CARNet
HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Metode za povećanje sigurnosti operacijskog sustava Linux

CCERT-PUBDOC-2009-09-277

+CERT.hr u suradnji s LSS

Sigurnosni problemi u računalnim programima i operacijskim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada je i ovaj dokument, koji je nastao suradnjom CARNet CERT–a i LS&S–a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr

Nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.LSS.hr

Laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

Operacijski sustav Linux razlikuje se od Windowsa po tome što administrator ima znatno veće mogućnosti prilagodbe sustava prema specifičnim potrebama primjene (osobno računalo, poslovni poslužitelj, poslužitelj za industrijska okruženja itd.). To istovremeno znači veću slobodu korištenja i veću odgovornost. Pritom se posebna pažnja treba posvetiti zaštiti sustava. Nju je potrebno uvesti na različitim razinama: od fizičke sigurnosti i zaštite kod podizanja sustava do praćenja rada mrežnih programa i usluga.

Najčešće opasnosti prijete iz nesigurne računalne mreže, zato je posebno važno osigurati (enkripcija, autentifikacija) i filtrirati (vatrozid) mrežnu komunikaciju. Jednom kada napadač uspješno izvede napad i stekne pristup računalu putem lokalnog korisničkog računa, može pokušati steći veću razinu ovlasti (npr. administratorske ovlasti) kako bi pristupio osjetljivim i zaštićenim informacijama. Zbog toga je potrebno ograničiti prava pristupa i dozvoljena ponašanja lokalnih korisnika. Još jedna mjera zaštite je redovito praćenje događaja u sustavu, pomoću u tu svrhu razvijenih alata, čime se omogućuje pravovremeno otkrivanje pokušaja napada i provođenje potrebnih radnji.

Uz standardnu zaštitu na Linux sustavima moguće je provesti specifične postupke ojačavanja sustava i jezgre (ručno i programskom nadgradnjom). Jednom kad je zaštita uspostavljena, može se ispitati odgovarajućim alatima (često istim onim koje koriste napadači).

Ovaj dokument oblikovan je s ciljem davanja općenitog uvida u sigurnosna pitanja Linux sustava i dostupne načine njihovih rješavanja. Predložene metode uključuju različite sigurnosne protokole i njihove besplatne izvedbe, ugrađene Linux sigurnosne mehanizme te dostupnu sigurnosnu nadogradnju.

2. Linux

Operacijski sustav Linux zasnovan je na sustavu Unix čija je jezgra oblikovana po uzoru na Solaris OS. Jezgru Linux sustava razvio je Linus Torvalds, a izvorni kod objavljen je na Internetu 1991. godine. Od tada se Linux slobodno razvija i distribuira kao besplatan sustav pod GPL licencom. Češće ga odabiru napredniji korisnici zato što pruža mnogo više mogućnosti prilagođavanja specifičnim potrebama korištenja od Windows sustava. Uz to, alati za Linux su najvećim dijelom besplatno dostupni. S druge strane, rad s Linuxom zahtijeva bolje poznavanje računala od rada s Windowsima, što ponovno dovodi do njegove veće popularnosti među naprednijim korisnicima.

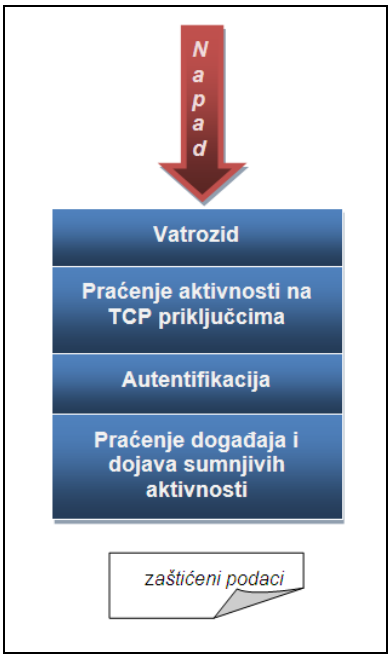
Briga i održavanje Linux OS-a između ostalog uključuju brigu o sigurnosti. Ovaj dokument daje opći pregled sigurnosnih mehanizama i elemenata o kojima valja voditi računa na Linux sustavima, a navode se i dostupni protokoli i alati koji doprinose sigurnosti.

2.1. Motivacija i pristupi zaštiti OS-a

Ukoliko računalo ima više korisnika ili je fizički dostupno nepouzdanim osobama, postoji značajna opasnost od napada. Isto vrijedi i za računala koja se spajaju na nesigurnu mrežu izravno ili preko zaštićene lokalne mreže. Neoprezno ponašanje korisnika može dovesti do iskorištavanja njihovih korisničkih računa kako bi se stekao pristup osjetljivim informacijama pohranjenim u sustavu. Pritom veću prijetnju predstavljaju korisnički računi s većim ovlastima. Logičan je zaključak tada da korisnicima ne treba davati veće ovlasti od onih koje su im nužne za obavljanje definiranih zadataka. Nadalje, opasnosti prijete i od mrežnih programa i usluga čije se eventualne ranjivosti mogu iskoristiti za narušavanje sigurnosti sustava. Primjerice, nepravilne izvedbe protokola mogu dovesti do DoS (eng. Denial of Service) stanja, a neodgovarajuća zaštita lozinki do njihovog otkrivanja i stjecanja neovlaštenog pristupa. Korisnici često puta nisu niti svjesni koliko opasnosti prijeti njihovu računalu, odnosno podacima na njemu jer oni su konačni cilj napada. Kod zaštite podataka u prvom redu važna je edukacija korisnika o odgovornom ponašanju. Osim toga, potrebno je uvesti odgovarajuće programske sigurnosne mehanizme koji će automatski i redovito poduzimati potrebne akcije kako bi se rizici i opasnosti smanjili na najmanju moguću mjeru.

Jedan od pristupa koji se koriste u uspostavljanju računalne zaštite je tzv. „*Defense In Depth*“. Riječ je o pristupu koji je razvila američka agencija NSA (eng. National Security Agency) po uzoru na vojnu strategiju koja odgađa obranu i napad, ostavljajući djelomično prostor napadaču. Pritom se prati ponašanje napadača te dobivaju informacije i vrijeme potrebni za organiziranje obrane. U računarstvu ovo uključuje pristup zaštiti koji ne zatvara potpuno sustav, već dopušta pristup do određene razine pri čemu se akcije mogućeg napadača pažljivo bilježe i prate. Na taj način, dok napadač iskušava sustav, oni zaduženi za njegovu zaštitu imaju vremena otkriti pokušaje napada i poduzeti potrebne zaštitne mjere. Osim toga, *Defense In Depth* znači zaštitu na više razina računalnog sustava: od fizičke sigurnosti, preko autentifikacijskih mehanizama do antivirusnih alata, IDS (eng. Intrusion Detection Systems) sustava i sustava za praćenje korisničkih aktivnosti i događaja (eng. logging and auditing). Primjer sigurnosnih mehanizama koje može uključivati dan je na slici 1.

Osim opisanog pristupa, zaštiti sigurnosti može se doprinijeti minimalističkim pristupom dodjeli ovlasti (eng. least privilege). Radi se o tome da se svakom korisniku, programu ili procesu dodjeljuju samo one ovlasti koje su nužne za njegovo legitimno djelovanje. Na taj način izbjegava se neopravdano stvaranje prostora za narušavanje sigurnosti sustava. Valja napomenuti kako nije u svakoj situaciji moguće savršeno procijeniti najmanje potrebne ovlasti. Ponašanje računalnih programa je općenito nepredvidivo pa su i zahtjevi koje će on postaviti sustavu nepredvidivi. Zato uporaba ovakvog pristupa zahtijeva dovoljno stručnog znanja pomoću kojeg se može odrediti koliko duboko u razvoju sigurnosne strategije ima smisla ići.



Slika 1. Primjer višeslojne zaštite

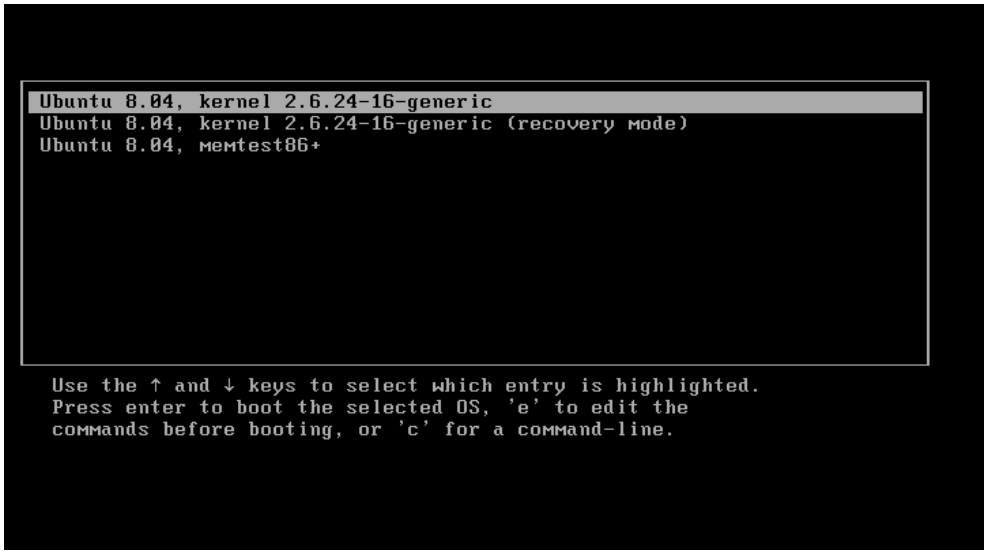
2.2. Osnovni zaštitni mehanizmi

U ovom poglavlju razmotrit će se osnovni mehanizmi zaštite Linux sustava koje treba poznavati svaki korisnik. Oni uključuju zaštitu datoteka i datotečnog sustava, sigurno upravljanje korisničkim računima, zaštitu od neovlaštenog podizanja sustava i slične mehanizme.

2.2.1. Podizanje sustava

Fizička sigurnost računala uključuje sigurnosna ograničenja kod podizanja operacijskog sustava na njemu. Ukoliko uspije steći dovoljnu razinu pristupa, napadač može pokrenuti ponovno podizanje sustava (eng. boot/reboot) s vanjskih uređaja (disketa, CD, USB) i uspostavljanje vlastitog operacijskog sustava što uključuje i administratorski račun. Kako bi spriječili ili bar otežali ovu vrstu napada preporuča se koristiti tzv. „boot loader“ alate koji omogućuju zaštitu podizanja sustava lozinkom. Primjerice LILO (eng. Linux Loader) alat omogućuje postavljanje zahtjeva za lozinkom prilikom podizanja sustava. Pritom je potrebno prikladno zaštiti konfiguracijsku datoteku ovog alata kako ju neovlašteni korisnici ne bi mogli čitati jer ona sadrži lozinku. No zaštita samog procesa podizanja sustava lozinkom ne otklanja mogućnost ponovnog podizanja sustava preko vanjskog uređaja.

Drugi alat za podizanje sustava je GNU Grub. Pomoću njega se može onemogućiti pokretanje *reboot* postupka bez poznavanja lozinke. Ona se pak nalazi u konfiguracijskoj datoteci, no moguće je (i preporučljivo), pohraniti ju u sažetom kriptiranom obliku (eng. hash) čime ona postaje nečitljiva korisnicima. Prilikom unosa ona se automatski prevodi u kriptirani oblik (s tim da je iz kriptiranog oblika praktički nemoguće saznati izvornu lozinku) i uspoređuje s pohranjenom vrijednošću.



Slika 2. GNU Grub

Izvor: Wikipedia

2.2.2. Korisnički računi

Korisnički računi osjetljiva su točka sustava Linux zato što korisnici mogu biti nedovoljno upućeni i neodgovorni (ne paze na lozinke, koriste nezaštićene veze i slično) što napadač može iskoristiti za stjecanje pristupa njihovim računima. Nakon što je stekao lokalni pristup, napadač ga može iskoristiti za dodatno povećanje razine ovlasti. Zato kod upravljanja korisničkim računima valja voditi računa o:

- dodjeljivanju najmanjih potrebnih prava svakom korisniku,
- praćenju kada i odakle korisnici pristupaju sustavu,
- otklanjanju neaktivnih računa,
- korištenju istih korisničkih imena na svim računalima i mrežama jer olakšava održavanje te o
- izbjegavanju korištenja skupnih korisničkih računa jer to otežava otkrivanje odgovornosti.

Posebno osjetljiv korisnički račun na Linux sustavima je tzv. „root“ ili administrator jer on ima najviše ovlasti nad sustavom ili mrežom. Zbog njegove osjetljivosti preporuča se korištenje ovog računa samo za vrlo kratke i specifične aktivnosti koje se nikako ne mogu obaviti s manjim ovlastima (npr. poslužitelji koji primaju usluge na TCP priključcima manjim od 1024 : FTP – 21, POP3 – 110, itd. moraju biti pokrenuti kao *root* procesi). Posebno se ne preporuča pokretati s administratorskim ovlastima nesigurne naredbe za udaljeni rad na računalu kao što su rsh/rlogin/rexec. Također, nije poželjno koristiti *.rhosts* datoteke koje omogućuju direktan pristup računu s određenih računala bez poznavanja lozinke. Preporuča se ograničiti i broj terminala s kojih se moguće prijaviti se administratorskim računom. Datoteka */etc/securetty* sadrži popis takvih terminala. Može se i potpuno isključiti udaljeni administratorski pristup. Tek nakon prijave korisnika omogućuje se ili onemogućuje preuzimanje administratorske ljuske. Ukoliko je potrebno nekom korisniku omogućiti neka administratorska prava to se može učiniti pomoću *sudo* programa. Njime se definira ograničena skupina administratorskih prava koju korištenjem tog programa korisnik može steći.

2.2.3. Zaštita datoteka

Osnovna ograničenja pristupa datotekama na Linux sustavima zadaju se preko varijable *umask*. Njezina vrijednost označava inicijalna ograničenja pristupa koja se daju novostvorenim datotekama, a zadaje se s tri znamenke (0-7). Primjer ograničenja i prava pristupa datoteci je sljedeći:

```
drwxrwxrwx
```

Prvi bit pritom označava je li riječ o direktoriju (d) ili datoteci(-), druge tri skupine po tri bita označavaju prava čitanja (r), pisanja (w) i izvođenja (x) za: vlasnika, korisničku skupinu kojoj datoteka ili direktorij pripada te za ostale lokalne korisnike. Maska se postavlja tako da se željena prava postave na 0, a ono što se želi ograničiti označi sa 1. Potom se izračuna oktalna vrijednost tog broja, npr. najrestriktivnija ograničenja zadaju se maskom 077 (-000111111) tako da samo vlasnik ima pristup datoteci. Postavljanje prava na 0, a ograničenja na 1 može biti naizgled nelogično, no radi se o tome da se ovlasti pomoću varijable *umask* računaju preko njezinog komplementa. Znači konačno ovlasti zadane maskom 077 su (d ili -)rwx------.

Navedena prava pristupa imaju malo izmijenjena značenja za datoteke i direktorije:

- r – čitanje datoteke/čitanje sadržaja direktorija (imena datoteka),
- w – pisanje datoteka/stvaranje, brisanje i premještanje datoteka u direktoriju,
- x – pokretanje izvođenja datoteka/ pretraživanje sadržaja(imena) datoteka i poddirektorija..

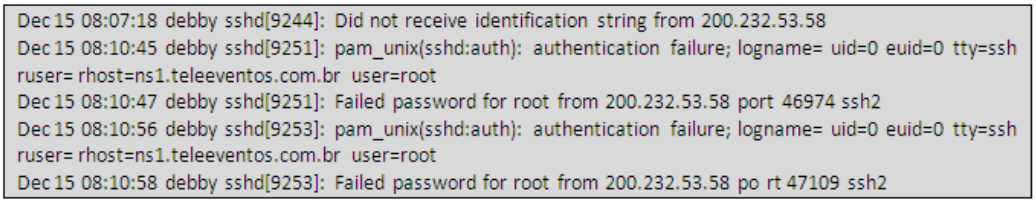
Osobitu pažnju treba voditi o izvršnim programima sa SUID/SGID ovlastima jer se korisnicima, odnosno procesima koji pokreću takve programe privremeno dodjeljuju povišene ovlasti (ovlasti korisnika koji je vlasnik programa). Ranjivosti SUID programa često se iskorištavaju za stjecanje administratorskog pristupa sustavu. Zato je potrebno onemogućiti pokretanje takvih programa u korisničkim „/home“ direktorijima.

2.2.4. Praćenje događaja u sustavu

Važna aktivnost u održavanju sigurnosti sustava i otklanjanju sigurnosnih rizika je praćenje događaja u sustavu. Riječ je o tzv. „log“ zapisima koji prate korisničke aktivnosti, pokušaje prijave na sustav i podizanja sustava, korištenje pojedinih programa i slično. Datoteke koje sadrže zapise o praćenju aktivnosti i događaja u sustavu smještaju se u direktorij *„/var/log“*. Njega je također potrebno primjereno zaštiti kako napadači ne bi mogli prikrivati tragove. Primjeri nekih *log* datoteka su:

- /var/log/auth.log – autentifikacija korisnika,
- /var/log/kern.log – aktivnosti jezgre,
- /var/log/cron.log – automatski pokretani poslovi,
- /var/log/boot.log – aktivnosti podizanja sustava,
- /var/log/httpd/ - pristup Apache web poslužitelju,
- /var/log/mysqld.log – pristup MySQL poslužitelju baze podataka.

Primjer sadržaja *log* datoteke je sljedeći (auth.log):



Slika 3. Primjer auth.log daoteke
Izvor:LinuxQuestions.org

Moguće je koristiti i alate za napredno, sigurnosno-orijentirano praćenje događaja. Riječ je o IDS (eng. Intrusion Detection System) sustavima o kojima će biti riječi i u nastavku dokumenta.

3. Zaštita sustava u nesigurnoj mreži

U ovom poglavlju predstavit će se dodatni i napredniji mehanizmi zaštite. To ne znači nužno da ih prosječan korisnik može zaobići već se u pravilu radi o metodama zaštite koje se koriste na računalima koja su povezana na mrežu. Osnovni i neizostavni element mrežne zaštite je vatrozid čija je osnovna funkcija provjera i filtriranje mrežnih paketa. Također, komunikacija u nesigurnoj mreži zahtjeva zaštitu od mogućih napadača koji prisluškuju promet ili se lažno predstavljaju. Za obranu od takvih opasnosti koriste se metode enkripcije i autentifikacije.

3.1. Enkripcija i autentifikacija

Enkripcija omogućuje zaštitu podataka bilo da se oni nalaze na računalu ili se šalju mrežom. Primjerice, kod udaljene autentifikacije potrebno je poslati lozinku mrežom što nije uputno činiti u nekriptiranom (čitljivom) obliku. Isto vrijedi za razmjenu bilo kakvih osjetljivih podataka. Osim toga, podaci kao što su lozinke moraju biti pohranjeni na nekom mjestu u računalu. Nije sigurno pohraniti ih u izvornom obliku zbog mogućnosti da netko iskorištavanjem ranjivosti stekne pristup datotekama u kojima su pohranjeni. Zato se koristi kriptografija. Uz kriptografiju usko je vezana i autentifikacija jer metode provjere identiteta korisnika koriste kriptografiju, a zaštićena komunikacija ima smisla onda kada se odvija između autenticiranih korisnika. Kriptografija se na Linux sustavima koristi za ostvarivanje zaštićenih komunikacijskih kanala u mreži (npr. VPN), za zaštitu elektroničke pošte, lozinki, podataka na računalu i dr.

3.1.1. Kriptografija na Linux sustavima

Na Linux sustavu mogu se koristiti različiti protokoli čija je namjena kriptografska zaštita podataka. Oni se ostvaruju na različitim razinama mrežne arhitekture, a najpoznatiji su:

- **SSH** (eng. Secure Shell) – aplikacijski protokol koji se koristi za sigurnu udaljenu komunikaciju i siguran rad s udaljenim računalom. Za Linux sustave dostupno je besplatno programsko rješenje - OpenSSH.
- **SSL** (eng. Secure Sockets Layer) - protokol transportnog sloja, slične namjene kao i SSH, koji kriptira komunikaciju, a ostvaren je u besplatnom paketu OpenSSL
- **IPSec** (eng. IP Security) – mrežni protokol koji kriptira IP pakete u Linux sustavima, a može se koristiti pomoću paketa FreeS/Wan. Nešto manje opsežno i moćno, ali korisno ostvarenje IP kriptiranja sadržano je u paketu CIPE (eng. Cryptographic IP Encapsulation). Kriptiranje IP paketa u programu CIPE ostvaruje se tzv. „*tuneliranjem*“ preko UDP protokola.

Za kriptiranje, dekriptiranje i potpisivanje poruka elektroničke pošte na Linux sustavima mogu se koristiti sljedeći protokoli:

- **S/MIME** – standard kojim se propisuje kriptiranje i potpisivanje elektroničkih poruka koje se uklapaju u format MIME. Na Linux sustavima, pomoću programa *openssl smime*, moguće je rukovati elektroničkom poštom zaštićenom prema ovom standardu.
- **OpenPGP** – program je iste namjene kao i S/MIME (za zaštitu elektroničke pošte). Razlikuje se od S/MIME-a utoliko što PGP nije službeni standard i zasniva se na različitim metodama provjere javnih ključeva. Kriptografski algoritmi koji se pritom koriste podjednake su sigurnosti za obje metode. PGP se također može koristiti za kriptiranje datoteka.

Kod kriptografije se javljaju problemi razmjene tajnih ključeva i provjere pripadnost javnih ključeva korisnika. Prvi problem rješava se posebno oblikovanim algoritmima (Diffie-Hellman), a drugi problem rješava se najčešće pomoću digitalnih certifikata. Njih izdaju

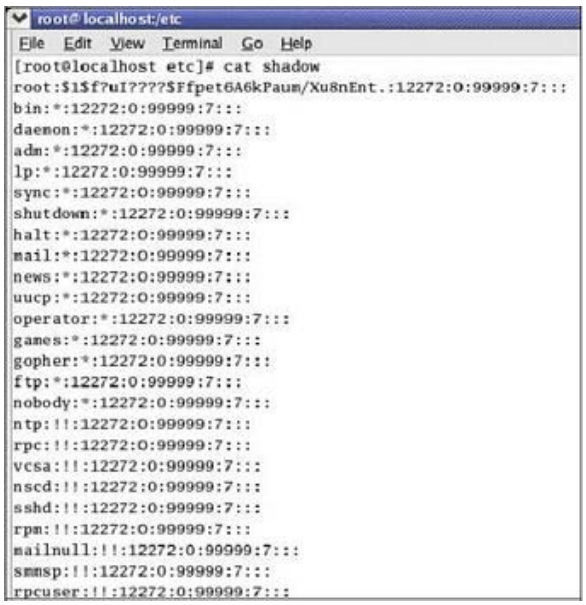
ovlaštena certifikacijska tijela. Najšire korišten format certifikata je X.509, a koriste ga SSL, SSH, IPSec, S/MIME i brojni drugi protokoli.

Osim podataka u mrežnoj komunikaciji, moguće je štiti datoteke i direktorije na računalu. Za tu vrstu zaštite koristite se CFS (eng. Cryptographic File System) sustav i njegova naprednija inačicu TCFS (eng. Transparent Cryptographic File System). Riječ je o alatima koji omogućuju kriptiranje direktorija zajedno s poddirektorijima te pohranjivanje kriptiranih datoteka u takve direktorije.

3.1.2. Autentifikacija na Linux sustavima

Kod klasične autentifikacije korisnika (uporabom lozinke) važno je kako i gdje se lozinke pohranjuju te tko i kako može njima pristupiti. Osim toga, lozinke moraju biti oblikovane tako da ih je teško pogoditi kako bi se smanjila mogućnost napada pretraživanjem svih mogućih rješenja (eng. brute force). Dobro oblikovana lozinka mora biti dovoljno duga te ju se ne smije moći lako pogoditi pretraživanjem rječnika. To znači da se treba odabrati tako da bude besmislena u jezičkom i značenjskom smislu. Imena kućnih ljubimaca primjerice nisu dobre lozinke. U oblikovanju sigurne lozinke pomaže korištenje brojeva, slova i posebnih znakova. *Passwd* je koristan program, najčešće uključen u Linux OS, koji onemogućuje zadavanje jednostavnih lozinki. Osim njega, provjera jakosti lozinke može se izvesti pomoću tzv. cracker programa koji isprobavaju napad pretraživanjem mogućih rješenja. Primjeri takvih programa su *Crack* i *John the Ripper*. Lozinke se najčešće pohranjuju u kriptiranom ili nepovratno izmijenjenom (eng. hash) obliku. Za kriptiranje lozinki koriste se posebni algoritmi: MD2, MD5, SHA-1, SHA-256 i drugi. Budući da su kod nekih algoritama otkrivene ranjivosti (MD2, MD5) preporuča se korištenje sigurnijih metoda (SHA). Prilikom unosa lozinke, ona se odgovarajućim algoritmima prevodi u oblik u kojem je pohranjena i tek tada uspoređuje s pravom lozinkom. Na taj način onemogućuje se njihovo otkrivanje.

Dodatno, lozinke je moguće zaštititi korištenjem posebne datoteke za njihovu pohranu kojoj pristup ima samo administrator. Najčešće se radi o */etc/shadow* datoteci. Nije praktično onemogućiti pristup svih korisnika *passwd* datoteci jer njoj pristupaju mnogi programi bez administratorskih ovlasti. Zato se samo lozinke mogu spremiti u zaštićenu *shadow* datoteku. Pohranjuje ih se u sažetom (eng. hash) obliku, a u *passwd* datoteku umjesto lozinke upisuje se znak *x* ili *. Sadržaj zaštićene datoteke su korisnička imena, kriptirane lozinke te različite informacije vezane uz datume izmjena i isteka lozinki i korisničkog računa.

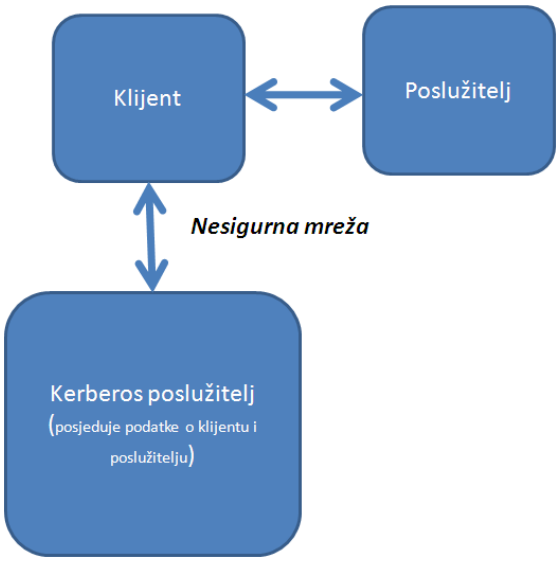


Slika 4. Primjer shadow datoteke

Izvor: www.mhsv.org

Drugi autentifikacijski protokoli podržani na Linux sustavima su:

- **Kerberos** – protokol koji omogućuje komunikaciju između računala u nesigurnoj mreži. Osmišljen je tako da pruža obostranu autentifikaciju, onemogućuje prisluškivanje i lažno predstavljanje. Problem kod korištenja ovog sustava je zahtjev za postojanjem apsolutno sigurnog Kerberos poslužitelja što može biti teško izvedivo.

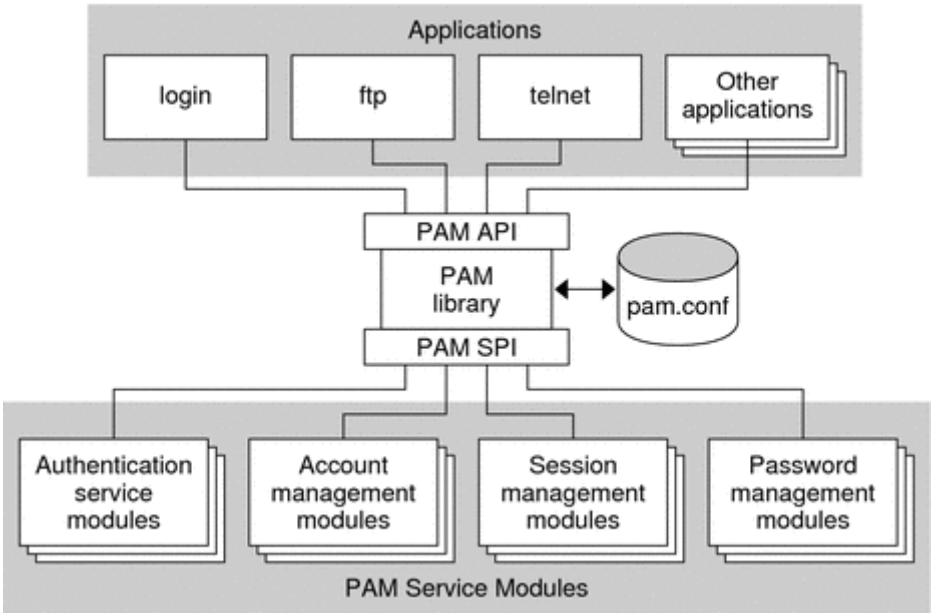


Slika 5. Shema Kerberos komunikacije

- **LDAP** (eng. Lightweight Directory Access Protocol) - protokol aplikacijske razine koji omogućuje pristup informacijama u objektno-orijentiranim bazama podataka, a uključuje autentifikaciju korisnika. Osobito je koristan jer centralizirano pohranjuje i logično organizira podatke o korisnicima. Za Linux sustave dostupna je besplatna izvedba ovog protokola - OpenLDAP.

Kad se govori o autentifikaciji korisnika na Linux operacijskim sustavima važno je spomenuti PAM (eng. Pluggable Authentication Modules) sustav. Riječ je skupini biblioteka koje dolaze u sklopu većine modernih Linux distribucija, a pružaju sučelje prema autentifikacijskim protokolima. PAM administratorima omogućuje izmjenu autentifikacijskih protokola koji se koriste u aplikacijama bez ponovnog prevođenja tih aplikacija. PAM značajno olakšava upravljanje autentifikacijskim sustavima, ali korist donosi samo kod programa koji su oblikovani s PAM podrškom. Četiri neovisna dijela PAM sustava omogućuju:

- provjeru korisničkih računa – jesu li istekli, valjani te je li dopušten pristup korisnika traženoj usluzi,
- autentifikaciju korisnika – ovjera identiteta provjerom lozinke ili korištenjem nekog dostupnog autentifikacijskog protokola,
- održavanje lozinki – uključuje obnavljanje lozinki i postavljanje zahtjeva za snažnim lozinkama,
- upravljanje sjednicama – uspostavljanje (nakon uspješne autentifikacije) i završavanje sjednica.



Slika 6. PAM sustav
Izvor: Linuxtopia

3.2. Mrežna sigurnost

Pod mrežnom sigurnošću pretpostavlja se sigurnost od napadača koji kao metu odabiru mrežne priključke. Osnovna i neizostavna mrežna zaštita je vatrozid, pa ga je zato važno pažljivo konfigurirati. Pritom ga ne treba shvatiti kao jedinu potrebnu i dostatnu zaštitu. Vatrozid štiti mrežu na razini mrežnih usluga kojima se pristupa i IP paketa koji se kroz nju prenose. Ne štiti, primjerice, od virusa i crva niti od iskorištavanja ranjivosti različitih programa. Zaštiti računala u nesigurnoj mreži doprinosi i isključivanje nesigurnih mrežnih usluga koje se mogu iskoristiti za napad. Kao i na ostalim sustavima, na Linuxu je moguće koristiti VPN kriptirane mrežne veze koje štite računala i promet između njih.

3.2.1. Isključivanje nesigurnih usluga

Prvi korak zaštite računala u mreži je otklanjanje nepotrebnih mrežnih usluga koje mogu biti meta napada. Budući da aktivni mrežni poslužitelji osluškuju zahtjeve na određenim priključcima, preko njih se mogu izvoditi različiti napadi, npr. izazivanje DoS stanja slanjem posebno oblikovanih paketa ili otkrivanje informacija koje se šalju preko nesigurne usluge. Najjednostavniji način da se ovo učini je označiti kao komentar linije s nesigurnim uslugama u glavnom poslužiteljskom programu *inetd*. Aktivne usluge definirane su u datoteci */etc/inetd.conf* . Primjer sadržaja takve datoteke je sljedeći (oznaka komentara je znak „#“ na početku linije):

```
…
ftp        stream  tcp   nowait root  /usr/libexec/ftpd       ftpd -l
ntalk      dgram   udp   wait   root  /usr/libexec/ntalkd     ntalkd
#telnet    stream  tcp6  nowait root  /usr/libexec/telnetd  telnetd
shell      stream  tcp46  nowait root  /usr/libexec/rshd rshd
…
```

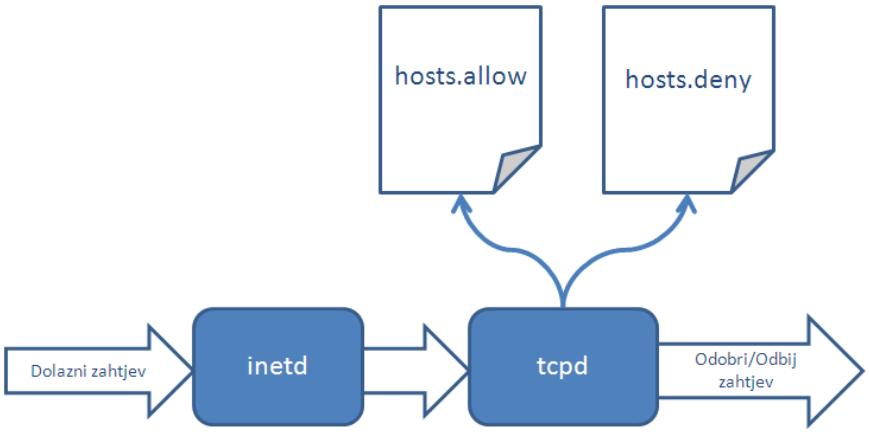
Primjer nesigurnih usluga i protokola su :

- rsh – udaljeno pokretanje naredbi,
- rlogin – udaljena prijava na sustav i
- rcp – udaljeno kopiranje datoteka.

Riječ je o naredbama koje šalju podatke između dvaju udaljenih računala u nektriptiranom obliku (engl. plaintext) tako da ih svatko može presresti i otkriti. Umjesto ovih načina

komunikacije, sigurnije je koristiti SSH protokol koji omogućuje izvođenje svih navedenih akcija, ali uz enkripciju koja komunikaciju čini zaštićenom. Primjer nesigurnog protokola je i POP (eng. Post Office Protocol) koji se koristi za dobavljanje elektroničke pošte, a lozinku šalje u nekriptiranom obliku. APOP (eng. Authenticated Post Office Protocol) je sigurnija alternativa ovom protokolu. Kad se spominje elektronička pošta važno je napomenuti i nesigurnost često korištenog Linux programa *Sendmail*. Riječ je o programu za rukovanje elektroničkom poštom koji je više puta bio meta napada i njegova se upotreba ne preporuča zbog brojnih ranjivosti. Umjesto njega bolje je koristiti alate *Postfix* ili *Qmail*.

Za ograničavanje pristupa uslugama mogu se koristiti i tzv. TCP omotači (eng. TCP wrappers). Radi se o programu *tcpd* koji se poziva iz *inetd* poslužitelja, koji provjerava zahtjeve za pristupom određenoj usluzi koja se pokreće preko *inetd* programa. Nakon što se *tcpd* pozove on provjerava dozvole klijenta na temelju mrežne adrese ili imena klijenta. Dozvoljeni klijenti zapisuju se u datoteku */etc/hosts.allow*, a zabranjeni u */etc/hosts.deny*.



Slika 7. Shema rada TCP omotača

3.2.2. Vatrozid

Osnovni i neizostavni element sigurnosti računala i lokalne mreže koji se povezuju na Internet je vatrozid. Riječ je o uređaju ili programu koji prati mrežni promet između sigurne i nesigurne mreže te ga pritom filtrira, kriptira, dekriptira ili maskira mrežne adrese. Vatrozid može filtrirati IP pakete, provjeravati uspostavljanje TCP/UDP veza i rad specifičnih protokola i usluga (npr. FTP, TELNET) ili maskirati mrežne adrese kako bi zaštitio unutarnji mrežni prostor od napadača. Radi se o sakrivanju mrežnih adresa računala u lokalnoj mreži kako bi im se onemogućio izravan pristup izvana.

Operacijski sustav Linux ima ugrađeni vatrozid i mehanizme za rukovanje njime. U jezgrama inačica 2.0 i 2.2 riječ je o korisničkim alatima *ipfwadm* i *ipchains*. Od inačice 2.4 dostupan je *iptables*, napredniji alat za upravljanje vatrozidom.

Ipfwadm je alat za upravljanje vatrozidom razvijen kao zamjena za *ipfw* alat iste namijene koji se koristio u starijim inačicama sustava. Korisnicima omogućuje:

- mijenjanje pretpostavljenih pravila za sve elemente zaštite,
- automatsko dodavanje dodatnih pravila za poslužitelje s više IP adresa,
- zadavanje adrese i imena sučelja koja se povezuju s pravilima,
- preusmjeravanje i maskiranje paketa i
- zadavanje dodatnih pravila u različitim formatima

Funkcionalnost prosljeđivanja IP adresa i posrednog posluživanja (eng. proxy) nije potpuno dostupna u ovom alatu.

Ipchains je poboljšana inačica paketa *Ipfwadm* koja se koristi u inačicama Linux jezgre 2.2 i više. Poboljšanja uključuju:

- filtriranje fragmentiranih paketa,
- povećanje najviše vrijednosti brojača paketa,

- širi raspon podržanih protokola (ne samo TCP, UDM i ICMP kao kod *Ipfwadm*),
- mogućnost analize paketa na temelju inverznih pravila,
- dodane skripte za lakše održavanje.

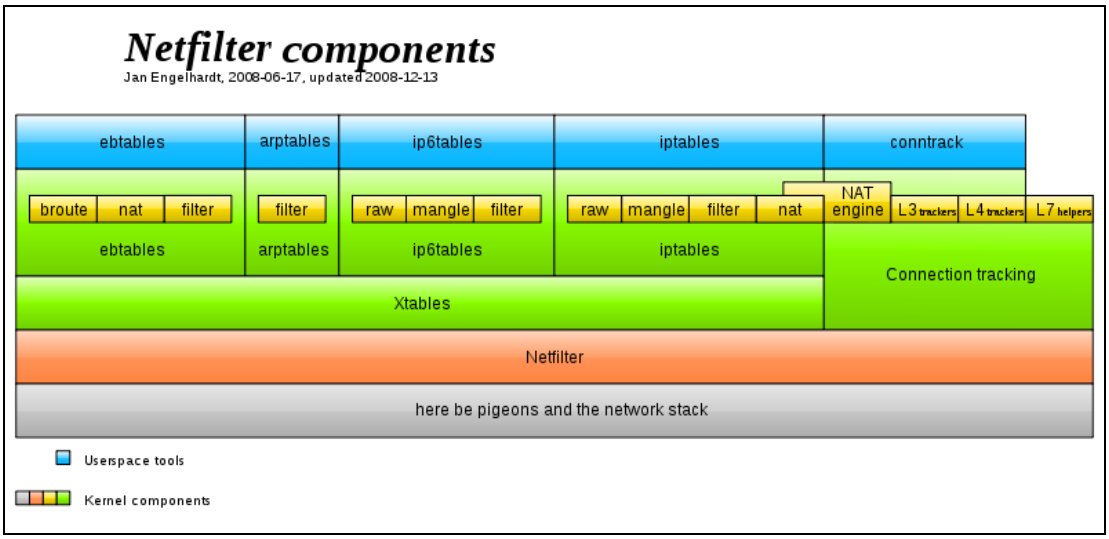
IPchains uvodi i koncept lanaca pravila prema kojima se paketi ispituju, a koji omogućuju pozivanje drugih lanaca prilikom ispitivanja paketa. Ova funkcionalnost omogućuje učinkovitije i prilagodljivije ispitivanje paketa u odnosu na linearno ispitivanje koje se provodi *Ipfwadm* alatom.

3.2.3. *Iptables*

Iptables je najnoviji alat za upravljanje postavkama vatrozida na operacijskom sustavu Linux. Koristi se u jezgrama od inačice 2.4 nadalje. Postoji više vrsta tablica za različite protokole:

- iptables za IPv4,
- ip6tables za IPv6,
- arptables za ARP i
- ebtables za Ethernet

Iptables se mora izvoditi pod administratorskim ovlastima, a upravlja postavkama vatrozida koji je sadržan u Netfilter modulima jezgre. Netfilter je programsko rješenje koje se ugrađuje u jezgru Linux operacijskog sustava i stvara sučelje prema korisniku, a namijenjeno je filtriranju i različitim vrstama rukovanja IP paketima (prevođenje mrežnih adresa, prosljeđivanje TCP priključaka i sl.). Iptables je jedan od elemenata Netfilter rješenja.



Slika 8. Netfilter arhitektura i Iptables

Izvor: Wikipedia

Za razliku od prethodno opisana dva alata, *Iptables* omogućuje analizu paketa u kontekstu veze u kojoj se oni šalju. Riječ je o tzv. „stateful“ analizi kod koje se paketi provjeravaju uzimajući u obzir kontekst njihova slanja (što omogućuje naprednije rukovanje paketom). Analiza koja svaki paket promatra zasebno naziva se „stateless“ analiza. Primjer koristi od načina rada, koji uzima u obzir kontekst, može biti FTP protokol koji prilikom komunikacije legitimno zahtjeva uspostavu veze na proizvoljnim TCP priključcima. Vatrozid koji zna da se taj zahtjev šalje u kontekstu FTP protokola propustit će zahtjev, dok će vatrozid koji nema tu informaciju otežati ili onemogućiti uspostavu veze.

Tablice pravila ostvaruju se pomoću modula Xtables. On sadrži izvorno definirana pravila, a korisnik može dodavati svoja. Pravila su povezana u lance i tako čine niz provjera kroz koje paket prolazi. Također, određene provjere povezuju se s određenim tipom paketa pa svi paketi ne prolaze kroz iste provjere.

Iptables omogućuje i mijenjanje IP adresa NAT (eng. Network Address Translation) postupkom. Kod ovog postupka u zaglavljima paketa izmjenjuje se IP adresa tako da se IP

jednog odredišta u lokalnoj mreži ili cijelog njezinog adresnog prostora „skriva“ iza jedne (najčešće javne) IP adrese. Na taj način paketi koji izlaze iz mreže naizgled potječu od usmjerivača, odnosno računala koje povezuje sigurnu i nesigurnu mreži. Time se ograničava pristup računalima u zaštićenoj mreži. Omogućeno je i pristupanje poslužiteljima u zaštićenoj mreži pomoću tzv. “port forwarding” metode.

Način upotrebe iptables alata opisani su u pripadnim *man* (eng. manual) stranicama pomoći. Nekoliko osnovnih naredbi za rad s ovim alatom i za njegovu konfiguraciju dano je u nastavku poglavlja.

Pokretanje alata tijekom podizanja sustava može se zadati naredbom:

```
chkconfig iptables on.
```

Ukoliko se alat zaustavlja ili pokreće nakon što je sustav podignut koriste se naredbe

```
service iptables start, service iptables stop i service iptables restart.
```

Status vatrozida može se provjeriti naredbom

```
service iptables status.
```

Jednostavni primjeri konfiguracije vatrozida su sljedeći:

- omogućavanje slanja i primanja ICMP zahtjeva:

```
iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A INPUT  -p icmp --icmp-type echo-reply   -j ACCEPT
```

- omogućavanje prihvaćanja svih TCP paketa koji dolaze na sučelje *eth0* prema adresi 188.189.2.1:

```
iptables -A INPUT -s 0/0 -i eth0 -d 188.189.2.1  -p TCP -j ACCEPT
```

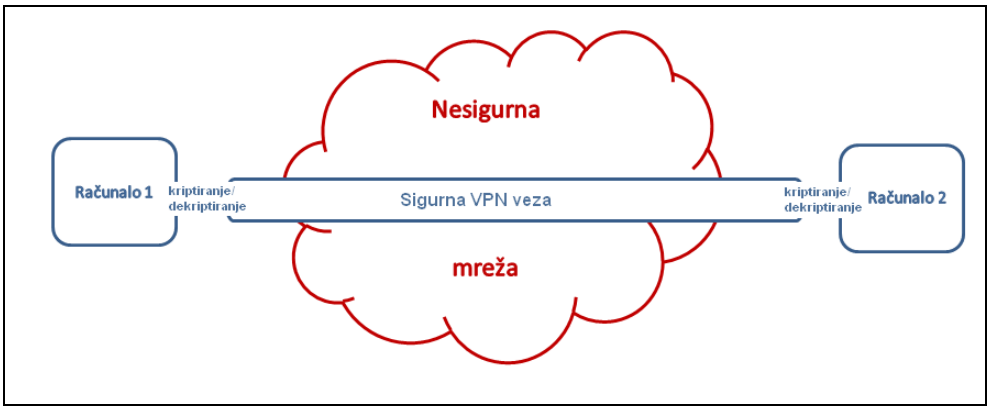
Opcija „–A“ znači dodavanje pravila na kraj lanca, INPUT/OUTPUT govori radi li se o odlaznim ili dolaznim paketima, ACCEPT znači prihvaćanje paketa koji zadovoljavaju parametre definirane u pravilu (IP adrese izvora, odredišta, vrsta zahtjeva, mrežno sučelje itd.).

U razvoju je trenutno alat nftables koji bi u budućnosti trebao zamijeniti iptables. Glavna promjena koju novi alat uvodi je zamjena iptables, ip6tables, arptables i ebtables tablica jednom tablicom koja će virtualno objediniti njihove funkcionalnosti. To znači da će usluge funkcionirati odvojeno, ali će im se pristupati kao da je riječ o jednoj tablici.

3.2.4. VPN

Često korišten način zaštite podataka u računalnoj mreži je VPN povezivanje. Riječ je o uspostavljanju zaštićene (kriptirane) veze između računala u nesigurnoj meži kojom se simulira sigurna, lokalna povezanost. Na taj način omogućuje se razmjena osjetljivih podataka. VPN se na Linux sustavima može ostvariti na više načina:

1. pomoću SSH protokola (eng. VPN over SSH) ili
2. pomoću gotovih programskih rješenja kao što su:
    a. *vpnd* – pozadinski program koji povezuje dvije mreže putem TCP/IP protokola ili virtualne veze između SLIP (Serial Line Internet Protocol) sučelja. Za enkripciju se koristi simetrični algoritam Blowfish
    b. *FreeS/Wan* – Linux izvedba protokola IPSEC i IKE (eng. Internet Key Exchange). Uključene su autentifikacija i enkripcija na razini mrežnog IP protokola čime se ostvaruje sigurna i zaštićena komunikacija na Internetu.



Slika 9. Shema VPN veze

4. Očvršćivanje jezgre sustava

Očvršćivanje operacijskog sustava Linux podrazumijeva uključivanje sigurnosnih mehanizama koji su dostupni u izvornom sustavu, ali u pravilu nisu uključeni. Radi se primjerice o konfiguraciji vatrozida i IDS (eng. Intrusion detection system) sustava, otklanjanju nepotrebnih i rizičnih programa, detaljnijem praćenju i bilježenju događaja u sustavu, poboljšanju sigurnosti postupka prijave na sustav i sličnim postupcima.

Očvršćivanje sustava može se izvesti pomoću programskih paketa koji automatiziraju ovaj postupak (Bastille Linux) ili ručnim prilagođavanjem postavki sustava što zahtjeva veću stručnost. Osim toga, dostupne su unaprijed očvrsnute distribucije Linux sustava kao što je *Immunix*. Očvršćivanje sustava, u mjeri koja ovisi o prilikama u kojima se on koristi, preporuča se za svaki sustav. No treba imati na umu da ovaj pristup čini otežanim napade na sustav izvana, ali ne i iznutra. Napadač koji uspije pomoću ranjivog programa ili SUID mehanizama steći povećane ovlasti na sustavu, odnosno neovlašteno ući u sustav, zaobišao je vanjsku zaštitu i ništa ga više ne sprječava da napravi štetu. Posebice ako se radi o administratorskom pristupu koji, kao što je već objašnjeno, uključuje apsolutne ovlasti.

Kako bi se smanjila šteta koju neovlašteni pristup može izazvati, moguće je koristiti dodatnu sigurnosnu zaštitu na razini jezgre. Riječ je o tzv. „očvršćivanju jezgre sustava“. Ovaj je postupak složeniji od očvršćivanja samog sustava pa se ne preporuča za sustave koji ne sadrže osobito osjetljive podatke. Složenost ne proizlazi samo iz postupka očvršćivanja jezgre već i iz kasnijeg održavanja takvog sustava koje za nedovoljno stručnog administratora može biti problematično.

4.1. Programski dodaci za očvršćivanje jezgre

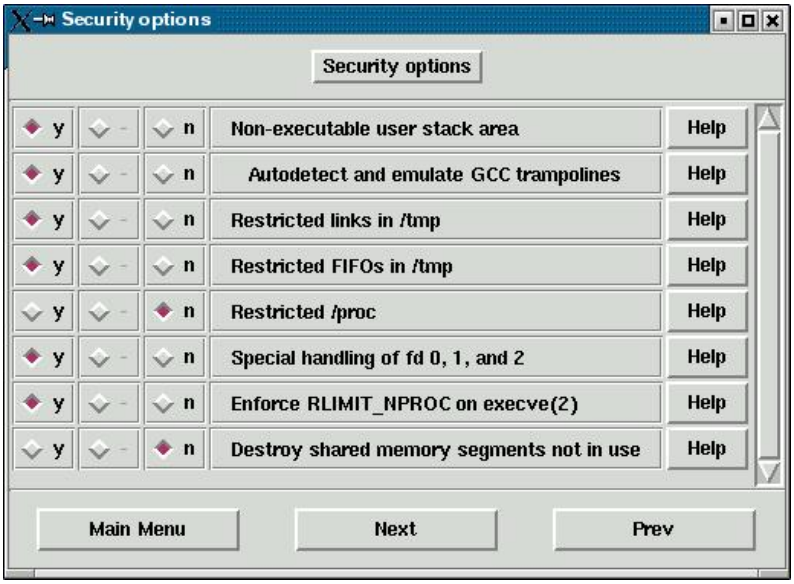
Očvršćivanje jezgre izvodi se posebnim programskim dodacima koji mogu uvoditi nove sigurnosne mehanizme ili mogu samo otklanjati postojeće nesigurnosti i ranjivosti. Ukoliko se sustav nadograđuje novim sigurnosnim mehanizmima, u pravilu se radi o uvođenju MAC (eng. Mandatory Access Control) provjera. Radi se o mehanizmima koji elemente sustava dijele na subjekte (procesi i dretve) i objekte(datoteke, direktoriji, TCP/UDP priključci). Svakom subjektu i objektu dodjeljuju se sigurnosna obilježja na temelju kojih se pristup subjekta objektu dozvoljava ili ne dozvoljava. Ta se obilježja provjeravaju prilikom svakog pokušaja pristupa. MAC pravila postavlja administrator sustava. Ona uključuju klasifikaciju objekata i korisnika prema povjerljivosti i domeni kojoj pripadaju (npr. određenom projektu). Ukoliko korisnik i objekt imaju podudarajuće sigurnosne oznake, pristup se odobrava. U suprotnom, pristup se odbija (npr. visoko povjerljiv korisnik ne može pristupati visoko povjerljivim podacima ukoliko ne dijele domenu). Iako očvršćivanje jezgre predstavlja znatno ojačanje zaštite sustava, ne znači i apsolutnu zaštitu. Postojanje ovakve zaštite može stvoriti pretjeran osjećaj sigurnosti koji dovodi do nepažljivog ponašanja. Također, opasnost postoji i u samoj sigurnosnoj nadogradnji koja, poput bilo kojeg drugog programskog proizvoda, može sadržavati ranjivosti. Na kraju krajeva, ukoliko napadač uspije postojeće ranjivosti iskoristiti za podizanje drugog operativnog sustava, sva je zaštita uništena. Postoji čitav niz dodataka za očvršćivanje jezgre, a neki od popularnijih su:

- Medusa DS9 Security Systems,
- Rule Set Based Access Control for Linux (RSBAC),
- LIDS (eng. Linux Intrustion Detection System) i
- Openwall Project.

Uvid u zaštitu koju stvara očvršćivanje jezgre dan je u sljedećim poglavljima na primjeru rješenja LIDS i Openwall Project.

4.1.1. Openwall

Openwall Project dostupan je za inačice 2.0 i 2.2 jezgre operacijskog sustava Linux. Ova sigurnosna nadogradnja ne uvodi MAC mehanizme, već samo otklanja postojeće ranjivosti u sustavu kao što je mogućnost pisanja po nepripadnim dijelovima memorije (FIFO – First In First Out), mogućnost pokretanja programa preko korisničkih stogova, mogućnost saznavanja informacija o procesima drugih korisnika, neoprezno rukovanje SUID mehanizmima i privremenim datotekama, nepostojanje ograničenja na korištenje CPU i memorijskih resursa i slično. Primjerice, otklanjanjem posljednje navedene ranjivosti (neograničeno korištenje CPU-a i memorije) i uvođenjem spomenutih ograničenja onemogućuje se znatan broj DoS (eng. Denial of Service) napada. Također, zabrana pokretanja memorije preko korisničkih stogova znatno otežava iskorištavanje tzv. „*buffer overflow*“ ranjivosti. Osiguravanje direktorija „/tmp“ onemogućuje neovlašteno pristupanje datotekama preko simboličkih veza.



Slika 10. Openwall konfiguracijski GUI
Izvor: SecurityFocus

Osim što uvodi poboljšanja, Openwall može uzrokovati probleme u radu pojedinih programa, posebice onih koji rade s bazama podataka. Zato je važno pažljivo izvesti nadogradnju i informirati se o svim mogućim problemima na stranicama projekta [8]. Također, iako Openwall sigurnosni dodaci otklanjaju mogućnost napada iskorištavanjem pojave prepisivanja spremnika u pojedinim programima, ne štite od prepisivanja spremnika uređenog po principu stoga (eng. heap based buffer overflow) niti od napada koji su oblikovani upravo s ciljem zaobilaženja ove zaštite. Bez obzira na mogućnost napada, ovakva programska nadogradnja predstavlja značajan doprinos sigurnosti sustava pa se stoga preporuča svim naprednijim korisnicima.

4.1.2. LIDS

Za razliku od prethodno opisane zaštite, LIDS (eng. Linux Intrusion Detection System) uvodi MAC mehanizme u rad jezgre. Uz MAC ograničenja, LIDS omogućava i značajnije smanjenje administratorskih ovlasti gdje je to moguće realzirati. Uz detaljnu MAC provjeru prava pristupa, omogućuje se i postavljanje vremenskih ograničenja u kojima subjekt može pristupati objektu (npr. korisnik može pristupati određenoj datoteci samo 1 sat dnevno, i to u periodu od 13 do 15 sati). Onemogućuje se učitavanje štetnih modula jezgre, a omogućuje se i stalno praćenje aktivnosti na TCP/UDP priključcima, skrivanje procesa i datoteka.

Iako LIDS omogućuje širok raspon različitih sigurnosnih postavki i ograničenja, zahtjeva izuzetno dobro poznavanje Linux sustava. Primjerice, administrator mora odrediti najmanje ovlasti potrebne za rad nekog programa. Ukoliko je potrebna ovakva snažna razina zaštite i dostupan je stručni kadar koji će je moći ugraditi i održavati, LIDS je svakako preporučljivo rješenje.

Slika 11. LIDS sučelje
Izvor: SecurityFocus

4.2. SELinux

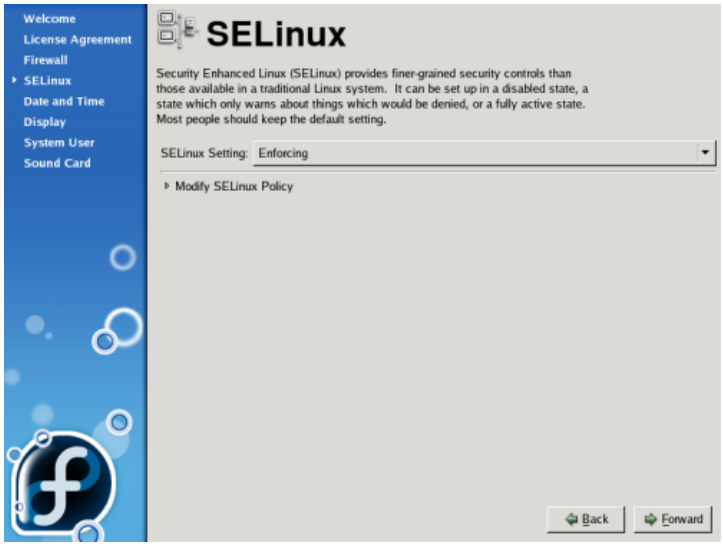
SELinux (eng. Security Enhanced Linux) je prvotno razvijen kao skupina sigurnosnih dodataka za jezgru operacijskog sustava, no danas je on potpuno integriran u inačicu 2.6 jezgre. Sustav je razvijen u američkoj agenciji za nacionalnu sigurnost (eng. National Security Agency), a od 2000. godine distribuira se kao besplatna programska podrška otvorenog koda (eng. open source). SELinux izgrađen je na FLASK (eng. Flux Advanced Security Kernel) sigurnosnoj arhitekturi, a dostupan je u različitim distribucijama Linux operacijskih sustava (Red Hat Enterprise Linux, Debian, Fedora, Ubuntu, Yellow Dog Linux, OpenSUSE). U nastavku ovog poglavlja razmotrit će se poboljšanja koja uvodi SELinux, ali i problemi koji se javljaju kod njegovog korištenja.

Ojačanja jezgre uvode se kroz MAC mehanizme u LSM (eng. Linux Security Modules) modulima jezgre. LSM je programsko okruženje koje omogućuje nadogradnju jezgre različitim sigurnosnim modelima. MAC mehanizmi uvode se u većini podsustava jezgre, a razvijeni su s ciljem odvajanja i kategoriziranja podataka s obzirom na njihovu osjetljivost, odnosno kritičnost očuvanja integriteta i tajnosti. Na ovaj način želi se olakšati prepoznavanje mogućih ranjivosti kako bi se lakše uveli i zaštitni mehanizmi.

Važno je napomenuti i kako SELinux ne otklanja postojeće ranjivosti jezgre i ne rješava specifična sigurnosna pitanja. On samo stvara pogodno okruženje i mehanizme za uvođenje različitih zaštitnih metoda koje administrator odabire i postavlja prema zahtjevima sustava i svrhe za koju se on koristi. SELinux tako funkcionira neovisno o klasičnim Linux sigurnosnim mehanizmima i otporan je na ranjivosti programa koji imaju visoke ovlasti. Jedino o čemu učinkovitost SELinux zaštite ovisi je ispravnost jezgre i sigurnosnog modela.

Neki od naprednih sigurnosnih mehanizama uključenih u SELinux OS su:

- kontrola mrežnih sučelja, priključaka i poruka,
- kontrola datotečnog sustava i otvorenih opisnika datoteka,
- kontrola procesa (inicijalizacija, nasljeđivanje, izvođenje),
- sigurnost na više razina (integritet i tajnost podataka štite se odvojenim mehanizmima),
- dodijeljena prava i donesene odluke spremaju se u privremenu memoriju radi učinkovitosti te
- neovisnost o specifičnim formatima sigurnosnih oznaka i sadržaja.



Slika 12. SELinux sučelje

Izvor: Fedora Core 5 Installation Guide

Uz svu sigurnosnu dobrobit ovog sustava postoje brojne zamjerke na složenost njegovog korištenja i održavanja. Zato se SELinux preporuča korisnicima kod kojih postoji realna potreba za snažnom zaštitom i sposobnost održavanja ovakvog sustava.

5. Alati za ispitivanje sigurnosti

Uspostavljena zaštita računala ili mreže može se ispitati i provjeriti pomoću raznih programskih alata. Među njima su neki komercijalni, a neki besplatno dostupni. Kod druge skupine alata češće se uočavaju propusti, a ukoliko se preuzimaju s neprovjerenih izvora može se raditi i o zamaskiranim štetnim programima (trojanski konji). Ipak među besplatno dostupnim alatima ima i kvalitetnih rješenja. U ovom poglavlju razmotrit će se tri takva alata:

1. **Rootkit Hunter** – alat za otkrivanje programa koji ozbiljno narušavaju sigurnost sustava tako što omogućuju napadaču neovlašten pristup sustavu (eng. backdoor), izmjenu osjetljivih dijelova OS-a (eng. rootkit) i slično.
2. **Snort** – alat za analizu i filtriranje prometa s naprednim mogućnostima koje ga razlikuju od običnog vatrozida.
3. **Nmap** – alata za pretraživanje poslužitelja i usluga na mreži te zaštite koju različiti sustavi koriste.

5.1. Rootkit Hunter

Rootkit Hunter je alat otvorenog programskog koda za otkrivanje štetnih programa kao što su *rootkit* programi i trojanski konji. Razvoj i održavanje alata odvija se u sklopu projekta *SourceForge The Rootkit Hunter*. Osim na Linux sustavima, može se koristit i na drugim Unix operacijskim sustavima kao što su FreeBSD i Mac OS X.

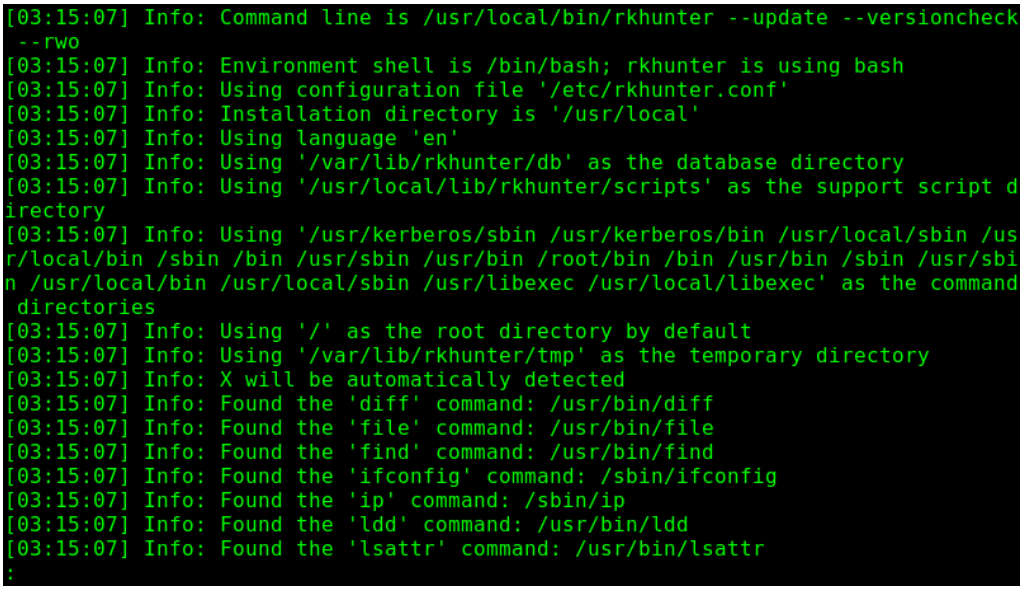
Rootkit je vrsta štetnog programa kojem je cilj sakriti zlonamjerne aktivnosti na sustavu. Uobičajeno se koristi u svrhu sakrivanja zloćudnih programa i njihovog djelovanja na sustav. Pritom mora nadomjestiti procese i podatke sustava bez znanja administratora. To znači da se rootkit može podmetnuti tek ako je već stečen pristup osjetljivim dijelovima sustava. Rootkit programi često su ujedno i tzv. trojanski konji koji se predstavljaju kao korisni i sigurni programi, a zapravo sadrže štetan kod (na taj način korisnik se navodi na njihovu instalaciju). Ova vrsta alata može uključivati i tzv. „backdoor“ ulaz u sustav iskorištavanjem ranjivosti podsustava za prijavu korisnika. Programi se često instaliraju u operacijski sustav kao pogonski programi ili moduli jezgre

Rootkit Hunter pregledava sustav tako da uspoređuje sažetke (eng. hash) programa instaliranih na sustavu sa poznatim sažecima izvornih inačica tih programa. Usporedba takvih sažetaka može otkriti odgovara li programski kod instaliran na sustav kodu izvorne i sigurne inačice programa. Ukoliko ne odgovara, Rkhunter javlja upozorenje administratoru. Alat je pouzdan i visoke učinkovitosti, a uključuje i sljedeće sigurnosne provjere:

- usporedbu MD5 sažetaka,
- pretraživanje sumnjivih dozvola pristupa binarnim datotekama,
- pretraživanje sumnjivih znakovnih nizova u LKM (eng. Loadable Kernel Module) modulima,
- pretraživanje skrivenih datoteka,
- pretraživanje tekstualnih i binarnih datoteka,
- pretraživanje poznatih crva, trojanskih konja, rootkit, backdoor i drugih štetnih programa.

Alat se može koristiti tako da se sustav svakodnevno provjerava, a izvještaj se prosljeđuje na adresu elektroničke pošte koju zada administrator.

Rootkit Hunter može ponekad korisne i sigurne datoteke prijaviti kao sumnjive i štetne. To se može dogoditi primjerice u slučaju nadogradnje nekih programa koji još nisu registrirane. U tom slučaju potrebno je usporediti rezultate ovog alata s drugim alatima koji provjeravaju integritet podataka (ukoliko se oni koriste na sustavu). Moguće je i usporediti nadograđene pakete s izvornim kodom.



Slika 13. Rkhunter zapis o aktivnosti (eng. log)
Izvor: SysAdmin.MD

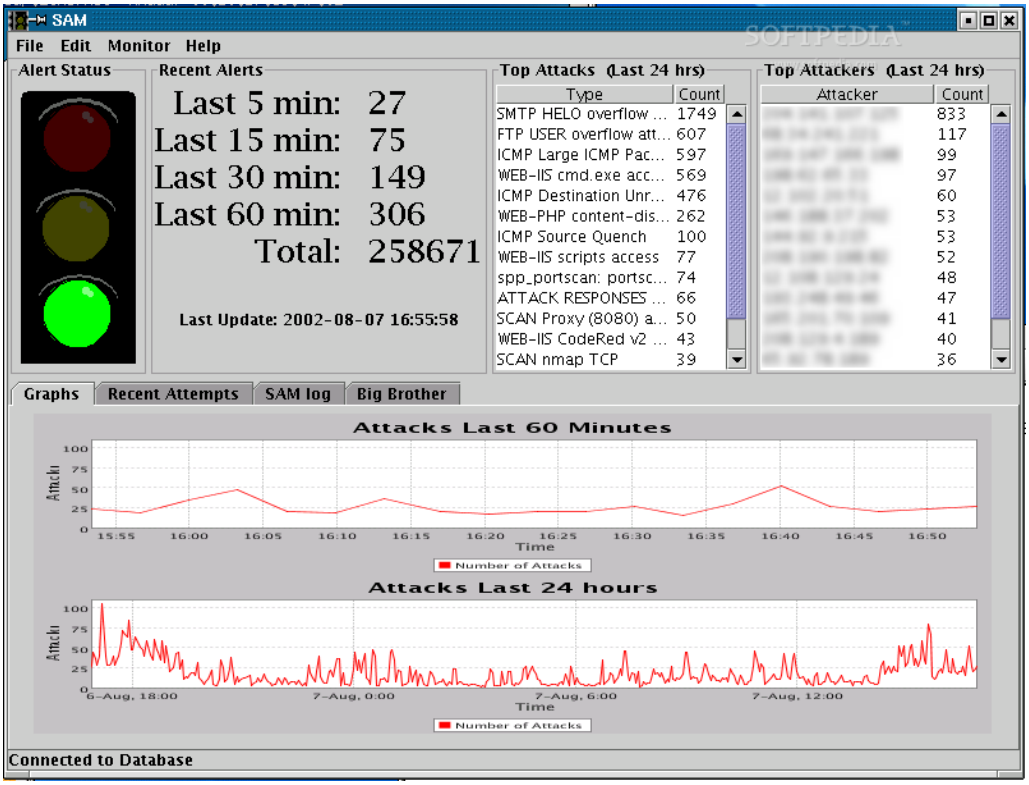
Ukoliko na sustavu postoji rootkit, tada je najbolje rješenje ponovna instalacija jer je širinu štetnog djelovanja teško odrediti, a svaki propust može značiti neuspješan oporavak sustava. Prije instalacije potrebno je stvoriti kopiju svih podataka. Pritom nisu uključene binarne datoteke (izvršne datoteke, biblioteke i slično) čiji integritet nije moguće sa sigurnošću utvrditi jer one mogu biti izvor štetnog djelovanja. Prilikom instalacije i konfiguracije sustava i usluga potrebno je ograničiti pristup mreži vatrozidom i sigurnosnim mehanizmima kao što je primjerice PAM. Poželjno je posebnu pažnju posvetiti pregledavanju dnevničkih zapisa (eng. log files) i programima koji su bili ranjivi u vrijeme uspješnog napada na sustav.

5.2. Snort

Snort je besplatan i moćan alat za praćenje mrežnog prometa. Namijenjen je prvenstveno Unix operacijskim sustavima, no dostupne su i inačice za Windows OS. Na stranicama projekta dostupni su brojni dokumenti koji uvode korisnike u mogućnosti i načine upotrebe alata Snort. Raspon usluga koje Snort nudi seže od pasivnog praćenja paketa i zapisivanja rezultata, do IDS/IPS (eng. Intrusion Prevention/Detection System) funkcionalnosti. U pasivnom načinu rada paketi se mogu samo osluškivati i eventualno zapisivati u datoteke koje se mogu naknadno analizirati. IDS sustav analizira mrežni promet i na temelju rezultata stvara upozorenja i dnevničke zapise o mogućim opasnostima. Pritom korisnik ima mogućnost samostalno oblikovati pravila prema kojima će ocjenjivati rizičnost paketa. Način zadavanja pravila opisan je u dokumentaciji alata. Pravila se upisuju u tekstualnom obliku u konfiguracijske datoteke, a omogućuju propuštanje paketa, podizanje upozorenja te automatsko aktiviranje novih pravila koja su do tada bila isključena (tzv. „dinamična pravila“). Osim toga mogu se uvesti i akcije odbacivanja prometa:

- tako da se promet registrira, ali ne propusti,
- da se promet registrira, odbaci i pošalje dojava izvoru paketa i
- da se paket odbaci bez ikakvih dodatnih akcija.

Pravila mogu provjeravati IP adrese paketa, TCP/UDP priključke, ali i sadržaj samih paketa. Osim filtriranja paketa, može se koristiti i filtriranje specifičnog prometa kao što je SMTP ili SSH.



Slika 14. Snort sučelje

Izvor: Softpedia

Snort također zahtijeva postojanje dodatne programske podrške na sustavu za svoj rad. Ona uključuje:

- Libpcap – biblioteka koja sadrži API (eng. Application Programming Interface)za praćenje mrežnog prometa
- PCRE – biblioteka za rukovanje regularnim izrazima napisana u jeziku C
- Libnet – generički API za pristup različitim protokolima
- Barnyard – program za rukovanje izlaznim datotekama koje stvara Snort

Snort je izuzetno moćan alat, no napredne mogućnosti upotrebe često znače i složenije korištenje i održavanje, zato se njegova upotreba preporuča naprednim korisnicima i onima koji su spremni ozbiljnije se potruditi kako bi iskoristili sve mogućnosti koji alat pruža.

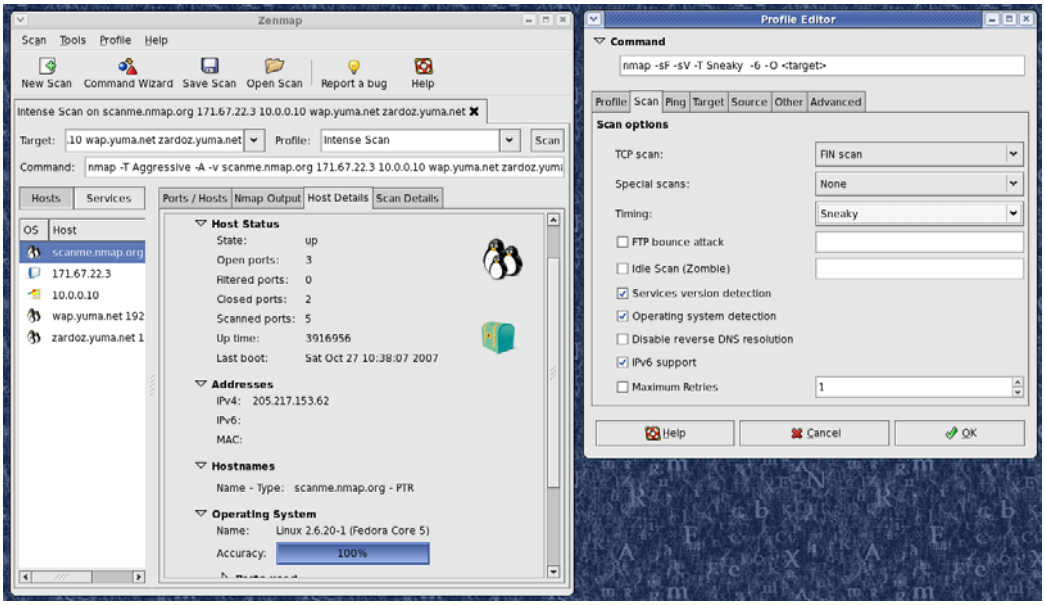
5.3. Nmap

Nmap je besplatni alat za različite operacijske sustave (Unix, Linux, Windows) namijenjen pretraživanju mreže, dostupnih usluga i računala u njoj. Odlikuju ga jednostavnost korištenja te prilagodljivost različitim veličinama mreža. Podjednako je učinkovit u velikim i malim mrežama (npr. jedno računalo). Pomoću ovog alata moguće je uočiti:

- aktivne poslužitelje na mreži,
- operacijske sustave pokrenute na računalima,
- dostupne usluge,
- MAC adrese i imena poslužitelja na temelju IP adrese,
- vrste vatrozida koji se koriste,
- dostupne mrežne servise i radne inačice
- te druge informacije iskoristive u očuvanju ili narušavanju sigurnosti.

Zbog svojih mogućnosti otkrivanja otvorenih priključaka i usluga Nmap se koristi u svrhu poboljšanja sigurnosti mreža i računala, ali i u svrhu planiranja napada na sustav. Osim rada

putem naredbenog retka Nmap se može koristiti i pomoću grafičkog korisničkog sučelja. Odlikuje ga i detaljna korisnička dokumentacija čije proučavanje se svakako preporuča.



Slika 15. Nmap korisničko sučelje
Izvor:Insecure.org

Nmap ispituje TCP/UDP priključke na dostupnim računalima i na temelju njihova odgovora doznaje jesu li oni otvoreni (čeka li neki poslužiteljski program zahtjeve na tom priključku), zatvoreni ili možda zaštićeni vatrozidom. Primjeri jednostavnih skeniranja Nmap programom su sljedeći:

- Ping pretraživanje aktivnih računala u mreži :

  nmap -sP –v 0.0.0.1

- Skeniranje inačica programa koji osluškuju na priključcima u nekom razmaku, npr. 20-30.

  nmap -sV -p 20-30 0.0.0.1

- Pretraživanje TCP servisa u mreži 192.168.0.0/24 , pri čemu parametrom „–v“ dodajemo zahtjev za detaljnim ispisom podataka. Znak „*“ znači da se ispituju IP adrese od 192.168.0.0 do 192.168.0.255.

  nmap -sS -v "192.168.0."*

Osim ovih jednostavnih načina pretraživanja mreža Nmap se može naprednije koristiti za pretraživanje dodatnih IP protokola kao što su EGP (eng. Exterior Gateway Protocol), IGP (eng. Interior Gateway Protocol), pretraživanje i identifikaciju RPC (eng. Remote Procedure Call) servisa, pretraživanje preko tzv. zombi računala (mrežu naizgled pretražuje zombi računalo) i dr.

6. Zaključak

Podizanje zaštite operacijskog sustava nezaobilazan je posao svakog administratora. Što će ta zaštita uključivati, ovisi o specifičnosti primjene sustava. Primjerice, neće biti potrebno niti poželjno na svakom sustavu uvoditi ojačanja jezgre koja bitno kompliciraju njegovo održavanje, ali neki sigurnosni mehanizmi moraju se podrazumijevati uvijek. To su primjerice zaštita lozinki (nezaštićene lozinke nemaju smisla), korištenje vatrozida (iz Internet mreže uvijek potencijalno prijete opasnosti), enkripcija komunikacije ukoliko se razmjenjuju osjetljivi podaci, autentifikacija udaljenih korisnika i slično.

U ovom dokumentu nisu razmotreni antivirusni alati za Linux sustav (npr. ClamAV) jer oni imaju zasebnu ulogu otklanjanja zlonamjernih programa, ali ne integriraju se u sam OS. Što ne znači da se njihova primjena ne preporuča, samo da nisu obuhvaćeni obrađenom temom. Posljednja, ali ne i najmanje važna stvar koju treba naglasiti jest redovita nadogradnja. Programski proizvodi uvijek su podložni ranjivosti. Zato ih je važno redovito nadograđivati, uključujući i programe koji se koriste u svrhu zaštite sigurnosti (i oni sami mogu biti ranjivi).

Ideja ovog dokumenta bila je dati općeniti uvod u metode zaštite Linux sustava koji se može iskoristiti kao polazišna točka, a detalji o posebnim zaštitnim mehanizmima spomenutim u ovom dokumentu mogu se potražiti pomoću tražilice u drugim dokumentima objavljenim na CERTovim stranicama (www.cert.hr).

7. Reference

1. Kevin Fenzi, Dave Wreski, Linux Security HOWTO, http://www.tldp.org/HOWTO/Security-HOWTO/index.html, rujan 2009.
2. Anton Chuvakin, Linux Kernel Hardening, http://www.securityfocus.com/infocus/1539, rujan 2009.
3. Snort, http://www.snort.org/, rujan 2009.
4. Nmap, http://nmap.org/, rujan 2009.
5. Rkhunter, http://rkhunter.sourceforge.net/, rujan 2009.
6. Wikipedia, http://en.wikipedia.org/wiki/Main_Page, rujan 2009.
7. Quick HOWTO:Ch14:Linux Firewalls Using iptables, http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:_Ch14_:_Linux_Firewalls_Using_iptables, rujan 2009.
8. Openwall, http://www.openwall.com/, rujan 2009.