HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Samhain programski paket

CCERT-PUBDOC-2003-04-14

CARNet CERT u suradnji s LS&S

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

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

CARNet CERT, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.lss.hr - laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. UVOD 4

2. INSTALACIJA 4

3. NAČINI RADA 5

3.1. SAMOSTALNI NAČIN RADA 5
3.2. KLIJENT/POSLUŽITELJ NAČIN RADA 5
3.3. NEVIDLJIVI NAČIN RADA 6

4. KONFIGURACIJA 6

4.1. GENERIRANJE UPOZORENJA I BILJEŽENJE LOG ZAPISA 6
4.2. SIGURNOSNA POLITIKA 8
4.3. PROVJERA SUID/SGID DATOTEKA 9
4.4. DETEKCIJA KERNEL ROOTKIT MALICIOZNIH PROGRAMA 9
4.5. DETEKCIJA PRIJAVLJIVANJA I ODJAVLJIVANJA U SUSTAV 10
4.6. BILJEŽENJE LOGOVA U BAZU PODATAKA 10
4.7. OSTALI PARAMETRI 11

5. KLIJENT/POSLUŽITELJ NAČIN RADA 11

6. PROVJERA INTEGRITETA LOG ZAPISA I E-MAIL PORUKA 13

7. ZAKLJUČAK 15

1. Uvod

Samhain programski paket besplatan je sigurnosni alat namijenjen Linux operacijskim sustavima, koji osim provjere integriteta datotečnog sustava (eng. *file integrity check*) omogućuje i detekciju jednostavnijih malicioznih aktivnosti na sustavu.
U okviru detekcije neovlaštenih aktivnosti (eng. *intrusion detection*), samhain program omogućuje praćenje prijavljivanja i odjavljivanja korisnika u sustav (*login/logoff* akcije), detekciju neregularnih SUID/SGID datoteka te detekciju *kernel rootkit* malicioznih programa.
U kombinaciji s provjerom integriteta datotečnog sustava, samhain programski paket može izvrsno poslužiti za podizanje sigurnosnog nivoa na Linux mrežnim poslužiteljima.
Osim osnovnog samostalnog (eng. *standalone*) načina rada, program je moguće podesiti da radi i u tzv. klijent/poslužitelj (eng. *client/server*) načinu rada, što je posebno praktično u mrežnim okolinama sa većim brojem poslužitelja koje je potrebno istovremeno nadzirati.
Razvijen je i nevidljivi (eng. *stealth*) način rada, kojem je osnovna namjena da u potpunosti ukloni tragove postojanja programa na sustavu.
U nastavku dokumenta opisani su osnovni postupci instalacije i konfiguracije programa, zajedno s načinima primjene.

2. Instalacija

Samhain programski paket dostupan je u tar.gz paketu, kojega je moguće dobaviti sa sljedeće URL adrese - http://samhain.sourceforge.net.
Prije same instalacije programa, potrebno je dobaviti odgovarajući PGP ključ sa odgovarajućeg poslužitelja, kako bi se na taj način omogućila provjera integriteta dobavljenog izvornog koda. Spomenuti PGP ključ moguće je dobaviti sa search.keyserver.net poslužitelja sljedećom naredbom:

```
# gpg --keyserver search.keyserver.net --recv-key 0F571F6C
```

Korišteni gpg alat dio je GnuPG (*GNU Privacy Guard*) programskog paketa, *open source* implementacije poznatog PGP programa za enkripciju i digitalno potpisivanje podataka pri njihovoj razmjeni.
Nakon što je dobavljen traženi PGP ključ potrebno je provjeriti integritet izvornog koda programa:

```
# gpg --verify samhain-1.7.1.tar.gz.asc samhain-1.7.1.tar.gz
```

Nakon što je potvrđen integritet izvornog koda moguće je standardnim postupkom otpakirati tar.gz arhivu samhain programa:

```
# tar –xzvf samhain-current-1.7.x
```

Nakon toga potrebno je ući u novonastali samhain-1.7.x direktorij te u njemu obaviti prevođenje i instalaciju programa.
Postupak je slijedeći:

```
#./configure [argumenti]
#make
#make install
```

Prosljeđivanjem različitih argumenata configure skripti moguće je precizno definirati tip instalacije programa. U nastavku (Tablica 1) je dana lista nekih važnijih parametara configure skripte, zajedno s njihovim kratkim objašnjenjem.

Ime argumenta	Značenje
`--prefix=PREFIX`	mjesto instalacije programa.
`--enable-network=[client|server]`	odabir tipa instalacije (klijent ili poslužitelj) – vidjeti napomenu.
`--enable-suidcheck`	uključi provjere SUID/SGID datoteka.
`--enable-install-name=NAME`	definiranje proizvoljnog imena NAME pod kojim će program biti instaliran. Opcija namijenjena prikrivanju programa.
`--enable-login-watch`	omogući kontrolu prijavljivanja i odjavljivanja u sustav (*login/logoff*).
`--with-kcheck=SYSTEM_MAP`	omogući provjeru integriteta jezgre operacijskog sustava (detekcija *kernel rootkit* programa).
`--with-database=[mysql|postgresql]`	omogući bilježenje log zapisa u mysql ili postgresql bazu podataka.

Tablica 1 - Argumenti configure skripte samhain programa

Napomena:
Instalaciju klijenta (`samhain`) i poslužitelja (`yule`) nije moguće obaviti istovremeno. Za svaki od navedenih programa potrebno je posebno pokrenuti `configure`, `make`, `make install` naredbe sa odgovarajućim parametrima.
Nakon pokretanja `configure` skripte, potrebno je obaviti prevođenje i instalaciju programa (`make && make install`) na način kako je to upravo spomenuto.
Ukoliko se želi omogućiti pokretanje `samhain` pri svakom podizanju sustava, potrebno je nakon `make install`, zadati i sljedeću naredbu:

```
# make install-boot
```

Program `make` prima i argument `uninstall` kojim je po potrebi moguće deinstalirati program.

3. Načini rada

`Samhain` programski paket podržava tri osnovna načina rada. To su:

- samostalni (eng. *standalone*)
- klijent/poslužitelj (eng. *client/server*)
- nevidljivi (eng. *stealth*)

U nastavku je ukratko opisan svaki od navedenih načina, zajedno sa svojim osnovnim karakteristikama.

3.1. Samostalni način rada

U ovom slučaju `samhain` programski paket instalira se samostalno na poslužitelju koji se želi nadzirati.
Uređivanjem konfiguracijske datoteke programa (`/etc/samhainrc`) definiraju se objekti (datoteke i direktoriji) koji se žele pratiti te ostali parametri uključeni u provjeru stanja sustava.
Na temelju definirane konfiguracije potrebno je kreirati jedinstveni "otisak" sustava na pomoću kojeg će se periodički provoditi provjere integriteta sustava. Učestalost provjera i način izvještavanja moguće je definirati putem konfiguracijske datoteke programa.
Ovo je najjednostavniji način korištenja programa, i zadovoljiti će većinu potreba gdje je potrebno nadzirati samo jedno ili manji broj računala.

3.2. Klijent/poslužitelj način rada

Klijent/poslužitelj način rada namijenjen je mrežnim okolinama u kojima se pomoću `samhain` programa istovremeno želi nadzirati više poslužitelja. Na svakom od poslužitelja potrebno je instalirati klijentsku aplikaciju (`samhain`), koja će na temelju zadane konfiguracije obavljati periodičke provjere integriteta sustava.

Na jednom od poslužitelja instalira se poslužiteljska aplikacija (`yule`), koja će predstavljati centralni dio cijelog sustava.
Zadaća `yule` poslužitelja je da prikuplja rezultate provjera sa svih klijent aplikacija te da pohranjuje njihove konfiguracijske datoteke i "otiske". Na taj način omogućena je centralizirana administracija svih klijent sustava.
Komunikacija klijent-poslužitelj dozvoljena je samo za računala koja su registrirana unutar konfiguracijske datoteke `yule` poslužitelja. Registracija se provodi dodavanjem posebno formiranog zapisa unutar konfiguracijske datoteke, i to za svaki od klijenata.
Svi pokušaji komunikacije sa onih računala koja nisu registrirana u konfiguracijskoj datoteci poslužitelja biti će prekinuti, a legitimne veze biti će potpisane prethodno dogovorenim ključem.
S obzirom na brojne kvalitete i prednosti koje nudi klijent/poslužitelj način rada, isti će biti detaljnije opisan poglavlju 5.

3.3. Nevidljivi način rada

Nevidljivi način rada namijenjen je korisnicima koji žele održavati maksimalnu sigurnost na svojim poslužiteljima.
Ukoliko neovlašteni korisnik primijeti da je na sustavu pokrenut neki od alata za provjeru integriteta, postoji realna mogućnost da zaobiđe sigurnosne mjere nametnute programom. Samim time program u određenoj mjeri gubi svoj smisao, budući da je napadač svjestan mogućnosti detekcije promjena na sustavu.
Korištenje programa u nevidljivom načinu rada uklanja ovaj problem, budući da će program biti vješto prikriven na sustavu. Posebnim metodama biti će prikrivene sve datoteke `samhain` programa, čime se bitno utječe na njegovu učinkovitost.
Da bi se program koristio u nevidljivom načinu rada potrebno ga je prevesti sa `--with-stealth=`*`xor_val`* opcijom. `Xor_val` vrijednost je proizvoljan broj između 128 i 255 koji će biti korišten za maskiranje sadržaja datoteka programa.
Provođenjem XOR logičke operacije između sadržaja pojedinih datoteka i navedene `xor_val` vrijednosti, identifikacija `samhain` programa biti će znatno otežana.
Nad čistim tekstualnim datotekama biti će cijeli sadržaj kriptiran spomenutom XOR operacijom, dok će se u binarnim datotekama kriptirati samo pojedini znakovni nizovi koji omogućuju identifikaciju programa.
Konfiguracijska datoteka programa u ovom je slučaju posebnim steganografskim postupcima prikrivena unutar PostScript slike, a prema potrebi moguće je i log datoteku spojiti s nekom drugom `exe` ili `jpeg` datotekom.
Prosljeđivanjem `--enable-install-name=NAME` parametra `configure` skripti moguće je dodatno prikriti postojanje programa. U ovom slučaju program će se na sustavu pojavljivati pod proizvoljno definiranim imenom `NAME` (a ne `samhain`), čime će dodatno biti otežana njegova detekcija.

4. Konfiguracija

Konfiguracijska datoteka `samhain` programskog paketa zove se `samhainrc`, i u inicijalnoj instalaciji nalazi se u `/etc` direktoriju. Uređivanjem ove konfiguracijske datoteke definiraju se svi parametri bitni za rad programa. Definiraju se objekti sustava uključeni u provjeru, atributi koji se žele kontrolirati, način generiranja upozorenja, učestalost provjera te brojni drugi parametri.
Konfiguracijska datoteka organizirana je u sekcije od kojih svaka ima svoj značaj

4.1. Generiranje upozorenja i bilježenje log zapisa

Za svaki događaj (eng. *event*) koji se prema konfiguraciji programa smatra neregularnim, biti će generirano odgovarajuće upozorenje. Ovisno o prioritetu upozorenja, poruka će biti zabilježena jednom od podržanih metoda bilježenja log zapisa.
U sljedećoj tablici (Tablica 2) dana je lista prioriteta pod kojima `samhain` program generira upozorenja.

Prioritet	Značenje
none	Poruke generirane pod ovim prioritetom se ne bilježe.
debug	Poruke niskog prioriteta vezane za otklanjanje grešaka.
info	Informativne poruke.
notice	Poruke koje indiciraju normalna stanja sustava.
warn	Upozorenja.
mark	Vremenske oznake (eng. *timestamp)*.
err	Poruke koje indiciraju grešku.
crit	Poruke koje indiciraju kritična stanja.
alert	Poruke koje indiciraju pokretanje i zaustavljanje servisa, neuobičajena stanja programa te slične ostale greške.
inet	Poruke koje yule poslužitelj prima od klijenata.

Tablica 2 – Prioriteti poruka samhain programa

Kao što je već ranije spomenuto svako upozorenje biti će generirano s određenim prioritetom (definiranim konfiguracijom programa) te će ovisno o njemu biti zabilježeno nekom od podržanih metoda logiranja.

U sljedećoj tablici (Tablica 3) dana je lista podržanih metoda putem kojih je moguće bilježenje log zapisa:

Metoda	Opis
e-mail	Upozorenja se administratoru šalju putem e-mail poruka.
syslog	Bilježenje zapisa putem syslog poslužitelja.
console	Log zapisi ispisuju se na konzolu sustava.
log file	Log zapisi pohranjuju se u posebnu datoteku.
log server	Log zapisi šalju se centralnom yule poslužitelju.
external	Mogućnost bilježenja log zapisa putem nekog drugog programa.
SQL server	Bilježenje log zapisa u MySQL i Postgres bazu.

Tablica 3 - Metode bilježenja log zapisa.

Za svaku od navedenih metoda definiran je korisnički podesiv prag koji je potrebno preći da bi poruka bila zabilježena dotičnom metodom. To znači da će sve poruke određenog prioriteta biti zabilježene onim metodama čiji je prag manji, ili jednak onome pod kojim je poruka generirana.

U nastavku je prikazan dio `samhainrc` konfiguracijske datoteke (sekcija `[Log]`) kojim se definiraju pragovi spomenutih metoda bilježenja log zapisa.

```
[Log]
  #
  # Threshold for E-mails (none = switched off)
  #
  MailSeverity=none
  #
  # Threshold for log file
  #
  LogSeverity=err}
  LogClass=RUN FIL STAMP
  #
  # Threshold for console
  #
  PrintSeverity=info
  #
  # Threshold for syslog (none = switched off)
  #
  SyslogSeverity=none
  #
  # Threshold for forwarding to the log server
  #
```

```
ExportSeverity=crit
#
# Threshold for invoking an external program
#
ExternalSeverity=crit
#
# Threshold for logging to a SQL database
#
DatabaseSeverity=err
```

4.2. Sigurnosna politika

Prilikom korištenja bilo kojeg alata za provjeru integriteta datotečnog sustava potrebno je definirati sigurnosnu politiku prema kojoj će se provoditi provjere.
Sigurnosna politika definira koji će se objekti i atributi na sustavu pratiti te predstavlja osnovu rada programa. Loše definirana sigurnosna politika može uzrokovati veliki broj lažnih upozorenja (eng. *false positives*), koja mogu negativno utjecati na pouzdanost programa.
`Samhainrc` datoteka definira nekoliko osnovnih parametara kojima je moguće definirati sigurnosnu politiku provjere. Svaki od parametara odgovara drukčijoj sigurnosnoj politici i prilagođen je različitim tipovima datoteka i direktorija.
U sljedećoj tablici (Tablica 4) dana je lista podržanih parametara s njihovim kratkim opisom.

Parametar	Opis
`ReadOnly`	Sve promjene na objektu osim vremena pristupa biti će prijavljene kao upozorenja.
`LogFiles`	Sve promjene na objektu osim vremenskih oznaka, veličine i potpisa biti će prijavljene kao upozorenja. Parametar prilagođen praćenju log datoteka.
`GrowingLogFiles`	Sve promjene na objektu osim vremenskih oznaka i potpisa biti će prijavljene kao upozorenja. Promjena veličine biti će prijavljena samo ukoliko se veličina datoteke smanjila. Parametar također prilagođen praćenju log datoteka.
`Attributes`	Prijavljuju se promjene prava pristupa ili vlasnika objekta.
`IgnoreAll`	Ne prijavljuje se niti jedna promjena na objektu. Ukoliko se datoteka izbriše biti će prijavljen njen nestanak.
`IgnoreNone`	Biti će prijavljena bilo koja promjena nad objektom (vremena pristupa, veličina, potpis, ...).
`User0 i User1`	Korisnički podesivi parametri.

Tablica 4 - Parametri provođenja sigurnosne politike.

Osim korisnički podesivih parametara (`User0 i User1`), postoji i posebna sintaksa sa kojom je moguće modificirati inicijalno značenje osnovnih parametara. Na ovaj način korisniku se omogućuje prilagođavanje definiranih parametara svojim potrebama.
Ukoliko se prilikom provjere integriteta sustava utvrdi nelegitimna promjena u odnosu na inicijalno zabilježeno stanje, biti će prijavljeno upozorenje.
Prioritet generiranog upozorenja ovisiti će o tome koji je od gore navedenih parametara prekršen te će ovisno o definiranim pragovima biti zabilježen odgovarajućom metodom (Poglavlje 4.1).
Dio konfiguracijske datoteke (sekcija `[Event Severity]`) kojom se definira prioritet upozorenja za svaki parametar priložen je nastavku.

```
[EventSeverity]
  #
  SeverityReadOnly=crit
  SeverityLogFiles=crit
  SeverityGrowingLogs=warn
  SeverityIgnoreNone=crit
  SeverityIgnoreAll=info
  #
```

```
# these are access errors
#
SeverityFiles=err
SeverityDirs=err
#
```

4.3. Provjera SUID/SGID datoteka

Ukoliko je omogućena provjera SUID/SGID datoteka na sustavu (`--with-suidcheck` opcija), potrebno je unutar konfiguracijske datoteke definirati parametre provjere.
Dio konfiguracijske datoteke vezan za SUID/SGID provjere (sekcija `[SuidCheck]`) prikazan je u nastavku.

```
[SuidCheck]
  # activate (0 for switching off)
  SuidCheckActive=1
  # interval between checks (in seconds, default 7200)
  # SuidCheckInterval=86400
  # scheduled check at 01:30 each night
  SuidCheckSchedule=30 1 * * *
  SeveritySuidCheck=crit
  # you may manually exclude one directory
  SuidCheckExclude=/net/localhost
  # limit on files per seconds
  SuidCheckFps=250
```

U sljedećoj tablici (Tablica 5) dan je kratki opis navedenih parametara.

Parametar	Opis
`SuidCheckActive`	Omogućavanje i onemogućavanje SUID/SGID provjera.
`SuidCheckInterval`	Interval provjere u sekundama.
`SuidCheckSchedule`	Ovim parametrom moguće je definirati učestalost provjere prema sintaksi cron poslužitelja.
`SeveritySuidCheck`	Prioritet pod koji će biti generirano upozorenje, ukoliko se detektiraju nelegitimne datoteke.
`SuidCheckExclude`	Isključivanje SUID/SGID provjera unutar definiranog direktorija.
`SuidCheckFps`	Ograničavanje broja datoteka koje se provjeravaju u sekundi – vidjeti napomenu.

Tablica 5- Parametri SUID/SGID provjera

Napomena:
Budući da su SUID/SGID provjere prilično zahtjevne na I/O resurse, `SuidCheckFps` parametrom moguće je ograničiti broj datoteka koje se provjeravaju u sekundi.
Ukoliko je omogućena SUID/SGID provjera, program će prilikom inicijalizacije baze napraviti listu svih SUID/SGID datoteka na sustavu (iz provjere su isključeni ISO9660, `proc`, `vfat` i `nfs` datotečni sustavi).
Periodičkom provjerom u zadanim intervalima (parametri `SuidCheckInterval` i `SuidCheckSchedule`) provoditi će se usporedba trenutnog stanja s ranije kreiranim "otiskom" te će se prijaviti sve nelegitimne SUID/SGID datoteke. Upozorenje će biti generirano pod prioritetom definiranim `SeveritySuidCheck` parametrom.

4.4. Detekcija kernel rootkit malicioznih programa

Rootkits su maliciozni programi koje neovlašteni korisnik instalira na kompromitiranom sustavu, kako bi prikrio svoje djelovanje, ili kako bi omogućio neautorizirani pristup istom sustavu.
Kernel rootkit programi specifični su po tome što se integriraju s jezgrom operacijskog sustava (eng. *kernel*) te ih je na taj način vrlo teško detektirati bez korištenja specijaliziranih alata.

Samhain programski paket, između ostalih mogućnosti, nudi i detekciju upravo opisanih malicioznih programa (ukoliko je preveden s opcijom `--with-kcheck`).
Primjer dijela konfiguracijske datoteke kojim se definiraju parametri *kernel rootkit* provjere priložen je u nastavku (sekcije `[Kernel]`).

```
[Kernel]
  # activate (0 for switching off)
  KernelCheckActive=1
  # interval between checks (in seconds, default 300)
  KernelCheckInterval=20
  SeverityKernel=crit
```

Značenje parametara identično je onima kod SUID/SGID provjera (Poglavlje 4.3).

4.5. Detekcija prijavljivanja i odjavljivanja u sustav

Samhain program omogućuje i praćenje prijavljivanje i odjavljivanja korisnika u sustav (*login/logoff*). Na ovaj način moguće je detektirati sumnjive pristupe poslužitelju, što vrlo često može pomoći u detekciji i prevenciji malicioznih aktivnosti.
Dio konfiguracijske datoteke kojim se definiraju parametri *login/logoff* provjere priložen je u nastavku (sekcija `[Utmp]`).

```
[Utmp]
  #
  # activate (0 for switching off)
  #
  LoginCheckActive=1
  #
  # interval between checks (in seconds)
  #
  LoginCheckInterval=600
  #
  #
  SeverityLogin=info
  SeverityLogout=info
  #
  # multiple logins by same user
  #
  SeverityLoginMulti=crit
```

4.6. Bilježenje logova u bazu podataka

Osim standardnih mogućnosti bilježenja log zapisa, `samhain` program omogućuje i bilježenje log zapisa u MySQL ili Postgres bazu podataka. U tu svrhu program je potrebno prevesti s `--with-database` opcijom, koja kao argument prima vrijednosti koje označavaju koji će se tip baze koristiti, `mysql` ili `postgresql`.
U tom slučaju unutar konfiguracijske datoteke potrebno je definirati parametre koje će omogućiti komunikaciju programa s odabranom bazom (sekcija `[Database]`).

```
[Database]
  SetDBName=samhain
  SetDBTable=log
  SetDBHost=localhost
  SetDBUser=samhain
  SetDBPassword=
```

Značenje navedenih konfiguracijskih parametara dano je u sljedećoj tablici (Tablica 6).

Parametar	Opis
SuidCheckActive	Omogućavanje i onemogućavanje SUID/SGID provjera.
SuidCheckInterval	Interval provjere u sekundama.
SetDBName	Ime baze podataka.
SetDBTable	Ime tablice u koju se bilježe podaci.
SetDBHost	Ime poslužitelja baza podataka.
SetDBUser	Korisničko ime pod kojim se pristupa bazi.
SetDBPassword	Korisnička zaporka za pristup bazi.

Tablica 6- Parametri bilježenja log zapisa u bazu podataka

4.7. Ostali parametri

Osim parametara opisanih u prethodnim poglavljima, koji su točno vezani za određeno svojstvo programa, postoji još mnoštvo parametara kojima je moguće preciznije definirati način rada programa (sekcija `[Misc]`).

U sljedećoj tablici (Tablica 7) navedeni su neki od njih s kratkim opisom.

Parametar	Opis
Daemon	Ovim parametrom definira se da li će program raditi kao poslužitelj (eng *daemon*), ili ne.
SetFilecheckTime	Definiranje intervala provjere integriteta datotečnog sustava.
SetMailTime	Interval u kojem će se slati e-mail poruke s rezultatima provjera.
SetMailAddress	E-mail adresa na koju se šalju rezultati provjere.
SetMailRelay	Mail poslužitelj putem kojeg će se slati e-mail poruke.
SetLogServer	IP adresa centralnog poslužitelja na koji se šalju log zapisi.
SetTimeServer	IP adresa vremenskog poslužitelja za sinkronizaciju vremena.
SyslogFacility	Tip poruke koji će se slati syslog poslužitelju (ukoliko se isti koristi).
DigestAlgo	Algoritam koji će se koristiti za potpisivanje podataka. Inicijalno se koristi TIGER algoritam, a ovim parametrom moguće je definirati, ili SHA1, ili MD5 algoritam.

Tablica 7 - Ostali parametri samhain programa.

5. Klijent/poslužitelj način rada

U ovom poglavlju biti će nešto detaljnije opisan već ranije spomenuti klijent/poslužitelj način rada, budući da se isti pokazao kao vrlo praktično rješenje u okolinama s većim brojem poslužitelja.

Na poslužitelje koji se žele kontrolirati ovim putem potrebno je instalirati `samhain` klijent aplikaciju koja će obavljati provjere integriteta prema zadanoj sigurnosnoj politici, a jedno računalo potrebno je odabrati za centralni poslužitelj (`yule`) koji će predstavljati jezgru cijelog sustava.

Poslužiteljsku aplikaciju potrebno je u tom slučaju prevesti s `--enable-network=server`, a klijent aplikaciju sa `–enable-network=client` opcijom (ostale opcije ovisiti će o potrebama korisnika).

Nakon toga potrebno je sve klijente registrirati kod poslužitelja, kako bi se na taj način omogućila njihova međusobna komunikacija. Sve veze prema poslužitelju sa onih klijenata koji nisu registrirani biti će prekinute.

Postupak registracije klijenata nešto je složeniji i biti će opisan u nastavku poglavlja.

Unutar konfiguracijske datoteke `yule` poslužitelja (`/etc/yulerc`) nalazi se sekcija `[Clients]` unutar koje je potrebno dodati posebno formirani zapis za svakog od klijenata. Npr.

```
[Clients]
```

```
#
# A client
#
Client=HOSTNAME_CLIENT1@salt1@verifier1
#
# another one
#
Client=HOSTNAME_CLIENT2@salt2@verifier2
```

Kako se može uočiti iz navedenog primjera, zapis za svakog klijenta sljedećeg je formata.

```
Client=HOSTNAME@salt@verifier
```

Polje `HOSTNAME` potrebno je zamijeniti s FQDN imenom svakog od klijenata, dok je preostala polja (*salt* i *verifier*) potrebno zasebno generirati.
U prvom koraku potrebno je odabrati odgovarajuću zaporku koja će biti ugrađena u `samhain` aplikaciju klijenta za kojeg se generira zapis. Svaka klijent aplikacija ima u sebi ugrađenu inicijalnu zaporku koja je postavljena za vrijeme instalacije programa (zaporka je korisnički podesiva putem `--enable-base` parametra) i vrlo je važna za rad programa.
Zapis je moguće kreirati na temelju inicijalne zaporke programa (ukoliko je ista poznata), ili je moguće generirati novu. U svrhu generiranja nove zaporke može pomoći `yule` poslužitelj, ukoliko je pokrenut s `–gen-password` parametrom.
Npr.

```
# yule –gen-password
05CB7E4CA9A5DB49
```

Dobivenu zaporku potrebno je ugraditi u `samhain` aplikaciju klijenta koji se želi registrirati, budući da će povezanost zaporke sa generiranim zapisom biti ključna za legitiman pristup `yule` poslužitelju.
Ugradnju zaporke potrebno je obaviti pomoću za to predviđenog `samhain_setpwd` programa. Sintaksa korištenja programa je sljedeća,

```
# samhain_setpwd <filename> <suffix> <new_password>
```

gdje proslijeđeni parametri imaju sljedeće značenje,

- `filename` – ime programa unutar kojega se želi ugraditi zaporka (`samhain`);
- `suffix` – nastavak kojim će biti označena verzija programa s novo ugrađenom zaporkom (`samhain.new`);
- `new_password` – nova zaporka koja se ugrađuje u program.

Na konkretnom primjeru to izgleda ovako,

```
# samhain_setpwd samhain new 05CB7E4CA9A5DB49
```

Rezultat pokretanja ove naredbe biti će datoteka `samhain.new`, u koju će biti ugrađena navedena zaporka.
Nakon ugradnje zaporke potrebno je pomoću `yule` programa kreirati odgovarajući zapis koji će odgovarati upravo toj klijent aplikaciji, odnosno zaporci. Zapis se generira na sljedeći način:

```
# yule -P 05CB7E4CA9A5DB49

Client=HOSTNAME@E25F7EA09A755B14@1EC6D15FE27D890D31FD4B2259C635F2097
990E9DC573016280015B3CB30F0ACB09C7F9FE61A006ACFD9A43DF169015C385FD42
E5D4EEE9774A2A17D97AB909CF33CA46A67DA39888060C38511BF35FADCA75937CBE
84F88EBEB11880F461A8DBA1D0FF70233CEA72EE7E6F6B2921E04A09CCA55E45E4A7
75DDC5F646472C275
```

Polje HOSTNAME potrebno je zamijeniti FQDN imenom računala na kojem se klijent nalazi te je tako formirani zapis potrebno kopirati u konfiguracijsku datoteku yule poslužitelja unutar [Client] sekcije.
Cijeli postupak bitno je jednostavniji ukoliko se ne želi mijenjati inicijalno ugrađena zaporka. U tom slučaju potrebno je u zadnjem koraku kao argument yule programu proslijediti postojeću, inicijalno ugrađenu zaporku, umjesto one novo kreirane. Inicijalna zaporka korisniku se prikazuje nakon pokretanja configure skripte te ju je u tom slučaju potrebno zapamtiti.
Opisani postupak potrebno je ponoviti za svaki od klijenata.
Nakon što je klijent registriran kod yule poslužitelja, potrebno je unutar njegove konfiguracijske datoteke omogućiti komunikaciju s poslužiteljem.

```
[Log]
  #
  # Threshold for forwarding to the log server
  #
  ExportSeverity=crit

  [Misc]

  SetLogServer=192.168.1.1
```

U slučaju velikog broja klijenata, praktično je na poslužitelju instalirati podršku za bilježenje log zapisa u bazu podataka (MySQL ili Postgres). Na taj način sva će upozorenja s udaljenih računala biti pohranjena u bazu koju je onda moguće pretraživati prema različitim kriterijima.
Ovisno o tipu, bazu je moguće kreirati na jedan od sljedećih načina.
MySQL

```
# mysql -p -u samhain < samhain.mysql.init
```

Postgres

```
# su postgres
# createdb samhain
# createuser samhain
# psql -d samhain < samhain.postgres.init
# exit
```

Ovim postupkom kreirati će se samhain baza podataka, koja će omogućiti pohranjivanje podataka. Logiranje u bazu podataka moguće je realizirati i na klijentskim računalima, ukoliko se za to ukaže potreba.
Nakon što je na klijentskim računalima podešeno slanje podataka na centralni yule poslužitelj, moguće je na njima onemogućiti lokalno bilježenje upozorenja, kako bi se smanjila redundancija podataka.

6. Provjera integriteta log zapisa i e-mail poruka

Prilikom razvoja samhain programskog paketa posebna pažnja posvećena je sigurnosti. Integritet generiranih upozorenja jedan je od elemenata o kojima se posebno vodilo računa. Na taj način željela se u minimalizirati mogućnost koruptiranja log zapisa, što je jedan od ključnih elemenata za pouzdanost programa ovakvog tipa.
Kao jednu od mogućnosti obavješćivanja administratora, samhain program koristi e-mail servis. Putem ugrađenog SMTP poslužitelja program će rezultate provjera slati na adresu administratora sustava.
Razlog korištenja zasebnog mail poslužitelja još je jedan od načina kojim se željela povećati pouzdanost programa.
Poznatiji Linux programi za slanje e-mail poruka (npr. sendmail) koriste praksu da sve poruke koje trenutno ne mogu ispostaviti, pohrane lokalno na tvrdi disk.

Iako je ovakav pristup posve bezopasan sa stanovišta mail servisa, u ovom slučaju ta mogućnost unosi određeni sigurnosni rizik. Budući da poruke sadrže povjerljive informacije o stanju sustava, potrebno je osigurati njihov integritet, odnosno nemogućnost promjene. Upravo iz tog razloga `samhain` program koristi zasebni mail poslužitelj, koji sve takve poruke umjesto na tvrdom disku, čuva u radnoj memoriji.
Tijelo poruke sastoji se od nekoliko segmenata koji opisuju rezultat provjere, zajedno s digitalnim potpisom koji garantira njezin integritet. Digitalni potpis generiran je na temelju sadržaja poruke i slučajno generiranog ključa. Svaka nova poruka biti će kriptirana novim ključem koji je dobiven iz prethodnog (eng. *hash chain*). Inicijalni ključ kojim je kriptirana prva poruka poslan je u prvoj poruci i kriptiran je *one-time pad* algoritmom.
E-mail poruka koja sadrži inicijalni ključ sljedećeg je formata.

```
-----BEGIN MESSAGE-----
  message
  -----BEGIN LOGKEY-----
  Key(48 chars)[timestamp]
  -----BEGIN SIGNATURE-----
  signature
  ID TRAIL_ID:hostname
  -----END MESSAGE-----
```

Ostale poruke vrlo su slične s osnovnom razlikom što ne sadrže sekciju `BEGIN LOGKEY`.

```
-----BEGIN MESSAGE-----
  first message
  second message
  ...
  -----BEGIN SIGNATURE-----
  signature
  ID TRAIL_ID:hostname
  -----END MESSAGE-----
```

Integritet poruke moguće je provjeriti naredbom,

```
# samhain -M /mailbox
```

gdje ime `mailbox` označava korisnički spremnik poruka.
Budući da postoji veza između ključa ugrađenog u `samhain` aplikaciju i ključa kojim se potpisuju e-mail poruke, provjeru integriteta moguće je obaviti samo sa onom aplikacijom koja je poruku generirala. Ukoliko to nije slučaj biti će prijavljena greška.
Na sličan način `samhain` program može rezultate provjere bilježiti i u lokalnu log datoteku (`/var/log/samhain_log` prema inicijalnoj instalaciji).
Svaka poruka sadržavati će u tom slučaju digitalni potpis koji je dobiven na temelju same poruke i istog ključa kojim je kriptirana prva e-mail poruka. Svaka sljedeća poruka potpisuje se novim ključem koji je izveden iz prethodnog ključa (identično kao i kod mail poruke).
Primjer log zapisa generiranih `samhain` programom dan je u nastavku:

```
CRIT   :  [2003-01-07T16:17:05+0100] msg=<No such file or
directory>, interface=<lstat>, path=</etc/shadow.backup>
F4A0D2133F79BA4C29DBA6B2E6E33021D60402208B438E31
CRIT   :  [2003-01-07T16:17:05+0100] msg=<POLICY MISSING>,
path=</etc/shadow.backup>
96771EBFC23EAD5ECB12D239153C449E09E8E613DDCE7654
CRIT   :  [2003-01-07T16:17:05+0100] msg=<No such file or
directory>, interface=<lstat>, path=</etc/passwd.backup>
651E4B2428F2E6224F98718F369DB54FB461F114E7054A24
```

```
CRIT   :  [2003-01-07T16:17:05+0100] msg=<POLICY MISSING>,
path=</etc/passwd.backup>
6C8A125BD3C35318E95B005565BB4EDABEA3CB1FCAB29CCA
```

Integritet log zapisa moguće je provjeriti naredbom:

```
# samhain -L /var/log/samhain_log
```

Zadavanjem ove naredbe korisnik će biti zatražen da unese ključ koji je poslan u prvoj poruci. Ukoliko ključ ne odgovara onome kojim su potpisane poruke, biti će prijavljena greška.

7. Zaključak

`Samhain` programski paket pokazao se kao prilično kvalitetan alat namijenjen detekciji i prevenciji neovlaštenih aktivnosti. Program podržava nekoliko različitih načina rada, koji će zadovoljiti većinu potreba korisnika.Mogućnost centraliziranog nadzora više `samhain` klijenata posebno je praktična mogućnost kojom se program odlikuje.
Iako program osim provjere integriteta datotečnog sustava nudi još neke mogućnosti, čini se da je ipak to njegova najveća kvaliteta. Ostala svojstva kao što su detekcija SUID/SGID datoteka, detekcija *kernel rootkit* programa, praćenje prijavljivanja i odjavljivanja u sustav dodaci su koji programu daju dodatnu snagu, i koji ga čine drugačijim od ostalih programa za provjeru integriteta (Tripwire, AIDE, i sl.).