ŠOLSKI CENTER VELENJE

VIŠJA STROKOVNA ŠOLA

DIPLOMSKA NALOGA

Velenje, Januar 2010 Damjan CASAR

ŠOLSKI CENTER VELENJE
VIŠJA STROKOVNA ŠOLA
Trg Mladosti 3
3320 VELENJE

PODJETJE: ŠOLSKI CENTER VELENJE

DIPLOMSKA NALOGA
V VIŠJEŠOLSKEM STROKOVNEM IZOBRAŽEVALNEM PROGRAMU
INFORMATIKA

VARNOSTNA ZAŠČITA DEBIAN STREŽNIKA

Avtor: Damjan CASAR, elektrotehnik elektronik

Mentor: doc. dr. Renato LUKAČ

Somentor: Anita URAN, univ. dipl. inž. el.

Velenje, Januar, 2010

Zahvala

Zahvaljujem se mentorju doc. dr. Renatu Lukaču in somentorici Aniti Uran, univ. dipl. inž. el. za strokovno pomoč in napotke pri izdelavi diplomske naloge. Zahvaljujem se tudi vsem sodelavcem laboratorija za računalniške arhitekture in jezike na Fakulteti za elektrotehniko, računalništvo in informatiko Maribor.

Še posebej se zahvaljujem prijateljem in staršem, ki so me podpirali in vzpodbujali pri mojem delu.

Ključne besede: varnost, zaščita, debian, strežnik

VARNOSTNA ZAŠČITA DEBIAN STREŽNIKA

Povzetek

V nalogi predstavljamo avtomatizacijsko skripto za začetno namestitev, nastavitev in nadzor strežnika, temelječega na distribuciji GNU/Linux Debian.

Jedro naloge je pripraviti programsko skripto Bash za nastavitev osnovnih funkcij delovanja strežnika. Poudarek je predvsem na varnosti. Skripta poskrbi za nameščanje programske opreme za dostop do računalnika, spletnega strežnika in programa za izdelavo časovno nastavljivih sporočil. Vso nameščeno programje nastavi za čim bolj varno delovanje, onemogoči varnostno pomanjkljive sistemske storitve in pregleda celovitost sistema.

Keywords: safety, protection, debian, server

SAFE PROTECTION OF THE DEBIAN BASED SERVER

Abstract

Thesis presents an automatisation script for the initial install, setup and control of the server, based on a GNU/Linux Debian distribution.

The core of our work is to prepare a Bash script for setting up basic functions of the server. Emphasis is mainly on systems 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 optimaly secure operation, the script then disables the *safety lack* system services and it checks the sistem's integrity.

Kazalo

Slike

Simboli in oznake

GPL & GNU General Public License – licenca GPL za prosto dostopno programsko opremo

BASH – prostodostopna ukazna lupina napisana za GNU projekt

Debian – odprtokodna GNU/Linux distribucija

SSH – protokol za varno povezavo dveh omrežij

Logwatch – program za pošiljanje dnevnih poročil sprememb sistema

Apache – odprtokodni spletni strežniški HTTP program

Finger – preprost omrežni protokol za posredovanje uporabniških informacij

Telnet – protokol za povezavo dveh omrežij

FTP – protokol za prenos datotek preko dveh omrežij

IP – protokol za prenos podatkov preko omrežij

SUID ali SETUID – uporabniške pravice sistemov UNIX

SGID ali SETGID – uporabniške pravice sistemov UNIX

Flags – zastavice, dodatne možnosti programov

CD-ROM – optični medij

RCP – ukaz na sistemih UNIX za oddaljeno kopiranje

X11 – računalniški programski grafični omrežni protokol

Root – sistemski administrator

HTTP – aplikacijski omrežni protokol

Inetd – sistemski servis za upravljanje z internetnimi storitvami

TCP – transportni kontrolni protokol

read-only – omogočena je samo možnost branja

1. Uvod

Operacijski sistem GNU/Linux. Dejstvo je, da ga premalo ljudi pozna [1], prav tako ne poznajo njegovih prednosti pred komercialnimi operacijskimi sistemi [2]. Cilj naloge je predstaviti varno sistemsko konfiguracijo na čim enostavnejši način, tako da bi se čim več ljudi odločalo za te alternativne sisteme.

Sistemska administracija, t.i. upravljanje in kontroliranje računalnika ali omrežja, je dandanes že zelo razširjena, vendar z njeno razširitvijo prihaja še več računalnikov, katere je potrebno vzdrževati. Administracija petih računalnikov je lahko preprosta in jo je možno opraviti tudi ročno. Ko imamo naenkrat po več sto računalnikov potrebnih administracije, pa t.i. ročna rešitev ni več primerna. Takrat se moramo poslužiti avtomatizacije oz. skript, t.i. programov, ki opravijo to delo namesto nas.

Torej skripta, ki jo opisuje naloga, olajša naše delo in tako avtomatizira postopek osnovnega nameščanja programja, konfiguracijo nameščenih programov in sistemsko preverjanje celovitosti.

2. O odprtokodnih sistemih

Odprtokodni operacijski sistem GNU/Linux [3], ki ga je razvil Linus Torvalds s pomočjo razvijalcev s celotnega sveta, je razvit pod licenco GNU GPL [4] (General Public Licence). Pomeni, da je GNU/Linux razvit pod licenco odprte kode in je prosto dostopen širši javnosti, ali drugače, vzamemo poljubno distribucijo GNU/Linux-a in jo po svoji volji spreminjamo, ji dodajamo, odvzemamo programsko opremo, jo konfiguriramo ter jo delimo z drugimi. Bistvo GNU GPL licence je, da če dobimo GNU/Linux distribucijo in jo spremenimo tako ali drugače, jo moramo distribuirati pod istimi pogoji, kot smo jo dobili, torej z drugimi moramo deliti celotno kodo brez omejitev pri spreminjanju in jih seznaniti z zgornjim pogojem GNU GPL.

2.1 Zgodovina sistemov GNU/Linux

Začelo se je z operacijskim sistemom UNIX [5] imenovanim MINIX [6]. Linus Torvalds se je odločil, da razvije sistem, ki presega standarde operacijskega sistema MINIX, saj je bil slednji "*premajhen*" za njegove zahteve. Svoje delo okrog novega jedra za operacijski sistem, ki ga je poimenoval Linux [7], je začel leta 1991 z izidom verzije 0.02 in nato s trdim delom leta 1994 dosegel izdajo jedra verzije 1.0. Do danes je bilo razvitih že več kot petsto distribucij [8], katerih avtorji so podjetja oz. uporabniki. Nekatere so plačljive, spet druge niso. Vsem pa je skupno isto jedro. Uporabljeno jedro v diplomski nalogi, je verzije 2.6.26-2-686.

2.2 Zakaj GNU/Linux?

Operacijski sistem GNU/Linux smo izbrali zaradi svoje vsestranskosti in odprtosti. Ena od velikih prednosti GNU/Linux sistemov je manj zlonamerne kode kot so npr. virusi, saj jih za GNU/Linux obstaja nekje pod 1000 [9], medtem ko je za Microsoft Windows znanih preko 1.000.000 [10] različnih zlonamernih programov. Če smo še vedno zaskrbljeni glede varnosti, lahko namestimo protivirusni program. Prednost sistemov GNU/Linux je tudi njihovo delovanje na raznoliki strojni opremi. Poleg primerjalne opreme, na kateri lahko poganjamo operacijske sisteme Microsoft, teče GNU/Linux tudi na procesorjih, kot so npr. SPARC, MIPS,... Distribucije GNU/Linux je možno nameščati tako, da omogočajo dvojni zagon (angl. dual-boot) sistema. Ko namestimo operacijski sistem Windows, porabimo še ure in ure za namestitev vseh dodatnih programov, medtem ko pri GNU/Linux-u dobimo večino programov že nameščenih, ostale lahko namestimo pozneje, za povprečnega uporabnika [11] nameščenih že ob prvem zagonu novega sistema. GNU/Linux je odlična izbira kot operacijski sistem za strežnik in razvojno okolje [12]. Privzeto je integriran požarni zid, ki ga moramo le prilagoditi, za kar pa imamo na voljo kar precej različnih grafičnih in tekstovnih nastavitvenih programov. Če želimo uporabljati operacijski sistem za igranje računalniških igric, potem bomo najverjetneje uporabili Windows operacijske sisteme, če pa smo uporabnik, ki potrebuje različne strežniške storitve operacijskega sistema, se bomo posluževali uporabnosti GNU/Linux operacijskih sistememov.

3. O diplomski nalogi

3.1 Opis ideje in cilja

GNU/Linux je bil nekoč tabu, ki je za uporabo potreboval ogromno znanja, dela in vzdrževanja. Če primerjamo Microsoftove izdelke, so stvari podobne. V današnjem času so se GNU/Linux distribucije razvile že do takšne meje, da lahko brez težav konkurirajo plačljivim Microsoftovim izdelkom. Ogromna prednost GNU/Linux-a je torej že njegova cena, saj nam zanj ni potrebno plačati nič, oziroma imamo tudi profesionalne različice ponavadi za delovne postaje in strežnike, ki so plačljive. Prednosti plačljivih distribucij so boljša podpora, stabilnost, varnost in so ponavadi namenjena bolj podjetjem kot navadnim uporabnikom. Tako kot ostali operacijski sistemi imajo tudi GNU/ Linux distribucije možnost namestitve Desktop ali Server različice. Desktop ali namizna različica je namenjena navadnim uporabnikom. Vključuje uporabniške programe, ki bi jih lahko uporabnik uporabljal za večino opravil. Ima pa možnost namestitve še dodatnih programov. Strežniška (angl. server) različica pa je namenjena strežnikom. Njen cilj ni uporabniška udobnost in prijaznost, ampak čim večja varnost in stabilnost sistema.

Varnost [13] je pri strežnikih ključnega pomena. Zato morajo imeti distribucije, ki so namenjene strežnikom, dobro varnost že na začetku namestitve. Ker se strežniki glede na uporabo razlikujejo so ob namestitvi začetne nastavitve [14] posplošene. Tako moramo sistem nastaviti za naše potrebe. Cilj diplomske naloge je bil narediti primer skripte, ki bo sistem pregledala, namestila vso potrebno programsko opremo, nas obveščala o morebitnih spremembah sistema in bo dnevno izvajala varnostne preglede, ter ugotovitve zapisala v dnevnik.

3.2 Opis skripte

V podpoglavju 2.1 smo omenili, da obstaja na stotine različnih distribucij operacijskega sistema GNU/Linux. Mi uporabljamo distribucijo Debian GNU/Linux [15], zato smo to distribucijo uporabili za testiranje naše skripte diplomske naloge. Skripta je napisana v [16] ukazni lupini Bash, ki je ponavadi privzeta lupina GNU/Linux distribucij.

Skripta omogoča avtomatično administracijo sistema [17]. Avtomatizacija je v tem času najboljša rešitev, saj nam prihrani čas, medtem ko se mi posvetimo drugim problemom na sistemu. Skripto sestavlja pregled sistema za nadgradnjami in njihovo namestitev. Onemogoči branje lokalnih medijev CD-ROM iz datoteke sistemskih virov in nam vse korake sproti izpisuje. Nato pregleda nameščene programe, kot so: ssh-server, logwatch [18], apache2 [19] in če jih na sistemu ne najde, nam ponudi možnost njihove namestitve. Skripta izklopi storitve, ki zmanjšajo varnost sistema, kot so npr: finger [20], telnet [21], ftp [22]. Namesti in nastavi programe ssh, apache2 in logwatch, s katerimi izboljšamo varnost sistema. Nastavi se še privzeti uporabnik, ki bo dobival dnevna poročila sistemskih dogodkov v obliki elektronske pošte. Nato imamo možnost dodajanja naslovov IP in domenskih imen v datoteko `/etc/hosts.allow`, kajti skripta pod privzeto onemogoči dostop iz kateregakoli naslova IP ali domenskega imena s datoteko `/etc/hosts.deny`. Za celovitost sistema je v skripti poskrbljeno z iskanjem datotek SUID in SGID [23] in njihovo hranjenje na naslovu, ki ga določimo mi. V skripti je integrirana še ena skripta, ki se skopira na disk in se dnevno zaganja za preverjanje sistema. Ob vsakodnevnem pregledu se ustvari datoteka o spremljanju dejavnosti sistema, v katero se potem shranijo informacije delovanja skripte. Na koncu skripte se zažene še čiščenje začasnih lokalnih namestitvenih paketov.

4. Delovanje skripte po odsekih

4.1 Inicializacija

Spodnji stavek nastavi ukazno lupino. Podana je pot, kjer se na sistemu nahaja BASH ukazna lupina, ki je vmesnik, kamor vpisujemo sistemske ukaze, ki jih želimo izvajati.

```
#!/bin/bash
#
```

4.2 Nadgradnja sistema

Sledeči del varnostne skripte na zaslon izpiše sporočilo, da sistem poskuša posodobiti nameščeno programsko opremo. Ukaz `echo` služi za izpis na zaslon. V naslednji vrstici še imamo ukaz za pregled vseh možnih posodobitev, `apt-get update -qq`. Zastavice (angl. flags) `-qq` so pri ukazu zaradi izpisa, saj brez njih sistem izpiše vse podatke, na katerih strežnikih se nahajajo paketi za nadgradnjo, tako pa tega ne izpiše. Ko pregled konča, nadaljujemo z namestitvijo posodobitev, nato se izvrši ukaz `apt-get upgrade -qqy`, kjer `-qq` pomeni isto v prejšnjem primeru, dodana zastavica `y`, pa omogoča avtomatsko potrditev namestitve paketov.

```
echo "Pregledujem posodobitve sistema!"
apt-get update -qq
echo "Nameščam vse možne posodobitve!"
apt-get upgrade -qqy
```

4.3 Sistemski programski viri

Sistemski programski viri [24] so lokacije shramb programske opreme (repozitorij), ki jo je možno prenesti in namestiti na sistem.

V operacijskem sistemu Debian imamo datoteko `/etc/apt/source.list`, ki vsebuje naslove vseh repozitorijev, ki jih v podpoglavju 4.2 sistem pregleda z ukazom `apt-get update` in nato prenese z ukazom `apt-get upgrade`. Pod privzeto možnostjo je v tej datoteki dodan medij Debian CD-ROM za lokalno namestitev. Ker so na CD-ROM-u lahko programi že zastareli, oz. imamo na internetnih strežnikih novejše verzije določenih programov, z naslednjim segmentom ukazov skripta preveri, če datoteka `/source.list` vsebuje vključen Debian CD-ROM in jo izključi iz seznama virov za posodabljanje. To se izpiše tudi na zaslon. Naslednja vrstica podaja pripadajoči ukaz:

```
sed -i 's/deb cdrom/#&/g' /etc/apt/sources.list
```

Ta ukaz doda pred vrstico z imenom "cdrom" znak # in s tem "zakomentira" vrstico, kar pomeni, da je sistem več ne upošteva pri posodobitvah. Če je vrstica z začetkom imena "cdrom" že "zakomentirana", skripta samo izpiše sporočilo, da je vir že onemogočen.

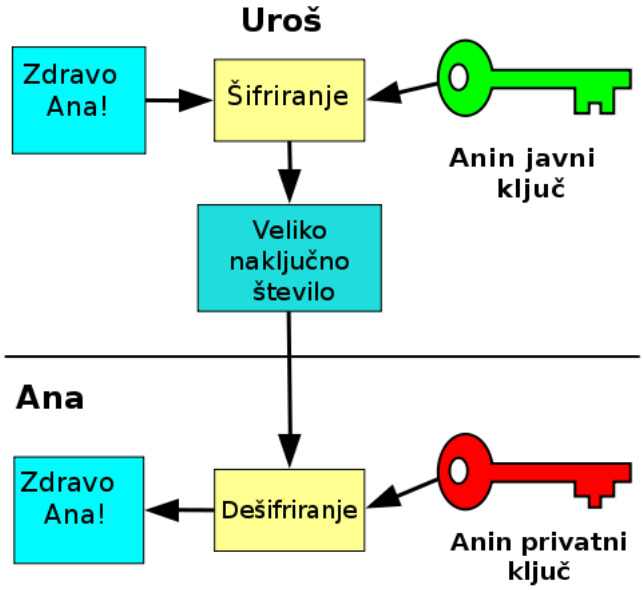
```
source_isk=‘cat /etc/apt/sources.list | grep "cdrom" |
grep -v "#" | wc -l‘

if [ $source_isk -eq 1 ]
   then
    echo "Brišem vnos optičnega pogona za vire nadgradnje sistema!"
    sed -i 's/deb cdrom/#&/g' /etc/apt/sources.list
   else
    echo "Viri optičnega pogona za nadgradnjo sistema so že
    izbrisani!"
fi
```

4.4 Varna lupina

Varna lupina (angl. Secure Shell, krajše SSH) [25] je mrežni protokol za varno povezavo med dvema mrežnima napravama. V sistemih GNU/Linux je to najboljši standarden način omrežnega terminalskega dostopa do računalnika. Nekoč se je za to uporabljal telnet, vendar je njegova največja pomanjkljivost ta, da pošilja vse podatke po omrežju nezaščitene oz. nekriptirane, poslane kot navadno besedilo. To pomeni, da če nekdo sledi naš promet, lahko hitro zasledi tudi geslo, ki se mu izpiše v tekstovni obliki. Protokol SSH uporablja šifriranje in povezavo z izmenjavo ključev. Služi lahko kot zamenjava za FTP (angl. File transfer protocol), protokol za prenos datotek preko omrežij , RCP (angl. Remote copy) kopiranje preko omrežja, s SCP (Secure copy) [26] varno kopiranje prek omrežja. Prav tako dovoljuje tuneliranje [27] in posredovanje prometa, npr. povezave X11, ki omogočajo poganjanje grafičnih programov na oddaljenem sistemu.

4.4.1 Izmenjava šifrirnih ključev pri varni lupini

Za izmenjavo šifrirnih ključev potrebuje vsak uporabnik dva šifrirna ključa [28]. 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, dešifrirana pa so lahko le s pripadajočim privatnim ključem. Povezava med ključema je matematična, vendar iz javnega ključa privatnega ključa ne moremo dešifrirati.



Slika 4.1: Postopek generiranja šifrirnih ključev pri SSH-ju

Večina distribucij GNU/Linux ima odjemalca (angl. client) SSH že nameščenega na sistem. Če želimo do sistema dostopati z oddaljenega računalnika, potrebujemo tudi strežnik SSH. Skripta preveri, če je storitev (angl. daemon) z imenom "sshd" že zagnan. Z iskanjem po mapi `/etc/init.d`, kjer se nahajajo vse storitve, shranimo v spremenljivko z imenom `iskanje` rezultat poizvedbe. Ob nobenem zadetku se na zaslon ne izpiše nič, zato dodamo še dodaten ukaz `wc -l`. Ta prešteje število izpisanih vrstic, ki jih nato lahko preverjamo s pogojnim stavkom `if` [29]. Če na sistemu ni zagnana storitev "ssh", vrne ukaz `wc -l` vrednost 0. Ta pogojnemu stavku pove, naj izpiše `"Program ssh-server je nameščen!"`, sicer pa na zaslonu dobimo izpis za možnost namestitve programa.

```
iskanje=‘find /etc/init.d -name "ssh" | wc -l‘

if [ $iskanje -gt 0 ]
   then
    echo "Program ssh-server je nameščen!"
   else
    echo "Program ssh-server ni nameščen, ga namestim?
    (D za DA, N za NE)"
    read vnos1
     if [ $vnos1 == "d" || $vnos1 == "D"]
        then
         echo "Nameščam ssh-server...!"
         apt-get -qqy install ssh
     elif [ $vnos1 == "n" || $vnos1 == "N" ]
        then
         echo "Izhod...!"
        else
         echo "Nedovoljena izbira!"
     fi
   echo "Program ssh-server je nameščen!"
fi
```

4.5 Nastavitev strežnika SSH

Varna lupina SSH je sama po sebi precej varna, saj uporablja šifriran prenos podatkov po omrežju. Ima pa kot privzeto možnost, da omogoča prijavo na oddaljeni računalnik kot sistemski administrator (angl. root). 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 lahko nato že popoln oddaljen nadzor nad računalnikom. S spreminjanjem vrednosti 'PermitRootLogin' v datoteki `/etc/ssh/sshd_config` iz "yes" na "no" spodnji del skripte to možnost prijavljanja na sistem kot uporabnik root onemogoči. Na koncu moramo storitev `ssh` še ponovno zagnati, da uveljavimo morebitne spremenjene nastavitve.

```
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
   else
    echo
fi
```

4.6 Logwatch

Logwatch je sistemski nadzornik, ki spremlja vse spremembe na sistemu. Ob določeni uri, ki jo nastavimo v konfiguracijski datoteki, lahko pošlje uporabniku elektronsko sporočilo v obliki poročila. Skripta najprej poišče, če je na sistemu konfiguracijska datoteka `logwatch.conf`. Če datoteka obstaja na sistemu, skripta izpiše, da je program logwatch že nameščen. V primeru, ko vrnjena poizvedba datoteke ne najde, skripta ponudi možnost namestitve programa z vnosom črke "D" za potrditev in črko "N" za odklonitev namestitve. Ob vnosu katerega koli drugega znaka ali številke, skripta vrne izpis `"Nedovoljena izbira!"`, zanko prekinemo, program pa se nadaljuje.

```
isk_log=‘find / -name "logwatch.conf" | wc -l‘

if [ $isk_log -gt 0 ]
   then
    echo "Program logwatch je nameščen!"
   else
    echo "Program logwatch ni nameščen, ga namestim?
    (D za DA, N za NE)"
    read vnos2
     if [ $vnos2 == "d" || $vnos2 == "D" ]
        then
         echo "Nameščam logwatch...!"
         apt-get -qqy install  logwatch
     elif [ $vnos2 == "n" || $vnos2 == "N" ]
        then
         echo "Izhod...!"
        else
         echo "Nedovoljena izbira!"
     fi
   echo "Program logwatch je nameščen!"
fi
```

4.7 Nastavitev storitve Logwatch

Logwatch lahko pošilja dnevna, tedenska ali mesečna poročila stanja opazovanega sistema. Skripta tukaj ponudi možnost spremembe elektronskega naslova za pošiljanje poročil. Najprej poiščemo datoteko `logwatch.conf`. Dobimo dva iskalna zadetka, ki ju omejimo z ukazom `grep "default"`. Skripta nato preišče datoteko, izključi vse nepotrebne vrstice in spremeni v vrstici, ki se začne z `Output`, besedo `stdout` v `mail`. To spremeni obnašanje programa `logwatch`, da ne izpisuje poročil na standardni izhod (zaslon), ampak jih pošilja po elektronski pošti. Nastavitev shranimo v spremenljivko `mailto` pri uporabniku, ki je zapisan v datoteki `logwatch.conf` kot privzeti prejemnik poročil. Nato uporabnika izpišemo na zaslon, polek tega se izpiše še možnost spreminjanja tega uporabnika. Če uporabnika želimo spremeniti, vnesemo črko ”d“, v nasprotnem primeru pa ”n“. Če izberemo možnost ”d“, dobimo na zaslon izpis za vnos uporabnikovega naslova eletronske pošte, kamor želimo, da se poročila pošiljajo. Skripta nato zamenja trenutnega uporabnika z vnešenim. Slednjega potem tudi izpiše na zaslon.

```
isk_log=‘find / -name "logwatch.conf" | grep "default"‘
mailto=‘cat $isk_log | grep "MailTo" | awk -F " " ’{print $3}’‘

cat $isk_log | grep "Output" | grep -v "#" |
sed -i ’s/stdout/mail/g’ $isk_log
echo "Trenutno je kot prejemnik poročil programa Logwatch nastavljen
uporabnik: "$mailto

echo "Če želite trenutnega uporabnika spremeniti oz.
če želite dodati vaš e-poštni naslov pritisnite črko d,
drugače pa črko n!"
read crka
```

```
if [ $crka == "d" ] || [ $crka == "D" ]
   then
    echo "Vnesite željenega uporabnika oz. poštni naslov kamor
    želite prejemati poročila programa Logwatch:"
    read uporabnik
    sed -i 's/'$mailto'/'$uporabnik'/g' $isk_log
elif [ $crka == "n" ] || [ $crka == "N" ]
   then
    echo "Nastavljen je uporabnik: "$mailto
   else
    echo "Nedovoljena izbira...Vnesite črki d ali n!"
fi
```

4.8 Apache2

Apache je spletni strežnk. Večina distribucij ima na sistemu nameščen ta program. Namenjen je tako podjetjem kot tudi domačim uporabnikom. Njegova funkcionalnost in uporabnost se še zveča z dodajanjem modulov.

V sledečem delu programske kode skripta preveri, če je apache2 nameščen na sistemu. Če je nameščen nam izpiše na zaslon, v obratnem primeru dobimo možnost namestitve po opisanem postopku iz podpoglavja 4.4. Edini dodatek v tem delu skripte je ukaz

```
update-rc.d apache2 defaults
```

Prikazan ukaz po namestitvi paketa `apache2`, nastavi spletni strežnik na privzete nastavitve.

```
isk_apa=‘find /etc/init.d -name "apache2" | wc -l‘

if [ $isk_apa -gt 0 ]
   then
    echo "Program apache2 je nameščen!"
   else
    echo "Program apache2 ni nameščen, ga namestim?
    (D za DA, N za NE)"
    read vnos3

     if [ $vnos3 == "d" || $vnos3 == "D"]
        then
         echo "Nameščam apache...!"
         apt-get -qqy install apache2
         update-rc.d apache2 defaults
         echo "Apache servis nastavljen na privzeto vrednost!"
     elif [ $vnos3 == "n" || $vnos3 == "N" ]
        then
         echo "\n"
        else
         echo "Nedovoljena izbira!"
     fi
 echo "Program apache2 je nameščen!"
fi
```

4.9 Konfiguracija Apache2

Spletni strežnik Apache ima nastavljene privzete varnostne nastavitve dokaj dobro, vendar so lahko še varnejše. Odvisno je le, kaj in koliko od privzetih možnosti smo pripravljeni "žrtvovati" za bolj varen spletni strežnik. Privzete nastavitve podajajo na strežnikovi spletni strani podatke o operacijskem sistemu, nameščeni verziji spletnega strežnika, podatkovni bazi mysql, ali je omogočen php in še marsikaj. Iz varnostnih razlogov skripta te informacije izklopi. Čeprav so ti podatki lahko koristni za administratorja so koristni tudi za potencialnega napadalca na sistem. Skripta najprej poišče pot, kjer se nahaja datoteka `apache.conf` in z dodanim ukazom `wc -l` shrani poizvedbo. Slednjo nato preverimo, če obstaja. Ob obstoju v iskano datoteko dodamo besedilo `ServerSignature Off`. Prav tako po enakem postopku spremenimo vrednost za "Timeout", ki določa, kako dolgo pusti strežnik odrto trenutno sejo ob neaktivnosti. Ker ne želimo, da bi bilo mogoče po kakšni poti priti do dostopa preko neaktivnosti prijavljenega uporabnika, smo čas neaktivne seje zmanjšali iz 300 na 45 sekund.

```
isk_apache=‘find / -name "apache2.conf"‘
isk_sig=‘cat $isk_apache | grep ServerSignature | wc -l‘

if [ $isk_sig -eq 1 ]
   then
    echo "Dodajam vnos podatkov v datoteko apache.conf...!"
    sed -i ’$a\ServerSignature Off’ $isk_apache
   else
    echo "Vnos ’ServerSignature Off’ je že dodan v datoteki
    apache.conf!"
fi
```

```
isk_time=‘cat $isk_apache | grep "Timeout" | grep -v "#"
| grep -v "KeepAliveTimeout" | awk -F " " ’ {print $2}’‘

if [ $isk_time -gt 50 ]
   then
    echo "Spreminjam vrednost podatkov v datoteki apache.conf...!"
    cat $isk_apache | grep "Timeout" | sed -i ’s/300/45/g’ $isk_apache
   else
    echo "Vrednost TimeOut v datoteki apache.conf je že nastavljena!"
fi
```

4.10 Finger

Sistemska storitev z imenom `finger` nam podatke o trenutno prijavljenih uporabnikih na sistemu. Ti podatki so lahko zelo pomembni za napadalca na oddaljenem sistemu, zato to storitev izklopimo. Najprej skripta pregleda zagnane procese in če je `finger` med njimi, ga prekine. Skrbniška storitev `Inetd` [30] je namenjena delovanju nekaterih lahkih internetnih storitev, njihovo konfiguracijo in zagon. Ponavadi se na sistemih UNIX nahaja v mapi `/usr/sbin/inetd`. Skripta tako uporabi ukaz:

```
/usr/sbin/update-inetd --disable finger
```

in tako onemogoči nadaljni zagon servisa `finger`. Sledi le še izpis, da je bila storitev `finger` izklopljena.

```
isk_fin=‘ps aux | grep finger | wc -l‘

if [ $isk_fin -gt 1 ]
   then
    killall finger
    /usr/sbin/update-inetd --disable finger
    echo "Finger storitev izklopljen!"
fi
```

Primer uporabe in izpisa podatkov z ukazom finger:

```
Login    Name    Tty     Idle     Login Time     Office     Office Phone
damjanc          tty7    2:47     Nov 27 09:13   (:0)
damjanc          pts/0   11       Nov 27 09:14   (:0.0)
damjanc          pts/1   1:26     Nov 27 10:32   (:0.0)
damjanc          pts/2            Nov 27 11:49   (:0.0)
```

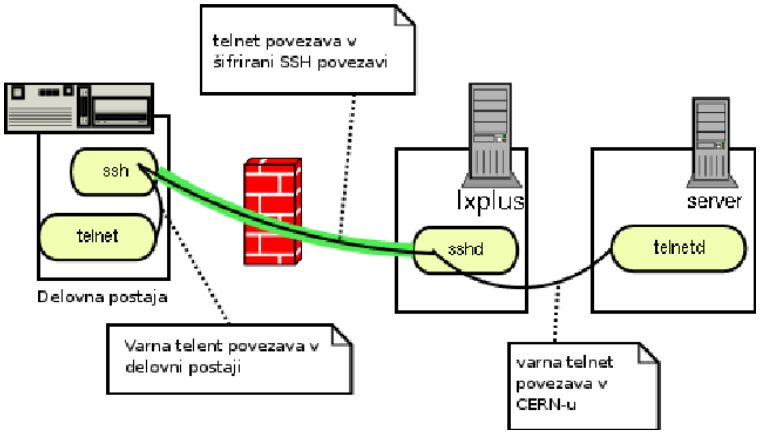
4.11 Telnet

`Telnet` se je uporabljal, oz. ponekod se še vedno uporablja, za delo na oddaljenih sistemih. Njegovo delovanje je podobno, kot pri omenjeni varni lupini, v podpoglavju 4.4. Njegova velika slabost je, da so vsi poslani podatki nešifrirani. Tako je lahko vsak, ki ima dostop do podatkovnih paketov v omrežju, bere podatke kot navadno tekstovno besedilo. Razvit je bil že leta 1969, od 1995 pa ga večinoma zamenjuje SSH.

Spodnji del skripte ima enako delovanje kot skripta za onemogočanje servisa `finger` v podpoglavju 4.10 z razliko, da tukaj onemogočamo servis `telnet`.

```
isk_tel=‘ps aux | grep telnet | wc -l‘

if [ $isk_tel -gt 1 ]
   then
    killall telnet
    /usr/sbin/update-inetd --disable telnet
    echo "Telnet servis izklopljen!"
fi
```

Na sliki 4.2 je prikazana povezava med dvema spletno povezanima lokalnima omrežjema. V lokalnem omrežju, ki je varno, lahko uporabljamo `telnet`. Ko se povezujemo prek spletne povezave do drugega lokalnega omrežja, uporabimo varno `SSH` povezavo.



Slika 4.2: Primer šifriranega dostopa preko telnet-a

4.12 Protokol za prenos datotek

`FTP` ali File Transfer Protocol je standradni mrežni protokol za izmenjavo in manipulacijo datotek preko omrežij. Pri večini distribucij GNU/Linux je `FTP` prisoten kot storitev. `FTP` se je najbolj uporabljal za prenos večjih datotek in pa za anonimno povezavo na strežnik `FTP`. Zaradi možnega anonimnega prijavljanja na strežnik in posledičnega anonimnega nalaganja datotek na strežnik, je storitev `FTP` tudi potrebno izklopiti. Skripta tukaj vključuje koraka iz podpoglavij 4.4 in 4.10. Na zaslon dobimo izpis možnosti, če želimo izklopiti `FTP` servis ali ne. Tukaj ni avtomatskega izklopa, saj se `FTP` ponekod uporablja še danes, čeprav ga zamenjuje `SCP`.

```
isk_ftp=‘ps aux | grep ftp | wc -l‘
if [ $isk_ftp -gt 1 ]
   then
   echo "Želite izklopiti ftp storitev? (D za DA, N za NE)?"
   read vnos4

   if [ $vnos4 == "d" || $vnos4 == "D" ]
      then
      killall ftp
      /usr/sbin/update-inetd --disable ftp
      echo "Ftp storitev izklopljen!"
   elif [ $vnos4 == "n" || $vnos4 == "N" ]
      then
       echo "Ftp storitev je vklopljen!"
   fi
 else
  echo "Nedovoljena izbira!"
fi
```

4.13 Datoteka *hosts.deny*

Datoteko `/etc/hosts.deny` uporabljamo za zavrnitev določenih povezav z določenih omrežij. To možnost imenujemo angl. TCP wrapper. Skripta prebere vse vrstice v datoteki, v katerih prvi znak ni enak znaku "#". Če je število vrstic manjše od ena, potem dodamo besedilo "ALL:ALL", kar pomeni, da vsem storitvam onemogočimo dostop s katerekoli domene do računalnika. Če datoteka vsebuje več kot eno vrstico brez znaka "#", izpišemo sporočilo "Vrednosti v datoteki že nastavljene".

```
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 "Vrednosti v datoteki hosts.deny so že nastavljene!"
 fi
```

4.14 Datoteka *hosts.allow*

Tako kot opisana datoteka `/etc/hosts.deny` v podpoglavju 4.13 onemogoča dostop za določene domenske naslove, jih datoteka `/etc/hosts.allow` omogoča. Skripta doda v datoteko domenska imena ali naslove IP, ter s tem omogoča njihov dostop do določenih storitev z določenih omrežij. Na zaslon izpiše navodilo kako vnesti podatke, če želimo dostopati do sistema. Konec vnašanja podatkov označimo s pritiskom na tipko, ki je ”q“ (ang. quit), sledi potrditev s tipko ”enter“.

```
echo " Vnesite IP naslov/e ali domenska imena za računalnike,
katerim želite dovoliti dostop do tega strežnika, ter pritisnite
tipko q za konec ali za preskok te možnosti!\n npr:
test.domain.com ali 123.456.78.90
(popravljanje teh možnosti je možno v datoteki /etc/hosts.allow)"
read hostname

while [ $hostname != "q" ]
   do
    echo "ALL:" $hostname >> /etc/hosts.allow
    read hostname
done
```

4.15 Super user ID in Super Group ID

SUID oz. SETUID je okrajšava za "Set User ID", SGID oz. SETGID pa okrajšava za "Set Group ID". Prvi ukaz omogoča uporabnikom poganjanje datotek tipa S v privilegiranem načinu. Drugi ukaz omogoča isto funkcionalnost skupini uporabnikov. Če določeno datoteko, označimo z bitom S so to lahko velika varnostna tveganja. Zato je najboljše imeti na sistemu čim manj takšnih datotek. Tiste pa, ki že morajo biti, naj so pod skrbnim nadzorom. Iz tega vidika je tudi v skripti dodan del:

```
echo "Vnesite pot kamori želite da se shranijo SUID in SGID datoteke,
ki so potrebne za nadaljno primerjavo za neokrnjenost sistema: "
echo "Trenutna pot je:"
pwd
read suid_pot
echo "Iščejo se SUID in SGID datoteke........."

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
echo "SUID in SGID datoteke shranjene!"
```

ki zahteva vnos poti, kamor želimo shraniti seznam datotek označenih z bitom S. Nato skripta poišče vse datoteke, ki so v lasti uporabnika root `-perm -4000`, ter datoteke, ki so v lasti uporabniške skupine root `-perm -2000`. Programsko zastavico (flag) `-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 shrani na "read-only", to je samo bralni medij, kot je CD-ROM ali disketa. Sledi še izpis, da so datoteke shranjene.

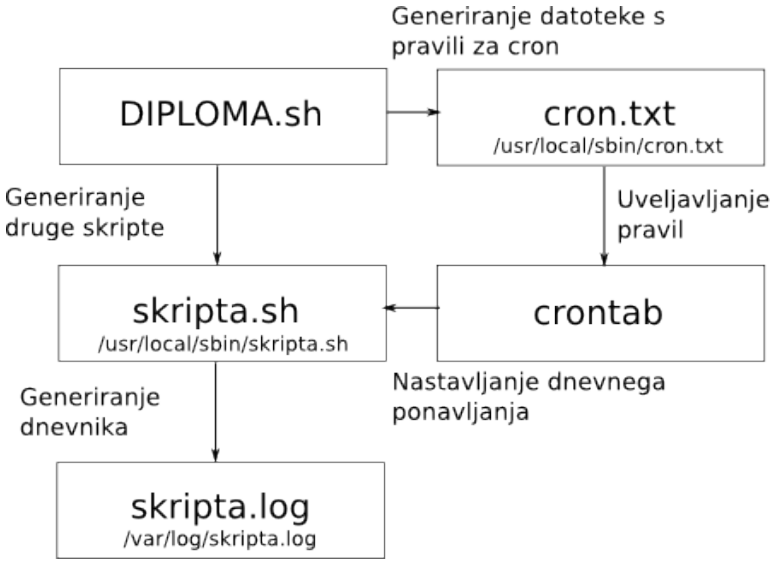
4.16 Vsakodnevni pregled in dnevnik

Ko je večji del funkcionalnosti sistema že nastavljen, skripta s pomočjo ukaza `touch` generira v mapi `/etc/local/sbin` novo skripto z imenom `skripta.sh`. V kreirani skripti so ukazi, predstavljeni skozi podpoglavja. Najprej v dnevniški mapi `/var/log` kreiramo datoteko `skripta.log`, v katero shranjujemo vse izpise. `Skripta.log` je dnevnik, kamor vnašamo datum, dobljen z ukazom `date`. Drugi vnos se navezuje na nameščanje programske opreme. Nato onemogočimo sistemski storitvi `finger` in `telnet`. Za storitev `FTP` izpišemo samo trenutno stanje na sistemu, t.j. če je omogočen ali ne. Nato med seboj primerjamo `s_datoteke.txt`, ki smo jih shranili v glavni skripti. Slednjo ugotovitev le še shranimo v datoteko `skripta.log`. Za konec nastavimo pravice datoteke `skripta.sh` tako, da jo lahko izvajamo oz. poganjamo.

```
echo "Ustvarjam skripto za vsakodnevni pregled sistema......"

touch /usr/local/sbin/skripta.sh
echo 'touch /var/log/skripta.log
date >> /var/log/skripta.log
echo "Pregledujem sistem za potencialnimi nadgradnjami!"
 >> /var/log/skripta.log
apt-get update -qq
echo "Nameščam vse možne nadgradnje!" >> /var/log/skripta.log
apt-get upgrade -qqy

isk_fin=‘ps aux | grep finger | wc -l‘

if [ $isk_fin -gt 1 ]
   then
    killall finger
    echo "Finger storitev izklopljen!" >> /var/log/skripta.log
   else
    /usr/sbin/update-inetd --disable finger
    echo "Finger storitev izklopljen!" >> /var/log/skripta.log
```

```
fi

isk_tel=‘ps aux | grep telnet | wc -l‘

if [ $isk_tel -gt 1 ]
   then
    killall telnet
    echo "Telnet storitev izklopljen!" >> /var/log/skripta.log
   else
    /usr/sbin/update-inetd --disable telnet
    echo "Telnet storitev izklopljen!" >> /var/log/skripta.log
fi

isk_ftp=‘ps aux | grep ftp | wc -l‘

if [ $isk_ftp -gt 1 ]
   then
    echo "Ftp storitev vklopljen!" >> /var/log/skripta.log
   else
    echo "Ftp storitev izklopljen!" >> /var/log/skripta.log
fi

s_dat=‘find / -name "s_datoteke.txt"‘
s_dat2=‘find / -name "s_datoteke2.txt"‘

razlika=‘diff $s_dat $s_dat2 | wc -l‘
```

```
if [ $razlika -eq 0 ]
   then
    echo "Sistemske datoteke so enake!" >> /var/log/skripta.log
   else
    echo "Sistemske datoteke niso enake!!!" >>
    /var/log/skripta.log
fi

apt-get clean

echo "**************************************************
**************************** >> /var/log/skripta.log' >>
/usr/local/sbin/skripta.sh

chmod +x /usr/local/sbin/skripta.sh
```



Slika 4.3: Diagram poteka ustvarjanja datotek

4.17 Časovni strežniški program za zagon ukazov Cron

Cron [31] je sistemsko integrirani strežnik časovnih opravil. Glavna skripta ustvari datoteko z imenom `cron.txt`, v kateri so vpisana pravila, ki določajo, kdaj se naj generirana skripta poganja avtomatsko. Nato skripta z ukazom `crontab` uvozi pravila [32] zapisana v omenjeni datoteki, v programskem strežniku cron in potem še pot do datoteke `cron.txt`. Celotno dogajanje sproti nato izpisujemo še na zaslon.

```
touch /usr/local/sbin/cron.txt
echo "0 3 * * * sh /usr/local/sbin/skripta.sh" >>
/usr/local/sbin/cron.txt
crontab /usr/local/sbin/cron.txt
echo "Dnevna ponovitev skripte je nastavljena!"
```

4.18 Čiščenje sistema

Na koncu skripte poženemo ukaz `apt-get clean`, ki počisti začasne namestitvene pakete, shranjene na računalniku.

5. Zagon in izpis skripte

Primer zagona skripte:

```
debian: # ./DIPLOMA.sh
Pregledujem sistem za potencialnimi nadgradnjami!
Nameščam vse možne nadgradnje za sistem!
Brišem vnos optičnega pogona za vire nadgradnje sistema!
Program ssh-server je nameščen!
Program logwatch ni nameščen, ga namestim?  (D za DA, N za NE)
d
Nameščam logwatch...!
Selecting previously deselected package libdatemanipperl.
(Reading database ...  19212 files and directories currently
installed.)
Unpacking libdatemanipperl (from .../libdatemanip-perl_5.54-1_all
.deb) ...
Selecting previously deselected package logwatch.
Unpacking logwatch (from .../logwatch_7.3.6.cvs200807022_all.deb) ...
Processing triggers for mandb ...
Setting up libdate-manip-perl (5.54-1) ...
Setting up logwatch (7.3.6.cvs20080702-2) ...
Program logwatch je nameščen!
Program apache2 ni nameščen, ga namestim?  (D za DA, N za NE)
d
Nameščam apache...!
Preconfiguring packages ...
Selecting previously deselected package openssl.
(Reading database ...  19533 files and directories currently
installed.)
Unpacking openssl (from .../openssl_0.9.8g-15+lenny5_i386.deb) ...
Selecting previously deselected package openssl-blacklist.
Unpacking openssl-blacklist (from .../openssl-blacklist_0.4.2_all.deb)
...
```

```
Selecting previously deselected package libapr1.
Unpacking libapr1 (from .../libapr1_1.2.12-5+lenny1_i386.deb) ...
Selecting previously deselected package libexpat1.
Unpacking libexpat1 (from .../libexpat1_2.0.1-4+lenny1_i386.deb) ...
Selecting previously deselected package mysql-common.
Unpacking mysql-common (from .../mysqlcommon_5.0.51a-24+lenny2_all
.deb) ...
Selecting previously deselected package libmysqlclient15off.
Unpacking libmysqlclient15off (from .../libmysqlclient15off_5.0.51a-24
+lenny2_i386.deb) ...
Selecting previously deselected package libpq5.
Unpacking libpq5 (from .../libpq5_8.3.8-0lenny1_i386.deb) ...
Selecting previously deselected package libaprutil1.
Unpacking libaprutil1 (from .../libaprutil1_1.2.12+dfsg-8+lenny4_i386
.deb) ...
Selecting previously deselected package apache2-utils.
Unpacking apache2-utils (from .../apache2-utils_2.2.9-10+lenny6_i386
.deb) ...
Selecting previously deselected package apache2.2-common.
Unpacking apache2.2-common (from .../apache2.2-common_2.2.9-10+lenny6
_i386.deb) ...
Selecting previously deselected package apache2-mpm-worker.
Unpacking apache2-mpm-worker (from .../apache2-mpm-worker_2.2.9-10+
lenny6_i386.deb) ...
Selecting previously deselected package apache2.
Unpacking apache2 (from .../apache2_2.2.9-10+lenny6_all.deb) ...
Selecting previously deselected package ssl-cert.
Unpacking ssl-cert (from .../ssl-cert_1.0.23_all.deb) ...
Processing triggers for man-db ...
Setting up openssl (0.9.8g-15+lenny5) ...
Setting up openssl-blacklist (0.4.2) ...
Setting up libapr1 (1.2.12-5+lenny1) ...
Setting up libexpat1 (2.0.1-4+lenny1) ...
Setting up mysql-common (5.0.51a-24+lenny2) ...
Setting up libmysqlclient15off (5.0.51a-24+lenny2) ...
Setting up libpq5 (8.3.8-0lenny1) ...
Setting up libaprutil1 (1.2.12+dfsg-8+lenny4) ...
Setting up apache2-utils (2.2.9-10+lenny6) ...
Setting up apache2.2-common (2.2.9-10+lenny6) ...
```

```
Enabling site default.
Enabling module alias.
Enabling module autoindex.
Enabling module dir.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module status.
Enabling module auth_basic.
Enabling module deflate.
Enabling module authz_default.
Enabling module authz_user.
Enabling module authz_groupfile.
Enabling module authn_file.
Enabling module authz_host.
Setting up apache2-mpm-worker (2.2.9-10+lenny6) ...
Starting web server:  apache2.
Setting up apache2 (2.2.9-10+lenny6) ...
Setting up ssl-cert (1.0.23) ...  System startup links for
/etc/init.d/apache2 already exist.
Apache servis nastavljen na privzeto vrednost
Program apache2 je nameščen
Restarting OpenBSD Secure Shell server:  sshd.
Dodajam vnos podatkov v datoteko apache.conf...
Spreminjam vrednost podatkov v datoteki apache.conf...
Vrednosti v datoteki hosts.deny so že nastavljene
Vnesite IP naslove ali domenska imena za računalnike katerim
želite dovoliti dostop do tega strežnika
ter pritisnite tipko q za konec ali za preskok te možnosti npr
test.domain.com ali 123.456.78.90
(popravljanje teh možnosti je možno v datoteki /etc/hosts.allow)
marmor.uni-mb.si
93.103.167.112
q
Trenutno je kot prejemnik poročil programa Logwatch nastavljen
uporabnik  root
Če želite trenutnega uporabnika spremeniti oz.  če želite dodati vaš
e-poštni naslov pritisnite
črko d, drugače pa črko n
d
```

```
Vnesite željenega uporabnika oz.  poštni naslov, kamor želite
prejemati poročila programa Logwatch
test
Vnesite pot, kamor želite, da se shranijo SUID-bit in SGID datoteke,
ki so potrebne za nadaljno primerjavo neokrnjenosti sistema:
Trenutna pot je
/root
/root
Iščejo se SUID in SGID datoteke.........
find:  proc/3082/task/3082/fd/5  No such file or directory
find:  proc/3082/task/3082/fd/5  No such file or directory
find:  proc/3082/task/3082/fdinfo/5  No such file or directory
find:  proc/3082/task/3082/fdinfo/5  No such file or directory
find:  proc/3082/fd/5  No such file or directory
find:  proc/3082/fd/5  No such file or directory
find:  proc/3082/fdinfo/5  No such file or directory
find:  proc/3082/fdinfo/5  No such file or directory
find:  proc/3083/task/3083/fd/5  No such file or directory
find:  proc/3083/task/3083/fd/5  No such file or directory
find:  proc/3083/task/3083/fdinfo/5  No such file or directory
find:  proc/3083/task/3083/fdinfo/5  No such file or directory
find:  proc/3083/fd/5  No such file or directory
find:  proc/3083/fd/5  No such file or directory
find:  proc/3083/fdinfo/5  No such file or directory
find:  proc/3083/fdinfo/5  No such file or directory
SUID in SGID datoteke shranjene
Dnevna ponovitev skripte je nastavljena!
```

Primer izpisa datoteke `skripta.log`:

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

6. Zaključek

Varnost računalniških sistemov je dandanes ključnega pomena. Povečanje varnosti zahteva izključevanje določenih sistemskih storitev oziroma nastavitev teh. Osnovna funkcija varnosti je ohranjanje integritete podatkov pred nepooblaščenimi osebami ali organizacijami. Da poskrbimo za varnost na sistemu, se torej poslužujemo določenih metod za ohranjanje varnosti. Poznamo fizične zaščite, kjer je fizični dostop do strežnika omejen ali celo onemogočen. Druga zaščita pa onemogoča napade s svetovnega spleta preko omrežnih povezav. In s slednjo se želimo zaščititi z določenimi algoritmi. Ali sami nastavimo sistem za čim boljšo varnost ali pa se poslužujemo določenih programov ali skript, ki to metodo olajšajo. Avtomatizacija oz. samodejno nastavljanje je najboljši način za nastavitev več računalnikov ali celotnih računalniških infrastruktur.

Sistemska administracija [33] je prisotna tako ob ročni nastavitvi kot tudi pri samodejni nastavitvi sistemskih strežnikov. Vedno lahko pride do kakšnih nepravilnosti pri delovanju, zato je administratorska navzočnost nujna. Dnevniški zapisi delovanja so zato pomembni, ob napakah delovanja ali spremembah nastavitev sistema, za odpravo napake oz. izvora prekinitve delovanja. V diplomski nalogi opisujemo skripto Bash namenjeno avtomatizaciji namestitve in nastavitve programja, ter sistemskih storitev.

Iz izpisa delovanja skripte, opisane v nalogi, so razvidne možnosti njene nadgradnje. Določenih izpisov ne potrebujemo oziroma so odveč. Izpis na zaslonu bi bilo možno omejiti, da bi bil bolj pregleden in izčrpen. Ob namestitvi dodatnega programja nam vsakodnevni sistemski pregled javlja spremenjeno integriteto operacijskega sistema, kar lahko odpravimo le s ponovnim zagonom osnovne skripte. To so tudi smernice za nadalnje delo.

Literatura

[1] "How many desktop Linux users?," *http://blogs.computerworld.com/how_many_desktop_linux_users (dne 10.12.2009).*

[2] "Linux Advantages," *http://linux.about.com/cs/linux101/a/linux_2.htm (dne 10.12.2009).*

[3] "GNU Linux," *http://en.wikipedia.org/wiki/Linux (dne 7.12.2009).*

[4] "GNU General Public Licence," *http://www.gnu.org/licenses/gpl.html*, 2007.

[5] "Unix," *http://en.wikipedia.org/wiki/Unix (dne 8.12.2009).*

[6] "Minix," *http://en.wikipedia.org/wiki/Minix (dne 8.12.2009).*

[7] "What is Linux," *http://www.linux.org/info/index.html (dne 17.11.2009)*, 2007.

[8] "LWN Linux distributions list," *http://lwn.net/Distributions/ (dne 26.11.2009).*

[9] "Linux malware," *http://en.wikipedia.org/wiki/Linux_malware (dne 25.11.2009)*, 2009.

[10] DaBoss, "Number of Viruses," *http://www.cknow.com/cms/vtutor/number-of-viruses.html (dne 17.11.2009)*, 2009.

[11] "Povprečen uporabnik," *http://www.lubica.net/bigwhale/blog/?p=856 (dne 10.12.2009).*

[12] J. Jack Hackett and D. Gunter, *Using Linux, Second edition.* Que Corporation, 1996.

[13] T. B. Ivan Verdonik, *Hekerski vdori in zaščita.* Založba Pasadena, 2005.

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

[15] "Debian GNU/Linux," *http://www.debian.org/ (dne 9.12.2009).*

[16] "BASH," *http://en.wikipedia.org/wiki/Bash (dne 9.12.2009).*

[17] J. F.-S. Peña, *Securing Debian Manual v3.6.* 2006.

[18] "Logwatch Configuration in Debian," *http://www.debianhelp.co.uk/logwatch1.htm (dne 30.11.2009).*

[19] "Apache HTTP server," *http://en.wikipedia.org/wiki/Apache_HTTP_Server (dne 26.11.2009).*

[20] "Finger protocol," *http://en.wikipedia.org/wiki/Finger_protocol (dne 27.11.2009).*

[21] "Telnet," *http://en.wikipedia.org/wiki/Telnet (dne 27.11.2009).*

[22] "File Transfer Protocol," *http://en.wikipedia.org/wiki/Ftp (dne 7.12.2009).*

[23] "setuid & setgid," *http://en.wikipedia.org/wiki/Setuid (dne 2.12.2009).*

[24] "Software repository," *http://en.wikipedia.org/wiki/Software_repository (dne 10.12.2009).*

[25] "Secure Shell-SSH," *http://en.wikipedia.org/wiki/Secure_Shell (dne 17.11.2009)*, 2009.

[26] "SCP: an FTP Alternative," *http://rimuhosting.com/howto/scp.jsp (dne 30.11.2009).*

[27] "Tuneliranje," *http://mreze.layerx.com/s0601000.html (dne 10.12.2009).*

[28] "Public-key cryptography," *http://en.wikipedia.org/wiki/Public-key_cryptography (dne 20.11.2009)*, 2009.

[29] P. Anderson, *Kako v Linuxu.* Založba Pasadena (LUGOS), 2002.

[30] "Inetd," *http://en.wikipedia.org/wiki/Inetd (dne 26.11.2009).*

[31] "How do I add jobs to cron under Linux and UNIX operating system?," *http://www.cyberciti.biz/faq/howdoiaddjobstocronunderlinuxorunixoses/ (dne 30.11.2009).*

[32] S. Kemp, "Command scheduling with cron," *http://www.debian-administration.org/articles/56 (dne 30.11.2009).*

[33] "System administrator," *http://en.wikipedia.org/wiki/System_ administrator (dne 10.12.2009).*