već sve podatke drže u lokalnoj bazi podataka;
- izrada inkrementalne pričuvne kopije (engl. *Incremental backup*) – stvara pričuvnu kopiju samo onih datoteka koje su se promijenile od zadnje izrade pričuvne kopije. Ovo uključuje i sve nove datoteke koje su napravljene od izrade zadnje pričuvne kopije. Budući da se izrađuje pričuvna kopija samo onih datoteka koje su se promijenile, izrada inkrementalnih pričuvnih kopija općenito je najbrža, no može se vidjeti da vraćanje cijelog operacijskog sustava u ovom slučaju zahtijeva i vraćanje svih pričuvnih kopija koje su prethodile inkrementalnoj pričuvnoj kopiji, što može znatno otežati i produljiti postupak vraćanja sustava.



Pri izradi pričuvne kopije promijenjenih ili novih datoteka, kao i u prethodnom slučaju, postavlja se njihova arhivna zastavica kako bi se znalo da je izrađena pričuvna kopija;

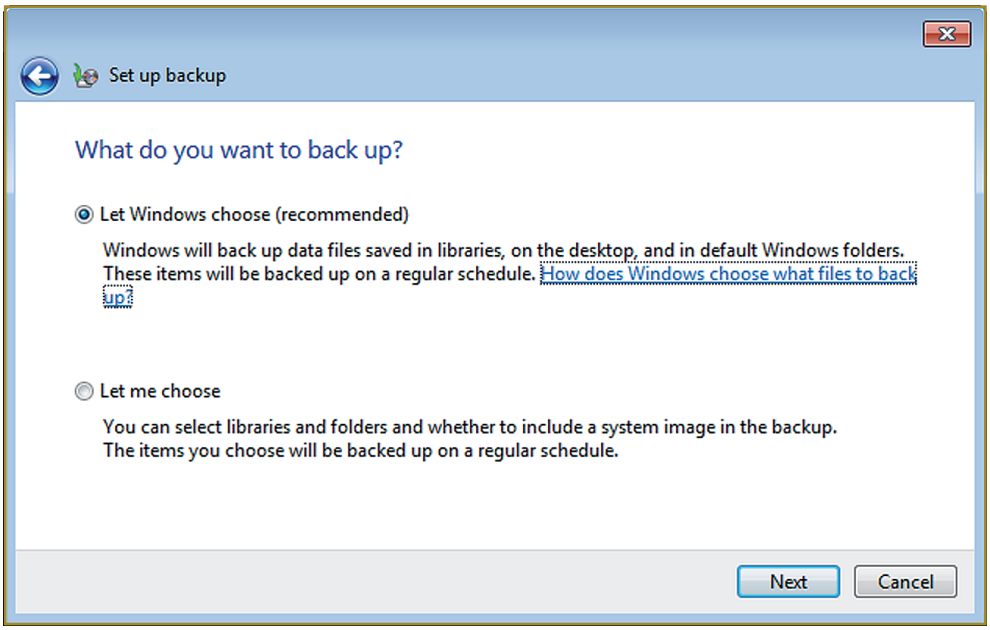
- diferencijalne pričuvne kopije (engl. *Differential backup*) – rade pričuvne kopije svih datoteka koje su se promijenile ili su novonapravljene od zadnje izrade pune pričuvne kopije. Kada se radi ovakva pričuvna kopija, arhivna zastavica datoteka ne postavlja se tako da jedna diferencijalna

pričuvna kopija ima jednake datoteke kao i prethodna diferencijalna pričuvna kopija, s dodatkom onih koje su se promijenile u međuvremenu;



- izravno kopiranje (engl. *Copy backup*) svih datoteka na neki drugi medij – ovdje je riječ o običnom kopiranju koje također ne modificira arhivnu zastavicu. Izravno kopiranje obično se rabi za izradu pričuvnih kopija na drugim brzim medijima poput vanjskih tvrdih diskova.

Na sljedećoj je slici prikazan Backup program koji dolazi s Windows operacijskim sustavima. Navedeni Backup program omogućuje izradu punih, inkrementalnih, diferencijalnih i izravnih pričuvnih kopija.



Slika 3.22. Windows Backup program

Budući da različite metode prave pričuvne kopije na različite načine, općenito ovisi o pojedinoj tvrtki kojom će se metodom izrade pričuvnih kopija koristiti. U većini slučajeva rade se pričuvne kopije cijelog sustava na tjednoj bazi, da bi se nakon toga svaki dan radila inkrementalna kopija. Na taj se način postiže optimalna ušteda u potrošenim medijima te ušteda vremena potrebnog za izradu pričuvne kopije budući da se pričuvna kopija cijelog sustava može raditi preko vikenda kada su poslužitelji u načelu neopterećeni, a inkrementalne kopije preko tjedna traju kratko. Primjer izrade pričuvnih kopija dan je u sljedećoj tablici.

Dan u tjednu	Tip pričuvne kopije
Ponedjeljak	Puna kopija
Utorak	Inkrementalna kopija
Srijeda	Inkrementalna kopija
Četvrtak	Inkrementalna kopija
Petak	Inkrementalna kopija
Subota	Inkrementalna kopija
Nedjelja	Bez izrade pričuvne kopije

Prikazana tablica omogućuje kvalitetnu izradu pričuvnih kopija s minimalnim troškom za medije na koje se pohranjuje pričuvna kopija. Budući da mediji mogu predstavljati znatan trošak, obično se upotrebljava određena rotacijska shema medija koja omogućuje ponovno upotrebljavanje medija na kojima je pohranjena neka od starih pričuvnih kopija. Ovo je posebno uobičajeno kada je riječ o trakama na koje se pohranjuju pričuvne kopije.

Najčešće korištena shema rotacije medija naziva se Djed-Otac-Sin (engl. GFS – *Grandfather, Father, Son*). Kod ove se sheme rotacije medija puna pričuvna kopija smatra Ocem, dok se dnevna izrada pričuvnih kopija smatra Sinovima. Pričuvna kopija koja se naziva Djed predstavlja još jednu punu pričuvnu kopiju koja se dodatno pohranjuje na neku udaljenu lokaciju i čuva određeno vrijeme. Na taj se način omogućuje vraćanje starih podataka čak i nakon što su mediji koji su se upotrebljavali za pohranjivanje Otac i Sin kopija vratili natrag u upotrebu.

Na kraju, potrebno je napomenuti da su pričuvne kopije dobre samo onoliko koliko je pouzdana sama procedura izrade pričuvne kopije te naročito vraćanje podataka. Vrlo je čest slučaj da se pričuvne kopije rade godinama bez pravog testiranja tako da u slučaju katastrofe dolazi do problema vraćanja podataka, makar su pričuvne kopije bile napravljene. U svrhu provjere ispravnosti izrade pričuvnih kopija, administratori bi trebali u regularnim intervalima pokušati napraviti vraćanje sustava, npr. svakih šest mjeseci ili godinu dana.

4. Aplikacijski sloj

Da bi informacije pohranjene u informacijskom sustavu imale svrhu, potrebno im je pristupati i obrađivati ih. Sučelje koje korisnici upotrebljavaju u svrhu pristupa informacija nalazi se na aplikacijskom sloju – riječ je o aplikacijama i servisima koji komuniciraju koristeći se računalnom mrežom, ali i drugim metodama komunikacije koje omogućuju moderni operacijski sustavi. Aplikacijski je sloj zadužen za upravljanje, uspostavljanje, koordiniranje i prekidanje komunikacijskih veza između aplikacija koje se nalaze na obje strane veze.

Aplikacije uključuju sve servise i sučelja kojima se koriste krajnji korisnici, ali i bilo koje druge procese koji uspostavljaju komunikacijske veze s drugim elementima informacijskog sustava, bez obzira na tehnologiju ili programski jezik u kojem su izvedene.

Kao što je slučaj sa svim ostalim slojevima, i aplikacijski se sloj naslanja na temelje koje mu pruža podatkovni sloj, a koji su vezani za prava pristupa informacijama pohranjenim u informacijskom sustavu. Drugim riječima, pri pristupanju aplikacije nekom podatku koji je pohranjen u datoteku operacijski sustav zaštitama na podatkovnom sloju osigurava da samo aplikacije koje su pokrenute pod autoriziranim korisničkim računom (korisničkim računom koji ima pravo pristupa datoteci) zbilja i mogu ostvariti to pravo pristupa.

Budući da aplikacije predstavljaju sučelje prema krajnjim korisnicima, jasno je da su one danas i najizloženije neovlaštenim aktivnostima potencijalnih napadača. Kao što smo već rekli, napadaču ultimativni cilj predstavlja dolazak do željenih informacija pohranjenih na informacijskom sustavu. Iako kompromitiranjem operacijskog sustava napadač automatski dobiva i pristup željenim informacijama, današnji, moderni, operacijski sustavi predstavljaju popriličan izazov u smislu da se tvrtke koje razvijaju ove operacijske sustave (npr. Microsoft) već godinama služe dobrim praksama i tehnikama razvijanja softvera (engl. SDLC – *Software Development Life Cycle*) pa su i ranjivosti samih operacijskih sustava postale rijetkost.

No budući da aplikacije trebaju imati neometan pristup podacima (na kraju, korisnici upotrebljavaju upravo navedene aplikacije kao sučelje prema pohranjenim podacima, a ne pristupaju podacima izravno), jasno je da napadač koji uspije kompromitirati aplikaciju može pristupiti i željenim podacima, bez obzira zna kompromitiranje samog operacijskog sustava.

Još jedan od problema jest i činjenica da se aplikacije obično razvijaju tako da je u prvom planu njihova funkcionalnost, a ne sigurnost. Da bi se ostvarila optimalna zaštita, sigurnost bi trebala biti dio dizajna aplikacije jednako kao i funkcionalnost, no danas nažalost još uvijek mali broj tvrtki posvećuje ovome dovoljno pažnje. Zbog toga se komponenta sigurnosti u aplikacije danas vrlo često dodaje na samom kraju razvojnog ciklusa, kao sučelje prema krajnjem korisniku, što u krajnjem slučaju jednom autenticiranom korisniku praktički ostavlja otvorene ruke.

Aplikacije obavljaju različite zadatke – mogu kontrolirati ulazne podatke, procesirati ih, komunicirati s drugim aplikacijama preko računalne mreže ili drugih komunikacijskih metoda, pristupati podacima i pohranjivati ih te konačno razmjenjivati poruke s operacijskim sustavom. Kao što je već rečeno, pri razvoju aplikacija treba imati na umu potencijalne prijetnje i rizike s kojima će se susretati aplikacija. Ovo se posebno odnosi na javne servise koji su izravno izloženi napadačima na Internetu.

Kao i u prethodnim slučajevima, sigurnosni mehanizmi koje možemo primjenjivati mogu biti administrativne i fizičke prirode, no u kontekstu aplikacija najviše je pažnje potrebno posvetiti tehničkim metodama zaštite, što je i predmet ove cjeline.

Cilj implementacije ispravnih sigurnosnih mehanizama je smanjivanje broja sigurnosnih ranjivosti (te izravno i sigurnosnog rizika upotrebe aplikacije) i mogućnosti narušavanja integriteta pohranjenih podataka. Sigurnosne kontrole koje se implementiraju u vidu ovih sigurnosnih mehanizama mogu biti preventivne, detektivne i korektivne.

- Preventivne sigurnosne kontrole pokušavaju spriječiti neku sigurnosnu ranjivost bez obzira na to je li ona prisutna na sustavu ili ne. Na primjer, ograničavanjem korisničkog unosa u nekoj formi na 20 znakova automatski je spriječen pokušaj prepisivanja drugih bitnih struktura aplikacije kao što je to slučaj kod napada prepisivanjem spremnika (više o napadima prepisivanja spremnika u poglavlju 4.1.1).
- Detektivne sigurnosne kontrole imaju za cilj otkrivanje potencijalnih napada. Za razliku od prethodnog slučaja kad je korisnički unos npr. ograničen na određeni broj znakova, detektivne sigurnosne kontrole upozoravaju administratora na pokušaj napada. Detekcija se može postići na različite načine, npr. analizom ulaznih podataka.
- Korektivne sigurnosne kontrole uklanjaju sigurnosne ranjivosti za koje se zna da postoje u aplikaciji. Ove se kontrole najčešće implementiraju u

vidu nekih vanjskih sigurnosnih mehanizama – npr. ako aplikacija prenosi korisničko ime i zaporku u čistom tekstualnom obliku preko računalne mreže, moguća je primjena enkripcije na mrežnoj razini (više o tome u poglavlju 6.6) kako bi se ispravila inherentna sigurnosna ranjivost aplikacije.

Specifične kontrole koje će biti implementirane u pojedinu aplikaciju ovise, naravno, o samoj aplikaciji i njezinom načinu rada. Osim toga, kod današnjih modernih aplikacija pojedine ranjivosti ovise i o programskom jeziku/okruženju u kojem se razvija aplikacija. Tako npr. napadi prepisivanja spremnika obilježavaju programske jezike koji nemaju automatsku kontrolu veličine varijabli/objekata te dopuštaju rukovanje pokazivačima u memoriji poput C i C++ programskih jezika, dok objektno orijentirani programski jezici poput Jave, C# ili Pythona nisu uopće ranjivi na navedene napade.

Ključni zahtjevi koji se postavljaju pred siguran razvoj aplikacija uključuju sljedeće:

- razumijevanje sigurnosnih zahtjeva aplikacije kako bi se u razvoj mogli uključiti pravilni sigurnosni mehanizmi, i to ne samo u aplikaciju već i u sve ostale slojeve modela sigurnosti (npr. enkripcija na razini računalne mreže;
- implementiranje ispravnih sigurnosnih kontrola i mehanizama;
- detaljno testiranje implementiranih sigurnosnih mehanizama i njihove integracije s aplikacijom. Detaljno testiranje uključuje ne samo funkcionalnost već i sigurnosne značajke implementiranih sigurnosnih kontrola;
- pridržavanje poznatih i prihvaćenih metodologija razvoja aplikacija;
- omogućavanje sigurne i pouzdane distribucije konačnog programskog paketa.

4.1. Sigurnosne ranjivosti aplikacija

Kao što je već rečeno u prethodnom poglavlju, tehničke sigurnosne ranjivosti aplikacija ovise o programskom jeziku odnosno okruženju u kojem su aplikacije razvijene. U nastavku ovog poglavlja dan je pregled najčešćih sigurnosnih ranjivosti današnjih aplikacija.

Sigurnosne ranjivosti prepisivanja spremnika vrlo su česte u aplikacijama pisanim u C i C++ programskim jezicima budući da isti ne uvjetuju nikakvu kontrolu veličine spremnika.

Sigurnosne ranjivosti web-aplikacija predstavljaju posebnu kategoriju sigurnosnih ranjivosti. Kako je zadnjih godina sve vidljiviji trend prijelaza na web kao sučelje prema korisniku (odnosno odmak od tradicionalnog modela *fat client* aplikacija), tako je potrebno i posebnu pažnju posvetiti upravo zaštiti ovih web-aplikacija koje imaju jedinstvene sigurnosne ranjivosti.

4.1.1. Prepisivanje spremnika

Sigurnosne ranjivosti prepisivanja spremnika (engl. *buffer overflow*) postale su najozloglašenije sigurnosne ranjivosti. Iako su detalji o sigurnosnim ranjivostima prepisivanja spremnika i metodama iskorištavanja ovih ranjivosti poznati već preko 15 godina, ove se ranjivosti i dalje pojavljuju na svakodnevnoj bazi. Kada se pogleda povijest kritičnih sigurnosnih ranjivosti u operacijskim sustavima i aplikacijama, odnosno servisima, može se vidjeti da su upravo sigurnosne ranjivosti prepisivanja spremnika na prvom mjestu po brojnosti.

Prepisivanje spremnika nastaje kada aplikacija (ili operacijski sustav, odnosno bilo koje sučelje kojem je korisnik u stanju dati ulazne podatke) prihvaća preveliku količinu podataka.

Sam spremnik (engl. *buffer*) predstavlja alocirani segment memorije koji je definirane, ograničene veličine. Primjerice, kada programer aplikacije želi alocirati segment memorije kako bi u njega pohranio neke varijable (npr. korisničko ime), ovisno o programskom jeziku u kojem se razvija aplikacija, sam mora specificirati koliko će velik biti navedeni memorijski segment.

Primjer koda u C programskom jeziku koji alocira devet okteta memorije za pohranu korisničkog imena (osam okteta za pohranu imena i zadnji oktet memorije za oktet čiji će sadržaj biti 0, kako bi se označio kraj znakovnog niza) dan je nastavku:

```
#include <stdio.h>
int main(int argc, char **argv) {
      char username[9];
      strcpy(username, „test“);
      printf(„Korisnicko ime je %s.\n“, username);
      return 0;
}
```

Kao što se u primjeru može vidjeti, program alocira varijablu pod imenom „username“, kopira u nju tekst „test“ pomoću funkcije strcpy() te na kraju ispisuje korisničko ime pomoću funkcije printf().

Rezultat izvođenja ovog programa je, očekivano:

```
Korisnicko ime je test.
```

Ključan problem navedenog programa je upravo u upotrebi funkcije strcpy(). Navedena funkcija kao argumente prima ciljni spremnik (u ovom slučaju *username*) te spremnik iz kojeg se podaci kopiraju (u ovom slučaju doslovno znakovni niz „test“). Ako u potonjem programu modificiramo redak strcpy()na sljedeći način:

```
strcpy(username, “Ovo je dugacki niz znakova”);
```

rezultat izvođenja programa bit će sljedeći:

```
Korisnicko ime je Ovo je dugacki niz znakova.
Segmentation fault
```

Segmentation fault predstavlja grešku u pristupu memoriji programa – riječ je o sigurnosnoj funkciji operacijskog sustava (u ovom primjeru Linuxa) koji je uočio da program pokušava pristupiti nedopuštenoj memoriji te prekinuo njegovo izvođenje. Jednak bi rezultat bio i na drugim operacijskim sustavima poput Windowsa.

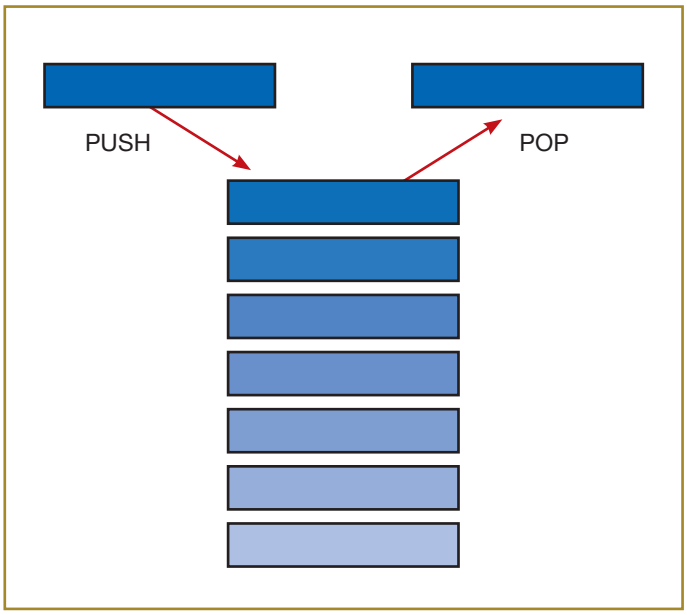
Zašto se dogodila ova greška? Budući da funkcija strcpy() uopće ne provjerava je li ciljni spremnik dovoljno velik za primanje željenog ulaznog niza, ona je samo uzela navedeni dugački niz znakova i prepisala memorijski spremnik varijable *username*. Budući da je operacijski sustav za ovaj memorijski spremnik alocirao samo devet okteta (koliko je programer izvorno tražio), funkcija strcpy() je zapravo „nesvjesno“ prepisala i neki drugi dio memorije koji je doveo do rušenja programa sa *Segmentation fault* greškom.

Prikazani primjer samo jednostavno pokazuje kako je lagano napraviti grešku koja dovodi do mogućnosti prepisivanja spremnika. Iako je u primjeru kopiran statički niz znakova, u pravoj aplikaciji ulazni niz koji se kopira u spremnik

username može biti doslovno korisničko ime koje upisuje napadač, npr. putem sučelja aplikacije.

Pravi sigurnosni rizik koji nastaje od napada prepisivanjem spremnika postaje jasan kada se pogleda slika memorije koju operacijski sustav stvara pri ovakvoj alokaciji varijabli. Ključan dio memorije predstavlja stog (engl. *stack*) na koji operacijski sustav pohranjuje lokalne varijable u funkcijama, kao i adrese na koje se program vraća nakon izvedene funkcije. Dakle, kada u programu programer poziva neku funkciju, adresu na kojoj je program bio operacijski sustav pohranjuje na stog te skače na funkciju. Kada je sav programski kod u funkciji izvršen, operacijski sustav uzima pohranjenu adresu sa stoga da bi znao gdje se treba vratiti te nastavlja izvođenje izvornog koda koji je pozvao funkciju.

Budući da je stog tzv. LIFO struktura (engl. *Last-In-First-Out*), operacijski sustav može jednostavno postavljati bilo kakve vrijednosti na stog (obično tzv. *Push* instrukcijom) i znati da će uzimanje sa stoga (obično tzv. *Pop* instrukcijom) uvijek vratiti zadnju varijablu ili adresu koja je bila postavljena na stog. Stog je jednostavno prikazan na sljedećoj slici.

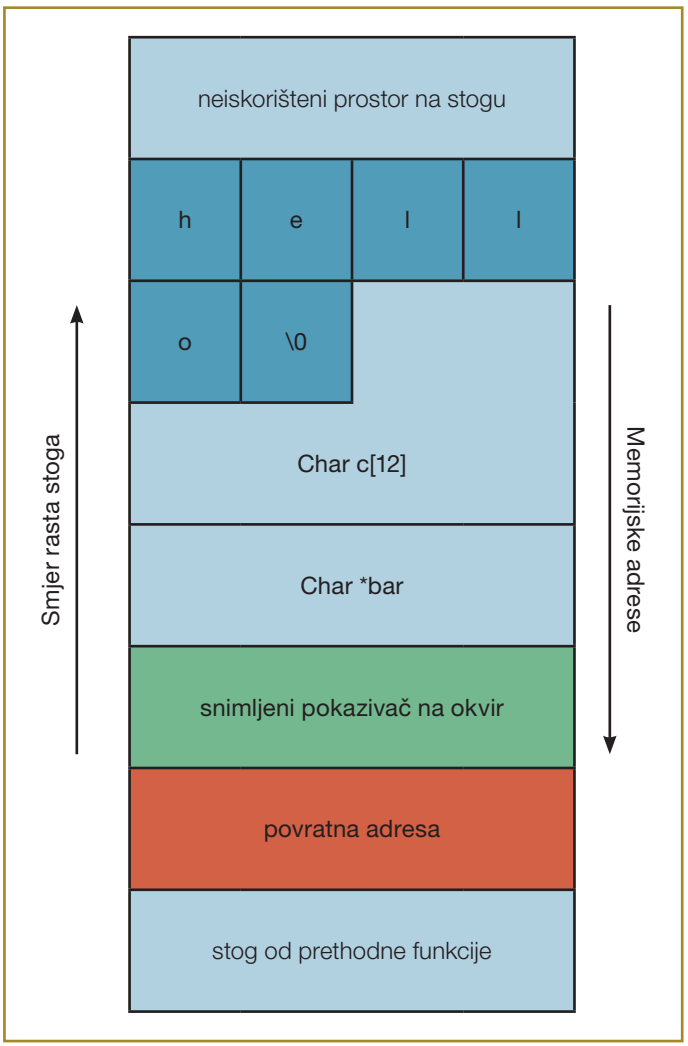


Slika 4.1. Način rada stoga

Još jedna osobina stoga koju je potrebno naglasiti na današnjim Intel temeljenim računalima je da stog u memoriji raste od viših memorijskih adresa prema nižima, dok sve druge varijable rastu od nižih memorijskih adresa prema višima. Na taj način, kada naredba strcpy() prepisuje varijablu koja je bila locirana

u memoriji, moguće je prepisati i ostatak sadržaja stoga. Sjetimo se da je na stogu bila pohranjena i adresa na koju se operacijski sustav treba vratiti što znači da vrlo pažljivim računanjem i postavljanjem određenih podataka napadač može „natjerati“ operacijski sustav da izvrši proizvoljnu adresu čime u konačnici preuzima potpunu kontrolu nad izvršavanjem aplikacije i može izvesti proizvoljni programski kod.

Na sljedećoj je slici prikazan izgled stoga gdje se može vidjeti da prepisivanje sadržaja varijable c (u ovom primjeru) u konačnici može rezultirati i prepisivanjem povratne adrese koja je označena crvenom bojom.



Slika 4.2. Napad prepisivanjem spremnika

Budući da se aplikacije ranjive na napad prepisivanjem spremnika pojavljuju na praktički dnevnoj bazi, jasno je da ovakve sigurnosne ranjivosti predstavljaju veliki problem, pogotovo što, kao što se moglo vidjeti, napadaču omogućuju potpuno preuzimanje kontrole nad aplikacijom.

Zbog toga su proizvođači operacijskih sustava, kompajlera, ali i samog hardvera omogućili određene mjere zaštite od ovih iznimno opasnih sigurnosnih ranjivosti.

4.1.1.1 Zaštita varijabli stoga

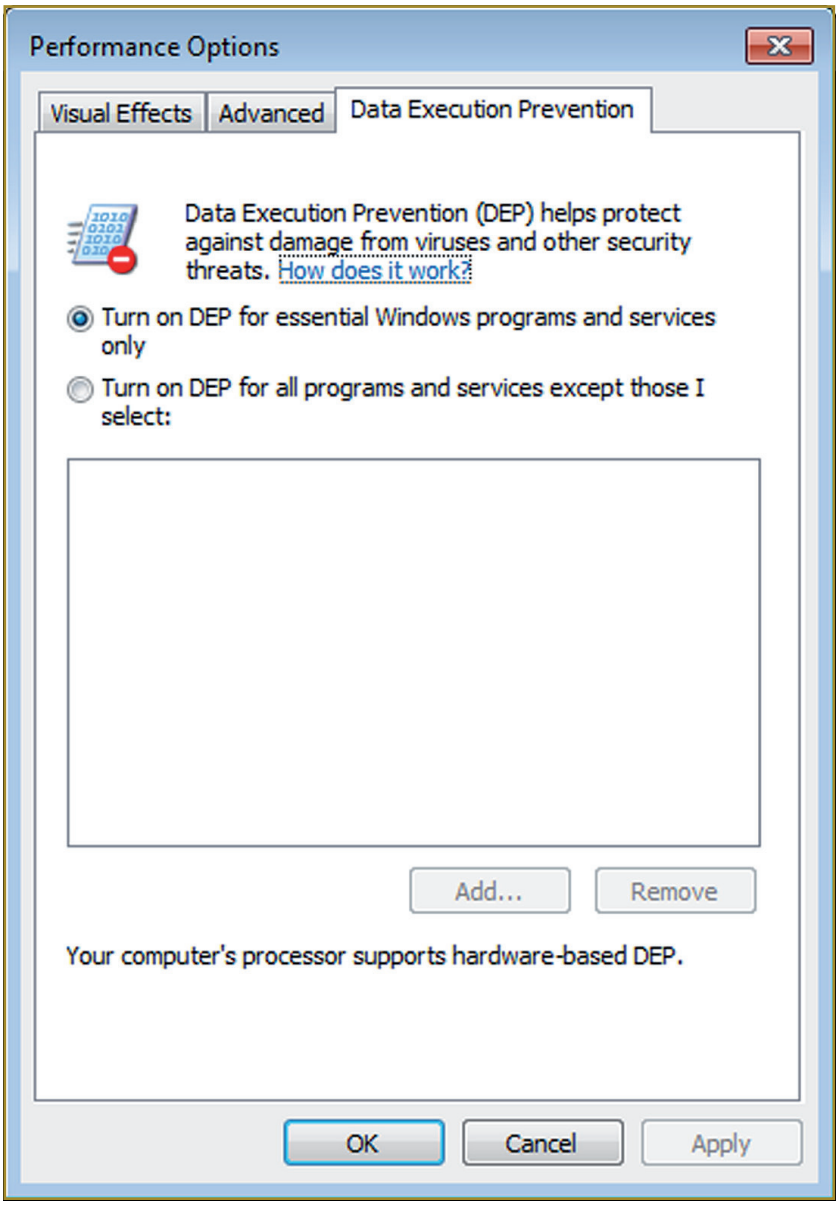
Kao što se može vidjeti, za uspješno iskorištavanje ranjivosti potrebno je prepisati varijable na stogu. Jednako tako, zbog velikog broja ovakvih programerskih grešaka, proizvođači kompajlera dodali su mogućnost zaštite varijabli stoga gdje prevodioci (engl. *compiler*) na stog ubacuju slučajne vrijednosti (engl. *stack cannary*) tijekom izvođenja programa i iste provjeravaju. Ako su podaci na stogu prepisani, biti će prepisane i te slučajne vrijednosti tako da se napad može detektirati i zaustaviti.

Microsoft Visual Studio prevodilac podržava /GS opciju koja automatski štiti varijable stoga.

4.1.1.2 Isključive memorijske dozvole

Od 2003. gotovo svi procesori koji se upotrebljavaju omogućuju postavljanje isključivih memorijskih dozvola na pojedine memorijske stranice. Memorijske dozvole uključuju čitanje, pisanje ili izvršavanje pojedinih stranica memorije. Drugim riječima, ako programer (ili kompajler) unaprijed zna da će u određenoj memorijskoj stranici biti uvijek pohranjeni samo podaci i da se u njoj nikada ne može nalaziti programski kod, moguće je postaviti memorijsku dozvolu samo na čitanje i pisanje te memorijske stranice. Ako se sada proba izvršiti neki memorijski kod iz te stranice, procesor će na hardverskoj razini to zabraniti.

Na ovaj način funkcionira hardverska zaštita koju podržava većina modernih operacijskih sustava – DEP (engl. *Data Execution Prevention*). Počevši od Windows XP, Microsoft podržava DEP i u softverskom i u hardverskom načinu rada. Osnovne postavke ovih operacijskih sustava uključuju DEP samo za kritične Windows aplikacije i servise, kao što se može vidjeti na sljedećoj slici, dok bilo koja druga aplikacija može pri pokretanju zatražiti od operacijskog sustava uključivanje DEP-a.



Slika 4.3. DEP funkcija Windows operacijskih sustava

Potrebno je napomenuti da navedene metode zaštite od napada prepisivanjem spremnika ne jamče sigurnost već samo podižu razinu sigurnosti. U konačnici, ranjivi programski kod uvijek je potrebno promijeniti tako da se ranjivost ukloni dok sve dodatne metode zaštite mogu pomoći u sprječavanju napadača, ali ga ne mogu i ultimativno zaustaviti.

4.2. Sigurnosne ranjivosti web-aplikacija

Veliki broj korisnika danas čak poistovjećuje internet s webom. Činjenica je da web-aplikacije danas predstavljaju sučelje prema raznim servisima za veliku većinu korisnika, pa je čak vidljiv i trend migriranja starih aplikacija na web-temeljena sučelja. Web-aplikacije nude brojne prednosti od kojih je vjerojatno najbitnija činjenica da su dostupne od bilo gdje bez potrebe za instalacijom ikakvih komponenti na lokalno računalo osim web-preglednika.

Upravno navedena dostupnost od bilo gdje predstavlja i veliki izazov koji se stavlja pred zaštitu web-aplikacija. Naime, ako se želi dopustiti pristup s bilo kojeg mjesta na internetu, jasno je da tradicionalni sustavi zaštite poput vatrozida više ne rade nikakvu razliku budući da se svim korisnicima interneta treba dopustiti pristup na web-aplikaciju, koja je zatim sama odgovorna za implementirane sigurnosne kontrole, autentikaciju i autorizaciju korisnika.

Web-aplikacije danas se mogu razvijati u čitavom nizu programskih jezika, a najpopularniji su Microsoftov .NET koji omogućuje razvijanje u podržanim jezicima poput C# ili Visual Basica, Java i PHP.

Iako svako od navedenih razvojnih okruženja ima neke specifične sigurnosne probleme, ranjivosti navedene u nastavku ovog poglavlja svojstvene su za sve web-aplikacije, bez obzira na programski jezik u kojem su napisane.

Ako se sjetimo ultimativnog cilja napadača koji je dolazak do osjetljivih podataka, jasno je da web-aplikacije predstavljaju još jedno sučelje koje napadaču omogućuje ostvarenje svojih ciljeva. Naime, bez obzira na format u kojem su podaci pohranjeni u pozadini (npr. obične tekstualne datoteke ili tablice u bazama podataka), ako napadač uspije kontrolirati web-aplikaciju, može doći i do željenih podataka.

Još jedna značajka web-aplikacija koju je potrebno spomenuti jesu sjednice (engl. *sessions*). Sjednice omogućuju identifikaciju korisnika koji pristupaju web-aplikaciji. Naime, kada novi korisnik otvori svoj web-preglednik te se prijavi na aplikaciju, zbog inherentnih ograničenja protokola poput HTTP i HTTPS, koji služe pristupu web-aplikacijama, web-aplikacija mora sama voditi brigu o tome koji je točno korisnik prijavljen. Današnje web-aplikacije obično podatke o sjednicama pohranjuju u kolačiće (engl. *cookies*), posebne datoteke koje podržava svaki web-preglednik i šalje ih pri svakom upitu. Dakle, nakon što se korisnik prijavio na web-aplikaciju, ona otvara novu sjednicu za njega i podatke o sjednici pohranjuje u kolačić koji se šalje natrag korisnikovom web-pregledniku u odgovoru.

Web-preglednik zatim pri bilo kojem sljedećem upitu, odnosno kliku korisnika na web-aplikaciju šalje ovaj isti kolačić natrag web-aplikaciji. Na osnovi podataka u kolačiću web-aplikacija može zaključiti koji je korisnik poslao navedeni upit.

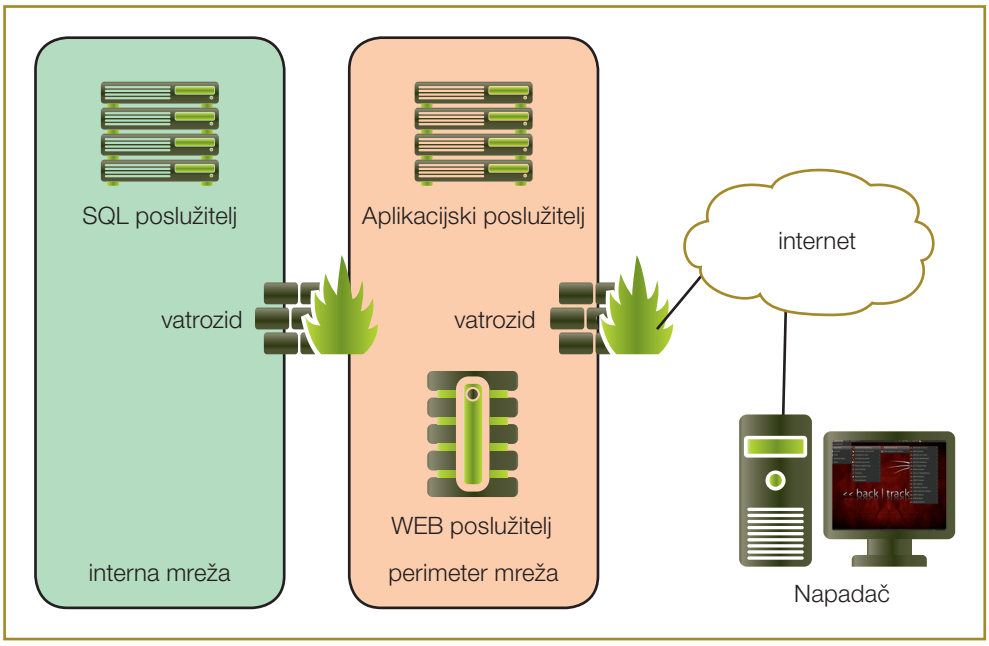
Budući da su HTTP i HTTPS protokoli bez stanja (tzv. engl. *stateless protocols*), web-aplikacije moraju se osloniti na kolačiće. Ovdje se može vidjeti i da se nakon početne prijave korisnika (recimo prijave korisničkim imenom i zaporkom) svi budući upiti korisnika definiraju samo prema sadržaju kolačića. Drugim riječima, kada se u web-aplikaciju istodobno prijave npr. običan korisnik i administrator (koji ima veća prava nad aplikacijom), jedina razlika između njihovih upita je sadržaj kolačića koji postaje sigurnosno vrlo bitan – sama krađa kolačića može korisniku dati pristup aplikaciji pod tuđim korisničkim računom. Kao što se može naslutiti, veliki broj napada na web-aplikacije usmjeren je upravo na kolačiće.

OWASP organizacija (engl. *Open Web Application Security Project*) već godinama prati i objavljuje metode napada na web-aplikacije. Iz njihovog se godišnjeg izvještaja može vidjeti da su dvije kategorije ranjivosti prisutne u najvećem broju web-aplikacija: umetanje proizvoljnog koda i tzv. XSS (engl. *Cross Site Scripting*) ranjivosti. Oba su tipa ranjivosti vrlo opasna i napadaču mogu omogućiti preuzimanje kontrole nad aplikacijom ili čak i izvođenje proizvoljnog koda.

4.2.1. Ranjivosti umetanjem proizvoljnog koda

Iako su sigurnosne ranjivosti umetanjem proizvoljnog programskog koda generičke te u ovu grupu spadaju sve sigurnosne ranjivosti koje napadaču omogućuju umetanje vlastitog koda i na taj način kontroliranja rada aplikacije, kod web-aplikacija ovdje je najčešće riječ o sigurnosnim ranjivostima umetanja SQL naredbi, tzv. engl. *SQL Injection* sigurnosne ranjivosti.

SQL Injection ranjivosti polaze od činjenice da velika većina web-aplikacija podatke pohranjuje u neku bazu podataka. Kada korisnici pristupaju web-aplikaciji, podaci se dohvaćaju i pohranjuju u navedenu bazu podataka. Baze podataka predstavljaju vrlo jednostavno i prihvatljivo rješenje za rukovanje s velikim brojem podataka pa danas programeri u baze podataka pohranjuju gotovo sve podatke, počevši od korisničkih imena i zaporki pa sve do ostalih osjetljivih podataka. Ovakav rad, odnosno implementacija i integracija web-aplikacije, poslužitelja i baze podataka zapravo su vrlo standardni i prikazani su na sljedećoj slici.



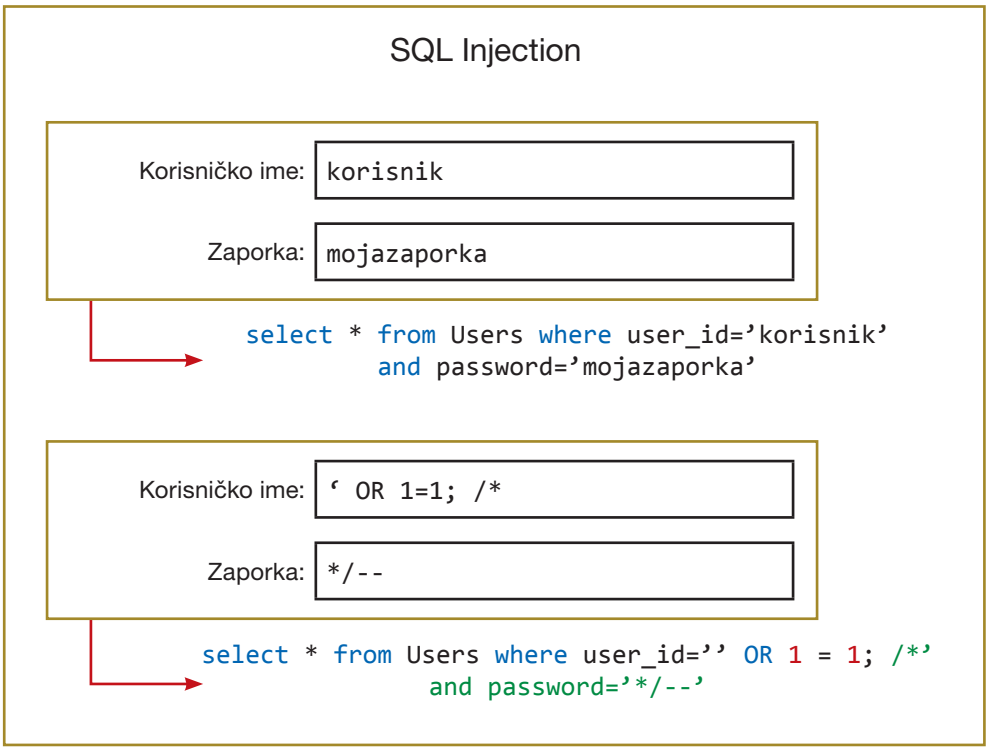
Slika 4.4. Klasična komunikacija klijent, web-aplikacija, baza podataka

U ovom slučaju, sigurnosnih mehanizmi podatkovnog sloja, koji su objašnjeni u poglavlju 3, moraju biti implementirani unutar same web-aplikacije, budući da baza podataka klasično daje potpuni pristup svim pohranjenim podacima.

SQL Injection sigurnosne ranjivosti predstavljaju vrstu ranjivosti web-aplikacija u kojima je napadač u mogućnosti modificirati SQL upit koji izvodi web-aplikacija nad podacima pohranjenim u bazi podataka.

Ove sigurnosne ranjivosti nastaju kada web-aplikacija prihvaća ulazne podatke dobivene od korisnika i ne provodi nikakve (ili nedovoljne) sigurnosne kontrole nad ovim podacima prije njihove upotrebe u SQL upitima koji se šalju na bazu podataka. Drugim riječima, napadač koji može poslati ulazne podatke web-aplikaciji može vrlo pažljivim odabirom ulaznih podataka modificirati SQL upit koji će se izvršiti upravo zahvaljujući nikakvim ili nedovoljnim sigurnosnim provjerama ulaznih podataka i njihovom izravnom upotrebom u formiranju SQL upita.

Na sljedećoj je slici prikazan SQL upit koji u izvornom obliku uzima korisničko ime i zaporku koju je korisnik unio putem web-forme aplikacije. Aplikacija ima unaprijed definiran SQL upit čiji je cilj provjeriti postoje li navedeni korisnik i upisana zaporka u bazi podataka, čime bi se potvrdio autentikacijski proces korisnika koji se želi prijaviti na web-aplikaciju.



Slika 4.5. Primjer SQL Injection napada

Kao što se može vidjeti na potonjem primjeru, pažljivim unosom proizvoljnih podataka napadač može modificirati SQL upit koji će na kraju biti izvršen na bazi podataka. Prikazani SQL upit omogućit će napadaču da se na web-aplikaciju prijavi kao prvi korisnik u bazi podataka, bez potrebe za poznavanjem korisničkog imena ili zaporke, samo na osnovi toga što će unesena ILI operacija uvijek rezultirati istinom!

Ovakvim pažljivim modificiranjem SQL upita napadač može čitati, modificirati ili stvarati proizvoljne zapise u bazi podataka ili čak i izvršavati programski kod, ako mu to sama baza podataka omogućuje. Kako je u ovom slučaju dolazak do osjetljivih podataka ultimativni cilj napadača, može se vidjeti da kroz web-aplikaciju s nedovoljnim sigurnosnim kontrolama napadač može pročitati bilo koje podatke iz baze podataka te tako ostvariti svoj cilj.

Sprječavanje umetanja proizvoljnog koda

Osnovna metoda sprječavanja umetanja proizvoljnog koda, ali i drugih napada koji su zasnovani na procesiranju ulaznih podataka koje postavlja korisnik, jest kontrola tih podataka (engl. *input validation*).

Pri svakoj obradi ulaznih podataka koji su zaprimljeni od korisnika, aplikacija bi trebala provjeriti zadovoljavaju li primljeni ulazni podaci određena pravila, odnosno ograničenja:

- provjera duljine ulaznih podataka spriječit će provođenje napada poput prepisivanja podataka u spremniku gdje korisnik unosi više ulaznih podataka nego što aplikacije očekuje;
- provjera tipa uzlaznih podatka spriječit će napade koji pokušavaju uzrokovati greške u aplikaciji unošenjem pogrešnog tipa podataka (npr. znakovni niz umjesto broja);
- provjera sintakse ulaznih podataka spriječit će napade koji se zasnivaju na podacima koji ne odgovaraju onima koje aplikacija očekuje, poput prethodno prikazanog primjera *SQL Injection* napada. Na primjer, ako korisničko ime ne može sadržavati znak ‘ koji predstavlja opasan znak za SQL upit, potrebno je dodati kontrolu koja će detektirati pokušaj upotrebe ovakvog znaka i spriječiti napad. Ako je taj znak potrebno upotrebljavati, moguće je upotrijebiti druge metode generiranja SQL upita koje nisu ranjive na *SQL Injection* napade;
- konačno, potrebno je provjeriti odgovaraju li ulazni podaci svrsi koju aplikacija treba. Primjerice, ako je riječ o aplikaciji koja prenosi novac s jednog računa na drugi, potrebno je provjeriti pokušava li korisnik prenijeti npr. negativnu količinu novca, što bi rezultiralo krađom novca s ciljnog računa.

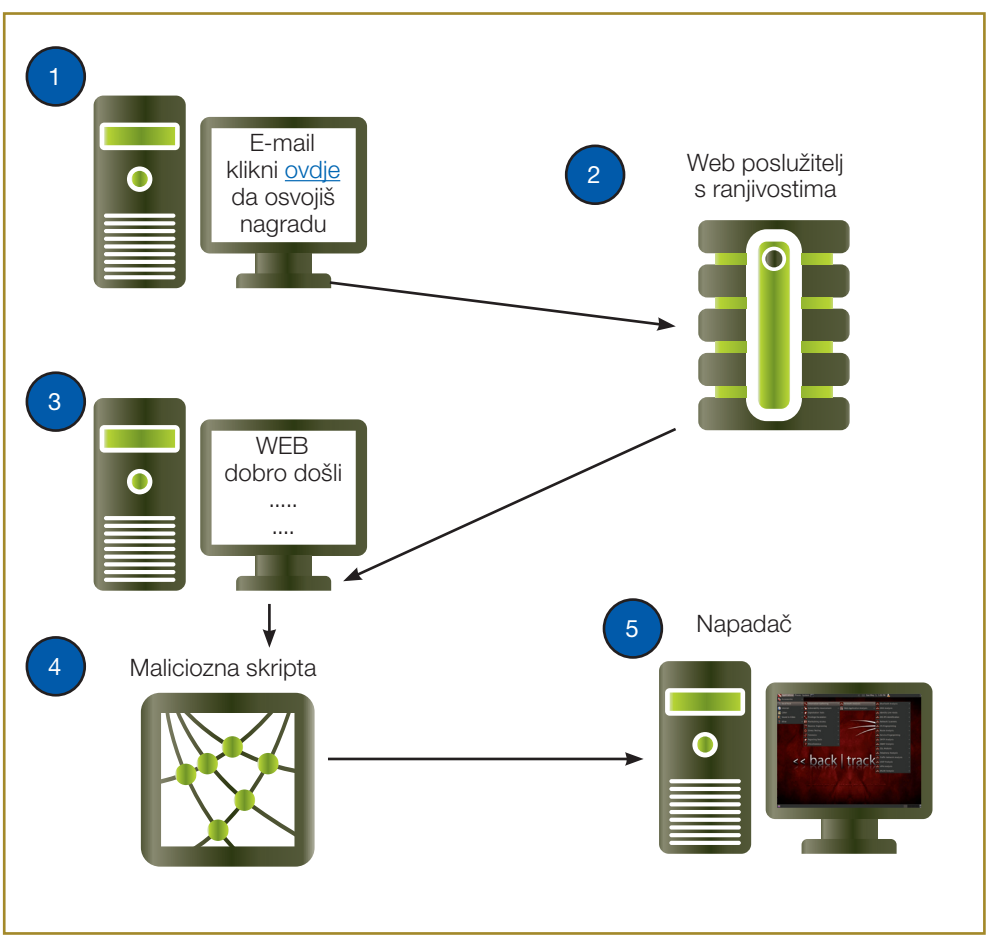
4.2.2. Cross Site Scripting (XSS) ranjivosti

Cross Site Scripting napadi predstavljaju najčešće napade na korisnike web-aplikacija. Cilj XSS napada je napasti krajnjeg korisnika, a ne izravno web-aplikaciju te su zbog toga specifični. Ako se pitate koja je svrha napadanja krajnjeg korisnika, sjetimo se da njegov web-preglednik informacije o sjednici pohranjuje u kolačiću, koji je zapravo obična tekstualna datoteka. Ako napadač uspije ukrasti sadržaj kolačića, može se web-aplikaciji predstaviti kao korisnik čiji je kolačić ukrao i time pristupiti osjetljivim podacima.

Postoji više kategorija XSS napada, no u najvećem je broju slučajeva riječ o reflektirajućem XSS napadu. Kod ovog napada napadač polazi od činjenice da web-aplikacija uzima neke ulazne podatke od korisnika pri slanju upita i onda ih prikazuje natrag, bez provođenja ikakvih sigurnosnih kontrola. Na taj način napadač može u upit zapravo ubaciti proizvoljni maliciozni kod u vidu JavaScript ili VBScripta koji će biti prikazan natrag korisniku. Ovako ubačen maliciozni kod bit će izvršen u korisnikovom web-pregledniku (budući da web-preglednik

taj kod vidi kao da ga je dobio u odgovoru od web-aplikacije). Naravno, budući da sada napadač može izvesti praktički bilo koji JavaScript ili VBScript kod u web-pregledniku, može vrlo jednostavno ukrasti sadržaj kolačića, a time i cijelu sjednicu.

Iskorištavanje XSS ranjivosti klasično, dakle od napadača, zahtijeva izradu posebne URL adrese koja poziva ranjivu web-aplikaciju te u sebi sadržava maliciozni kod. No još se uvijek postavlja pitanje kako navesti korisnika da posjeti tu malicioznu URL adresu. Napadači danas u tu svrhu najčešće rabe lažno napravljene poruke elektroničke pošte ili čak postavljaju maliciozne URL adrese na druge web-stranice na internetu (npr. popularne blogove i slično), ne bi li korisnika naveli na posjećivanje iste. Metoda napada putem poruke elektroničke pošte prikazana je na sljedećoj slici.



Slika 4.6. XSS napad putem poruke elektroničke pošte

4.2.2.1 Sprječavanje XSS sigurnosnih ranjivosti

Budući da se XSS sigurnosne ranjivosti temelje na činjenici da napadač može poslati proizvoljni ulazni niz web-aplikaciji koja će ga zatim prikazati posjetitelju, osnovna metoda zaštite je pravilno enkodiranje izlaznih znakova. Naime, znakovi poput < i > ključni su za HTML elemente.

U slučaju da je ove znakove potrebno prikazati na ekranu, njih web-aplikacija treba ispravno enkodirati, i to znak < u &lt, a znak > u &gt. Na taj način web-preglednik bilo kakav tekst između ovih znakova neće interpretirati kao HTML te će se jednostavno spriječiti mogućnost ubacivanja malicioznih JavaScript programa i sličnog.

Osim navedenih znakova potrebno je slično enkodirati i druge osjetljive znakove koji uključuju:

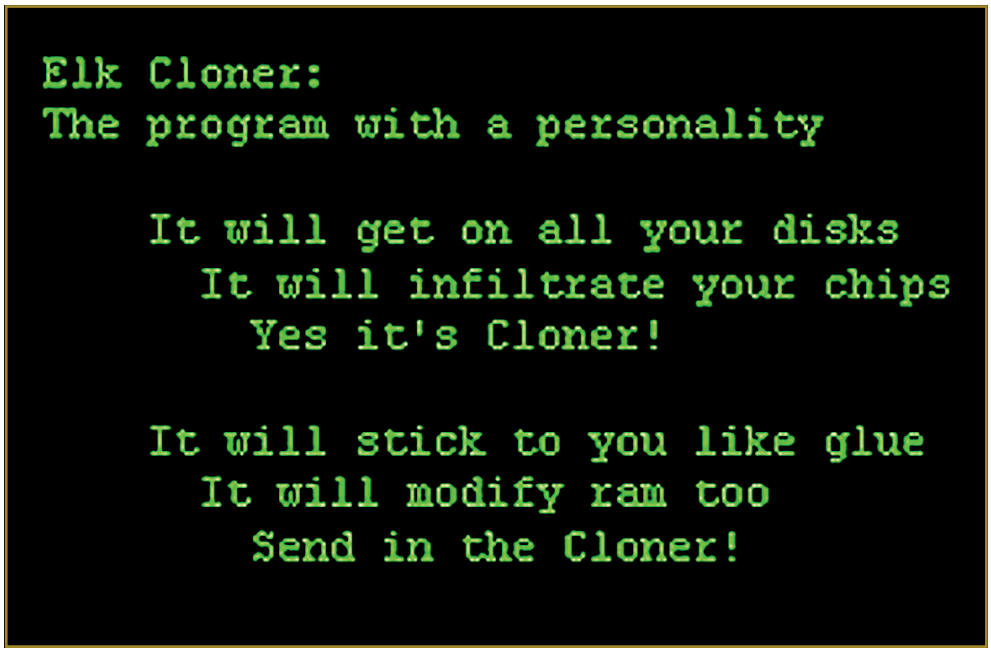
< > „ , ‘ &

4.3. Maliciozni programi

Maliciozni programi danas nesumnjivo predstavljaju najveći izazov koji se postavlja pred zaštitu informacijskih sustava. Ovi se programi, kao što im i ime govori, ne razlikuju ni po čemu od drugih legitimnih programa osim po svojoj svrsi i namjeni koja je maliciozna: maliciozni programi (engl. *malicious programs*, *malware*) uz ostale aktivnosti koje provode na neki način narušavaju jednu ili više osnovnih sigurnosnih komponenti, CIA trokut (više o CIA trokutu bilo je rečeno u poglavlju 1.1).

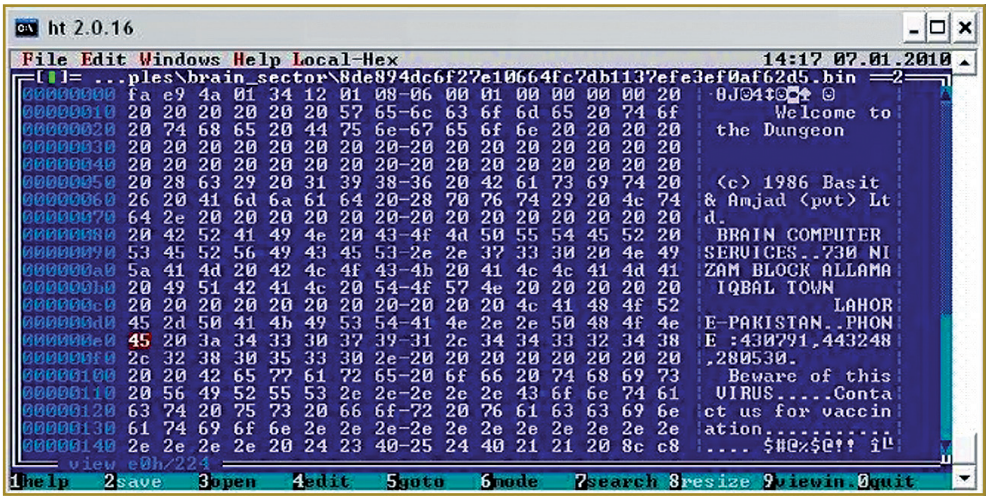
Povijesno gledajući, prvi su se maliciozni programi u vidu virusa pojavili još osamdesetih godina prošlog stoljeća. Tada je uglavnom bila riječ o nekakvim eksperimentalnim programima gdje su programeri htjeli vidjeti što se događa kada napišu programski kod koji se sam može replicirati (što je jedna od glavnih značajki virusa, kao što je objašnjeno u poglavlju 4.3.1) ili je jednostavno bila riječ o šalama ili podvalama.

Tako ne treba čuditi činjenica da je zapravo prvi raširenији virus bio napisan za Apple II računala koja su upravo osamdesetih godina bila dominantna kućna računala. Riječ je o virusu pod imenom Elk Cloner koji se samo širio preko disketa te pri svakom pedesetom pokretanju prikazao jednostavnu pjesmicu, kao što se može vidjeti na sljedećoj slici.



Slika 4.7. Poruka Elk Cloner virusa, jednog od prvih značajnih virusa

Prvi raširen virus za PC računala bio je Brain. Riječ je bila o još jednom virusu koji se širio preko disketa (tzv. boot sektor virus). Zanimljivo je da je Brain izvorni boot sektor pomaknuo na neiskorišten dio diskete te ove sektore označio kao neispravne. U modificiranom boot sektoru virus je ostavio poruku, uključujući čak i punu adresu autora virusa, kao što se može vidjeti na sljedećoj slici.



Slika 4.8. Boot sektor diskete inficirane Brain virusom

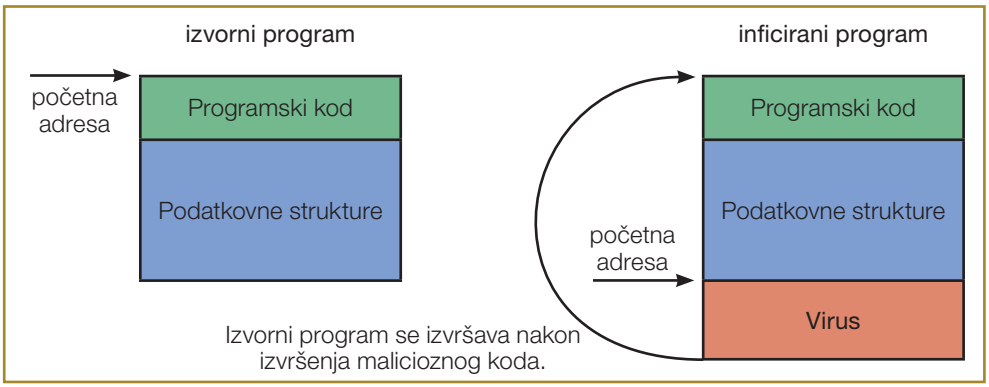
Početkom ovog stoljeća slika malicioznih programa znatno se promijenila. Veliki utjecaj interneta, međusobna povezanost računala i jednostavno popularnost i dostupnost kućnih računala omogućili su nove metode širenja malicioznih programa što je rezultiralo pravom eksplozijom broja malicioznih programa. Jednako tako, sve veća ekspanzija elektroničkog poslovanja i pohranjivanje osjetljivih podataka ne samo na poslužitelje već i na osobna računala otvorili su nove mogućnosti. Napadači su brzo prepoznali moguću korist krađe i manipulacije ovakvih podataka tako da su se maliciozni programi zadnjih godina počeli upotrebljavati gotovo isključivo u kriminalne svrhe. Kriminalni su krugovi napravili cijeli ekosustav koji uključuje programere posebnih malicioznih programa kojima je cilj krađa informacija te posebne kriminalne grupe koje se nakon toga bave „unovčavanjem“ ovih informacija, bilo da je riječ o doslovno ukradenim brojevima kreditnih kartica ili privatnim podacima korisnika.

Iako se danas bilježe doslovno desetci, ako ne i stotine milijuna različitih primjeraka malicioznih programa, općenito ih je moguće svrstati u nekoliko grupa prikazanih u nastavku, s naglaskom da u velikom broju slučajeva maliciozni programi zapravo dijele značajke više grupa.

- Virusi
- Crvi
- Trojanski konji
- Spyware i adware programi

4.3.1. Virusi

Glavna značajka virusa je činjenica da se njegovo cijelo tijelo “priljepljuje” (inficira) na legitimni program, bilo da je riječ o drugom programu ili fizičkim sektorima na nekom mediju (npr. tvrdom disku ili disketi). Drugim riječima, virus je u stanju identificirati početnu/ulaznu adresu programa koji se inficira, dodati svoje tijelo na kraj (ili na neku drugu lokaciju, gdje ima dovoljno prostora, ovisno o pojedinoj inačici virusa) tog programa te promijeniti početnu/ulaznu adresu programa tako da se pri pokretanju prvo pokrene sam virus, kao što je pokazano na priloženoj slici.



Slika 4.9. Načelo rada virusa

Osim obilježja virusa da inficira druge datoteke, još jedna ključna značajka ovog tipa malicioznih programa je da se mogu širiti samostalno, dakle da za širenje ne trebaju nikakvu aktivnost korisnika (osim pokretanja inficirane datoteke, što se u slučaju npr. boot sektor virusa može svesti samo na pokretanje sustava upotrebom navedene diskete).

Prema načinu inficiranja drugih datoteka, viruse dijelimo na sljedeće kategorije:

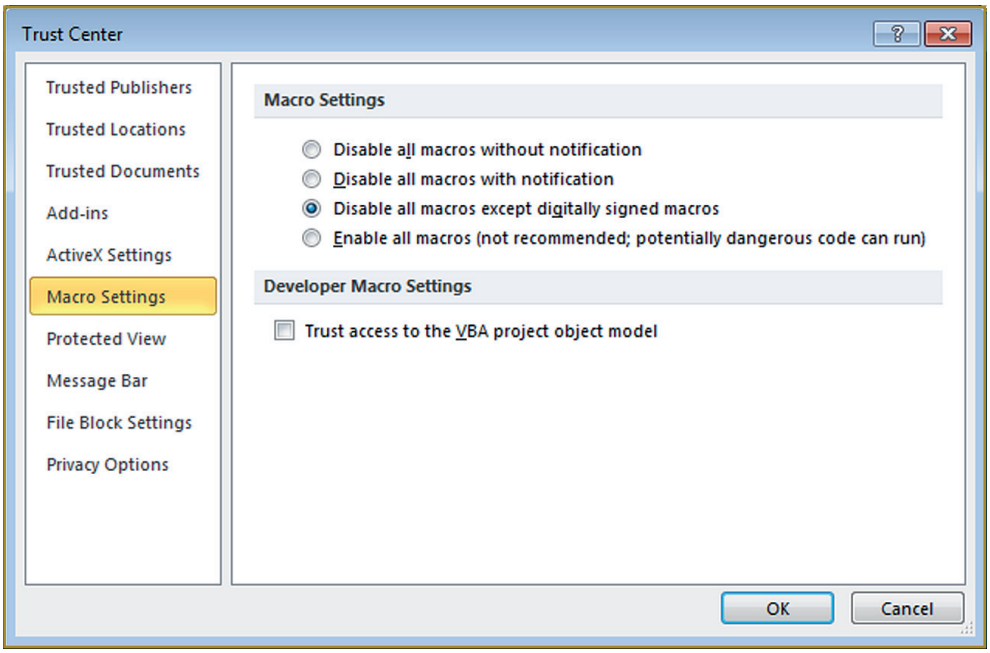
- viruse koji inficiraju datoteke (engl. *file infector viruses*) – rade na načelu prikazanom na slici 38 te su u stanju inficirati sve izvršne datoteke poput EXE, COM pa čak i DLL datoteka na Windows operacijskim sustavima. Jednako je tako potrebno napomenuti da virusi postoje za praktički sve platforme tako da je moguće naći viruse koji inficiraju datoteke i na Linux operacijskim sustavima, u kojem slučaju inficiraju tzv. ELF datoteke, koje predstavljaju izvršne datoteke na Linux operacijskom sustavu;
- boot sektor virusi – inficiraju boot sektor (prvi sektor) različitih medija. Povijesno gledano, ovo su bili najčešći virusi kada su se programi razmjenjivali na disketama. Budući da BIOS računala prvo učitava boot sektor diskete, virus infekcijom ovog sektora osigurava da će on biti pokrenut prvi.

  Kako su se diskete sve manje upotrebljavale, i broj boot sektor virusa počeo je padati, sve do početka 2008. kada su se počeli pojavljivati boot sektor virusi koji se instaliraju na prvi sektor (engl. *Master Boot Records – MBR*) tvrdih diskova te na taj način opet osiguravaju da će biti pokrenuti prije samog operacijskog sustava. Moderni boot sektor virusi mogu napasti čak i zadnje inačice operacijskih sustava poput Windows 7 i Windows 2008 te često sadržavaju i komponente *rootkita* (više o *rootkitima* u poglavlju 4.3.5), što ih čini naročito opasnim;

- makrovirusi – ova kategorija virusa predstavlja maliciozne programe koji su napisani u makrojezicima. *Macro* jezici predstavljaju vrlo jednostavne programske jezike kojima se može proširiti funkcionalnost nekih dokumenata. Tako npr. Microsoft Office paketi dopuštaju dodavanje *macro* programa dokumentima kako bi se neke aktivnosti unutar dokumenta automatizirale (npr. ispunjavanje formulara), ali omogućuju i čitav niz drugih funkcionalnosti.

  Macro virusi, dakle, ovise o programu nositelju, odnosno programu koji podržava navedeni *macro* jezik i inficiraju dokumente ili predloške kojima se koristi navedeni program nositelj. Najveći broj *macro* virusa napisan je za Microsoft Office programske pakete (Word, Excel i slični), no potrebno je napomenuti da bilo koja aplikacija koja podržava *macro* jezik s određenom funkcionalnošću (otvaranje drugih datoteka i njihovo modificiranje) može biti ranjiva na *macro* viruse. Tako su napisani *macro* virusi za čitav niz aplikacija, uključujući i poslovne aplikacije poput AutoCAD-a ili Microsoftovog Visual Studia.

  Zbog velike rasprostranjenosti Microsoft Office *macro* virusa u novijim inačicama Microsoft Office paketa Microsoft je promijenio osnovni način pokretanja *macroa* te se dopuštaju samo *macroi* koji su ispravno digitalno potpisani, kao što se vidi na sljedećoj slici.



Slika 4.10. Microsoft Office dopušta samo pokretanje macroa koji su digitalno potpisani

Navedene tri kategorije nisu ekskluzivne. Dapače, vrlo je čest slučaj da autori malicioznih programa kombiniraju ove mogućnosti tako da pojedini virus može inficirati druge datoteke, ali i boot sektor tvrdog diska. Ovakve su aktivnosti logične budući da autori malicioznih programa pokušavaju na što je moguće više načina osigurati uspješno rasprostranjivanje virusa koji su napisali.

Virusi koji imaju značajke više podgrupa (npr. inficiraju i datoteke i boot sektore disketa) nazivaju se *multi-partite* virusi.

4.3.2. Crvi

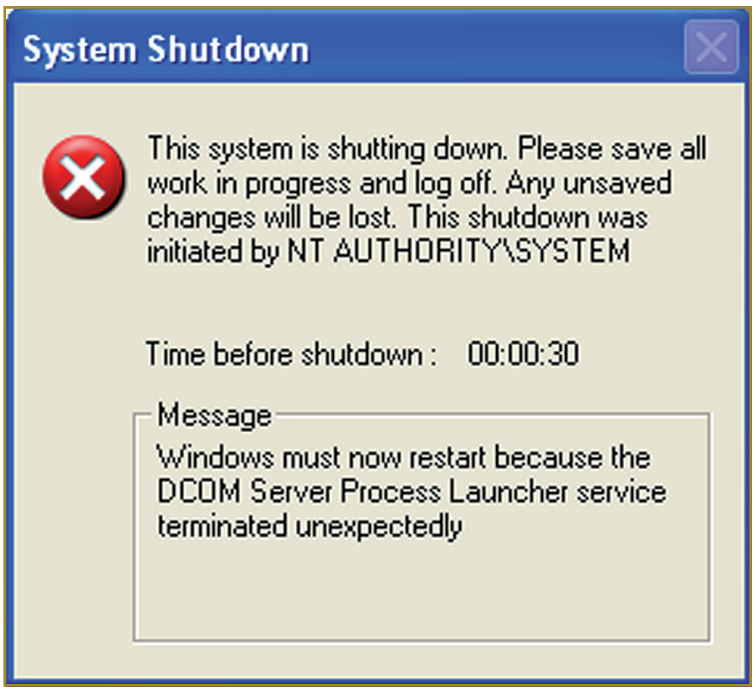
Za razliku od virusa, crvi predstavljaju posebnu kategoriju malicioznih programa koji ne inficiraju druge datoteke. Dakle, crvi su prisutni isključivo kao zasebni maliciozni programi te nikada ne inficiraju druge izvršne datoteke. Još jedna značajka crva jest da se šire putem računalne mreže, i to putem jedne od sljedećih kategorija širenja:

- crvi koji iskorištavaju sigurnosne ranjivosti neispravno (odnosno nesigurno) implementiranih aplikacija ili servisa – primjer ovakvog crva je SDbot – riječ je o crvu koji se u svrhu širenja preko računalne mreže služi slabim administratorskim zaporkama na Windows operacijskim sustavima što mu omogućuje da automatski kopira svoje tijelo preko računalne mreže na udaljeno računalo i pokrene se;
- crvi koji iskorištavaju sigurnosne ranjivosti u mrežnim servisima – riječ je o crvima koji se koriste tehnikama poput prepisivanja podataka (više o ovim tehnikama objašnjeno je u poglavlju 4.1.1). Ovakvi crvi obično pregledavaju cijelu lokalnu računalnu mrežu (ili se pokušavaju spojiti na druga računala preko interneta) te, ako detektiraju ranjivi servis, upotrebljavaju posebno napravljen kod koji im omogućuje kopiranje svog tijela na ranjivo računalo te pokretanje. Na taj je način ranjivo računalo inficirano i crv se širi dalje.

  Jedan od poznatijih crva koji je iskorištavao sigurnosnu ranjivost u Microsoft SQL Server servisu (bazi podataka) i koji je napravio veliku štetu na internetu je svakako SQL Slammer. Riječ je o crvu koji se počeo širiti 2003. iznimno agresivno, toliko da je većina mrežnog prometa na internetu bila usporena. Crv je iskorištavao upravo sigurnosnu ranjivost prepisivanja podataka na stogu. SQL Slammer uspio je usporiti internet čak i više od također poznatog crva pod imenom Code Red, koji je pak iskorištavao sigurnosnu ranjivost u Microsoft IIS web-poslužitelju, a širio se 2001. Ono što je svojstveno jest da su u oba slučaja crvi iskorištavali sigurnosne ranjivosti koje su već bile poznate i za koje je postojala sigurnosna zakrpa koju je Microsoft izdao. Velika uspješnost širenja ovih crva uputila je na

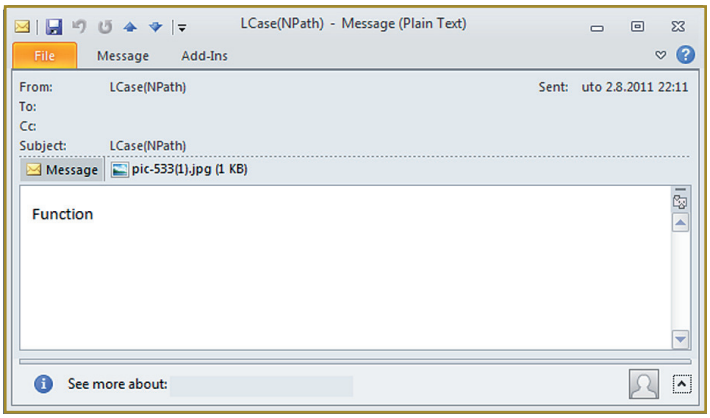
probleme koje tvrtke imaju u instaliranju sigurnosnih zakrpi na poslužitelje, što svakako predstavlja jedan od osnovnih zahtjeva višeslojnog strateškog modela sigurnosti.

Jedna od nuspojava SQL Slammer crva bila je i prisilno rušenje sustava zbog greške u LSASS procesu koji je kritičan proces za Windows operacijski sustav. Greška koja bi se prikazala na ekranu računala prikazana je u nastavku;



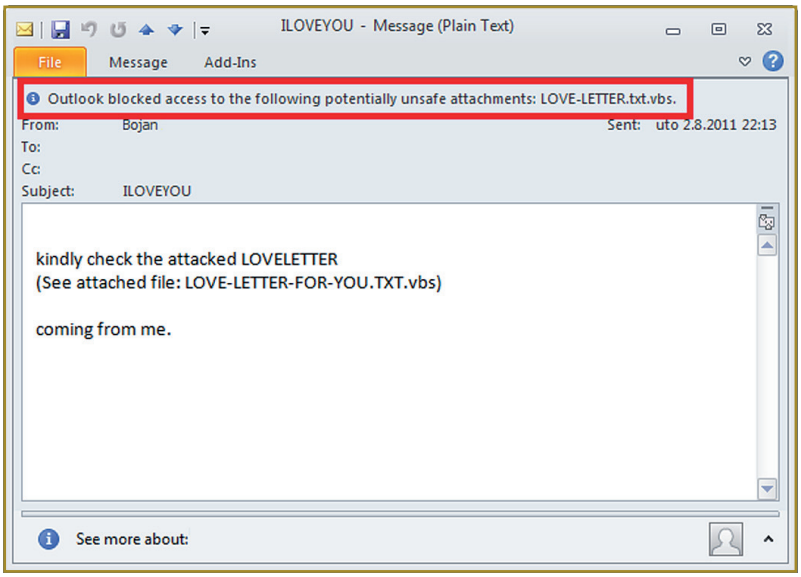
Slika 4.11. Greška u LSASS procesu uzrokuje rušenje operacijskog sustava

- crvi koji iskorištavaju sigurnosne ranjivosti u aplikacijama – riječ je o kategoriji crva koja je vrlo slična prethodnoj, no za razliku od crva koji iskorištavaju sigurnosne ranjivosti u mrežnim servisima, ova kategorija crva napada isključivo aplikacije. To znači da se ovi crvi mogu širiti samo poluautomatski budući da zahtijevaju određenu aktivnost od korisnika.
  Primjer crva iz ove kategorije je crv Klez koji iskorištava sigurnosnu ranjivost u Microsoft Outlook Express aplikaciji za čitanje elektroničke pošte. Pri otvaranju poruke elektroničke pošte koja sadržava crva Klez, crv se automatski pokreće, kupi sve adrese iz adresara Outlook Express aplikacije te se šalje na sve dostupne adrese elektroničke pošte. Na sljedećoj je slici prikazana klasična poruka elektroničke pošte koju šalje crv Klez;

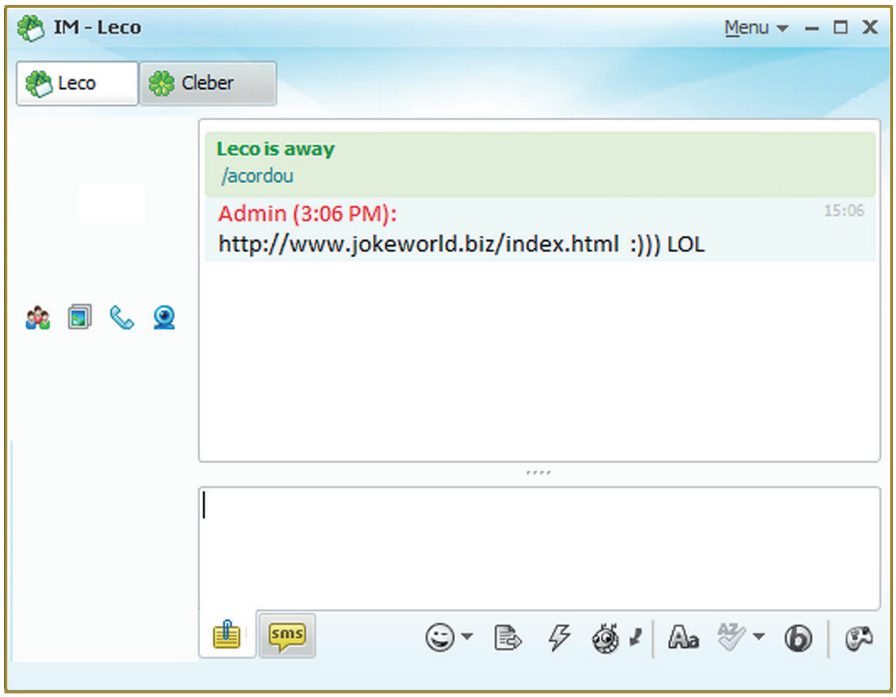


Slika 4.12. Poruka elektroničke pošte generirana od crva Klez

- crvi koji koriste metode socijalnog inženjeringa – zadnju kategoriju crva predstavljaju crvi koji ne iskorištavaju tehničke sigurnosne ranjivosti, već svoje širenje zasnivaju na pokušaju prijevare korisnika, odnosno navođenja korisnika na samostalno pokretanje crva. Crvi u ovoj kategoriji mogu se širiti putem praktički bilo koje metode socijalnog inženjeringa, a vrlo su česti crvi koji se šire putem elektroničke pošte ili pomoću servisa za komuniciranje poput ICQ-a, Skypea ili MSN Messangera. Na sljedećim su slikama dani primjeri dva crva koji se koriste metodom socijalnog inženjeringa.



Slika 4.13. Poznati ILOVEYOU crv navodi korisnika na pokretanje privitka



Slika 4.14. Bizex crv širi se putem ICQ poruka

4.3.3. Trojanski konji

Osnovna značajka malicioznih programa koji spadaju u kategoriju trojanskih konja jest da se nisu uopće u stanju sami širiti. Drugim riječima, trojanski konji svoje širenje zasnivaju isključivo na aktivnostima korisnika, odnosno činjenici da će ih korisnici sami kopirati na ciljna računala i pokrenuti.

Da bi postigli ovakvu metodu širenja, trojanski konji pokušavaju zavarati korisnika te ga uvjeriti da je njihova funkcija benigna, odnosno da nisu maliciozni. Trojanski konji dobili su ime upravo prema povijesnom događaju u Troji, kada su grčki ratnici napravili drvenog trojanskog konja kojeg su stanovnici Troje prihvatili kao miroljubivi poklon. No u trojanskom su konju bili skriveni grčki ratnici koji su po noći, nakon što je konj unesen unutar zidina Troje, otvorili ulazna vrata te omogućili napad.

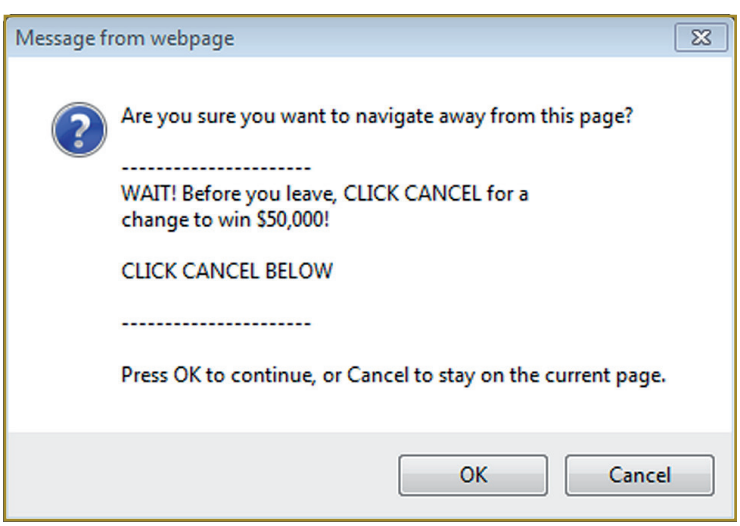
Danas je ova kategorija malicioznih programa iznimno popularna te se zapravo s njom najčešće susrećemo. Trojanski konji vrlo često imaju i posebne module koji napadačima nakon uspješne infekcije računala omogućuju udaljenu kontrolu u svrhu krađe osjetljivih podataka na računalu ili iskorištavanja resursa računala. Ovakvi se moduli obično nazivaju stražnjim vratima (engl. *backdoor*).

4.3.4. Spyware i adware programi

Kategoriju *spyware* i *adware* malicioznih programa zapravo je najteže definirati budući da graniče s trojanskim konjima. Riječ je ponovno o malicioznim programima koji se ne mogu sami širiti, već svoje širenje temelje na korisnicima koji ih sami i pokreću.

Spyware programi tradicionalno pokušavaju prikupiti osjetljive informacije o korisniku inficiranog računala. Prvi *spyware* programi zapravo su bili relativno benigni te su samo bilježili podatke o npr. web-stranicama koje korisnik inficiranog računala posjećuje te su na taj način zadirali u privatnost korisnika, ali nisu krali nikakve osjetljive podatke. Današnji *spyware* programi mogu biti itekako opasni te nerijetko prikupljaju čitav niz osjetljivih podataka s inficiranog računala koji mogu uključivati lokalno pohranjene podatke, korištena korisnička imena i zaporke (npr. Facebook, Twitter i slični servisi) te, u određenim slučajevima, mogu uključivati čak i instalaciju modula za nadzor pritisnutih tipki (engl. *keylogger*). Ovakvi moduli mogu prikupljati sve što korisnik upiše na računalu te to neprimjetno slati na neki središnji poslužitelj na internetu gdje napadač dalje može odlučiti što s ukradenim podacima.

Adware programi u načelu se upotrebljavaju za prikazivanje usmjerenih reklama. Ovi programi također prate aktivnosti korisnika, posebno u vidu web-stranica koje korisnik posjećuje, no ti se podaci prikupljaju isključivo u svrhu prikazivanja budućih reklama. Napadači koji stoje iza *adware* programa obično zarađuju na osnovi prikazivanja reklama te pomoću ove ilegalne metode zapravo nastoje povećati svoju zaradu.



Slika 4.15. Primjer prozora ubačenog od strane **adware** programa na inficiranom računalu

4.3.5. Rootkit programi

Rootkit programi pripadaju posebnoj kategoriji malicioznih programa s naprednim mogućnostima koji se zapravo vrlo često integriraju s drugim tipovima malicioznih programa. Tako danas npr. crvi ili trojanski konji koji imaju *rootkit* funkcionalnost nisu rijetkost već ih zapravo srećemo vrlo često.

Osnovni je zadatak *rootkit* funkcionalnosti onemogućavanje detekcije programa na inficiranom računalu putem normalnih metoda detekcije. Kada je primjerice neki crv pokrenut na inficiranom računalu te se pokušava širiti dalje, a ako nema *rootkit* funkcionalnost, vidljiv je samo kao još jedan proces te ga korisnik može vidjeti i zaustaviti putem tradicionalnih programa za upravljanje procesima na računalu poput Task Managera na Windows operacijskim sustavima ili ps naredbe na Linux operacijskim sustavima.

Rootkit programe (odnosno mogućnosti) dijelimo prema njihovom stupnju integracije s operacijskim sustavom na korisničke *rootkit* programe i kernel *rootkit* programe.

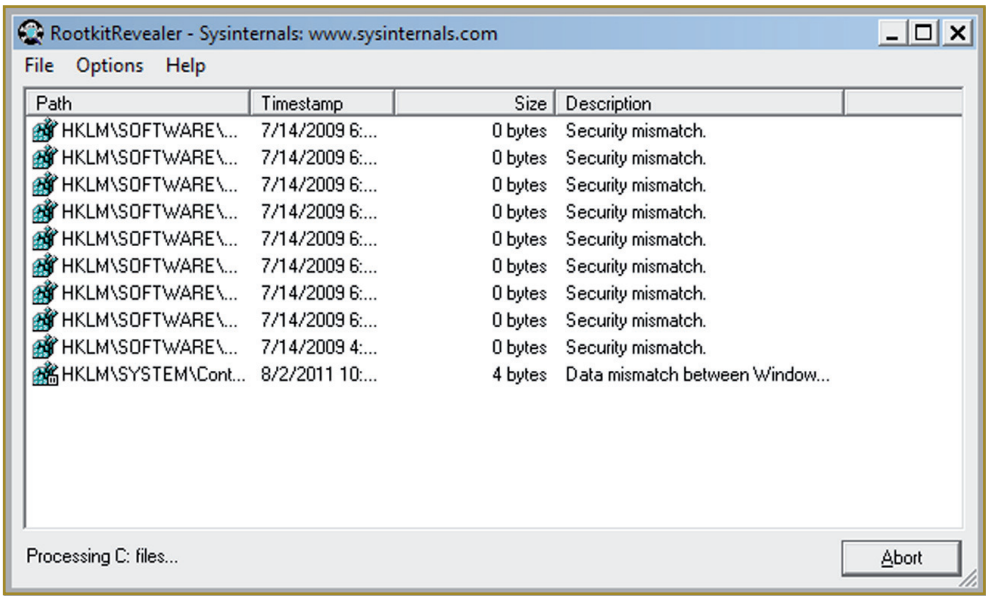
Korisnički *rootkit* programi zasnovani su na zamjeni legitimnih aplikacija koje bi korisnik mogao upotrebljavati u svrhu detekcije malicioznog programa. Na primjer, već spomenuti Task Manager na Windows operacijskim sustavima nakon inicijalne infekcije računala maliciozni program može zamijeniti s lažnim Task Manager programom koji će nastaviti prikazivati sve procese na računalu ispravno osim onih koji predstavljaju sam maliciozni program. Na taj način, ako korisnik pokrene Task Manager, program neće biti u stanju vidjeti maliciozni proces dok će sve ostalo funkcionirati kako treba, što napadaču omogućuje dulje razdoblje kontrole inficiranog računala budući da će se korisnik trebati koristiti nekim naprednijim metodama detekcije malicioznih programa.

Kernel *rootkit* programi pokušavaju postići jednak učinak kao i korisnički *rootkit* programi s razlikom da se kernel *rootkit* programi, kao što im i ime govori, integriraju usko s jezgrom operacijskog sustava, kernelom. Kernel *rootkit* programi ne modificiraju nikakve druge aplikacije na sustavu, već pokušavaju preciznim modificiranjem struktura i objekata u jezgri operacijskog sustava onemogućiti detekciju malicioznog programa. Na primjer, lista svih pokrenutih procesa na računalu u kernelu se drži kao lista povezanih pokazivača na objekte koji opisuju pojedine procese. Pažljivim modificiranjem ove liste kernel *rootkit* program može onemogućiti detekciju aplikacijama poput Task Managera koje samo prolaze kroz ovu listu i ispisuju rezultate. Na sličan način kernel *rootkit* programi mogu modificirati bilo koje objekte ili liste koje kernel drži te tako sakriti ne samo procese već i datoteke, mrežne komunikacijske veze i slično. Korisnički *rootkit* programi također mogu pokušati sakriti sve ove zna-

kove koji upućuju na postojanje malicioznog programa, ali ovakve aktivnosti od njih zahtijevaju znatne modifikacije aplikacija koje se upotrebljavaju (npr. Task Managera za pregledavanje procesa, Windows Explorera za pregledavanje datoteka i slično).

Kernel *rootkit* programi mogu predstavljati velike probleme pogotovo pri pokušaju čišćenja inficiranog računala (uklanjanja malicioznog programa), zbog činjenice da korisnik ne može biti siguran je li uistinu uklonio maliciozni program ili je na računalu još uvijek prisutan *rootkit* koji skriva neke aktivnosti malicioznog programa. Zbog ove se činjenice danas u slučaju infekcije računala gotovo uvijek preporučuje potpuna reinstalacija računala, a u slučajevima kada to nije moguće, čišćenje računala, i to tako da se isto podigne s nekog medija kojem se vjeruje, npr. CD ili DVD medija, što osigurava da *rootkit* nije pokrenut.

Iako je u svrhu detekcije *rootkita* moguće upotrebljavati klasične antivirusne programe, obično se u ovu svrhu rabe specijalizirani *rootkit* detektori. Na sljedećoj je slici prikazan Microsoftov Rootkit Revealer alat koji je uspješno detektirao skrivene datoteke koje pripadaju Hacker Defender alatu. *Rootkit* detektori inače funkcioniraju tako da pokušavaju na što je moguće više načina provesti istu akciju detekcije procesa te analiziraju rezultate. Kod traženja datoteka, *rootkit* detektori često otvaraju sektore na tvrdom disku izravno, analiziraju njihov sadržaj te uspoređuju s rezultatima korisničkih aplikacija poput Windows Explorera. Ako je rezultat različit, na računalu je detektiran *rootkit*.



Slika 4.16. Rootkit Revealer pri detekciji računala inficiranog **rootkitom**

Bootkit programi

Najnovija kategorija *rootkit* programa pod nazivom *bootkit* napravljena je prvenstveno kako bi se zaobišle sigurnosne kontrole pune enkripcije tvrdog diska (više o ovoj metodi zaštite podataka dostupno je u poglavlju 3.4.2). Za razliku od klasičnih *rootkita* koji se pokreću nakon što je pokrenut sam operacijski sustav, *bootkiti* se instaliraju na prvi sektor tvrdog diska te se pokreću prije samog operacijskog sustava što im omogućuje modificiranje kernela još pri pokretanju te provođenje napada čak i na aplikacije koje se rabe za enkripciju svih podataka na tvrdom disku.

Napredne mogućnosti malicioznih programa

Autori malicioznih programa služe se različitim naprednim mogućnostima ne bi li otežali detekciju antivirusnim programima, ali i običnim korisnicima. Jasno je da je maliciozni program koji se ne mijenja lakše detektirati od malicioznog programa koji se stalno mijenja. Ovo vrijedi i za ručnu detekciju, ali i za antivirusne programe, pogotovo kada se koriste metodom detekcije malicioznih programa temeljenom na potpisima, koja je detaljnije opisana u poglavlju 4.4.

U svrhu otežavanja detekcije, danas se najčešće rabe sljedeće napredne mogućnosti malicioznih programa:

- polimorfizam – mogućnost malicioznog programa da se neprestano mijenja. Ovdje se ne misli na promjenu funkcionalnosti malicioznog programa, koja će zapravo uvijek ostati ista (npr. ako je funkcija malicioznog programa krađa osjetljivih podataka, on će to uvijek činiti), već na sam programski kod koji čini maliciozni program.
  Naime, isti je programski kod moguće prikazati na puno (praktički beskonačno) načina tako da program može modificirati sam sebe upravo u svrhu izbjegavanja detekcije od strane potpisa antivirusnih programa. Jednostavan primjer istog programskog koda je uvećavanje varijable: ako želimo varijablu A uvećati za 5, to možemo napraviti na puno načina:
  - A = A + 5
  - A = A + 7 – 2
  - A = A + 2; A = A + 3 itd.

  Jasno je da mogućnosti ima beskonačno što uvelike otežava pisanje potpisa antivirusnim proizvođačima;
- umetanje beskorisnog koda – još jedna metoda čiji je cilj otežavanje pisanja potpisa u svrhu detekcije antivirusnim programima. Ovdje je riječ jednostavno o beskorisnom kodu koji ne radi ništa niti je bitan za funkcionalnost malicioznog programa, no ovaj je kod moguće konstantno mije-

njati te u slučaju ručne analize stručnjaka antivirusne kuće može itekako odužiti i otežati proces analize;

- upotreba virtualnih strojeva – jedna od najkompleksnijih metoda otežavanja detekcije. Ovdje nije riječ o punim virtualnim strojevima kao što je to riječ u poglavlju 5.4, već o simuliranim virtualnim strojevima koji se rabe za postizanje određenih funkcionalnosti. Navedeni virtualni strojevi mogu biti vrlo teški za detekciju antivirusnim programima, pogotovo ako se upotrebljavaju neke metode koje su prisutne i kod legitimnih programa.

4.4. Antivirusni programi

Antivirusni programi predstavljaju osnovnu liniju obrane od malicioznih programa. Iako je instalaciju malicioznih programa moguće spriječiti nekim drugim procesima poput onemogućavanja pokretanja nedopuštenih programa ili izbjegavanjem upotrebe administratorskih korisničkih prava pri radu na računalu, u konačnici antivirusni programi pružaju zadnju liniju obrane od malicioznih programa.

Višeslojni strateški model sigurnosti, ali i drugi prihvaćeni standardi poput već opisanog ISO 27001 standarda i PCI DSS-a istaknuli su nužnost upotrebe antivirusnih programa ne samo na klijentskim računalima već i na svim poslužiteljskim računalima koji su pod povećanim rizikom od malicioznih programa, što danas uglavnom uključuje Windows operacijske sustave, ali ne znači da se u budućnosti neće više viđati maliciozni programi za druge platforme poput Linuxa ili Mac OS X-a.

Bitno je napomenuti da antivirusni programi ne osiguravaju stopostotnu zaštitu od malicioznih programa. Niti jedan od danas na tržištu dostupnih dvadesetak antivirusnih programa nema savršenu detekciju – naprotiv, vrlo često se događa da pri pojavljivanju novih malicioznih programa tek vrlo mali broj antivirusnih programa iste uspješno detektira. Razlog ovome su brojne metode koje maliciozni korisnici koriste, kao i činjenica da autori malicioznih programa mogu modificirati i testirati svoje programe sve dok ne postignu oblik koji ima najslabiju (ili nikakvu) detekciju.

Današnji se moderni antivirusni programi koriste čitavim nizom metoda detekcije malicioznih programa, od kojih su one najčešće korištene objašnjene u nastavku.

- Detekcija potpisima (engl. *Signature detection*) najstariji je način detekcije malicioznih programa koji podržavaju praktički svi antivirusni proizvodi danas. Pri analizi malicioznog programa moguće je napisati jedinstveni

potpis analiziranog programa. Potpis u ovom smislu znači neku definiciju slijeda okteta koji se mogu naći u malicioznom programu, njihovu lokaciju i još neke parametre kojima se antivirusni programi služe.
Prednost detekcije potpisima je njihova brzina – pri pregledavanju stotina i tisuća programa, što je danas zadatak modernih antivirusnih programa, svako usporavanje ovog pregledavanja je bitno budući da antivirusni programi moraju zbog zahtjeva korisnika raditi što je brže te trošiti što je moguće manje računalnih resursa. Potpisi su pritom idealni jer omogućuju brzu detekciju.

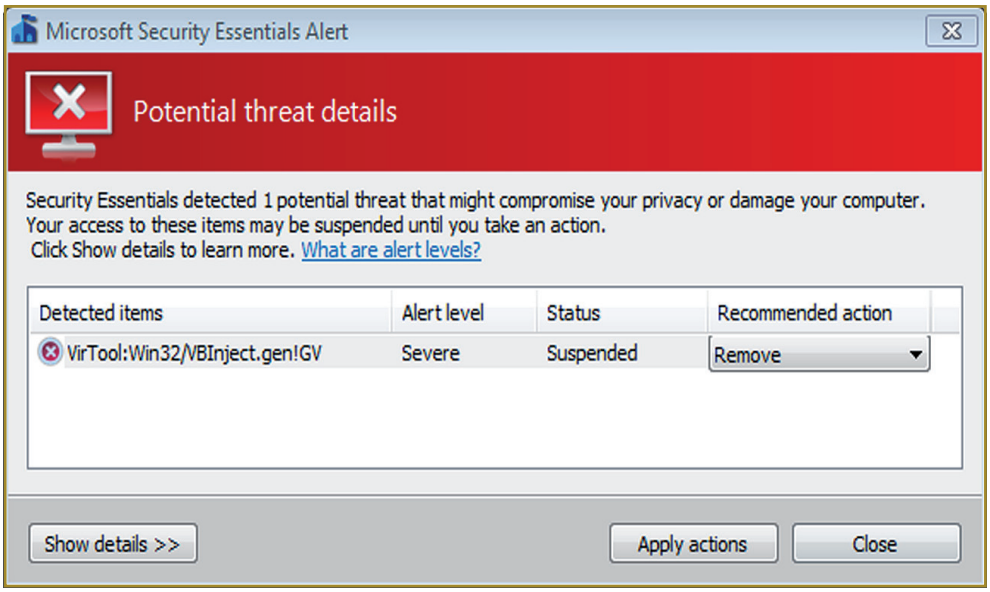
S druge strane, nedostatak potpisa je također očit – ako potpis ovisi o nekom mjestu programskog koda koje autor malicioznog programa može jednostavno promijeniti (ili se mijenja automatski, npr. korištenjem polimorfizma), potpis će biti u stanju detektirati samo jednu određenu vrstu malicioznog programa. U svrhu što bolje detekcije antivirusne tvrtke zato nastoje što je moguće češće napraviti tzv. generičke detekcije na dijelovima malicioznih programa koji ostaju statički.

- Detekcija aktivnosti (ponašanja) malicioznog programa (engl. *Behavioral, heuristic detection*) pokušava detektirati nove i nepoznate maliciozne programe ili one koji su modificirani toliko da ih nije moguće detektirati potpisima.
  Detekcija aktivnosti pri zahtjevu za pokretanjem nekog programa ne pokreće program izravno, već ga pokreće u posebnom, izoliranom okruženju koje stvara antivirusni program. Antivirusni program ujedno i analizira sve tražene aktivnosti programa koji je pokrenut u izoliranom okruženju te na osnovi ove analize zaključuje je li program benigan ili maliciozan. Na primjer, ako program pokušava izbrisati sve datoteke na računalu ili poslati osjetljive podatke na neko računalo na internetu, jasno je da je program maliciozan.

Navedene metode detekcije antivirusnih programa danas su općenito prihvaćene tako da većina antivirusnih programa na tržištu podržava obje metode detekcije, kao i čitav niz vlasničkih metoda pojedinih antivirusnih proizvođača koje se drže tajnima.

Većina današnjih antivirusnih programa integrira se s operacijskim sustavom kako bi omogućila detekciju na što je moguće nižoj razini, upravo zbog problema koje mogu izazvati *rootkit* programi, kao što je objašnjeno u poglavlju 4.3.5. Gledajući način rada antivirusnih programa pri detekciji, razlikujemo dva načina rada:

- detekciju malicioznih programa u stvarnom vremenu (engl. *Real time detection* – najčešći način rada antivirusnih programa i nužno zahtijeva usku integraciju s operacijskim sustavom. Antivirusni programi koji podržavaju detekciju malicioznih programa u stvarnom vremenu pri pokretanju bilo kojeg programa ili samo odabira programa u npr. Windows Exploreru (čak i bez pokretanja) automatski provode detekciju nad odabranim programom.
  Na ovaj je način moguće spriječiti infekciju računala čak i ako je korisnik pokušao pokrenuti maliciozni program. Osnovni problem detekcije u stvarnom vremenu je narušavanje performansi sustava budući da antivirusni program mora neprestano pregledavati sve datoteke koje korisnik ili aplikacije na sustavu pokušavaju otvoriti ili pokrenuti. Na sljedećoj je slici prikazan prozor upozorenja Microsoft Security Essentials antivirusnog programa pri detekciji malicioznog programa;



Slika 4.17. Microsoft Security Essentials detekcija malicioznog programa

- detekcija na zahtjev (engl. *On-Demand Detection*) – obično se upotrebljava za temeljitije pregledavanje svih datoteka na tvrdom disku. Tradicionalno, detekcija na zahtjev bila je detaljnija od detekcije u stvarnom vremenu, no povećavanjem brzine procesora i resursa koji su antivirusnim programima na raspolaganju, današnji antivirusni programi zapravo ne rade nikakvu razliku između detekcije u stvarnom vremenu i detekcije na zahtjev.

Detekcija na zahtjev se još uvijek, međutim, često upotrebljava na poslužiteljima, pogotovo kada je riječ o datotečnim poslužiteljima na koje korisnici pohranjuju svoje datoteke. Detekcija na zahtjev obično se pokreće u vrijeme kada je sustav neaktivan (npr. noću na poslužiteljima) i rabi se za detekciju možebitnih malicioznih programa koji nisu bili detektirani u stvarnom vremenu (npr. pohranjeni su na sustav kada još nisu postojali potpisi za njih).

5. Računalni sloj

Sigurnost samih poslužitelja ključna je i u održavanju sigurnosti podataka te izravno utječe na sve tri komponente CIA trokuta, budući da kompromitiranjem samog operacijskog sustava napadač dobiva potpunu kontrolu i samim time pristup osjetljivim podacima koji su njegov ultimativni cilj.

Povijesno gledajući, operacijski sustavi odnosno poslužitelji predstavljali su glavni cilj napadača, što je bilo i opravdano većim brojem sigurnosnih ranjivosti istih. No kroz vrijeme su proizvođači operacijskih sustava počeli uvoditi dobre prakse razvoja i pratiti sigurnosne trendove što je rezultiralo povećanjem razine sigurnosti operacijskih sustava odnosno računalnog sloja.

Greške pri implementaciji su, međutim, još uvijek dosta česte pa se nerijetko može vidjeti poslužitelje s operacijskim sustavom koji je sam po sebi siguran, ali koji su neispravno konfigurirani te dopuštaju neovlašteni pristup.

5.1. Autentikacija korisnika

Ključni zadatak koji se postavlja pred operacijske sustave je autentikacija korisnika i upravljanje njihovim pravima. Njihova su prava u konačnici definirana prema podacima kojima nastoje pristupiti, kao što je opisano u poglavljima 3.1 i 3.2, no da bi ispravno dodijelili prava bilo kojem korisniku, potrebno je provesti i predkorak autentikacije korisnika.

Cilj autentikacije je identifikacija korisnika koji se prijavljuje na sustav, da bi mu se na osnovi identiteta dodijelio ispravan pristup drugim objektima na sustavu poput npr. pristupa datotekama. Autentikacija je, dakle, provjera identiteta korisnika koji se prijavljuje, odnosno provjera je li korisnik zbilja onaj koji tvrdi da je.

Autentikacija se općenito može provesti na tri načina, odnosno prema tri tipa autentikacije.

- Autentikacija „Nešto što znaš“ (engl. *Something you know*) jest klasična autentikacija korisnika zaporkom, što u ovom slučaju predstavlja nešto što korisnik zna. Zaporka je općenito najprihvaćeniji način autentikacije korisnika zbog jednostavne upotrebe, prihvatljivosti od strane korisnika i niske cijene budući da svi operacijski sustavi podržavaju metodu autentikacije zaporkama.

  Da bi se sigurno pohranile na operacijske sustave, zaporke se obično pohranjuju u nekom načinu pohrane koji ne omogućuje njihovo jedno-

stavno čitanje niti administratorima. Naime, budući da administrator ima praktički potpuna prava na sustavu, kada bi zaporke bile pohranjene u čistom tekstualnom obliku, administrator bi jednostavno mogao pročitati zaporku nekog drugog korisnika, predstaviti se kao on i napraviti neke maliciozne aktivnosti. Ono što je još gore je da u tom slučaju nije moguće zadržati načelo neporecivosti, što znači da korisnik koji napravi malicioznu aktivnost uvijek može tvrditi da je to napravio administrator koji može pročitati njegovu zaporku.

Zbog toga se, kao što je već rečeno, zaporke najčešće pohranjuju kao sažetci, kao što je već opisano u poglavlju 3.3.3. U tom slučaju, nakon što je korisnik unio zaporku, operacijski sustav ponovno računa sažetak te ga uspoređuje s pohranjenim – ako je sažetak isti, korisnik je unio ispravnu zaporku, u protivnom je zaporka neispravna. Ovakav način pohranjivanja zaporki sprječava administratora u otkrivanju korisničkih zaporki, za što bi trebao provesti napad na zaporke (više detalja o napadima na zaporke dano je u poglavlju 5.1.1). Ovisno o metodi pohranjivanja zaporki (kriptografskom algoritmu sažimanja) te snazi zaporki ovakav napad može biti težak ili nemoguć za provođenje.

Snaga zaporke definirana je njezinom duljinom i različitim znakovima koji se upotrebljavaju. Jasno je da zaporka ne smije biti jednaka nečemu što napadač može upotrebljavati pa je za snažne zaporke potrebno rabiti slučajne nizove znakova. Preporuke za snažne zaporke dane su u nastavku.

- Zaporke bi trebale imati minimalno osam znakova, a uzevši u obzir snagu današnjih procesora, preporučuju se i dulje zaporke.
- Nikako se ne preporučuje upotreba bilo kakve riječi u zaporci, a pogotovo ne pojmova koji se mogu povezati s korisnikom poput imena članova obitelj i slično.
- Zaporka bi trebala sadržavati barem tri od četiri dostupna skupa znakova (mala i velika slova engleske abecede, brojke i posebni znakovi (!"#$%&/).
- Za snažne se zaporke danas preporučuje upotreba i cijelih fraza (engl. *passphrase*) koje predstavljaju čak i spojene rečenice, što itekako otežava provođenje napada na zaporke.

- Autentikacija „Nešto što imaš“ (engl. *Something you have*) – proširuje snagu procesa autentikacije sa zaporke na još jednu komponentu koju korisnik ima. Ovakav način autentikacije često se naziva i dvofaktorska autentikacija budući da korisnik treba znati zaporku (ili npr. PIN broj ure-

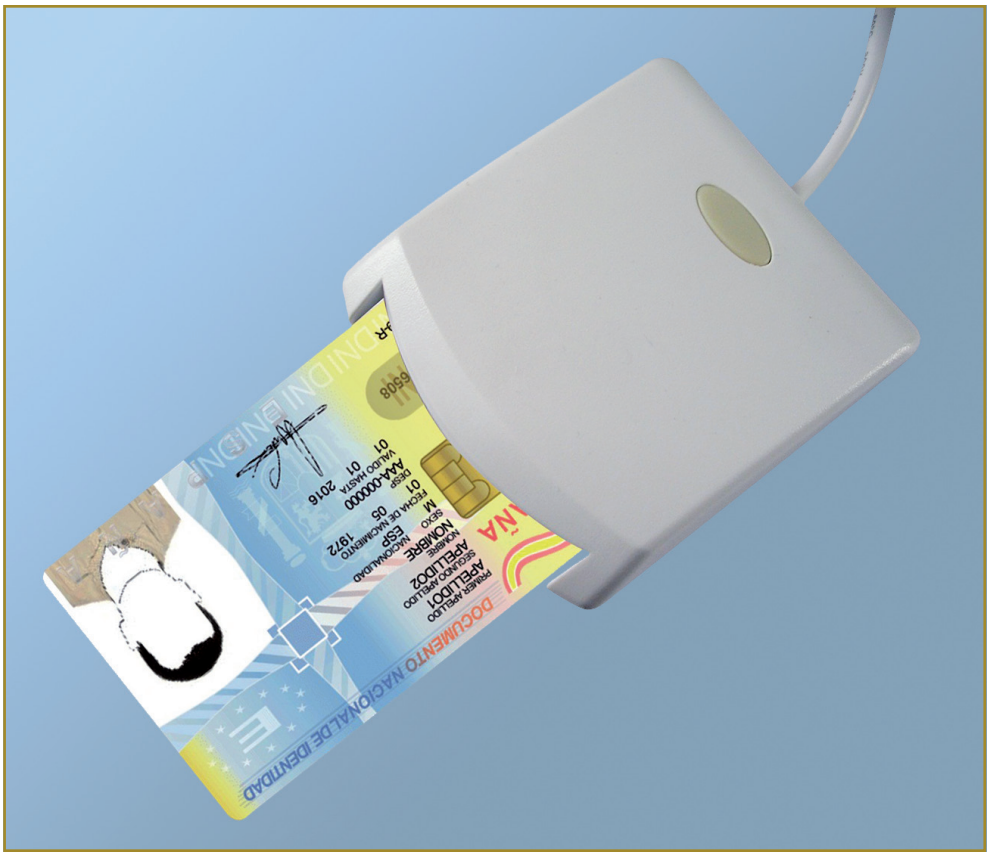
đaja), što je prvi faktor te imati sam token, odnosno uređaj koji mu omogućuje autentikaciju, što je drugi faktor.

Token je uređaj koji omogućuje generiranje jedinstvenih zaporki ili drugih mehanizama koji omogućuju autentikaciju. Klasično je token izveden u obliku malog uređaja kojeg korisnik može jednostavno nositi te koji generira dinamičke zaporke. Ove su zaporke obično ovisne o vremenu te se mijenjaju u regularnim intervalima što znači da sam uređaj mora imati vrijeme sinkronizirano s poslužiteljem. Ovdje je potrebno naglasiti da uređaj ne treba znati točno vrijeme, već treba samo ispravno brojiti vrijeme (sekunde). Primjer ovakvog uređaja koji je dosta popularan je token tvrtke RSA koji generira jedinstveni šesteroznamenkasti broj svakih 60 sekundi. Korisnik zatim upisuje svoju zaporku i generirani broj kako bi dokazao svoj identitet. Sigurnost ovakvog načina autentikacije može se odmah vidjeti budući da napadač više ne može samo ukrasti zaporku korisnika (što bi mogao npr. pomoću programa koji prati sve što korisnik upisuje na tipkovnicu, *keyloggera*) već mu je za ispravnu prijavu potreban i jedinstveni broj generiran od strane tokena, što znači da je potrebno nekako i ukrasti token. Jednako je tako potrebno napomenuti da kada se generirani broj s tokena jednom iskoristi, nije ga moguće opet iskoristiti.

Slika 5.1. RSA token za dvofaktorsku autentikaciju

Zbog svoje su jednostavnosti i relativno niske cijene dvofaktorska autentikacija, odnosno tokeni našli primjenu na cijelom nizu osjetljivih aktivnosti, pogotovo kada je riječ o servisima poput internetskog bankarstva gdje je upotreba dvofaktorske autentikacije nezaobilazna.

Osim već spomenutih tokena, dvofaktorska autentikacija može se upotrebljavati i putem pametnih kartice, koje se danas sve više rabe. Riječ je o karticama veličine kreditne kartice koje imaju poseban čip i procesor koji omogućuje pohranjivanje podataka i procesiranje. Problem s ovakvom upotrebom dvofaktorske autentikacije je da korisnik mora osim pametne kartice imati i poseban uređaj za čitanje (tzv. čitač za pametne kartice), što poskupljuje implementaciju ovakvog sustava i dodatno ju otežava po korisnika koji se bez čitača pametne kartice ne može prijaviti na sustav ili obaviti transakcije, ako je npr. riječ o internetskom bankarstvu, čak i ako ima pametnu karticu sa sobom.



Slika 5.2. Pametna kartica sa čitačem

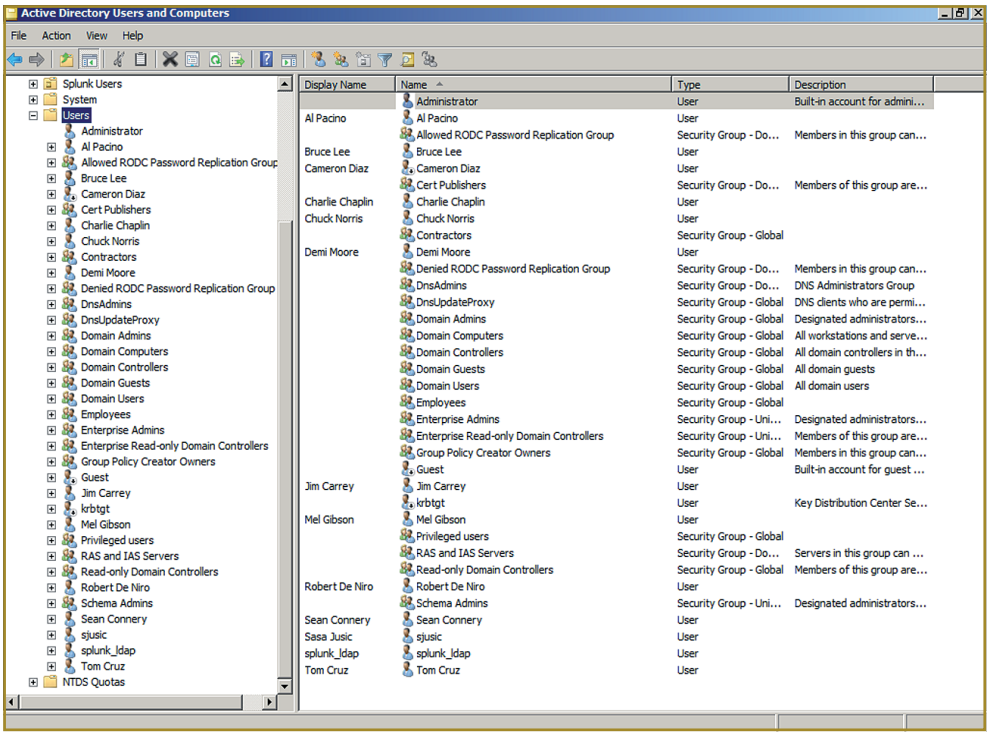
- Autentikacija „Nešto što jesi“ (engl. *Something you are*) – biometrijska autentikacija o kojoj je već bilo riječi u poglavlju 2.3. Zbog visoke cijene biometrijske autentikacije te većeg zadiranja u privatnost korisnika danas su biometrijski sustavi još uvijek relativno rijetki, ako se izuzmu jednostavni čitači otiska prsta koji polako postaju sve prisutniji na prijenosnim računalima.

Uz uspješnu metodu autentikacije na sustavima je potrebno ispravno implementirati prava pristupa, kao što je bilo objašnjeno u poglavlju 3, Podatkovni sloj.

Pri autentikaciji korisnika operacijski sustav klasično sadržava lokalno pohranjene podatke o korisničkim računima i sažetke zaporki. Jasno je da ovakav pristup nije skalabilan te da je za informacijske sustave s velikim brojem korisnika i poslužitelja upravljanje korisničkim računima potrebno centralizirati.

Centralizirani sustavi za upravljanje korisničkim računima u osnovi funkcioniraju isto kao i prethodno spomenuti, samostalni sustavi. Ključna je razlika da u centraliziranom sustavu pojedini poslužitelj pri autentikaciji korisnika konzultira središnji, glavni poslužitelj te lokalno ne sadržava nikakve podatke o korisničkom računu, osim možda prava pristupa te lokalnih grupa kojima taj korisnički račun pripada. Središnji poslužitelj (ili više njih, što se obično rabi u svrhu redundancije) je taj koji lokalno ima pohranjene podatke o korisničkim računima, uključujući i sažetke zaporki i koji ultimativno provjerava je li korisnik upisao dobru zaporku te ga autenticira. Danas se najčešće upotrebljavaju sljedeća dva centralizirana sustava za upravljanje korisničkim računima:

- *Active Directory* – Microsoftovo rješenje za upravljanje korisničkim računima pomoću kojeg je moguće integrirati i druge operacijske sustave poput npr. Linuxa. *Active Directory* omogućuje višestruke poslužitelje koji međusobno sinkroniziraju podatke o korisničkim računima.
- *Active Directory* u svrhu autentikacije upotrebljava protokol koji se zove Kerberos. Kerberos je autentikacijski protokol koji je moguće upotrebljavati i na drugim operacijskim sustavima i zasnovan je na kartama za pristup (engl. *Tickets*). Kod Kerberosa postoji hijerarhija poslužitelja koji omogućuju definiranje pristupa pojedinim servisima. Kada se korisnik prijavljuje, dobiva tzv. TGT (engl. *Ticket Granting Ticket*) kartu koja mu omogućuje traženje pristupa drugim resursima. Kada taj (autenticirani) korisnik želi pristupiti nekom servisu, njegovo računalo automatski zatraži kartu za pojedini servis koju dobiva na osnovi TGT-a. Riječ je o kompleksnom protokolu čija je glavna osobina da omogućuje sigurnu autentikaciju i razmjenu upita za servisima preko nesigurnih računalnih mreža;



Slika 5.3. Active Directory Users and Computers aplikacija koja omogućuje upravljanje korisničkim računima

- LDAP (engl. *Lightweight Directory Access Protocol*) – protokol koji omogućuje pristup imeničkim direktorijima preko računalnih mreža. Imenički direktorij u ovom smislu predstavljaju bilo koji organizirani podaci – LDAP je tako moguće upotrebljavati za autentikaciju korisnika, gdje organizirane podatke predstavljaju informacije o korisničkim računima uključujući i sažetke zaporki. Osim za autentikaciju, LDAP je moguće upotrebljavati i za čitav drugi niz primjena, kao npr. za telefonske imenike i slično.
  LDAP omogućuje slično upravljanje korisnicima kao što je to bio slučaj kod *Active Directorya* gdje se na jednom, središnjem mjestu drže podaci o svim korisnicima.

5.1.1. Napadi na zaporke

Budući da korisničke zaporke često predstavljaju prvu crtu obrane prema sustavima, postoji čitav niz napada koje je moguće provesti. Pojedini napadi na zaporke ovise o određenoj implementaciji, no napadi opisani u nastavku ovog poglavlja generički su te se mogu primijeniti na sve zaporke korisnika.

- Svi napadi na zaporke korisnika zasnovani su na pogađanju zaporke korisnika. Kao što je rečeno u poglavlju 5.1, zaporke bi trebale imati postavljena određena pravila kako bi se osiguralo da korisnici upotrebljavaju snažne zaporke. Mnogi sustavi, uključujući i Windows i Linux operacijske sustave, danas omogućuju centralno postavljanje zahtjeva na snagu zaporki, gdje se korisnicima onemogućuje postavljanje jednostavnih zaporki (npr. zaporke koje su jednake njihovom korisničkom imenu i slično). Jednako tako, budući da se napadi na zaporke (u slučaju da napadač nije uspostavio kontrolu nad operacijskim sustavom te došao do sažetaka zaporki) temelje na pogađanju zaporki, na središnjem je poslužitelju moguće definirati broj puta koliko se dopušta unošenje neispravne zaporke za pojedini korisnički račun, nakon čega se korisnički račun automatski zaključava. Potrebno je napomenuti da ovakav pristup treba pažljivo implementirati budući da napadač s ovakvim sustavom može namjerno zaključati sve korisničke račune na sustavu (ako samo zna njihova korisnička imena do kojih je obično lakše doći), što će u konačnici dovesti do uskraćivanja pristupa sustavu.

Bez obzira na sigurnosne kontrole koje su implementirane, napadači danas primjenjuju sljedeće tri metode pri pokušaju razbijanja korisničkih zaporki, odnosno njihovog pogađanja.

- Napad sirovom snagom (engl. *Brute force attack*) je najučinkovitiji, ali ujedno i najsporiji napad na zaporke. Riječ je o vrlo jednostavnom napadu koji uzima sve moguće vrijednosti za zaporke i isprobava ih na sustavu, jednu po jednu – npr. prvo uzima slovo „a“ za zaporku, pa slovo „b“ itd. Ovakav pristup može biti vrlo uspješan jer će proći kroz sve moguće zaporke, no jednako tako može trajati danima, mjesecima, godinama i tisućama godina.

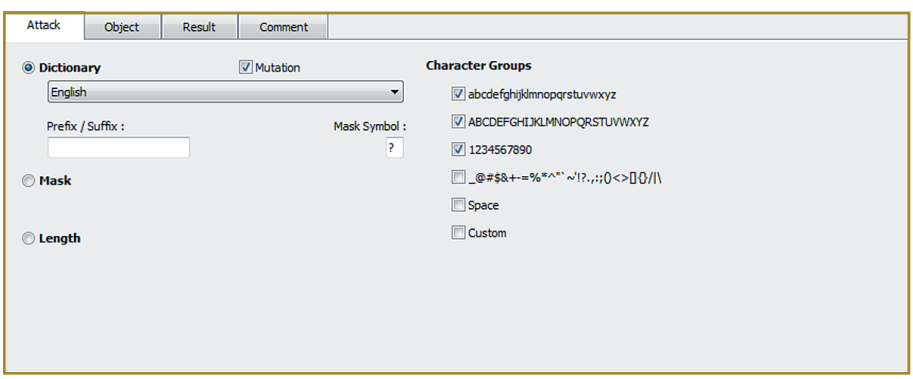
  Napad sirovom snagom ovisi o duljini zaporke – što je zaporka dulja, to će i napad trajati dulje. Na primjer, ako napadač može isprobati 1000 zaporki u sekundi, što je relativno velik broj ako je riječ o isprobavanju preko računalne mreže (dakle, kada nije riječ o lokalnom napadu na zaporke), u slučaju da zaporka može imati četiri znaka koji mogu uključivati mala i velika slova engleske abecede, brojeve i posebne znakove (! @ # $ % ^ & * ( ) - _ + = ~ ` [ ] { } | \ : ; ” ’ < > , . ? /), broj mogućih kombinacija je (26+26+10+33) ^ 4 = 81450625, što uz isprobavanje od 1000 zaporki u sekundi maksimalnu duljinu napada stavlja na puna 22 sata.

Ako se sada za primjer uzme maksimalna duljina zaporke od osam znakova, broj kombinacija raste na 95 ^ 8 što je približno 6,5 * 10 ^ 14, a što duljinu napada stavlja na 210 tisuća godina!

Čak i ako se brzina pogađanja znatno poveća, vrijeme potrebno za isprobavanje svih kombinacija za zaporku od osam znakova preveliko je za napadača, a svaki dodatni znak potrebno vrijeme još dodatno eksponencijalno povećava.

Zbog svega navedenog napad sirovom snagom obično je zadnji izbor napadača, koji često prvo isprobavaju kombinaciju narednih metoda.

- Napadi rječnikom (engl. *Dictionary attacks*) u svrhu pogađanja zaporki upotrebljavaju rječnike iz kojih grade zaporke koje rabe pri pogađanju. Pretpostavka ovih napada je da većina korisnika upotrebljava neku riječ kako bi lakše zapamtili zaporku.
  Umjesto isprobavanja svih znakova, napadač sada može uzeti unaprijed definiran rječnik s npr. 10,000 najčešće korištenih riječi te na taj način prvo isprobati ovakve slučajeve ne bi li povećao mogućnost pogađanja zaporki. Na internetu je moguće naći rječnike za sve jezike, uključujući i hrvatski, a napadač može i napraviti vlastiti rječnik.
- Kombinirani napadi (engl. *Hybrid attacks*) služe se rječnicima koje zatim proširuju određenim znakovima. Ovakve su metode danas najuspješnije budući da veliki broj korisnika upotrebljava zaporke koje su poznate riječi, na koje dodaju neke znakove da bi zadovoljili zahtjeve za snagom zaporke. Jedan primjer takve zaporke je npr. „skola123“.
  Kombinirani napadi u ovom slučaju uzimaju riječi iz rječnika te na njih dodaju znakove (u ovom primjeru niz „123“) i pokušavaju tako pogoditi zaporku. Današnji programi za napade na zaporke omogućuju čitav niz modifikacija riječi iz rječnika, kao što je prikazano na sljedećoj slici.



Slika 5.4. Modificiranje zaporke pri kombiniranom napadu

5.2. Ojačanje sigurnosti operacijskih sustava

Osim već spomenutih zaporki korisnika, u svrhu zaštite na računalnom sloju potrebno je ispravno konfigurirati i implementirati same poslužitelje. Nakon početne instalacije poslužitelja, s operacijskim sustavima vrlo često dolazi preinstaliran čitav niz različitih aplikacija i servisa koji su u stvarnom radu za taj specifičan poslužitelj nepotrebni. Na primjer, ako se instalira DNS poslužitelj, onda nema nikakve svrhe da su na njemu instalirani i web i FTP servisi. Ovakvo uklanjanje suvišnih servisa još više dolazi do značaja kad se zna da današnji poslužitelji uglavnom imaju jednu ili dvije uloge.

Proces ojačanja sigurnosti operacijskih sustava (engl. *System hardening*) predstavlja proces koji bi trebalo uključiti za svaki poslužitelj pri njegovoj inicijalnoj instalaciji. Ovaj proces uključuje sve korake koje je potrebno provesti kako bi se sigurnost poslužitelja dovela na zadovoljavajuću, inicijalnu razinu te bi kao takva trebala biti dijelom cjelokupnog upravljanja informacijskom sigurnošću sustava, kao što je bilo spomenuto u poglavlju 1.4.1.

Budući da proces ojačanja sigurnosti operacijskih sustava ovisi o pojedinom operacijskom sustavu, obično je preporučljivo napraviti dokument koji će opisivati korake koji se moraju provesti za svaki pojedini operacijski sustav. Ovaj je dokument potrebno redovito osvježavati, budući da se i zahtjevi na sigurnost i sigurnosni rizici svakodnevno mijenjaju.

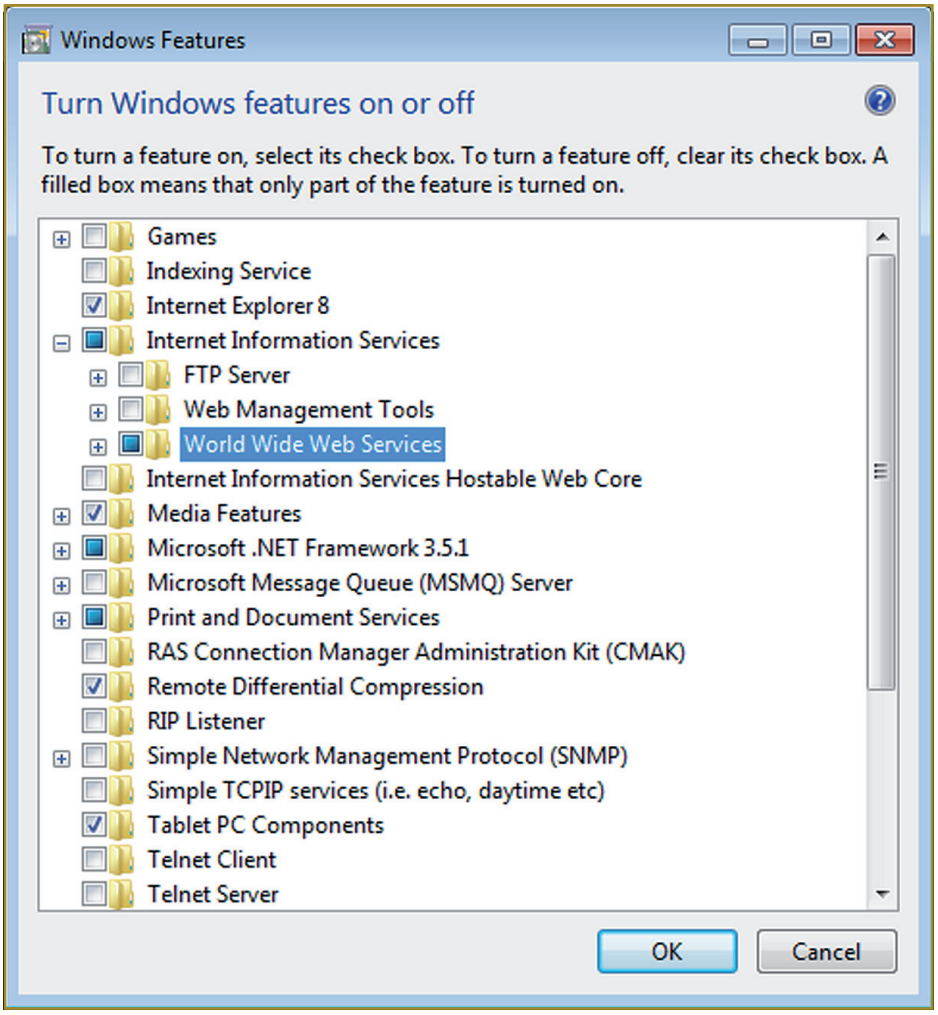
Organizacija Center for Internet Security (CIS) u svrhu povećanja sigurnosti računala i poslužitelja na internetu nudi besplatne dokumente za ojačavanje sigurnosti operacijskih sustava koji se mogu iskoristiti kao detaljni predlošci pri izradi ovakvih dokumenata. CIS-ovi predlošci dostupni su na njihovim web-stranicama http://cisecurity.org/.

Potrebno je napomenuti da prijedlozi za ojačanje sigurnosti operacijskih sustava dani u nastavku ovog poglavlja nikako ne jamče sigurnost sustava već samo predstavljaju osnovne korake koje je potrebno poduzeti u svrhu ojačanja sigurnosti.

Iako koraci procesa ojačanja sigurnosti operacijskih sustava ovise o pojedinom operacijskom sustavu, sam je proces u načelu generički pa se može opisati sljedećim koracima.

- Inicijalnu instalaciju operacijskog sustava potrebno je provesti tako da se omoguće osnove za postavljanje sigurnosti operacijskog sustava. Na Windows operacijskim sustavima ovo uključuje obavezno uključivanje NTFS datotečnog sustava koji omogućuje postavljanje prava pristupa korisnicima.

- Uklanjanje nepotrebnih servisa i aplikacija jedan je od najbitnijih koraka ojačavanja sigurnosti operacijskog sustava. Ovisno o pojedinom operacijskom sustavu, nepotrebne aplikacije i servise moguće je isključiti tijekom prvog procesa, inicijalne instalacije, što je optimalan postupak budući da se na računalo nikada niti ne instaliraju suvišni programski paketi.
  U slučaju da tijekom instalacije nije bilo moguće ukloniti nepotrebne programske pakete, to je potrebno napraviti nakon inicijalne instalacije. Sljedeća slika prikazuje Windows Components Wizard koji omogućuje uklanjanje i dodavanje pojedinih komponenti Windows operacijskog sustava.



Slika 5.5. Windows Components Wizard

Na Linux operacijskim sustavima programske pakete koji će biti instalirani moguće je detaljno odabrati tijekom instalacijske procedure. U slučaju da je programske pakete potrebno dodati ili ukloniti nakon instalacije, procedura ovisi o distribuciji koja se upotrebljava (npr. na RedHat/Fedora distribucijama može se rabiti paket yum, dok se na Debian/Ubuntu distribucijama može upotrebljavati paket apt-get ili aptitude).

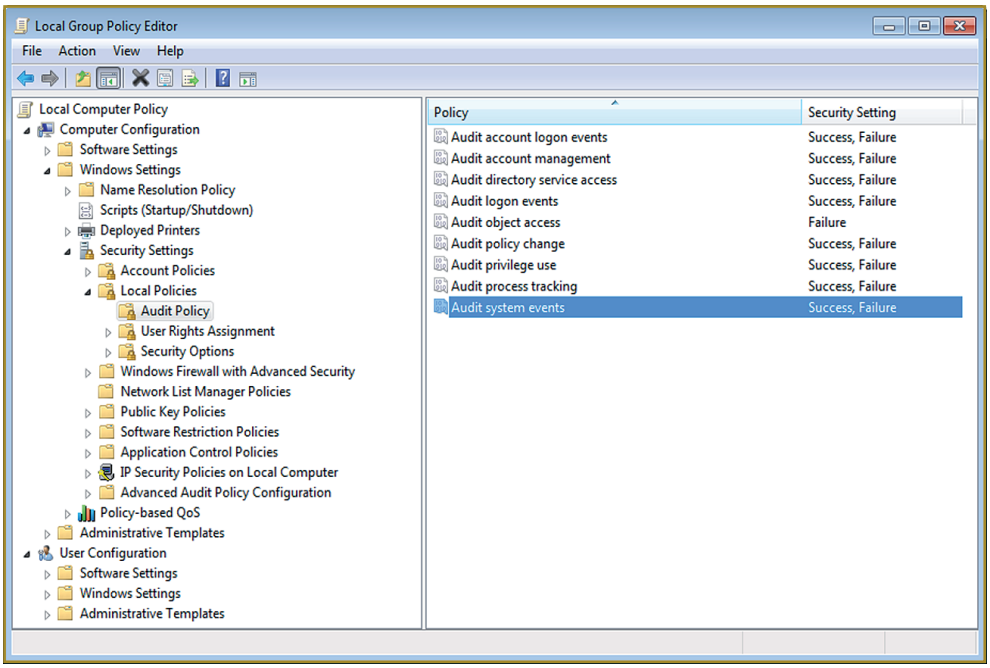
- Uklanjanje suvišnih ili nepotrebnih korisničkih računa. Operacijski sustavi obično dolaze s određenim unaprijed napravljenim korisničkim računima. Ako navedeni korisnički računi nisu potrebni, preporučuje ih se u potpunosti ukloniti ili, ako to nije moguće, potvrditi da su navedeni korisnički računi zaključani.

  Na Windows operacijskim sustavima ovo uključuje onemogućavanje gost korisničkog računa (engl. *guest*), dok je na Linux operacijskim sustavima preporučljivo provjeriti servisne korisničke račune te potvrditi da se oni ne mogu koristiti ljuskom sustava.

  Budući da je u ovoj fazi instalacije poslužitelj još praktički ranjiv, pogotovo nakon inicijalne instalacije kada na poslužitelj nisu instalirane nikakve sigurnosne zakrpe, provođenje koraka procesa navedenih u nastavku preporučuje se na izoliranom segmentu računalne mreže, gdje ostala računala i poslužitelji ne mogu pristupiti poslužitelju koji se instalira. Za implementiranje izoliranog segmenta računalne mreže moguće je koristiti se različitim metodama filtriranja mrežnog prometa, kao što je detaljnije objašnjeno u poglavlju 6.2.

  Konačno, na sustavu je potrebno postaviti politike sigurnosti vezane za korisničke račune, zaporke i bilježenje sigurnosnih događaja, o čemu je više riječi dano u poglavlju 5.3.

  - Sigurnosnu politiku snage zaporki potrebno je postaviti tako da odgovara zahtjevima kompletnog informacijskog sustava. Ova sigurnosna politika uključuje minimalno duljinu zaporki te zahtjeve za njihovu kompleksnost.
  - Bilježenje sigurnosnih događaja potrebno je konfigurirati tako da se bilježe svi uspješni i neuspješni pokušaji prijave na sustav. Na Linux operacijskim sustavima ovo je automatski uključeno dok je na Windows operacijskim sustavima starijim od Windows 7 ili Windows 2008 navedeno bilježenje potrebno ručno uključiti u sigurnosnoj politici sustava, kao što je prikazano na sljedećoj slici.



Slika 5.6. Postavljanje ispravnog bilježenja sigurnosnih događaja na Windows poslužitelju

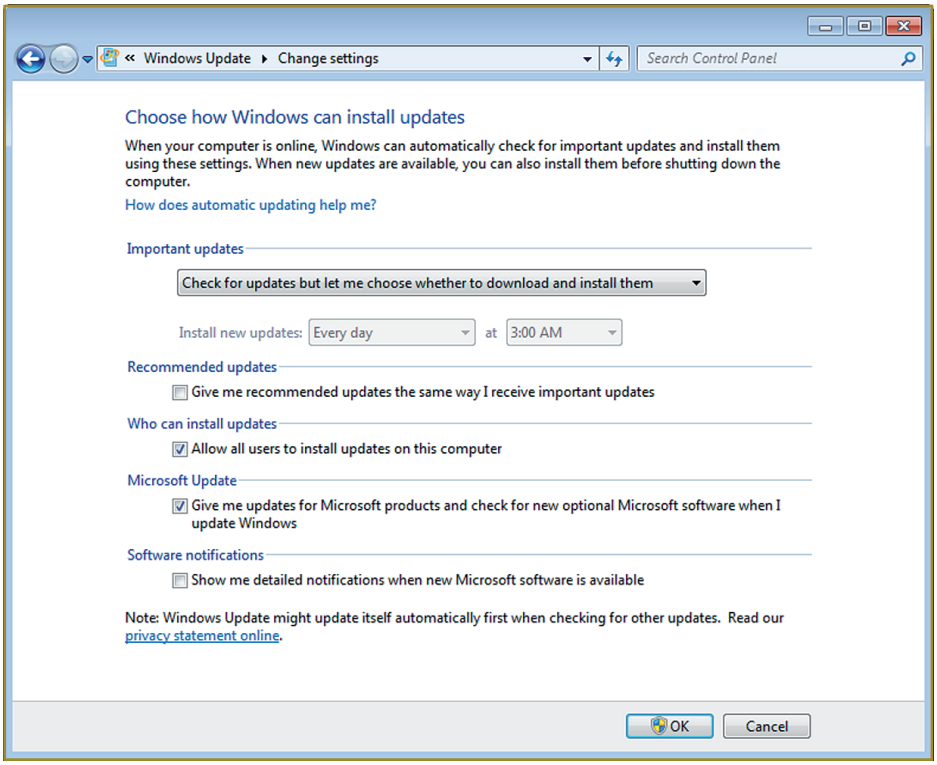
Ovisno o operacijskom sustavu koji se sigurnosno ojačava, u ovom je koraku moguće podesiti još i neke dodatne postavke poput, u slučaju Windows operacijskog sustava, onemogućavanja anonimnog enumeriranja SAM korisničkih računa, što napadaču omogućuje dolazak do liste korisničkih računa na sustavu.

- Nakon što su s poslužitelja uklonjeni svi suvišni programski paketi i servisi u prvom koraku, na poslužitelj je potrebno instalirati sve dostupne sigurnosne zakrpe. Ako su dostupni, tijekom ovog bi postupka trebalo prvo instalirati servisne pakete (npr. Service Pack 2 za Windows 2003 poslužitelj), nakon čega je kroz cijeli postupak potrebno proći još jednom kako bi se potvrdilo da su na poslužitelj instalirane sve dostupne sigurnosne zakrpe. Navedeni je postupak jednak za sve operacijske sustave.

  Na svim je poslužiteljima potrebno konfigurirati proces koji će redovito (svakodnevno ili više puta dnevno) provjeravati jesu li dostupne nove sigurnosne zakrpe te ih automatski skinuti na poslužitelj. Ovisno o kritičnosti poslužitelja, automatska instalacija sigurnosnih zakrpi se ne preporučuje (već samo skidanje, tako da su zakrpe spremne za instalaciju). Samu instalaciju treba provoditi administrator sustava ručno,

prema unaprijed definiranom ciklusu instalacije sigurnosnih zakrpa, što treba biti navedeno i u dokumentaciji sustava.

Na sljedećoj je slici prikazana politika provjere i skidanja sigurnosnih zakrpi na Windows 2003 poslužitelju, gdje se zakrpe samo skidaju, a administrator ih treba ručno instalirati.



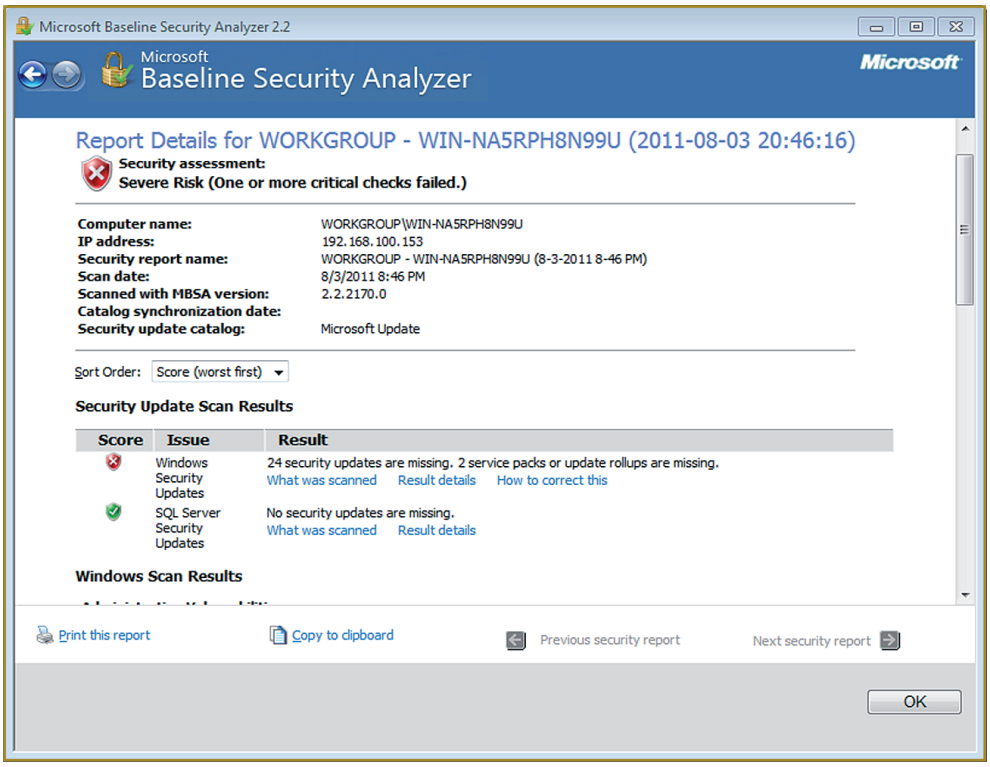
Slika 5.7. Postavke za automatsko skidanje sigurnosnih zakrpi na Windows poslužiteljima

Na Linux operacijskim sustavima isti je postupak moguće konfigurirati pomoću yum ili apt-get aplikacija.

- Provjera i podešavanje sustava za sinkronizaciju točnog vremena. Točno vrijeme ključno je ne samo za ispravan rad sustava već i za razne aktivnosti poput bilježenja dnevničkih zapisa (više o dnevničkim zapisima u poglavlju 5.3). U svrhu podešavanja sinkronizacije vremena rabe se različiti programski paketi, ovisno o pojedinom operacijskom sustavu. Windows operacijski sustavi u domeni automatski će imati vrijeme sinkronizirano s domenskim poslužiteljima zbog zahtjeva Kerberos protokola, koji se upotrebljava za autentikaciju (kao što je već bilo objašnjeno u poglavlju 5.1), za sinkroniziranim vremenom.

Na Linux distribucijama u svrhu sinkronizacije vremena upotrebljava se NTP (engl. *Network Time Protocol*) programski paket koji je potrebno zasebno instalirati i konfigurirati. Ovaj protokol omogućuje točnu sinkronizaciju vremena između NTP poslužitelja koji se može nalaziti bilo gdje na internetu i lokalnog poslužitelja.

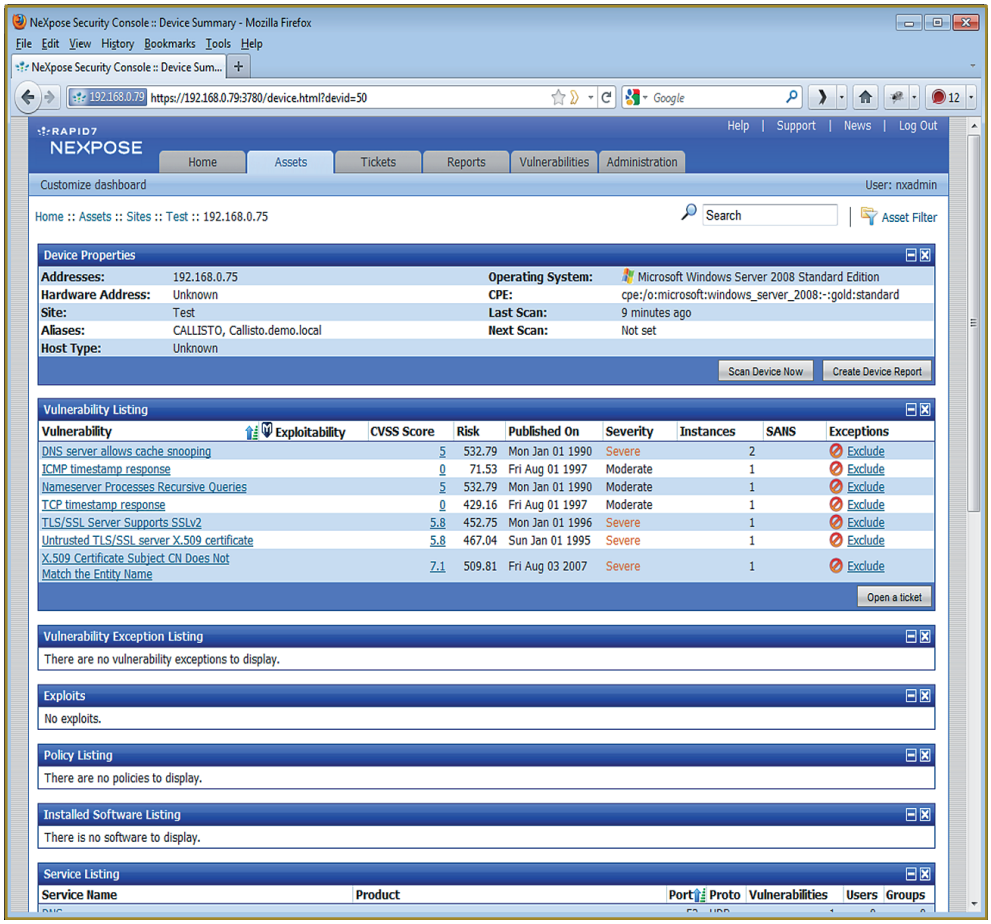
- Zadnji korak u procesu ojačanja sigurnosti operacijskih sustava je provjera konfiguracije odnosno ranjivosti sustava. U svrhu provjere ranjivosti sustava moguće je služiti se nekim od dostupnih besplatnih programskih paketa, poput Microsoftovog Baseline Security Analyzera (MBSA) za Microsoftove operacijske sustava.
  MBSA je alat koji administratoru dopušta provjeru sigurnosnih postavki, ali i potvrdu da su sve dostupne sigurnosne zakrpe instalirane na poslužitelj. Vrlo korisna mogućnost ovog alata je i udaljena provjera poslužitelja, gdje administrator treba samo unijeti podatke o administratorskom korisničkom računu poslužitelja, bez potrebe za fizičkom prijavom na poslužitelj koji se provjerava. MBSA alat prikazan je na sljedećoj slici.



Slika 5.8. MBSA alat prijavljuje sigurnosne zakrpe koje nedostaju

Osim MBSA alata postoji još čitav niz specijaliziranih alata koji se mogu upotrebljavati u svrhu provjere sigurnosti udaljenih računala. Neki od poznatijih alata koji su dostupni besplatno uključuju NeXpose, Nessus i OpenVAS.

Navedeni alati omogućuju provjeru sigurnosti praktički bilo kojeg poslužitelja, bez obzira na instalirani operacijski sustav. Kao i na primjeru MBSA alata, administrator treba samo upisati IP adresu poslužitelja čija se sigurnost provjerava te pokrenuti automatsku provjeru sigurnosti, nakon čega će alat generirati izvještaj koji će sadržavati sve identificirane sigurnosne ranjivosti.



Slika 5.9. Izvještaj NeXpose alata za provjeru ranjivosti

Proces ojačavanja sigurnosti operacijskih sustava predstavlja skup koraka koje je preporučljivo provesti prije stavljanja poslužitelja u produkcijsku ulogu i na-

stavka rada na njemu. Naravno, kao što je već rečeno, ovaj proces ne jamči sigurnost poslužitelja već samo uspostavlja određenu minimalnu razinu sigurnosti poslužitelja.

Kod većih je implementacija moguće razmotriti i specijalizirane mogućnosti centralizacije upravljanja sigurnosnim postavkama poslužitelja. Primjerice, na Windows operacijskim sustavima većinu postavki navedenih u koracima procesa ojačavanja sigurnosti operacijskih sustava moguće je centralizirati putem domenskih politika, nakon čega je poslužitelj potrebno samo staviti u odgovarajuću grupu poslužitelja u Windows domeni, što će automatski uzrokovati postavljanje ispravnih sigurnosnih postavki na navedeni poslužitelj.

5.3. Sistemski i operativni dnevnički zapisi

U svrhu praćenja aktivnosti korisnika, ali i bilo kojih drugih aktivnosti na poslužiteljima, upotrebljavaju se dnevnički zapisi. Dnevnički zapisi su datoteke koje sadržavaju zapise događaja koji su se dogodili na bilo kojoj komponenti informacijskog sustava – poslužiteljima, aplikacijama, mrežnim uređajima poput preklopnika i usmjerivača ili nečem četvrtom.

Dnevnički su zapisi iznimno bitni pri pokušaju rekonstrukcije događaja. Na primjer, ako se dogodio sigurnosni incident, vrlo su često dnevnički zapisi jedini način na koji je moguće ustanoviti što se dogodilo, tko je odgovoran te koliki je utjecaj sigurnosnog incidenta. Kao što se može vidjeti, kvalitetni dnevnički zapisi koji su pritom pohranjeni na siguran način ključan su dio sigurnosti informacijskog sustava. Značaj dnevničkih zapisa prepoznale su i ključne organizacije, poput već spomenutih ISO 27001 standarda i PCI DSS-a, koje zahtijevaju uspostavu sustava za upravljanje dnevničkim zapisima, kao i njihovo redovito sigurnosno arhiviranje.

Standardni poput PCI DSS-a još i dodatno specificiraju da se dnevnički zapisi moraju čuvati na tvrdim diskovima minimalno za zadnja tri mjeseca, a arhive moraju biti dostupne za zadnjih godinu dana.

Sustav za upravljanje dnevničkim zapisima zapravo definira četiri koraka u životnom ciklusu dnevničkih zapisa koji su navedeni u nastavku.

5.3.1. Generiranje i slanje dnevničkih zapisa

Generiranje i slanje dnevničkih zapisa je osnovni zahtjev koji se postavlja pred operacijske sustave, aplikacije, mrežne uređaje i bilo koje druge komponente informacijskog sustava.

Tip, količina i izgled dnevničkih zapisa ovise o pojedinom operacijskom sustavu odnosno uređaju. Nažalost, brojni pokušaji standardizacije izgleda dnevničkih zapisa nisu urodili plodom pa danas praktički svaki proizvođač generira drukčije dnevničke zapise.

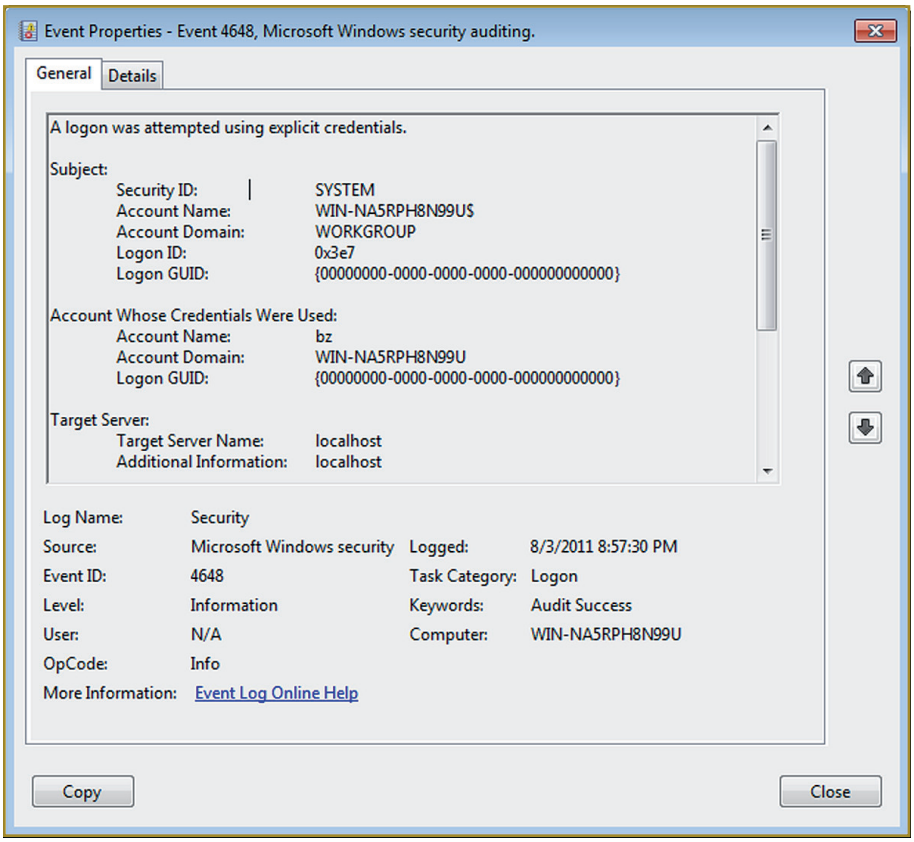
Bez obzira na tip dnevničkih zapisa, svi zapisi trebaju zadovoljiti određene zahtjeve:

- točan datum i vrijeme;
- vrstu događaja. Ovisno o pojedinom događaju, administrator mora biti u stanju interpretirati događaj prema vrsti (npr. neuspjeli pokušaj prijave na sustav ili prestanak rada servisa);
- stanje događaja odnosno predstavlja li događaj uspješno ili neuspješno stanje;
- izvor događaja gdje je potrebno zabilježiti servis ili aplikaciju koja je generirala događaj kao i poslužitelj ili uređaj na kojem se događaj odvio, pogotovo u slučaju kada se dnevnički zapisi skupljaju na središnji poslužitelj, kao što je opisano u poglavlju 5.3.2;
- sam dnevnički zapis koji opisuje određeni događaj. U svrhu lakšeg automatskog obrađivanja generiranih dnevničkih zapisa obično se preporučuje strukturiranje događaja, no u prsaksi je vrlo često rukovanje nestrukturiranim dnevničkim zapisima.

Iako ne postoji standard generiranja dnevničkih zapisa, oni se ipak mogu svrstati u određene zajedničke grupe.

- Sigurnosni dnevnički zapisi (engl. *Security event logs*) sadržavaju zapise o događajima poput ispravnih prijava na sustav te neispravnih pokušaja prijava na sustav, kao i druge sigurnosne događaje poput uspješnih i neuspješnih pokušaja pristupa datotekama.

  Sigurnosni dnevnički zapisi obično se zapisuju na određene lokacije na sustavu. Na Windows operacijskim sustavima to je Security komponenta Windows Event Loga, servisa koji je zadužen za rukovanje dnevničkim zapisima. Na Linux operacijskim sustavima lokacija sigurnosnih dnevničkih zapisa ovisi o servisu koji se upotrebljava, no u najvećem broju slučajeva riječ je o /var/log/secure datoteci u koju se bilježe minimalno uspješni i neuspješni pokušaji prijave na sustav. Na sljedećoj je slici prikazan izgled jednog sigurnosnog dnevničkog zapisa na Windows operacijskom sustavu.



Slika 5.10. Sigurnosni dnevnički zapis generiran na Windows poslužitelju

Zbog svojeg značaja, pogotovo u slučaju sigurnosnog incidenta, pri implementaciji sustava potrebno je ispravno konfigurirati zapisivanje svih sigurnosnih dnevničkih zapisa, a iste je potrebno i regularno arhivirati, prema unaprijed propisanim procedurama prihvaćenim za cijeli informacijski sustav.

- Dnevnički zapisi događaja sustava (engl. *System event logs*). U ovu grupu dnevničkih zapisa spadaju svi događaji generirani od samog operacijskog sustava i njegovih komponenti, a koji nisu sigurnosni događaji.
  Pri podizanju poslužitelja dnevnički se zapisi prvo zapisuju u ovu grupu događaja te uključuju općenite podatke o okruženju, kao i zapise pojedinih servisa o njihovog stanju.

  Bilo koja komponenta operacijskog sustava također zapisuje dnevničke zapise u ovu grupu, poput npr. problema u radu nekog upravljačkog programa (engl. *Driver*).

Na Windows operacijskim sustavima dnevnički zapisi događaja sustava zapisuju se u System komponentu Windows Event Loga, dok se na Linux operacijskim sustavima ovi zapisi obično pohranjuju u /var/log/messages datoteku.

- Aplikacijski dnevnički zapisi (engl. *Application event logs*) sadržavaju zapise o događajima koje su generirale različite aplikacije instalirane na poslužitelj.

Zbog velikog je broja različitih aplikacija ovaj tip dnevničkih zapisa nažalost najmanje standardiziran. Iako neki operacijski sustavi poput npr. Windowsa imaju standardno mjesto namijenjeno za pohranjivanje aplikacijskih dnevničkih zapisa (Application komponenta Windows Event loga), vrlo često aplikacije svoje dnevničke zapise pohranjuju u lokalne direktorije na tvrdom disku poslužitelja. Čak i aplikacije i servisi koje je razvio Microsoft pohranjuju dnevničke zapise u lokalne direktorije, tako npr. Microsoftov IIS web-poslužitelj pohranjuje zapise u C:\Windows\System32\LogFiles direktorij.

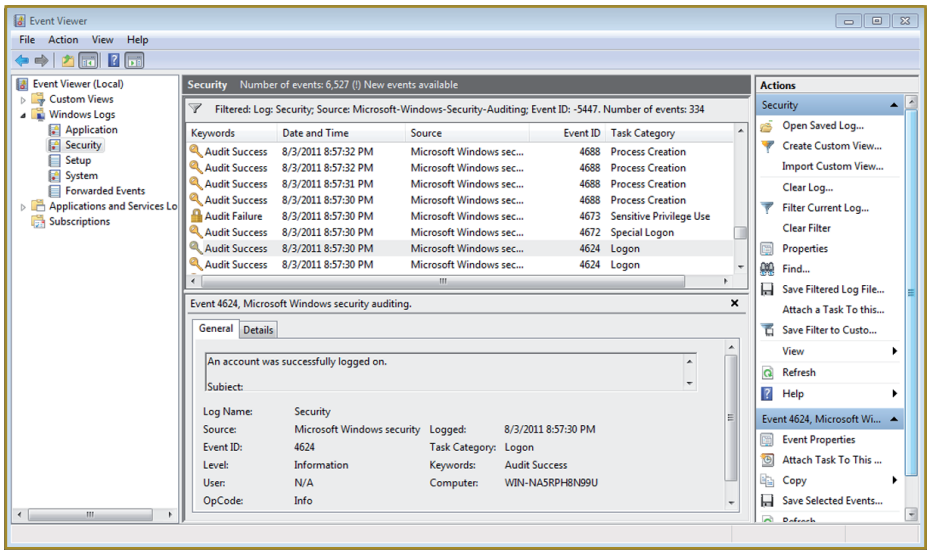
Za većinu aplikacija lokacija dnevničkih zapisa kao i njihov izgled predstavljaju konfigurabilne opcije tako da administrator može sam definirati kakav mu tip dnevničkih zapisa najviše odgovara te lokaciju njihova pohranjivanja.

Aplikacijski zapisi na Linux operacijskom sustavu također mogu biti pohranjeni na različitim mjestima, no standardno je da se nalaze u /var/log direktoriju, gdje većina aplikacija napravi poddirektorije za svoje dnevničke zapise. Tako npr. Apache web-poslužitelj dnevničke zapise pohranjuje u /var/log/httpd ili /var/log/apache direktorij, ovisno o pojedinoj distribuciji.

5.3.2. Pohranjivanje dnevničkih zapisa

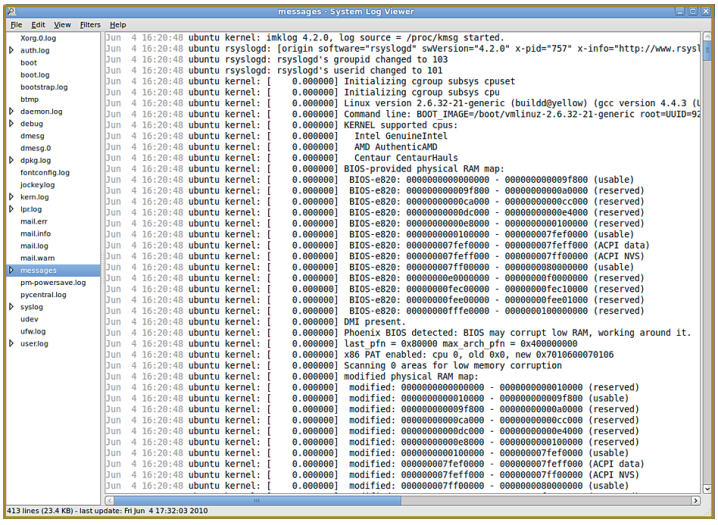
Pohranjivanje dnevničkih zapisa predstavlja drugi bitan korak upravljanja dnevničkim zapisima. Nakon što su ovi zapisi generirani od sustava, aplikacija ili uređaja, potrebno ih je ispravno pohraniti kako bi se osigurala mogućnost njihove analize u slučaju incidenta.

Windows operacijski sustavi, kao što je već bilo spomenuto, pohranjuju sve dnevničke zapise generirane od strane operacijskog sustava i određenih aplikacija/servisa u binarni spremnik koji se naziva Event Log. Riječ je o binarnim datotekama koje upotrebljavaju Microsoftov format zapisivanja dnevničkih zapisa. U svrhu nadgledanja istih može se rabiti Microsoft Event Viewer aplikacija koja omogućuje i pohranjivanje dnevničkih zapisa u druge formate odnosno njihov izvoz. Na sljedećoj je slici prikazan Microsoft Event Viewer



Slika 5.11. Microsoft Event Viewer

Linux operacijski sustavi dnevničke zapise pohranjuju u datoteke i poddirektorije /var/log direktorija. Kako je ovdje uglavnom riječ o običnim tekstualnim datotekama (ne upotrebljava se binarni format), ove je dnevničke zapise moguće pregledavati bez potrebe za upotrebom vanjskih programa, no za jednostavnije pregledavanje dnevničkih zapisa obično se rabe aplikacije poput Log Viewera koji navedene tekstualne datoteke prikazuju u grafičkom okruženju. Log Viewer prikazan je na sljedećoj slici.



Slika 5.12. Log Viewer prikazuje Linux dnevničke zapise

Moderni informacijski sustavi u svrhu bolje zaštite dnevničkih zapisa od možebitnog neželjenog događaja kompromitiranja poslužitelja osim lokalnog pohranjivanja dnevničkih zapisa automatski zapise šalju i na središnji poslužitelj za pohranu dnevničkih zapisa.

Na ovaj se način osigurava dostupnost, čak i u slučaju da je napadač kompromitirao poslužitelj i pobrisao dnevničke zapise koji upućuju na njegove maliciozne aktivnosti. Naime, isti će dnevnički zapisi biti pohranjeni i na središnji poslužitelj za prikupljanje dnevničkih zapisa što napadaču predstavlja još jedan poslužitelj koji u tom slučaju treba kompromitirati.

Budući da središnji poslužitelj za prikupljanje dnevničkih zapisa treba imati samo tu ulogu, prateći korake opisane za ojačavanje sigurnosti operacijskog sustava u poglavlju 5.2 moguće je sigurnosni rizik od kompromitiranja ovog poslužitelja znatno reducirati te na taj način osigurati jednu kopiju dnevničkih zapisa. U izrazito osjetljivim okruženjima moguće je čak koristiti se i posebnim uređajima koji automatski zapisuju dnevničke zapise u stvarnom vremenu na medij koji dopušta samo pisanje (ali ne i modificiranje), čime se i na hardverskoj razini napadač sprječava u možebitnom brisanju dnevničkih zapisa.

Sustavi koji centralizirano prikupljaju dnevničke zapise danas vrlo često omogućuju i upravljanje s jednog mjesta te analizu i korelaciju, kao što je navedeno u nastavku. Ovakvi se sustavi nazivaju SIEM (engl. *Security Information and Event Management*). Neki od poznatijih SIEM paketa uključuju ArcSight, Splunk, Loglogic i RSA enVision.

5.3.2.1 *Syslog*

Syslog protokol predstavlja jedan od pokušaja standardizacije dnevničkih zapisa i omogućavanja jednostavnog slanja istih na središnji poslužitelj. Syslog protokol definiran je još osamdesetih godina prošlog stoljeća i danas je uglavnom prihvaćen na Linux operacijskim sustavima, iako postoje agenti i za Windows operacijske sustave koji mogu izvorne Windows dnevničke zapise generirane u Windows Event Log formatu konvertirati u Syslog te na taj način jednostavno slati dnevničke zapise na udaljeni središnji poslužitelj.

Glavna značajka Syslog protokola je definicija dva parametra svakog dnevničkog zapisa.

- Vrsta dnevničkog zapisa upotrebljava se za određivanje izvora navedenog dnevničkog zapisa. Ovaj je parametar koristan kada se dnevnički zapisi prosljeđuju na neki središnji poslužitelj te omogućuje identifikaciju

izvora zapisa (npr. je li to bio autentikacijski servis, operacijski sustav ili nešto treće).

- Kritičnost dnevničkog zapisa definirana je na skali od 0 do 7, gdje 0 predstavlja kritičan zapis dok 7 predstavlja zapis najmanje kritičnosti. Kritičnost nam dopušta automatiziranje odluka pri primanju navedenog dnevničkog zapisa. Na primjer, ako je s udaljenog poslužitelja primljen zapis s kritičnosti 0, to je znak da se administrator treba uzbuniti budući da je riječ o događaju koji je znatno utjecao na dostupnost poslužitelja.

Iako je sam protokol vrlo jednostavan te postoji čitav niz aplikacija koje omogućuju njegovu upotrebu, danas se njime uglavnom koristi na Linux operacijskim sustavima, gdje je podržan inicijalnom instalacijom.

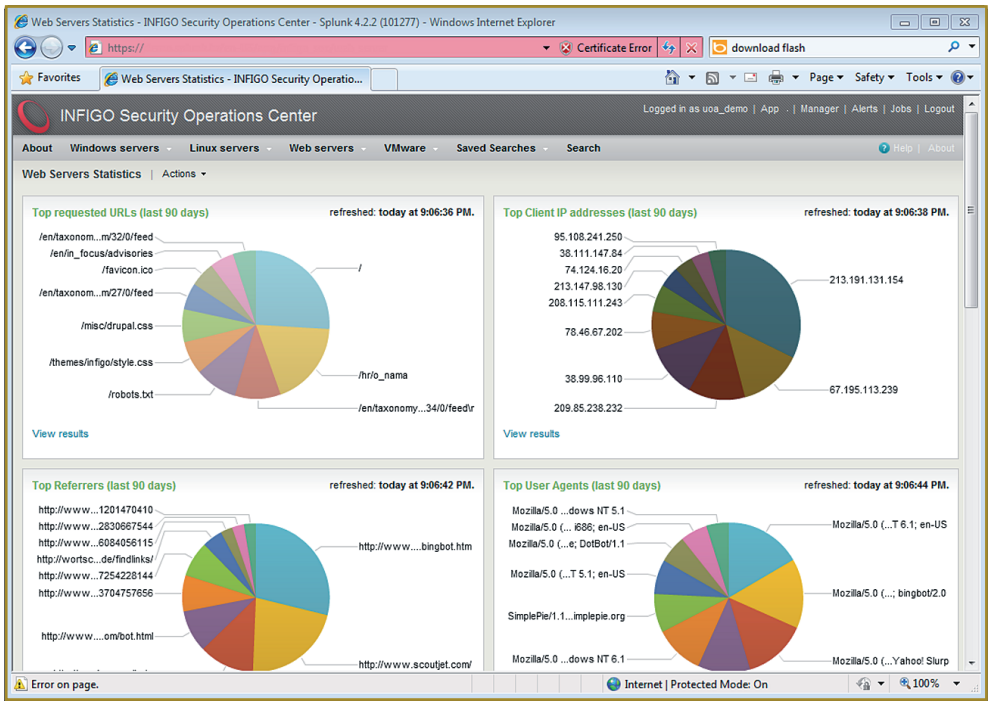
5.3.3. Analiza i korelacija dnevničkih zapisa

Nakon što su dnevnički zapisi prikupljeni na središnji poslužitelj vrlo se često nad njima provodi analiza i korelacija. Cilj analize i korelacije je identifikacija događaja koji odstupaju od uobičajenih ili predstavljaju upozorenje na potencijalni sigurnosni incident.

Na primjer, ako se za pojedini korisnički račun identificira veliki broj neispravnih pokušaja prijava, vrlo je jednostavno zaključiti da je riječ o jednom od prije spomenutih napada na zaporke u poglavlju 5.1.1. Korelacija se ne ograničava na ovako relativno jednostavne analize već može sadržavati i vrlo kompleksne uvjete na osnovi kojih je moguće identificirati neuobičajeno stanje sustava.

Cilj analize i korelacije dnevničkih zapisa je, uz rano uzbunjivanje i upućivanje na neuobičajene događaje, i olakšavanje posla administratoru. Naime, današnji sustavi nerijetko tijekom normalnog rada generiraju tisuće, desetke tisuća pa nekada čak i milijune događaja dnevno. Jasno je da se ovako velika količina događaja ne može ručno pregledavati pa se administratori danas služe prethodno spomenutim SIEM alatima.

Osim analize i korelacije dnevničkih zapisa, SIEM alati se često upotrebljavaju i za vizualizaciju događaja. Pri obradi velikog broja zapisa, njihovo vizualno prikazivanje može administratoru itekako olakšati posao identifikacije neuobičajenog događaja. Zbog toga se za prikaz svih događaja vrlo često upotrebljavaju različite vizualizacije, od kojih je vizualizacija Splunk SIEM alata prikazana na sljedećoj slici.



Slika 5.13. Vizualizacija Splunk SIEM alatom

5.3.4. Arhiviranje dnevničkih zapisa

Zadnji korak u procesu upravljanjem dnevničkim zapisima jest njihovo arhiviranje. Kao što je već rečeno, različiti standardi, ali i zakonske regulative, postavljaju različite zahtjeve pred arhiviranje dnevničkih zapisa. Pri odlučivanju koliko će se dugo prikupljeni dnevnički zapisi držati arhivirani na trakama ili drugim medijima potrebno je proučiti relevantne standardne i druge zakonske regulative.

Pri arhiviranju, ali i skupljanju od strane SIEM sustava, izvorni se dnevnički zapisi komprimiraju kako bi se smanjila diskovna potrošnja pohrane istih. Ovdje je potrebno napomenuti da svi SIEM sustavi moraju uvijek sačuvati izvorni oblik dnevničkog zapisa, tako da proces komprimiranja mora biti bez gubitaka. Ovo je potrebno kako bi se osigurala vjerodostojnost dnevničkih zapisa u slučaju ozbiljnog sigurnosnog incidenta koji može završiti na sudu. Ovisno o pojedinoj organizaciji, neke su obvezne dulje čuvati određene zapise (npr. telekomi moraju čuvati informacije o korisnicima ADSL veza kroz razdoblje propisano zakonom).

5.4. Virtualizacija

U posljednjih nekoliko godina virtualizacija je postala dijelom svake tvrtke. Velike mogućnosti uštede te jednostavnost proizvoda koji nude virtualizaciju doveli su do velike popularnosti ovih proizvoda.

Virtualizacija se odnosi na simuliranje hardvera i softvera na kojem su pokrenute druge softverske aplikacije. Ovo simulirano okruženje naziva se virtualnim strojem (engl. *Virtual machine*). Postoji puno načina virtualizacije koji se upotrebljavaju već čitav niz godina, no tema ovog poglavlja je način virtualizacije koji se naziva još i potpunom virtualizacijom. Potpuna virtualizacija simulira virtualni stroj na koji se instalira operacijski sustav s pripadajućim aplikacijama, kao da je riječ o fizičkom poslužitelju. Operacijski sustav u načelu nije niti „svjestan" da je pokrenut na virtualnom poslužitelju, koji simulira sva sučelja koja operacijski sustav i inače upotrebljava s fizičkim poslužiteljima.

Moderna arhitektura procesora omogućila je jednostavnu i relativno sigurnu implementaciju pune virtualizacije, no potrebno je napomenuti da su pojedini sigurnosni rizici i potencijalne sigurnosne ranjivosti navedene u nastavku ovog poglavlja specifične za virtualne strojeve.

5.4.1. Tipovi virtualizacije

Ovisno o pojedinoj implementaciji, razlikuju se dva tipa virtualizacije: izravna i posredna.

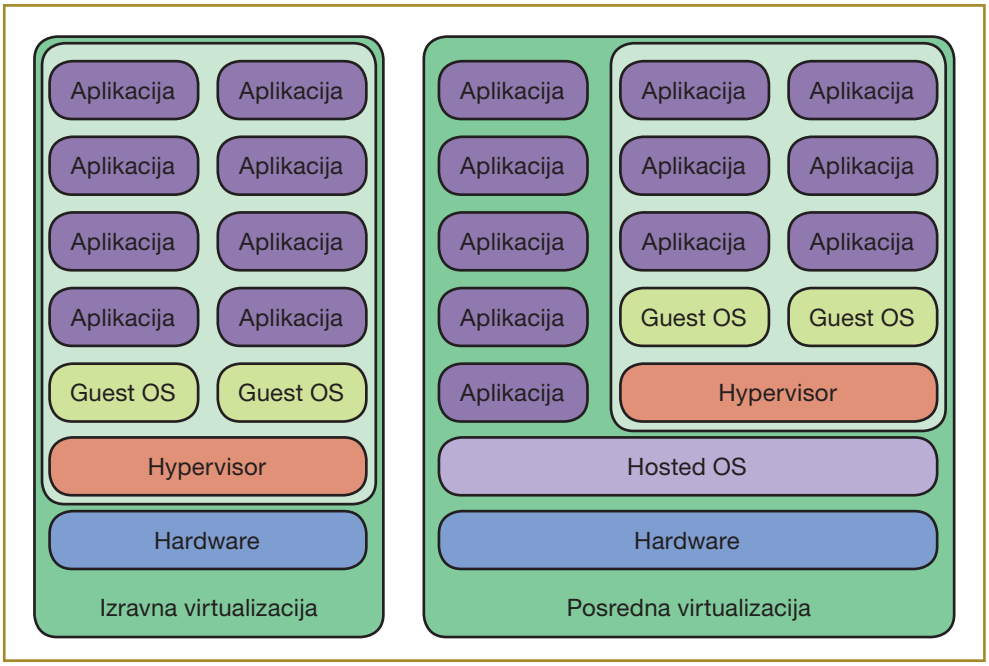
- Izravna virtualizacija na hardveru (engl. *Bare metal*) – sustav za upravljanje virtualnim strojevima, koji se još naziva i VMM (engl. *Virtual Machine Manager*), odnosno Hypervisor pokrenut je izravno na hardveru poslužitelja koji služi za virtualizaciju.

  Ovakve implementacije virtualizacije omogućuju najbolje performanse budući da troše minimalnu količinu resursa za upravljanje virtualnim strojevima.

  Primjeri ovakve virtualizacije uključuju VMWare ESXi (vSphere) programski paket i Microsoftov Hyper-V.
- Posredna virtualizacija (engl. *Hosted virtualization*) zasniva se na programskim paketima koji se instaliraju na već postojeći operacijski sustav poput Windowsa ili Linuxa. Programski paketi za virtualizaciju dalje omogućuju instaliranje virtualnih strojeva koji su pokrenuti u virtualnom okruženju, jednako kao i u prethodnom slučaju.

Postoji veliki broj programskih paketa koji primjenjuju posrednu virtualizaciju, poput VMWare Workstation, Microsoft Virtual PC, VirtualBox i drugih.

Na sljedećoj je slici prikazano načelo rada oba tipa virtualizacije.



Slika 5.14. Tipovi virtualizacije

Kao što se može vidjeti na potonjoj slici, u oba slučaja upravljanje virtualnim strojevima provodi Hypervisor, posebni dio koda koji omogućuje simuliranje hardverskih komponenti te normalan rad virtualnog stroja. Iako Hypervisor predstavlja kompleksan paket, osnovna funkcionalnost je relativno jednostavna: kada operacijski sustav u virtualnom stroju izvršava bilo kakve standardne programe, Hypervisor dopušta njihovo izvršavanje izravno na fizičkom procesoru poslužitelja. No ako operacijski sustav na virtualnom stroju zahtijeva pristup nekim hardverskim komponentama, npr. mrežnoj kartici, Hypervisor presreće ovaj upit, obavlja traženi zadatak te vraća odgovor operacijskom sustavu virtualnog stroja koji nije bio niti svjestan da odgovor nije dobio od fizičke mrežne kartice (u navedenom primjeru) već od Hypervisora.

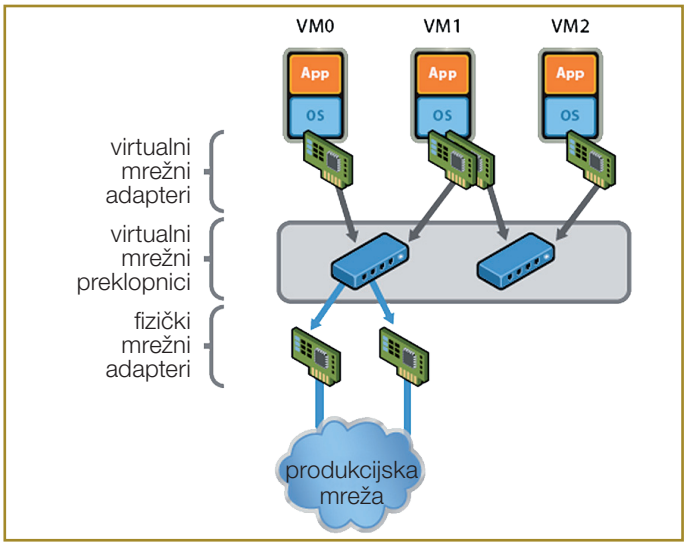
Iz opisanog je modela rada jasno da implementacija Hypervisora izravno na hardveru ostvaruje bolje performanse od posredne implementacije, budući da kod posredne implementacije i operacijski sustav instaliran na fizičkom računalu troši performanse računala.

Kod oba se tipa virtualizacije operacijski sustavi koji su instalirani u virtualnim strojevima, koji se još nazivaju i gostujući strojevi, koriste simuliranim hardverom te nisu „svjesni“ činjenice da su u virtualnom stroju. Simuliran hardver uključuje sve komponente normalnog, fizičkog poslužitelja:

- procesor,
- memoriju,
- tvrde diskove, uključujući i vanjske jedinice poput disketnih ili CD-ROM uređaja,
- mrežne kartice, odnosno mrežni priključci,
- grafičke i zvučne kartice,
- tipkovnicu i miš.

Ovisno o pojedinom virtualizacijskom rješenju, pri je implementaciji moguće upotrebljavati virtualne računalne mreže, kao i virtualne tvrde diskove.

Virtualne računalne mreže omogućuju uspostavljanje komunikacijskih veza između virtualnih strojeva bez potrebe za stvarnim, fizičkim računalnim mrežama. Naravno, ako je potrebno ostvariti komunikaciju s drugim poslužiteljima, preko računalnih je mreža potrebno virtualnu računalnu mrežnu spojiti na pravu, fizičku. Većina virtualnih rješenja ovakve kompleksnije postavke omogućuje upotrebom virtualnih preklopnika te izravnim spajanjem virtualnih strojeva na fizičku računalnu mrežu ili putem mogućnosti prepisivanja IP adresa (engl. *Network Address Translation*, više o NAT-u dostupno je u poglavlju 6).



Slika 5.15. Primjer implementacije kompleksne virtualne mreže na VMWare ESXi poslužitelju upotrebom virtualnih preklopnika

5.4.2. Sigurnost virtualnih strojeva

Osnovni zahtjevi za sigurnost koji se postavljaju pred virtualne strojeve ili poslužitelje na kojima su pokrenuti programski paketi za virtualizaciju jednaki su fizičkim poslužiteljima te zadržavaju sve standardne sigurnosne zahtjeve CIA trokuta, kao što je opisano u poglavlju 1.1.

No zbog kompleksnih postavki i same implementacije koja može biti vrlo složena posebnu je pažnju potrebno obratiti na sigurnost sljedećih komponenti sustava:

- sigurnost Hypervisora – Hypervisor je kritična komponenta sustava virtualizacije budući da osim upravljanja virtualnim strojevima predstavlja i njihova sučelja prema fizičkim komponentama poslužitelja. Sigurnosna ranjivost u Hypervisoru kritični je sigurnosni rizik s obzirom da potencijalnom napadaču koji je kompromitirao gostujući stroj omogućuje preuzimanje kontrole nad Hypervisorom, a samim tim i preuzimanje kontrole nad svim ostalim gostujućim strojevima, budući da Hypervisor ima neograničenu kontrolu nad svim aspektima gostujućih strojeva, uključujući njihovu memoriju i tvrdi disk.
  Iako su popularni virtualizacijski paketi do sada imali vrlo mali broj ovakvih sigurnosnih ranjivosti, nikako ih se ne smije zanemariti. U tu je svrhu potrebno redovito pratiti zbivanja vezana za virtualizacijske proizvode te instalirati potrebne sigurnosne zakrpe za Hypervisore;
- sigurnost gostujućih strojeva – budući da je ovdje riječ o virtualnim strojevima, sa sigurnosnog se pogleda oni niti u čemu ne razlikuju od klasičnih, fizičkih poslužitelja. To znači da je na njima potrebno poduzeti sve aktivnosti koje bi se i standardno poduzele na fizičkim poslužiteljima, poput procesa ojačavanja sigurnosnih postavki poslužitelja, konfiguriranja ispravnog bilježenja dnevničkih zapisa, instaliranja antivirusnih programa i slično;
- osiguravanje komunikacije između virtualnih poslužitelja i drugih poslužitelja – budući da se pri ostvarivanju komunikacije između virtualnih strojeva i drugih poslužitelja u većim informacijskim sustavima uvijek upotrebljavaju kombinacije virtualnih i stvarnih računalnih mreža, posebnu je pažnju potrebno posvetiti sigurnosti virtualnih preklopnika te konfiguraciji navedenih virtualnih mreža.
  Naime, pri korištenju virtualnim mrežama ljudska pogreška predstavlja povećani sigurnosni rizik budući da sada administrator Hypervisora može pogrešnom konfiguracijom postaviti gostujući stroj u neku drugu računalnu mrežu na kojoj se možda nalaze osjetljivi podaci.

Jednako tako, budući da se u svrhu implementacije virtualnih računalnih mreža gotovo uvijek rabi segmentiranje na načelu definiranja VLAN-ova (engl. *Virtual LAN*), posebnu je pažnju potrebno posvetiti mapiranju istih na fizičke mrežne kartice poslužitelja;

- Ovisno o osjetljivosti podataka odnosno virtualnih strojeva koji se implementiraju preporučuje se segmentiranje fizičkih strojeva, odnosno upotreba pojedinih fizičkih strojeva samo za virtualne strojeve koji imaju jednake zahtjeve na sigurnost.
  Ovo znači da ako se upotrebljavaju virtualni strojevi za implementaciju javnih servisa dostupnih s interneta (npr. javne web-stranice tvrtke), onda je potrebno izbjegavati da se na isti fizički poslužitelj instalira i virtualni stroj na kojem se nalaze neke interne baze podataka s osjetljivim podacima. Na ovaj se način pokušava reducirati sigurnosni rizik koji nastaje od potencijalnog kompromitiranja virtualnog stroja dostupnog s interneta te, u najgorem scenariju, napadačevo preuzimanje kontrole Hypervisora, a time i preuzimanje kontrole nad cijelim fizičkim poslužiteljem i drugim virtualnim strojevima koji su pokrenuti na tom poslužitelju.

6. Mrežni sloj

Mrežna komunikacija između poslužitelja i klijentskih računala predstavlja osnovni zahtjev informacijske sigurnosti: dostupnost podataka. Bez računalne mreže korisnici ne mogu pristupiti svojim podacima (osim u slučaju da su pohranjeni na lokalnom računalu) niti mogu provoditi svoje poslovne zadatke; bez računalne mreže elektroničko poslovanje, o kojem ovisi današnja ekonomija, ne može funkcionirati.

Iz navedenih je zahtjeva jasno da se pred mrežnu komunikaciju odnosno pred mrežni sloj višeslojnog strateškog modela sigurnosti stavljaju veliki sigurnosni zahtjevi.

Kada danas govorimo o računalnim mrežama, uglavnom se govori o mrežama temeljenim na IPv4 (engl. *Internet Protocol version 4*) protokolu. Iako se u pojedinim informacijskim sustavima tvrtki mogu naći i drugi, zastarjeli mrežni protokoli, elektroničko poslovanje, kao i komunikacija putem interneta danas ovise o IPv4 protokolu.

IPv6 protokol, nasljednik IPv4, iako se u zadnje vrijeme sve više upotrebljava, danas još uvijek nije jako rasprostranjen te se vrlo rijetko može naći na internim računalnim mrežama tvrtki. Iako je adresni prostor IPv4 protokola znatno ograničen na maksimalan teoretski broj od 4,294,967,296 IPv4 adresa (s time da su određene adrese ograničene ili se rabe u posebne svrhe), IPv6 još uvijek nije dovoljno zaživio.

Iako se već dulje vremena predviđa nestanak slobodnih IPv4 adresa, što je svakako točno gledajući rezervaciju slobodnih IPv4 prostora, činjenicu da je komunikacija na internetu moguća, iako je broj računala i poslužitelja premašio maksimalne četiri milijarde, možemo zahvaliti tehnologijama poput NAT-a (engl. *Network Address Translation)*.

NAT omogućuje izgradnju lokalnih računalnih mreža koje se služe privatnim IP adresama te, u slučaju da računala na takvim lokalnim računalnim mrežama trebaju komunicirati s drugim računalima na internetu, prepisuje adrese u javne, rutabilne i dostupne adrese. Više o NAT funkcionalnosti objašnjeno je u poglavlju 6.2.2, budući da se NAT funkcionalnost danas obično implementira na vatrozidima.

Mrežna komunikacija koja predstavlja, kao što je već rečeno, osnovni zahtjev na dostupnost podataka, ujedno je i najveći sigurnosni problem – uz legitimne korisnike, dostupnost preko računalne mreže mogu iskoristiti i napadači ne bi li ostvarili neovlašteni pristup poslužiteljima te došli do svog ultimativnog cilja, osjetljivih podataka.

Ograničavanje, odnosno kontrola pristupa jedan je od glavnih sigurnosnih mehanizama koji se implementiraju na mrežnom sloju. U svrhu kontrole pristupa često se upotrebljavaju specijalizirani uređaji poput vatrozida, sustava za detekciju i prevenciju neovlaštenih aktivnosti i slični, o kojima je više riječi dano u nastavku.

Ako već spomenuti CIA trokut preslikamo na zahtjeve koji se postavljaju pred mrežni sloj, dobit ćemo sljedeće zahtjeve:

- povjerljivost podataka – osigurava da podatke mogu vidjeti samo ona tijela u mrežnoj komunikaciji koja imaju dozvolu. Zahtjev za povjerljivošću podataka treba spriječiti nadgledanje podataka od neautoriziranog tijela, odnosno napadača u njihovom prijenosu preko računalne mreže;
- integritet podataka – cilj je prenijeti podatke u njihovom izvornom obliku, bez promjene ili gubitka podataka, bilo da je riječ o namjernoj promjeni (npr. od strane malicioznog korisnika) ili o grešci u prijenosu podataka;
- dostupnost podataka – postiže se da svi korisnici, bez obzira na njihovu lokaciju, mogu pristupiti željenim servisima i podacima putem računalne mreže.

Ako se podaci prenose preko interneta, jasno je da se prenose i preko velikog broja različitih računalnih mreža, koje su pod kontrolom različitih entiteta. Zbog toga je pri dizajnu svakog novog servisa potrebno analizirati sigurnosne zahtjeve koji se postavljaju pred navedeni servis, odnosno pred podatke kojima će taj servis upravljati. Osiguravanje svake od spomenutih komponenti CIA trokuta ovisit će upravo o navedenim sigurnosnim zahtjevima.

Neki od zahtjeva ispunjeni su automatski pomoću prikladnog protokola. Na primjer, putem TCP (engl. *Transmission Control Protocol*) protokola za prijenos podataka, greške u prijenosu bit će automatski detektirane. TCP protokol će ujedno osigurati i da su podaci predani servisu kojih ih prima u ispravnom redoslijedu, čak i ako paketi koji su prenosili te podatke nisu stigli u ispravnom redoslijedu zbog problema u funkcioniranju računalne mreže. S druge strane, niti jedan od navedenih

mehanizama ne postoji u UDP protokolu (engl. *User Datagram Protocol*), u kojem slučaju programer servisa koji upotrebljava UDP mora sam osigurati i implementirati navedene mehanizme kako bi se očuvao integritet podataka.

Isto vrijedi i za povjerljivost podataka, gdje postoji čitav niz protokola, poput SSL-a, koji mogu služiti za osiguravanje povjerljivosti podataka, a koji su detaljnije obrađeni u poglavlju 6.6.

6.1. Praćenje mrežnog prometa

Praćenje mrežnog prometa korisna je aktivnost koja administratorima omogućuje identifikaciju problema u radu računalne mreže. No praćenje mrežnog prometa jednako tako može i malicioznom korisniku dopustiti pregledavanje i analizu paketa.

Ako neki korisnik pristupa osjetljivim podacima preko računalne mreže, a napadač ima pristup mrežnim uređajima preko kojih je uspostavljena komunikacijska veza, jasno je da je narušen osnovni zahtjev za povjerljivošću podataka.

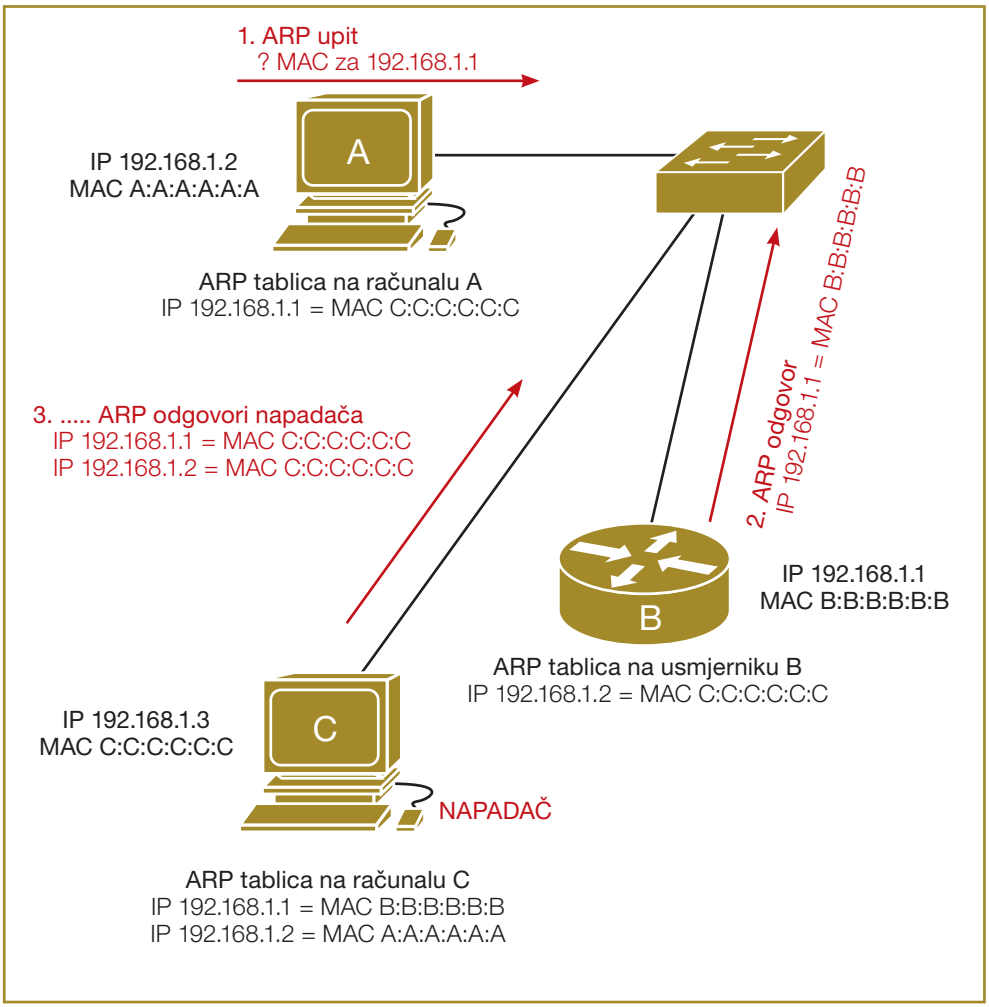
Analiza mrežnih paketa napadaču omogućuje ne samo dolazak do povjerljivih podataka (ako isti nisu bili ispravno zaštićeni od neovlaštenog praćenja) već i provođenje drugih različitih napada na protokol koji se upotrebljava.

Praćenje mrežnog prometa od napadača zahtijeva kontrolu nad mrežnim uređajima kroz koje je uspostavljena komunikacijska veza između klijentskog računala i poslužitelja čiji mrežni promet napadač želi nadgledati. Ako je riječ o lokalnoj računalnoj mreži, moguća su sljedeća dva scenarija:

- upotreba uređaja poput *huba* neovlaštenom korisniku omogućuje vrlo jednostavno pregledavanje mrežnog promet zbog načina funkcioniranja *huba*. Budući da *hub* funkcionira tako da mrežni promet primljen na jednom sučelju automatski šalje na sva ostala sučelja, ako napadač ima pristup na bilo koje računalo spojeno na *hub* može vidjeti mrežni promet svih drugih računala koja su spojena na isti *hub*;
- ako se za spajanje računala upotrebljava preklopnik, kao što je to danas gotovo uvijek slučaj, jedno računalo može vidjeti samo promet usmjeren prema tom računalu, budući da preklopnik u memoriji drži tablicu MAC (engl. *Media Access Control* – jedinstvenih adresa mrežnih kartica) adresa svih spojenih računala te će mrežni promet poslati na određeno sučelje samo ako je zbilja bio namijenjen MAC adresi mrežne kartice računala na tom sučelju.

Kod ovog scenarija, kao što se može vidjeti, ako napadač ostvari pristup na jedno računalo, nije u stanju vidjeti promet namijenjen drugim računalima, osim ako ne provodi određene napade na preklopnik, kao što je navedeno u nastavku.

- Ako napadač ostvari pristup na administracijsko sučelje preklopnika, moguće je služiti se naprednim mogućnostima preklopnika poput zrcaljenja mrežnog prometa (engl. *port mirroring*). Zrcaljenje mrežnog prometa služi upravo za pregledavanje mrežnog prometa i administratoru omogućuje konfiguriranje preklopnika tako da sav promet s jednog ili više sučelja zrcali na treće sučelje. Na taj način napadač može vrlo jednostavno pregledavati mrežni promet namijenjen drugim sučeljima.
- Iz navedenog je jasno da je i na mrežnu opremu, i općenito bilo koje druge uređaje u informacijskom sustavu potrebno primijeniti pravila ojačavanja (više u poglavlju 5.2) kako bi se spriječilo provođenje napada na opremu.
- Napadač može probati provesti neke mrežne napade usmjerene na preklopnike poput preplavljivanja MAC tablica. Naime, preklopnici imaju ograničenu količinu memorije u koju pohranjuju parove MAC adresa, sučelje. Ako napadač počne generirati veliku količinu lažnih MAC adresa, može doći do preplavljivanja MAC tablica u memoriji preklopnika što u konačnici rezultira time da će se preklopnik početi ponašati kao *hub*, budući da neće znati kojem sučelju pripada pojedini paket. Ovakav napad dovodi i do degradacije performansi preklopnika.
- Konačno, napadač može provesti napade na druga klijentska računala u istoj lokalnoj mreži. Riječ je o napadima slanja lažnih ARP paketa. Budući da se za razlučivanje IP adresa u MAC adrese upotrebljava ARP protokol, napadač može slati lažne ARP pakete ne bi li promijenio legitimnu postavku određene IP adrese (za napad se obično rabi IP adresa usmjerivača, što napadaču omogućuje pregledavanje mrežnog prometa koji nije usmjeren lokalnoj računalnoj mreži). Na ovaj način napadač može promijeniti postavke IP na MAC adresa proizvoljnih računala na lokalnoj računalnoj mreži. Primjer napada lažiranjem ARP zapisa dan je na sljedećoj slici.



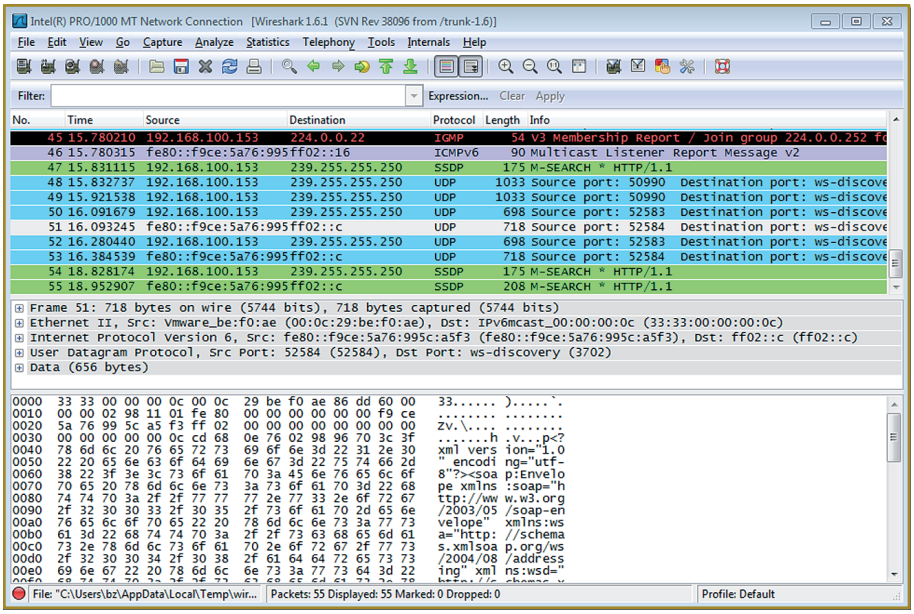
Slika 6.1. Napad lažiranjem ARP zapisa

Ovakav napad naziva se još i engl. *Man-in-the-Middle* (*MitM*) napad, budući da napadaču dopušta pozicioniranje između dvije točke koje žele uspostaviti komunikacijsku vezu te pregledavanje i po potrebi modificiranje njihovog mrežnog prometa.

U slučaju bilo kojeg uspješno provedenog napada na mrežnu opremu napadač je u stanju pregledavati mrežni promet. U svrhu pregledavanja mrežnog prometa obično se rabe specijalizirane aplikacije koje omogućuju prikazivanje i analizu mrežnog prometa te podržavaju čitav niz protokola kako bi olakšali ovaj posao. Dva su najpopularnija takva programska paketa, Wireshark i Tcpdump, opisana u nastavku.

6.1.1.1 Wireshark

Wireshark je najpopularniji grafički program za pregledavanje mrežnog prometa. Dostupan je za brojne operacijske sustave, uključujući i Windows i Linux operacijske sustave. U svrhu prikupljanja mrežnog prometa Wireshark zahtijeva instalaciju libpcap biblioteke na Linux operacijskim sustavima ili Winpcap biblioteke na Windowsima. Wireshark je prikazan na sljedećoj slici.



Slika 6.2. Wireshark

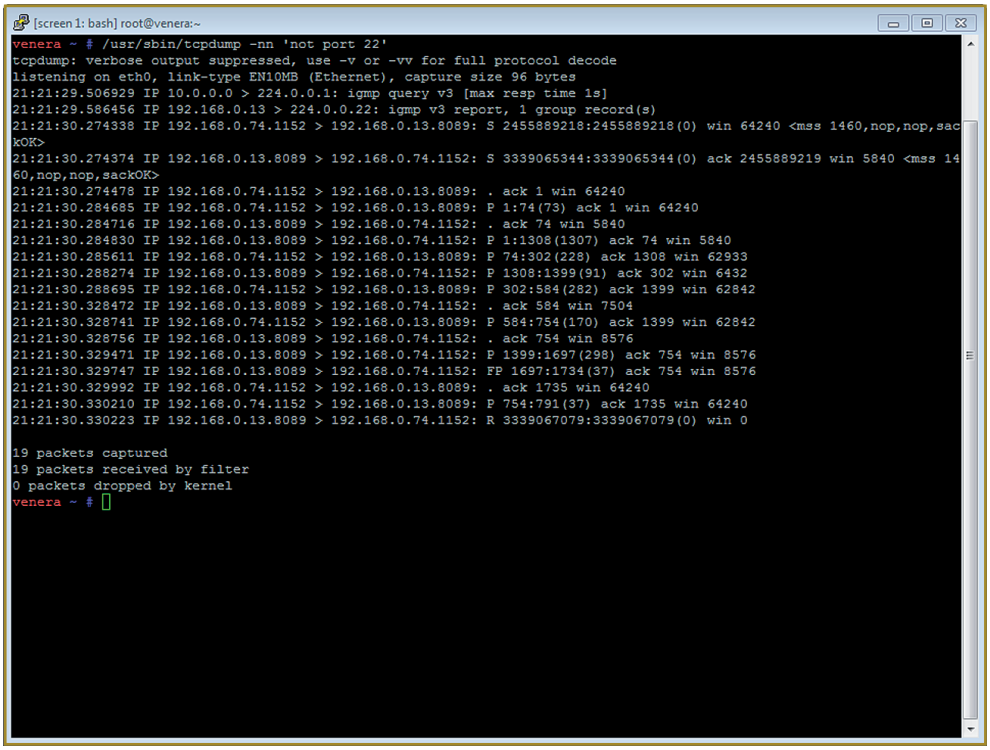
Glavna prednost Wiresharka pred sličnim programima za analizu mrežnog prometa je u iznimno velikom broju modula koji omogućuju analizu različitih protokola (engl. *Dissectors*). Zahvaljujući ovim modulima, Wireshark može ispravno prikazati zaglavlja više od stotinu različitih protokola. Jednako tako, zbog mogućnosti precizne analize različitih protokola Wireshark može jednostavno prikazati kompletne sjednice pojedinih protokola, bez obzira na način njihove uspostave. Na primjer, pri praćenju TCP komunikacijske veze Wireshark može izdvojiti sve pakete koji pripadaju pojedinoj komunikacijskoj vezi. Wireshark može uspješno analizirati i više protokole, poput HTTP-a, gdje je moguće čak i izvoženje datoteka detektiranih u HTTP sjednicama.

U Wiresharku je pronađen veliki broj sigurnosnih ranjivosti, uglavnom vezanih za prepisivanja podataka u spremniku (više o ovoj kategoriji sigurnosnih ranjivosti dostupno je u poglavlju 4.1.1), zbog čega se preporučuje redovito instaliranje najnovije inačice ovog programskog paketa.

6.1.1.2 *Tcpdump*

Tcpdump je program u naredbenom retku koji omogućuje jednostavno pregledavanje mrežnog prometa. Riječ je o programu koji je naročito popularan na Linux distribucijama, budući da dolazi s inicijalnom instalacijom većine Linux distribucija i ne zahtijeva instalaciju nikakvih dodatnih paketa. Poput Wiresharka, Tcpdump na Linux operacijskim sustavima za ispravno funkcioniranje zahtijeva libpcap biblioteku. Na Windows sustavima dostupna je Windows inačica programa Windump, koja za funkcioniranje zahtijeva već spomenutu Winpcap biblioteku.

Za razliku od Wiresharka, Tcpdump omogućuje samo osnovnu analizu paketa, odnosno IP, TCP i UDP zaglavlja te još nekih protokola poput ARP-a, ICMP-a i slično. Sve ostale protokole Tcpdump nije u stanju analizirati te može prikazati samo sadržaj paketa, što znači da korisnik ovoga paketa mora sam provesti analizu takvog mrežnog prometa. Tcpdump je još uvijek vrlo koristan program zbog već rečene činjenice da je inicijalno instaliran na većini Linux distribucija te zbog iznimno malih zahtjeva za resursima. Na sljedećoj je slici prikazan Tcpdump paket.



Slika 6.3. Tcpdump program

6.2. Filtriranje mrežnog prometa

U svrhu ograničavanja pristupa računalnim mrežama i poslužiteljima na računalnim mrežama upotrebljavaju se različite metode filtriranja mrežnog prometa.

Pravila filtriranja mrežnog prometa mogu se postaviti na različitim točkama pristupa računalnoj mreži: na usmjerivačima, vatrozidima te lokalnim, poslužiteljskim vatrozidima.

6.2.1. Filtriranje na usmjerivačima

Usmjerivači podržavaju osnovne mogućnosti filtriranja prema pristupnim listama (engl. *Access Control List* – ACLs). Riječ je o jednostavnim listama koje administrator usmjerivača može definirati za pojedina sučelja na usmjerivaču. Liste dopuštaju ili blokiraju pojedini promet, a administrator ih može definirati prema izvorišnim ili ciljnim IP adresama te portovima, kao i kompletnim računalnim mrežama.

ACL-ovi predstavljaju izvrstan način osnovnog filtriranja neželjenog prometa te ih se definitivno preporučuje implementirati na graničnim usmjerivačima, ali i na svim drugim mjestima gdje se može ograničiti mrežni promet.

Pri implementiranju ACL-ova potrebno je slijediti politiku bijelih lista (engl. *white lists*) – usmjerivači bi trebali inicijalno blokirati sav promet, osim onog koji je eksplicitno dopustio administrator. Na taj se način postiže najviša razina sigurnosti. Drugi pristup, koji se rjeđe rabi, je pristup crnih lista (engl. *black lists*), gdje se dopušta sav promet osim onog koji je eksplicitno zabranjen. Ovaj se pristup ne preporučuje jer je puno teže ili praktički nemoguće napraviti listu svog mrežnog prometa koji se ne želi propuštati. Dodatno, veliki ACL-ovi opterećuju usmjerivače tako da pristup bijelih lista zahtijeva puno manje resursa, budući da će u konačnici i lista dopuštenog mrežnog prometa biti relativno mala u usporedbi sa zabranjenim mrežnim prometom.

Primjer ACL-ova za usmjerivač koji se koristi bijelim listama (zadnja naredba eksplicitno brani sav mrežni promet osim onoga koji je dozvoljen) dan je na sljedećoj slici.

```
!--- This command is used to permit IP traffic from 10.1.1.0
!--- network to 172.16.1.0 network. All packets with a source
!--- address not in this range will be rejected.

access-list 101 permit ip 10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255
```

```
!--- This command is used to permit IP traffic from 10.1.1.0
!--- network to 172.16.1.0 network. All packets with a source
!--- address not in this range will be rejected.

access-list 102 permit ip 10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255
access-list 102 deny ip any any
```

Slika 6.4. Primjer ACL liste

Ovdje je potrebno naglasiti da usmjerivači nisu u stanju analizirati protokole već mogu samo primijeniti ACL-ove na dolazni i odlazni mrežni promet. Drugim riječima, kompleksni protokoli poput FTP-a, koji zahtijevaju dinamičko otvaranje portova te istodobno upotrebljavaju više portova mogu predstavljati problem za filtriranje na razini usmjerivača. Zbog toga se često rabe napredniji uređaji za filtriranje mrežnog prometa koji su u stanju analizirati mrežne pakete, poput vatrozida.

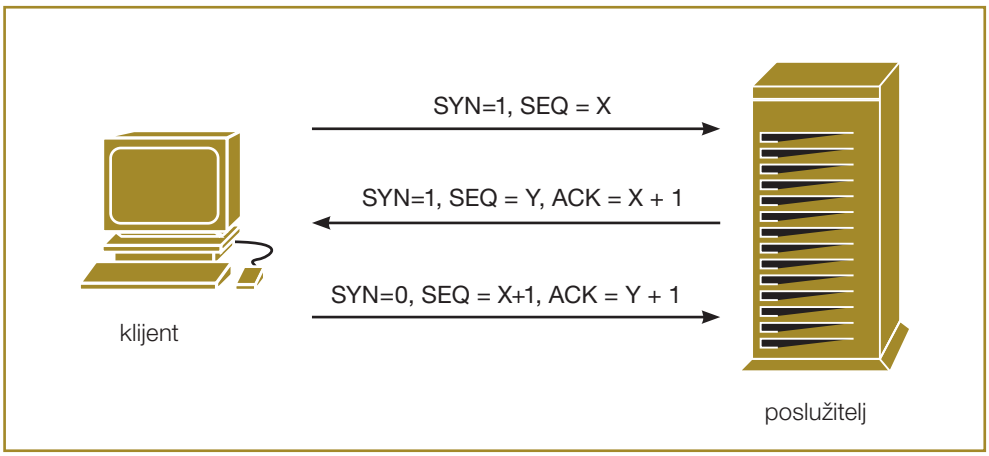
6.2.2. Filtriranje pomoću vatrozida

Vatrozid (engl. *Firewall*) je osnovni sigurnosni uređaj namijenjen kontroli pristupa putem računalnih mreža. Zbog svoje specifične uloge vatrozidi danas predstavljaju neizostavan element svakog informacijskog sustava i kontrola mrežnog pristupa obično ovisi o ovim uređajima.

Vatrozidi su obično napravljeni u vidu hardverskih uređaja iako postoje i mnogi softverski vatrozidi. Ključna osobina vatrozida je da razdvaja dvije ili više računalnih mreža. Kod klasičnih, fizičkih vatrozida ovo se razdvajanje najčešće postiže putem više fizičkih mrežnih sučelja – jedna računalna mreža spojena je na prvo mrežno sučelje, druga na drugo itd. Vatrozid zatim provodi usmjeravanje mrežnih paketa između ovih sučelja te ujedno, tijekom usmjeravanja, nad mrežnim paketima provodi i sigurnosnu politiku vatrozida. Poznatiji hardverski vatrozidi danas uključuju Check Point, koji je vodeći proizvod na tržištu, Cisco ASA te Juniper vatrozide.

Sigurnosna politika vatrozida u osnovi je vrlo jednaka ACL-ovima usmjerivača objašnjenim u prethodnom poglavlju, no ključna je razlika činjenica da vatrozid detaljno analizira mrežni promet te propušta pakete samo ako zadovoljavaju parametre određenog protokola. Ovakva se analiza mrežnog prometa naziva engl. *stateful analysis*. U nastavku je dan primjer gdje detaljna analiza prometa može detektirati potencijalni napad.

- Kod TCP mrežnog prometa svaka komunikacijska veza mora početi s trostrukim rukovanjem (engl. *Three way handshake*), kao što je prikazano na sljedećoj slici.



Slika 6.5. Slika 66: Trostruko rukovanje kod TCP veza

Kao što se vidi, prvi mrežni paket mora biti poslan od klijentskog računala te mora imati postavljenu SYN zastavicu. Nakon što je TCP komunikacijska veza uspostavljena (nakon trostrukog rukovanja), svi ostali mrežni paketi sadržavaju podatke koji se prenose između klijenta i poslužitelja te imaju samo postavljenu ACK zastavicu (osim zadnjih paketa koji signaliziraju prekid komunikacijske veze).

Usmjerivači, kao što je rečeno u poglavlju 6.2.1, svoje sigurnosne politike temelje na ACL listama. ACL liste se na pakete primjenjuju izravno – ako izvorišna ili ciljna IP adresa paketa koji je usmjerivač primio odgovara nekoj IP adresi ili računalnoj mreži u ACL listi, onda se pravila definirana ACL listom i primjenjuju. No usmjerivači ovdje imaju jedan ključni problem – budući da ne prate stanje mrežnih sjednica, kada prime paket s ACK zastavicom postavljenom, misle da je riječ o prethodno uspostavljenoj sjednici te ga automatski propuštaju. Ovaj problem u radu usmjerivača napadači mogu iskoristiti te slanjem mrežnih paketa s postavljenom ACK zastavicom zaobići ACL-ove i na taj način ipak poslati neke mrežne pakete u štićenu mrežu.

Za razliku od usmjerivača, vatrozidi u memoriji drže podatke o svim uspostavljenim mrežnim sjednicama. Drugim riječima, kada vatrozid primi paket s postavljenom ACK zastavicom, on prvo u memoriji traži pripa-

dajuću sjednicu – ako je ovaj paket legitiman, moralo mu je prethoditi trostruko rukovanje (zbog činjenice da sve TCP komunikacijske veze moraju početi trostrukim rukovanjem). Ako vatrozid pronađe podatke o sjednici u memoriji, riječ je o legitimnom paketu, u protivnom se paket odbacuje bez obzira na to je li možda čak i dopušten sigurnosnom politikom vatrozida.

Na ovaj način vatrozidi omogućuju puno preciznije provođenje sigurnosne politike, a samim time i višu razinu sigurnosti informacijskog sustava. Dodatno, moderni vatrozidi danas mogu pravilno analizirati i kompleksne protokole poput FTP-a, koji zahtijevaju uspostavljanje dvije sjednice (jedne kontrolne i jedne podatkovne) te, ako su tako konfigurirani, čak i automatski dopuštati mrežni promet prema pojedinim portovima, ako komunikacijski protokol koji se analizira to zahtijeva. Zbog svega navedenog vatrozidi predstavljaju nezaobilaznu komponentu sigurnosti informacijskih sustava danas.

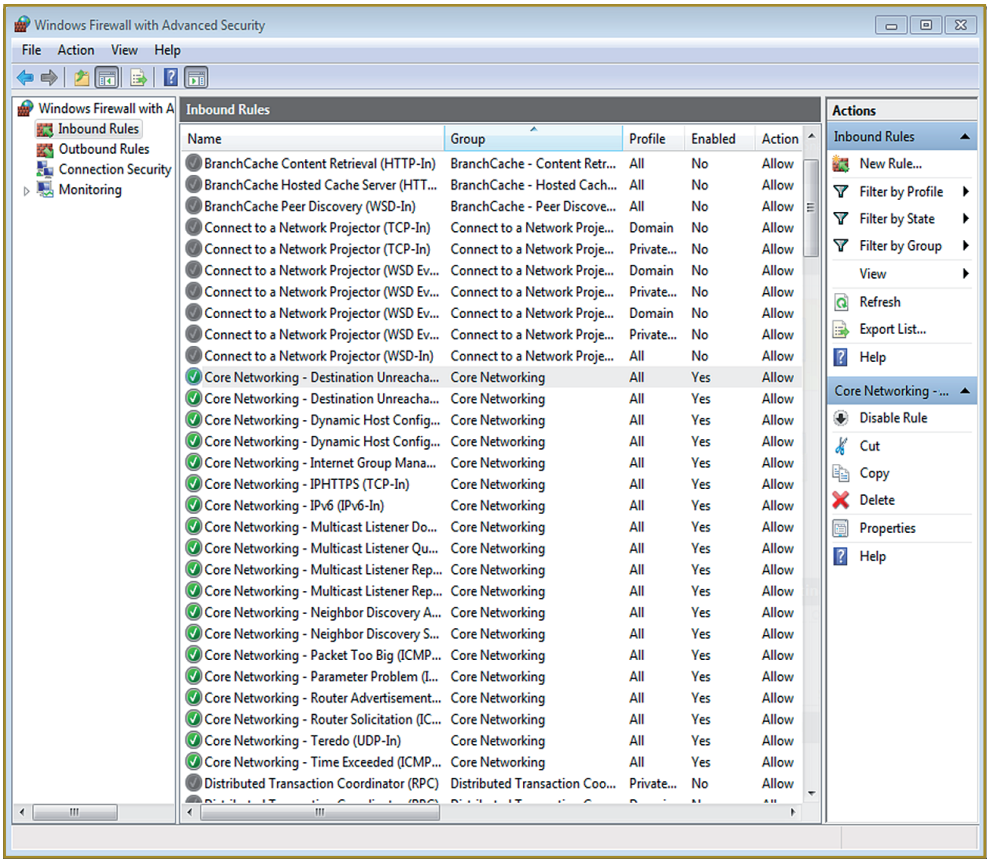
Osim navedenih hardverskih vatrozida, danas se često upotrebljavaju i softverski vatrozidi. Riječ je zapravo samo o programskom paketu koji simulira hardverski vatrozid – ovakvi se paketi obično instaliraju na ojačane operacijske sustave (više o postupku ojačavanja sigurnosti operacijskih sustava spomenuto je u poglavlju 5.2) te se, jednako kao i u prethodnom slučaju, služe višestrukim mrežnim karticama kako bi odvajali računalne mreže odnosno primjenjivali sigurnosne politike na njih.

Porastom popularnosti virtualizacije danas se vrlo često sreću i virtualni vatrozidi. Osim što je riječ o, ponovno, softverskim proizvodima koji se u nekim slučajevima mogu čak instalirati i u virtualne strojeve, glavna je značajka virtualnih vatrozida da svoje sigurnosne politike primjenjuju na pojedine VLAN-ove. Ovo znači da virtualni vatrozid, čak i kada je implementiran na fizičkom računalu, ne mora nužno imati višestruke mrežne kartice za odvajanje računalnih mreža već mu je dovoljna jedna mrežna kartica koja podržava 802.1Q standard te na taj način može ispravno procesirati i kreirati mrežne pakete s različitim VLAN zaglavljima. Ostale funkcionalnosti ovakvih vatrozida identične su onim standardnim, hardverskim, s jedinom razlikom što se sigurnosne politike vatrozida u ovom slučaju mogu primijeniti i na pojedine VLAN-ove.

Konačno, poslužiteljski vatrozidi (engl. *Host based firewalls*) predstavljaju softverske komponente koje se instaliraju izravno na poslužitelje. Riječ je o vatro-

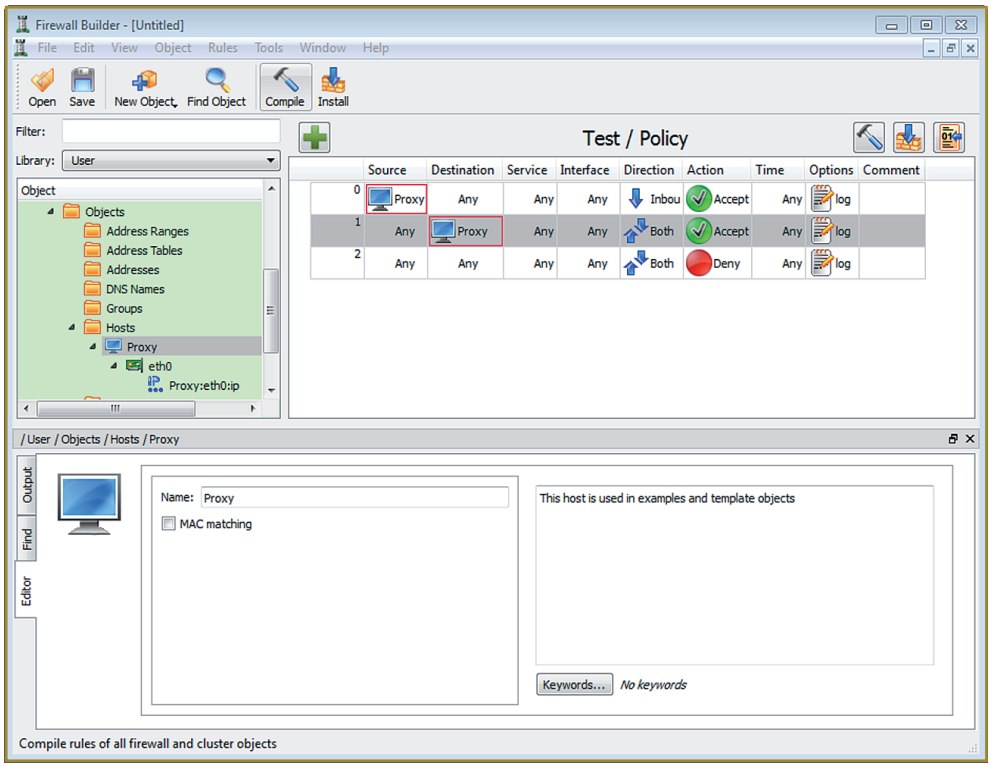
zidima koji ne odvajaju višestruke računalne mreže već im je funkcija u zaštiti pojedinog poslužitelja. Ovi vatrozidi, dakle, samo filtriraju dolazni i odlazni promet prema i s poslužitelja na kojem su instalirani te na taj način dodatno štite navedeni poslužitelj čak i od drugih računala u istoj lokalnoj računalnoj mreži (sjetimo se da standardni vatrozidi filtriraju promet između različitih računalnih mreža, što znači da je sav mrežni promet generiran u lokalnoj računalnoj mreži „nevidljiv“ za ove vatrozide).

Počevši od Windows XP SP2, Microsoft s Windows operacijskim sustavima automatski instalira i poslužiteljski vatrozid koji nakon inicijalne instalacije odmah blokira sav dolazni promet, dok je sav odlazni promet dopušten. S Windows 7 i Windows 2008 poslužiteljima ovaj je vatrozid dodatno proširen tako da je moguće filtrirati i odlazni mrežni promet, što prijašnje inačice vatrozida nisu mogle. Na sljedećoj je slici prikazan Windows vatrozid.



Slika 6.6. Vatrozid na Windows 7/2008 operacijskim sustavima

Na Linux operacijskim sustavima klasično se upotrebljava iptables poslužiteljski vatrozid. Riječ je o moćnom poslužiteljskom vatrozidu koji može obavljati čitav niz funkcija, osim primjenjivanja sigurnosne politike, poput usmjeravanja paketa i prepisivanja zaglavlja paketa prema definiranim politikama. Konfiguracija iptables paketa zadaje se u tekstualnom obliku, no postoji čitav niz grafičkih sučelja koja omogućuju jednostavniju konfiguraciju iptables paketa. Na sljedećoj je slici prikazan alat Firewall Builder, moćan alat za konfiguraciju vatrozida koji podržava iptables vatrozid, kao i čitav niz drugih vatrozida.

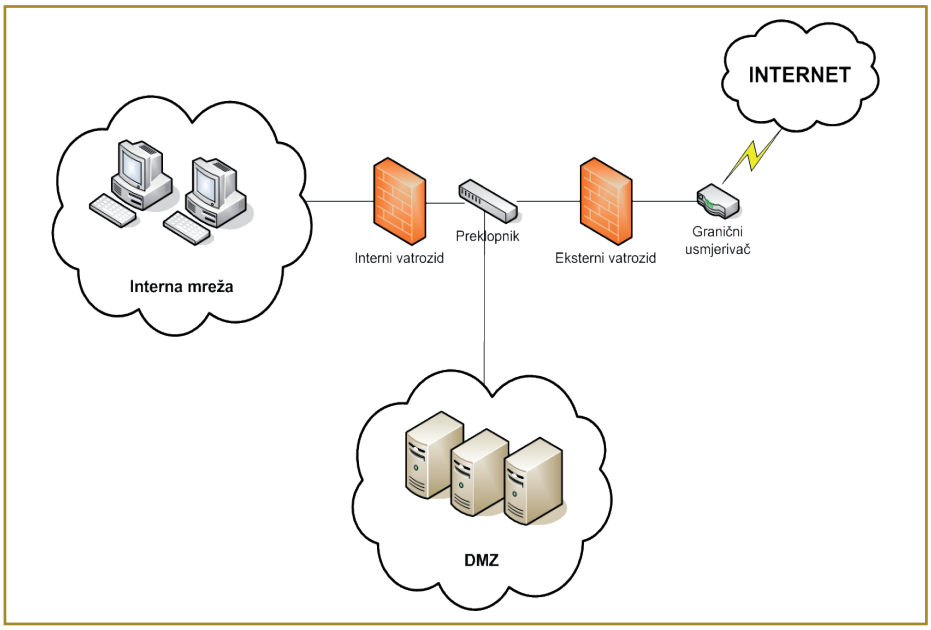


Slika 6.7. Firewall Builder alat za upravljanje konfiguracijama vatrozida

6.2.2.1 Implementacije vatrozida

Ovisno o broju vatrozida te njihovoj funkciji danas se općenito primjenjuju dvije metode implementacije vatrozida: implementacija s višestrukim vatrozidima te tzv. *three-leg* implementacija.

Implementacija s višestrukim vatrozidima, kao što i samo ime govori, zahtijeva višestruke hardverske ili softverske vatrozide. Shema ove implementacije prikazana je na sljedećoj slici.



Slika 6.8. Implementacija s višestrukim vatrozidima

Prednost ovakve implementacije je razdvajanje sigurnosnih zona pomoću višestrukih (obično jednog internog i jednog eksternog) vatrozida. Eksterni vatrozid u ovom slučaju primjenjuje sigurnosnu politiku na javni mrežni promet koji dolazi s interneta prema računalima u DMZ-u i internoj računalnoj mreži te odlazni promet prema internetu. DMZ (engl. *Demilitarized Zone*, demilitarizirana zona) naziv je za posebnu računalnu mrežu u koju se stavljaju javno dostupni poslužitelji, dakle poslužitelji dostupni s interneta. Budući da su javno dostupni, ovi poslužitelji predstavljaju i viši sigurnosni rizik pa ih se obično stavlja u zasebnu zonu na koju se mogu primijeniti posebna pravila, kao i dodatno nadgledanje sigurnosti i mrežnog prometa.

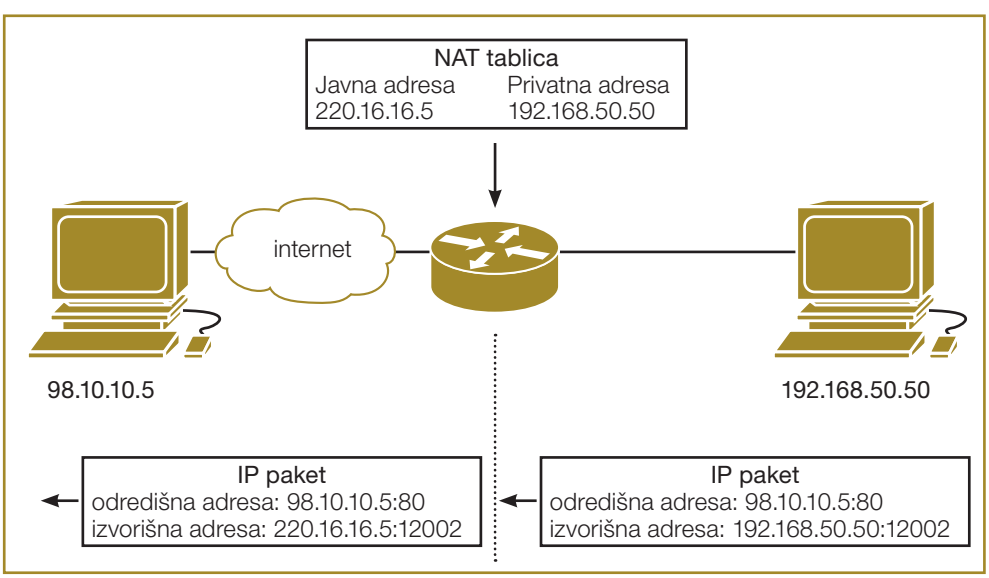
Prema potonjoj shemi interni vatrozid samo kontrolira promet prema internim računalima/poslužiteljima, i to promet koji dolazi s poslužitelja u DMZ-u. Kod ovakve se implementacije obično u potpunosti zabranjuje mrežni promet prema računalima na internoj računalnoj mreži koji dolazi s interneta. Na taj se način postiže najviša razina sigurnosti računala na internoj računalnoj mreži. Kao što se može vidjeti, napadač koji želi napasti računala ili poslužitelje na internoj računalnoj mreži u ovom slučaju treba prvo zaobići eksterni vatrozid, zatim kompromitirati poslužitelj u DMZ-u te potom preko njega zaobići interni vatrozid, što cijeli postupak čini teškim i kompleksnim za napadača te u skladu s time i smanjuje rezultirajući sigurnosni rizik.

Nedostatak ovakvog načina implementacije je prvenstveno u trošku implementacije budući da se zahtijeva instalacija dva vatrozidna proizvoda. U vrlo osjetljivim okruženjima često se zahtijeva da bude riječ o dva različita proizvoda, što dodatno podiže cijenu implementacije ne samo zbog investicije u proizvode već i zbog potrebe za održavanjem dva različita vatrozida.

Kao što je već spomenuto u poglavlju 6, interne računalne mreže gotovo se uvijek koriste privatnim adresnim prostorom, tzv. RFC 1918 adresnim prostorom koji uključuje sljedeće računalne mreže:

- 10.0.0.0/8,
- 172.16.0.0/12 i
- 192.168.0.0/16.

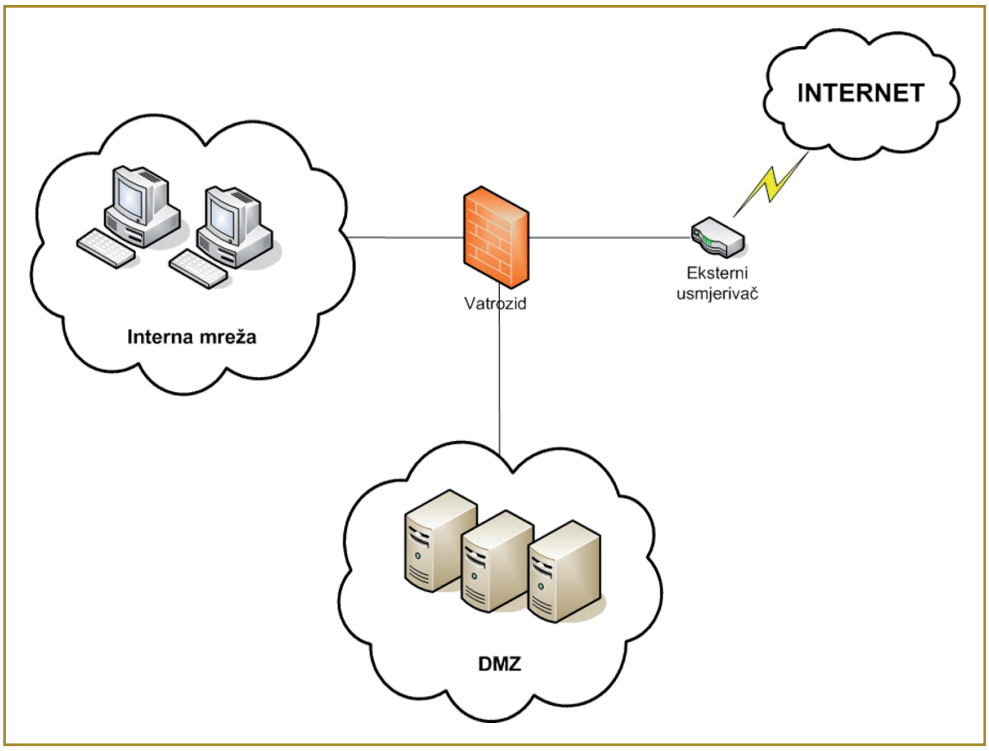
Budući da su navedene mreže nerutabilne na internetu, ako se želi ostvariti komunikacija s drugim poslužiteljima na internetu potrebno je uspostaviti NAT funkciju (engl. *Network Address Translation*). NAT funkcija prevodi interne IP adrese u one javno dostupne, na javnom sučelju vatrozida. Ovaj se postupak provodi u stvarnom vremenu, transparentno po korisnika koji niti ne zna da su njegovi paketi prevedeni. Funkcija prijevoda zapravo samo mijenja izvornu adresu IP paketa u onu na javnom sučelju vatrozida. To omogućuje primanje povratnog paketa ciljnog poslužitelja s interneta, nakon čega vatrozid opet prevodi IP adresu u internu i šalje mrežni paket natrag računalu koje je poslalo izvorni mrežni paket. NAT funkcija prikazana je na sljedećoj slici.



Slika 6.9. NAT funkcionalnost

Na potonjem primjeru implementacije NAT funkcionalnost može provoditi bilo koji od vatrozida. Klasične postavke NAT funkcionalnost stavljaju na eksterni vatrozid koji u potpunosti skriva sve interne i DMZ poslužitelje te dopušta promet iniciran s interneta samo prema onim poslužiteljima za koje su na eksternom vatrozidu eksplicitno definirana pravila prepisivanja i dozvole mrežnog prometa.

Drugi tip implementacije vatrozida koji se danas puno češće sreće, pogotovo kada je riječ o manjim tvrtkama, je tzv. *three leg* implementacija vatrozida. U ovom je slučaju, kao što se može vidjeti na sljedećoj slici, riječ o jednom vatrozidu koji ima tri mrežne kartice (koje mu omogućuju odvajanje tri računalne mreže).



Slika 6.10. Slika 71: Three leg implementacija vatrozida

Ovdje je potrebno napomenuti da vatrozid zapravo ne mora nužno imati tri fizičke mrežne kartice – kao što je rečeno u prethodnom poglavlju, moguće je implementirati i virtualni vatrozid koji sigurnosnu politiku primjenjuje prema VLAN-ovima.

Prikazana implementacija zasnovana na jednom vatrozidu ima nekoliko prednosti, ali i nedostataka. Ključna prednost je u smanjenju troška, budući da je

u ovom slučaju potrebno investirati samo u jedan vatrozid. Nedostatak ovakve implementacije je u činjenici da sama sigurnosna politika vatrozida, odnosno pravila koja definiraju koji će mrežni promet vatrozid propustiti a koji blokirati, postaje kompliciranija budući da sada jedna sigurnosna politika definira mrežni promet za i od sva tri segmenta računalnih mreža. Kompleksnija sigurnosna politika znači i da je rizik od ljudske pogreške automatski veći, što povećava i kompletan rezultirajući sigurnosni rizik s obzirom na to da pogreška kod ovakve implementacije može npr. dopustiti izravni promet između interneta i interne računalne mreže. Osim toga, može se vidjeti da u ovom slučaju vatrozid predstavlja kritičnu točku dostupnosti – u slučaju kvara na vatrozidu korisnicima na internoj računalnoj mreži bit će nedostupni i poslužitelji u DMZ-u i internet, dok kod prethodne implementacije u slučaju kvara jednog vatrozida dio servisa još uvijek funkcionira.

U navedenoj implementaciji vatrozid mora obavljati i NAT funkciju, što dodatno komplicira samu implementaciju.

6.2.3. Proxy poslužitelji/vatrozidi

Zbog potrebe za sve detaljnijom inspekcijom mrežnog prometa na višim razinama (npr. HTTP mrežni promet, koji danas predstavlja većinu mrežnog prometa gledajući prema protokolima koji se upotrebljavaju), danas se sve češće susrećemo sa specijaliziranim vatrozidima koji su zapravo kombinacije klasičnih vatrozida i *proxy* poslužitelja.

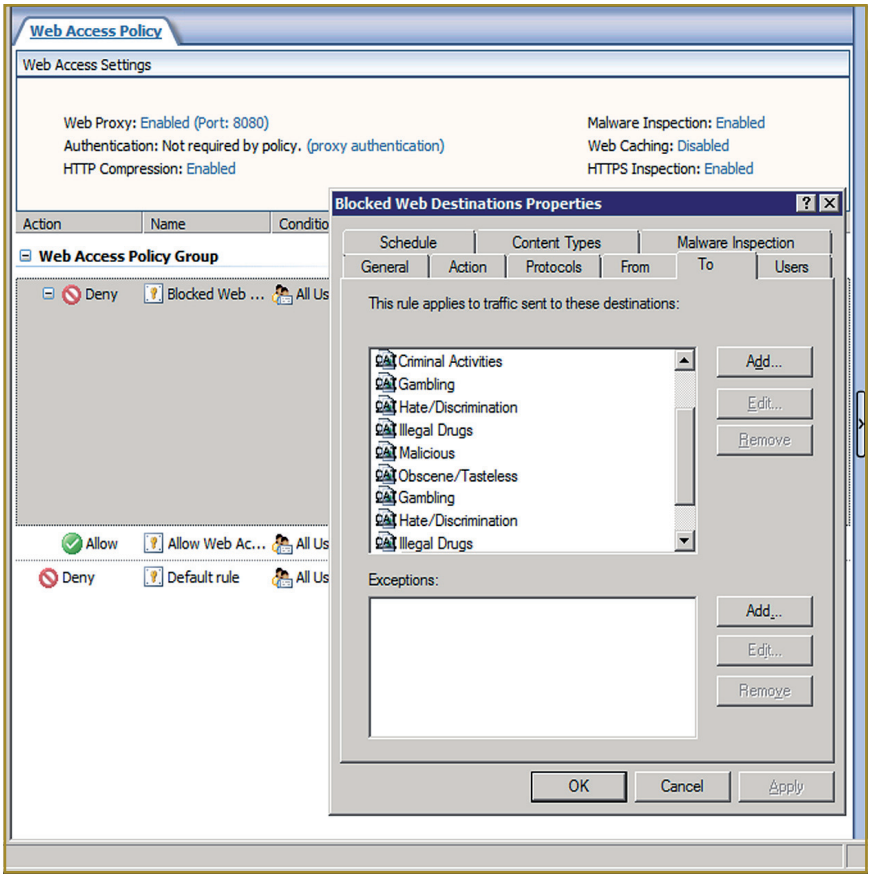
Glavna razlika između *proxy* poslužitelja i vatrozida je u činjenici da vatrozid transparentno provodi svoju sigurnosnu politiku (dakle, u načelu ne mijenja mrežne pakete koji prolaze kroz njega, osim ako nije riječ i o funkciji NAT-a, gdje vatrozid minimalno mijenja mrežni promet), dok *proxy* poslužitelj stvara dvije komunikacijske veze (sjednice). Pri upotrebi *proxy* poslužitelja jedna se sjednica stvara između klijentskog računala i *proxy* poslužitelja, a druga između *proxy* poslužitelja i ciljnog poslužitelja na internetu (ili nekoj drugoj računalnoj mreži).

Prednost *proxy* poslužitelja u odnosu na klasični vatrozid je jasna: zbog činjenice da se na njemu terminira mrežni promet, *proxy* poslužitelj može provoditi vrlo detaljnu analizu istoga, odnosno analizu aplikacijskog sloja te na taj način detektirati možebitne napade ili kršenja sigurnosne politike na višim razinama. *Proxy* poslužitelji vrlo se često upotrebljavaju upravo za analizu HTTP prometa, gdje se administratorima omogućuje upotreba naprednih mogućnosti poput URL filtriranja.

Microsoft TMG (bivši Microsoft ISA) vatrozid najčešće je korišteni vatrozid koji kombinira mogućnosti klasičnog vatrozida i *proxy* poslužitelja. Microsoft TMG

tako može primjenjivati klasične sigurnosne politike gdje se kontrolira mrežni promet prema sučeljima vatrozida, ali se isto tako može upotrebljavati i kao *proxy* poslužitelj gdje omogućuje nadgledanje HTTP, ali HTTPS (enkriptiranog) mrežnog prometa. U slučaju nadgledanja HTTP mrežnog prometa Microsoft TMG nudi čitav niz naprednih mogućnosti:

- URL filtriranje prema dopuštenim i nedopuštenim kategorijama – pokušava se povećati produktivnost zaposlenika. Microsoft TMG može spriječiti pokušaj pristupa web-stranicama u određenim kategorijama (npr. *online* klađenje i slično). Prozor koji omogućuje definiranje kategorija web-stranice koje se žele blokirati prikazan je na sljedećoj slici;



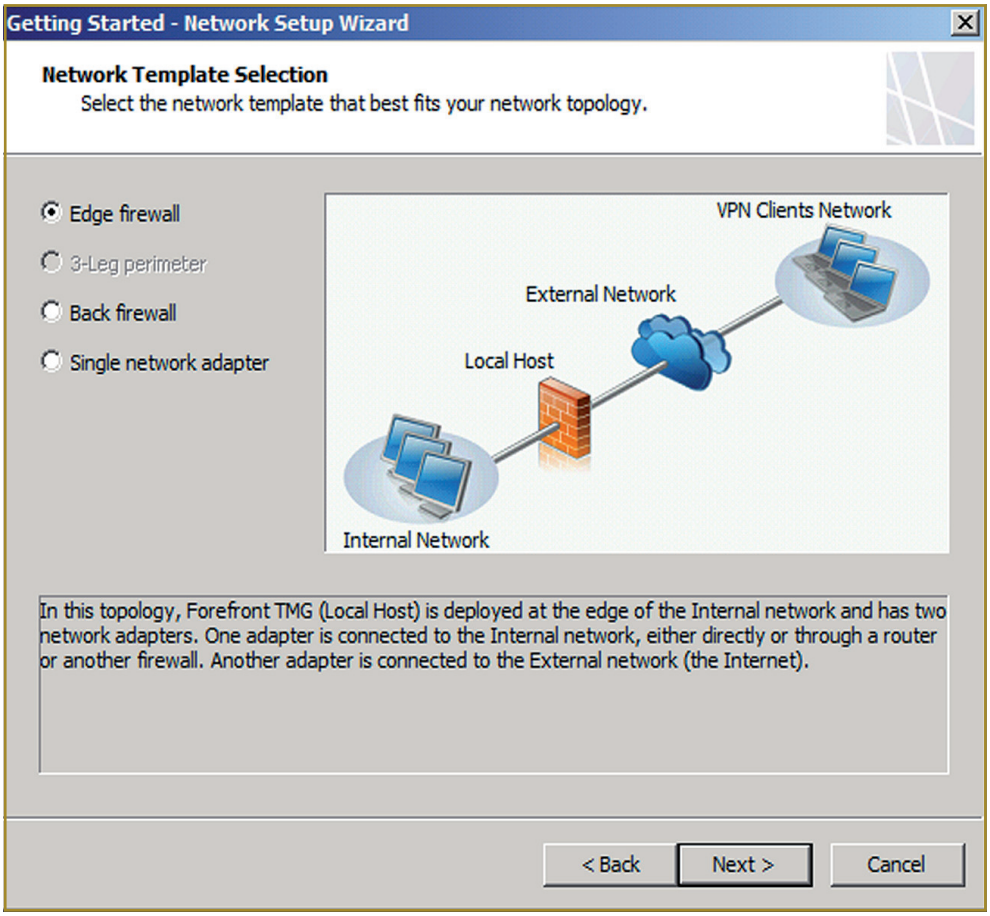
Slika 6.11. Microsoft TMG mogućnost blokiranja pristupa web-stranicama prema kategoriji web-stranica

- pregledavanje mrežnog prometa u svrhu detektiranja pokušaja iskorištavanja sigurnosnih ranjivosti – ovu funkciju Microsoft naziva NIS (engl.

Network Inspection System) i omogućuje detektiranje poznatih napada, funkcionalnost sličnu sustavima za detekciju i prevenciju neovlaštenih aktivnosti, koji su detaljnije objašnjeni u nastavku;

- pregledavanje mrežnog prometa u svrhu detekcije malicioznih programa poput virusa, trojanskih konja i crva – Microsoft TMG može se integrirati s poznatim antivirusnim programima kako bi se omogućila inspekcija mrežnog prometa i blokiranje malicioznih programa u slučaju njihove detekcije.

Zbog svog specifičnog načina rada, odnosno kombinacije klasičnog vatrozida i *proxy* poslužitelja, Microsoft TMG omogućuje implementaciju čak i samo s jednom mrežnom karticom, kao što se može vidjeti na sljedećem ekranu koji pokazuje mogućnosti instalacije Microsoft TMG vatrozida.



Slika 6.12. Mogućnosti implementacije Microsoft TMG vatrozida

U slučaju da poslužitelj na kojem je instaliran Microsoft TMG vatrozid ima samo jednu mrežnu karticu, programski je paket moguće instalirati isključivo u *proxy* načinu rada. U tom slučaju sva klijentska računala moraju imati definiran ovaj poslužitelj kao *proxy* poslužitelj u web-preglednicima kako bi se osiguralo da sav HTTP i HTTPS mrežni promet prolazi kroz *proxy* poslužitelj. Ovo je moguće jednostavno postići pomoću domenskih (grupnih) politika koje su bile detaljnije obrađene u poglavlju 5.2.

6.3. Sustavi za detekciju i prevenciju neovlaštenih aktivnosti

Sustavi za detekciju i prevenciju neovlaštenih aktivnosti (engl. *Intrusion Detection / Prevention Systems* – IDS/IPS) su specijalizirani sustavi čija je svrha, kao što im i ime govori, sprječavanje bilo kakvih neovlaštenih aktivnosti.

Ovdje je najčešće riječ opet o specijaliziranim hardverskim uređajima koji pregledavaju i analiziraju mrežni promet u svrhu detekcije napada. IDS sustavi namijenjeni su samo detekciji potencijalnih neovlaštenih aktivnosti nakon čega mogu uzbuniti administratora, dok IPS sustavi predstavljaju proširenje IDS sustava te nakon detekcije neovlaštene aktivnosti istu mogu i spriječiti.

Današnji najpoznatiji IDS i IPS sustavi uključuju one od proizvođača Sourcefire, McAfee i Tipping Point. Riječ je o komercijalnim proizvodima koji mogu predstavljati vrlo visoke investicije, pogotovo kada se uzmu u obzir računalne mreže na koje se namjeravaju instalirati. Naime, jedan od osnovnih parametara IDS i IPS uređaja jest njihova propusnost koja mora odgovarati računalnoj mreži na koju su postavljeni. Na primjer, ako je veza na internetske tvrtke 100 mbit/s, onda je potrebno implementirati i IDS/IPS uređaj koji podržava 100 mbit/s, u protivnom je moguće da će doći do gubljenja paketa što može uzrokovati probleme u detekciji.

Što se tiče besplatnih IDS sustava, najpoznatiji mrežni IDS sustav na kojem je temeljena i većina komercijalnih jest Snort, paket dostupan na adresi http://www.snort.org.

IDS i IPS sustavi dijele se prema tehnologiji nadzora na sljedeće dvije grupe, slično vatrozidima:

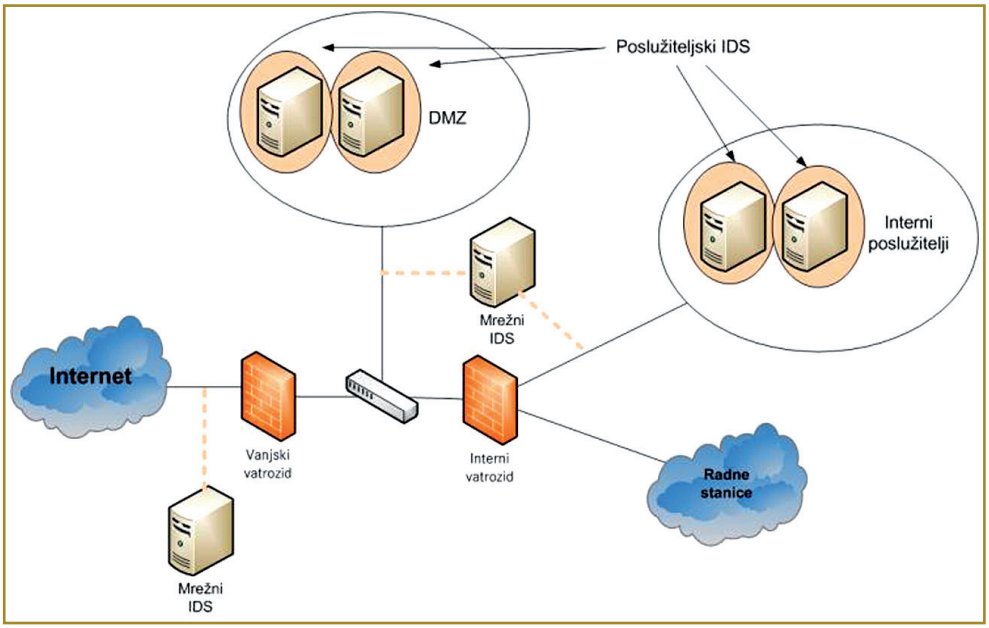
- mrežni IDS i IPS sustavi – pregledavaju izravno mrežni promet koristeći se nekim od tehnologija spomenutih u poglavlju 6.1. U slučaju IDS sustava riječ je uglavnom o pasivnom pregledavanju mrežnog prometa tako da se oni najčešće realiziraju u vidu hardverskih uređaja koji imaju posebna

sučelja koja se spajaju na zrcalna sučelja na preklopnicima kako bi bili u stanju vidjeti mrežni promet koji trebaju analizirati.

Budući da su IPS sustavi aktivni u smislu da moraju blokirati detektirane neovlaštene aktivnosti, njihova je implementacija na računalnim mrežama malo drukčija. Ovi se sustavi tako najčešće postavljaju na granična mjesta (perimeter) kako bi nadgledali sav odlazni ili dolazni mrežni promet, slično vatrozidima. Razlika je, kao što je već rečeno, u činjenici da IPS sustavi detaljno analiziraju sav mrežni promet te u slučaju detekcije malicioznog mrežnog prometa isti automatski blokiraju;

- poslužiteljski IDS i IPS sustavi – slično poslužiteljskim vatrozidima, instaliraju se u vidu softverskih aplikacija izravno na poslužitelje. Ovakvi IDS i IPS sustavi pregledavaju mrežni promet koji je usmjeren prema poslužitelju na kojem su instalirani i koji je generiran od strane tog poslužitelja. Prednost ovakvog pristupa je da poslužiteljski IDS i IPS sustavi izravno mogu nadgledati promet usmjeren poslužitelju te ga pravodobno blokirati. Nedostatak je potreba za instaliranjem na svaki poslužitelj što, osim dodatne potrošnje resursa poslužitelja, povećava i zahtjeve za upravljanjem IDS i IPS sustavima.

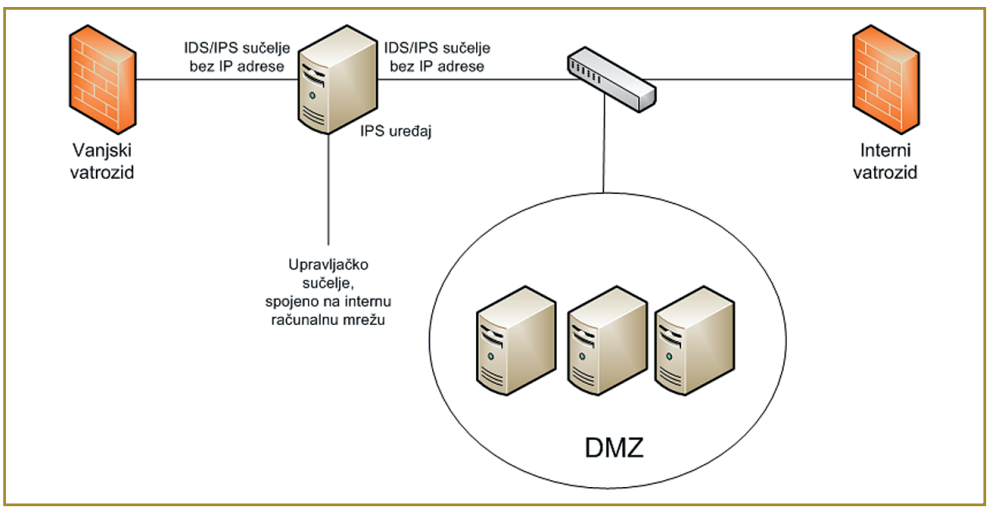
Na sljedećoj su slici prikazana najčešća mjesta implementacije mrežnih i poslužiteljskih IDS i IPS sustava.



Slika 6.13. Implementacija IDS sustava

Osim samog pregledavanja mrežnog prometa, analize i detektiranja potencijalnih neovlaštenih aktivnosti, od IDS i IPS sustava obično se očekuje i generiranje dnevničkih zapisa koji upućuju na detektirane događaje kao i zapisivanje svih dodatnih informacija koje se mogu iskoristiti tijekom kasnije analize incidenta. Naime, budući da mrežni IDS i IPS sustavi temelje svoju analizu na pregledavanju mrežnog prometa, jedan od zahtjeva koji se postavljaju pred ove uređaje je i zapisivanje izvornog mrežnog prometa kako bi administrator poslije i sam mogao pregledati pakete koji su uzrokovali detekciju. Navedeni snimljeni mrežni promet obično se pohranjuje u tzv. PCAP formatu koji omogućuje izravno učitavanje u programe za analizu mrežnog prometa poput već spomenutih Wiresharka i Tcpdumpa (poglavlja 6.1.1.1 i 6.1.1.2).

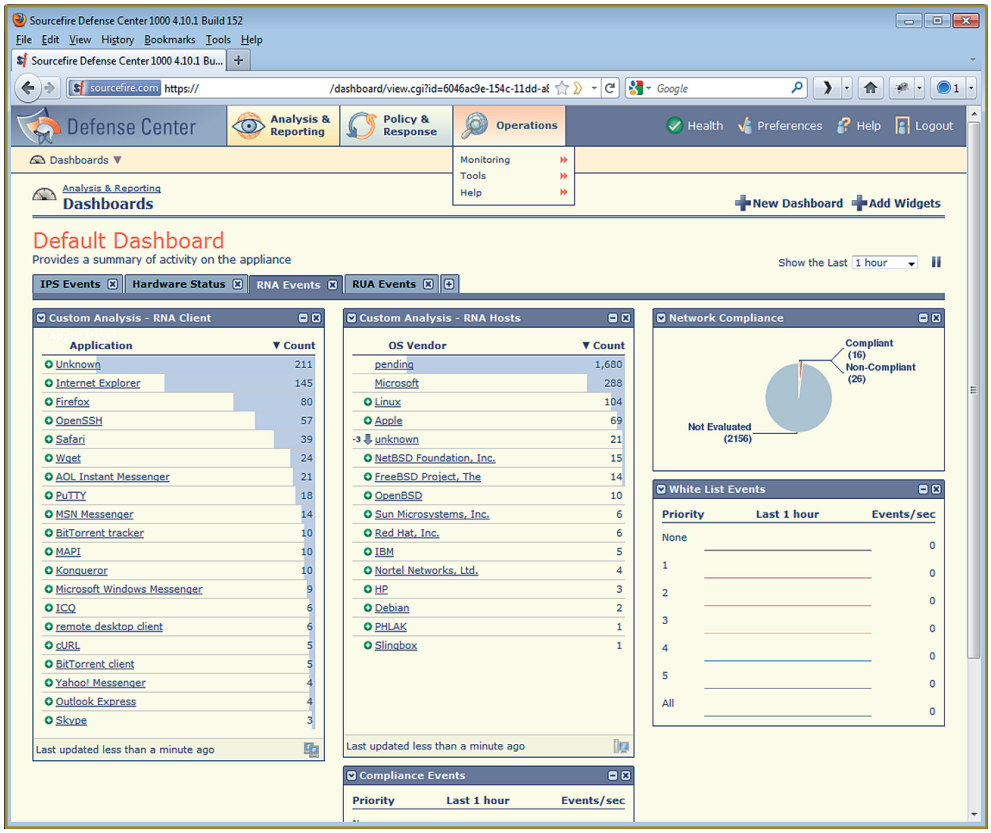
U slučaju IPS sustava očito je, međutim, da je potrebno ne samo pregledavanje mrežnog prometa nego i blokiranje u slučaju da je detektiran napad. U većini slučajeva ovo se postiže tako da se IPS sustav postavlja izravno na komunikacijsku vezu (engl. *Inline implementation*), kao i vatrozid. U ovom slučaju IPS uređaj ima dvije mrežne kartice – jednu ulaznu i jednu izlaznu te se obično postavlja iza vatrozida. Ovakav način implementacije omogućuje blokiranje mrežnog prometa u kojem su detektirane maliciozne aktivnosti: budući da svi mrežni paketi koji dolaze prema štićenoj računalnoj mreži ili odlaze s nje moraju proći kroz IPS sustav, moguće je svaki mrežni paket pregledati te napad, u slučaju da je detektiran, jednostavno blokirati (IPS sustav takav maliciozni mrežni paket ne propušta na izlazno sučelje). Ovakav način postavljanja IPS sustava prikazan je na sljedećoj slici.



Slika 6.14. Implementacija **inline** IPS sustava

U slučaju detekcije neovlaštenih aktivnosti IPS sustav može, osim blokiranja paketa, pokušati i aktivno prekinuti komunikacijsku vezu (npr. u slučaju TCP mrežnog prometa moguće je poslati pakete za prekid, RST pakete na obje strane komunikacije). Ovakav se pristup, međutim, izbjegava budući da napadaču može pažnju usmjeriti na postojanje IPS sustava.

Upravljanje IDS i IPS sustava obično se provodi putem centralizirane konzole. Naime, u slučaju većeg broja instaliranih agenata ili senzora, upravljanje svakog uređaja zasebno može biti vrlo kompleksno tako da se danas u tu svrhu obično rabe specijalizirane konzole. Konzola jednog takvog IPS sustava proizvođača Sourcefire koji svoje uređaje zasniva na besplatnom Snort IDS sustavu otvorenog koda prikazana je na sljedećoj slici.



Slika 6.15. Konzola Sourcefire IDS/IPS sustava

Osim centralnog upravljanja IDS i IPS senzorima, ovakve konzole obično omogućuju i pregled svih detektiranih neovlaštenih aktivnosti na jednom mjestu te generiranje izvještaja.

6.3.1. Mehanizmi detekcije IDS i IPS sustava

Današnji IDS i IPS sustavi detekciju temelje na mehanizmima vrlo sličnim onima kojima se koriste antivirusni programi opisani u poglavlju 4.4. Većina IDS i IPS sustava tako detekciju zasniva na sljedeća tri mehanizma:

- detekcija potpisima (engl. *Signature Detection*) – funkcionira vrlo slično istoimenoj detekciji kod antivirusnih programa. I ovdje je riječ o posebno napisanim potpisima koji definiraju izgled i sadržaj mrežnih paketa. Pri pregledavanju mrežnog prometa IDS/IPS uređaji uspoređuju primljeni mrežni promet s potpisima iz baze potpisa te se, ako je detektiran paket s jednakim značajkama, diže uzbuna.

Prednost detekcije potpisima je, jednako kao i kod antivirusnih programa, njihova brzina. Zbog toga nerijetko IDS/IPS sustavi imaju i po nekoliko desetaka tisuća potpisa koji definiraju razne mrežne napade i tipove malicioznog mrežnog prometa. Problem ovakve detekcije je ovisnost o potpisima – ako potpis za pojedini maliciozni paket ne postoji ili je riječ o novom napadu, IDS/IPS sustav neće biti u stanju detektirati ga. Napadači vrlo često provode različite metode mijenjanja mrežnog prometa ne bi li zaobišli detekciju od strane IDS/IPS sustava;

- detekcija anomalija (engl. *Anomaly Detection*) u računalnoj mreži zasnovana je na profilima. Pri upotrebi ovog tipa detekcije, IDS/IPS sustav prvo se postavlja neko vrijeme na računalnu mrežu bez ikakvih pravila. Tijekom ovog razdoblja stvara se profil mrežnog prometa: gledaju se značajke mrežnog prometa, statistika upotrebe pojedinih protokola i slično.

Kada je profil mrežnog prometa napravljen, sustav dalje funkcionira tako da, uz već spomenute potpise, pokušava detektirati i anomalije u korištenim protokolima i slično. Na ovaj je način moguće detektirati napade koji su bili modificirani ili u potpunosti nove napade;

- detaljna analiza mrežnih protokola (engl. *Stateful analysis*) – postupak je sličan onome već spomenutom kod vatrozida. U ovom slučaju IDS/IPS poslužitelji detaljno analiziraju mrežne protokole koji se rabe u pregledavanom mrežnom prometu te su u stanju detektirati bilo kakav izlazak iz očekivanih okvira protokola. Primjerice, ako određeno polje definirano protokolom prima ulazni niz od 10 znakova, a IDS/IPS sustav u istom polju detektira niz od 200 znakova, moguće je zaključiti da je riječ o pokušaju provođenja napada prepisivanjem podataka u spremniku, objašnjenih u poglavlju 4.1.1 te automatski blokirati ovaj mrežni promet ili uzbuniti administratora.

Kao što je bio slučaj s antivirusnim programima, tako niti IDS i IPS sustavi nisu savršeni. Razlikujemo dva slučaja kada dolazi do grešaka u detekciji ovih sustava:

- lažno pozitivne detekcije (engl. *False positive detections*) – događaju se kada IDS/IPS sustavi detektiraju napade u mrežnom prometu u kojem napada nije bilo, npr. zbog loše napisanog potpisa ili greške u detekciji anomalijama. Iako je ovakva greška nepoželjna, administrator ipak može uvijek modificirati rad IDS/IPS sustava te čak, ako je to potrebno, isključiti pravilo ako ono stvara probleme u radu;
- lažno negativne detekcije (engl. *False negative detections*) – događaju se kada IDS/IPS sustav ne detektira napad u mrežnom prometu u kojem je napad (odnosno iskorištavanje sigurnosne ranjivosti) zbilja postojao. Ovakvi su tipovi grešaka puno opasniji budući da omogućuju provođenje malicioznih aktivnosti bez detekcije IDS/IPS sustava. Jednako kao i u prethodnom slučaju, administrator može modificirati pravila ili profil mrežnog prometa kako bi poboljšao detekciju sustava.

Osim mogućnosti grešaka u detekciji, jedan od ključnih nedostataka IDS i IPS sustava je definitivno i nemogućnost pregledavanja enkriptiranog mrežnog prometa. Naime, veza se pri uspostavi enkriptirane komunikacijske veze uspostavlja između dva krajnja entiteta (npr. u slučaju HTTPS veze između klijentskog računala i poslužitelja). Iako IDS/IPS sustav može vidjeti sav mrežni promet, sve informacije koje klijentsko i poslužiteljsko računalo razmjenjuju preko enkriptirane komunikacijske veze bit će skrivene od IDS/IPS sustava koji neće biti u mogućnosti detektirati maliciozne aktivnosti. Današnji moderni IDS/IPS sustavi omogućuju dekripciju sličnu onoj *proxy* poslužitelja, koja je objašnjena u poglavlju 6.2.3, no potrebno je napomenuti da to zahtijeva provođenje određenih predkoraka na klijentskim računalima kako bi se omogućilo umetanje lažnih certifikata i pregledavanje enkriptiranog mrežnog prometa. Više o SSL-u i certifikatima objašnjeno je u poglavlju 6.6.

6.4. Kontrola pristupa računalnoj mreži

U velikim tvrtkama, koje nerijetko mogu imati stotine, pa i tisuće klijentskih računala, održavanje ovih računala sigurnima predstavlja ozbiljan zadatak. S druge strane, u tako velikim organizacijama dosta je čest slučaj da zaposlenici imaju prijenosna računala koja mogu biti izvan tvrtkine organizacije i dulja razdoblja. Jednako tako, vrlo često dolaze vanjski suradnici koji mogu unutar tvrtkinih

fizičkih prostorija donositi svoju opremu te ju u konačnici spajati izravno na internu računalnu mrežu.

Kao što je već spomenuto u poglavlju 5.2, ojačanje sigurnosti operacijskih sustava postupak je koji treba provesti za sve nove poslužitelje. No neke je od koraka kao što su redovito instaliranje sigurnosnih zakrpi na računala te osvježavanje definicija antivirusnih programa potrebno redovito provoditi, a ne samo pri inicijalnoj instalaciji.

Zbog stalnog pojavljivanja novih malicioznih programa, računala bez odgovarajuće antivirusne zaštite ili bez instaliranih sigurnosnih zakrpi predstavljaju naročit sigurnosni rizik za cijelu organizaciju. Naime, infekcija jednog klijentskog računala može napadaču omogućiti kontrolu putem interneta te pokretanje daljnjih napada upravo s tog inficiranog računala.

Upravo zbog činjenice da takva računala predstavljaju povećan sigurnosni rizik za organizaciju pojavila se potreba kontrole računala kojima se dopušta spajanje na računalnu mrežu organizacije. Ovakva kontrola mrežnog pristupa (engl. *Network Access Control* – NAC, ili ponekad engl. *Network Access Protection* – NAP) danas se postiže pomoću različitih metoda i uređaja.

Cilj kontrole pristupa računalnoj mreži jest postizanje određenih minimalnih zahtjeva na sigurnost računala kojima se dopušta spajanje. Drugim riječima, NAC odnosno NAP proizvodi prije nego što dopuste pristup računalnoj mreži klijentskim računalima provjeravaju jesu li zadovoljeni minimalni zahtjevi za sigurnost koji najčešće uključuju sljedeće:

- provjeru instaliranih sigurnosnih zakrpi – za računala koja se žele priključiti na računalnu mrežu provjerava se imaju li instalirane sve tražene sigurnosne zakrpe;
- provjeru stanja antivirusnog programa – stanje antivirusnog programa provjerava se da bi se potvrdilo jesu li instalirane zadnje dostupne definicije za antivirusni program te je li sam antivirusni program pokrenut i funkcionira li ispravno;
- provjeru poslužiteljskog vatrozida i drugih sigurnosnih zahtjeva, ovisno o pojedinoj implementaciji NAC/NAP rješenja.

Različita NAC/NAP rješenja služe se različitim metodama ustanovljavanja zadovoljava li klijentsko računalo postavljene sigurnosne zahtjeve. Najsigurnija metoda provjere temelji se na agentu koji je u tom slučaju potrebno instalirati na svako klijentsko računalo. Agent zatim može provjeriti stanje računala (instalirane zakrpe, antivirusne definicije, stanje vatrozida i slično) te rezultate poslati na središnji NAC/NAP poslužitelj koji može zatim odlučiti zadovoljava li klijentsko računalo sigurnosne zahtjeve. Problem ovog pristupa je u instalaciji agenta, što

može biti teško ili čak neprihvatljivo za klijentska računala koja nisu u vlasništvu tvrtke, poput npr. klijentskih računala koja donose suradnici.

Ako se ne upotrebljava agent na klijentskom računalu, moguće je provesti čak i provjeru sigurnosti računala preko računalne mreže, no potrebno je naglasiti da ovakva rješenja mogu vrlo lako dovesti do lažno pozitivnih ili lažno negativnih rezultata. Glavni razlog za to je činjenica da je bez postojanja lokalnog agenta putem računalne mreže vrlo teško donijeti zaključak o sigurnosnom stanju klijentskog računala.

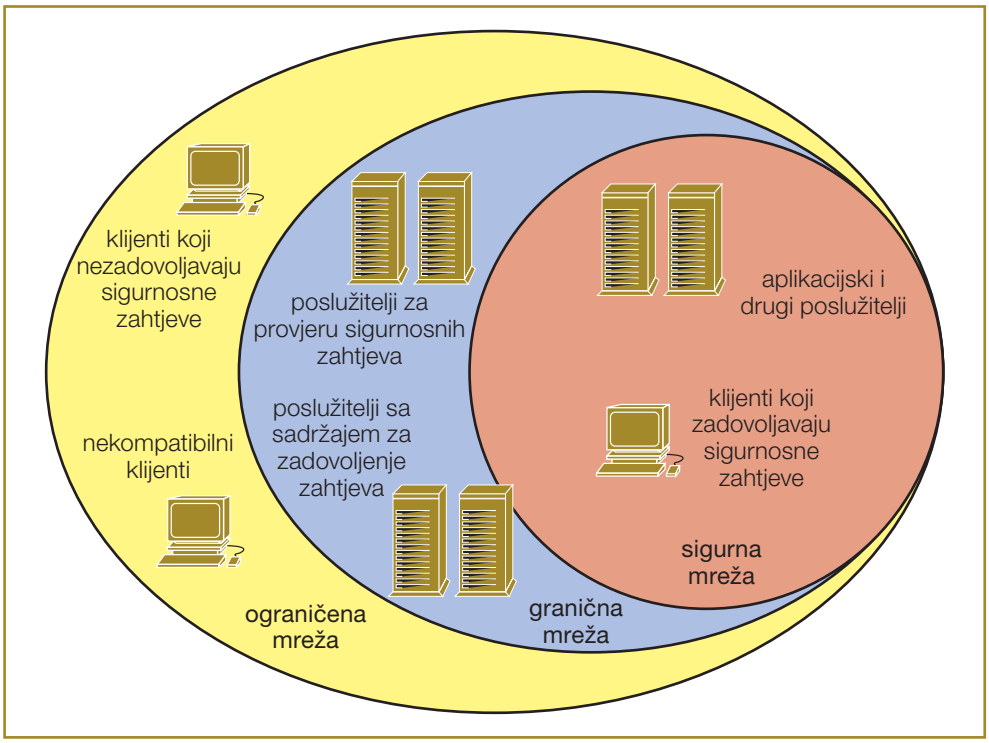
U slučaju da je rezultat sigurnosne provjere bio negativan, odnosno da je otkriveno da klijentsko računalo koje se želi spojiti na računalnu mrežu ne zadovoljava sigurnosne zahtjeve, općenito se upotrebljavaju dva pristupa:

- postavljanje računala u karantenu – karantena je izolirani segment računalne mreže kojem je zabranjena mrežna komunikacija s drugim, internim segmentima računalne mreže organizacije te mu se klasično samo dopušta pristup na internet.

  Karantena se obično implementira u vidu posebnog VLAN-a, a NAC/NAP sustav ima mogućnost kontrole mrežnog sustava u vidu u kojem se VLAN-u nalazi pojedino mrežno sučelje. Tako se sva sučelja na kojima su računala koja nisu zadovoljila sigurnosne zahtjeve ili prazna sučelja stavljaju u VLAN karantene. Tek nakon što je ustanovljeno da klijentsko računalo zadovoljava sigurnosne zahtjeve, mrežno sučelje preklopnika na koje je to računalo spojeno spaja se u drugi VLAN te mu se dopušta normalna komunikacija;
- zatočenički portali (engl. *Captive portal*) – u ovom je slučaju i dalje riječ o izoliranim mrežnim segmentima, ali se dodatno presreće (obično HTTP) mrežni promet te usmjerava na tzv. zatočenički portal. Cilj ovog portala je omogućiti korisniku klijentskog računala skidanje svih potrebnih aplikacija za zadovoljavanje sigurnosnih zahtjeva, kao što su sigurnosne zakrpe, antivirusne definicije i slično. Sav ostali mrežni promet je onemogućen tako da korisnik ne može pristupiti ničemu osim sigurnosnim programima potrebnim za zadovoljavanje sigurnosnih zahtjeva.

  Zatočenički portali danas su vrlo popularni kod ISP-eva, pogotovo u slučajevima kada je detektirano inficirano računalo korisnika. U tom se slučaju ne želi u potpunosti onemogućiti rad računala već se ono šalje na poseban portal na kojem se obično nude posebni programi za uklanjanje malicioznih programa i slično.

Na sljedećoj je slici prikazana shema konfiguracije računalne mreže koja se obično preporučuje za implementaciju s NAC/NAP rješenjima.



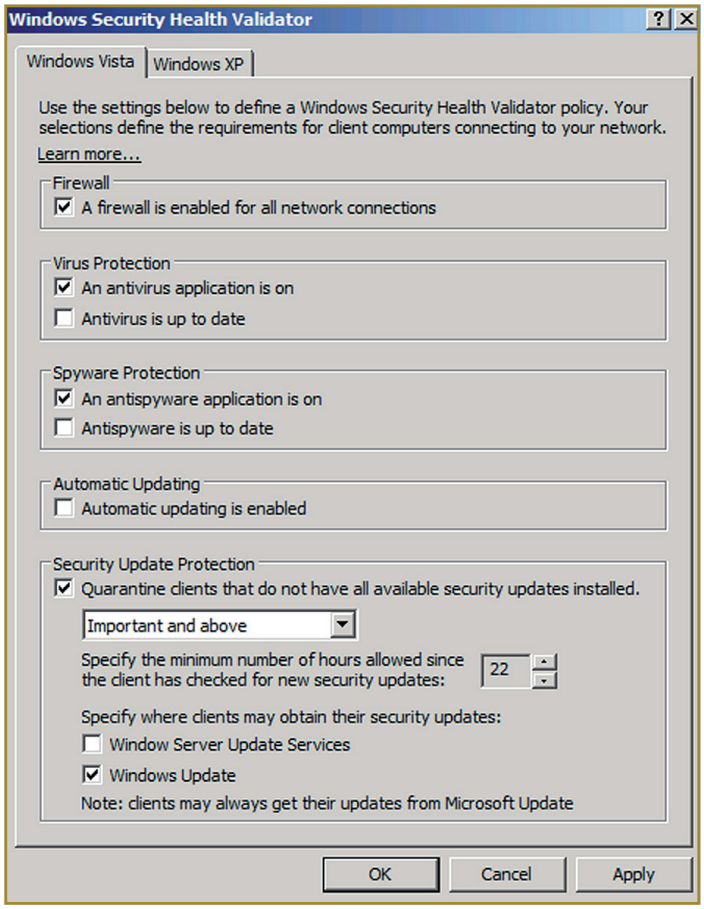
Slika 6.16. Primjer NAC/NAP implementacije

Na sigurnoj se računalnoj mreži nalaze klijentska računala koja su zadovoljila sigurnosne zahtjeve. Granična mreža sadržava poslužitelje koji služe za zadovoljavanje sigurnosnih zahtjeva, odnosno zatočeničke portale i slično koji omogućuju skidanje sigurnosnih zakrpi, antivirusnih definicija i drugih programskih paketa. Konačno, na ograničenoj računalnoj mreži nalaze se sva klijentska računala koja nisu zadovoljila sigurnosne zahtjeve.

Od dostupnih NAC/NAP proizvoda danas se najčešće upotrebljava Cisco Network Admission Control uređaj. Riječ je o uređaju koji omogućuje uspostavljanje karantena i kontrolu krajnjih, klijentskih računala, no za uspostavu nekih naprednih funkcija zahtijeva u potpunosti Ciscovu mrežnu opremu.

Microsoft je također ponudio svoje NAP rješenje kao dio Windows Server 2008 operacijskog sustava. Microsoftovo NAP rješenje može se koristiti različitim metodama ograničavanja pristupa od kojih neke ne zahtijevaju nikakvu dodatnu ili posebnu mrežnu opremu, no svaka od podržanih metoda ograničavanja (ograničavanje putem DHCP, IPSec, VPN ili 802.1X metoda) ima svoje prednosti i mane. Što se tiče klijentskih računala, Microsoft podržava NAP od Windows Viste, dok je za starije operacijske sustave poput Windows XP potrebno insta-

lirati dodatni paket. Sljedeća slika prikazuje mogućnosti konfiguriranja sigurnosnih zahtjeva koji se postavljaju pred klijentska računala koja se žele spojiti na računalnu mrežu.



Slika 6.17. Windows Security Health Validator omogućuje specificiranje sigurnosnih zahtjeva koji se postavljaju pred klijentska računala u NAC/NAP okruženju

6.5. Sigurni mrežni protokoli

Kao što je već rečeno u poglavlju 6.1, jedan od glavnih napada koji neovlašteni korisnici pokušavaju provesti na lokalnim računalnim mrežama jest pregledavanje mrežnog prometa. Cilj provođenja ovakvih i sličnih napada je jasan – ako se podaci računalnom mrežnom prenose u čistom tekstualnom obliku, napadač niti ne treba kompromitirati ciljno računalo već može jednostavnim pregledavanjem mrežnog prometa doći do cilja.

Još jedan od velikih sigurnosnih rizika predstavlja slanje autentikacijskih parametara računalnom mrežom. Naime, kada se administrator treba prijaviti na udaljeni poslužitelj ili mrežni uređaj, autentikacija administratora odvija se putem računalne mreže. Ovdje se može vidjeti da, ako je proces autentikacije napravljen na nesiguran način, napadač koji ima mogućnost pregledavanja mrežnog prometa može na vrlo jednostavan način doći do administratorskog korisničkog računa (zaporke) te se nakon toga na udaljeni sustav i sam prijaviti kao administrator.

Povijesno, veliki broj protokola provodio je autentikaciju korisnika i prijenos podataka u čistom tekstualnom obliku. Danas je očito da je ovakav način prijenosa podataka računalnom mrežom neprikladan te su zbog toga tijekom vremena za praktički sve nesigurne protokole razvijene alternative koje se služe naprednim autentikacijskim mehanizmima te enkripcijom za prijenos osjetljivih podataka. Ako se enkripcija ispravno upotrebljava, činjenica da napadač može doći do mrežnog prometa ne povećava rizik budući da je isti za njega nedostupan.

U sljedećoj je tablici dana lista protokola koji prenose podatke u čistom tekstualnom obliku, njihovih sigurnih alternativa te komentara vezanih za pojedinu implementaciju. Pri upotrebi bilo kojeg od navedenih protokola preporučuje se uvijek upotreba sigurne alternative protokola budući da pruža jednaku funkcionalnost (u nekim slučajevima i puno više).

Izvorni protokol	Sigurni protokol	Komentar
telnet	SSH	Telnet protokol ne bi se trebao više nigdje upotrebljavati budući da sve podatke šalje u čistom tekstualnom obliku. Alternativu telnet protokolu predstavlja SSH (engl. *Secure Shell*) koji omogućuje punu funkcionalnost telnet protokola i k tome osigurava da su svi podaci koji se prenose preko računalne mreže enkriptirani. SSH protokol također omogućuje i identifikaciju udaljenih poslužitelja prema ponuđenom javnom ključu (više o javnim ključevima u poglavlju 3.3.2) te na taj način omogućuje detekciju *Man-in-the-Middle* napada gdje se napadač predstavlja kao ciljni poslužitelj.
rlogin	SSH	Rlogin predstavlja stari protokol koji je SSH zamijenio u potpunosti. Rlogin protokol potrebno je u potpunosti onemogućiti.
rsh	SSH	Rsh predstavlja stari protokol koji je SSH potpuno zamijenio. Rsh protokol potrebno je u potpunosti onemogućiti.

Izvorni protokol	Sigurni protokol	Komentar
FTP	SFTP/SCP	FTP protokol kompleksan je protokol koji zahtijeva otvaranje dvije komunikacijske veze (sjednice), jedne kontrolne i jedne podatkovne. Zbog toga je povijesno stvarao velike probleme pri konfiguraciji vatrozida. Dodatno, FTP protokol sve podatke, uključujući i one autentikacijske, računalnom mrežnom prenosi u čistom tekstualnom obliku. Upotreba FTP protokola preporučuje se samo u slučaju javnih FTP poslužitelja na kojima se ne nalaze osjetljivi (privatni) podaci. U protivnom, preporučuje se upotreba SFTP/SCP protokola koji su dio SSH-a. SFTP omogućuje snažnu autentikaciju te automatsko korištenje enkripcijom pri prijenosu podataka. Konačno, SFTP rabi samo jedan mrežni port (port 22, jednako kao i SSH) te ga je zbog toga lako propuštati na vatrozidima.
HTTP	HTTPS	HTTP protokol sve podatke šalje u čistom tekstualnom obliku. U slučaju da je potrebno osigurati povjerljivost prenošenih podataka (npr. autentikacijski podaci ili drugi osjetljivi podaci), preporučuje se korištenje HTTPS protokolom koji omogućuje provjeru identiteta poslužitelja i klijenta te enkripciju podataka koji se prenose računalnom mrežom. Više o HTTPS protokolu, odnosno SSL-u koji se rabi za zaštitu podataka dano je u poglavlju 6.6.
POP3	POP3S	POP3 protokol koji se često upotrebljava za čitanje poruka elektroničke pošte preporučuje se zamijeniti POP3S protokolom. Riječ je o identičnom protokolu koji rabi SSL za zaštitu podataka.
IMAP4	IMAP4S	Kao i u prethodnom slučaju, umjesto IMAP4 protokola za čitanje elektroničke pošte preporučuje se upotreba IMAP4S protokola koji štiti podatke pomoću SSL-a.
LDAP	LDAPS	LDAP protokol upotrebljava se za pristup imeničkim servisima te sve podatke računalnom mrežom šalje u čistom tekstualnom obliku. U svrhu zaštite podataka preporučuje se korištenje LDAPS protokolom koji štiti podatke pomoću SSL-a.

6.6. SSL/TLS protokoli

Kao što je prikazano u prošlom poglavlju, većina protokola koji žele osigurati razmjenu autentikacijskih podataka kao i slanje i primanje osjetljivih podataka preko računalne mreže upotrebljavaju SSL (engl. *Secure Sockets Layer*).

SSL protokol objavljen je još davne 1995. Riječ je o v2.0 protokola, dok je v1.0 bila interna i nikad nije javno objavljena. Inačica 2.0 imala je, međutim, brojne

sigurnosne ranjivosti tako da je tek inačica 3.0 postala opće prihvaćena. Inačica 3.0 izdana je 1996. i izdao ju je Netscape, tvrtka koja je napravila prvi popularni web-preglednik. Osnovni razlog za razvoj SSL-a bila je upravo činjenica da je bilo potrebno prenositi osjetljive podatke, što je preko HTTP-a kao protokola bilo nemoguće.

TLS protokol (engl. *Transport Layer Security*) je nadogradnja na SSL protokol i, sa stanovišta sigurnosti, mogu se smatrati ekvivalentnima. Razlike između ovih protokola su minorne, a TLS inačice 1.0 može čak komunicirati pomoću v3.0 SSL protokola. TLS je preuzeo IETF (engl. *Internet Engineering Task Force*) kako bi se osiguralo da ovaj protokol iznimno bitan za elektroničko poslovanje i općenito funkcioniranje velikog broja ostalih servisa na internetu ne bude u privatnom vlasništvu.

SSL/TLS protokol omogućuje čitav niz naprednih funkcionalnosti od kojih su najbitnije:

- mogućnost identifikacije klijentske i poslužiteljske strane putem certifikata (tzv. obostrana autentikacija, engl. *Mutual authentication*);
- mogućnost upotrebe različitih enkripcijskih algoritama za zaštitu povjerljivosti podataka koji se prenose.

SSL/TLS protokol temelji svoju sigurnost na asimetričnoj i simetričnoj kriptografiji koje se obje rabe tijekom uspostavljanja i za vrijeme sigurne sjednice. Asimetrična kriptografija upotrebljava se za autentikaciju dok se simetrična kriptografija upotrebljava za enkripciju podataka.

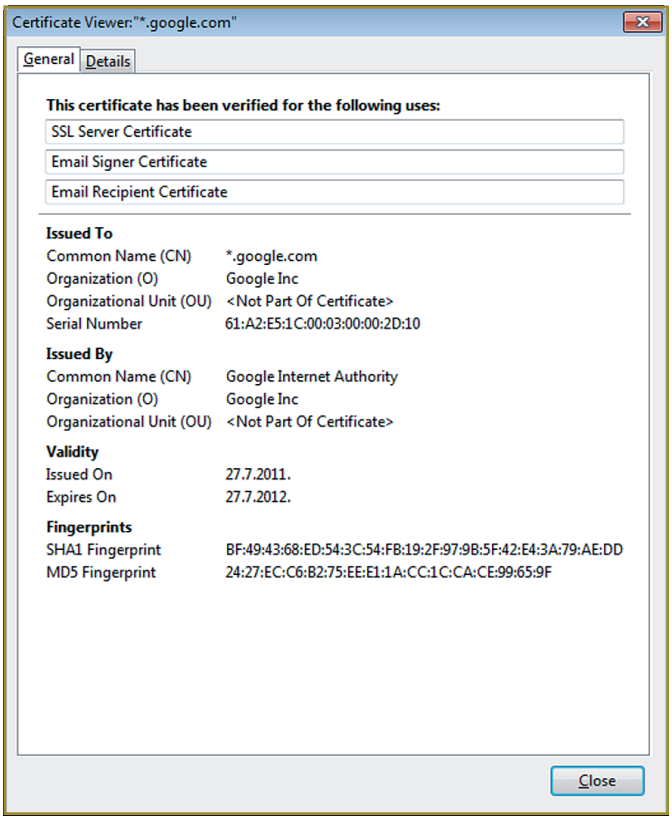
Sigurna sjednica uspostavlja se putem sljedećih koraka:

- klijentsko računalo koristi se TCP/IP komunikacijskom vezom da bi se spojilo na poslužiteljski servis koji podržava SSL ili TLS. Ako je riječ o SSL servisu, upotrebljava se zaseban port za navedeni servis (npr. mrežni port 443 za HTTPS). TLS komunikacijska veza omogućuje upotrebu protokola u čistom tekstualnom obliku u prvom koraku te uz posebne naredbe (npr. STARTTLS u SMTP-u za slanje elektroničke pošte) označava početak sigurne sjednice;
- klijentsko računalo sada poslužitelju šalje popis podržanih enkripcijskih algoritama. Ovaj je korak napravljen kako bi se omogućila kompatibilnost s različitim uređajima. Klijentsko računalo ujedno generira i slučajan broj (RN) koji će se upotrebljavati poslije u generiranju tajnog ključa. RN se šalje poslužitelju;
- poslužitelj odabire najsnažniji podržani enkripcijski algoritam, generira svoj dio slučajnog broja RN i oznaku sjednice te klijentskom računalu natrag šalje certifikat. Certifikat predstavlja digitalni dokument kojim po-

služitelj može potvrditi svoj identitet. U certifikatu je doslovno zapisano ime poslužitelja kojem taj certifikat pripada (npr. www.google.com), kao i javni ključ asimetričnog algoritma koji se upotrebljava;

- na osnovi dobivenog certifikata i digitalnog potpisa certifikata za koji se upotrebljava infrastruktura javnog ključa, PKI (detaljnije o PKI-u dostupno je u poglavlju 6.7) klijentsko računalo može potvrditi da zbilja komunicira s poslužiteljem s kojim je trebalo uspostaviti komunikaciju. Certifikati se rabe za sprječavanje napada presretanja mrežnog prometa (*Man-in-the-Middle* napada).

Primjer certifikata dan je na sljedećoj slici:



Slika 6.18. Certifikat za www.google.com domenu

- nakon što je klijentsko računalo dobilo natrag od poslužitelja poslužiteljski dio slučajnog broja RN, njegov certifikat koji sadržava i javni ključ poslužitelja te potvrdilo da je certifikat ispravan, klijentsko računalo generira simetrični tajni ključ na osnovi prethodno generiranih slučajnih brojeva RN (klijentskog i poslužiteljskog).

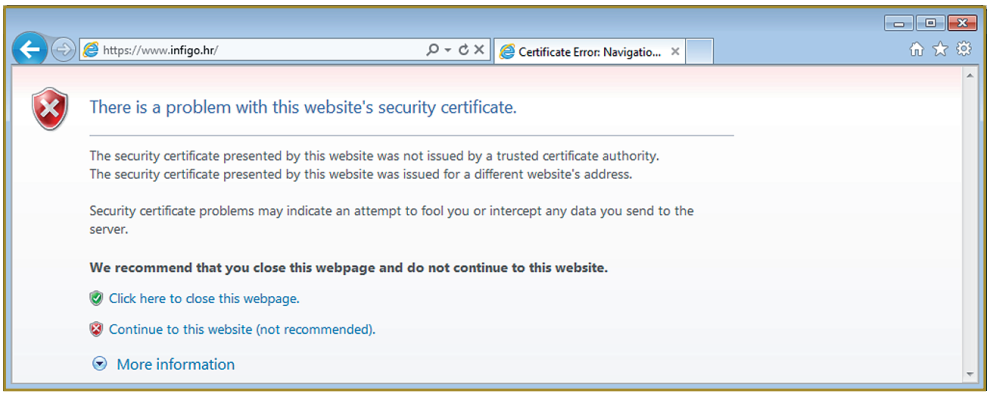
Ovaj generirani simetrični tajni ključ zatim se enkriptira javnim ključem dobivenim s certifikatom. Na ovaj se način postiže to da samo vlasnik para ključeva koji su potpisani certifikatom može dekriptirati generirani simetrični tajni ključ. Klijentsko računalo enkriptirani objekt šalje natrag poslužitelju;

- poslužitelj sada prima enkriptirani objekt. Ovaj je objekt moguće dekriptirati samo s tajnim asimetričnim ključem koji ima samo poslužitelj. Kada se objekt dekriptira, dobiva se generirani simetrični tajni ključ koji će se dalje upotrebljavati za enkripciju podataka između klijenta i poslužitelja. Simetrični se kriptografski algoritam rabi za enkripciju zbog svoje brzine.

Iz navedenog se postupka može vidjeti da su klijentsko i poslužiteljsko računalo na siguran način razmijenili tajni simetrični ključ koji obje strane moraju znati (zbog značajki simetričnih enkripcijskih algoritama, kao što je objašnjeno u poglavlju 3.3.1).

Čak i ako je maliciozni korisnik mogao cijelo vrijeme pregledavati mrežni promet, nije u stanju doći do navedenog simetričnog ključa budući da je isti bio poslan samo kao objekt enkriptiran javnim asimetričnim ključem poslužitelja.

Kao što se može vidjeti, SSL/TLS protokoli omogućuju uspostavljanje sigurne komunikacijske sjednice preko nesigurnih računalnih mreža, čak i u slučaju kada napadač pregledava mrežni promet. U slučaju da napadač pokušava modificirati mrežni promet ubacivanjem između klijentskog i poslužiteljskog računala, klijentsko će računalo detektirati takav pokušaj zbog neispravnog certifikata. Ovakve su detekcije česte pogotovo u web-preglednicima kada na poslužitelj nije postavljen certifikat koji je digitalno potpisan od strane tijela kojem se vjeruje. Primjer prozora s greškom prikazan je na sljedećoj slici.



Slika 6.19. Upozorenje Internet Explorer web-preglednika o neispravnom certifikatu

6.7. Infrastruktura javnog ključa

Kao što je prikazano u prethodnom poglavlju, ključan element uspostave sigurne sjednice pomoću SSL/TLS protokola je identificiranje jedne ili obje strane (SSL/TLS, kao što je već rečeno, dopušta obostranu autentikaciju ali se danas uglavnom rabi samo za autentikaciju poslužiteljske strane).

Da bi se poslužitelji uspješno mogli identificirati, potrebno je na njih postaviti ispravan certifikat. Certifikat predstavlja digitalni dokument u kojem je navedeno ime poslužitelja (njegovo puno DNS ime, npr. www.google.com), kao i njegov javni ključ te još neki drugi parametri poput vremena valjanosti certifikata.

Identifikacija poslužitelja u ovom slučaju polazi od asimetrične kriptografije (više o asimetričnoj kriptografiji dostupno je u poglavlju 3.3.2), a infrastruktura koja omogućuje uspostavu ovakvog sustava identifikacije naziva se infrastruktura javnog ključa (engl. *Public Key Infrastructure* – PKI).

PKI predstavlja temelje elektroničkog poslovanja te identifikacije poslužitelja (ali i drugih tijela, kao i krajnjih korisnika) na internetu. PKI omogućuje uspostavu sljedećih sigurnosnih načela:

- neporecivost (engl. *Non-repudiation*) – jedan od osnovnih zahtjeva elektroničkog poslovanja. Neporecivost omogućuje jednoznačno povezivanje nekog korisnika (ili aplikacije, servisa ili poslužitelja) s nekom aktivnosti, odnosno transakcijom.

  Primjer gdje je neporecivost kritična je internetsko bankarstvo. Kod ovog je servisa kritično da korisnici ne mogu poreći aktivnosti koje su proveli (npr. potrošnju novca). Neporecivost se zasniva na certifikacijskom centru (engl. *Certification Authority* – CA), koji hijerarhijski potvrđuje aktivnosti svojim digitalnim potpisom;
- povjerljivost podataka – kontekstu PKI-a, povjerljivost podataka postiže se asimetričnom kriptografijom. Drugi se protokoli mogu služiti proizvoljnim kriptografskim algoritmima, ali identifikacija subjekata biti će temeljena na asimetričnoj kriptografiji;
- integritet podataka – čuva se upotrebom i potpisivanjem algoritama sažimanja, kao što je objašnjeno u poglavlju 3.3.3).

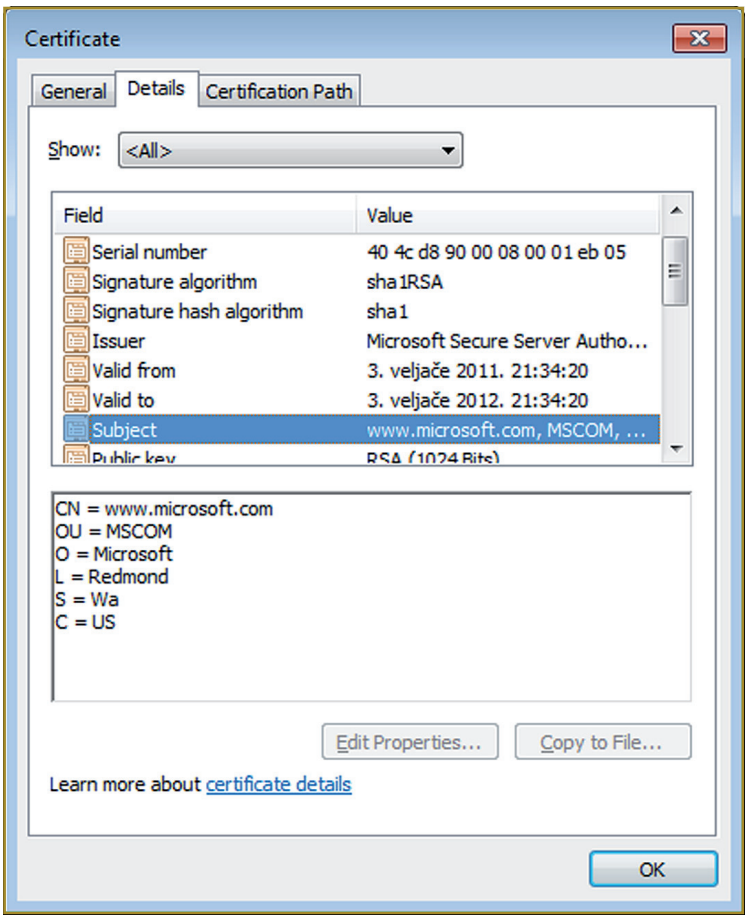
Osnovnu komponentu PKI infrastrukture predstavlja certifikacijski centar. Osim certifikacijskog centra, elemente PKI infrastrukture čine i registracijski centar te krajnji korisnik.

Krajnji korisnik je subjekt čiji se identitet želi provjeriti i potvrditi. On generira par ključeva pomoću podržanog asimetričnog algoritma (npr. RSA), pohranjuje

tajni ključ na sigurno mjesto te certifikacijskom centru šalje zahtjev za izdavanje certifikata za njegov identitet, zajedno s generiranim javnim ključem.

Certifikacijski centar sada provjerava identitet korisnika koristeći se nekim drugim načinom komunikacije koji niti ne mora biti povezan s internetom. Na primjer, zaposlenik u certifikacijskom centru može tražiti osobnu kartu korisnika te na taj način potvrditi njegov identitet. Ako je identitet bio ispravan, certifikacijski se centar sada služi svojim tajnim ključem kako bi digitalno potpisao javni ključ korisnika. Ovaj se digitalni potpis naziva certifikat.

U certifikatu se dakle nalazi digitalno potpisan javni ključ korisnika (poslužitelja), javni ključ certifikacijskog centra (kako bi se mogao provjeriti digitalni potpis) te neki drugi podaci poput vremena trajanja i svrhe certifikata. Detaljni podaci o certifikatu prikazani su na sljedećem primjeru.



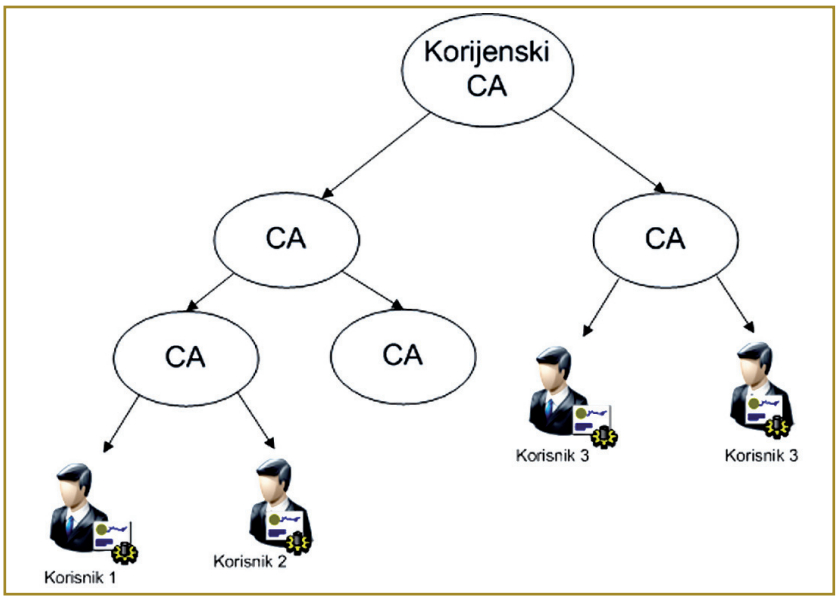
Slika 6.20. Certifikat stranice www.microsoft.com

Uloga registracijskog centra je zapravo samo u primanju zahtjeva za izdavanjem certifikata i provjeri identiteta korisnika koji je poslao zahtjev. Naime, kod velikih PKI sustava ove se dvije funkcije (izdavanje certifikata i provjera identiteta) mogu podijeliti između certifikacijskog i registracijskog centra.

Između ostalih zadataka, certifikacijski centar drži i repozitorij izdanih certifikata te certifikata koji su bili opozvani. Može biti više razloga zbog kojih bi se pojedini certifikat opozvao, poput činjenice da je možda bio kompromitiran tajni ključ korisnika ili da je korisnik npr. promijenio posao (ako je certifikat bio izdan stvarnoj osobi, a ne poslužitelju) te stari certifikat više nije važeći. Lista opozvanih certifikata naziva se CRL (engl. *Certificate Revocation List*) te bi ju klijentski programi (npr. web-preglednici) zapravo trebali pregledavati pri svakoj uspostavi nove sigurne sjednice.

6.7.1. PKI hijerarhija

Kao što se može vidjeti, certifikacijski centar ključno je tijelo kojem svi korisnici PKI infrastrukture moraju vjerovati. Jasno je da jedan takav certifikacijski centar ne može zadovoljiti potrebe svih korisnika na internetu. PKI zato omogućuje definiranje hijerarhije gdje je moguć veći broj certifikacijskih centara koji su organizirani u više slojeva u obliku stablaste strukture, kao što je prikazano na sljedećoj slici.



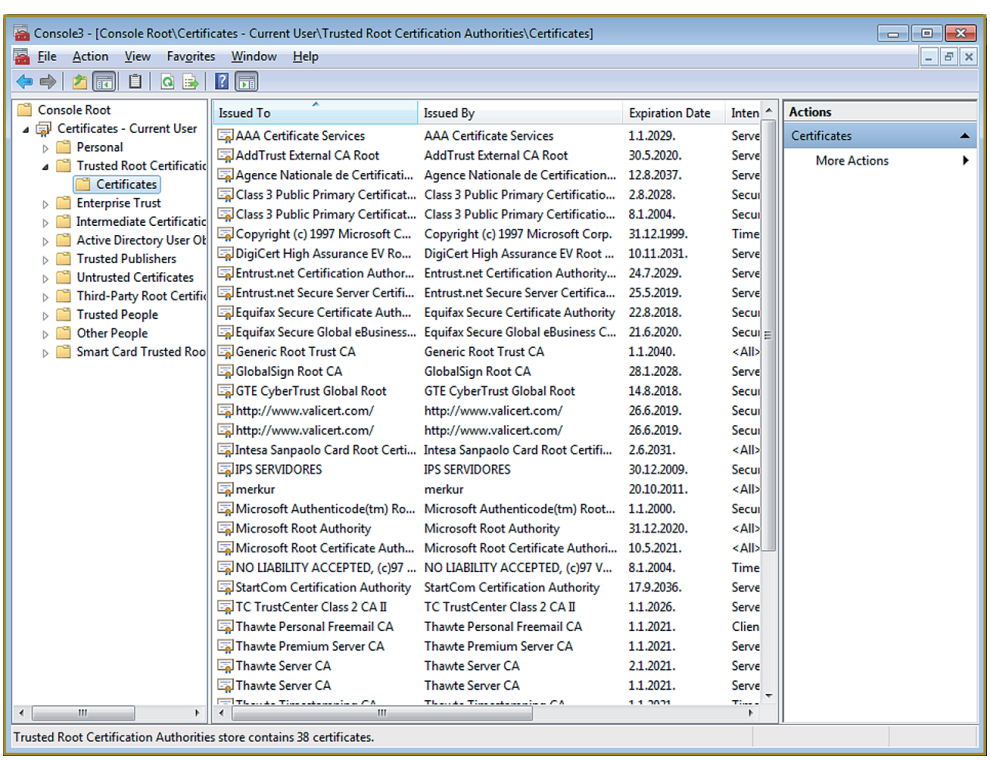
Slika 6.21. Hijerarhijska PKI infrastruktura

Na ovaj je način moguće raspodijeliti opterećenje između višestrukih certifikacijskih centara čime se postiže i veća skalabilnost i sigurnost sustava. Da bi se zadržao model vjerovanja, korijenski certifikacijski centar izdaje poseban certifikat certifikacijskim centrima ispod sebe koji to mogu načiniti i dalje.

Kada je potpisan korisnikov (krajnji) certifikat, on u sebi sadržava kompletnu putanju certifikacijskih centara koji su potpisali navedeni certifikat. Klijentsko računalo zbog toga treba imati pohranjene samo certifikate korijenskih certifikacijskih centara za uspješno funkcioniranje kompletnog sustava.

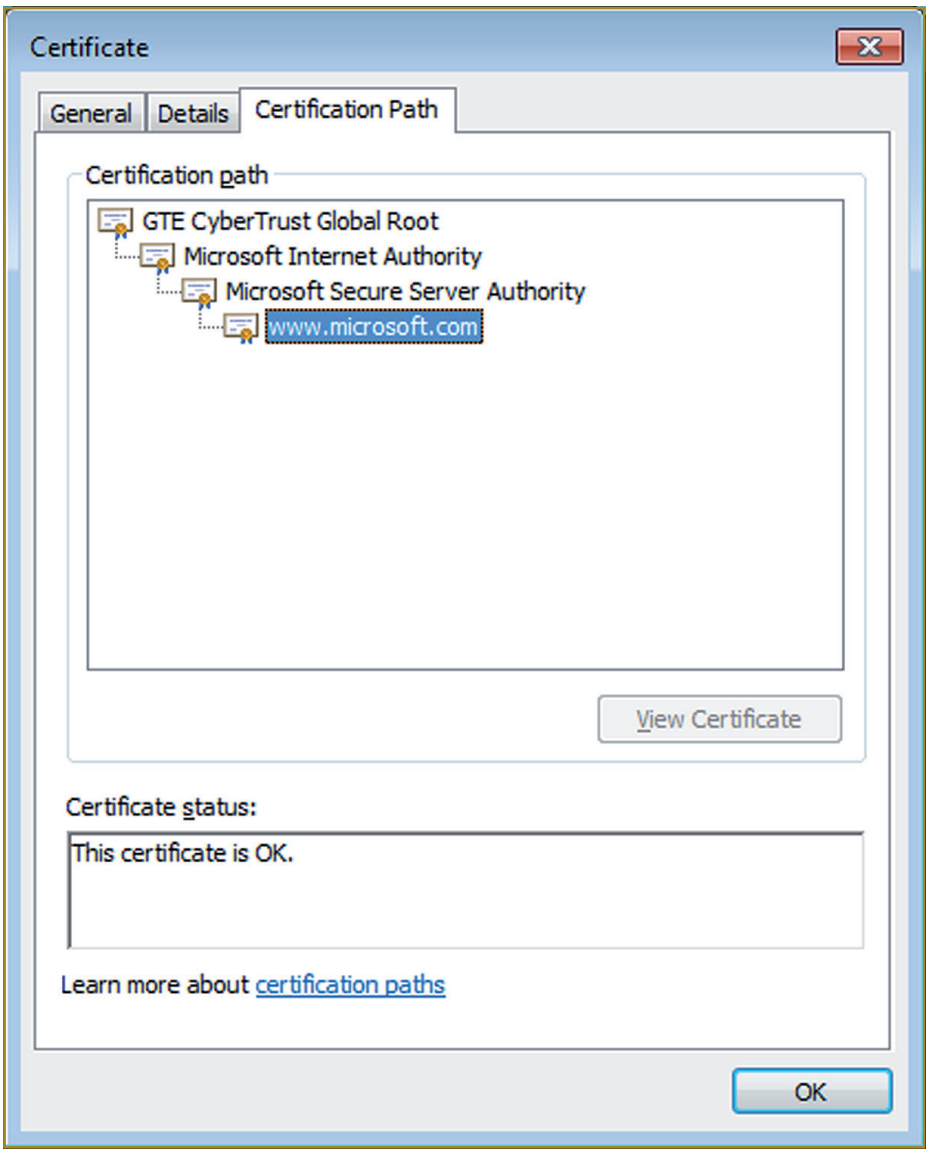
Svi operacijski sustavi dolaze već s određenom listom korijenskih certifikacijskih centara, koja može brojati i nekoliko stotina certifikata. Aplikacije mogu upotrebljavati i svoje posebne liste, tako da npr. na Windows operacijskim sustavima Internet Explorer rabi listu koja dolazi s operacijskim sustavima, dok Mozilla Firefox web-preglednik uvijek ima svoju privatnu listu certifikata korijenskih certifikacijskih centara, bez obzira na operacijski sustav na kojem je instaliran.

Na Windows operacijskim sustavima listu certifikata moguće je pogledati pomoću Microsoft Management Console alata, odnosno Certificates dodatka, kao što je prikazano na sljedećoj slici.



Slika 6.22. MMC alat omogućuje upravljanje certifikatima na računalu

Pri upotrebi bilo kojeg certifikata moguće je vidjeti kompletan certifikacijski put (engl. *Certification path*) koji pokazuje put certificiranja, sve od vršnog, korijenskog certifikacijskog centra pa do zadnjeg koji je potpisao prikazani certifikat. Na sljedećoj je slici prikazan certifikacijski put certifikata korištenog na www.microsoft.com stranici. Može se vidjeti da je korijenski certifikacijski centar „GTE CyberTrust Global Root“, a da je zadnji certifikacijski centar koji je zapravo i potpisao www.microsoft.com certifikat „Microsoft Secure Server Authority“.



Slika 6.23. Certifikacijski put www.microsoft.com certifikata

6.8. Sigurnost bežičnih računalnih mreža

Bežične računalne mreže zadnjih godina postaju iznimno popularne zbog brojnih prednosti koje pružaju, od kojih su najbitnije jednostavnost i fleksibilnost. Sva su prijenosna računala danas opremljena bežičnom mrežnom karticom, a bežične se računalne mreže često sreću i u tvrtkama.

Danas ima mnogo bežičnih standarda koji se mogu upotrebljavati, kao i sigurnosnih postavki od kojih se neke smatraju nedovoljnima za ispunjavanje današnjih sigurnosnih zahtjeva.

Svi standardi bežičnih računalnih mreža koji se danas upotrebljavaju zasnovani su na osnovnom 802.11 standardu. Standardi se služe različitim frekvencijama te omogućuju uspostavljanje različitih brzina, kao što je prikazano u sljedećoj tablici.

IEEE standard	Frekvencija	Brzina	Komentar
802.11	2.4 GHz	2 Mbps	Izvorni standard, ne upotrebljava se.
802.11a	5 GHz	54 Mbps	Pokušaj stvaranja standarda za brze bežične računalne mreže. Nije kompatibilan s 802.11b i 802.11g standardima.
802.11b	2.4 GHz	11 Mbps	Prvi raširen standard bežičnih mreža. Cilj mu je bio omogućiti brzine jednake 10 Mbps Ethernet brzinama koje su bile standardne u doba razvoja.
802.11g	2.4 GHz	54 Mbps	Dodatak 802.11b standardu kako bi se povećale njegove brzine.
802.11n	2.4 i 5 GHz	300 Mbps	Novi standard koji omogućuje još veće brzine bežičnih mreža. Kompatibilan je sa svim prethodnim standardima (a, b i g).
802.11i	-	-	Dodatak za sigurnost, izbačen nakon što su otkriveni ozbiljni sigurnosni problemi u WEP-u. Ovaj standard omogućuje WPA/WPA2 zaštitu bežičnih računalnih mreža.

Zbog svoje prirode, uz standardne zahtjeve i prijetnje koji se pojavljuju pred žičanim računalnim mrežama (zadovoljavanje zahtjeva CIA trokuta kao što je navedeno u poglavlju 1.1, odnosno sprječavanje DAD trokuta prema poglavlju

1.2), pred bežične se računalne mreže postavlja čitav niz novih zahtjeva vezanih za sigurnost.

- Zbog prirode prometa neovlašteni korisnik može vrlo jednostavno pregledavati mrežni promet. To znači da određene mjere zaštite moraju biti konstantno primijenjene za sav mrežni promet. Ovaj je zahtjev otežan činjenicom da je vrlo teško ili gotovo nemoguće kontrolirati fizičko područje na kojem je dostupan signal bežičnih računalnih mreža: napadači se mogu koristiti specijaliziranim, usmjerenim bežičnim antenama koje omogućuju pregledavanje bežičnog mrežnog prometa s velikih udaljenosti.
- Iako su napadi uskraćivanjem računalnih resursa prisutni i na žičnim računalnim mrežama, kod implementacije bežičnih računalnih mreža potrebno je obratiti posebnu pažnju na ovu kategoriju napada.
  Napadač može iskoristiti činjenicu da može praktički sa svoje fizičke lokacije slati bežični mrežni promet te na taj način probati zauzeti ograničen prostor koji bežične računalne mreže imaju na raspolaganju. Jednako je tako moguće i da drugi uređaji koji nemaju nikakve veze s bežičnim računalnim mrežama, ali se služe istim ili približnim frekvencijama rada unesu šum i smetnju u bežične računalne komunikacije.
  U svrhu detektiranja kolizija današnje bežične mreže primjenjuju CSMA/CA metodu (engl. *Carrier Sense Multiple Access with Collision Avoidance*). Iako se u klasičnim sustavima pa tako i u običnom Ethernetu implementira CSMA/CD metoda (engl. *CSMA with Collision Detection*), u bežičnim je mrežama ova metoda nepraktična zbog činjenice da sve radne stanice ne vide komunikaciju od svih drugih uređaja u bežičnoj računalnoj mreži. Razlog ovome je jednostavan – zbog fizičke je pozicije lako moguće da dvije radne stanice komuniciraju bežičnom računalnom mrežom preko pristupne točke (engl. *Access Point*) te se međusobno ne vide: svaka vidi samo pristupnu točku.
  CSMA/CA, za razliku od CSMA/CD metode primjenjuje aktivni pristup detektiranju kolizija. CMSA/CA prije slanja mrežnog prometa provjerava komunicira li bilo koji drugi uređaj već preko bežične računalne mreže. Ako je komunikacijski kanal slobodan, radna stanica koja želi slati mrežni promet prvo traži dozvolu te, kada ju dobije, nastavlja sa slanjem mrežnog prometa. Na ovaj se način znatno smanjuje broj kolizija, no može se vidjeti da maliciozni korisnik još uvijek može namjerno izazvati kolizije na bežičnoj računalnoj mreži. Ovo je zapravo i dosta čest slučaj upravo kod napada na WEP protokol, kao što je opisano u poglavlju 6.9.
- Napadi ubacivanja u mrežni promet (engl. *Man-in-the-Middle*) na bežičnim računalnim mrežama predstavljaju još veću opasnost nego na žič-

nim računalnim mrežama. Naime, pri spajanju radne stanice na pristupnu točku kod bežičnih računalnih mreža, ako se ne upotrebljavaju snažni mehanizmi autentikacije pristupne točke, maliciozni se korisnik može vrlo jednostavno predstaviti kao pristupna točka. Ako u takvom slučaju maliciozni korisnik navede klijentsko računalo da se spoji na njegovu, lažnu pristupnu točku, moguće je vrlo jednostavno modificiranje daljnjeg mrežnog prometa i, naravno, njegovo pregledavanje.

- Konačno, ponovno slanje mrežnog prometa (engl. *Message Reply*) veliki je problem u bežičnim računalnim mrežama. Budući da potencijalni maliciozni korisnik može prema definiciji vidjeti i prikupljati sav mrežni promet (naravno, ako se ispravno primjenjivala enkripcija za zaštitu osjetljivih podataka, malicioznih ih korisnik ne može vidjeti, ali i dalje može prikupiti enkriptirane podatke), ništa ga ne sprječava u ponovnom slanju paketa u njihovom izvornom obliku. Drugim riječima, čak i kada maliciozni korisnik vidi samo enkriptirane podatke, može ih ponovno slati u bežičnu računalnu mrežu. Ovisno o višem protokolu koji se upotrebljavao ovo može predstavljati problem (npr. ako se upotrebljava TCP, paketi će biti identificirani kao kopije i odbačeni, ali ako se rabi UDP, paketi mogu biti čak i prihvaćeni od ciljnog poslužitelja). U svrhu sprječavanja provođenja ovakvih napada bežični protokoli danas upotrebljavaju niz sigurnosnih mehanizama, kao što je navedeno u nastavku poglavlja.

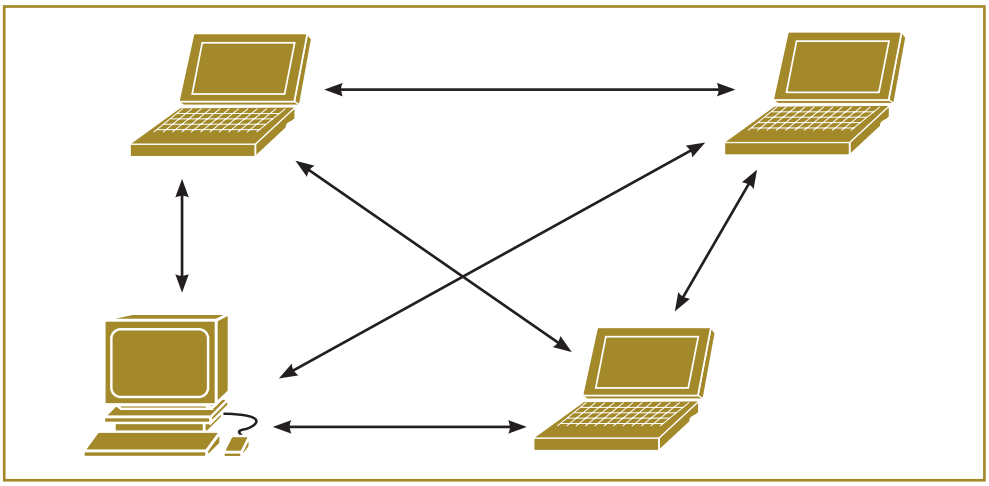
Kao što se može vidjeti, prijetnje na bežične računalne mreže puno su veće od onih na žične računalne mreže te u skladu s time zahtijevaju i više investicija te kompleksnije sigurnosne mehanizme.

6.8.1. Bežične mrežne konfiguracije

802.11 standard definira dva entiteta u bežičnim mrežnim konfiguracijama: radnu stanicu (engl. *Station* – STA), koja predstavlja klijentsko računalo koje se spaja na bežičnu računalnu mrežu (u ovom kontekstu to može biti klijentsko računalno poput PC-a ili poslužitelj) te pristupnu točku (engl. *Access Point* – AP), koja predstavlja središnje mjesto komunikacije.

Standard također definira dva tipa mrežne konfiguracije koji definiraju kako entiteti u bežičnoj računalnoj mreži međusobno komuniciraju: *Ad-hoc* i infrastrukturna bežična mrežna konfiguracija.

Ad-hoc bežična mrežna konfiguracija namijenjena je manjim grupama radnih stanica (STA) koje se sve nalaze u međusobnom dometu, dakle svaka radna stanica može izravno komunicirati s drugom radnom stanicom u navedenoj grupi. Primjer *ad-hoc* bežične mrežne konfiguracije dan je na sljedećoj slici.



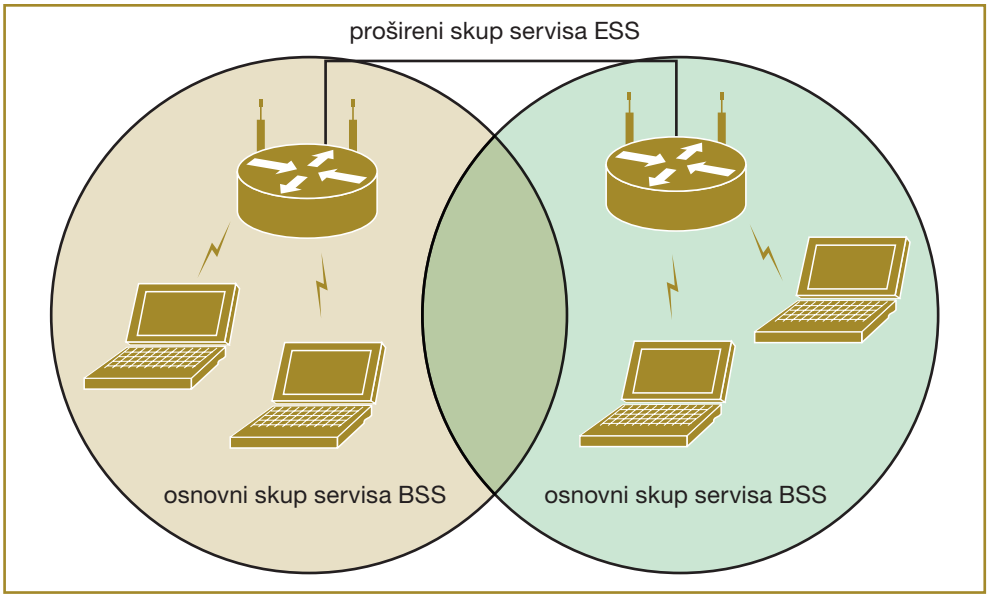
Slika 6.24. Ad-hoc bežična mrežna konfiguracija

Prednost *ad-hoc* bežične mrežne konfiguracije je njezina jednostavnost: korisnici praktički sami mogu iznimno jednostavno implementirati i konfigurirati ovakve bežične računalne mreže koje su, u skladu s time, namijenjene uglavnom manjim, kućnim bežičnim računalnim mrežama. Klasičan primjer upotrebe *ad-hoc* bežičnih računalnih mreža je kada korisnik želi spojiti pisač na svoje računalo ili jednostavno razmijeniti podatke između dva računala bez dizanja kompleksnije bežične računalne mreže.

Infrastrukturna konfiguracija bežičnih računalnih mreža namijenjena je, kako joj i ime govori, umrežavanju većeg broja radnih stanica (STA) primjenom pristupnih točaka (AP). Primjena pristupnih točaka prvenstveno omogućuje komunikaciju u slučajevima kada radne stanice ne mogu izravno komunicirati jedna s drugom (npr. međusobno su previše udaljene, ali je pristupna točka i dometu i jedne i druge radne stanice). Jednako tako, pristupna točka administratoru omogućuje jedinstveno definiranje konfiguracije i svih parametara sigurnosti bežične računalne mreže. Konačno, infrastrukturna se konfiguracija upotrebljava za povezivanje bežičnih i žičnih računalnih mreža gdje upravo pristupna točka predstavlja „most“ koji povezuje ove dvije mreže. Zbog svega navedenog infrastrukturna se konfiguracija bežičnih računalnih mreža danas rabi u najvećem broju slučajeva te pogotovo u tvrtkama gdje se zaposlenicima želi omogućiti fleksibilnost rada na bežičnoj računalnoj mreži.

Standard također definira i grupe računala spojenih u infrastrukturne bežične računalne mreže. Tako se jedna grupa spojena preko jedne pristupne točke naziva osnovnim skupom servisa (engl. *Basic Service Set* – BSS), dok se mre-

za više pristupnih točaka naziva proširenim skupom servisa (engl. *Extended Service Set* – ESS), kao što je prikazano na sljedećoj slici.



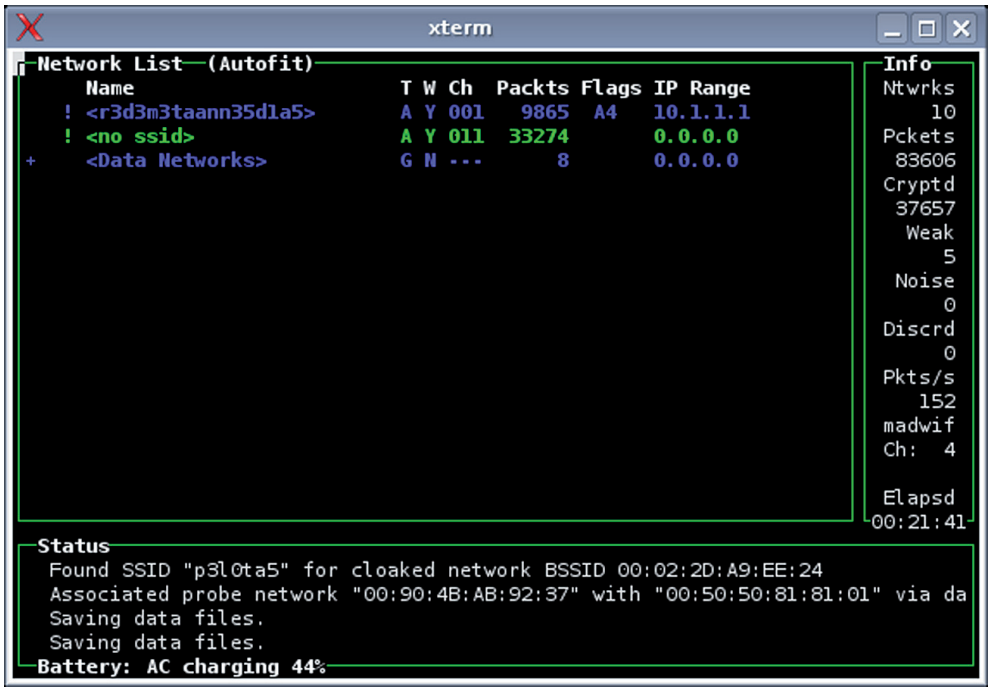
Slika 6.25. Infrastrukturna bežična računalna mreža

Budući da je moguće na jednoj fizičkoj lokaciji imati više (puno) bežičnih računalnih mreža, napravljen je mehanizam koji omogućuje razlikovanje bežičnih računalnih mreža, odnosno koji omogućuje krajnjem korisniku specificiranje bežične računalne mreže kojoj želi pristupiti. U tu se svrhu rabi oznaka računalne mreže, SSID (engl. *Service Set Identifier*). SSID predstavlja tekstualni naziv bežične računalne mreže kako bi se korisnicima olakšao odabir mreže. SSID omogućuje grupiranje bežičnog mrežnog prometa više radnih stanica – čak i kada se upotrebljava isti bežični kanal (frekvencija), bežični je promet moguće kategorizirati prema SSID-u. Naravno, ako puno bežičnih računalnih mreža rabi isti kanal, jasno je da je povećana i mogućnost kolizija što u konačnici može rezultirati degradiranjem performansi bežične računalne mreže.

Budući da pristupne točke također upotrebljavaju SSID oznaku da bi ustanovile kome je primljeni bežični mrežni promet namijenjen (budući da sve pristupne točke u dometu primaju navedeni mrežni promet), jasno je da SSID mora biti prenošen u čistom tekstualnom obliku. Zbog ove je činjenice također jasno da SSID ne predstavlja nikakvu mjeru zaštite – naime, određene pristupne točke dopuštaju konfiguriranje tako da se ne šalju signalni paketi (engl. *Beacon packets*). Signalni paketi šalju svim radnim stanicama u dometu pristupne točke

informacije o ponuđenim bežičnim računalnim mrežama, odnosno njihovom SSID-u. Na taj način korisnik može na radnoj stanici vidjeti popis bežičnih računalnih mreža koje su u dometu te se jednostavno spojiti na njih (ako zadovoljava ostale uvjete koji su navedeni u nastavku poglavlja).

Isključivanje slanja paketa s informacijama o bežičnim mrežama ne povisuje razinu sigurnosti bežične računalne mreže zbog jednostavne činjenice da svi ostali mrežni paketi koji se razmjenjuju između drugih radnih stanica i navedene pristupne točke moraju imati SSID informacije u čistom tekstualnom obliku. Budući da navedene bežične mrežne pakete zapravo mogu vidjeti svi korisnici u dometu, jasno je da navedeni korisnici mogu vrlo jednostavno doći do popisa bežičnih računalnih mreža u dometu, čak i kad su signalni paketi isključeni. Ovakav je postupak moguć jednostavnim pasivnim pregledavanjem bežičnog mrežnog prometa, sličnog onom spomenutom u poglavlju 6.1. U ovu su svrhu razvijeni posebni programi koji su u stanju nadgledati bežični mrežni promet (iako se jednaka funkcija donekle može postići i već opisanim programima poput Wiresharka i Tcpdumpa). Na sljedećoj je slici prikazan Kismet, paket namijenjen pregledavanju bežičnog mrežnog prometa koji će prikupiti sve dostupne pakete te pokazati popis detektiranih SSID-ova.



Slika 6.26. Kismet, paket za pregledavanje bežičnog mrežnog prometa

6.8.2. Autentikacija i kontrola pristupa bežičnim računalnim mrežama

802.11 standard definira dva načina autentikacije uređaja koji se žele spojiti na bežičnu računalnu mrežu. Potrebno je napomenuti da se izvorne specifikacije oba načina autentikacije danas ne smatraju sigurnima, osim ako se ne upotrebljava 802.11i sigurnosni dodatak standardu.

Dva podržana načina autentikacije su otvorena autentikacija te autentikacija dijeljenim tajnim ključem.

- Otvorena autentikacija dopušta pristup bežičnoj računalnoj mreži svim radnim stanicama koja zadovoljavaju sljedeća dva jednostavna uvjeta:
  - radna stanica mora specificirati točan SSID koji definira bežičnu računalnu mrežu kojoj želi pristupiti. Kao što je objašnjeno u prethodnom poglavlju (6.8.1), SSID se bežičnom računalnom mrežom šalje u čistom tekstualnom obliku te se kao takav ne može smatrati javnim čak ni u slučajevima kada pristupne točke ne šalju informacijske pakete;
  - MAC adresa radne stanice zadovoljava uvjete postavljene od stane pristupne točke. Administrator pristupne točke može postaviti konfiguraciju bijelim listama koja dopušta spajanje samo onim radnim stanicama čija je MAC adresa na listi, dok je svim ostalim radnim stanicama spajanje na navedenu pristupnu točku onemogućeno. Ovdje je također potrebno napomenuti da se MAC adresa šalje bežičnom računalnom mrežom u čistom tekstualnom obliku te je kao takva podložna napadu pregledavanja mrežnog prometa. Dodatno, MAC adresa bežične mrežne kartice može se vrlo jednostavno promijeniti, u većini slučajeva čak i bez potrebe za bilo kakvih vanjskim alatima, što napadaču u ovom slučaju omogućuje jednostavno spajanje na bežičnu računalnu mrežu.

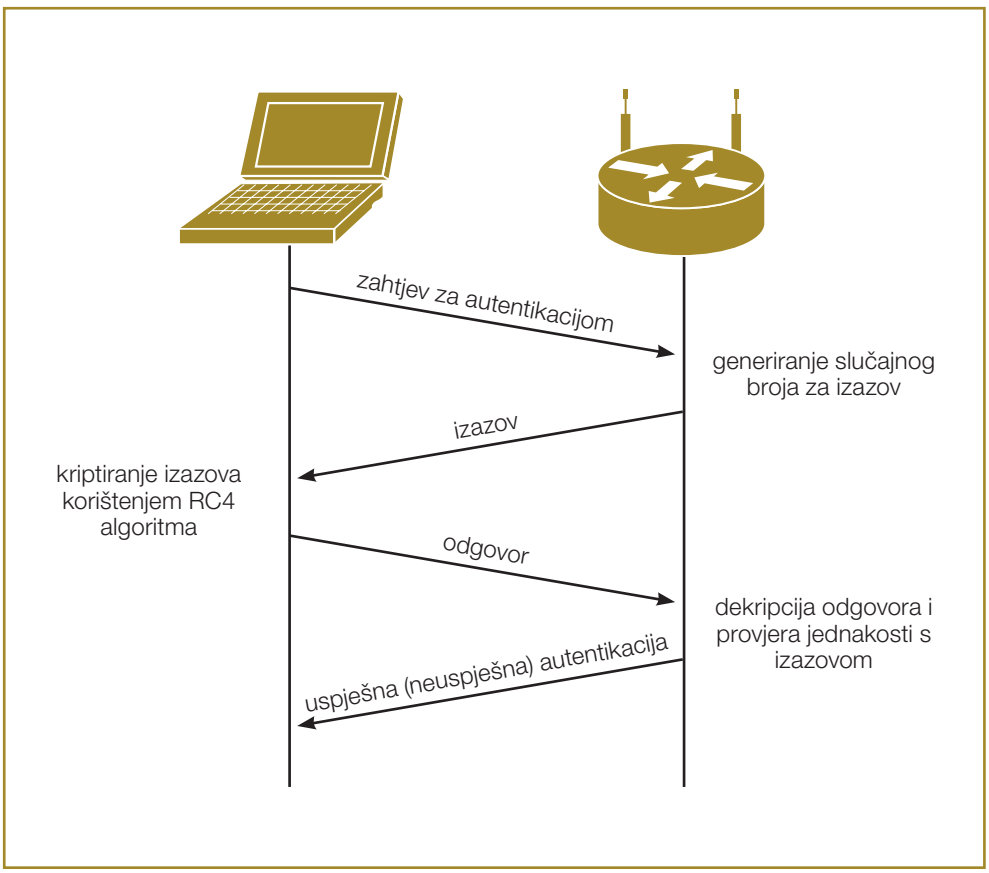
  Zbog svega navedenog otvorena se autentikacija treba upotrebljavati isključivo za javne pristupne točke, budući da maliciozni korisnik može vrlo jednostavno zaobići sve sigurnosne mehanizme ovog tipa autentikacije.
- Autentikacija dijeljenim tajnim ključem polazi od činjenice da, osim zadovoljavanja uvjeta navedenih kod otvorene autentikacije, radne stanice koje se žele spojiti na bežičnu računalnu mrežu moraju imati i dijeljeni tajni ključ.

  U ovom se slučaju najčešće rabi neka inačica tzv. izazov-odgovor algoritama (engl. *Challenge-Response*) kako bi se potvrdilo zna li korisnik radne stanice ispravni tajni ključ. Ovisno o sigurnosnom algoritmu koji je implementiran, autentikacija dijeljenim tajnim ključem može biti više ili ma-

nje sigurna. Na primjer, WEP sigurnosni algoritam ima brojne nedostatke koji su navedeni u sljedećem poglavlju te se autentikacija dijeljenim tajnim ključem u kombinaciji s WEP algoritmom također ne može smatrati sigurnom.

lako je danas najčešće korišten oblik autentikacije krajnjih korisnika, u većim okruženjima autentikacija dijeljenim tajnim ključem predstavlja velike probleme upravljanju tajnim ključem. Naime, u slučaju da zaposlenik napusti tvrtku, potrebno je promijeniti tajni ključ što zahtijeva i promjenu na svim radnim stanicama koje se služe istom bežičnom mrežom. U ovakvim se velikim okruženjima zato preporučuje upotreba naprednijih algoritama za autentikaciju koje podržavaju WPA i WPA2 sigurnosni algoritmi, kao što je navedeno u poglavlju 6.10.

Primjer autentikacije dijeljenim tajnim ključem kod WEP sigurnosnog algoritma prikazan je na sljedećoj slici.



Slika 6.27. Autentikacija dijeljenim tajnim ključem

6.9. WEP sigurnosni protokol

WEP (engl. *Wired Equivalent Privacy*) protokol definiran je u svrhu zaštite povjerljivosti i integriteta podataka koji se šalju bežičnom računalnom mrežom. Kao što mu i samo ime govori, WEP je napravljen u svrhu pružanja jednake razine sigurnosti bežičnim računalnim mrežama kao što ju imaju žične računalne mreže. Na žalost, kao što je objašnjeno u nastavku ovog poglavlja, WEP protokol ima brojne nedostatke tako da se danas može smatrati da ne nudi nikakvu sigurnost te ga je potrebno u potpunosti zaobići i primjenjivati isključivo sigurnije protokole poput WPA i WPA2.

U svrhu zaštite povjerljivost podataka WEP rabi RC4 simetrični kriptografski algoritam koji je temeljen na enkriptiranju toka podataka (više o RC4 algoritmu dostupno je u poglavlju 3.3.1). Za očuvanje integriteta podataka upotrebljava se CRC32 algoritam (engl. *Cyclic Redundancy Check*). CRC32 algoritam obično služi za detekciju grešaka u prijenosu te je potrebno napomenuti da nije riječ o kriptografski snažnom algoritmu. Ova činjenica u kombinaciji s nekim drugim inherentnim sigurnosnim problemima tokovnih simetričnih algoritama poput RC4 omogućuje provođenje složenih, ali vrlo uspješnih napada protiv WEP-a.

WEP algoritam definira tri metode zaštite povjerljivosti podataka prenošenih bežičnom računalnom mrežom:

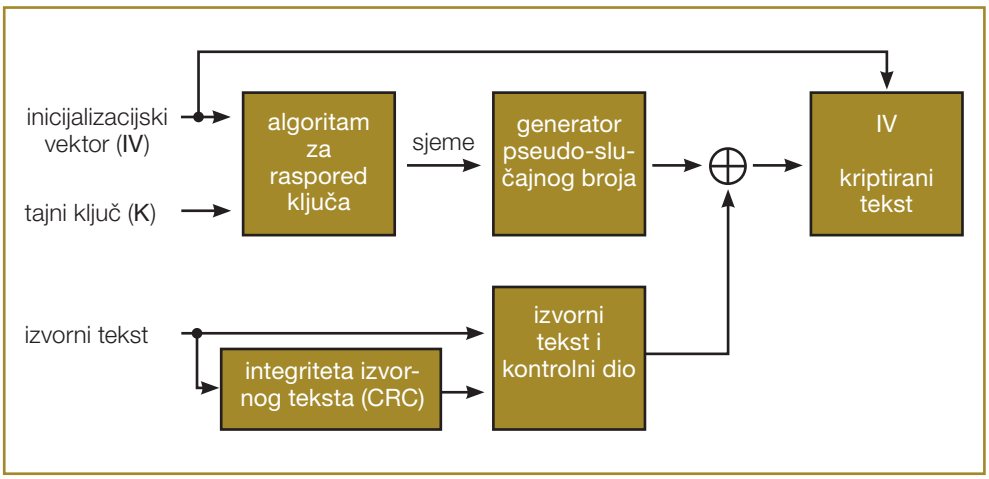
- implementacija bez enkripcije, kao što joj ime govori, ne nudi nikakvu zaštitu povjerljivosti prenesenih podataka;
- upotreba 40-bitnog tajnog ključa;
- upotreba 128-bitnog (u praksi 104-bitnog) tajnog ključa.

Nakon što su identificirani prvi sigurnosni problemi WEP algoritma pojavili su se proizvođači koji su nudili čak i nestandardnu upotrebu 256-bitnog tajnog ključa. Potrebno je napomenuti da, međutim, zbog sigurnosnih problema u samom algoritmu veličina ključa ne znači veliku razliku u sigurnosti podataka već samo produljuje vrijeme napada. Drugim riječima, bez obzira na veličinu WEP tajnog ključa, maliciozni napadač u konačnici može doći do ovog ključa i na taj način u potpunosti kompromitirati bežičnu računalnu mrežu koja rabi WEP za zaštitu.

Kao što je već rečeno, WEP protokol za zaštitu povjerljivosti informacija upotrebljava RC4 simetrični kriptografski algoritam koji enkriptira tokove podataka. Kod RC4 algoritma ključan parametar pri enkripciji predstavlja tzv. inicijalizacijski vektor (engl. *Initialization Vector*), broj koji se zajedno s tajnim ključem rabi u svrhu generiranja pseudo-slučajnog niza brojeva. Nakon generiranja ovog niza

brojeva on jednostavno provodi XOR operaciju s ulaznim podacima da bi se dobili enkriptirani podaci.

Na drugoj strani koja treba dekriptirati podatke generira se opet isti pseudo-slučajni niz (koji ovisi o inicijalizacijskom vektoru i tajnom ključu), provodi se XOR operacija s enkriptiranim podacima i dobiva se natrag izvorni, neenkriptirani tekst. WEP enkripcijski proces prikazan je na sljedećoj slici.



Slika 6.28. RC4 enkripcija kod WEP algoritma

Kao što se može vidjeti, da bi se podaci uspješno enkriptirali, obje strane koje razmjenjuju podatke moraju znati dva ključna parametra: inicijalizacijski vektor i tajni ključ.

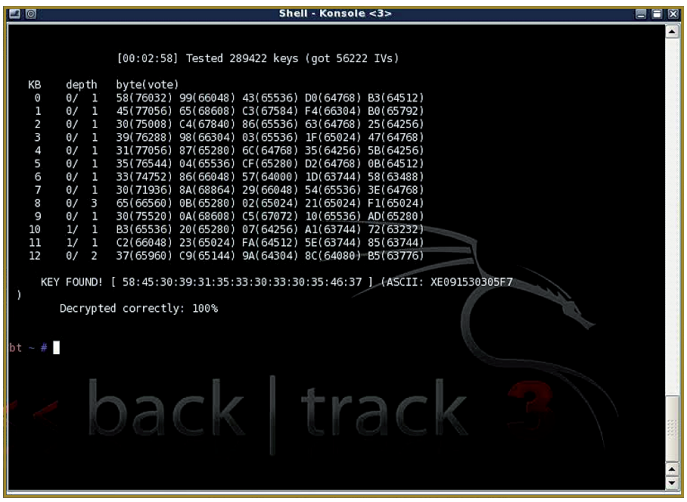
Tajni je ključ moguće unaprijed konfigurirati na obje strane, no inicijalizacijski vektor treba zapravo pri svakoj enkripciji biti što je moguće više slučajan. Kod RC4 algoritma koji se rabi u WEP-u, inicijalizacijski vektor je veličine 24 bita, što omogućuje generiranje 2 ^ 24 vrijednosti, odnosno 16.777.216 različitih brojeva. Samo iz ovog primjera može se vidjeti da je vjerojatnost ponavljanja istog inicijalizacijskog vektora dosta velika, pogotovo kod bežičnih računalnih mreža koje prenose veliku količinu podataka – za svaki mrežni paket inicijalizacijski je vektor drukčiji. Jednako tako, budući da obje strane moraju znati inicijalizacijski vektor, on se bežičnom računalnom mrežom prenosi u čistom tekstualnom obliku, što znači da ga napadač koji pregledava mrežni promet može vidjeti bez ikakvog problema.

Ova 24 bita predstavljaju dio sveukupnog tajnog ključa, tako da je u slučaju kada je tajni ključ veličine 40 bitova kompletan ključ velik 64 bita.

Ključan događaj koji napadaču omogućuje dolazak do tajnog ključa predstavlja slučaj kada se rabi inicijalizacijski vektor koji je već prije bio upotrijebljen te ako je napadač u stanju pogoditi dijelove izvornog teksta. Na žalost, kada je riječ o mrežnom prometu, posao napadača dodatno je olakšan zbog činjenice da mrežni paketi imaju velike dijelove zaglavlja koji su uvijek jednaki – npr. kod IP mrežnog prometa IP zaglavlje ima dosta polja koja su uvijek jednaka, pogotovo kada je riječ o *broadcast* mrežnom prometu.

Kao što se može vidjeti, napadač treba samo dovoljno dugo pregledavati mrežni promet jer je zbog relativno malog konačnog broja inicijalizacijskih vektora velika vjerojatnost da će se isti inicijalizacijski vektor ponoviti. Dodatno, napadač može probati slati lažirane pakete u bežičnu računalnu mrežu ne bi li tako „natjerao“ pristupnu točku u slanje dodatnih paketa i generiranje novih inicijalizacijskih vektora.

Kad napadač prikupi dovoljni broj mrežnih paketa, moguće je provesti napredne kriptoanalitičke napade ne bi li se otkrio tajni ključ. Postoji čitav niz alata koji automatiziraju ovakve napade tako da napadač ne mora imati nikakvo znanje. Osim toga, zbog napretka u napadima na WEP protokol danas se do tajnog ključa može doći u vrlo kratkom razdoblju, čak i na bežičnim računalnim mrežama koje ne generiraju puno mrežnog prometa. Na sljedećoj je slici prikazan aircrack-ng program, dio paketa namijenjenog napadanju bežičnih računalnih mreža. Na slici se može vidjeti da je paket uspješno došao do izvornog tajnog ključa uz samo 130 tisuća paketa prikupljenih s bežične računalne mreže koja se napadala. Jednako se tako može vidjeti i da je bio korišten vrlo kompleksan ključ veličine 104 bita.



Slika 6.29. Aircrack-ng paket pri provođenju napada na WEP bežičnu računalnu mrežu

Zbog svega bi navedenog upotrebu WEP protokola za zaštitu bežičnih računalnih mreža trebalo zaobilaziti te se preporučuje upotreba WPA/WPA2 protokola, kao što je navedeno u nastavku.

6.10. WPA/WPA2 sigurnosni protokoli

Nakon javnog objavljivanja detalja o sigurnosnim ranjivostima WEP protokola postalo je jasno da je WEP nedovoljan za odgovarajuću zaštitu bežičnih računalnih mreža. IEEE je zbog toga razvio 802.11i sigurnosni dodatak u kojem su definirane brojne mogućnosti podizanja razine sigurnosti bežičnih računalnih mreža.

WPA (engl. *Wi-Fi Protected Access*) napravljen je kao prijelazni standard čija je namjena bila ponuditi mogućnost više razine zaštite bežičnih računalnih mreža uz maksimalnu moguću kompatibilnost prema postojećem hardveru. Naime, budući da je u trenutku objavljivanja 802.11i sigurnosnog dodatka u svijetu postojao već izrazito veliki broj pristupnih točaka koje upotrebljavaju WEP za zaštitu, bilo je potrebno omogućiti i vlasnicima ovih pristupnih točaka podizanje sigurnosti svojih bežičnih računalnih mreža.

WPA protokol donosi mnogo novih mogućnosti, ali osnovnu enkripciju i dalje zasniva na RC4 simetričnom algoritmu, uz eliminaciju sigurnosnih problema prisutnih u WEP protokolu. Ovo je napravljeno zbog činjenice da se pristupne točke uglavnom koriste specijaliziranim hardverskim dodacima za enkripciju, što znači da je enkripcijski algoritam nemoguće promijeniti na starim uređajima (mora ostati RC4), ali se ostatak algoritma može promijeniti, i to obično putem osvježavanja *firmwarea*.

WPA2 protokol, s druge strane, dizajniran je bez ovakvih ograničenja tako da se niti ne koristi RC4 algoritmom za enkripciju podataka, već je povjerljivost osigurana upotrebom AES simetričnog enkripcijskog algoritma.

Kao što se može vidjeti na primjeru WEP algoritma, dijeljeni tajni ključevi predstavljaju problem u smislu da sigurnost cijelog protokola ovisi o njima. Zato su WPA i WPA2 algoritmi dodali upotrebu posebne funkcije za generiranje slučajnih tajnih ključeva određene veličine. Ovakva se funkcija naziva PBKDF2 (engl. *Password-Based Key Derivation Function*) i prmjenom kompleksnih matematičkih funkcija omogućuje generiranje snažnih ključeva. Ovi ključevi kao jedan ulazni parametar uzimaju čak i ime bežične mreže (SSID), što ih čini otpornima na određene klase napada na zaporke.

Uz generiranje kompleksnih ključeva, jedan od problema WEP-a je i činjenica da je ključ statičan, odnosno da se ne mijenja kroz cijeli životni vijek navedenog ključa. Kod WPA protokola ovaj se problem rješava upotrebom TKIP protokola (engl. *Temporal Key Integrity Protocol*), koji uz neke dodatne zaštite omogućuje i promjenu tajnog ključa koji se rabi za zaštitu svakog pojedinog mrežnog paketa. Na ovaj se način dodatno otežava provođenje napada na bežične računalne mreže koje se koriste WPA algoritmom za zaštitu.

Kod WPA2 algoritma upotrebljava se još napredniji algoritam zaštite pod imenom CCMP (engl. *Counter Mode with Cipher Block Chaining MAC Protocol*). CCMP pruža čitav niz metoda zaštite, uz već spomenutu upotrebu AES algoritma sa 128-bitnim tajnim ključem (generiranom upotrebom PBKDF2 funkcije).

Sljedeća tablica prikazuje usporedbu korištenih kriptografskih algoritama, veličine tajnog ključa te podržanih metoda autentikacije za WEP, WPA i WPA2 algoritme.

Mogućnost	WEP	WPA	WPA2
Kriptografski algoritam	RC4	RC4	AES
Veličina tajnog ključa	40-bitni ili 104-bitni	128-bitni	128-bitni
Zaštita integriteta	Enkriptirani CRC32	MICHAEL algoritam	CCMP
Autentikacija	Otvorena ili dijeljeni tajni ključ	EAP ili dijeljeni tajni ključ	EAP ili dijeljeni tajni ključ
Distribucija tajnih ključeva	Ručna	Ručna ili 802.1X	Ručna ili 802.1X

6.11. Zaštita kritičnih mrežnih servisa

Dostupnost računalne mreže, osim o mrežnim uređajima bez kojih računalna mreža uopće ne može funkcionirati (preklopnici, usmjerivači), uvelike ovisi i o kritičnim mrežnim servisima. Ispravan rad ovih servisa ključan je i za rad računalne mreže. Osim omogućavanja pristupa mrežnim resursima, kritični mrežni servisi navedeni u ovom poglavlju bitni su i za procese poput autentikacije.

Dva kritična mrežna servisa bez kojih drugi servisi uopće nisu dostupni ili ne mogu obavljati svoju funkciju u današnjim su računalnim mrežama svakako DNS i DHCP. Uz navedena dva servisa potrebno je još spomenuti i HTTP(S)

te FTP servise, koji se danas također rabe za ostvarivanje pristupa podacima i aplikacijama koje korisnici svakodnevno upotrebljavaju.

6.11.1. DNS servis

DNS servis zadužen je za razlučivanje DNS imena u IP adrese, proces bez kojeg korisnici uopće ne mogu pristupiti resursima kojima žele pristupiti. Osim što korisnici ne mogu pristupiti željenim resursima bez funkcioniranja ovog servisa, mnogi su drugi protokoli temeljeni na DNS-u. Tako npr. u Microsoft Windows domeni, pri upotrebi Kerberosa za autentikaciju, DNS imena poslužitelja odnosno resursa ključna su za traženje karata za pristup (više o Kerberos autentikaciji dostupno je u poglavlju 5.1).

DNS servis je jedan od najstarijih servisa na internetu. Budući da se pri izvornom dizajnu ovog servisa nije obraćalo previše pažnje na sigurnost informacijskih sustava, DNS servis doživio je brojne promjene, ali je sama jezgra još uvijek jednaka onoj prvotnoj.

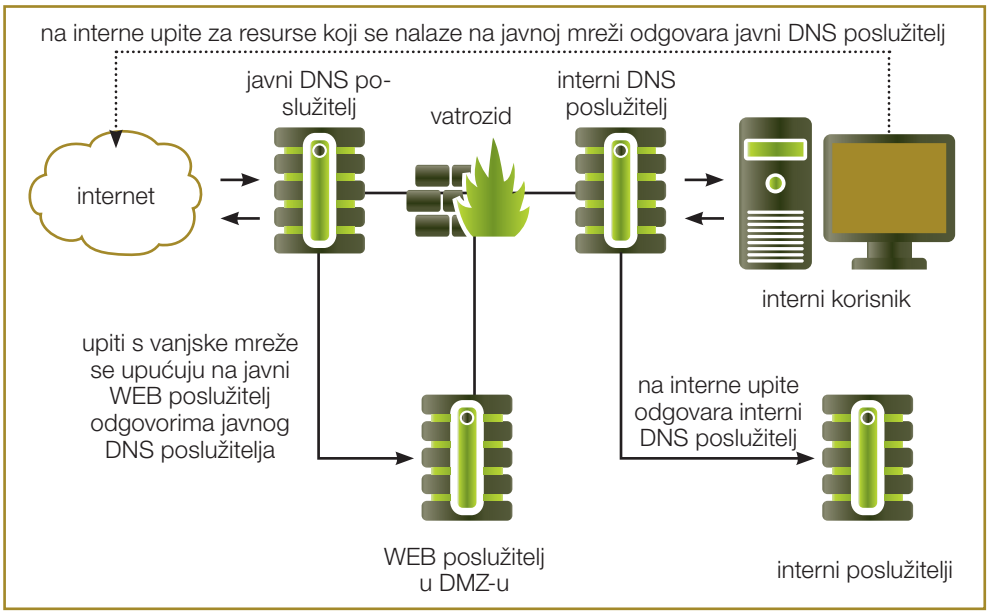
Kao i kod drugih servisa, jedan od osnovnih zahtjeva za postavljanje sigurnog DNS servisa je pravodobno i redovito instaliranje sigurnosnih zakrpi koje se odnose na ovaj servis.

Na Microsoft Windows operacijskim sustavima riječ je o servisu koji se automatski instalira na domenske poslužitelje (zbog, kao što je već rečeno, ovisnosti Kerberos autentikacije o DNS servisu).

Na Linux operacijskim sustavima obično se upotrebljava jedan od dostupnih DNS programskih paketa, s tim da je Bind programski paket najpopularniji i najviše zastupljen.

Osim redovitog instaliranja sigurnosnih zakrpi posebnu je pažnju potrebno posvetiti i konfiguraciji ovog servisa. Kako DNS servis zapravo samo omogućuje dolazak do informacija koje su pohranjene u njemu, potrebno je obratiti pažnju oko prava slanja upita. Naime, interni podaci u DNS sustavu ne bi trebali biti dostupni korisnicima na internetu već samo onim korisnicima koji su na internoj računalnoj mreži tvrtke. Ove se dozvole jednostavno postavljaju prema IP adresi izvornog upita tako da se slanje upita na određene DNS zone (npr. tvrtka.local) dopušta samo ako IP adresa upita pripada određenom adresnom prostoru (internoj računalnoj mreži tvrtke).

U pojedinim se implementacijama instaliraju čak i višestruki DNS poslužitelji u tzv. odvojenoj shemi (engl. *Split DNS*). Odvojena shema prikazana je na sljedećoj slici.



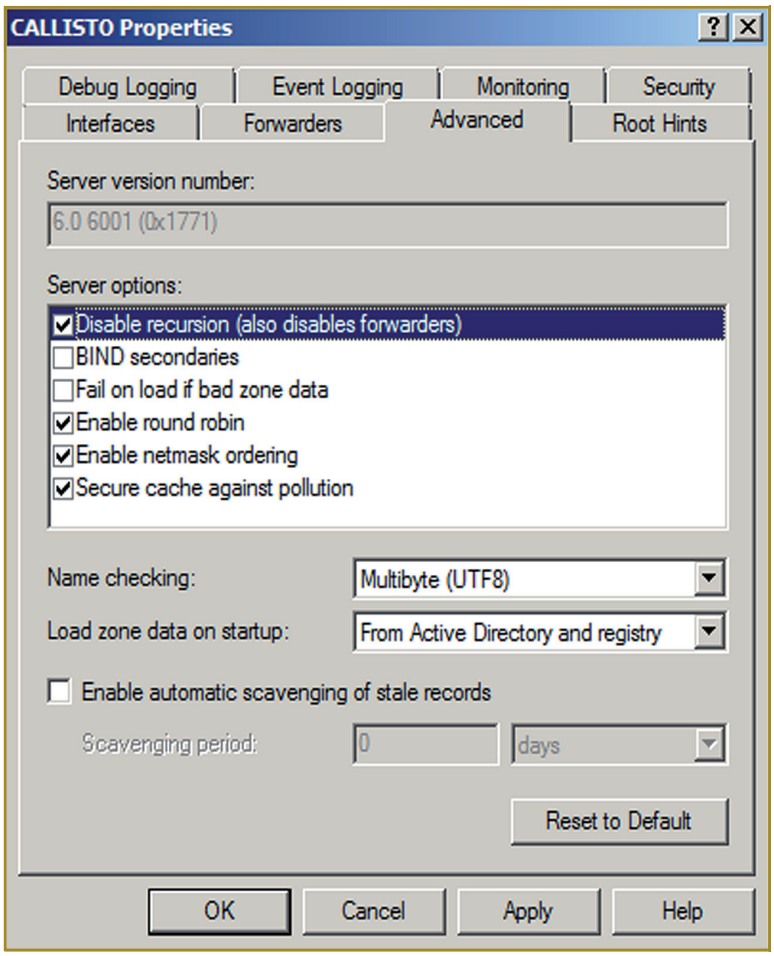
Slika 6.30. Implementacija odvojenih DNS poslužitelja

Na ovaj je način moguće vrlo jednostavno konfigurirati ispravne DNS zone koje poslužitelji pokrivaju: javni DNS poslužitelj imat će samo podatke o javno dostupnim poslužiteljima i servisima tvrtke (onima koji se nalaze u DMZ-u), dok će interni DNS poslužitelj imati sve ostale podatke o internim resursima i poslužiteljima. Kada interna računala šalju DNS upite, iste je moguće procesirati na internom DNS poslužitelju, a moguće je konfigurirati i tzv. usmjeravanje gdje interni DNS poslužitelj prima upit, šalje ga javnom DNS poslužitelju na razlučivanje i kada dobije odgovor, vraća ga klijentskom računalu koje je izvorno poslalo upit.

Kada se na računalnim mrežama upotrebljava DHCP servis, vrlo se često dopušta automatsko modificiranje DNS zapisa u Windows domeni. Tako npr. ako klijentsko računalo dobije određenu IP adresu, zapis u DNS zoni bit će automatski modificiran da za navedeno klijentsko računalo pokazuje na ispravnu IP adresu. Iako je ova mogućnost korisna, potrebno ju je ograničiti samo na računala u Windows domeni kako bi se spriječilo zloupotrebljavanje navedene funkcije. Računalima u domeni korisnici neće moći mijenjati imena pa će ona uvijek odgovarati onima upisanima u zoni.

Konačno, kod ojačanja sigurnosti DNS servisa potrebno je i obratiti pažnju na IP adrese kojima se dopušta razlučivanje adresa preko navedenog DNS poslužitelja. Razlučivanje adresa znači da klijentska računala mogu poslati upit za bilo kojim DNS imenom poslužitelju, koji zatim provodi korake razlučivanja te

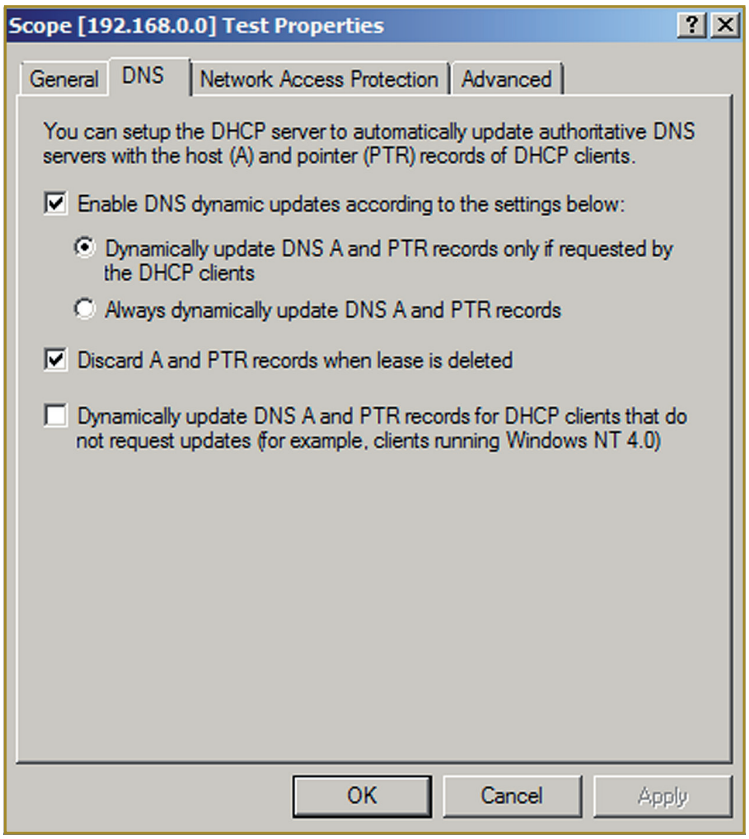
šalje konačan odgovor klijentskom računalu. Ovaj je postupak potrebno dopustiti samo za legitimne korisnike. Drugim riječima, javni DNS poslužitelji trebali bi biti tako konfigurirani da odbijaju upite za bilo koje DNS ime osim za ono koje pripada domeni za koju su nadležni. Na Windows DNS poslužiteljima ovo se postiže onemogućavanjem rekurzije, kao što je prikazano na sljedećoj slici.



Slika 6.31. DNS postavke za onemogućavanje rekurzivnih upita

6.11.2. DHCP servis

DHCP servis je još jedan bitan servis zadužen za dodjeljivanje IP adresa klijentskim računalima. Riječ je o relativno jednostavnom servisu koji sam po sebi nije imao znatnijih sigurnosnih propusta već su napadi na ovaj servis uglavnom usmjereni na provođenje napada uskraćivanjem računalnih resursa.



Slika 6.32. Automatsko modificiranje DNS zapisa potrebno je dopustiti samo domenskim računalima

Jedan od najčešćih napada na DHCP servis je iskorištavanje svih dostupnih IP adresa. Kod ovog napada napadač stalno mijenja MAC adresu svoje mrežne kartice te traži nove IP adrese od DHCP poslužitelja ne bi li potrošio sve raspoložive IP adrese. U svrhu sprječavanja ovog napada obično se preporučuje iskorištavanje funkcije sigurnosti sučelja (engl. *Port security*) na preklopniku. Navedena funkcija može ograničiti broj MAC adresa po pojedinom sučelju i tako efektivno spriječiti navedeni napad.

Drugi česti napad na DHCP servis temelji se na dizanju lažnog DHCP servisa gdje napadačev DHCP servis počinje izdavati IP adrese korisnicima, a samim time i mijenjati njihove postavke što napadaču omogućuje provođenje daljnjih napada. U svrhu sprječavanja ovakvih napada moderna mrežna oprema omogućuje nadgledanje DHCP paketa (engl. *DHCP snooping*). Jednako je tako moguće upotrebljavati specijalizirane programske pakete koji omogućuju pregledavanje mrežnog prometa i detekciju lažnih DHCP poslužitelja.

6.11.3. HTTP(S) sigurnost

HTTP protokol također je (gledajući sigurnosne značajke i povijest protokola) još jedan protokol koji nije zabilježio veće sigurnosne ranjivosti. S druge strane, najveći broj napada na poslužitelje te uspješnih kompromitacija u zadnjih se nekoliko godina dogodio upravo preko web-servisa. Razlog ovome nisu sigurnosne ranjivosti u HTTP protokolu već u web-aplikacijama koje se instaliraju na web-poslužitelje (najčešće Microsoft IIS na Windows operacijskom sustavu te Apache na Linux operacijskim sustavima).

Pri instalaciji web-servisa tako je zapravo najviše pažnje potrebno obratiti upravo web-aplikacijama koje će se upotrebljavati. Postoji čitav niz sigurnosnih ranjivosti specifičnih za web-aplikacije, a koje su detaljnije opisane u poglavlju 4.2. Zbog toga je potrebno osigurati da se web-aplikacije instalirane na web-poslužitelj također redovito osvježavaju te da se sigurnosne zakrpe za iste redovito instaliraju.

Ako je potrebno osigurati povjerljivost podataka koje web-aplikacija razmjenjuje s korisnicima (dakle, ako nije riječ o javnim web-stranicama), svakako je potrebno implementirati HTTPS, odnosno upotrebu SSL-a. Konfiguracija HTTPS-a u načelu je relativno jednostavna te moduli koji omogućuju ovaj rad automatski dolaze s većinom HTTP poslužitelja. Osim ispravne konfiguracije pri instalaciji HTTPS servisa potrebno je pribaviti i ispravan certifikat koji će biti instaliran na poslužitelj. Ako je riječ o javnom poslužitelju, preporučuje se generiranje ključeva te dobavljanje certifikata od jednog od prepoznatih certifikacijskih centara koji su automatski upisani na Windows operacijske sustave te popularne web-preglednike poput Mozilla Firefoxa. Na taj će se način omogućiti ispravno uspostavljanje SSL komunikacijske veze između klijentskog računala i poslužitelja bez ikakvih upozorenja i grešaka. U protivnom, ako poznati certifikacijski centar nije potpisao certifikat, korisniku će biti prikazano upozorenje. Ako je riječ o internim HTTPS poslužiteljima, moguća je upotreba internog certifikacijskog centra, čiji se certifikat zatim može ubaciti na klijentska računala kako bi se spriječilo upozorenje web-preglednika. Kako je nakon ubacivanja ovog certifikata interni certifikacijski centar praktički postao ekvivalentan javnim certifikacijskim centrima, s njegovim je tajni ključem potrebno vrlo pažljivo rukovati.

6.11.4. FTP servis

FTP servis obično se upotrebljava za dopuštanje skidanja i postavljanja velikih datoteka. Budući je riječ o, kao što je već bilo spomenuto u poglavlju 6.5, protokolu koji sve informacije šalje u čistom tekstualnom obliku, upotreba FTP poslu-

žitelja preporučuje se samo za javne datoteke. Drugim riječima, FTP poslužitelj trebao bi omogućavati skidanje samo anonimnim korisnicima.

Ako postoji potreba za autentikacijom korisnika, preporučuje se upotreba alternativnih servisa FTP-u, poput SFTP-a. SFTP omogućuje prenošenje svih podataka u enkriptiranom obliku te snažnu autentikaciju korisnika, koja je također enkriptirana te tako zaštićena od možebitnih napada pregledavanja mrežnog prometa.

Konačno, ako je riječ o javnim FTP poslužiteljima (poslužiteljima dostupnim anonimnim korisnicima na internetu), potrebno je posebnu pažnju obratiti na mogućnosti pohranjivanja datoteka koje je potrebno onemogućiti. Naime, danas je vrlo čest slučaj da se javno dostupni FTP poslužitelji zloupotrebljavaju od strane malicioznih korisnika koji na njih nerijetko pohranjuju ilegalne datoteke u svrhu razmjene s drugim malicioznim korisnicima na internetu. Ovakve datoteke dovode u opasnost tvrtku koja je odgovorna za sadržaje postavljene na poslužitelje koji su u njezinom vlasništvu. Zbog toga je pohranjivanje datoteka na FTP poslužitelje potrebno dopustiti samo s internih javnih adresa ili, još bolje, samo administratorima sustava koji izravno mogu postaviti datoteke na FTP poslužitelj.

7. Operativni sloj

Sigurnosne procedure, politike i drugi dokumenti osnova su zaštite informacijskog sustava koja je objašnjena na tehničkoj razini u prethodnim poglavljima. Bez kvalitetne sigurnosne politike nije moguće niti organizirati ispravnu zaštitu informacijskog sustava i upravo je jedna od najvećih grešaka koju puno tvrtki čini nepostojanje kvalitetne sigurnosne politike iza koje stoji i samo vodstvo odnosno uprava tvrtke.

Potrebu kvalitetne sigurnosne politike prepoznale su i krovne organizacije koje kroz različite standarde zahtijevaju izradu ovih dokumenata – jedan takav primjer je i ISO 27001 standard koji je već bio objašnjen u poglavlju 1.3.3.

Kada se govori o sigurnosnim politikama i procedurama, iznimno je bitno istaknuti da kvalitetne sigurnosne politike trebaju postaviti cilj koji je dosežan tvrtci. Drugim riječima, sigurnosne politike ne smiju predstavljati samo mrtvo slovo na papiru koje se ne može ili ne želi ostvariti već pravi, realni cilj tvrtke. Bilo koji zahtjev koji se stavlja pred informacijski sustav, ali i njegove korisnike, kroz sigurnosne politike tako mora biti analiziran te postavljen samo ako ima smisla.

Primjerice, vrlo se često u sigurnosnim politikama rade greške čiji učinak na kompletnu sigurnost informacijskog sustava nije dobro procijenjen. Jedan takav primjer jest redovito mijenjanje zaporki korisnika, zahtjev koji se tehnički može vrlo jednostavno implementirati npr. na domenskom poslužitelju Windows domene. No pitanje koje se ovdje postavlja je upravo koliko bi trebalo biti razdoblje nakon kojeg korisnici moraju promijeniti svoju zaporku? Klasično vrijeme koje se postavlja za promjenu zaporke je tri mjeseca, no nekada administratori postavljaju čak i puno kraća razdoblja od samo mjesec dana, a možda i kraće.

U ovom je slučaju potrebno napraviti analizu rizika koji se nastoji eliminirati uvođenjem ovakvog zahtjeva kao i posljedica na informacijski sustav, zaposlenike, ali i administratore koje će ovaj zahtjev imati.

Tjeranjem korisnika na mijenjanje zaporke nastoji se umanjiti rizik od malicioznog korisnika koji je uspio doći do nečije zaporke. Pravo pitanje koje je potrebno ovdje postaviti je kako je maliciozni korisnik došao do te zaporke? Ako je riječ o napadu kada maliciozni korisnik pokušava pogoditi zaporku, kao što je bilo opisano u poglavlju 5.1.1, navedeni je napad puno učinkovitije riješiti primjenjivanjem drugih tehničkih metoda zaštite poput zaključavanja korisničkih računa te postavljanjem zahtjeva za odabir snažne zaporke.

Ako problem predstavlja mogućnost da korisnici dijele zaporke, i ovdje je sigurnosni problem moguće riješiti na druge načine: sigurnosna bi politika, kao prvo, trebala u potpunosti zabraniti dijeljenje zaporki tako da je korisnicima jasno da zaporke predstavljaju informaciju o kojoj su sami odgovorni i koju ne smiju nikome dati. Dodatno, ako se rabe grupni korisnički računi, njih je također potrebno zabraniti, odnosno dopustiti samo uz posebne zahtjeve koji moraju biti opravdani i odobreni od odgovorne osobe.

Kao što se može vidjeti, odgovor na pitanje koliko često treba mijenjati zaporke nije nimalo jednostavno dati. Ako je razdoblje predugo, izlažemo se riziku da napadač može doći do neke zaporke, a ako je prekratko, povećavamo mogućnost da korisnici zaborave zaporke, ne mogu se prijaviti na sustav te tako opterećuju i helpdesk tvrtke. Osim toga, generiranje snažne zaporke korisnicima nije jednostavno te se vrlo često može kao nuspojava dogoditi da korisnici počnu zapisivati zaporke na papir jer ih ne mogu zapamtiti, što predstavlja još veći sigurnosni rizik za tvrtku.

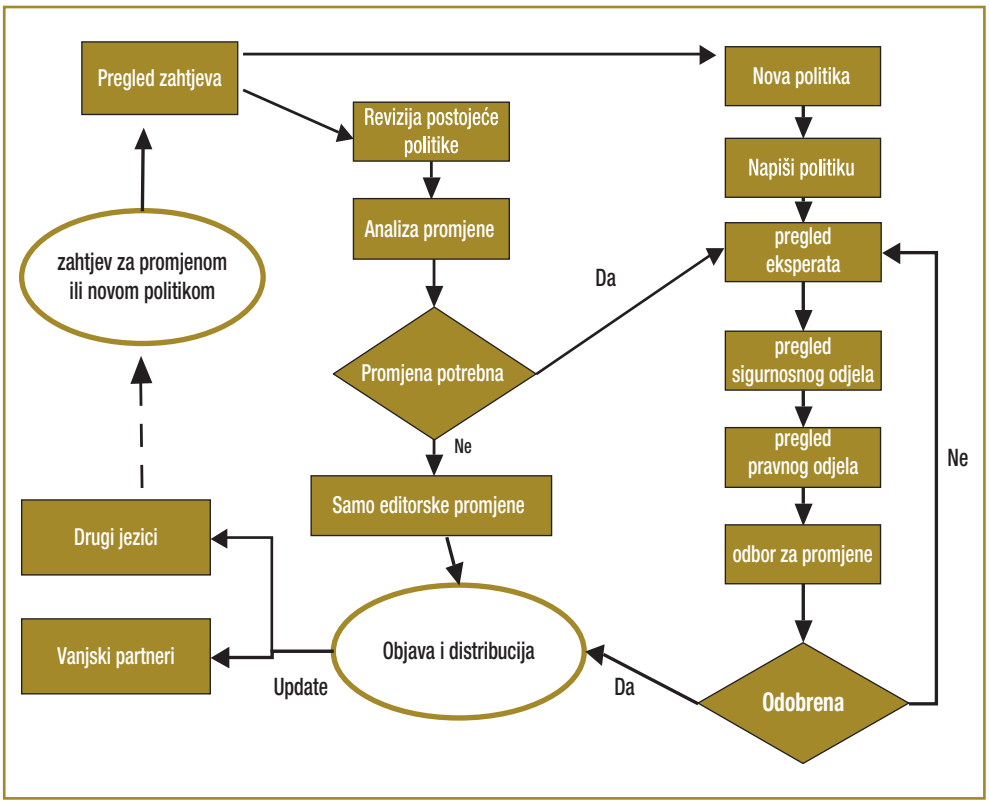
Kada se navedeni faktori izvažu – mogućnost pogađanja zaporke od strane napadača ili provođenja nekog drugog napada, može se vidjeti da prečesto mijenjanje zaporki ima kontraproduktivni učinak te da razdoblje od tri, pa čak i šest mjeseci ima optimalan učinak, pogotovo kada se kombinira s drugim sigurnosnim zahtjevima koji su navedeni.

Još jedan sličan primjer predstavlja pitanje na koliko snažnim zaporkama treba inzistirati? Većina tvrtki danas zadovoljit će se zaporkama od osam znakova, koje se sastoje od malih i velikih slova, brojki te jednog posebnog znaka. No nerijetko je moguće susresti se i sa slučajevima kada su administratori, bez propisane sigurnosne politike, sami konfigurirali zahtjeve za snagom zaporki te su, u želji osiguravanja samog informacijskog sustava, postavili nerealne zahtjeve na snagu zaporki.

Ovo je primjer još jedne tehničke mjere zaštite koja može imati kontraproduktivne učinke jer će korisnici opet početi zapisivati zaporke na papir. Zbog toga je i ovaj zahtjev potrebno detaljno analizirati te donijeti sigurnosnu politiku koja će definirati snagu zaporki na kojoj će zatim biti temeljene sve tehničke konfiguracije sustava.

Kada se pišu sigurnosne politike, potrebno je imati na umu da je riječ o dokumentima koji bi trebali biti dugo na snazi te se relativno rijetko mijenjati. Tako npr. sigurnosna politika ne bi trebala doticati tehnička rješenja već samo navoditi generalne preporuke i zahtjeve, dok će pojedine sigurnosne procedure i drugi dokumenti opisivati tehnička rješenja.

Jednako tako, tijekom vremena se pokazalo da je nepraktično sve zahtjeve držati u jednom dokumentu, već je puno kvalitetnije imati niz specijaliziranih dokumenata koje je onda u skladu s time moguće posebno održavati (po potrebi). Osim toga, kategoriziranjem sigurnosnih procedura i politika moguće ih je grupirati prema ulogama korisnika – svi zaposlenici tvrtke morat će biti upoznati i potpisati da su pročitali određene sigurnosne politike, dok će oni zaposlenici koji imaju veća prava pristupa (npr. administratori informacijskog sustava) morati biti upoznati s dodatnim sigurnosnim politikama.



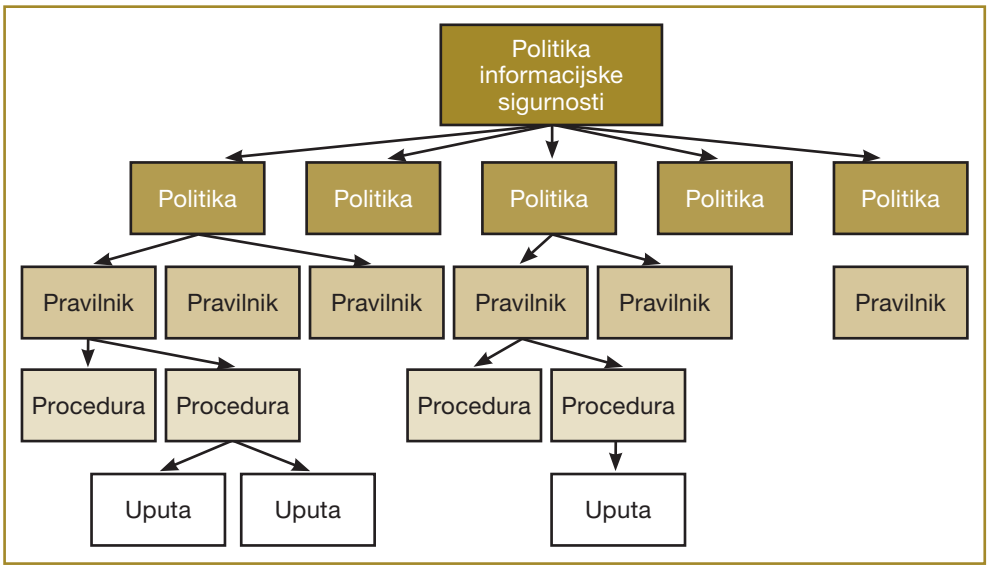
Slika 7.1. Životni ciklus sigurnosne politike

7.1. Hijerarhija sigurnosnih politika

Kao što je već rečeno, u svrhu kvalitetnijeg održavanja dokumentacije preporučuje se napraviti hijerarhija sigurnosnih politika. Na vrhu se nalazi glavna sigurnosna politika tvrtke – riječ je o vrlo kratkom dokumentu, ne duljem od jedne

ili dvije strane koji samo potvrđuje potporu uprave u uvođenju informacijske sigurnosti u tvrtku te dužnosti svakog zaposlenika u poštovanju iste.

Nakon navedene krovne sigurnosne politike slijedi čitav niz drugih sigurnosnih politika, iz kojih se grade pravilnici, procedure i upute, kao što je prikazano na sljedećoj slici.



Slika 7.2. Hijerarhija sigurnosnih politika

Sigurnosne politike definiraju zahtjeve koji se postavljaju pred pojedina područja informacijske sigurnosti. Neki primjeri sigurnosnih politika koje se najčešće definiraju u tvrtkama navedeni su u nastavku.

- Politika primjerene upotrebe informacijskog sustava definira koja su prava i obaveze svih korisnika informacijskog sustava. Ovdje je potrebno obuhvatiti sve zaposlenike te definirati što korisnici smiju, a što ne smiju raditi u informacijskom sustavu (na generičkoj razini). Ova politika također treba definirati i sankcije u slučaju kršenja navedenih zahtjeva.
- Politika klasifikacije informacija definira kako sve informacije pohranjene u informacijski sustav moraju biti klasificirane. Na osnovi klasifikacije informacija moguće je odrediti rukovanje njima – npr. politika može odrediti da se sve informacije odnosno dokumenti koji su označeni klasifikacijom „tajno“ ne smiju slati računalnom mrežom u čistom tekstualnom obliku, odnosno da moraju biti enkriptirani.

- Politika kontrole pristupa definira procese odobravanja i dodjele prava pristupa, reviziju te uklanjanje prava pristupa. Ova politika, u skladu s politikom klasifikacije informacija, definira tko čemu smije pristupiti.

Pravilnici, kao što im kaže i samo ime, definiraju pravila vezana za određeni dio informacijskog sustava. Za razliku od sigurnosnih politika, koje daju samo smjernice te postavljaju strategiju implementacije informacijske sigurnosti, pravilnici konkretno definiraju zahtjeve koji se postavljaju u procesima te kakve se sigurnosne kontrole moraju uspostaviti u navedenom informacijskom sustavu.

Primjer razlike između pravilnika i sigurnosne politike najbolje se može prikazati na sigurnosnoj politici kontrole pristupa, odnosno pravilniku upravljanja pristupom korisnika. Sigurnosna politika tako samo zahtijeva da korisnici imaju jedinstvene, snažne zaporke te da zaporke predstavljaju njihovu odgovornost. Pravilnik, s druge strane, definira da se sigurnim zaporkama smatraju zaporke koje su dugačke osam znakova ili više te imaju barem jedno malo slovo, veliko slovo, brojku i poseban znak.

Na ovaj se način postiže trajnost sigurnosnih politika te omogućuje osvježavanje pravilnika. Drugim riječima, sigurnosne je politike potrebno pisati općenito tako da vrijede kroz dulje razdoblje. Pravilnici bi također trebali vrijediti što je moguće dulje, no ako dođe do promjena prijetnji zbog npr. neke nove sigurnosne ranjivosti ili, u navedenom primjeru sa zaporkama, nekog novog načina pogađanja zaporki koji je iznimno učinkovit, pravilnik je moguće osvježavati po potrebi.

Procedure su još detaljniji dokumenti u kojima su navedene i tehničke informacije. Cilj procedura je opisati korake potrebne za ispunjavanje određenog zahtjeva definiranog sigurnosnom politikom, odnosno pravilnikom. U navedenom bi primjeru postavljanja snažne zaporke procedura sadržavala informacije zaposlenicima kako mogu postaviti ovakvu zaporku – npr. na Windows operacijskim sustavima zaposlenik treba pritisnuti CTRL+ALT+DEL tipke te odabrati opciju *Change a password* nakon čega treba upisati snažnu zaporku koja je u skladu sa zahtjevima definiranim u pravilniku, odnosno sigurnosnoj politici.

Konačno, upute se primjenjuju kada je neke korake definirane procedurama potrebno još detaljnije opisati. Primjerice, upute mogu sadržavati pojedine korake koje je potrebno provesti pri izvršavanju procedure, primjere i dodatne informacije kao i popis alata koji služe za izvršavanje procedure.

Navedena hijerarhija dokumenata koji čine sigurnosnu politiku osnova je sigurnosti informacijskog sustava na kojoj se sve daljnje aktivnosti zasnivaju. Pri izradi sigurnosnih politika danas se uglavnom primjenjuju dva pristupa:

- pristup od vrha prema dnu (engl. *Top-down approach*) – temelji se na zakonskim i regulatornim zahtjevima. Kod ovog se pristupa prvo razmatraju navedeni zahtjevi koji se postavljaju pred tvrtku te se nakon toga izrađuju sigurnosna politika, pravilnici i procedure koji zadovoljavaju navedene zakonske i regulatorne zahtjeve.

  Problem ovakvog pristupa obično je taj da se napravljene sigurnosne politike, pravilnici i procedure vrlo teško provode u praksi. Naime, budući da su ovi dokumenti napisani isključivo prema zakonskim i regulatornim zahtjevima, vrlo se lako može dogoditi da nisu primjenjivi ili da znatno otežavaju poslovanje u već postojećim informacijskim sustavima;
- pristup od dna prema vrhu (engl. *Bottom-up approach*) – suprotan je pristupu od vrha prema dnu. U ovom se slučaju pisanje sigurnosnih politika, pravilnika i procedura temelji na ulaznim podacima dobivenim od administratora, zaposlenika tvrtke ili analizom sigurnosnih incidenata koji su se dogodili.

  Problem navedenog pristupa je taj da je vrlo usko vezan za postojeći informacijski sustav. Sigurnosne politike, pravilnici i procedure koje su napisane primjenom pristupa od dna prema vrhu vrlo često imaju problem neadekvatnosti čak i kod najmanjih promjena informacijskog sustava.

Kao što se može pretpostaviti, najbolji način pristupa izradi sigurnosnih politika, pravilnika i procedura je upravo kombinacija oba navedena pristupa. Pri izradi je potrebno uzeti u obzir zakonske i regulatorne zahtjeve, budući da u konačnici tvrtka mora biti usklađena s njima, no istodobno je potrebno analizirati i trenutne poslovne procese, provesti intervjue s administratorima i zaposlenicima te konačno analizirati bilo kakve sigurnosne incidente koji su se možebitno dogodili u tvrtci.

Na ovaj će način biti moguće napraviti najrealniju sigurnosnu politiku koja će zadovoljiti zakonske i regulatorne zahtjeve te će istodobno biti primjenjiva u praksi, a neće ostati samo „mrtvo slovo na papiru".

Konačno, za zaposlenike je potrebno redovito organizirati sigurnosno osvješćivanje. Riječ je o procesu u kojem se zaposlenike upozorava na sigurnosne politike te opisuju prijetnje kojima su i sami izloženi. Sigurnosno osvješćivanje organizira se ovisno o grupama zaposlenika (npr. zaposlenici koji su izloženi većem sigurnosnom riziku ili rukuju s naročito osjetljivim podacima trebaju imati posebno prilagođeno sigurnosno osvješćivanje) te ga je potrebno provesti za

sve zaposlenike, uključujući i vodstvo tvrtke. Na ovaj se način povisuje sveukupna razina sigurnosne osviještenosti tvrtke što u konačnici rezultira i većom razinom sigurnosti samog informacijskog sustava, budući da je praksa pokazala da čak i kratki seminari sigurnosnog osvješćivanja mogu znatno utjecati na aktivnosti zaposlenika i konačnu sigurnost informacijskog sustava.

7.2. Socijalni inženjering

Socijalni inženjering je posebna kategorija napada koja se ne temelji na nikakvim tehničkim sigurnosnim ranjivostima već na mogućnosti prijevare, odnosno manipuliranja ljudima (zaposlenicima tvrtke) u svrhu provođenja aktivnosti koje napadač želi provesti.

Kada je riječ o napadima socijalnim inženjeringom, zapravo se najčešće govori upravo o nekim vrstama prijevara koje su, međutim, zasnovane na dobro osmišljenim scenarijima kako bi se zaposlenika koji ima pristup nekim informacijama navelo da npr. te informacije svojevoljno preda napadaču. Ključna značajka napada socijalnim inženjeringom je upravo ta da zaposlenik (žrtva) svojevoljno provodi neku aktivnost – napadač ga je dakle uvjerio da je riječ o benignoj aktivnosti koju zaposlenik treba ispuniti zbog ovog ili onog razloga (neki primjeri scenarija napada socijalnim inženjeringom koji su bili provedeni u praksi dani su u nastavku poglavlja).

7.2.1. Napadi Kevina Mitnicka

Jedan od najpoznatijih napadača, prema čijim je aktivnostima djelomično i zasnovan pojam socijalni inženjering, svakako je Kevin Mitnick. Riječ je o čovjeku čiji je cijeli život bio isprepleten napadima socijalnim inženjeringom na velike informatičke tvrtke. Još kada je imao 12 godina, Kevin Mitnick sprijateljio se s vozačem autobusa u Los Angelesu koji mu je otkrio kako može doći do aparata za bušenje karata kakve su upotrebljavali kondukteri. To mu je omogućilo da se godinama besplatno vozi javnim prijevozom u Los Angelesu. No to je bio samo početak napada socijalnim inženjeringom koje je Mitnick provodio.

Iako se Mitnick služio i tehničkim sigurnosnim ranjivostima tijekom provođenja napada, glavna metoda njegovih napada ipak je bila socijalni inženjering. Jedan od najvećih napada, zbog kojeg je Mitnick kasnije i završio u zatvoru, bio je na Digital Equipment Corporation (DEC), veliku informatičku tvrtku 1979. Mitnickov napad na ovu tvrtku sastojao se od cijelog niza pažljivo isplaniranih koraka. Mitnick je tako prvo saznao za jednog stručnjaka za računalnu sigurnost koji je imao čitav niz još neobjavljenih ranjivosti za DEC operacijski sustav. Budući

da je znao da se navedeni sigurnosni stručnjak zapravo želi zaposliti u DEC-u, Mitnick ga je nazvao i predstavio se kao zaposlenik kadrovske službe DEC-a te sigurnosnom stručnjaku ponudio niz intervjua s ciljem možebitnog zaposlenja DEC-a. U ovom se napadu socijalnog inženjeringa Mitnick nije libio ni fizičkog kontakta s žrtvom (većina napada socijalnog inženjeringa za komunikaciju upotrebljava telefon, elektroničku poštu ili slične medije).

Nakon nekog je vremena sigurnosni stručnjak Mitnicku svojevoljno dao informacije o sigurnosnim ranjivostima koje je napravio, što je, kao što je već ranije u poglavlju spomenuto, osnovni cilj socijalnog inženjeringa. Sigurnosni je stručnjak navedene informacije dao Mitnicku jer je mislio da mu to pomaže u možebitnom dobivanju posla. Mitnick je dobivene informacije iskoristio kako bi napadao DEC-ovu računalnu mrežu i došao do povjerljivih informacija o novim proizvodima.

Tijekom narednih godina Mitnick je proveo čitav niz napada socijalnim inženjeringom. Još jedan od poznatih napada bio je na Motorolu, gdje je Mitnick putem napada socijalnim inženjeringom provedenog preko telefona uspio navesti jednog od voditelja razvojnih timova u Motoroli da mu preko interneta pošalje izvorni kod programskog paketa koji se rabio za upravljanje Motorolinim mobilnim telefonima.

Kao što se može vidjeti iz navedenih primjera, napadi socijalnim inženjeringom, ako su dobro planirani, mogu biti iznimno opasni. Ono što je najbitnije napomenuti kod ovih napada jest činjenica da ih klasični tehnički sigurnosni mehanizmi zapravo ne mogu spriječiti: u svakom su prikazanom scenariju napadaču osjetljive podatke svojevoljno predali zaposlenici tvrtke koji imaju legitimni i autorizirani pristup navedenim osjetljivim podacima!

7.2.2. Napadi lažnim antivirusnim programima

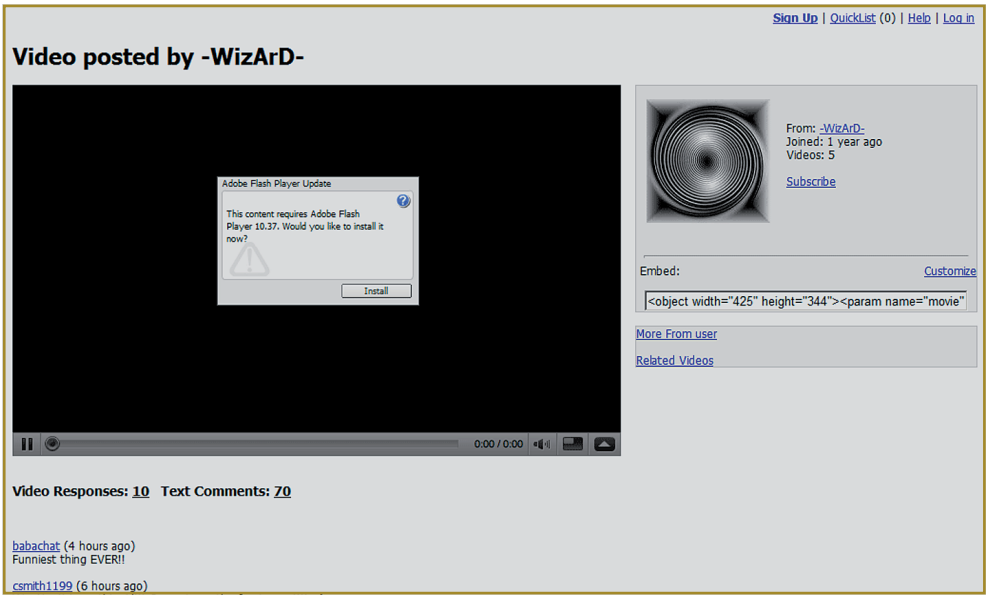
Jedan od današnjih najpoznatijih primjera napada socijalnog inženjeringa na korisnike putem web-stranica na internetu svakako su lažni antivirusni programi. Riječ je o napadima koji su se prvi put pojavili tijekom 2007. da bi danas predstavljali gotovo apsolutnu većinu napada socijalnim inženjeringom na internetu.

Scenarij iza ovog napada slijedi klasični scenarij napada socijalnim inženjeringom: korisnika koji ništa ne sumnja nastoji se navesti da provede aktivnost koju napadač želi provesti. U ovom je slučaju ta aktivnost instaliranje lažnog antivirusnog programa na računalo korisnika što kasnije napadaču omogućuje kontrolu tog (inficiranog) računala ili čak navođenje korisnika na plaćanje ovog lažnog antivirusnog programa što napadaču daje izravnu mogućnost zarade.

Napad socijalnim inženjeringom u ovom je slučaju malo drukčije izveden. Putem posebno napravljenih web-stranica korisnika se želi prestrašiti i uvjeriti da je njegovo računalo već inficirano nekim malicioznim programom. Kao rješenje korisniku se pruža instalacija lažnog antivirusnog programa. Korisnik koji ne sluti da je riječ o napadu socijalnog inženjeringa tako će svojevoljno instalirati navedeni lažni antivirusni program, koji je zapravo sam maliciozan.

Na ovaj način zapravo funkcionira i veliki broj trojanskih konja koji su također zasnovani na napadu socijalnim inženjeringom: na određeni se način, putem određenog scenarija, korisnika želi uvjeriti da je potrebno instalirati ponuđeni program da bi uspio napraviti ono u što ga se uvjerilo: zaštititi svoje računalo, instalirati program koji mu omogućuje gledanje videozapise ili nešto treće. Nekoliko primjera ovakvih napada dano je u nastavku poglavlja.

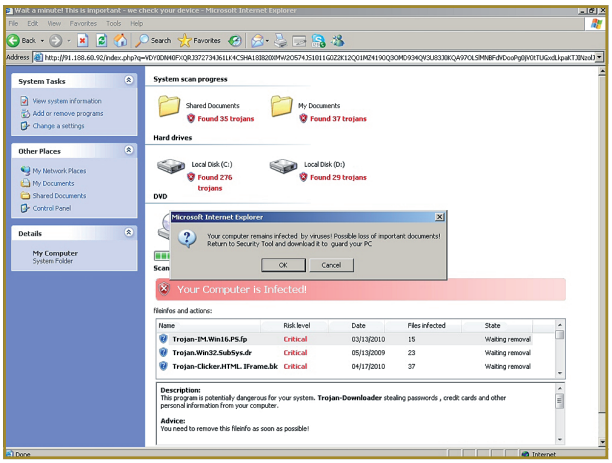
Sljedeća slika predstavlja upravo navedeni napad socijalnim inženjeringom na korisnika kojeg se na web-stranicu dovelo nuđenjem pregledavanja videozapisa. Kao što se može vidjeti iz priloga, videozapis se ne može vidjeti, a korisniku se sugerira da treba instalirati ponuđeni pogonski program za navedeni format videozapisa kako bi ga mogao vidjeti.



Slika 7.3. Socijalni inženjering putem videozapisa

Sljedeće dvije slike prikazuju poznati napad socijalnim inženjeringom koji se služi lažnim antivirusnim programom. Kao što se može vidjeti, web-stranice napadača vrlo vjerno simuliraju izgled legitimnih prozora Windows i Macintosh

operacijskih sustava čime se postiže vjerodostojnost napisanih informacija. Cilj ovakvog napada je navođenje korisnika na pokretanje i instaliranje lažnog antivirusnog programa.



Slika 7.4. Socijalni inženjering lažnog antivirusnog programa na korisnike Windows OS-a

Slika 7.5. Socijalni inženjering lažnog antivirusnog programa na korisnike Mac OS-a

Mogućnosti provođenja napada socijalnim inženjeringom praktički su beskonačne. Budući da uspješnost ovog napada zapravo ovisi o krajnjem korisniku, čovjeku, najbolji način sprječavanja provođenja ovakvih napada je u implementaciji višeslojnog strateškog modela sigurnosti, kao što je navedeno u poglavlju 1.5.

Ispravne sigurnosne kontrole potrebno je implementirati na svim slojevima, a posebnu je pažnju potrebno posvetiti krajnjim korisnicima sustava, zaposlenicima tvrtke. Zbog toga je potrebno redovito održavati već spomenuto sigurnosno osvještavanje zaposlenika kako bi im se približile sve prijetnje i sigurnosni rizik koji može prouzročiti bilo koja njihova aktivnost.

8. Sigurnost mobilnih uređaja

Današnji mobilni uređaji uvelike su promijenili način korištenja informacijskim resursima te otvorili nove mogućnosti za elektroničko poslovanje. Zbog svojih iznimnih mogućnosti koje su se razvile u vrlo kratkom razdoblju, današnji se mobilni uređaji vrlo često upotrebljavaju u praktički iste svrhe kao i osobna ili prijenosna računala. U skladu s time, prijetnje i sigurnosni rizik upotrebe mobilnih uređaja vrlo su slični sigurnosnom riziku upotrebe prijenosnih računala, a zbog nekih drugih značajki koje su navedene u nastavku te nedostatka percepcije o sigurnosnom riziku korisnika još su i veći.

Budući da mobilni uređaji mogu pohranjivati velike količine osobnih, ali i poslovnih povjerljivih podataka, potencijalno su zanimljiv cilj napadačima. Ovo je još više potkrijepljeno činjenicom da mobilni uređaji, iako mogu imati pohranjene vrlo osjetljive podatke, još uvijek nemaju implementirane sigurnosne kontrole slične onima na osobnim računalima. Jednako tako, administratori informacijskog sustava danas još uvijek imaju malu ili nikakvu kontrolu nad mobilnim uređajima što još više povećava rizik njihove upotrebe te dozvole pristupanja informacijskom sustavu tvrtke.

Mobilni su uređaji danas postali sastavnim dijelom života, a International Telecommunication Union procijenio je da je u svijetu krajem 2009. bilo preko 4.5 milijardi mobilnih uređaja. Uz već spomenute znatne računalne resurse koje ovi uređaji pružaju, počevši od procesorske snage, memorije te mogućnosti pohranjivanja velike količine podataka, današnji mobilni uređaji mogu komunicirati s drugim uređajima primjenom različitih protokola: od mobilnih podatkovnih protokola, preko bežičnih računalnih mreža pa do Bluetooth računalnih mreža.

Kao što se može vidjeti, kompleksnost mobilnih uređaja dostigla je, pa i prestigla kompleksnost prosječnog osobnog računala. U skladu s rastom kompleksnosti ovih uređaja jasno je da raste i sigurnosni rizik te mogućnost potencijalnih sigurnosnih ranjivosti. Kako danas mobilne uređaje zapravo pokreću mobilne inačice već poznatih operacijskih sustava (npr. Windows Mobile koji je zasnovan na Windows operacijskom sustavu, Appleov iOS koji se temelji na BSD operacijskom sustava te Google Android koji je zasnovan na Linux operacijskom sustavu), jasno je da sigurnosne prijetnje koje vrijede za osobna računala danas vrijede i za mobilne uređaje. Navedene sigurnosne prijetnje uključuju sigurnosne ranjivosti u instaliranim aplikacijama, maliciozne programe, napade koji primjenjuju socijalni inženjering i slično.

Povijesno gledano, maliciozni programi za mobilne uređaje nisu predstavljali znatnu prijetnju. Prvenstveni razlog za ovo bio je u velikoj različitosti ovih uređaja te nepostojanju kritične mase koja bi opravdala ulaganje u razvoj napadača pa su maliciozni programi za starije generacije mobilnih uređaja uglavnom predstavljale samo prototipove.

Rastom popularnosti mobilnih uređaja, ujednačavanjem zastupljenih platformi te rastom mogućnosti ovih uređaja (njihove procesorske snage, memorije i slično) stvorila se kritična masa ovih uređaja. U skladu s time, početkom 2011. počeo se primjećivati sve veći broj malicioznih programa za mobilne uređaje koji prednjače za mobilne uređaje koji upotrebljavaju otvorene operacijske sustave poput Google Androida. Za navedeni je operacijski sustav identificiran iznimno veliki broj trojanskih konja koji na različiti način oštećuju korisnika. Nakon inficiranja mobilnog uređaja primjenom jedne od spomenutih metoda (npr. napad socijalnim inženjeringom gdje korisnik svojevoljno instalira trojanski konj na mobilni uređaj misleći da je riječ o legitimnom programu) maliciozni program ima potpunu kontrolu nad mobilnim uređajem (naravno, ako je korisnik tijekom instalacije dopustio takvu kontrolu) te može napraviti čitav niz aktivnosti.

- Vrlo čestu pojavu predstavljaju trojanski konji za mobilne uređaje automatski šalju SMS uređaje ili uspostavljaju telefonske pozive na vrlo skupe destinacije izravno oštećujući korisnika. Identičan tip malicioznih programa koji su uspostavljali telefonske pozive na skupe destinacije već je godinama prisutan na osobnim računalima – riječ je o tzv. engl. *dialer* kategoriji programa koji koriste modemsku vezu za uspostavljanje telefonskih poziva.

  Napadač može koristiti ove mogućnosti da korisnika inficiranog mobilnog uređaja pretplati na skupe servise (npr. slanjem SMS poruke na određeni broj), što napadaču omogućuje ostvarivanje zarade.

  Ovakvi trojanski konji gotovo se uvijek instaliraju kroz napade socijalnim inženjeringom. Jedan od razloga je i činjenica da operacijski sustavi u mobilnim uređajima automatski blokiraju aktivnosti instaliranih aplikacija koje mogu utjecati na sigurnost – npr. slanje i čitanje SMS poruka i adresara, uspostavljanje telefonskih poziva, čitanje podataka s telefona i slično. No, kao što se može vidjeti iz primjera u poglavlju 7.2.2, napadači danas vrlo uspješno koriste metode socijalnog inženjeringa ne bi li korisnike naveli na dopuštanje navedenih aktivnosti.
- Krađa osobnih podataka danas predstavlja vrlo čest napad na mobilne uređaje koji pohranjuju veliku količinu osobnih podataka korisnika, ali i

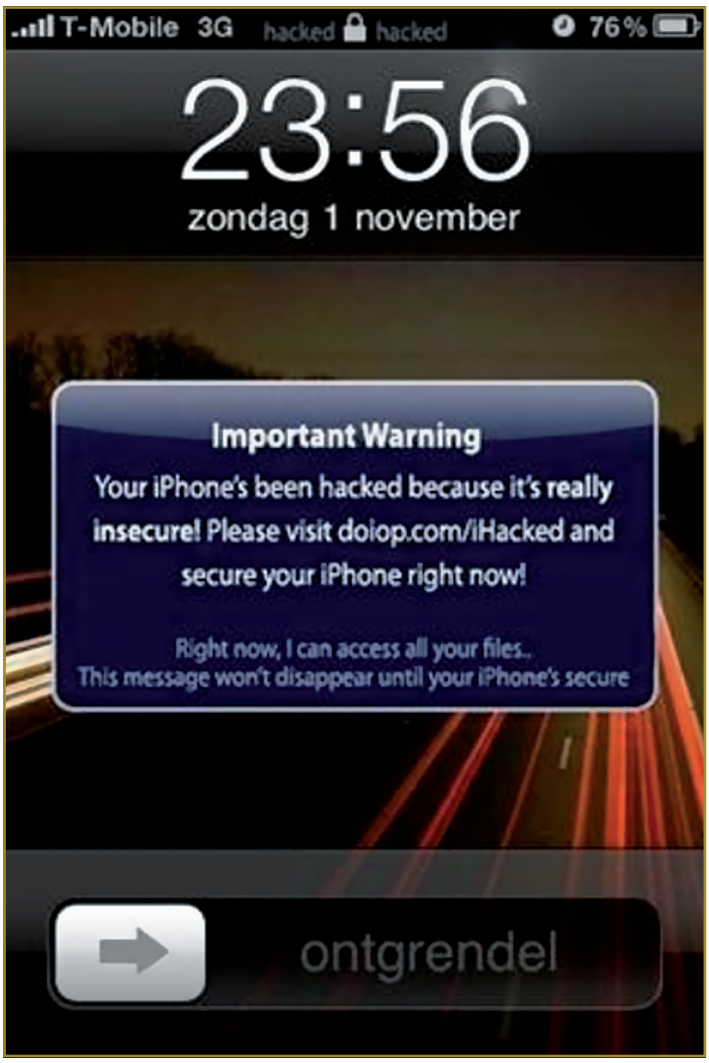
poslovnih informacija pogotovo ako se mobilni uređaji upotrebljavaju za pristup elektroničkoj pošti.

- Udaljeno upravljanje inficiranim mobilnim uređajem napadaču omogućuje provođenje proizvoljnih daljnjih napada, pogotovo ako je mobilni uređaj spojen na bežičnu računalnu mrežu tvrtke. Potrebno je naglasiti da se današnji mobilni uređaji za komunikaciju koriste IP računalnim mrežama te su kao takvi također ekvivalentni osobnim računalima.
- Konačno, posebnu je pažnju potrebno posvetiti mobilnim aplikacijama. Iako je u osnovi riječ o klasičnim aplikacijama, te su sigurnosne ranjivosti i prijetnje zapravo jednake aplikacijama na osobnim računalima, a zbog specifičnosti funkcioniranja mobilnih uređaja postoje i dodatne kategorije sigurnosnih ranjivosti koje vrijede samo za mobilne uređaje.

Još jedan od problema koje je potrebno navesti jest i činjenica da korisnik ima praktički potpunu kontrolu nad mobilnim uređajem. Kako su sigurnosne kontrole koje su implementirane na ove uređaje još uvijek ograničene na one implementirane na osobnim računalima, administratori ne mogu uspostaviti jednaka prava pristupa i sigurnosne mehanizme na mobilnim uređajima i osobnim računalima.

Kao primjer sigurnosnog rizika koji proizlazi iz potpune kontrole korisnika nad mobilnim uređajem može se navesti postupak mijenjanja operacijskog sustava mobilnog uređaja (tzv. engl. *jail breaking*). Promjenom operacijskog sustava, odnosno instaliranjem modificirane inačice operacijskog sustava na mobilni uređaj korisnik može praktički neograničeno instalirati aplikacije svoj uređaj te tako može zaobići sve sigurnosne kontrole, čak i one koje su primijenjene na legitimnom operacijskom sustavu mobilnog uređaja. Ovaj je postupak svojstven za iPhone mobilne uređaje gdje korisnici instaliraju modificirani operacijski sustav kako bi bili u stanju instalirati aplikacije bez ograničenja.

Osim što instaliranje modificiranog operacijskog sustava također predstavlja sigurnosni rizik, dodatne aplikacije ovaj rizik uvelike povećavaju. Tako je krajem 2009. zabilježen jedan od prvih crva za iPhone mobilne uređaje koji je automatski inficirao korisnike ovih mobilnih uređaja koji su ostavili otvoreni SSH servis na uređaju (kao što je već rečeno, uređajima se može jednostavno pristupiti preko IP računalne mreže, čak i kada su samo spojeni preko mobilne podatkovne komunikacijske mreže). Crv je automatski inficirao ranjive iPhone mobilne uređaje te je korisniku prikazivao poruku koja od njega zahtijeva plaćanje 5 USD (riječ je malicioznom programu koji iznuđuje novce, tzv. engl. *ransom ware*). Sljedeća slika prikazuje inficirani iPhone mobilni uređaj navedenim crvom.



Slika 8.1. Inficirani iPhone mobilni uređaj

Bez obzira na sigurnosne ranjivosti koje su inherentne i mobilnim uređajima, danas su vjerojatno najveći sigurnosni rizik neznanje i nesvjesnost korisnika ovih uređaja o sigurnosnim rizicima istih. Metode podizanja razine sigurnosti mobilnih računala vrlo su slične onima osobnih računala te uključuju sljedeće:

- instaliranje svih dostupnih sigurnosnih zakrpi za operacijske sustave mobilnih uređaja kao i mobilnih aplikacija;
- instaliranje antivirusnih aplikacija ako one postoje za navedene mobilne uređaje. Nažalost, danas još uvijek antivirusne aplikacije nisu dostupne za čitav niz mobilnih uređaja (npr. iPhone) koji su kao takvi automatski

izloženi većem sigurnosnom riziku bez obzira na druge sigurnosne kontrole koje je proizvođač implementirao, poput npr. ograničavanja instaliranja aplikacija samo iz certificiranih izvora (AppStore za iPhone uređaje);
- enkripciju svih pohranjenih podataka na mobilnom uređaju, ako mobilni uređaj to podržava;
- automatsko zaključavanje radne površine mobilnog uređaja te autentikaciju korisnika PIN-om;
- provođenje postupaka ojačanja mobilnog uređaja slično postupku opisanom u poglavlju 5.2, što uključuje isključivanje automatskog spajanja na bežične računalne mreže, isključivanje vidljivosti pri upotrebi Bluetooth mreža i slično.

Provođenje sigurnosnog osvješćivanja korisnika u svrhu podizanja svijesti o sigurnosnim ranjivostima mobilnih uređaja ključno je za podizanje cjelokupne razine sigurnosti. Korisnici ovih uređaja moraju biti svjesni da mobilni uređaji ne predstavljaju više samo telefone već su zapravo određena vrsta prijenosnih računala.