**FORENZIČNA RAZISKAVA DISKA**

Ko smo prejeli disk v roke(ali sliko diska) ga lahko začnemo forenzično raziskovat. Prvi korak je priklopitev diska v načinu samo za branje. To naredimo z ukazom

mount –r *medij direktorij*

Zastavica –r pomeni da priklopimo disk v read-only načinu(lahko uporabimo tudi –o ro), *medij* je absolutna pot do diska katereg smo priklopili, *direktorij* je mesto na katereg bomo priklopili disk. Za primer – če nam se korenski datotečni sistem nahaja v /dev/hdd2 in ga želimo priklopit na /mnt, potem ukaz izgleda:

mount –r /dev/hdd2 /mnt

Ko smo priklopili disk, dobro je spoznat se z Linuxom in njegovimi posebnosti.

* 1. **Pridobivanje zbrisanih podatkov**

V Linux-u ne obstaja file slack – ko se naredi nova datoteka preostali del sektorja se napolni z ničlami in se ta prostor označi kot nealociran. Eden od načinov na kateri lahko pridemo do zbrisanih podatkov je da poiščemo vse zbrisane inode in pridobimo nazaj podatke.

examiner1% ils -f linux-ext2 /e1/case2/ext2-bitstream.dd | more

class|host|device|start\_time

ils|case|ext2-bitstream.dd|1054082181

st\_ino|st\_alloc|st\_uid|st\_gid|st\_mtime|st\_atime|st\_ctime|st\_dtime|st\_mode|st\_nli

nk|st\_size|st\_block0|st\_block1

1|a|0|0|973385730|973385730|973385730|0|0|0|0|0|0

24|f|500|500|973695537|973695537|973695537|973695537|40700|0|0|308|0

25|f|500|500|954365144|973695521|973695537|973695537|100600|0|28587|309|310

26|f|500|500|954365144|973695521|973695537|973695537|100600|0|340|338|0

2049|f|500|500|973695537|973695537|973695537|973695537|40700|0|0|8489|0

2050|f|500|500|953943572|973695536|973695537|973695537|100600|0|4178|8490|8491

2051|f|500|500|960098764|973695521|973695537|973695537|100600|0|52345|8495|8496

2052|f|500|500|953943572|973695537|973695537|973695537|100600|0|4860|8548|8549

2053|f|500|500|959130680|973695521|973695537|973695537|100600|0|28961|8553|8554

**2054|f|500|500|959130680|973695521|973695537|973695537|100600|0|87647|8583|8584**

2055|f|500|500|961959437|973695521|973695537|973695537|100600|0|30799|8670|8671

2056|f|500|500|959130680|973695521|973695537|973695537|100600|0|50176|8702|8703

2057|f|500|500|953943572|973695537|973695537|973695537|100600|0|21700|8752|8753

2058|f|500|500|959130680|973695521|973695537|973695537|100600|0|22865|8775|8776

2059|f|500|500|959130680|973695521|973695537|973695537|100600|0|14584|8799|8800

2060|f|500|500|953943572|973695521|973695537|973695537|100600|0|12276|8815|8816

2061|f|500|500|959130680|973695521|973695537|973695537|100600|0|10840|8827|8828

2062|f|500|500|959130680|973695521|973695537|973695537|100600|0|26027|8838|8839

Ko pridemo do številke inode, lahko pridemo do podatkov na katere kaže inode z komando icat.

examiner1% **icat -f linux-ext2 ext2-bitstream.dd 2054**

/\*

dcc.c –– handles:

activity on a dcc socket

disconnect on a dcc socket

...and that's it! (but it's a LOT)

dprintf'ized, 27oct95

\*/

/\*

This file is part of the eggdrop source code

copyright (c) 1997 Robey Pointer

and is distributed according to the GNU general public license.

For full details, read the top of 'main.c' or the file called

COPYING that was distributed with this code.

\*/

#if HAVE\_CONFIG\_H

#include <config.h>

Tukaj lahko pride do problemov ker Linux pri brisanju podatkov odstranjuje referenco z inode na sektorje na katerim so zapisani podatki.

Obstaja še eden pogosto uporabljen način pridobivanja zbrisanih podatkov – preverijo se direktoriji in iščejo se zbrisani vnosi, če obstajajo.

Obstajajo številna forenzična orodja za pridobivanje zbrisanih podatkov katera uporabljajo eden od opisana dva načina, ali uporabljajo oboje(Sleuth Kit, The Linux disk editor, debugfs, The SMART tool, The Autopsy Forensic Browser…).

Kot še eden način pridobivanja zbrisanih podatkov lahko uporabimo file carving.

File carving je postopek pridobivanja zbrisanih podatkov uporabljajoč klasne karakteristike. Komande scalpel in foremost se najbolj pogosto uporabljajo za file carving – sprehajajo se čez bitstream kopijo diska in pregledujejo glave in noge datotek iskajoč klasne karakteristike. Za primer, če glava datoteke se začne z D0 CF, potem smo našli skrito(izbrisano) .doc datoteko in pročitamo vse podatke do EOF znaka.

Primer file carvinga:

examiner1% **foremost –o carved-foremost –v floppycopy.dd**

foremost version 0.62

Written by Kris Kendall and Jesse Kornblum.

Using output directory: /e1/carved-foremost

Verbose mode on

Using configuration file: foremost.conf

Opening /e1/linuxpractical.dd.

Total file size is 1474560 bytes

/e1/case2/floppycopy.dd: 100.0% done (1.4 MB read)

A doc was found at: 17408

Wrote file /e1/case2/carved-foremost/00000000.doc -- Success

A doc was found at: 37888

Wrote file /e1/case2/carved-foremost/00000001.doc -- Success

A jpg was found at: 76800

Wrote file /e1/case2/carved-foremost/00000002.jpg -- Success

A jpg was found at: 77230

Wrote file /e1/case2/carved-foremost/00000003.jpg -- Success

A jpg was found at: 543232

Wrote file /e1/case2/carved-foremost/00000004.jpg -- Success

A gif was found at: 990208

Wrote file /e1/case2/carved-foremost/00000005.gif -- Success

A jpg was found at: 1308160

Wrote file /e1/case2/carved-foremost/00000006.jpg -- Success

Foremost is done.

Linux operacijski sistem ima eno zelo korisno lastnost – Linux uporablja grupe blokov za shranjevanje podatkov in podatki na ta način ustvarjajo clusterje na disku. Za primer, če sumimo da je zlikovec obrisal eno log datoteko z diska, lahko preberemo celoten blok z diska na katereg je shranjena vsebina /var/log in poiščemo vse zbrisane datoteke.

* 1. **Skrivanje podatkov**

Prikazali bomo par načini skrivanja podatkov – dva preprosta z manipulacijo imen datotek, skrivanje v file slack, steganografijo in alternativni tok podatkov.

Najpreprostejši način skrivanja je preimenovanje končnice datoteke, npr. datoteko child.gif preimenujemo v child.doc . Ta način je tudi najlažje odkriti ker lahko pogledamo glavo datoteke. Vsak format datoteke se začne z drugo vrednostjo. V tabeli je začetek glave za 3 priljubljena formata:



Če pogledamo glavo child.txt in vidimo da se začne z 47 49 46, potem se dela o .gif datoteki.

Drugi način je tudi enostaven – če na konec imena datoteke dodamo ~ (tilda), Linux označi datoteko kot backup in jo zaradi tega skrije. Če pa preimenujemo datoteko na način da se začne z . (pika), potem Linux prepozna datoteko kot skrito in jo ne prikaže. Oba načina se lahko odkrijejo če uporabljamo izpis z skriti datotekami.

Tretji način skrivanja je skrivanje podatkov v file slack. File slack je prostor kateri ostane izmed EOF oznake datoteke in konca sektorja, npr če imamo sektor od 1024 bajtov in datoteko velikosti 1010 bajtov, potem imamo file slack od 14 bajtov v katereg lahko skrijemo podatke. Podatki se lahko skrijejo z komanda bmap in zastavico slack. Za primeru bomo v file slack skrili informacijo »cybercriminal«:

[root@tortilla slack]# bmap --slack file1.txt

getting from block 139522

file size was: 10

slack size: 1014

block size: 1024

cybercriminal

Skrivanje podatkov v file slack se redko uporablja zaradi dveh razlogov:

* v file slack lahko skrijemo podatke majhne velikosti
* če spremenimo velikost datoteke katera se nahaja v sektorju(dodamo znake), lahko povozimo skrite podatke v file slacku.

Četrti način skrivanja podatkov je steganografija.

Steganografija je znanost skrivanja podatkov v informacijo na način da do podatkov lahko pridejo le pošiljatelj in prejemnik informacije. Steganografija obstaja od antične zgodovine in se je razvijala v različnim oblikami(pisma, nevidna črnila...). V seminarski nas zanima digitalna steganografija.

Digitalna steganografija uporablja skrivanje podatkov v že obstoječimi datotekami. Obstajajo 3 metode:

* Metoda vstavljanja – sporočilo se vstavi v datoteko na način da ne vpliva na uporabo datoteke, npr za EOF znakom
* Metoda generiranja – generiramo novo datoteko glede na sporočilo katereg skrivamo
* Metoda zamenjave – sporočilo vnesemo v datoteko tako da prepišemo najmanj pomembne bite posameznih bajtov

Metoda zamenjave se posebej uporablja pri digitalnimi slika. Razlog je precej enostaven – če se spremenijo najmanj pomebmne informacije(biti) slike, človeško oko nemore zaznat spremembe.

Obstajajo številna ogrodja za steganografijo, katera so sestavljena iz:

* Podatkov katere želimo skriti
* Nosilec – datoteka v katero želimo skrit podatke, npr digitalna slika v .png formatu. Nosilec moramo izabrat na način da spremenjeni nosilec z skriti podatki in orginalni nosilec bojo dokaj podobni
* Verige – podatki so lahko skriti v nosilcu na različnimi mesti
* Enkripcijski in dekripcijski algoritem

Najbolj poznato prosto dostopno orodje za steganografijo je OpenPuff, kateri omogoča skrivanje podatkov v slike, avdio in video podatke. Za skrivanje uporablja več enkripcijskih algoritmov – CSPRNG, hash enkripcijo...

Za primer steganografije so priložene dve slike – orginalna v katero je podtaknjena skrita slika.

 

Peti način skrivanja podatkov je uporaba alternativneg toka podatkov(alternative data stream – ADS). ADS je funkcija NTFS datotečneg sistema katera je razvita da bi datotečni sistem bil kompaktibilen z Machintosh računalniki. Ampak obstajajo virusi in trojanci kateri izkoriščjajo ADS za dodajanje ene datoteke na drugo datoteko na način da ni vidljiva uporabnikom. Obstajajo številna forenzična orodja katera zaznavajo podatke v ADS.

Najbolj siguren način skrivanja podatkov je uporaba močnih enkripcijskih algoritmov. Tudi če vemo da obstaja datoteka in kje se nahaja, še vedno nemoremo prit do vsebine datoteke če ne poznamo ključa za dektriptiranje.

* 1. **Vremenske zanke v Linux sistemi**

Pri forenzični raziskavi diska so nam vremenske zanke od velikeg značaja ker iz njih lahko dobimo veliko informaciji o podatki zapisanimi na disku. V spodnji tabeli so opisani načini spremembe vremenskih zank pri manipulaciji podatkov na disku:



V Linux sistemi obstajajo 2 vrsti vremena: ctime in mtime. Ctime je vremenska zanka katera se spreminja če se delajo akcije z datoteko(npr. kopiranje datoteke), mtime je vremenska zanka katera se spreminja če se spreminja vsebina datoteke.

Enkrat ko se zbriše datoteka se prekine povezava izmed datoteke in njeneg inode. Ker inode ni dostopen za pregled v datotečnem sistemu, podatek kdaj je zbrisana datoteka lahko ostane zapisana v inode dokler se inode znova ne uporabi.

Ko se datoteka doda ali odstrani iz mape, inode spremeni vremensko zanko direktorija in vreme zadnjeg pristopa in izmenjave datoteke. Zaradi tega se spremeni ctime direktorija če se zbriše kakšna datoteka znotraj direktorija. To nam se lahko pokaže zelo koristno v preiskavi, ker lahko primerjamo ctime direktorija in vseh njemu pripadajočih datotek in vidimo ali je znotraj direktorija obrisana kakšna datoteka.

* 1. **Log datoteke v Linux**

V Linux operacijskimi sistemi obstaja veliko število log datotek katere beležijo podatke kateri nam olajžajo raziskavo diska. Log datoteke se nahajajo v direktoriju /var/log.

Primeri log datotek:

/var/log/message: generalna sporočila in sporočila povezana z sistemom

/var/log/auth.log: autentikacija

/var/log/kern.log: jedro

/var/log/cron.log: Cron

/var/log/maillog: poštni strežnik

/var/log/qmail/ : Qmail

/var/log/httpd/: Apache strežnik in sporočila o napakami

/var/log/boot.log : System boot sporočila

/var/log/mysqld.log: MySQL server sporočila

/var/log/secure: autentikacija

/var/log/utmp ali /var/log/wtmp: login sporočila

Log datoteke lahko tudi imajo aplikacije in se ponavadi nahajajo na mestu določenem v aplikaciji.

Log datoteke so bogata riznica informacij, ampak moramo z njimi delat pažljivo zaradi tega ker se informacije v log datoteka lahko poneverijo.

* 1. **Disk zaščiten z gesli in enkripcijo**

Podatki na disku so lahko zaščiteni z gesli ali z enkripcijo. V Linuxu obstaja komada crpyt, katera dela enostavno enkripcijo.

% crypt -key 'guessme' < plaintext> ciphertext

Klasične distribucije Linuxa so za enkripcijo uporabljale DES enkripcijski algoritem, kateri se danes obravnava kot šibek enkripcijski algoritem katereg je zelo lahko dekriptirat uporabljajoč brute-force napad. Novejše distribucije Linuxa uporabljajo močne enkripcijske algoritme, kot MD5(po najnovejšimi istraživanji se tudi smatra šibkim algoritmom!), Blowfish, SHA-256 ali SHA-512.

Pogosto se orginalni tekst samo zbriše z diska, in uporabljajoč že opisane metode lahko pridemo do teksta tudi brez dekriptiranja. Za dekripcijo lahko tudi probamo uganit skriti ključ, ker se pogosto za ključ uporabi beseda katero je zelo enostavna, ali pri raziskavi diska pridemo do podatka kateri bi lahko bil skriti ključ. Če lastnik diska nima velikeg tehničneg znanja, potem sigurno obstaja nekje zapisani ključ, ali ga je uporabil na kakšnem drugem mestu. Pogosto se za skrite ključe in gesla tudi uporabljajo besede katere imajo nekakšen pomen lastniku. Številni hekerji danes uporabljajo tudi metode socijalneg istraživanja lastnikov diski da bi prišli do gesli.

Po nekaterimi istraživanju, še vedno je več kot 40% vseh gesli na svetu katera so zelo lahko ugibljiva, kot »password«, »12345678«, »qwerty«...

Po zakoni Republike Slovenije je lastnik diska dolžan dati preiskovalcem vsa gesla in skrite ključe, v sprotnem ga se lahko kaznuje.

Na Internetu obstaja veliko število orodji za razbijanje enkripcije, najbolj poznata so Crack in Jack the Ripper.

**Primer raziskave diska**

Forenzično raziskavo diska bomo prikazali na živem primeru.

Ko smo priklopili disk, lahko si izpišemo vse direktorije na disku z naredbo ls

# ls -lat /mnt

total 73

drwxr-x--- 17 root root 1024 May 1 09:01 root

drwxrwxrwt 6 root root 1024 May 1 04:03 tmp

drwxr-xr-x 8 root root 34816 Apr 30 04:02 dev

drwxr-xr-x 34 root root 3072 Apr 29 14:17 etc

drwxr-xr-x 2 root root 2048 Apr 26 16:52 bin

drwxr-xr-x 2 root root 1024 Apr 26 11:12 boot

drwxr-xr-x 3 root root 3072 Apr 21 04:01 sbin

drwxr-xr-x 4 root root 3072 Apr 21 03:56 lib

drwxrwxr-x 2 root root 1024 Mar 3 13:27 cdrom

drwxr-xr-x 2 root root 1024 Oct 9 1999 home

drwxr-xr-x 2 root root 12288 Oct 9 1999 lost+found

drwxr-xr-x 4 root root 1024 Oct 9 1998 mnt

drwxr-xr-x 2 root root 1024 Oct 9 1999 proc

drwxr-xr-x 20 root root 1024 Aug 2 1998 usr

drwxr-xr-x 18 root root 1024 Aug 2 1998 var

Če preverimo imena direktorijev – root, tmp, dev, etc, bin..., lahko pridemo do zaključka da je disk katereg raziskujemo zares korenski datotečni sistem.

V /etc/fstab so vse particije našeg datotečneg sistema.

# less /mnt/etc/fstab

. . .

/dev/hda1 /dosc msdos defaults 0 0

/dev/hda2 / ext2 defaults 1 1

/dev/hda4 /home ext2 defaults 1 2

/dev/hda3 swap swap defaults 0 0

/dev/cdrom /cdrom iso9660 noauto,user,ro 0 0

/dev/fd0 /floppy ext2 noauto,user,rw 0 0

none /proc proc defaults 0 0

none /dev/pts devpts mode=0622 0 0

Lotimo se raziskave diska. Najprej preverimo kaj je zapisano v /etc/passwd v katerem so zapisani uporabniški računi. Tukaj so lahko tudi zapisani računi katere je ustvaril zlikovec.

# less /mnt/etc/passwd

. . .

root:x:0:0:root:/root:/bin/bash

bin:x:1:1:bin:/bin:

daemon:x:2:2:daemon:/sbin:

adm:x:3:4:adm:/var/adm:

lp:x:4:7:lp:/var/spool/lpd:

sync:x:5:0:sync:/sbin:/bin/sync

shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown

z:x:0:0::/:/bin/bash

halt:x:7:0:halt:/sbin:/sbin/halt

mail:x:8:12:mail:/var/spool/mail:

news:x:9:13:news:/var/spool/news:

uucp:x:10:14:uucp:/var/spool/uucp:

operator:x:11:0:operator:/root:

r00t:x:598:500:::/bin/bash

games:x:12:100:games:/usr/games:

y:x:900:100::/tmp:/bin/bash

gopher:x:13:30:gopher:/usr/lib/gopher-data:

ftp:x:14:50:FTP User:/home/ftp:

nobody:x:99:99:Nobody:/:

gdm:x:42:42::/home/gdm:/bin/bash

xfs:x:100:233:X Font Server:/etc/X11/fs:/bin/false

user1:x:500:500:User 1:/home/user1:/bin/tcsh

user2:x:501:501:User 2:/home/user2:/bin/tcsh

user3:x:502:502:User 3:/home/user3:/bin/tcsh

named:x:25:25:Named:/var/named:/bin/false

Preverimo kateri računi nam izgledajo zelo sumnjivi, kot r00t (sumnjivo ime), y (domači imenik mu je /tmp), named (zadnji račun) in x (uid=0 in gid=0, isto kot korenski uporabnik!).

Vzemimo za primer y in gremo pogledat kaj je v /tmp direktoriju.

# ls -lat /mnt/tmp

total 156

drwxrwxrwt 6 root root 1024 May 1 04:03 .

-r--r--r-- 1 root gdm 11 Apr 29 14:17 .X0-lock

drwxrwxrwt 2 root gdm 1024 Apr 29 14:17 .X11-unix

drwxrwxrwt 2 xfs xfs 1024 Apr 29 14:17 .font-unix

drwxr-xr-x 25 y root 1024 Apr 28 23:47 ..

drwx------ 2 user1 user1 1024 Apr 26 17:36 kfm-cache-500

-rw-rw-r-- 1 user1 user1 12288 Apr 26 16:37 psdevtab

drwxrwxrwt 2 root root 1024 Apr 21 11:12 .ICE-unix

-rwx------ 1 root root 138520 Apr 20 20:15 .fileMFpmnk

Zadnja datoteka ima zelo čudno ime in je izvršna. Z komando string preverimo kaj je vpisano v datoteki.

# strings - /mnt/tmp/.fileMFpmnk

/lib/ld-linux.so.2

\_\_gmon\_start\_\_

libpam.so.0

\_DYNAMIC

\_GLOBAL\_OFFSET\_TABLE\_

pam\_set\_item

free

\_\_ctype\_tolower

malloc

strcmp

pam\_end

pam\_start

. . .

File

Compressed

Block

Stream

[nowhere yet]

ftpd

:aAvdlLiop:P:qQr:sSt:T:u:wWX

bad value for -u

option -%c requires an argument

unknown option -%c ignored

. . .

VirtualFTP Connect to: %s [%s]

banner

logfile

email

/var/log/xferlog

connection refused (server shut down) from %s

%s FTP server shut down -- please try again later.

lslong

/bin/ls -la

lsshort

lsplain

/bin/ls

greeting

full

terse

brief

%s FTP server (%s) ready.

%s FTP server ready.

FTP server ready.

. . .

FTP LOGIN REFUSED (already logged in as %s) FROM %s, %s

Already logged in.

/etc/ftphosts

FTP LOGIN REFUSED (name in %s) FROM %s, %s

anonymous

FTP LOGIN REFUSED (anonymous ftp denied on default server) FROM %s, %s

FTP LOGIN REFUSED (ftp in denied-uid) FROM %s, %s

/etc/ftpusers

Datoteka izgleda kot FTP strežnik, katere se ponavadi imenujejo ftpd ali .ftpd. Zaradi tega datoteko označimo kot sumnjivo.

Potem je dobro preverit nastavitvene datoteke v /dev direktoriju, kje se lahko nahajajo datoteke katere so ostale za zlikovcem.

# cd /mnt/dev

# ls -lat | head -30

total 116

drwxr-xr-x 8 root root 34816 Apr 30 04:02 .

srw-rw-rw- 1 root root 0 Apr 30 04:02 log

crw------- 1 root root 4, 1 Apr 29 14:17 tty1

crw------- 1 root root 4, 2 Apr 29 14:17 tty2

crw------- 1 root root 4, 3 Apr 29 14:17 tty3

crw------- 1 root root 4, 4 Apr 29 14:17 tty4

crw------- 1 root root 4, 5 Apr 29 14:17 tty5

crw------- 1 root root 4, 6 Apr 29 14:17 tty6

srwxrwxrwx 1 root root 0 Apr 29 14:17 gpmctl

srw------- 1 root root 0 Apr 29 14:17 printer

crw-r--r-- 1 root root 1, 9 Apr 29 14:17 urandom

prw------- 1 root root 0 Apr 29 14:14 initctl

drwxr-xr-x 25 y root 1024 Apr 28 23:47 ..

crw-rw-rw- 1 root tty 3, 2 Apr 28 11:44 ttyp2

crw-rw-rw- 1 root tty 3, 0 Apr 28 11:43 ttyp0

crw-rw-rw- 1 root tty 3, 1 Apr 28 11:43 ttyp1

-rw-r--r-- 1 root root 18 Apr 27 22:58 ptyp

drwxr-xr-x 4 r00t root 1024 Apr 27 22:58 ...

crw-rw-rw- 1 root tty 3, 4 Apr 27 12:02 ttyp4

crw-rw-rw- 1 root tty 3, 3 Apr 27 11:56 ttyp3

crw------- 1 root root 5, 1 Apr 21 11:09 console

lrwxrwxrwx 1 root root 5 Apr 21 04:02 mouse -> psaux

drwxr-xr-x 2 root root 1024 Apr 20 15:21 rev0

-rw-r--r-- 1 root root 33 Apr 20 15:21 ptyr

lrwxrwxrwx 1 root root 9 Feb 28 02:23 isdnctrl -> isdnctrl0

lrwxrwxrwx 1 root root 5 Feb 28 02:23 nftape -> nrft0

lrwxrwxrwx 1 root root 3 Feb 28 02:23 fb -> fb0

lrwxrwxrwx 1 root root 15 Feb 28 02:23 fd -> ../proc/self/fd

lrwxrwxrwx 1 root root 4 Feb 28 02:23 ftape -> rft0

Broken pipe

Sumnjive so nam datoteke ptyp in ptyr (- kot prvi znak v opisu datoke), direktorij rev0 in skriti direktorij ..

Preverimo kaj se nahaja v ptyp:

# less ptyr

. . .

sp.pl

slice

ssynk4

rev0

bc1

snif

Datoteke ke se nahajajo v ptyp so že poznate kot konfiguracijske datoteke za trojanske konje. Lahko preverimo kje se vse uporabljajo datoteke:

# cd /mnt

# find . -ls | grep -f etc/ptyr

282058 1 drwxr-xr-x 2 root root 1024 Apr 20 15:21 ./dev/rev0

282059 1 -rw-r--r-- 1 root root 5 Apr 20 15:21 ./dev/rev0/sniff.pid

282061 20 -rw-r--r-- 1 root root 19654 Apr 20 20:23 ./dev/rev0/tcp.log

164753 9 -rwxr-xr-x 1 1080 users 9106 Sep 20 1999 ./dev/rev0/slice

164754 8 -rwxr-xr-x 1 1080 users 8174 Sep 20 1999 ./dev/rev0/smurf4

164755 8 -rwxr-xr-x 1 1080 users 7229 Sep 20 1999 ./dev/rev0/snif

164756 4 -rwxr-xr-x 1 1080 users 4060 Mar 5 1999 ./dev/rev0/sp.pl

164770 9 -rwxr-xr-x 1 root 1000 8268 Aug 10 1999 ./dev/.../blitznet/slice2

61907 2 -rwxr-xr-x 1 root root 2006 Mar 29 1999 ./usr/bin/sliceprint

255230 1 -rw-r--r-- 1 root root 900 Mar 21 1999 ./usr/include/python1.5/sliceobject.h

Nekatere od njih so sigurno legitimne sistemske datoteke, ampak nam je sumnjivo zakaj se uporabljajo v /dev, pa gremo en korak naprej in preverimo /dev.

# cd /mnt/dev

# less ptyp

. . .

3 egg

3 egg

3 bnc

Obstaja trojanski konj kateri skriva procese egg in bnc v izpisu komande ps. Preverimo kje se nahajajo izvršne datoteke z timi nazivi.

# cd /mnt/dev

# ls -lR ...

...:

total 2699

drwxr-sr-x 2 root 1000 1024 Aug 10 1999 blitznet

-rw-r--r-- 1 root root 30720 Apr 26 04:07 blitznet.tar

-rwxrw-r-- 1 r00t user1 22360 Apr 27 22:58 bnc

-rw-r--r-- 1 900 users 2693120 Apr 20 22:18 collision.tar

-rw-rw-r-- 1 r00t user1 976 Apr 27 22:58 example.conf

-rw-rw-r-- 1 user1 user1 5 Apr 28 20:35 pid.bnc

.../blitznet:

total 22

-rw-r--r-- 1 root 1000 3450 Aug 10 1999 README

-rw-r--r-- 1 root 1000 1333 Aug 10 1999 blitz.c

-rw-r--r-- 1 root 1000 3643 Aug 10 1999 blitzd.c

-rwxr-xr-x 1 root 1000 2258 Aug 10 1999 rush.tcl

-rwxr-xr-x 1 root 1000 8268 Aug 10 1999 slice2

Direktorij /dev/rev0 se nahaja v izpisu. Lahko ga preverimo.

# ls -lR rev0

rev0:

total 51

-rwxr-xr-x 1 1080 users 9106 Sep 20 1999 slice

-rwxr-xr-x 1 1080 users 8174 Sep 20 1999 smurf4

-rwxr-xr-x 1 1080 users 7229 Sep 20 1999 snif

-rw-r--r-- 1 root root 5 Apr 20 15:21 sniff.pid

-rwxr-xr-x 1 1080 users 4060 Mar 5 1999 sp.pl

-rw-r--r-- 1 root root 19654 Apr 20 20:23 tcp.log

# cd /mnt/usr/bin

# ls -lat | head

total 89379

drwxr-xr-x 6 root root 27648 Apr 21 04:01 .

-rwsr-xr-x 1 root root 20164 Apr 15 19:23 chx

lrwxrwxrwx 1 root root 8 Feb 28 02:28 netscape-navigator -> netscape

drwxrwxr-x 2 news news 1024 Feb 28 02:25 rnews.libexec

drwxrwxr-x 2 news news 1024 Feb 28 02:25 control

drwxrwxr-x 2 news news 1024 Feb 28 02:25 filter

lrwxrwxrwx 1 root root 4 Dec 30 13:06 elatex -> etex

lrwxrwxrwx 1 root root 5 Dec 30 13:06 lambda -> omega

lrwxrwxrwx 1 root root 3 Dec 30 13:06 latex -> tex

Broken pipe

# strings - chx

/lib/ld-linux.so.2

\_\_gmon\_start\_\_

libcrypt.so.1

libpam.so.0

. . .

/var/log/btmp

/usr/share/locale

util-linux

fh:p

login: -h for super-user only.

usage: login [-fp] [username]

/dev/tty

%s??

/dev/vcs

/dev/vcsa

login

login: PAM Failure, aborting: %s

Couldn't initialize PAM: %s

FAILED LOGIN %d FROM %s FOR %s, %s

Login incorrect

TOO MANY LOGIN TRIES (%d) FROM %s FOR %s, %s

FAILED LOGIN SESSION FROM %s FOR %s, %s

Login incorrect

.hushlogin

%s/%s

/var/run/utmp

/var/log/wtmp

/bin/sh

TERM

dumb

HOME

/usr/local/bin:/bin:/usr/bin

PATH

/sbin:/bin:/usr/sbin:/usr/bin

SHELL

/var/spool/mail

MAIL

LOGNAME

DIALUP AT %s BY %s

ROOT LOGIN ON %s FROM %s

ROOT LOGIN ON %s

LOGIN ON %s BY %s FROM %s

LOGIN ON %s BY %s

You have %smail.

new

login: failure forking: %s

setuid() failed

No directory %s!

Logging in with home = "/".

login: no memory for shell script.

exec

login: couldn't exec shell script: %s.

login: no shell: %s.

%s login:

login name much too long.

NAME too long

login names may not start with '-'.

too many bare linefeeds.

EXCESSIVE linefeeds

Login timed out after %d seconds

/etc/securetty

/etc/motd

/var/log/lastlog

Last login: %.\*s

from %.\*s

on %.\*s

LOGIN FAILURE FROM %s, %s

LOGIN FAILURE ON %s, %s

%d LOGIN FAILURES FROM %s, %s

%d LOGIN FAILURES ON %s, %s

. . .

Sporočila o napakami in referenca na hushlogin nam ukazuje da se dela o trojanskem konju, verzija login.

Informacije o simboli so vključene v kompajliranimi objekti če niso odstranjene. Preverimo:

# nm chx

chx: no symbols

Preverimo kaj se nahaja v dinamičnimi knjižnicami.

# ldd chx

libcrypt.so.1 => /lib/libcrypt.so.1 (0x40018000)

libpam.so.0 => /lib/libpam.so.0 (0x40045000)

libdl.so.2 => /lib/libdl.so.2 (0x4004d000)

libpam\_misc.so.0 => /lib/libpam\_misc.so.0 (0x40050000)

libc.so.6 => /lib/libc.so.6 (0x40054000)

/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)

Zdaj vidimo da se je uporabljal crypt() in Pluggable Authentication Module (PAM), kaj nedvomno ukazuje na trojanskeg konja.

Na koncu raziskave še enkrat izračunamo MD5 hash kodo in primerimo z orginalno kodo da zagotovimo nespremenjljivost podatkov.

Z tim smo zaključili raziskavo diska.

**LITERATURA**

1. Knjiga
2. Robertson, G., »An Introduction to Hiding and Finding Data on Linux« - GIAC Security Essentials Certification, SANS Institute, 2003.
3. Artz, D.,«Digital steganography: hiding data within data”, Internet Computing, volume 5 issue 3, 75-80, 2001.
4. Huebner, E., Bem, D., Kai Wee, C., “Data hiding in the NTFS file system”, Digital investigation, volume 3 issue 4, 211-226, 2006.
5. Dittrich, D., “Basic steps in forensics analysis of Unix systems”, <http://staff.washington.edu/dittrich/misc/forensics/>, datum pristopa: 13.5.2012.