Sigurnost Linux servera

Dokument najbolje prakse
(smernice i preporuke)

Izrađen u okviru AMRES tematske grupe za oblast sigurnost
(AMRES BPD 114)

Autori: Miloš Kukoleča, Miloš Zdravković
Saradnik: Ivan Ivanović

April 2014

Dokument broj: GN3plus-NA3-T2-AMRES-BDP-114
Verzija / datum: April 2014.
Izvorni jezik : Srpski
Originalni naslov: “Sigurnost Linux servera”
Originalna verzija / datum: Verzija 1 / 28. April 2014.
Kontakt: milos.kukoleca@amres.ac.rs, milos.zdravkovic@rcub.bg.ac.rs

AMRES/RCUB snosi odgovornost za sadržaj ovog dokumenta. U izradi dokumenta učestvovala je tematska grupa za oblast sigurnost organizovana u AMRESu radi sprovođenja zajedničkih aktivnosti na razvoju i širenju dokumenata sa tehničkim smernicama i preporukama za mrežne servise u višokoškolskim obrazovnim i istraživačkim ustanovama u Srbiji.

Dokument je nastao kao rezultat istraživanja koja su finansirana sredstvima Sedmog okvirnog programa Evropske zajednice (FP7/2007-2013) po ugovoru br. 238875, koji se odnosi na projekat „Multi-Gigabit European Research and Education Network and Associated Services (GN3plus)”.

Sadržaj

Executive Summary

This document consists of guidelines and recommended practices that system administrators should follow during initial Linux installation for a server environment. The purpose of the document is to help administrators protect the server and its services before going into production, using the already available protection mechanisms that Linux distributions offer. The document describes widely used practices that have stood the test of time and that offer general security in the Linux working environment.

Rezime

Ovaj dokument sadrži smernice i preporuke kojima se sistem administratori mogu voditi prilikom inicijalne instalacije i održavanja Linux operativnih sistema u serverskom okruženju. Cilj dokumenta je da se administratorima pomogne da zaštite server i servise, oslanjajući se na već dostupne sisteme zaštite koje Linux distribucije nude. Dokument opisuje široko rasprostranjene prakse koje su izdržale test vremena i koje pružaju opštu sigurnost u radu Linux servera.

Uvod

Linux predstavlja najčešći izbor kada je u pitanju odabir operativnog sistema za serversko okruženje. Granularnost i fleksibilnost u podešavanju, visoke performanse, pouzdanost i sigurnost su neke od komparativnih prednosti u odnosu na druge operativne sisteme. Najveći broj servisa koje akademske institucije nude svojim korisnicima se nalaze upravo na serverima sa Linux operativnim sistemom. Usled ograničene infrastrukture vrlo često se na jednom serveru nalazi više servisa što povećava izazov zaštite Linux servera. Od sistem administratora se očekuje da zaštite server od potencijalnih malicioznih aktivnosti koje bi mogle da ugroze ili kompromituju rad servisa. Ipak, zaštita Linux servera nije jednokratni proces već predstavlja proces koji traje dokle god je server u upotrebi i ovaj dokument sadrži smernice kojima se administrator može voditi prilikom postavljanja zaštite na Linux serveru u cilju povećanja sigurnosti i lakše detekcije problema.

1 Instalacija Linux servera

Linux operativni sistem je dostupan u velikom broju formata - takozvanih „distribucija“. Neke od najčešće korišćenih distribucija su Red Hat (Fedora, CentOS, Mandriva), Debian (Linux Mint, Ubuntu), OpenSUSE ili Arch Linux. Izbor distribucije najčešće zavisi od prethodnog iskustva administratora ili dostupne dokumentacije. Linux distribucije za servere uglavnom razlikuju sledeće instalacije:

- Standardna server instalacija
- Specifična server instalacija
- Minimalna instalacija.

Sistem administrator može koristiti bilo koju od ovih instalacija za svoje potrebe ali se u zavisnosti od izbora mora sprovesti određen skup koraka kako bi se server zaštitio od mogućih zloupotreba u radu. Osnovna ideja u cilju zaštite Linux servera jeste da sistem administrator kontroliše rad celog servera i koristi samo one pakete koji su neophodni za rad planiranih servisa.

Standardnom server instalacijom se dobija Linux operativni sistem sa velikim brojem najčešće korišćenih paketa (aplikacija). Ovo je najpogodnija instalacija ukoliko je server predviđen za rad više servisa. Ipak, više instaliranih paketa upućuje i na veći broj aplikacija koje se ne koriste, a mogu biti iskorišćene kako bi se narušila sigurnost servera. Ukoliko administrator koristi ovu instalaciju, neophodno je sačiniti spisak nepotrebnih paketa za rad servera i nakon instalacije ih isključiti ili ukloniti.

Specifična server instalacija nudi mogućnost instalacije Linux operativnog sistema koji sadrži one pakete koji su neophodni da bi se server koristio u usko određenim implementacijama, npr. veb server, server sa bazom podataka i sl. Ovakva instalacija je pogodna za servere koji će nuditi jedan osnovni servis. Nakon specifične server instalacije administrator treba da proveri spisak instaliranih paketa i ukloni one koji se neće koristiti u radu servera.

Minimalna instalacija Linux distribucije podrazumeva instalaciju osnovnih paketa neophodnih za rad operativnog sistema. Administrator treba da ima jasnu sliku koje servise želi da koristi na serveru i nakon instalacije operativnog sistema posebno instalira neophodne pakete. Ovakav vid instalacije je najpogodniji sa stanovišta sigurnosti ali zahteva i najviše poznavanje rada planiranih servisa i Linux operativnog sistema.

1.1 Isključivanje nepotrebnih servisa

Server mora imati jasno definisanu ulogu u mreži kako bi se napravila adekvatna strategija zaštite. Nakon definisanja svih operacija koje server treba da vrši potrebno je sastaviti listu svih servisa (paketa) koji su

neophodni da bi server vršio planiranu funkciju. Svi servisi koji nisu na listi neophodnih mogu se smatrati kandidatima za isključivanje ili uklanjanje.

Veoma je važno da se administrator upozna sa radom svih servisa na svom Linux serveru. Ukoliko se ne razumeju osnove rada nekog servisa, onda se ne mogu pojmiti ni posledice na sigurnosnom planu koje servis može izazvati. Izuzetno je važno da se za svaki pojedinačni servis uporede prednosti koje donosi sa sigurnosnim problemima koje može izazvati. Ovakav pristup zahteva da administrator uloži više vremena u instalaciju i konfiguraciju Linux servera, ali na duže staze dramatično smanjuje vreme održavanja jer se preventivno sprečavaju potencijalni sigurnosni problemi. Svaki dodatni servis može iznedriti novi sigurnosni problem, te je iz ovih razloga neophodno svesti broj korišćenih servisa na minimum koji omogućava sve predviđene funkcije servera.

Lista nepotebnih servisa zavisi od slučaja do slučaja i ne može se precizno definisati. Najčešće se isključuju:

- printer servis (*lpd* i *cups*), ukoliko server ne koristi mrežni štampač
- *BIND* (*named*), ukoliko se server ne koristi kao DNS (*Domain Name System*) server
- *FINGER*, jer je reč o veoma nesigurnom protokolu
- *Inetd*, *xinetd*, ukoliko korišćenje superservera nije zaista neophodno
- *TFTP*, jer je reč o prevaziđenom servisu koji ne pruža dovoljan nivo sigurnosti
- *avahi-daemon*, ukoliko nema potrebe da server automatski detektuje druge uređaje na mreži
- *bluetooth*, ukoliko se ne koristi ova tehnologija na serveru
- *NFS i svi pripadajući servisi kao što su nfsd, mountd, lockd, statd, nfslock ili portmap*
- *telnet, jer ne obezbeđuje dovoljan nivo sigurnosti u prenosu informacija*
- *FTP* ili *MTA* (*Mail Transfer Agent*), ukoliko se ovi servisi ne koriste na serveru.

Administrator može dobiti listu pokrenutih servisa sledećom komandom:

```
#service --status-all
```

Pregledom liste se utvrđuje u kom statusu se trenutno nalaze servisi. Nakon što administrator identifikuje nepotrebne aktivne servise može ih ugasiti komandom:

```
#service IME_SERVISA stop
```

Ovom komandom se servisi trenutno zaustavljaju, ali je neophodno osigurati da se nepotrebni servisi neće pokrenuti prilikom ponovnog pokretanja (*reboot*) operativnog sistema. U Red Hat distribucijama se to postiže ***chkconfig*** komandom:

```
#chkconfig $IME_SERVISA off
```

U Debian distribucijama se koristi *update-rc.d* komanda:

```
#update-rc.d -f $IME_SERVISA remove
```

Ovim komandama se ne uklanjaju paketi odnosno servisi već se samo manipuliše skriptama koje uređuju pokretanje servisa prilikom ponovnog pokretanja Linux operativnog sistema (*start-up* skripte). Bilo koji od ovih servisa se može ponovo pokrenuti ukoliko se ukaže potreba za tim.

1.2 Uklanjanje nepotrebnih paketa

Pored isključivanja nepotrebnih servisa moguće je primeniti drastičniju meru, a to je uklanjanje paketa koji regulišu ove servise. Uklanjanje predstavlja jednostavniji način jer ne postoji opasnost od ponovnog pokretanja servisa nakon ponovnog pokretanja operativnog sistema. Uklanjanje se preporučuje kada ne postoji apsolutno nikakva potreba za određenim servisima ili u slučajevima kada postoji alternativna opcija koja pruža veći nivo zaštite.

Najveću opasnost po Linux server predstavljaju nesigurni programi za izvršavanje komandi sa udaljenih lokacija. U pitanju su *rexec*, *rlogin* i *rsh* koji su u stručnoj literaturi poznatiji kao „R komande“. Ovi programi prenose korisnička imena, šifre i komande preko mreže bez prethodnog šifrovanja. U slučaju presretanja saobraćaja, informacije su vidljive napadaču što predstavlja ozbiljan sigurnosni problem. Preporučuje se da se ovi paketi uklone sa Linux severa i umesto njih koriste programi i protokoli koji koriste šifrovanu komunikaciju, npr. *SSH*.

Administrator servera treba dobro da razmisli da li je neophodno da pojedini protokoli kao što su *telnet* i *FTP* (*File Treansfer Protocol*) imaju i klijentsku i serversku podršku. Ukoliko je potrebno preuzimati fajlove sa udaljenih *FTP* servera dovoljno je imati samo *FTP* klijent. Pokretanje sopstvenog *FTP* servera koji se ne koristi predstavlja dodatnu opasnost od napada na Linux server.

1.3 *Kickstart/Preseed* instalacija sa specifičnim potrebama

Većina distribucija se isporučuje sa alatima koji administratorima omogućavaju automatizaciju instalacionog procesa. Dva najpoznatija mehanizma su *Preseed* za Debian distribucije i *Kickstart* za Red Hat i njegove derivate. Kickstrat/Preseed instalacija omogućava administratoru da kreira konfiguracioni fajl u kome će se nalaziti odgovori na sva pitanja koja se postavljaju tokom instalacije (uključujući npr. i sigurnosna podešavanja). Ovo omogućava administratoru da kreira uniformne Linux instalacije za određene potrebe u svojoj mreži. Konfiguracioni fajl se generiše ručno (iz tekst editora) ili pomoću grafičkih alata kao što je *Kickstart configurator*. Primer *Kickstart* konfiguracionog fajla koji odgovara datoj instalaciji se na Red Hat i srodnim distribucijama može naći u */root/* direktorijumu pod imenom *anaconda-ks.cfg*. Lokacija na kojoj se čuva konfiguracioni fajl (USB Flash, adresa FTP servera ili putanja unutar modifikovanog ISO diska) se zadaje kao *boot* parametar prilikom pokretanja instalacionog medijuma.

2 Menadžment Linux operativnog sistema

Administrator Linux servera treba da postavi strategiju upravljanja Linux operativnim sistemom sa posebnim osvrtom na sigurnost servera. Neophodno je definisati na koji način se administrator povezuje na server radi održavanja Linux operativnog sistema, potom siguran transfer fajlova ukoliko je to potrebno, zatim ažuriranje paketa i primena eventualnih zakrpa. Generalna praksa koju administrator treba da sprovodi je šifrovanje podataka koji se prenose preko mreže gde god je to moguće.

2.1 Sigurni protokol za udaljeni pristup (SSH ili Telnet)

Sistem administratori najčešće koriste *telnet* i *SSH* protokole za udaljeni pristup Linux serveru. *Telnet* je u većini napušten budući da ne šifrira podatke, uključujući korisničke kredencijale, koji se razmenjuju sa serverom. Bilo ko na mreži između računara administratora i servera koji komuniciraju može presresti pakete i doći u posed korisničkih kredencijala, a samim tim i do potpune kontrole servera. Ovi razlozi su doveli do opšte preporuke da se *telnet* ne koristi već se preporučuje *SSH* protokol.

SSH protokol nudi zaštitu u komunikaciji sa serverom koristeći sisteme za šifrovanje sa asimetričnim ključevima. Sva komunikacija između *SSH* klijenta i *SSH* servera je šifrovana i nerazumljiva za treća lica koja mogu presresti pakete koji se razmenjuju. Autentifikacija *SSH* klijenta se može obaviti šifrovanim prenosom korisničkih kredencijala ili korišćenjem ručno generisanih asimetričnih ključeva pri čemu za autentifikaciju nije neophodno korišćenje lozinke. Ukoliko se autentifikacija vrši pomoću ručno generisanih asimetričnih ključeva bez prenosa lozinke, neophodno je smestiti autentifikacioni ključ na uređaj sa koga će biti pristupano Linux serveru. Za prijavu korisnika (administratora) na server se preporučuje korišćenje šifrovanog prenosa korisničkih kredencijala budući da korisnici mogu menjati računare sa kojih se udaljeno prijavljuju na Linux server. Za komunikaciju i međusobnu autentifikaciju dva Linux servera, preporučuje se korišćenje ručno generisanih asimetričnih ključeva jer su strane u komunikaciji fiksne i izbegava se zapisivanje lozinke u fajlovima na serveru.

2.2 Sigurni protokoli za razmenu fajlova (SCP/SFTP ili FTP)

Ukoliko je neophodno obezbediti razmenu fajlova između Linux servera i druge udaljene mašine potrebno je obezbediti sigurne kanale komunikacije. *FTP* protokol je najpopularniji za transfer fajlova ali pati od istih sigurnosnih rizika kao i *telnet* (šifre se prenose u čitljivom formatu, mogući su „*man-in-the-middle*“ napadi i sl.). Kao alternativa se preporučuju sigurnije varijante *FTP* protokola: *FTPS*, *SFTP* ili *SCP* protokol za prenos fajlova.

SFTP (*Secure File Transfer Protocol* ili *SSH File Transfer Protocol*) ne treba mešati sa protokolom *Simple File Transfer Protocol*. *SFTP* je protokol koji je zamišljen kao nadogradnja *SSH* protokola i bazira se na njemu. Putem *SFTP* protokola se uspostavlja sigurna šifrovana komunikacija sa udaljenim uređajem kako bi se omogućio pristup, upravljanje i prenos fajlova. Protokol koristi *SSH* tunel kako bi vršio manipulaciju fajlovima na udaljenom serveru.

FTPS (*File Transfer Protocol Secure*) je nadogradnja *FTP* protokola koja se bazira na *TLS* (*Transport Layer Security*) i *SSL* (*Secure Sockets Layer*) sigurnim protokolima koji omogućavaju šifrovan prenos podataka. *FTPS* je u mnogome sličan *HTTPS* (*Hyper Text Transfer Protocol Secure*) protokolu i takođe zahteva upotrebu digitalnog sertifikata na serveru. Druga strana u komunikaciji mora verovati instaliranom digitalnom sertifikatu na serveru što može predstavljati problem u implementaciji celokupnog rešenja. Ukoliko se transfer fajlova koristi za potrebe sistem administratora i administriranje servera (npr. prenos log fajlova na udaljeni log server) onda se preporučuje *SFTP* protokol jer nudi više opcija za upravljanje fajlovima. Ukoliko se transfer fajlova sa Linux servera koristi kao servis za krajnje korisnike onda se preporučuje *FTPS* jer nudi autentifikaciju servera korišćenjem digitalnih sertifikata. *FTPS* koristi *tcp/udp* portove 989 i 990 za uspostavljanje konekcije i prenos fajlova pa je neophodno da ovi portovi budu otvoreni na Linux serveru.

SCP (*Secure Copy Protocol*) je protokol koji koristi *SSH* protokol kako bi se osigurala autentifikacija i šifrovanje prenetih podataka. Mehanizam je veoma sličan *SFTP* protokolu ali *SFTP* protokol ima mnogo veće mogućnosti od prostog prenosa fajlova. *SCP* koristi konekciju na *tcp* portu 22 kako bi se izvršio siguran prenos fajlova.

2.3 Sigurni protokoli za pristup veb serveru (HTTP vs. HTTPS)

Veb server je jedna od najčešćih implementacija na Linux serverima. Veb stranice danas sve više zahtevaju autentifikaciju korisnika i razmenu osetljivih podataka što čini *HTTP* protokol neprikladnim budući da ne vrši šifrovanje kredencijala i podataka koji se razmenjuju. Ukoliko se javi potreba za autentifikacijom i razmenom osetjlivih podataka, administratorima se preporučuje implementacija isključivo *HTTPS* protokola.

Implementacija *HTTPS* protokola na Linux veb serveru zahteva pribavljanje digitalnog serverskog sertifikata. Administratori imaju izbor da samostalno kreiraju samopotpisani digitalni sertifikat ili da digitalni serverski sertifikat pribave od zvaničnog sertifikacionog tela. Kreiranje samopotpisanog digitalnog sertifikata je jednostavno i može se obaviti veoma brzo na Linux serveru pomoću *openssl* alata. Problem samopotpisanog digitalnog sertifikata je što takvom sertifikatu ne veruje nijedan od Internet pregledača koje krajnji korisnici koriste. Prilikom pristupa veb serveru sa samopotpisanim digitalnim sertifikatom, korisnicima će Internet pregledač prikazati upozorenje da se veb serveru ne može verovati. Ovakvo obaveštenje je opasno iz dva razloga: korisnici mogu biti zbunjeni ovom porukom i odustati od pristupa veb serveru ili mogu razviti potencijalno opasnu naviku da ignorišu ovakva upozorenja. Samopotpisane digitalne sertifikate treba primenjivati samo u slučajevima kada administratori nisu u mogućnosti da pribave digitalni sertifikat od zvaničnog sertifikacionog tela kome veruju svi Internet pregledači.

Pri implementaciji *HTTPS* protokola dobru praksu predstavlja preusmeravanje svih *HTTP* zahteva na *HTTPS*, odnosno preusmeravanje saobraćaja sa porta 80 na port 443.

2.4 Korišćenje udaljenih fajl sistema

Često je potrebno da serveri poseduju dodatni prostor gde se mogu smeštati podaci koji se prikupljaju tokom rada servera. Administratori pribegavaju upotrebi udaljenih fajl sistema kako bi proširili kapacitet svojih servera. U tu svrhu se Linux serveri povezuju na udaljeni server za skladištenje podataka i „pozajmljuju“ deo fajl sistema. Udaljenim fajl sistemima se pristupa preko mreže, a sistem administratori imaju utisak kao da je udaljeni fajl sistem integralni deo Linux servera. Prenos podataka na relaciji „server – udaljeni fajl sistem“ mora biti adekvatno zaštićen kako ne bi došlo do kompromitovanja podataka koji se prenose. Sistem administratorima se preporučuje upotreba *SSHFS* (*Secure Shell File System*) klijenta koji omogućava korišćenje udaljenih fajl sistema na lokalnom Linux serveru.

SSHFS koristi *SSH* konekciju kako bi se izvršio siguran prenos podataka preko mreže. *SSHFS* u osnovi koristi *SFTP* protokol koji pruža siguran pristup, prenos podataka i upravljanje fajlovima na udaljenom fajl sistemu.

Nakon instalacije *SSHFS* klijenta na Linux serveru neophodno je kreirati direktorijum kome će biti pridružen udaljeni fajl sistem:

```
#mkdir /mnt/additional_storage
```

Sledeći korak bi bio vezivanje udaljenog fajl sistema i lokalnog kreiranog direktorijuma na Linux serveru:

```
#sshfs username@192.168.0.1:/home/spare_storage/ /mnt/additional_storage
```

Od administratora će biti zatraženo da upiše *SSH* šifru udaljenog fajl sistema. Nakon uspešnog pridruživanja, sav sadržaj na udaljenom fajl sistemu (IP adresa 192.168.0.1) u direktorijumu */home/spare_storage/* će biti vidljiv na lokalnom Linux serveru kao integralni deo lokalnog fajl sistema.

2.5 Ažuriranje sistema

Dobra obaveštenost je ključ uspešne zaštite Linux servera. Sistem administrator treba da obezbedi pouzdan izvor informacija o sigurnosnim trendovima Linux distribucije i instaliranih softverskih paketa. Vremenom se otkrivaju sigurnosni propusti u softverskim paketima koje napadači mogu iskoristiti kako bi kompromitovali server. Praćenje imejling lista koje se tiču sigurnosti u Linux distribucijama i softverskim paketima omogućava administratoru da pravovremeno reaguje i na vreme ažurira sistem, a samim tim ukloni sigurnosne propuste u Linux serveru. Preporučuje se praćenje imejling lista odgovarajućih Linux distribucija (Red Hat, Debian, SuSE i sl.) koje se bave sigurnošću sistema. Na osnovu prikupljenih informacija, administrator može odlučiti o vremenu i načinu ažuriranja sistema i softverskih paketa koji se koriste na serveru.

Za instalaciju i ažuriranje softverskih paketa preporučuje se upotreba paket menadžera kao što su *yum* (Red Hat distribucije) ili *apt-get* (Debian distribucije) jer omogućavaju jednostavniju instalaciju softvera i dodatnih neophodnih komponenti (*dependencies*). Pored toga olakšano je i uklanjanje softverskih paketa. Paket menadžeri automatski proveravaju digitalni potpis paketa i ne obavljaju instalaciju ukoliko potpis nije ispravan. Administratori treba da koriste zvanične repozitorijume koji su preporučeni od strane proizvođača. Preuzimanje sa ostalih repozitorijuma se može obavljati ali uz posebne mere opreza. Generalna preporuka je da se pre svakog ažuriranja prikupe informacije o novijoj verziji paketa kako bi se proverila stabilnost i moguće implikacije na sigurnosnom nivou. Ukoliko je neophodno instalirati softver koji se ne može naći u zvaničnom repozitorijumu, preporučuje se preuzimanje softverskog paketa direktno sa veb stranice proizvođača i obavezna provera

digitalnog potpisa. Na produkcioni Linux server nikako ne treba instalirati nezvanične pakete ili pakete koji su fazi testiranja. Dobru praksu predstavlja i čuvanje liste instaliranih paketa i njihovih verzija nakon svakog ažuriranja. Na ovaj način administratori imaju uvid u poslednje stabilne verzije softvera koje su se koristile na serveru. Ukoliko noviji paket ugrozi stabilnost servera, administrator u listi može pronaći raniju stabilnu verziju softverskog paketa i ponovo je instalirati.

Kreiranje fajla sa datumom i listom instaliranih paketa u Red Hat distribucijama:

```
$ yum list installed > pkg-installed-list-`date "+%Y-%m-%d"`.txt
```

Kreiranje fajla sa datumom i listom instaliranih paketa u Debian distribucijama:

```
$ apt-get --list > pkg-installed-list-`date "+%Y-%m-%d"`.txt
```

Administratori mogu podesiti automatsko ažuriranje paketa ali se ono generalno ne preporučuje. Bolja praksa je da se svako ažuriranje paketa pojedinačno verifikuje. Administrator može podesiti paket menadžer da šalje periodične imejlove u kojima se navode svi paketi koji se mogu ažurirati na serveru. Nakon što dobije listu paketa, administarator može za svaki paket pojedinačno doneti odluku o ažuriranju. Podešavanje periodičnog slanja imejlova u *yum* paket menadžeru se vrši u */etc/yum/yum-updatesd.conf* konfiguracionom fajlu. Pored vremenskog intervala za proveru neophodno je definisati i sledeću liniju koda u konfiguracionom fajlu:

```
# how to send notifications (valid: dbus, email, syslog)
emit_via = email
# who to send email to
email_to = linux-administrator@example.org
```

Nakon prepravljanja konfiguracionog fajla, neophodno je ponovo pokrenuti *yum-updatesd* servis.

Administratorima Debian distribucija se preporučuje instaliranje i korišćenje *apticron* softverskog paketa kako bi se uredila automatska provera ažurnosti paketa i obaveštavanje putem imejla. U konfiguracionom fajlu *apticron* softvera neophodno je navesti imejl adresu na koju će obaveštenja biti poslata.

Ukoliko administratori imaju dovoljno iskustva i mogu precizno da definišu potrebne pakete na serverima, dobra praksa je kreiranje sopstvenog repozitorijuma u mreži. Na jednom serveru se može krerati repozitorijum sa dobro poznatim, stabilnim verzijama paketa, a ostali Linux serveri u mreži ih mogu preuzimati. Ovakvo rešenje podrazumeva i detaljnu proveru svih paketa pre smeštanja u sopstveni repozitorijum.

2.6 Broj servisa na serveru

Linux serveri mogu istovremeno obavljati nekoliko funkcija i pružati više servisa. Ovakvo uređenje mreže dovodi do racionalizacije i maksimalnog iskorišćenja serverske infrastrukture. Ipak, sistem administratori treba da se drže zlatnog pravila sigurnosti: „Sistem je siguran koliko i najranjiviji servis na njemu“. Idealno rešenje bi bilo da svaki Linux server pruža jedan servis krajnjim korisnicima. Pošto je ovakav scenario teško ostvariv, neophodno je pažljivo planiranje izgradnje celokupne arhitekture servera u jednoj mreži. Sistem administratori treba da definišu servise koje će pružati krajnjim korisnicima, izvrše prioritetizaciju planiranih servisa, potom da definišu skup raspoloživih servera i na kraju izvrše raspodelu servisa na raspoložive servere. Prilikom raspodele servisa na raspoložive servere, najkritičnije servise bi trebalo održavati na Linux serverima posebno

ili u kombinaciji sa servisima koji su ocenjeni kao najsigurniji. Manje kritični servisi mogu deliti jedan server ili mogu biti rapoređeni u kombinaciji sa manje sigurnim servisima. Cilj planiranja je da se spreči situacija u kojoj bi probijanje jednog nesigurnog servisa dovela do probijanja servera i kontrole nad kritičnim servisom koji administrator održava. Generalna preporuka je da veb, imejl i *SQL* servisi budu međusobno razdvojeni na pojedinačnim serverima.

3 Alati za povećanje sigurnosti

Alati za povećanje sigurnosti se koriste kako bi se sprečio neovlašćeni pristup servisima i maliciozni pokušaji napada na Linux server. Linux poseduje ugrađene sigurnosne alate kao što su *Netfilter* ili *SE Linux*. *Netfilter* je radni okvir za presretanje i obradu mrežnih paketa koji se može naći na gotovo svim Linux distribucijama sa 2.4 ili kasnijom verzijom kernela. Uveden je kao zamena za starije alate za zaštitu mreže *ipchains* i *ipfwadm*. *Netfilter* obezbeđuje skup administrativnih alata među kojima je najzastupljeniji *iptables*.

3.1 Iptables

Iptables alat poseduje pet tabela od kojih svaka definiše poseban skup operacija nad mrežnim paketima:

- filter - vrši filtriranje paketa, najčešće se koristi
- nat - vrši translaciju adresa
- mangle - vrši specifične obrade paketa (npr. promene TOS i TTL polja)
- raw – vrši obrade paketa koje prethode i/ili zaobilaze ostale definisane tabele
- security – vrši filtriranje paketa prema *MAC* (*Mandatory Access Control*) kontroli pristupa.

Tabele sadrže predefinisane nizove pravila (*chains*). Pravilo se sastoji od uslova koji paket treba da ispuni i akcije koja se izvršava ukoliko je uslov ispunjen. Za pakete koji ne ispunjavaju nijedan uslov unutar niza pravila primenjuje se podrazumevana akcija za taj niz (tzv. polisa). Polise su za razliku od običnih pravila ograničene samo na dopuštanje ili odbacivanje paketa. Tačna sintaksa i redosled izvršavanja tabela i nizova se mogu pronaći u *man* stranama i zvaničnoj *Netfilter* dokumentaciji. Korišćenje iptables alata zahteva *root* privilegije, a najčešće primene su: filtriranje paketa, adresne translacije, evidentiranje saobraćaja, direktne promene pojedinih polja u zaglavljima paketa ili balansiranje saobraćaja. Treba napomenuti da se za IPv6 saobraćaj koristi *ip6tables* alat.

3.1.1 Sigurnosne preporuke za iptables

Sigurnosne preporuke koje su ovde date se fokusiraju na kriterijume po kojima se neželjeni saobraćaj detektuje i odbacuje. Opšta preporuka je da konfiguracija mrežne zaštite bude što striktnija jer se time smanjuje mogućnost eksploatacije eventualnih sigurnosnih propusta u servisima na serveru. Strane u komunikaciji treba opisati što je preciznije moguće, a to se prvenstveno odnosi na izvorišne i odredišne *IP* adrese i *tcp/udp* portove. Pakete treba identifikovati i prema drugim poljima zaglavlja ukoliko je njihov sadržaj unapred poznat. Na primer, *ICMP* nema portove ali se paketi za tu vrstu saobraćaja mogu ograničiti prema tipu i kodu poruka koje prenose.

Iptables nudi mogućnost definisanja pravila za manipulaciju paketa u zavisnosti od strane koja je inicirala konekciju. Ovu funkcionalnost obezbeđuje *connection tracking* modul (*conntrack*). U slučaju složenijih protokola kao što su *FTP*, *SIP* i *H.323* preporučuje se korišćenje pomoćnih *conntrack* modula (*connection tracking helpers*). U suprotnom bi npr. aktivna *FTP* sesija zahtevala da klijent nezavisno od kontrolne konekcije otvori ka sebi sve *tcp* portove veće od 1023.

Pre konačne implementacije *iptables* konfiguraciju treba detaljno testirati. Složenija ili manje jasna pravila treba proširiti komentarima (npr. **-m comment --comment "Administrativni opseg IP adresa"**). Srodna pravila se mogu grupisati u korisnički definisane nizove (*user-defined chains*) jer grupisanje popravlja čitljivost i olakšava održavanje *iptables* konfiguracije.

3.1.2 Najčešći propusti *iptables* konfiguracije

Poredak pravila unutar niza je bitan. Pravila sa strinktnijim uslovima treba smestiti bliže početku niza jer u suprotnom postoji opasnost da paketi nikad ne stignu do njih. Sve izmene u *iptables* konfiguraciji treba eksplicitno sačuvati jer se sadržaj tabela nepovratno gubi prilikom ponovnog pokretanja sistema. Za to mogu poslužiti komande kao što je **/etc/init.d/iptables save** na Red Hat distribucijama ili univerzalne **iptables-save** i **iptables-restore** komande. Pre brisanja filter tabele (**iptables -F**) treba proveriti polise za nizove koji se u njoj nalaze kako se ne bi onemugićio željeni saobraćaj. Složenije konfiguracije zahtevaju poznavanje redosleda po kojem paketi prolaze kroz tabele i nizove, pa je potrebno proveriti da li se, na primer, filtriranje po destinacionom portu obavlja pre ili nakon preusmeravanja.

3.2 Alati za zaštitu od *ARP spoofing* napada

Uređaj koji primi *ARP* odgovor će ažurirati svoju *ARP* tabelu nezavisno od toga da li je za taj odgovor prethodno poslat zahtev. *ARP* ne poseduje mehanizme za proveru autentičnosti poruka koje se razmenjuju, pa je ovakvo ponašanje otvorilo mogućnost za tzv. *spoofing* (*poisoning*) napade u kojima napadač nameće svoju MAC adresu kao odredišnu adresu za druge uređaje u mreži. *ARP spoofing* se obično koristi kao uvod u složenije *man-in-the-middle*, *denial-of-service* i *session hijacking* napade. Idealne zaštite od ovog tipa napada nema. Preporuke koje su ovde date mogu pomoći da se on detektuje i ograniči. Najsigurniji vid zaštite su statički ulazi u *ARP* tabeli koji se obično zadaju u */etc/ethers* fajlu. Međutim, ovakvo rešenje je nepraktično u mrežama sa velikim brojem uređaja. Problem se može ublažiti ako se statički zapisi konfigurišu samo za bitne uređaje kao što je npr. *default gateway*. *Arptables* alat se najčešće koristi da ograniči parove MAC i IP adresa sa kojima server želi da komunicira.

Nije retka pojava da se u mrežama adrese dinamički dodeljuju krajnjim korisničkim uređajima, što može biti dobro okruženje za *ARP spoofing* napad. *Antidote* je aplikacija koja po prijemu novog *ARP* odgovora proverava da li je prethodno naučena MAC adresa i dalje aktivna u lokalnoj mreži. Ukoliko jeste, nova MAC adresa se stavlja u listu zabranjenih adresa. Glavni nedostatak ovog algoritma je što se oslanja na činjenicu da je trenutni zapis u *ARP* tabeli legitiman što ne sprečava mogućnost utrkivanja između napadača i pravog odredišta prilikom kreiranja novih zapisa. Za nadgledanje *ARP* aktivnosti može poslužiti *arpwatch*. Ova aplikacija održava lokalnu bazu parova IP i MAC adresa i u slučaju nekakvih promena obaveštava administratora putem *syslog* i imejl poruka.

3.3 Alati za zaštitu od *brute force* napada

Zaštita servera od *brute force SSH* napada se može ostvariti *iptables* alatom ograničavanjem broja dolaznih konekcija u definisanom periodu vremena (*rate-limiting*). Međutim, postoje situacije kada napadi nisu tako očigledni i kada je cilj napada iskoristiti ranjivost softvera koji je instaliran na Linux serveru. Obezbeđivanje servera u tim situacijama zahteva:

- Detekciju brute force napada
- Blokiranje izvora napada.

Fail2Ban je interesantan alat koji omogućava veliku fleksibilnost pri podešavanju zaštitnih mehanizama. *Fail2Ban* prati informacije koje se upisuju u definisane log fajlove i na osnovu specifičnih log linija koje ukazuju na napade aktivira blokade u *iptables* ili *TCP wrapper* alatu (objašnjen u sekciji 3.4). Pored detekcije napada i aktiviranja blokade, *Fail2Ban* može poslati imejl sistem administratoru o svim akcijama koje se sprovode, pa administrator dobija informaciju o napadu u momentu kada se napad odigrava. Dodatna prednost je što alat sadrži predefinisane obrasce za zaštitu veb, imejl, *asterisk* i drugih aplikacija.

Implementacija *Fail2Ban* alata zahteva određenu pripremu. Sistem administrator treba da uredi logovanje informacija softvera koji želi da zaštiti i da locira log fajl u koji će se informacije upisivati. Potom je neophodno analizirati log informacije u fajlu i izdvojiti one koje ukazuju na napad. Izdvajanje ovih log linija nije jednostavno budući da vektori napada mogu biti različiti, ali će administrator najčešće upotrebiti iskustva iz ranijih slučajeva napada i tako doći do onih log linija koje ukazuju na pokušaj ugrožavanja rada aplikacije. Nakon toga se *Fail2Ban* alat konfiguriše da u log fajlu pretražuje ove linije, broji koliko puta se ponavljaju u određenom vremenskom intervalu i nakon toga konfiguriše blokiranje na određeni vremenski period IP adresu koja stoji iza ovih napada. Parametri kao što su broj ponavljanja log linija, vremenski interval u kome se one ponavljaju i period na koji se IP adresa blokira su konfigurabilni. Treba napomenuti da log linije koje ilustruju pokušaj napada treba da imaju informaciju o IP adresi napadača kako bi se mogla aktivirati blokada IP adrese. Najčešća primena Fail2Ban alata je blokiranje IP adrese nakon nekoliko neuspešnih prijavljivanja na veb server.

Fail2Ban treba koristiti sa određenom rezervom budući da su mogući tzv. *injection* napadi koji podrazumevaju da napadač kreira lažni IP paket u kojem će se predstaviti sa IP adresom žrtve. Generisanjem velikog broja ovakvih paketa, napadač može uzrokovati situaciju u kojoj će *Fail2Ban* konfigurisati blokiranje IP adrese žrtve. Žrtva može biti npr. sistem administrator ili neki drugi regularni korisnik servera. Upravo iz ovih razloga se administratorima preporučuje da u situacijama u kojima je to izvodljivo prvo onemogući pristizanje lažnih IP paketa u lokalnu mrežu (sprečavanje *spoofing* napada). Administrator potom treba da identifikuje sve IP adrese koje su bitne za rad servisa koji se štiti *Fail2Ban* alatom. Na kraju je potrebno iskoristiti funkcionalnost *Fail2Ban* alata koja omogućava definisanje skupa IP adresa koje nikada neće biti blokirane. Ovo je omogućeno definisanjem „*ignoreip*“ opcije u */etc/fail2ban/jail.conf* konfiguracionom fajlu alata.

DenyHosts je sličan alat ali se njegova primena ograničava samo na *SSH* pristup, pa se administratorima preporučuje *Fail2Ban* budući da mogu uspostaviti istovremenu zaštitu više servisa na serveru.

3.4 *TCP wrapper*

TCP wrapper predstavlja alat za kontrolu pristupa određenim demonskim procesima (*daemon*) i servisima na Linux serveru. Alat koristi *libwrap.so* biblioteku kako bi implementirao pravila pristupa. Radi na aplikativnom sloju TCP/IP referentnog modela i unosi dodatnu zaštitu servisa pored korišćenja alata kao što je *iptables*. Da

bi se *TCP wrapper* implementirao na određenom servisu, neophodno je da demonski proces koji pokreće servis podržava *libwrap.so* biblioteku. Provera podrške se vrši pomoću ***ldd*** komande, a na primeru je data provera *sshd* demonskog procesa:

```
#ldd /usr/sbin/sshd | grep libwrap*
```

U datom primeru, */usr/sbin/sshd* je lokacija na kojoj se nalazi *sshd*. Komanda ***ldd*** proverava da li je *libwrap.so* jedna od potrebnih biblioteka za rad demonskog procesa.

```
libwrap.so.0 => /lib64/libwrap.so.0 (0x00002b3f6617c000)
```

Ukoliko je rezultat izvršenja komande kao u primeru iznad, onda se može zaključiti da *sshd* demonski proces podržava *libwrap.so* biblioteku i da se može implementirati *TCP wrapper* zaštita na *SSH* servisu. Sledeći korak bi bio konfigurisanje *TCP wrappera*, a to se vrši izmenama u */etc/hosts.allow* i */etc/hosts.deny* fajlovima. U fajlu *hosts.allow* se definišu mašine koje imaju pravo da pristupe naznačenom servisu, dok se u *hosts.deny* fajlu definišu mašine kojima je eksplicitno zabranjeno da pristupe naznačenom servisu. Treba napomenuti da *TCP wrapper* prvo proverava */etc/hosts.alow* pa tek onda */etc/hosts.deny* fajl. Ukoliko je jedna mašina definisana u oba fajla, prednost ima */etc/hosts.allow* fajl i na osnovu toga se toj mašini dopušta pristup.

```
sshd:192.168.0.0/255.255.255.0 EXCEPT 192.168.0.99
```

Konfigurisanjem gore navedenog reda u */etc/hosts.deny* fajlu, uvodi se zabrana pristupa *SSH* servisu svima na mreži 192.168.0.0/24 osim IP adresi 192.168.0.99. Umesto IP adresa se mogu koristiti *DNS* imena ali se to ne preporučuje jer bi kvar *DNS* servisa onemogućio *TCP wrapper* da razreši imena što bi sva postavljena pravila učinilo ništavnim.

```
tcpdmatch sshd 192.168.0.99
```

Gore navedenom komandom se testiraju postavljena pravila, tako da administrator može proveriti da li su /etc/hosts.allow i /etc/hosts.deny fajlovi konfigurisni na odgovarajući način.

Sistem administratorima se preporučuje korišćenje *TCP wrapper* alata ali nikako kao zamena za *iptables*, već isključivo kao dopuna zaštite pojedinačnih servisa na Linux serveru.

3.5 SE Linux

SE Linux (*Security-Enhanced Linux*) predstavlja sigurnosni modul Linux kernela koji obezbeđuje dodatne mehanizme zaštite u radu servera. Dok standardni Linux kernel koristi restrikcije pristupa objektima na osnovu identiteta korisnika ili grupe (tzv. *Discretionary Access Control*), *SE Linux* uvodi tzv. mandatnu kontrolu pristupa (*Mandatory Access Control*) tj. ograničava privilegije pojedinačnim korisnicima ili procesima na minimalne neophodne za ispravno funkcionisanje.

SE Linux uvodi nove entitete za sve korisnike i procese na serveru: uloga, korisničko ime i domen. Uloga i korisničko ime nemaju nikakvu korelaciju sa standardnim korisničkim nalozima i grupama na Linux operativnom sistemu. Ceo mehanizam dozvole određenih akcija je uređen domenom. Akcije i privilegije na serveru su uređene podelom na specifične domene, a korisnik stiče pravo da ih koristi dozvolom korišćenja domena. Za sve korisnike i procese se definiše *SE Linux* polisa koja uređuje domene koje taj korisnik može da koristi.

Konačan rezultat uređenja polisa za pojedinačne korisnike i procese je ograničavanje delovanja. Ovo je veoma korisno jer se sprečava da, na primer, probijanjem veb servera napadač stekne kontrolu nad ostalim servisima na serveru. *SE Linux* se najčešće koristi za ograničavanje delovanja demonskih procesa i servisa koji pokreću veb servere, imejl servere ili servere baze podataka.

SE Linux zahteva izvesno vreme za razumevanje i sticanje dovoljnih veština za upravljanje. Ograničavanje procesa u radu servera može biti čest izvor nefunkcionalnosti određenih aplikacija na serveru. Iz ovih razloga se administratorima preporučuje da prvo dobro prouče rad *SE Linuxa* i konfigurišu ga na serverima čija je namena pružanje jednog servisa. Nakon što ovladaju veštinama upravljanja ovog alata, mogu ga koristiti i na drugim serverima. *SE Linux* može biti izuzetno moćan sigurnosni alat u rukama obučenog sistem administratora.

3.6 *AppArmor*

AppArmor predstavlja sigurnosni modul Linux kernela koji se može naći na Debian, SUSE i Mandriva distribucijama. Konceptualno je veoma sličan *SE Linux* modulu s tim što je fokus *AppArmor* modula da ograniči delovanje softverskih programa, a ne pojedinačnih korisnika. *AppArmor* funkcioniše formiranjem tzv. profila koji mogu ograničiti delovanje softvera ili samo prijaviti situacije u kojima je softver vršio akcije van svoje nadležnosti. Sistem administratorima se preporučuje da prouče rad *AppArmor* sigurnosnog modula ukoliko procene da im je neophodno ograničavanje delovanja pojedinačnih softverskih programa.

4 Menadžment korisnika

Ubedljivo najviše štete se može naneti kompromitovanjem korisničkih naloga. Sistem administrator mora biti upoznat sa svim korisničkim nalozima na serveru. Zato je neophodno formirati strategiju kreiranja i upravljanja nalozima korisnika koja će ispunjavati sledeće ciljeve:

- Svaki korisnik Linux servera mora biti jednoznačno identifikovan korisničkim nalogom
- Svaki korisnički nalog mora imati dovoljno sigurnu lozinku za autentifikaciju
- Na serveru mora postojati log informacija o prethodno autentifikovanim sesijama korisnika
- Svaki korisnik mora biti ograničen samo na skup neophodnih privilegija.

Administrator se može služiti pojedinim ugrađenim alatima koje Linux operativni sistem poseduje kako bi se automatski primenjivala određena pravila koja vode do povećanja sigurnosti korisničkih naloga.

4.1 Kreiranje politike korisničkih lozinki

Skoro sve Linux distribucije koje su danas u upotrebi skladište lozinke u zamaskiranom formatu (*shadow passwords*). Upotreba zamaskiranih lozinki obezbeđuje da prijavljeni korisnik ne može videti lozinke ostalih korisnika pregledom fajla */etc/passwd*. Administrator treba da proveri da li postoji fajl */etc/shadow* koji ukazuje na korišćenje zamaskiranih lozinki na Linux serveru. Ukoliko Linux server ne koristi zamaskirane lozinke administratoru se preporučuje da obavezno implementira ovo rešenje.

Dobre lozinke su osnova sigurnosti svakog korisničkog naloga. Sistem administrator treba da postavi jasna pravila i ograničenja koja će naterati korisnike da kreiraju jake lozinke.

4.1.1 Konstrukcija korisničkih lozinki

Korisničke lozinke treba da prate jasno ustanovljena pravila uz pomoć kojih se povećava otpornost probijanja. Ukoliko postoji veći broj korisnika Linux servera, dobra praksa je pisanje preporuka za kreiranje lozinki u vidu dokumenta. Dokument bi onda bio podeljen svim korisnicima koji imaju nalog na serveru. Da bi se smatrala dovoljno sigurnom, lozinka mora ispuniti sledeće kriterijume:

- ne sme biti reč ili više spojenih reči
- ne sme imati manje od 8 karaktera
- ne sme biti ime ili prezime
- ne sme biti slična korisničkom imenu ili deo korisničkog imena

- mora imati barem jedno veliko slovo
- mora imati barem jedan broj
- mora imati barem jedan specijalni karakter.

Dobra lozinka se može konstruisati od početnih slova neke omiljene fraze, uz kombinaciju malih i velikih slova, brojeva i specijalnih karaktera. Danas se ovakve lozinke smatraju najtežim za probijanje. Sistem administratori mogu koristiti i alate za proveru jačine lozinki kako bi osigurali da su sve lozinke na Linux serveru dovoljno sigurne. Preporučuje se korišćenje programa „*John the Ripper*“ koji će pokušati da probije lozinke svih naloga i obeležiti one koje su uspešno probijene. „*John the Ripper*“ i slični alati zahtevaju određeno procesorsko zauzeće te se ne preporučuje korišćenje ovog alata u toku vršnih radnih sati servera.

4.1.2 Ograničavanja u procesu kreiranja lozinki

Korišćenjem *PAM* (*Pluggable Authentication Module*) modula mogu se uspostaviti pravila koja ograničavaju korisnike u ponavljanju lozinki ili korišćenju delova ranijih lozinki. *PAM* se može pronaći na skoro svim Linux distribucijama, a ograničenja koja se tiču starih lozinki se mogu postaviti u fajlu */etc/pam.d/system-auth* na Red Hat distribucijama ili u fajlu */etc/pam.d/common-password* na Debian distribucijama. Dodavanjem sledećeg reda u ovim fajlovima pamti se prethodnih 10 lozinki svakog korisnika:

```
password sufficient pam_unix.so use_authtok md5 shadow remember=10
```

Važno je napomenuti da neke Linux distribucije mogu koristit *PAM 2* modul, pa je u tom slučaju neophodno u prethodnoj komandi umesto *pam_unix.so* modula koristiti *pam_unix2.so*. Linux operativni sistem će sve prethodne lozinke korisnika čuvati u */etc/security/opasswd* fajlu. Ukoliko ovaj fajl ne postoji, potrebno je kreirati ga i postaviti privilegije tako da samo *root* nalog može da čita i upisuje podatke u njega.

Sistem administrator može uvesti dodatna ograničenja poput minimalnog broja karaktera koji se moraju razlikovati u odnosu na prethodnu lozinku. Ovakvo ograničenje se uvodi korišćenjem *pam_cracklib.so* modula u */etc/pam.d/system-auth* ili */etc/pam.d/common-password* fajlovima, u zavisnosti od distribucije.

```
password required pam_cracklib.so difok=4 minlen=8 dcredit=-1 ucredit=-1 ocredit=-1 lcredit=0
```

Dodavanje gore navedenog reda u prethodno pomenute fajlove implementira pravilo da lozinka mora biti minimalne dužine 8 karaktera (minlen=8) pri čemu mora postojati barem jedna cifra (dcredit=-1), barem jedno veliko slovo (ucredit=-1) i barem jedan specijalan znak(ocredit=-1), dok bar 4 karaktera moraju biti promenjena u odnosu na prethodnu lozinku (difok=4). Modul *pam_cracklib.so* koristi sistem kreditiranja gde se svakom karakteru dodaje određeni broj poena u zavisnosti od tipa. Ukupan zbir kredita mora odgovarati minimalnoj dužini lozinke. Ukoliko se za kredit navode negativne vrednosti onda se uvode obavezne kvote za određeni karakter. Pored ovih ograničenja, *pam_cracklib.so* modul sprovodi dodatne provere, kao na primer da li je nova lozinka samo obrnuti red karaktera stare lozinke ili da li nova lozinka ima samo promenjenu veličinu slova u odnosu na staru.

4.1.3 Ograničavanje trajanja korisničkih lozinki

Dodatni sigurnosni mehanizam predstavlja vremensko ograničenje trajanja korisničkih lozinki. Periodična promena lozinke za sve korisnike se definiše u */etc/login.defs* fajlu. Komandom ***chage*** se mogu definisati parametri za pojedinačne korisnike.

Informacije o vremenskim atributima lozinke određenog korisnika se mogu dobiti sledećom komandom:

```
#chage -l username
```

Prethodnom komandom će se ispisati vreme kreiranja lozinke, period važenja i broj dana upozorenja pre nego što rok važnosti istekne. Postavljanje upozorenja par dana pre isteka je veoma korisna funkcija, a preporučuje se da period upozorenja traje 2 do 3 nedelje pre konačnog isteka lozinke.

Vremensko ograničenje pojedinačnog naloga se može obaviti komandom:

```
#chage -M 90 -m 7 -W 20 username
```

U gore navedenoj komandi se za korisnika postavlja trajanje lozinke od 90 dana, pri čemu je minimalni period trajanja lozinke 7 dana, a upozorenje o isteku lozinke će početi da se pojavljuje 20 dana pre isteka roka važnosti lozinke. Upozorenje se javlja prilikom prijave korisnika na Linux server. Minimalni period trajanja lozinke je koristan budući da nakon njenog isteka korisnici mogu promeniti lozinku na par minuta, a potom vratiti prethodnu novom promenom. Time se praktično obesmišljava čitava zamisao vremenskog ograničavanja trajanja lozinke. Međutim upotrebom mehanizma u kome je korisniku onemogućeno da koristi staru šifru rešava ovaj problem.

Ukidanje vremenskog ograničenja za određenog korisnika se može odraditi komandom:

```
#chage username
```

Svrsishodnost postavke vremenskog ograničenja je veoma upitna, budući da se dobre lozinke teško kreiraju, a poseban izazov može predstavljati kontinuirano kreiranje jakih lozinki. Česte promene lozinki mogu naterati korisnike da ih zapisuju, a to je veoma rizično ponašanje. Iz tog razloga preovladava mišljenje da je bolje imati jednu jaku lozinku nego često menjati slabe lozinke.

4.1.4 Zaključavanje korisničkih naloga

Linux operativni sistem nudi mogućnost zaključavanja korisničkih naloga ukoliko je pogrešna lozinka uneta predefinisani broj puta. Korišćenjem *PAM* modula *pam_tally.so* može se definisati maksimalni broj pokušaja unošenja lozinke i vreme koje će nalog nakon toga provesti u zaključanom stanju. Zaključani nalog ne može biti iskorišćen za pristup Linux serveru. Primena ovog mehanizma se uvodi izmenom fajla */etc/pam.d/system-auth* na Red Hat distribucijama ili fajla */etc/pam.d/common-password* na Debian distribucijama. Potrebno je dodati sledeće redove:

```
auth required pam_tally.so no_magic_root

account required pam_tally.so deny=5 no_magic_root lock_time=3600
```

Opcija „*no_magic_root*“ definiše da nije moguće zaključavanje *root* naloga. U primeru se postavlja logovanje neuspešnih pokušaja, uvodi se ograničenje do 5 neuspešnih pokušaja, a nakon toga se uvodi zaključavanje naloga na period od 3600 sekundi. Informacije o neuspešnim pokušajima prijave na Linux sistem se mogu dobiti komandom ***faillog***.

```
#faillog -u username

#faillog -a
```

Opcija „*-u*“ daje informacije o preciziranom korisniku, dok opcija „*-a*“ daje informacije o svim korisnicima Linux operativnog sistema. Brojači neuspešnih logovanja se mogu resetovati korišćenjem opcije „*-r*“.

```
#faillog -r
```

Ukoliko se koristi mehanizam automatskog zaključavanja naloga, dobra praksa je povremeno resetovanje brojača. Ubacivanjem gore navedene komande za resetovanje brojača u *crontab*, može se automatizovati ceo proces. Preporučuje se resetovanje svakog dana ili eventualno svake nedelje.

Pored automatskog zaključavanja naloga, moguće je i ručno zaključavanje naloga pomoću ***faillog*** komande.

```
faillog –l 3600 -u username
```

Opcijom „*-l*“ se nalog zaključava na određeni vremenski period u sekundama, dok se opcijom „*-u*“ precizira koji će korisnički nalog biti zaključan.

4.2 Sudo pristup

Root korisnički nalog je veoma moćan i predstavlja tzv. superkorisnika koji ima pristup svim sistemskim procesima i fajlovima u Linux operativnom sistemu. Red Hat distribucije zahtevaju kreiranje *root* naloga prilikom instalacije operativnog sistema, dok Debian distribucije kreiraju *root* nalog ali mu ne dodeljuju lozinku pa je nemoguće prijaviti se na Linux operativni sistem pomoću ovog naloga. Bez obzira na distribuciju koja se koristi, generalna preporuka je da se *root* nalog nikada ne koristi za prijavu na server. Ukoliko više korisnika koristi *root* nalog za prijavu i održavanje servera onda postaje nemoguće pratiti ko je odgovoran za određena podešavanja na serveru. Bolja praksa je da svaki korisnik poseduje korisnički nalog koji će koristiti za prijavu na sistem i administraciju servera. Korišćenjem ***sudo*** komande, regularni korisnički nalozi mogu dobiti privremenu dozvolu da pokreću komande za koje inače nemaju privilegije.

Sudo (*Super User Do*) omogućava privremeno sticanje *root* ili tačno određenih privilegija kako bi se izvršile komande na koje regularni korisnik nema pravo. Podešavanje sudo mehanizma se vrši u */etc/sudoers* fajlu.

Da bi regularni korisnik pokrenuo komandu koja zahteva veće privilegije neophodno je ispred komande dodati ***sudo***. Linux će potom od korisnika zatražiti da upiše šifru kako bi potvrdio svoj identitet i izvršio komandu. Uobičajeno je da korisnik upisuje šifru sopstvenog naloga, ali se */etc/sudoers* fajl može konfigurisati da zahteva upis *root* lozinke. Ovakvo podešavanje može biti korisno jer uvodi još jedan nivo zaštite ukoliko dođe do kompromitovanja određenog korisničkog naloga. Kada *sudo* zahteva *root* lozinku, nije dovoljno znati samo kredencijale običnog korisnika, već i lozinku *root* naloga. Podešavanje ***sudo*** komande sa *root* lozinkom se vrši dodavanjem sledećeg reda u */etc/sudoers* fajlu:

```
Defaults rootpw
```

U zavisnosti kako je konfigurisan fajl */etc/sudoers*, različiti korisnici mogu imati različite privilegije. Pregled svih *root* komandi koje su dozvoljene određenom korisničkom nalogu se mogu dobiti komandom:

```
# sudo -ll
```

Sudo alat je veoma koristan u situacijama kada je neophodno ograničiti uticaj pojedinih korisnika. Dobru praksu može predstavljati podela zaduženja među korisnicima Linux sistema i u skladu sa tim podešavanje *sudo* alata kako bi se osiguralo da nijedan korisnik neće preći svoju zonu odgovornosti.

4.3 Nadgledanje pristupa serveru

Informacije o prijavama korisnika na Linux server mogu biti veoma korisne prilikom eventualnih istraga i utvrđivanja odgovornosti u incidentnim situacijama. Linux operativni sistem ima ugrađene mehanizme praćenja prijave korisnika na server.

Komande ***who***, ***last*** i ***lastb*** daju pregled svih prijava na server, nezavisno da li se korisnik prijavio putem *telnet*, *SSH* ili konzolnog pristupa. S obzirom da je preporučeni mehanizam za udaljeni pristup korišćenje *SSH* protokola, sistem administrator može koristiti i informacije date u log fajlu */var/log/secure* gde se prate svi pokušaji *SSH* pristupa na Linux server. Korisna može biti i ***lastlog*** komanda koja za sve korisnike ispisuje vreme poslednje prijave na sistem.

4.4 Nadgledanje izvršenih komandi

Linux operativni sistem ima ugrađeni mehanizam pregleda ranije izvršenih komandi. Pregled izvršenih komandi se može dobiti korišćenjem komande ***history***.

Komande su poređane po redosledu izvršenja ali nema informacije o tačnom vremenu kada je određena komanda pokrenuta. Da bi se dobila informacija o vremenu, neophodno je u */etc/profile* fajlu dodati i sledeću liniju:

```
export HISTTIMEFORMAT='[%F] [%T] '
```

Komandom ***history*** se ispisuju sopstvene pokrenute komande, međutim sistem administrator ponekad ima potrebu da ustanovi koje komande su drugi korisnici pokretali. Uvid u komande drugih korisnika u Linux operativnom sistemu se mogu videti u skrivenom fajlu *.bash_history* koji se nalazi u početnom (*/home/*) direktorijumu korisnika koji se posmatra.

Napredni korisnici mogu prikriti svoje tragove izmenom *.bash_history* fajla i time podatke o izvršenim komandama učiniti netačnim. Ukoliko Linux server koriste napredni korisnici, dobar alat za praćenje aktivnosti korisnika je *psacct* (*Process Accounting*). *Psacct* alat prati koliko dugo su korisnici bili prijavljeni na sistem i koje komande su izvršavali. Alat je dostupan na svim najčešće korišćenim Linux distribucijama i sistem administratorima se preporučuje njegovo korišćenje ukoliko žele da imaju pouzdane informacije o aktivnostima svih korisnika. Nakon instalacije alata, Debian distribucije automatski pokreću servis nadgledanja dok je na Red Hat distribucijama nakon instalacije neophodno ručno pokrenuti servis.

Alat nudi i neke dodatne interesantne podatke kao na primer potrošnju RAM memorije i procesorsko zauzeće po korisniku i komandama. Korišćenjem ovog alata sistem administrator može steći dobar uvid u aktivnosti korisnika te se zbog toga preporučuje u situacijama kada jedan Linux server administrira više korisnika.

4.5 *LDAP* autentifikacija

Održavanje korisničkih naloga na jednom Linux serveru ne predstavlja preveliki zadatak za sistem administratora. Međutim, ukoliko je potrebno održavati korisničke naloge na više Linux servera moguće je kreiranje haotičnog sistema u kome korisnički nalozi neće biti konzistentni na različitim Linux serverima. Svaki novi Linux server donosi dodatni administrativni posao uređenja pojedinačnih korisničkih naloga. Novi problem

može predstavljati i uklanjanje korisničkog naloga jer se mora obaviti na nekoliko servera. Centralizovani sistem autentifikacije u velikoj meri rešava problem rukovanja korisničkim nalozima na više Linux servera. Svi nalozi se nalaze na jednom mestu i sinhronizovani su na svim Linux serverima. Dodatna prednost uvođenja centralizovane autentifikacije je mogućnost kreiranja pregledne dokumentacije o korisnicima budući da postoji jedna jedinstvena baza svih korisnika koji se povezuju na Linux servere. Za centralizovano upravljanje korisničkim nalozima se preporučuje *LDAP* (*Lightweight Directory Access Protocol*) baza koja bi sadržala sve korisničke naloge koji imaju pravo da se povežu na Linux server.

Linux posmatra korisničke naloge u udaljenoj *LDAP* bazi na identičan način kao i u fajlovima */etc/shadow* ili */etc/passwd*. Za podešavanje *LDAP* autentifikacije na Linux operativnom sistemu koristi se poseban *PAM* modul, *pam_ldap.so*. *SSH* pristup treba biti konfigursan tako da podržava *PAM* module, dok se konfiguracijom *PAM* autentifikacije i *LDAP* servisa na Linux operativnom sistemu konfigurišu parametri komunikacije između Linux servera i udaljenog *LDAP* servera. Konačan rezultat je realizacija sistema u kome će korisnici putem sigurnog *SSH* protokola pristupati Linux serveru, dok će Linux proveravati korisničke kredencijale i prava pristupa u centralizovanoj *LDAP* bazi na udaljenom serveru. U ovakvoj arhitekturi se preporučuje korišćenje LDAPS protokola (*LDAP Secure*) za razmenu informacija o korisničkim nalozima između Linux servera i udaljenog LDAP servera kako bi se omogućio siguran prenos korisničkih kredencijala kroz mrežu.

5 Nadgledanje Linux operativnog sistema

Praćenje rada servera i osluškivanje vitalnih sistemskih informacija predstavlja apsolutan prioritet kako bi se obezbedio stabilan i siguran rad servera. Informacije o radu servera predstavljaju ključ stabilnosti i važan zadatak administratora predstavlja prikupljanje sledećeg skupa podataka:

- Informacije o radu aplikacija na serveru
- Informacije o mrežnoj aktivnosti servera
- Informacije o potrošnji sistemskih resursa
- Informacije o radu operativnog sistema.

Prikupljene informacije mogu ukazati na greške prilikom konfiguracije Linux servera, ali mogu dovesti i do otkrivanja sigurnosnih pretnji po rad servera:

- Neuobičajeni rad aplikacije može otkriti napad koji se trenutno odvija
- Mrežna aktivnost servera može ukazati na napad i otkriti njegov izvor
- Nagli skok potrošnje sistemskih resursa može ukazati na aktivnosti napadača
- Nestabilan rad operativnog sistema može biti pokazatelj probijanja servera.

Kontinuiranim praćenjem rada Linux servera, administrator stiče iskustvo i osećaj za uobičajeno ponašanje servera. Informacije koje ukazuju na nepravilnosti u radu servera treba ispitati jer mogu otkriti ranjivosti sistema ili napad na Linux server.

5.1 Syslog

Log informacije su od neprocenljive vrednosti i mogu ukazati na rane znake kompromitovanja servera. Dodatno, u slučaju sigurnosnih incidenata logovi pružaju važne forenzičke podatke na osnovu kojih se otkrivaju ranjivosti sistema i rekonstruiše tok događaja koji je doveo do ugrožavanja sigurnosti servera. *Syslog* predstavlja nezaobilazan alat u nadgledanju rada servera, a prednost predstavlja činjenica da je ugrađen u svim Linux distribucijama. *Syslog* prikuplja podatke od kernela, lokalnih procesa i aplikacija i potom ih filtrira. Filtriranje log informacija je fleksibilno i omogućava administratoru da odluči koje informacije će beležiti, u kolikom obimu i na kom mestu.

Podešavanje *syslog* alata u Linux operativnom sistemu se uobičajeno vrši u */etc/syslog.conf* fajlu. Linije u konfiguracionom fajlu se sastoje od dva elementa: selektora i akcije. Selektor je konstruisan u formatu „*facility.priority*“ i njime se definiše koji podaci će biti beleženi i u kojoj meri. *Facility* predstavlja kategoriju informacija koja se prikuplja a administratorima se preporučuje da prikupljaju informacije iz sledećih kategorija:

- *Auth* – nosi informaciju o autentifikaciji korisnika i događajima vezanim za sigurnost servera
- *Authpriv* – nosi informaciju o kontroli pristupa serveru
- *Daemon* – nosi informaciju o sistemskim i demonskim procesima
- *Kern* – nosi informaciju o radu kernela.

U zavisnosti od potreba i namene servera mogu se koristiti i druge kategorije, ali ove navedene daju informacije koje mogu biti značajne sa sigurnosnog aspekta. Pored kategorija neophodno je definisati i prioritet poruka koje će biti beležene. *Syslog* razlikuje 8 nivoa prioriteta poruka: *emergencies* (0), *alerts* (1), *critical* (2), *errors* (3), *warnings* (4), *notifications* (5), *informational* (6) i *debugging* (7), pri čemu niža vrednost predstavlja viši prioritet. Treba imati na umu da beleženje velikog broja log poruka utiče na rad servera budući da se deo procesorske moći troši na obradu i beleženje log informacija. Administratorima se preporučuje da koriste bar prioritet nivoa *errors* (3) jer ovaj nivo beleži log informacije koje se tiču grešaka u radu procesa i aplikacija. Greške u radu su pokazatelj nestabilnosti sistema, a izvor nestabilnosti može biti sigurnosni incident. Ukoliko se sumnja da je sistem bezbedonosno ugrožen, administrator može privremeno omogućiti *debugging* (7) nivo čime se beleže sve log poruke i tako stiče puna slika o radu procesa ili aplikacije.

Tipičan red u */etc/syslog.conf* konfiguraciji bi bio:

```
auth.err  /var/log/auth.log
```

Konfiguraciona linija iznad definiše beleženje log poruka o autentifikaciji (*facility*), prioriteta nivoa *errors* (3) i svih viših prioriteta (*priority*), a log informacije se upisuju u */var/log/auth.log* fajl (akcija). *Syslog* se može konfigurisati i tako da se log informacije ispisuju na terminalu svih ili pojedinačnih korisnika koji su prijavljeni na sistem, ali se ovakvo podešavanje preporučuje samo za najviše nivoe prioriteta tj. kada se javljaju poruke o kritičnim greškama koje onemogućavaju rad procesa ili aplikacije. Prednosti ispisivanja log informacija na terminalu jeste što se informacije ne mogu izmeniti, za razliku od rešenja kada se upisuju u log fajl. Osoba koja ima pristup log fajlu može promeniti ili obrisati pojedine informacije iz fajla i time sakriti tragove svojih akcija. Mana ispisivanja log informacija na terminalu je nepregledan prikaz informacija i otežan rad na serveru budući da ispisivanje informacija može prekidati rad sistem administratora.

Neke aplikacije ne koriste *syslog* za prikupljanje log informacija te se zahteva posebno podešavanje u konfiguracionom fajlu. Tipičan primer je *Apache* aplikacija za veb servere gde se logovanje informacija podešava u */etc/httpd/httpd.conf* fajlu ili u */etc/apache2/apache2.conf* fajlu za *Apache2* verziju paketa. Administratorima se preporučuje da obavezno beleže log informacije o radu aplikacija bilo da se to obavlja putem *syslog* alata ili direktno.

5.2 Centralizovano syslog nadgledanje

U generalnoj praksi informacije o radu servera se upisuju lokalno na svakom pojedinačnom serveru. Ukoliko administrator održava veći broj servera, može biti prilično nepraktično paralelno pratiti rad različitih sistema. Dobro rešenje u takvim situacijama može predstavljati slanje log informacija na jednu centralnu lokaciju, tipično jedan Linux server koji je predviđen za čuvanje ovih informacija. *Syslog* podržava slanje log informacija na udaljeni *syslog* server, a podešavanja se vrše u konfiguracionom fajlu */etc/syslog.conf*.

```
*.err  @syslog-server.ac.rs
```

Konfiguraciona linija iznad definiše da se sve *syslog* poruke, prioriteta *errors* (3) i višeg prioriteta šalju na udaljeni server „*syslog-server.ac.rs*“. Udaljeni server takođe mora imati *syslog* alat koji će biti konfigurisan da

prihvata logove sa udaljenih mašina. Administratorima se preporučuje da obavezno ograniče pristup centralnom *syslog* serveru, a najefikasniji način je ograničavanje pristupa korišćenjem *iptables* alata. Centralni *syslog* server uobičajeno prima *syslog* poruke na destinacionom UDP portu 514. Pristup serveru po ovom portu treba biti dozvoljen samo onim mašinama koje koriste usluge centralnog *syslog* servera.

5.3 Syslog-ng

Syslog-ng (*syslog next generation*) je unapređena verzija osnovnog *syslog* alata koja pruža daleko veću programabilnost prilikom obrade logova, pa pored filtriranja i klasifikacije može vršiti i parsiranje ili prepravljanje log informacija. Parsiranje i prepravljanje nude velike mogućnosti prilagođavanja log informacija potrebama sistem administratora jer omogućavaju izbacivanje nepotrebnih podataka ili zamenu pojedinih delova. Prednost koju nudi *syslog-ng* je i mogućnost čuvanja log poruka u *SQL* bazama što omogućava integraciju sa aplikacijama za analiziranje log informacija. Korišćenjem SQL baze za smeštanje log informacija omogućuje se lakši pregled željenih podataka i pravljenje statistike rada pojedinačnih procesa, aplikacija ili čitavih uređaja.

Syslog-ng uvodi poboljšanja na nivou sigurnosti prilikom prenosa log informacija do centralnog *syslog* servera. Pouzdan prenos informacija se obavlja putem TCP protokola, za razliku od nepouzdanog UDP protokola koji se koristi u osnovnom *syslog* alatu. Dodatno se uvodi mogućnost šifrovanog prenosa *syslog* paketa korišćenjem *TLS* (*Transport Layer Security*) protokola što onemogućava napadača da presretne i čita log informacije sa različitih uređaja. Zbog svih navedenih prednosti, sistem administratorima se preporučuje upotreba *syslog-ng* alata u arhitekturama gde se koristi centralizovani monitoring.

Alternativa *syslog-ng* alatu može predstavljati *rsyslog* koji se može naći na većini najčešće korišćenih Linux distribucija. Alat predstavlja unapređenu verziju osnovnog *syslog* alata koja se odlikuje pouzdanim *tcp* prenosom podataka na udaljeni server, čuvanjem poruka u lokalnom baferu u slučaju da udaljeni server nije trenutno dostupan, integracijom sa bazama podataka i dodatnim opcijama za precizno filtriranje log poruka.

5.4 SNMP

SNMP (*Simple Network Management Protocol*) je protokol kojim se prenose podaci o raznim parametrima rada udaljenih mašina. Često se koriste za praćenje obima mrežnog saobraćaja, procesorske opterećenosti, zauzeća RAM memorije ili prostora za skladištenje na udaljenim Linux serverima. Podaci dobijeni putem *SNMP* protokola se konstantno prate i najčešće su prvi pokazatelj promene u radu servera. Nagli skok potrošnje sistemskih resursa je prvi pokazatelj nestabilnosti sistema što navodi administratora da ispita problematični server i proveri logove koji se prikupljaju na njemu. Neretko se događa da sumnja na sigurnosni incident potekne upravo na osnovu podataka prikupljenih putem SNMP protokola.

Danas se najčešće koriste SNMP verzije protokola v2, v2c i v3. Verzija 2 je unela novinu u odnosu na verziju 1 korišćenjem „*Inform*“ poruke umesto „*Trap*“. Obe poruke obaveštavaju udaljenog klijenta ili aplikaciju da se na serveru dogodilo nešto što zahteva posebnu pažnju. Za razliku od poruke „*Trap*“, „*Inform*“ zahteva potvrdu prijema čime se osigurava pouzdan prenos do udaljenog klijenta. Verzija v2c je veoma slična osnovnoj verziji s tim što je pojednostavila sigurnosni aspekt protokola i oslonila se na autentifikaciju putem „*community*“ atributa. Upravo iz razloga jednostavnijeg sigurnosnog modela, v2c je mnogo rasprostranjenija nego osnovna verzija 2.

Verzija 3 je najnovija vezija *SNMP* protokola koja je donela poboljšanja u sistemu sigurnosti uvođenjem identifikatora svakog uređaja (entiteta) u *SNMP* mreži. Svaki uređaj (entitet) poseduje jedinstveni „*EngineID*“ i

koristi sopstveni ključ za autentifikaciju poruka. Autentifikacija se koristi kako bi se *SNMP* poruke razmenjivale samo između željenih entiteta. Pored toga, verzija 3 je uvela šifrovanje *SNMP* poruka čime je omogućena tajnost podataka koji se prenose. Zbog pouzdanog prenosa poruka, autentifikacije, provere integriteta i šifrovanja *SNMP* poruka, sistem administratorima se preporučuje korišćenje *SNMP* protokola verzije 3.

6 Najčešće ranjivosti Linux servera

Linux serveri mogu imati različite primene, tipično se koriste kao kao veb, imejl, DNS serveri ili serveri sa bazom podataka. Aplikacije koje održavaju ove servise mogu biti sigurnosno porozne što napadači koriste i ugrožavaju rad servera. Tokom godina korišćenja ovih servisa s vremena na vreme su se pojavljivali novi sigurnosni propusti koji su rešavani usput pojavljivanjem novih zakrpa i novih verzija softvera. Ipak, sama arhitektura protokola koji podržavaju ove servise može biti takva da neadekvatno podešavanje servisa na Linux serveru može ostaviti sistem ranjivim i podložnim napadima sa Interneta. U ovom poglavlju se govori o nekim najčešćim napadima na Linux servere i rešenjima za njihovo zaustavljanje

6.1 Kompromitovanje korisničkih naloga

Verovatno najčešći pokušaj napada na Linux servere jeste pokušaj probijanja korisničkih naloga za udaljeni pristup serveru. Napadači tipično skeniraju otvorene portove na serveru i ukoliko detektuju otvorene portove 22 (*SSH*) i 23 (*telnet*) pokušavaju da se povežu na server korišćenjem standardnih kredencijala korisnika. Uobičajeno se pokušava sa korisničkim imenom „*root*“ ili vlastitim imenima (John, Michael, George) a kao lozinke se isprobavaju razne tipične vrednosti (pass123, john123456 i sl.). Termin „*dictionary attack*“ je nastao kao simbol napada gde se kao šifre isprobavaju razne reči u engleskom i drugim jezicima kao i uobičajene kombinacije imena i brojeva. Iznenađujuće, broj kompromitovanih naloga nije mali i pored višegodišnjeg upozoravanja na pravila za formiranje korisničkog imena i šifre.

Odbrana od ovih napada je data u ranijim poglavljima ovog dokumenta. Osnovno je da na serveru ne treba biti omogućen udaljeni pristup korišćenjem naloga „*root*“. Lozinke korisnika moraju biti dovoljno jake i ne smeju biti kombinacija uobičajenih reči i imena u govornom jeziku. Pojedini administratori se bore protiv ovakvih napada i promenom porta za pristup, pa se tako umesto porta 22 za *SSH* pristup koristi neki drugi port (npr. 2202). Rešenje izmeštanjem SSH porta nije dovoljno dobro budući da će skeniranjem i pokušajima pristupa po drugim portovima napdači naći „novi“ *SSH* port. Jedino sigurno rešenje jeste korišćenje *iptables* alata kako bi se ograničio pristup serverima po *SSH* portu samo iz sigurnih mreža. Sigurna mreža je tipično mreža u kojoj je sistem administrator nadležan. Ukoliko administrator želi da obezbedi pristup od kuće ili neke druge udaljene lokacije onda je najbolje rešenje implementacija *VPN* rešenja (*Virtual Private Networking*). Administrator se može iz bilo koje tačke na svetu povezati na *VPN* servis u svojoj mreži, dobiti lokalnu IP adresu i samim tim dobiti pristup lokalnom serveru. *VPN* servis se može implementirati na Linux serveru korišćenjem besplatnog *OpenVPN* rešenja.

6.2 DNS Amplification napad

Jedan od veoma čestih napada na *DNS* servere je „*DNS amplification*“ napad. Ovaj napad iskorišćava moguću funkcionalnost *DNS* servera da služi kao otvoreni rezolver. Otvoreni rezolver podrazumeva da *DNS* server odgovara na sve upite koje dobije, bez obzira odakle oni dolaze. Praktično bilo ko na Internetu može poslati *DNS* upit lokalnom serveru i dobiti odgovor. Napadači skeniraju mreže na Internetu u potrazi za otvorenim rezolverima, a kada ih pronađu koriste ih kao posrednike u napadu. U napadu se izdvajaju uloge napadača, žrtve i otvorenog rezolvera. Napadač kreira lažni *DNS* upit u kojem se predstavlja sa IP adresom žrtve i šalje *DNS* upit otvorenom rezolveru. Otvoreni rezolver prima *DNS* upit i vraća odgovor IP adresi žrtve, ni ne sluteći da žrtva nije ni tražila razrešavanje *DNS* imena. Žrtva počinje da prima neželjeni saobraćaj što u nekim situacijama može izazvati zagušenje odnosno klasičan *DoS* (*Denial of Service*) napad. Glavna poluga ovakvog napada je mehanizam gde mali *DNS* upit može izazvati veliki *DNS* odgovor što dovodi do situacije gde napadač sa malim ulaganjem mrežnog protoka može dobiti napad mnogo većeg mrežnog protoka.

Rešenje problema predstavlja pravilna konfiguracija *named* servisa koji je odgovoran za rad *DNS* servera na Linux operativnom sistemu. U */etc/named.conf* fajlu, u sekciji „*options*“ je neophodno konfigurisati rekurziju samo za opseg lokalnih adresa koje koriste posmatrani *DNS* server.

```
allow-recursion {
            A.B.C.D/E
      };
```

U navedenom delu konfiguracije A.B.C.D/E predstavlja lokalnu mrežu koja koristi posmatrani *DNS* server za razrešavanje *DNS* upita (rekurziju). Prilikom instalacije *DNS* servera, administrator treba da razmisli koji korisnici će koristiti server kao svoj *DNS* i potom omogućiti rekurziju samo tim korisnicima. Za sve ostale, *DNS* server ne treba da vrši razrešavanje *DNS* imena.

6.3 *NTP reflection* napad

NTP (*Network Time Protocol*) je protokol koji omogućava vremensku sinhronizaciju uređaja na mreži. Tipično se svi uređaji sinhronizuju prema jednom centralnom čvoru u mreži. *NTPD* je *NTP* demonski proces na Linux operativnom sistemu koji je zaslužan za implementaciju *NTP* protokola i može se naći na svim Linux distribucijama. Tačno vreme na serveru je veoma bitno budući da se sve log informacije upisuju sa vremenskom referencom, pa administratori mogu imati jasnu sliku o vremenu kada se određeni događaj desio na serveru. Sistem administratorima se obavezno preporučuje korišćenje *NTP* servisa na Linux serverima kako bi mogli da prate vremensku liniju događaja na serveru.

„NTP reflection“ napad u mnogome liči na „*DNS amplification*“ napad budući da mali upit napadača generiše veliki odgovor servera. Kao i kod „*DNS amplification*“ napada, napadač koristi ranjivi Linux server kao posrednika u napadu korišćenjem ***monolist*** komande u *NTP* servisu. Napadač lažira IP paket glumeći žrtvu i šalje ***monolist*** komandu ranjivom Linux serveru, a server odgovara na IP adresu žrtve sa listom od poslednjih 600 mašina koje su imale *NTP* konekciju sa ranjivim Linux serverom. Epilog napada je da žrtva dobija neželjeni saobraćaj koji može ugroziti njeno funkcionisanje.

Rešenje problema leži u pribavljanju novije verzije *NTPD* paketa, 4.2.7 ili novije. Administrator treba da proveri verziju *NTPD* paketa na Linux serveru i ukoliko koristi ranjivu verziju odmah je potrebno izvršiti ažuriranje paketa na sigurnu verziju. Ukoliko iz bilo kog razloga ovo nije moguće, alternativno rešenje predstavlja

isključivanje ***monolist*** komande u *NTP* servisu. U konfiguracionom *NTPD* fajlu /etc/ntp.conf potrebno je dodati sledeće opcije u „*restrict default*“ redove:

```
restrict default kod nomodify notrap nopeer noquery

restrict-6 default kod nomodify notrap nopeer noquery
```

6.4 Heartbleed ranjivost

Jedna od novije otkrivenih ranjivosti je „*Heartbleed*“ koji se javlja kod *openssl* alata. *Openssl* alat je veoma popularan „*open-source*“ softver koji se može naći na 2/3 veb servera na Internetu, a koristi se za implementaciju *SSL/TLS* protokola. *Openssl* se tipično koristi na serverima za uspostavu *HTTPS* konekcije, rad *VPN* servera ili za sigurnu razmenu podataka između *RADIUS* servera. *Heartbleed* ranjivost omogućuje napadaču da kroz ostvarenu *SSL/TLS* vezu sa ranjivim serverom pristupi radnoj memoriji veličine 64KB i iz nje prepozna i pročita kredencijale poslednjeg prijavljenog korisnika ili privatni ključ servera. Dobijanjem privatnog ključa servera, napadač može da dešifruje saobraćaj koji drugi korisnici razmenjuju sa serverom i na taj način dođe do poverljivih informacija, npr. korisničkih kredencijala ili finansijskih transakcija. Ova ranjivost je istorijski verovatno i najozbiljnija otkako se Internet masovno koristi, a posledice korišćenja ranjivog *openssl* alata mogu biti katastrofalne.

Rešenje problema predstavlja korišćenje verzija *openssl* alata koje nisu ranjive na ovaj problem. Sistem administrator treba obavezno da osigura da se na serveru ne koriste verzije *openssl* alata od 1.0.1 do 1.0.1f. Ukoliko se koristi neka od ranjivih verzija alata onda je potrebno pod hitno ažurirati softver na najnoviju sigurnu verziju, potom promeniti privatni ključ i digitalni sertifikat koji se koristi na serveru, a onda promeniti i sve korisničke lozinke koje su se razmenjivale *SSL/TLS* konekcijom sa serverom u prethodnom periodu.

6.5 Zanemarivanje IPv6 protokola

Internet protokol verzije 6 treba u budućnost da zameni IPv4 koji se danas dominantno koristi na Internetu. IPv6 pruža mnoge prednosti ali nažalost do danas nije preuzeo značajniji udeo u Internet saobraćaju. Čini se da je IPv6 pomalo zanemaren u sigurnosnim testiranjima raznih aplikacija koje se masovno koriste na Linux operativnom sistemu. Dodatan problem predstavlja automatsko pokretanje IPv6 podrške na skoro svim Linux distribucijama, dok administratori često zaboravljaju na ovo pa Linux server može ostati potpuno otvoren za napade preko IPv6 protokola. Najčešća greška u konfiguraciji Linux servera jeste zanemarivanje IPv6 protokola i podešavanje sigurnosnih mehanizama isključivo za IPv4 saobraćaj. Tipično, administratori postavljaju pravila za filtriranje IPv4 saobraćaja, dok IPv6 saobraćaj može ostati nefiltriran što omogućava napadačima otvoren ulaz i kompromitovanje servera.

Ukoliko servisi na Linux serveru nisu planirani za IPv6 saobraćaj, administratorima se preporučuje implementacija striktne mrežne politike korišćenjem *ip6tables* alata ili isključivanje podrške za IPv6 na nivou celog operativnog sistema. Isključivanje IPv6 podrške se vrši dodavanjem sledećih linija u */etc/sysctl.conf* fajl:

```
net.ipv6.conf.all.disable_ipv6 = 1

net.ipv6.conf.default.disable_ipv6 = 1
```

```
net.ipv6.conf.lo.disable_ipv6 = 1
```

Gore navedenim linijama se ne isključuje IPv6 modul u kernelu već se samo sprečava dodeljivanje i korišćenje IPv6 adresa na mrežnim interfejsima. Isključivanje IPv6 modula u kernelu se ne preporučuje budući da može izazvati probleme u radu pojedinih aplikacija koje pokušavaju da koriste IPv6.

Ukoliko se administratori odluče za isključivanje IPv6 podrške na mrežnim interfejsima, potrebno je u */etc/hosts* fajlu dodatno zakomentarisati sve IPv6 adrese. Nakon načinjenih promena potrebno je ponovo pokrenuti Linux operativni sistem.

7 Dijagnostikovanje rada Linux servera

Tokom održavanja servera, sistem administrator često ima potrebu da proveri trenutni rad operativnog sistema, bilo da se radi o proveri mrežnog dela, sistemskog dela ili proveri log fajlova. Ovo nije klasičan monitoring rada servera već se radi o dijagnostikovanju trenutnog stanja. Linux poseduje mnoge ugrađene alate za ovu svrhu ali se mogu naći i neki dodatni alati koji daju preglednije informacije. Alati za dijagnozu rada sistema se obično koriste prilikom ispitivanja rada određenih aplikacija, servisa ili prilikom istraživanja sigurnosnih incidenata. Sistem administratorima se preporučuje da probaju razne alate i na kraju izaberu one koji im najviše odgovaraju.

7.1 Provera potrošnje sistemskih resursa

Provera potrošnje sistemskih resursa se vrši kada se sumnja u neopravdano prekomerno opterećenje Linux servera. Izvor povećane potrošnje sistemskih resursa može biti određeni proces, a uzrok sigurnosni incident koji se dogodio na serveru. Administrator treba da otkrije koji proces je odgovoran za povećano opterećenje servera a potom daljim ispitivanjima procesa treba da utvrdi uzrok za ovakvo ponašanje. Pomoć u dijagnozi problema mogu dati razni Linux alati koji ispituju iskorišćenost sistemskih resursa.

7.1.1 Iskorišćenost sistemskih resursa

Povremeno se događa da Linux server otežano radi i usporeno obavlja predviđene funkcije. Prvi zadatak administratora je da proveri iskorišćenost sistemskih resursa kako bi utvrdio da li postoji neko odstupanje od uobičajenih vrednosti. Tipično se proveravaju opterećenost RAM memorije i procesora, a potom i performanse diskova za smeštanje podataka. Postoji veliki broj alata pomoću kojih sistem administrator može brzo ustanoviti koji deo servera ima povećanu potrošnju resursa.

Komanda ***free*** prikazuje trenutnu potrošnju RAM memorije, a dodatno se može videti i iskorišćenost swap memorije.

Komanda ***vmstat*** ispisuje informacije o potrošnji RAM memorije, blokovima upisa i ispisa na disk kao i o aktivnostima procesora. Komandi se može dodati i broj sekundi na koji će se ispisivati najnoviji rezultati potrošnje sistemskih resursa.

Komanda ***uptime*** daje informacije o proteklom vremenu od poslednjeg podizanja Linux operativnog sistema, potom broj korisnika koji su trenutno prijavljani na Linux operativni sistem, a na kraju daje i vrednost opterećenosti servera u proteklom periodu. Vrednost opterećenosti varira od servera do servera, ali se

uopšteno može uzeti da je prihvatljiva vrednost do 3 za servere sa jednim procesorom, a do 10 za servere sa više procesora.

Komanda ***iostat*** daje pregled iskorišćenosti procesora i statistiku aktivnosti diskova i particija na Linux serveru. Alat koji daje slične podatke je *dstat*.

Komanda ***sar*** ispisuje potrošnju procesorske moći u periodičnim intervalima. Komanda ***mpstat*** je veoma slična s tim što ispisuje podatke po svakoj procesorskoj jedinici.

7.1.2 Aktivnost pojedinačnih procesa

Nakon otkrivanja povećane potrošnje sistemskih resursa, sledeći logičan korak je pronalaženje procesa koji su odgovorni za opterećenost Linux servera. Postoji veliki broj dobrih alata i komandi, a ovde će biti predstavljeni neki najčešće korišćeni.

Komanda ***top*** izlistava procesorsko zauzeće pojedinačnih procesa u Linux operativnom sistemu. Procesi su poređani od najzahtevnijih ka manje zahtevnim. Postoji mnogo alata koji su slični kao na primer *htop* ili *atop* koji imaju atraktivniji i uredniji prikaz informacija, a pri tom mogu davati i više informacija o pojedinačnim procesima. Alat *atop* je posebno zanimljiv budući da može čuvati informacije o potrošnji resursa u log fajlovima pa sistem administrator može uporediti trenutnu potrošnju u odnosu na potrošnju od pre par dana. *Iotop* alat ima istu formu kao *top* ali se bavi analizom upisa i ispisa podataka na disk po aktivnim procesima.

Alat *nmon* prikazuje ukupnu potrošnju resursa kao i potrošnju po procesu. U pitanju je veoma atraktivan alat koji omogućava administratoru da se komandama na tastaturi kreće kroz različite statističke podatke i dobije informaciju o iskorišćenosti procesora, memorije, prostora za skladištenje, a prikazuje i procese koji najviše opterećuju sistem.

Alat *glances* je sličan *nmon* alatu i predstavlja kompletno rešenje za pregled opšte potrošnje sistemskih resursa kao i potrošnje po procesu. U pitanju je atraktivan i pregledan alat čija funkcionalnost obuhvata nekoliko alata kao što su *top*, *iostat*, *vmstat* i druge slični alati.

Komanda ***ps*** ispisuje sve aktivne procese u Linux operativnom sistemu. Komanda pruža sve najbitnije podatke o aktivnim procesima, veoma je konfigurabilna a nudi i mogućnost prikaza najzahtevnijih procesa po potrošnji memorije ili procesorskog vremena.

Komanda ***pmap*** ispisuje potrošnju memorije po izabranom procesu i često se koristi u kombinaciji sa komandom ***ps*** kako bi se dobili tačni podaci o potrošnji memorije po određenom procesu.

Komanda ***lsof*** ispisuje listu otvorenih fajlova i procesa koji su ih otvorili, a može da prikaže i listu otvorenih mrežnih soketa po procesima.

7.2 Provera mrežne aktivnosti

Većina sigurnosnih problema dolazi preko mreže, pa je neophodno imati uvid u trenutne mrežne aktivnosti Linux servera, interakciju servera sa drugim mašinama ali i identifikovati otvorene portove i servise.

Administrator treba da obezbedi alate pomoću kojih će dobiti informacije o mrežnom saobraćaju koji server razmenjuje sa udaljenim mašinama.

Komanda ***netstat*** ispisuje aktivne konekcije, portove na kojima server sluša, ruting tabelu i statistiku mrežnog saobraćaja. U pitanju je veoma moćna komanda koja će sistem administratoru pružiti sve bitnije informacije o mrežnom saobraćaju koji server razmenjuje sa drugim uređajima na mreži. Pored ***netstat*** komande preporučuje se ***ss*** komanda (*Socket Statistics*) koja će ispisati statističke podatke konekcija u veoma sličnom formatu.

Alat *iftop* prikazuje količinu razmenjenog saobraćaja po pojedinačnim aktivnim konekcijama koje Linux server uspostavlja sa udaljenim mašinama na mreži. Ono što je *top* za procesorsku potrošnju po aktivnim procesima, to je *iftop* za količinu razmenjenog mrežnog saobraćaja po aktivnim konekcijama. Veoma sličan alat predstavlja *nload* koji daje i osnovni grafički prikaz dolaznog i odlaznog saobraćaja kako bi se lakše uočile neke anomalije u mrežnom protoku informacija.

Iptraf alat je sveobuhvatni alat koji prati mrežnu aktivnost servera. Pomoću njega administrator može dobiti statističke podatke o razmenjenom saobraćaju, pojedinačnim konekcijama, mrežnim aktivnostima na pojedinačnim interfejsima, podatke po portovima ili veličini paketa. Korisni podaci mogu biti i podaci o razmenjenom saobraćaju sa pojedinačnim MAC adresama.

NetHogs alat izlistava informacije o mrežnom saobraćaju svakog pojedinačnog procesa na Linux operativnom sistemu i može biti veoma koristan prilikom eventualnih istraga o ponašanju pojedinih procesa.

Arpwatch alat prati *ARP* protokol na mreži gde se nalazi Linux server i beleži promene u parovima IP i MAC adresa. Može biti veoma koristan alat u detektovanju „*ARP spoofing*“ napada na mreži.

Alat *apachetop* je namenjen praćenju *http* zahteva koji pristižu na *Apache* veb server. Alat prikazuje statistiku, zahtevane URL adrese i protok što daje dobru opšu sliku o aktivnostima veb servera.

GoAccess je sličan *apachetop* alatu ali pruža više informacija o saobraćaju koji se razmenjuje sa veb serverom. Sistem administrator može videti koji udaljeni operativni sistemi ili Internet pregledači pristupaju veb serveru, potom najaktivnije IP adrese, najtraženije URL adrese ili fajlove na veb serveru.

Komanda ***tcpdump*** je veoma korisna prilikom praćenja određenog mrežnog saobraćaja na serveru. Ova komanda služi za ispitivanje pojedinačnih paketa koji dolaze na server i služi za najpreciznije ispitivanje mrežnog saobraćaja Linux servera. Alternativa može biti *Wireshark* alat za Linux koji služi istoj svrsi.

Nmap je nezamenljivi alat u testiranju mrežnih servisa i otvorenih portova. Dovoljno je imati *nmap* instaliran na jednom Linux serveru i skenirati sve ostale uređaje i servere na mreži. Administratorima se preporučuje korišćenje ovog alata kako bi testirali da li postavljene zabrane pristupa serverima po određenim portovima ispravno funkcionišu.

7.3 Provera tipičnih log informacija

Log fajlovi pružaju pregled rada pojedinačnih procesa i aplikacija na Linux serveru. Mogu se podesiti da budu sažeti ili opširniji u zavisnosti od nivoa detalja koje administrator želi da beleži. U trenucima kada server radi bez problema, sistem administratorima se preporučuje da pregledaju log fajlove kako bi naučili kako procesi i aplikacije funkcionišu i koje log poruke karakterišu stabilan rad servera. Kada se dogodi neki problem, kasno je

za sticanje opšteg razumevanja rada pojedinačnih procesa. Log fajlovi na Linux distribucijama se tipično nalaze u */var/log/* direktorijumu, a neki od osnovnih fajlova su:

- */var/log/messages* – Opšte informacije o Linux radu sistema
- */var/log/boot.log* – Informacije o procesima prilikom podizanja operativnog sistema
- */var/log/auth.log* ili */var/log/secure* – informacije o autentifikaciji korisnika
- */var/log/yum.log* ili */var/log/dpkg.log* – informacije o instalaciji paketa
- /*var/log/mail.log* – informacije o radu mail servera.

Sistem administrator podešava beleženje log informacija u *syslog* alatu kako je definisano u poglavlju 5. Određene aplikacije kreiraju samostalno log fajlove i obično ih smeštaju u direktorijume na lokaciji */var/log/*. Administratorima se preporučuje da povremeno pregledaju log informacije kako bi proverili da li ima nekih novih momenata u radu servera.

8 Procedura kreiranja sigurnosnih kopija

Kreiranje sigurnosnih kopija podataka (*backup*) je veoma važan aspekt održavanja sigurnosti na serveru. Ukoliko nešto krene po zlu i dođe do gubitka podataka dobro je imati mogućnost vraćanja izgubljenih fajlova. Razlozi gubitaka fajlova mogu biti raznoliki, od probijanja servisa na serveru do nenamernih grešaka korisnika. Cilj implementacije rešenja kreiranja sigurnosnih kopija je izbegavanje ponovnog podizanja celokupnog sistema, oporavak servera u kriznim situacijama i smanjenje vremena nefunkcionalnosti servisa koji se održavaju na serveru.

8.1 Strategija kreiranja sigurnosnih kopija

Sistem administrator treba da napravi strategiju kreiranja sigurnosnih kopija koja će odgovoriti na sledeća pitanja:

- Koji fajlovi se kopiraju i čuvaju?
- Gde se čuvaju sigurnosne kopije?
- Koju tehniku kreiranja sigurnosnih kopija treba primeniti?
- Koliko je vremensko trajanje ciklusa kreiranja sigurnosnih kopija?
- Koliko dugo se čuvaju sigurnosne kopije?
- Koliko prostora treba ostaviti za skladištenje sigurnosnih kopija?

U zavisnosti od namene servera razlikuje se i skup fajlova koje treba sačuvati. Lokacije fajlova variraju od servera do servera ali je uopšteno govoreći esencijalno sačuvati fajlove pojedinačnih korisnika, fajlove najbitnijih aplikacija, sistemske fajlove i sav sadržaj koji se koristi u svakodnevnom radu servera. Uobičajeno se podaci korisnika čuvaju u */home/* direktorijumu, fajlovi aplikacija u */usr/* direktorijumu, a konfiguracioni fajlovi demonskih procesa koji održavaju stabilan rad Linux operativnog sistema se nalaze na lokaciji */etc/*. Svaki server može imati drugačije prioritete kada je u pitanju izbor podataka za kopiranje. Na primer, za veb server je bitno sačuvati konfiguracione fajlove u */etc/apache2/* ili */etc/httpd/* direktorijumu, kao i sadržaj koji se nudi krajnjim korisnicima a koji se tipično nalazi na lokaciji */var/www/*. Ukoliko je svrha Linux servera održavanje *SQL* baze podataka onda je potrebno sačuvati podatke u bazi kao i konfiguracione fajlove *SQL* softvera. Ipak, bez obzira na konačnu namenu servera, sa sigurnosnog aspekta je bitno sačuvati fajlove u direktorijumu */etc/* jer se ovde nalaze konfiguracije korisničkih naloga i mrežnih filtara.

Kopirani podaci se mogu čuvati na samom serveru a kasnije prebaciti na neki drugi medijum kao što je optički medijum. Problem ovakvog načina skladištenja je neefikasno upravljanje resursima, zahtevanje fizičkog prostora gde će se čuvati optički diskovi i duža procedura prilikom povraćaja kopiranih podataka. Bolji način predstavlja čuvanje kopiranih podataka na nekom od raspoloživih servera za skladištenje podataka, a prenos podataka se može obaviti putem mreže. Prenos podataka preko mreže se može obaviti putem *FTP* protokola,

rsync alata ili dodavanjem udaljenog fajl sistema. Prednost predstavlja činjenica da svaki od ovih načina nudi opciju sigurnog prenosa podataka: *SFTP*, *rsync* preko *SSH* ili *SSHFS* za dodavanje udaljenog fajl sistema. Sistem administrator uvek treba da ima na umu da je jedan od najvažnijih uslova prenosa kopiranih podataka tajnost i sprečavanje neautorizovanih pristupa serveru gde se podaci skladište.

Pri odabiru tehnike kreiranja sigurnosnih kopija treba uzeti u obzir količinu podataka za kopiranje i raspoloživi prostor za skladištenje. Sistem administrator može izabrati jednu od dve tehnike: standardno sigurnosno kopiranje (*full backup*) ili inkrementalno sigurnosno kopiranje (*incremental backup*). Standardno sigurnosno kopiranje podrazumeva kopiranje kompletnih fajlova svaki put iznova. Pogodan je za jednostavnija rešenja koja zahtevaju čuvanje manje količine podataka. Tehnika inkrementalnog sigurnosnog kopiranja podrazumeva kopiranje samo onih fajlova koji su promenjeni u odnosu na prethodno sigurnosno kopiranje. Ovakvo rešenje zahteva početno standardno sigurnosno kopiranje, a nakon toga se vrše inkrementalna kopiranja promenjenih fajlova. Kada je potrebno povratiti informacije, kao osnova se uzima početna standardna sigurnosna kopija, prolazi se kroz sva inkrementalna kopiranja i pronalazi poslednja sačuvana verzija fajla. Pogodan je za sigurnosna kopiranja velike količine podataka budući da se ne vrši skladištenje celokupnih podataka već samo onih koji su u međuvremenu promenjeni čime se minimalno troši prostor za skladištenje. Dodatno, inkrementalno sigurnosno kopiranje značajno smanjuje vreme čitave procedure, a ako se fajlovi čuvaju na udaljenom serveru onda se dodatno smanjuje opterećenje mrežnog linka.

Učestalost sigurnosnog kopiranja je period između dve procedure kopiranja i podešava se u zavisnosti od učestalosti promena podataka na serveru. Ukoliko se sadržaj dinamički menja onda je dobra praksa dnevno kopiranje ili čak postavka kraćeg vremenskog perioda. Što je vremenski period kraći manji je rizik gubitka podataka, a povratak na pređašnje stabilno stanje je brže.

Ciklus sigurnosnog kopiranja predstavlja period između dva standardna sigurnosna kopiranja. Ciklus počinje sa prvim standardnim sigurnosnim kopiranjem, uključuje sva naredna inkrementalna kopiranja, a završava se prvim narednim standardnim sigurnosnim kopiranjem.

Vreme čuvanja sigurnosnih kopija nije jednostavno odrediti budući da u mnogome zavisi od realnih potreba i dostupnih resursa.Treba čuvati kopije bar za minimalni potrebni period reagovanja u slučaju nepredviđenih okolnosti. Ukoliko sistem administrator ima na raspolaganju dovoljno resursa onda se kopije mogu čuvati i duži vremenski period (par ciklusa sigurnosnog kopiranja).

Neophodni prostor za skladištenje je u direktnoj vezi sa veličinom podataka koji se čuvaju i sa definisanim vremenom čuvanja. Sigurnosne kopije mogu zauzeti mnogo prostora na serveru pa se administratorima preporučuje da naprave računicu pre nego što implementiraju konačno rešenje.

8.2 Alati za sigurnosno kopiranje

U zavisnosti od odabrane strategije, sistem administrator treba da izabere odgovarajući alat za implementaciju sigurnosnog kopiranja. Ovde će biti predstavljeni neki alati koji se najčešće koriste.

Komanda ***tar*** služi za arhiviranje podataka na Linux operativnom sistemu. Može se koristiti za sigurnosno kopiranje arhiviranjem i kompresijom željenih fajlova na serveru. Nakon arhiviranja potrebno je *.tar* fajl prebaciti na optički medijum ili udaljeni server, u zavisnosti od izabranog prostora za skladištenje. Ukoliko se transfer *.tar* fajla vrši preko mreže, administratorima se preporučuje upotreba *SFTP*, *SCP* ili *SSH* konekcije jer na taj način obezbeđuju tajnost i sigurnost prenetih podataka. Komandom ***tar*** se obično implementira tehnika standardnog

sigurnosnog kopiranja i preporučuje se u slučajevima kada je neophodno čuvati male količine podataka. Veće količine podataka bi trošile više sistemskih i mrežnih resursa.

Alat *Bacula* je veoma koristan za procedure inkrementalnog sigurnosnog kopiranja jer uvodi tzv. „diferencijalno“ sigurnosno kopiranje (*differential backup*). Diferencijalno sigurnosno kopiranje podrazumeva kopiranje svih fajlova koji su promenjeni u odnosu na poslednju standardnu sigurnosnu kopiju. Praktično, diferencijalno sigurnosno kopiranje objedinjuje sva inkrementalna sigurnosna kopiranja do tog trenutka. Primer ciklusa sigurnosnog kopiranja pomoću alata *Bacula* bi bio da se svakog prvog dana u mesecu radi standardno sigurnosno kopiranje, svakog dana inkrementalno kopiranje a svake nedelje diferencijalno sigurnosno kopiranje. Ciklus sigurnosnog kopiranja počinje standardnim sigurnosnim kopiranjem i uključuje sva naredna inkrementalna i diferencijalna kopiranja do narednog standardnog sigurnosnog kopiranja. Uvođenjem diferencijalnog sigurnosnog kopiranja, skraćuje se vreme povraćaja podataka, jer se za povraćaj uzima standardna sigurnosna kopija, potom primenjuje poslednja diferencijalna sigurnosna kopija i nakon toga vrše poslednja inkrementalna ažuriranja. Mana diferencijalnog sigurnosnog kopiranja je povećanje prostora neophodnog za skladištenje kopija jer diferencijalna kopiranja čuvaju veći broj fajlova u odnosu na inkrementalna kopiranja. *Bacula* može biti konfigurisana da koristi *SSH* konekciju kako bi se prenosili kopirani podaci. Administratorima se preporučuje upotreba *SSH* ključeva kako bi osigurali konekciju između Linux servera na kome se vrši procedura sigurnosnog kopiranja i servera na kome se skladište podaci.

Duplicity je takođe veoma dobar alat za inkrementalno sigurnosno kopiranje. U osnovi ovog alata je kreiranje šifrovanih *.tar* fajlova koji se mogu bezbedno preneti na udaljeni server za skladištenje podataka. *Duplicity* koristi *GPG* (*Gnu Privacy Guard*) sigurnosne ključeve kako bi se digitalno potpisali i šifrovali *.tar* fajlovi i na taj način sprečava kompromitovanje kopiranih podataka. Alat podržava *SSH* i *SCP* konekcije pa se administratorima preporučuje da ih koriste ukoliko se podaci skladište na udaljenim serverima.

Rsnapshot je interesantan alat za inkrementalno sigurnosno kopiranje koji se bazira na *rsync* alatu za sinhronizaciju fajlova. Alat omogućava ograničenje broja ciklusa sigurnosnog kopiranja što sprečava nekontrolisano povećanje potrebnog prostora za skladištenje podataka. Alat omogućava siguran prenos kopija pomoću *SSH* protokola.

Reference

[1] Kickstart instalacija: http://fedoraproject.org/wiki/Anaconda/Kickstart
[2] Preseed instalacija: http://www.debian.org/releases/stable/i386/apb.html.en
[3] SSHFS: https://help.ubuntu.com/community/SSHFS
[4] iptables: http://www.netfilter.org/projects/iptables/
[5] Antidote: http://antidote.sourceforge.net/
[6] Arptables: http://ebtables.sourceforge.net/misc/arptables-man.html
[7] Arpwatch: http://www.linuxcommand.org/man_pages/arpwatch8.html
[8] Fail2Ban: http://www.fail2ban.org/wiki/index.php/Main_Page
[9] DenyHosts: http://denyhosts.sourceforge.net/
[10] TCP Wrapper: https://www.centos.org/docs/4/html/rhel-rg-en-4/s1-tcpwrappers-access.html
[11] SE Linux: http://selinuxproject.org/page/Main_Page
[12] AppArmor: http://wiki.apparmor.net/index.php/Main_Page
[13] Sudo: http://www.sudo.ws/
[14] Syslog: http://man7.org/linux/man-pages/man3/syslog.3.html
[15] Syslog-ng: http://www.syslog-ng.org/
[16] OpenVPN: http://openvpn.net/
[17] Glances: https://github.com/nicolargo/glances
[18] Iptraf: http://iptraf.seul.org/
[19] Nload: http://www.roland-riegel.de/nload/
[20] NetHogs: http://nethogs.sourceforge.net/
[21] Apachetop: http://www.fr3nd.net/projects/apache-top/
[22] GoAccess: http://goaccess.prosoftcorp.com/
[23] Wireshark: http://www.wireshark.org/
[24] Nmap: http://nmap.org/
[25] Bacula: http://blog.bacula.org/
[26] Duplicity: http://duplicity.nongnu.org/
[27] Rsnapshot: http://www.rsnapshot.org/

Rečnik pojmova

ARP	Address Resolution Protocol
CPU	Central Processing Unit
IP	Internet Protocol
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
MAC	Media Access Control
NFS	Network File System
RAM	Random Access Memory
SQL	Structured Query Language
SSH	Secure Shell
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
UDP	User Datagram Protocol