Avtomatizacija varnostnih nastavitev strežnika Debian

Damjan Casar, Borko Bošković, Janez Brest, Viljem Žumer

Univerza v Mariboru
Fakulteta za elektrotehniko, računalništvo in informatiko
Smetanova ulica 17, 2000 Maribor
E-pošta: damjan.casar@uni-mb.si

Safe Protection of Debian Based Server

This paper presents an automatisation script for the initial install, setup and control of the server, based on a GNU/Linux Debian distribution. Emphasis is mainly on system security. The core of our work is to prepare a Bash script for setting up basic security. The script itself takes care of the instalation of the software needed for remote computer access, web server access and the program for making time related setup messages. All of the installed software is then configured for optimally secure operations, the script then disables the safety lack system services and it checks the sistem's integrity.

1 Uvod

Upravljanje in kontroliranje računalnika ali omrežja oz. sistemska administracija, je dandanes že zelo razširjena. Z vedno večjim številom računalnikov, se zvišuje tudi zahtevnost njihove administracije. Npr., administracija petih računalnikov je preprosta. Ko pa imamo več sto računalnikov, se moramo poslužiti avtomatizacije oz. skript, ki opravijo administracijo.

Cilj članka je predstaviti sistemsko konfiguracijo s pomočjo skripte, ki samodejno nastavi sistem tako, da je čim bolj varen. Naloga skripte je, da avtomatizira postopek nameščanja programske opreme, njeno konfiguracijo in preverjanje celovitosti operacijskega sistema. Skripta je namenjena operacijskemu sistemu debian, ki je le ena od mnogih GNU/Linux verzij oz. distribucij.

V drugem poglavju predstavimo odprtokodne operacijske sisteme. Nato v tretjem poglavju na kratko opišemo naše delo. Četrto poglavje je namenjeno predstavitvi uporabljenih programov. V petem poglavju opišemo celotno delovanje skripte, ter nato v šestem podamo kratki zaključek.

2 Operacijski sistemi Debian

Odprtokodni operacijski sistem Debian je ena od distribucij GNU/Linux, katerih jedro je razvil Linus Torvalds. Razvit je pod licenco GNU GPL (General Public Licence)(R. Stallman 1992), kar pomeni, da ima odprto kodo in je prosto dostopen širši javnosti.

Prednost odprtokodnih operacijskih sistemov GNU/Linux je njihova cena, saj nam za njih ni potrebno plačati. Za profesionalne različice, ki so namenjene podjetjem za delovne postaje in strežnike lahko plačamo tehnično podporo. Tako kot večina ostalih operacijskih sistemov imajo GNU/Linux distribucije možnost namestitve namizne ali strežniške različice. Namizna (angl. *Desktop*) različica je namenjena vsakodnevni uporabi z grafičnim vmesnikom. Vključuje uporabniške programe, ki jih porabnik lahko uporablja za večino opravil, z možnostjo namestitve dodatne programske opreme. Strežniška (angl. *Server*) različica je namenjena strežnikom. Njen cilj ni uporabniška udobnost in prijaznost, ampak čim večja stabilnost, ter varnost operacijskega sistema.

Varnost je pri strežnikih ključnega pomena. Zato morajo distribucije, namenjene strežnikom, zagotavljati varnost že ob začetku namestitve. Ker se strežniki glede na uporabo razlikujejo, so ob namestitvi začetne nastavitve poenostavljene. Vsak strežnik je tako potrebno posebej tudi nastaviti oz. prilagoditi na njegove naloge.

Operacijski sistem GNU/Linux smo izbrali zaradi svoje vsestranskosti in odprtosti. Ena od velikih prednosti GNU/Linux sistemov je ta, da obstaja manjše število zlonamerne kode, ki lahko zmanjša varnost sistema. Primer zlonamerne kode so npr. virusi. Ob morebitni zaskrbljenosti glede varnosti, lahko namestimo tudi kakšen odprtokodni prostodostopni protivirusni program. Ti operacijski sistemi so tudi odlična izbira za spletne strežnike in razvojna okolja. Privzeto imajo integriran požarni zid (angl. *firewall*), katerega je potrebno nastaviti, za kar pa imamo na voljo kar precej različnih grafičnih in tekstovnih nastavitvenih orodij oz. programov.

3 Kratek opis dela

Naš cilj je narediti skripto, ki bo pregledala sistem ali obstajajo sistemske posodobitve, namestila vso potrebno programsko opremo, nas obveščala o morebitnih spremembah sistema in bo dnevno izvajala varnostne preglede, ter ugotovitve zapisala v dnevnik.

Skripta bo administrirala sistem avtomatično in tako administratorju prihranila čas. Opravljala bo naslednja opravila:

- pregleda sistem ali obstajajo posodobitve in jih namesti,

- onemogoči branje lokalnih medijev *cd-rom* iz datoteke sistemskih virov in vse sproti izpisuje,
- poišče za nameščeno programsko opremo: ssh-server, logwatch, apache2. Če na sistemu določene programske opreme ni, dobimo ponujeno možnost, da jo namestimo,
- izklopi storitve, ki zmanjšajo varnost sistema, kot so npr: finger, telnet in z uporabniško odločitvijo tudi ftp,
- namesti programe ssh, apache2 in logwatch, ki izboljšajo varnost sistema,
- nastavi privzetega uporabnika, ki bo dobival dnevna poročila sistemskih dogodkov v obliki elektronske pošte,
- onemogoči dostop s kateregakoli spletnega naslova, zato dobimo možnost dodajanja naslovov IP in domenskih imen, ki jim želimo omogočiti dostop do sistema,
- na uporabniško določeno mesto shrani spisek SUID in SGID datotek za kasnejšo medsebojno primerjavo,
- nastavi vsakodnevno izvajanje preverjanja sistema,
- ob vsakodnevnem pregledu ustvari datoteko o spremljanju dejavnosti sistema, v katero se shranijo informacije o delovanju skripte,
- zažene še čiščenje začasnih lokalnih namestitvenih paketov.

4 Uporabljena orodja

Za omogočanje čim boljše varnosti smo uporabili določena orodja, ki jih opisujemo v tem poglavju. Vsako orodje ima svojo določeno nalogo, za povečanje varnosti sistema.

4.1 Varna lupina SSH

Varna lupina SSH (angl. *Secure Shell*) je varen način povezovanja med računalniki. Podobna storitev je telnet, ki pa ima veliko pomanjkljivost, da je pošiljanje podatkov nešifrirano.

Vsako morebitno poslušanje takšnega prometa, lahko privede do odkrivanja gesel in drugih zaupnih podatkov. Varna lupina uporablja za varno povezavo algoritem javnih in zasebnih ključev. Za izmenjavo šifrirnih ključev potrebuje vsak uporabnik dva šifrirna ključa. En ključ imenujemo javni ključ, delimo ga lahko javno, drugega pa imenujemo privatni ključ in ga skrivamo pred vsemi uporabniki. Sporočila šifriramo s prejemnikovim javnim ključem, dešifriramo pa jih lahko le s pripadajočim privatnim ključem. Povezava med ključema je matematična, vendar iz javnega ključa ne moremo dešifrirati privatnega ključa.

4.2 Logwatch

Logwatch je sistemski nadzornik, ki spremlja vse spremembe na sistemu. Ob določeni uri, ki jo nastavimo v konfiguracijski datoteki, lahko pošlje administratorju oz. določenemu uporabniku operacijskega sistema elektronsko sporočilo. Ta sporočila vsebujejo spremembe in povezave na sistem.

4.3 Strežnik časovnih opravil Cron

Velikokrat na sistemu želimo poganjati določene aplikacije oz. ukaze ob določeni uri, ko sistem ni obremenjen. Za to opravilo imamo na voljo program Cron, ki je sistemsko integrirani strežnik časovnih opravil. Tej storitvi lahko podamo kdaj se naj ukazi izvršijo in v katerem časovnem intervalu se bodo ponavljali. Čas kdaj se naj določena serija ukazov izvede podamo s ukazom "crontab", ki jih bomo prikazali v poglavju 5.

5 Avtomatizacija zaščite

Ob zagonu skripte se kot prvo zažene primerjava nameščenih programskih paketov s paketi v spletnih skladiščih, ki preveri ali je sistem posodobljen ali ne. Ker se s tem preveri le posodobljenost sistema, moramo na sistemu zastarelo programsko opremo tudi nadgraditi.

```
apt-get update -qq
apt-get upgrade -qqy
```

Preverjanje sistemskih virov na sistemu ima kot lokalni vir dodan *cd-rom*. Naš sistem je spletni strežnik, zato lokalne vire, kot je *cd-rom* onemogočimo.

```
source_isk=`cat /etc/apt/sources.list
 | grep "cdrom" | grep -v "#" | wc -l`
if [ $source_isk -eq 1 ]
then
  sed -i 's/deb cdrom/#&/g'
  /etc/apt/sources.list
fi
```

Večina distribucij GNU/Linux ima odjemalca (angl. *Client*) SSH že nameščenega na sistem. Če želimo do sistema dostopati z oddaljenega računalnika, potrebujemo tudi strežnik (angl. *Server*) SSH.

Skripta preveri, če je storitev (angl. *Daemon*) z imenom *sshd* že zagnana. V mapi */etc/init.d* se nahajajo vse storitve. To pot shranimo v spremenljivko z imenom *iskanje* kot rezultat poizvedbe. Če je storitev nameščena na sistem, potem dobimo izpis te storitve, v nasprotnem primeru ne dobimo izpisa. Nato z ukazom *wc* preberemo število izpisanih vrstic, ki jih nato s pogojem lahko obravnavamo kot nič ali več nameščenih storitev. Če storitev ni nameščena jo namestimo.

```
iskanje=`find /etc/init.d -name "ssh"
 | wc -l`
 if [ $iskanje -gt 0 ]
   then
     echo "Sshd is running."
   else
     apt-get -qqy install ssh
 fi
```

Varna lupina je sama po sebi precej varna, saj uporablja šifrirani prenos podatkov po omrežju. Kot privzeto možnost ima sistemski administrator (angl. *root*) omogočeno prijavo z oddaljenega računalnika. S tem postane računalnik zelo odprt za t. i. napade *brute force*, ki s poizkušanjem ugotavljajo uporabniška imena

in pripadajoča gesla. Ko napadalec preko uporabnika *root* ugotoviti geslo, ima popoln oddaljen nadzor nad računalnikom. Tako smo s spodnjim delom kode onemogočili neposredno oddaljeno prijavo uporabnika *root*. Po onemogočanju prijave smo storitev *sshd* ponovno zagnali, da so se spremenjene nastavitve uveljavile.

```
ssh_iskanje=`cat /etc/ssh/sshd_config
  | grep "PermitRootLogin"
  | awk -F " " '{print $2}'`
  if [ $ssh_iskanje == "yes" ]
    then
      cat /etc/ssh/sshd_config
      | grep "PermitRootLogin"
      | sed -i 's/yes/no/g'
      /etc/ssh/sshd_config
      /etc/init.d/ssh restart
  fi
```

Naslednja sistemska storitev za nameščanje je t. i. Logwatch, ki lahko pošilja dnevna, tedenska ali mesečna poročila o stanju opazovanega sistema. Storitev vsako spremembo na sistemu beleži in jo ob določenem času pošlje prednastavljenemu uporabniku na njegov elektronski naslov. Logwatch storitev ni v mapi */etc/init.d*, kakor je bilo v primeru *sshd strežnika*, zato preiščemo datoteko *logwatch.conf*, ki je konfiguracijska datoteka te storitve. Če ta datoteka obstaja je program že nameščen, drugače ga namestimo. Po namestitvi spremenimo v konfiguracijski datoteki uporabnika kamor se bodo pošiljala dnevna poročila.

```
isk_log=`find / -name "logwatch.conf"
  | wc -l`
    if [ $isk_log -gt 0 ]
      then
        echo "Logwatch installed."
      else
        apt-get -qqy install logwatch
    fi
isk_log=`find / -name "logwatch.conf"
 | grep "default"`
cat $isk_log | grep "Output"
  | grep -v "#" | sed -i 's/stdout/mail/g'
  $isk_log
mailto=`cat $isk_log | grep "MailTo"
  | awk -F " " '{print $3}'`
read uporabnik
sed -i 's/'$mailto'/'$uporabnik'/g'
 $isk_log
```

Apache je spletni strežnik namenjen tako podjetjem kot tudi domačim uporabnikom. Njegova funkcionalnost in uporabnost se zveča z dodajanjem in uporabo dodatnih modulov. Na sistem namestimo paket *apache2* s podobnim postopkom, kot pri sshd storitvi.

Nastavljene privzete varnostne nastavitve spletnega strežnika so dokaj dobre, vendar so lahko še varnejše. Odvisno je le, kaj in koliko od privzetih možnosti smo pripravljeni "žrtvovati" za bolj varen spletni strežnik oz. za zadovoljivo razmerje med varnostjo in uporabnostjo. Privzete nastavitve podajajo na strežnikovi spletni strani podatke o operacijskem sistemu, nameščeni verziji spletnega strežnika, podatkovni bazi mysql, php in drugo. Iz varnostnih razlogov te informacije s pomočjo skripte izklopimo. Čeprav so ti podatki lahko koristni za administratorja, so koristni tudi za potencialnega napadalca na sistem. Tako, poiščemo lokacijo kjer se nahaja datoteka *apache.conf*, s tem tudi preverimo njen obstoj. V datoteki spremenimo čas neaktivne seje iz 300 na 45 sekund, da zmanjšamo možnost neavtoriziranega dostopa.

```
isk_apache=`find / -name "security"
 | grep apache2`
isk_sig=`cat $isk_apache
 | grep ServerSignature | wc -l`
  if [ $isk_sig -eq 1 ]
    then
      sed -i '$a\ServerSignature Off'
      $isk_apache
  fi
isk_time=`cat $isk_apache
 | grep "Timeout" | grep -v "#"
 | grep -v "KeepAliveTimeout"
 | awk -F " " ' {print $2}'`
  if [ $isk_time -gt 50 ]
    then
      cat $isk_apache | grep "Timeout"
      | sed -i 's/300/45/g' $isk_apache
  fi
```

Določene sistemske storitve so na sistemu potencialne varnostne luknje, bodisi zaradi funkcionalnosti, bodisi zaradi njihovih pomanjkljivosti. Zaradi tega smo onemogočili na sistemu storitvi *finger* in *telnet*, ter storitev *ftp* podali kot možnost izklopa uporabniku.

Storitev *finger* nam izpiše podatke o trenutno prijavljenih uporabnikih na sistemu. Ti podatki so lahko pomembni za napadalca na oddaljenem sistemu, zato to storitev izklopimo. S skripto pregledamo zagnane procese, in če je *finger* med njimi, ga prekinemo. Po istem postopku prekinemo delovanje storitve *telnet*, ki je namenjen oddaljenemu dostopu do računalnika. Storitev *ftp*, pa je opcijska in jo uporabnik lahko onemogoči.

```
isk_fin=`ps aux | grep finger | wc -l`
  if [ $isk_fin -gt 1 ]
    then
      killall finger
      /usr/sbin/update-inetd
      --disable finger
  fi
```

Sistem nam omogoča dodeljevanje dostopnih pravic točno določenim domenskim imenom oz. IP naslovom kar imenujemo (angl. *TCP wrapper*). Za onemogočanje dostopa uporabljamo datoteko */etc/hosts.deny*, v katero zapišemo "ALL:ALL" in tako onemogočimo dostop do sistema s kateregakoli naslova. V datoteko */etc/hosts.allow*, preko poziva ukazne lupine vnašamo domenska imena oz. IP naslove, katerim želimo omogočiti oddaljeni dostop do sistema.

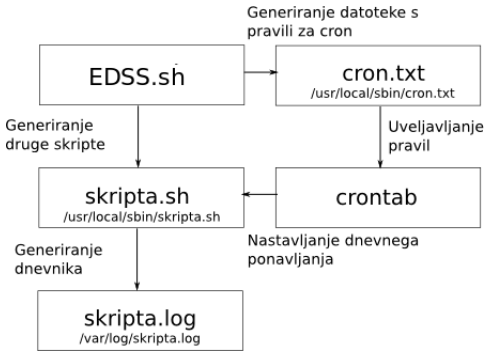
```
isk_host_deny=`cat /etc/hosts.deny
 | grep -v "#" | wc -l`
  if [ $isk_host_deny -eq 1 ]
    then
      sed -i '$a\ALL:ALL' /etc/hosts.deny
    else
      echo
  fi
while [ $hostname != "q" ]
  do
    echo "ALL:" $hostname >>
    /etc/hosts.allow
    read hostname
  done
```

SUID oz. SETUID je okrajšava za "Set User ID", SGID oz. SETGID pa okrajšava za "Set Group ID". To sta parametra zaščite datotek in omogočata uporabnikom poganjanje datotek, ki vsebujejo nastavitveni bit S v privilegiranem načinu. Če določeno datoteko označimo z bitom

S, so to lahko velika varnostna tveganja. Vsako datoteko z bitom S poganjamo s pravicami lastnika, čeprav nismo lastnik mi. Zato je najbolje imeti na sistemu čim manj takšnih datotek. Tiste pa, ki že morajo biti, naj so pod skrbnim nadzorom.

Ob namestitvi operacijskega sistema se na sistemu nahajajo datoteke, ki vsebujejo t.i. bit S. Ker vemo, da obstajajo jih moramo najti. Najdemo jih s sistemskim programom *find*, kateremu dodamo *-perm* zastavico (angl. *flag*) za iskanje pravic, ter vrednost pravice. V našem primeru je za uporabnika root vrednost *-4000*, za root uporabniško skupino pa vrednost *-2000*. Programsko zastavico "-o" uporabljamo, ko želimo dodati več iskalnih parametrov. Iz dela skripte je razvidno, da datoteke iščemo dvakrat, njihove poizvedbe pa shranjujemo v dve različni datoteki. Ideja je, da se ena datoteka s spiskom datotek, ki vsebujejo bit S shrani na "read-only", to je samo bralni medij, kot je *cd-rom* ali disketa. Nato primerjamo spisek shranjen na nespremenjenem *read-only* mediju s trenutnimi datotekami, ki vsebujejo bit S, na sistemu.

```
read suid_pot
find / \( -perm -4000 -o -perm -2000 \)
-print >> $suid_pot/s_datoteke.txt
find / \( -perm -4000 -o -perm -2000 \)
 -print >> /usr/local/sbin/s_datoteke2.txt
```



Slika 1: Diagram poteka ustvarjanja datotek

Diagram na sliki 1 nam pomaga razumeti delovanje celotne skripte, ki smo jo poimenovali Easy Debian Security Script oz. krajše EDSS. Glavna skripta *EDSS.sh* je namenjena namestitvi in nastavitvi uporabljenih programov. Poženemo jo enkrat, takoj po namestitvi operacijskega sistema. Za vsakodnevno ponovitev in pregled celovitosti sistema, pa uporabimo *skripto.sh*. Datoteka z imenom *cron.txt* vsebuje pravila za vsakodnevno zaganjanje *skripte.sh* s pomočjo časovnega strežnika cron. Z ukazom *crontab* pravila iz datoteke *cron.txt* uvozimo v časovni strežnik. *Skripta.sh* vpisuje kot dnevnik dogodkov (angl. *Event Log*) vsakodnevne zagone *skripta.sh*.

S *skripto.sh*, ki jo s pomočjo *cron-a* poganjamo vsakodnevno, pregledujemo sistem in iščemo morebitne posodobitve, ter jih namestimo. Prav tako pregledamo omogočene in onemogočene sistemske storitve (finger, telnet, ftp). Preverjamo spiska datotek, ki vsebujeta bite S na sistemu.

Primer izpisa datoteke *skripta.log*:

```
Wed Dec  2 13:32:15 CET 2009
Pregledujem sistem za potencialnimi nadgradnjami.
Nameščam vse možne nadgradnje.
Finger storitev izklopljena.
Telnet storitev izklopljena.
Ftp storitev izklopljena.
Sistemske datoteke so enake.
**********************************************
Wed Dec  2 13:36:20 CET 2009
Pregledujem sistem za potencialnimi nadgradnjami.
Nameščam vse možne nadgradnje.
Finger storitev izklopljena.
Telnet storitev izklopljena.
Ftp storitev izklopljena.
Sistemske datoteke so enake.
**********************************************
```

Na koncu skripte poženemo "apt-get clean", ki počisti začasne namestitvene pakete, shranjene na disku.

6 Zaključek

Varnost računalniških sistemov je ključnega pomena. Povečanje varnosti zahteva izključevanje določenih sistemskih storitev. Osnovna funkcija varnosti je ohranjanje celovitosti podatkov pred nepooblaščenimi osebami ali organizacijami. Da poskrbimo za varnost na sistemu, se torej poslužujemo določenih metod za ohranjanje varnosti. Za čim boljšo varnost lahko sistem nastavimo sami ali pa se poslužujemo določenih programov ali skript. Ker je slednji način boljši, ob upravljanju z več računalnikov, smo se ga poslužili tudi mi. S pomočjo avtomatizacijske skripte, ki je sistem preverila za posodobitvami in jih nato namestila, onemogočila določene sistemske storitve, ter namestila in nastavila določeno programsko opremo je sistem postal bolj varen.

Skripta je tudi javno dostopna na sistemu za odprtokodne programe Sourceforge na spletnem naslovu: (http://sourceforge.net/projects/edss/).

Literatura

[1] S. Powers, J. Peek, T. O'Reilly, M. Loukides: Unix Power Tools, Third Edition, 2002

[2] R. Flickenger: Linux Server Hacks, First Edition, 2003

[3] M. Welsh, P. Hughes, D. Bandel, B. Beletsky, S. Dreilinger, R. Kiesling, E. Liebovitch, H. Pierce: Linux Installation and Getting Started, 1996, http://tldp.org/LDP/gs/

[4] J. Fernández-Sanguino Peña: Securing Debian Manual, Version: 3.6, 2006

[5] C. Albing, JP Vossen, C. Newham: bash Cookbook, 2007

[6] W. von Hagen: Ubuntu Linux Bible, 2007

[7] M. Bauer, Linux server security, 2005.

[8] J. Tackett Jr., D. Gunter: Using Linux, Second Edition, Que Corporation, 1996

[9] I. Verdonik, T. Bratuša: Hekerski vdori in zaščita, Založba Pasadena, 2005

[10] P. Anderson: Kako v Linuxu?, Založba Pasadena (LUGOS), 2002

[11] A. Weeks: The Linux System Administrator's Guide, Custom Publishing, 2007

[12] http://www.debian-administration.org/articles/56

[13] http://www.debianhelp.co.uk/logwatch1.htm