Global Information Assurance Certification Paper

Intrusion Detection In-Depth

GCIA Practical Assignment for

SANS Parliament Square, London

June 20 – 23, 2001

Version 2.9

Philipp STADLER

Title: bypassing masquerading firewalls

TABLE OF CONTENTS

Assignment 1 – Network Detects

1. Trojan port scan

data:

```
2001-03-18-20:41:30  tcp   152.30.98.87:3908   ->    130.216.8.24:27374  S
2001-03-18-20:41:30  tcp   152.30.98.87:3909   ->    130.216.8.24:12345  S
2001-03-18-20:41:30  tcp   152.30 .98.87:3910  ->    130.216.8.24:139    S
2001-03-18-20:56:14  tcp   152.30.98.87:4560   ->  130.216.72.231:27374  S
2001-03-18-20:56:14  tcp   152.30.98.87:4561   ->  130.216.72.231:12345  S
2001-03-18-20:56:14  tcp   152.30.98.87:4562   ->  130.216.72.231:139    S
2001-03-18-21:57:40  tcp   152.30.98.87:4864   ->  130.216.108.39:27374  S
2001-03-18-21:57:40  tcp   152.30.98.87:4874   ->  130.216.108.39:12345  S
2001-03-18-21:57:40  tcp   152.30.98.87:4878   ->  130.216.108.39:139    S
2001-03-18-23:16:54  tcp   152.30.98.87:4911   ->  130.216.207.219:27374  S
```

1.1 Source of trace

GIAC: http://www.sans.org/y2k/031901.htm

1.2 Detect was generated by

Probably this output is generated by the portscan module of the SNORT Intrusion Detection System. (referenced to the SANS course book 3.3/3.4).

1.3 Probability the source address was spoofed

In this case the purpose of this network scan in to gather information about the destination systems (if the spec ified ports 139, 12345 and 27374 are listening). The sender of these packets is waiting for the response and so the source address is probably not spoofed. The only possibility that this address is spoofed is the „Kevin Mitnick“ – way. (Spoof the source ad dress of a silent Windows host)

1.4 Description of attack

The source is searching for some Trojan horses or file shares on the victim systems.
Because of the time gaps between the scans the attacker will likely scan the whole 130.216.x.x class B network. Also the increasing source ports (and the gaps between the scans) point to a whole class B scan.

The domain name of the source 152.30.98.87 is: rn098087.wcu.edu

The WHOIS query for the source domain name results:
Registrant: Western Carolina University (WCU-DOM)
B-10 Forsyth Bldg.
Cullowhee, NC 28723
US

Domain Name: WCU.EDU

Administrative Contact, Technical Contact, Billing Contact:
Swartzentruber, Scott (SS30838) scotts@WCU.EDU
Western Carolina University / Computer Center
Forsyth Bldg, Room B -10
Cullowhee, NC 28723
828.227.7282 (FAX) 828.227.7700

Record last updated on 07 -Aug-2000.
Record created on 26 -Jan-1990.
Database last updated on 4 -Sep-2001 22:17:00 EDT.

Domain servers in listed order:
COWEE.WCU.EDU 152.30.2.120
NS.GA.UNC.EDU 152.4.20.3
NCNOC.NCREN.NET 192.101.21.1

1.5 Attack mechanism

This is a network scan to three different ports. The source address is from the IP address space of Western Carolina University.
There are two Trojan horses related to the first port (27374), the most popular one is the SubSeven Trojan, the other one is BadBlood. The second port (12345) is related to NetBus and some more Trojans. The third port (139) is known as the „NETBIOS session service“, this port is used on Windows Systems for file sharing.
The attacker tried to begin the three -way handshake to the shown ports to search at the victims, if one of these Trojan horses is installed.

The NetBus Trojan is similar to the Back Orifice, this tool is promoted as a remote administration tool. You can fully control the remote host by this tool. The SubSe ven is also a „remote administration tool“, it can do everything that NetBus can do. (citation of reference 1)
This includes:

File controls
Upload / Download
Move, Copy, Rename, Delete
Erase hard drives and other disks
Execute programs
Monitoring
Can see your screen as you see it
Log any/all keypresses (even hidden passwords)
Open/close/move windows
Move mouse
Network control
Can see all open connections to and from your computer
Can close connections
Can 'bounce' or relay from their system to yours, so wherever they connect, it seems as if you are doing it. This is how they prevent getting caught breaking into other computer systems and you can get in trouble!

1.6 Correlations

This particular scan was detected by Security@auckland. This pattern has never been seen before in a scan, but the SYN -Scan used for this attack is a common way to look for listening ports. A known tool for this scan is NMAP. If the attacker used NMAP he wrote a command like this:
nmap –sS –p 27374,12345,139 < -P0>
The „–P0“ option is for suppressing the ping before scanning. The port scan is only a part of a whole scan process and so it's possible that the ping packets are not seen in the IDS.

1.7 Evidence of active targeting

We don't know the whole attack because only a part of it is shown in the IDS logs. If the shown IDS log would include all traces of this attack, a direct attack to a limited number of target systems could be assumed. But the publisher of the IDS log posted only a part of the data, so this scan is probably a general scan and not targeted to particular hosts. This is most likely not an active targeting attack.

1.8 Severity

GIAC's approach to determining severity is to apply the following formula:
(Criticality + Lethality) – (System + Net Countermeasures) = Severity
Each metric is assigned on a five -point scale (1 as the lowest and 5 as the highest).

Criticality -We don't know the real criticality of the systems and so I will set it to 3.
Lethality – This are Trojan horses with remote administration features, so if this Trojan is installed on the victim the attacker can do „everything“, therefore the Lethality is 5.
System Countermeasure - The system countermeasure is 5, because none of the systems are infected to this Trojans (or won't answer to the scanned ports). The problem of this attack is the unknown network infrastructure and we don't know if the attack is blocked by a firewall or the ports are closed.
Network Countermeasure -..will be 4 because none of the attacks are successful (we don't know if this packets are blocked by firewall or the system isn't infected)

(Criticality	+ Lethality)	– (System Countermeasure	+ Network Countermeasure)	= **Severity**
(3	+ 5)	– (5	+ 4)	= **-1**

1.9 Defence recommendation

One possibility is to create input filters on a firewall or a packet filtering device to block all known Trojan ports. The much better possibility is surely to block all incoming traffic except the used and needed ports and also the established connections.
Another way to defend against such problems is to do regular scanning of the own network for the Trojan ports. Also integrity checkers should be installed on the systems to recognize chances in the systems.
An IDS or a logging facility is already installed because without this we wouldn't get the detect.

1.10 Multiple choice test question

Which are the interesting values in the following packets?

```
2001-03-18-20:41:30  tcp   152.30.98.87:3908    ->    130.216.8.24:27374  S
2001-03-18-20:41:30  tcp   152.30.98.87:3909    ->    130.216.8.24:12345  S
2001-03-18-20:41:30  tcp   152.30.98.87:3910    ->    130.216.8.24:139    S
2001-03-18-20:56:14  tcp   152.30.98.87:4560    ->   130.216.72.231:27374   S
2001-03-18-20:56:14  tcp   152.30.98.87:4561    ->   130.216.72.231:12345  S
2001-03-18-20:56:14  tcp   152.30.98.87:4562    ->   130.216.72.231:139    S
2001-03-18-21:57:40  tcp   152.30.98.87:4864    ->   130.216.108.39:27374  S
2001-03-18-21:57:40  tcp    152.30.98.87:4874   ->   130.216.108.39:12345  S
2001-03-18-21:57:40  tcp   152.30.98.87:4878    ->   130.216.108.39:139    S
```

a. same source IP address in all packets
b. the source ports
c. the destination ports
d. the timestamps

Answer: c. the destination port – because there are some Trojans associated to this ports.

2. telnet probe

data:

```
 Feb  6 03:12:28 takahe snort[58999]: IDS127  - TELNET -
  Login Incorrect: 130.216.3.40:23  -> 210.55.97.124:4210
Feb  6 03:16:54 takahe snort[58999]: IDS376  - FINGER-root:
  210.55.97.124:4618  -> 130.216.7.15:79
Feb  6 03:17:30 takahe snort[58999]: IDS127  - TELNET -
  Login Incorrect: 130.216.7.15:23  -> 210.55.97.124:1503
Feb  6 03:18:31 takahe snort[58999]: IDS127  - TELNET -
  Login Incorrect: 130.216.7.15:23  -> 210.55.97.124:2754
Feb  6 03:19:41 takahe snort[58999]: IDS378  - FINGER-Probe0:
  210.55.97.124:1287  -> 130.216.7.15:79
Feb  6 03:19:58 takahe snort[58999]: IDS378  - FINGER-Probe0:
  210.55.97.124:1447  -> 130.216.7.15:79
Feb  6 03:21:08 takahe snort[58999]: IDS127  - TELNET -
  Login Incorrect: 130.216.7.15:23  -> 210.55.97.124:3299
Feb  6 03:26:30 takahe snort[58999]: IDS304  - TELNET -
```

SGI telnetd format bug: 210.55.97.124:2407 -> 130.216.21.44:23
Feb 6 03:26:32 takahe snort[58999]: IDS304 - TELNET -
SGI telnetd format bug: 210.55.97.124:2453 -> 130.216.21.44:23
Feb 6 03:26:40 takahe snort[58999]: IDS304 - TELNET -
SGI telnetd format bug: 210.55.97.124:2674 -> 130.216.21.44:23
Feb 6 03:26:42 takahe snort[58999]: IDS304 - TELNET -
SGI telnetd format bug: 210.55.97. 124:2684 -> 130.216.21.44:23
Feb 6 03:26:44 takahe snort[58999]: IDS304 - TELNET -
SGI telnetd format bug: 210.55.97.124:2706 -> 130.216.21.44:23

2.1 Source of Trace

GIAC: http://www.sans.org/y2k/020701 .htm

2.2 Detect was generated by

The detect was generated by various of SNORT Intrusion Detection systems with the rules turned on for finger probes and denied telnet login attempts. A lot of data is missing because the IDS don't sniff all systems, but the one, who detect this scan, wrote that he saw a lot of telnet and finger attempts from this address to his network in the network traffic logs.

The following SNORT rules are written for Snort 1.8 and because of not knowing the exact version which are used by the detect it's possible that the rules can differ in some options.

The **finger-probe** alerts are triggered by:

```
alert tcp $EXTERNAL any -> $INTERNAL 79 (msg:" IDS378 - FINGER-Probe0";flags: A+;
content:"0";reference:arachnids,378; classtype:attempted -recon; sid:325; rev:1;)
alert tcp $EXTERNAL any -> $INTERNAL 79 (msg:" IDS376 - FINGER-root";flags: A+;
content:"root";reference:arachnids,376; classtype:attempted -recon; sid:323; rev:1;)
```

The **telnet login attempts** are triggered by this rule:

```
alert tcp $INTERNAL 23 -> $EXTERNAL any (msg:"IDS127 - TELNET login incorrect";
content:"Login incorrect"; flags: A+; reference:arachnids,127; classtype:bad -unknown; sid:718;
rev:1;)
```

The last five log entries (**telnet format bug**) are generated by the following SNORT rul e:

```
alert TCP $EXTERNAL any -> $INTERNAL 23 (msg: "IDS304 - telnet_SGI telnetd format
bug"; flags: A+; content: "_RLD"; classtype: system -attempt; reference: arachnids,304;)
```

2.3 Probability the source address was spoofed

It isn't probable that the source address is spoofed because the attacker wanted to get additional information from the scanned systems. If a stateful inspection firewall is used and the target systems are not vulnerable to sequence number prediction attacks I'm pretty certain that the source address of this attack isn't spoofed. (normally telnet service isn't running on windows systems, and those are the only known ones today which are vulnerable to sequence number prediction attacks)

2.4 Description of attack

The telnet log entries on th e IDS are certainly responses to incorrect telnet access attempts, the finger log entries are stimulus to get finger information from the scanned systems. In this summary of data (merging different IDS log files and different alert types) it is very easy t o find the stimuli and the responses.
The attacker scanned the network and probably tried default logins on the telnet server, he also tried to get additional information with using the finger service. On the last section of the scan we can see that he als o attempted to exploit the telnet service with a format bug. (reference 4)
This is a bug where unexpected characters can cause unwanted (unwanted by the administrators) errors like crashing the system.
The scan took about three hours (as the publisher wrot e).

2.5 Attack mechanism

The attacker completed the three -way handshake to a lot of destination addresses to port 23 (telnet) and port 79 (finger). On both connections he tried to get information or access to the system. With the SGI format bug weakness h e attempted to break in the system, this bug only can exploit IRIX systems. (reference 4)

Attackers Address: 210.55.97.124
Domain Name: ip124 -97.quik.co.nz

A WHOIS search for the source address reveals the following result:

```
inetnum          210.55.96.0 - 210.55.99.255
netname          QUIKINT-NZ
descr            Quik Internet (NZ) Ltd
descr            PO Box 7515
descr            Auckland
country          NZ
admin-c          BJ10-AP, inverse
tech-c           WEB193-ORG, inverse
rev-srv          ns.quik.co.nz
rev-srv          ns.quik.com
notify           nic@netgate.net.nz, inverse
changed          don@nertgate.net.nz 19991217
source           APNIC
```

2.6 Correlations

This particular scan was detected by Security@auckland . This attack pattern has never been seen before in this manner, but all parts of it are known and this scan is only a sum of three different recognitions. The telnet login attempts can be tested manually by starting a telnet client to all servers, this can also be easily auto mated.

2.7 Evidence of active targeting

There is no evidence of active targeting because this is a global network scan (the publisher wrote that there are a lot of such attempts in his network trace log) and no single host is specified.

2.8 Severity

GIAC's approach to determining severity is to apply the following formula:
(Criticality + Lethality) – (System + Net Countermeasures) = Severity
Each metric is assigned on a five -point scale (1 as the lowest and 5 as the highest).

Criticality –because of checking a lot of system also very critical systems can be scanned. (=4)
Lethality – If the systems are secured with a good passwords and if no IRIX system is in the scanned address space the Lethality of this attack will be 2.
System Countermeasure - The system countermeasure will be 4 because the login attempts failed but we don't know if one of the systems return finger information back to the attacker.
Network Countermeasure -..will be 1 because none of the telnet packets are blocked and we don't know how the finger packets will be handled by the firewall (if there is one)

(Criticality	+ Lethality	) – (System Countermeasure	+ Network Countermeasure	) = **Severity**
(4	+ 2	) – (4	+ 1	) = **1**

2.9 Defensive recommendation

Because telnet is an unencrypted remote administration tool (username and password are transmitted in cleartext) I strongly recommend that telnet shouldn't be used any more. The better way to get a remote shell is SSH (secure shell) because you get an encrypted channel to the server and it's much harder to get the username and password combination by sniffing the SSH packets. Also the finger service should be turned off on all systems. If all this is done, the security of the system s increases by a high amount.

2.10 Multiple choice test question

What type of attack are normally associated with finger and telnet service?

a. buffer overflows
b. Denial of Service attacks
c. Backdoor attacks
d. Additional reconnaissance

Answer: d. Additional reco nnaissance – because with both finger and telnet you get information about the victim

3. DNS scan

data:

```
Jan 14 18:52:53 [firewall.ip.address] %PIX -6-106015: Deny TCP (no connection)
  from 12.3.146.212/53 to dmz.ip.addr.2/53 flags FIN SYN  on interface   outside
Jan 14 18:52:53 [firewall.ip.address] %PIX -7-106011: Deny inbound (No xlate)
  tcp src outside:12.3.146.212/53 dst outside:cidr.net.addr.101/53
Jan 14 18:52:53 [firewall.ip.address] %PIX -7-106011: Deny inbound (No xlate)
  tcp src outside:12.3.14 6.212/53 dst outside:cidr.net.addr.102/53
Jan 14 18:52:53 [firewall.ip.address] %PIX -7-106011: Deny inbound (No xlate)
  tcp src outside:12.3.146.212/53 dst outside:cidr.net.addr.104/53
Jan 14 18:52:53 [firewall.ip.address] %PIX -7-106011: Deny inbound (N o xlate)
  tcp src outside:12.3.146.212/53 dst outside:cidr.net.addr.105/53
Jan 14 18:52:53 [firewall.ip.address] %PIX -6-106015: Deny TCP (no connection)
  from 12.3.146.212/53 to 192.149.115.2/53 flags FIN SYN  on interface outside
```

3.1 Source of Trace

GIAC: http://www.sans.org/y2k/011901.htm

3.2 Detect was generated by

This detect was generated by the logging system of a Cisco PIX firewall. The log format is self explaining. The lines with „%PIX -6-106015“ are connection attempts from the outside to a demilitarised zone, the „%PIX -7-106011“ lines are probes from outside to the inside network.

3.3 Probability the source address was spoofed

It’s improbable that the source address is spoofed because the at tacker is waiting for a reply. Without getting the reply he wouldn’t get the information about a listening port and a poorly maintained firewall, and this is most likely what he want to know. (Excluding the rare case where an attacker is located between th e spoofed Source and the victim host – there it is possible to obtain the response by sniffing the passing answer packets)

3.4 Description of attack

It’s not likely that the detected packets are responses, because it’s not probable that 6 systems attempt to connect to the same DNS server and get the reply in the same second, so these are almost certain stimulus packets.

This attack is a scan for an open TCP -port 53 which is associated with the domain name service. A lot of know bugs are possible to this port (mostly buffer overflows). The attacker used port 53 as the source port, because at poorly maintained firewalls there are inbound connections allowed if the source port is 53. This is normally used for the reply packets of outbound TCP -connections from a domain name server. (also possible from clients, which expect a big DNS reply). So if all packets are permitted from source port 53 at the firewall, this could make this attack possible. The attacker set the SYN and the FIN bit, and so you can be sure that this is a crafted packet.

One time again the attacker hoped that he scanned through a poorly maintained firewall. Because some poor firewalls (or old ones) only block inbound connections with only the SYN bit set, and if other bits are set, the packet will be passed. The problem is that normal systems will reply to a SYN/FIN packet with a SYN/ACK packet. And so it is possible that a connection can be established from outside without permission.

One problem of BIND is, that with the change from version 4 to 8, the code has undergone a major redesign. This also introduces the possibility for new unknown bugs. With version 9 we can hope dough that the code stabilizes again. Version 4 should under no circumstances be used since it contains known vulnerabil ities and is no longer maintained.

3.5 Attack mechanism

The attacker (source address: 12.3.146.212) tried to connect to different official NAT addresses of the PIX firewall to the port 53 (well -known port for the domain name service). The attacker used the source port 53 for this scan. Also some connection attempts to the demilitarised zone were detected from the same attacker and ports. All attempts were in the same second.
The PIX firewall blocked the inbound connections (outside -> inside) because of h aving no translation entry for this address. The connections to the DMZ were blocked because of the access-list, which deny these packets. The packets to the DMZ had the SYN and the FIN bit set and because of that I expect that also the packets to the NAT address space are packets with SYN and FIN bit set.

A „WHOIS“ at ARIN (American Registry for Internet Numbers) at the source address results the following:

Syracuse Supply (NETBLK-SYRACUSES-146-208)
294 Ainsley Dr
Syracuse,, NY 13205
US
Netname: SYRACUSES -146-208
Netblock: 12.3.146.208 - 12.3.146.223
Coordinator:
Franz, Pat (PF21-ARIN) [No mailbox]
315-476-9981
Record last updated on 28 -Apr-1998.
Database last updated on 5 -Sep-2001 23:16:33 EDT.

3.6 Correlations

The attack was detecte d by Curt Wilson. This is a known SYN/FIN scan to the domain name service. There are a lot of tools which can generate this pattern (SYN/FIN flag set, source port 53). One of the tools are hping and you can generate the scan with hping (version 2) for exam ple by: hping2 –s 53 –p 53 –S –F target.host

3.7 Evidence of active targeting

My opinion of this scan is that there is no evidence of active targeting in this network scan, because the attacker scanned a lot of (or all) addresses of the victim location. A lso the attacker doesn't know that there is no running domain name system on port 53 at this address space.

3.8 Severity

GIAC's approach to determining severity is to apply the following formula:
(Criticality + Lethality) – (System + Net Countermeasures) = Severity
Each metric is assigned on a five -point scale (1 as the lowest and 5 as the highest).

Criticality –DNS is a critical (in this case internal) service, without DNS normally most of the services can't be used, but it's not a core service like DH CP/BOOTP, therefore it will be 4.
Lethality – There are some DNS vulnerability known, but if a DNS server is well maintained most of the vulnerabilities will be fixed. Because of the frequency of published exploits the lethality will be 3.
System Counterm easure- The system countermeasure cannot be valued because we don't know if the system have installed services on port 53 because the firewall blocked the packets. So the system countermeasure is 3.
Network Countermeasure -..will be 5 because none of the sc an packets will pass the firewall.

(Criticality	+ Lethality)	– (System Countermeasure	+ Network Countermeasure)	= **Severity**
(4	+ 3)	– (3	+ 5)	= **-1**

3.9 Defensive recommendation

No system can be reached from the outside to the port 53 of the addresses because they are blocked by the PIX firewall. This is a very good defence method for this scan. The only recommendation is, that if one of these systems is a DNS server, it should be well mai ntained. Otherwise a firewall that forwards packets with source port 53 ("pretending to be from a DNS server") opens the possibility to gain access to the system.

3.10 Multiple choice test question

Why does the attacker of this packet use source port 53 ?
Jan 14 18:52:53 [firewall.ip.address] %PIX -6-106015: Deny TCP (no connection) from 12.3.146.212/53 to dmz.ip.addr.2/53 flags FIN SYN on interface outside

a. to crash the destination host
b. to bypass the firewall
c. to hide the scan (stealth scan)
d. makes no sense for this scan

Answer: b. to bypass the firewall

4. Phorum attack

data:

a. SNORT alerts

[2001-07-28 14:24:38] 195.3.96.70:22215 -> target.host:80 [arachNIDS/205] WEB-MISC Phorum admin access
[2001-07-28 14:24:38] 195.3.96.70:22215 -> target.host:80 [arachNIDS/205] WEB-MISC Phorum admin access
[2001-07-28 14:24:38] 195.3.96.70:4524 -> target.host:80 [arachNIDS/205] WEB-MISC Phorum admin access
[2001-07-28 14:24:38] 195.3.96.70:4524 -> target.host:80 [arachNIDS/205] WEB-MISC Phorum admin acces s
[2001-07-28 14:24:30] 195.3.96.70:13483 -> target.host:80 [arachNIDS/205] WEB-MISC Phorum admin access
[2001-07-28 14:24:30] 195.3.96.70:13483 -> target.host:80 [arachNIDS/205] WEB-MISC Phorum admin access

b. packet #1

Generated by ACID v0.9.6b12 on Sat September 08, 2001 12:39:24

#(2 - 39651) [2001-07-28 14:24:38] [arachNIDS/205] WEB-MISC Phorum admin access
IPv4: 195.3.96.70 -> target.host
hlen=5 TOS=16 dlen=621 ID=0 flags=0 offset=0 TTL=255 chksum=2806
TCP: port=22215 -> dport: 80 flags=***AP*** seq=3372272455
ack=3372272455 off=5 res=0 win=32120 urp=0 chksum=18306
Payload: length = 581

```
000 : 47 45 54 20 2F 61 64 6D 69 6E 2E 70 68 70 33 3F    GET /admin.php3?
010 : 6F 70 3D 61 64 6D 69 6E 4D 61 69 6E 20 48 54 54    op=adminMain HTT
020 : 50 2F 31 2E 30 0D 0A 41 63 63 65 70 74 3A 20 69    P/1.0..Accept: i
030 : 6D 61 67 65 2F 67 69 66 2C 20 69 6D 61 67 65 2F    mage/gif, image/
040 : 78 2D 78 62 69 74 6D 61 70 2C 20 69 6D 61 67 65    x-xbitmap, image
050 : 2F 6A 70 65 67 2C 20 69 6D 61 67 65 2F 70 6A 70    /jpeg, image/pjp
060 : 65 67 2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F    eg, application/
070 : 76 6E 64 2E 6D 73 2D 65 78 63 65 6C 2C 20 61 70    vnd.ms-excel, ap
080 : 70 6C 69 63 61 74 69 6F 6E 2F 6D 73 77 6F 72 64    plication/msword
090 : 2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76 6E    , application/vn
0a0 : 64 2E 6D 73 2D 70 6F 77 65 72 70 6F 69 6E 74 2C    d.ms-powerpoint,
0b0 : 20 2A 2F 2A 0D 0A 52 65 66 65 72 65 72 3A 20 68    */*..Referer: h
0c0 : 74 74 70 3A 2F 2F 6F 72 67 61 67 75 69 64 65 2E    ttp://orgaguide.
0d0 : 6C 61 6E 70 61 72 74 79 2E 61 74 2F 61 64 6D 69    lanparty.at/admi
0e0 : 6E 2E 70 68 70 33 0D 0A 41 63 63 65 70 74 2D 4C    n.php3..Acc ept-L
0f0 : 61 6E 67 75 61 67 65 3A 20 64 65 2D 61 74 0D 0A    anguage: de -at..
100 : 41 63 63 65 70 74 2D 45 6E 63 6F 64 69 6E 67 3A    Accept -Encoding:
110 : 20 67 7A 69 70 2C 20 64 65 66 6C 61 74 65 0D 0A    gzip, deflate..
120 : 55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 6F 7A 69    User-Agent: Mozi
130 : 6C 6C 61 2F 34 2E 30 20 28 63 6F 6D 70 61 74 69    lla/4.0 (compati
140 : 62 6C 65 3B 20 4D 53 49 45 20 35 2E 30 3B 20 57    ble; MSIE 5.0; W
```

```
150 : 69 6E 64 6F 77 73 20 39 38 3B 20 44 69 67 45 78   indows 98; DigEx
160 : 74 29 0D 0A 48 6F 73 74 3A 20 6F 72 67 61 67 75  t)..Host: orgagu
170 : 69  64 65 2E 6C 61 6E 70 61 72 74 79 2E 61 74 0D  ide.lanparty.at.
180 : 0A 43 6F 6F 6B 69 65 3A 20 61 64 6D 69 6E 3D 5A  .Cookie: admin=Z
190 : 6D 56 77 63 44 70 6B 64 57 31  74 65 51 25 33 44   mVwcDpkdW1teQ%3D
1a0 : 25 33 44 3B 20 75 73 65 72 3D 4D 6A 70 6D 5A 58  %3D; user=MjpmZX
1b0 : 42 77 4F 6E 70 71 56 45 78 4A 4E 46 4E 55 64 30   BwOnpqVExJNFNUd0
1c0 : 64 69 4D 47 38 36 4E 54 70 6D 62 47 46 30 4F 6A   diMG86NTpmbGF0Oj
1d0 : 45 36 4D 44 6F 77 4F 6A 45 36 52 47 56 6D 59 58   E6MDowOjE6RGVmYX
1e0 : 56 73 64 44 6F 30 4D 44 6B 32 0D 0A 43 61 63 68   VsdDo0MDk2..Cach
1f0 : 65 2D 43 6F 6E 74 72 6F 6C 3A 20 6D 61 78 2D 73   e-Control: max-s
200 : 74 61 6C 65 3D 30 0D 0A 58 2D 46 6  F 72 77 61 72  tale=0..X-Forwar
210 : 64 65 64 2D 46 6F 72 3A 20 36 32 2E 34 36 2E 34   ded-For: 62.46.4
220 : 30 2E 38 32 0D 0A 56 69 61 3A 20 31 2E 30 20 77   0.82..Via: 1.0 w
230 : 69 65 6E 31 70 72 6F 78 79 2E 61 6F 6E 2E 61 74   ien1proxy.aon.at
240 : 20 0D 0A 0D 0A                                    ....
```

4.1 Source of Trace

This scan was detected at our external IDS in our company.

4.2 Detect was generated by

The detect was generated by the intrusion detection system SNORT and the following rule triggered the log entries:

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB -MISC Phorum admin access"; flags: A+; uricontent:"/admin.php3"; nocase; reference:arachnids,205; classtype:attempted-recon; sid:1134; rev:1;)

explanation:

This is a default SNORT rule, only the IP addresses were changed.

$EXTERNAL_NET	any, except our internal networks
$HTTP_SERVERS	external Web-Servers (accessible from the Internet)
flags: A+	match ACK-flag
uricontent:"/admin.php3"	match if the „/admin.php3" pattern is in the URI content
nocase	pattern matching is case insensitive
reference:arachnids,205	there is a reference for this vulnerability (http://www.whitehats.com/info/ids205)
classtype: attempted-recon	type of packet is an additional reconnaissance
sid: 1134	SNORT rule ID is 1134
rev:1	rule revision is 1

4.3 Probability the source address was spoofed

It's not probable that the source IP is spoofed because the three -way handshake was succ essfully done (packets with ACK/PUSH flag set).Next the attacker wanted to get additional information whether the admin.php3 script exists, so he needed the reply from the server to get this information.

4.4 Description of attack

This attack is a known at tack against a Web -Server with the Phorum tool installed. This exploit is reported on arachNIDS (reference 5). Phorum is a web based discussion software. The attacker tried to find the admin.php3 script that is used for administrative tasks of the Phorum t ool. Though a vulnerability in version 3.0.7 and early (www.whitehats.com) exists, that the attacker can change the administrative password without any rights on the system. After this he can read all files on the system that the user running the http server has access to. With the ".." parent directory trick it's also possible that the intruder can read for example the /etc/passwd file. This tool only works on UNIX systems with the Phorum tool installed.

To exploit the server it must be a UNIX Web -Server (i.e. apache) with PHP3 -support and the tool Phorum installed. The attacker must establish a connection to the server, after that he can request the URL:
http://target.host/admin.php3?step=4&option=pass&confirm=testpwd&newPassword=testpwd

Target.host	the attacked system
Admin.php3	script for administrative tasks
?	separator of file and options
&	separator of different options
option=pass	to set password
newPassword=testpwd	set new admin password to "testpwd"
confirm=testpwd	confirm setting of password "testpwd"

After setting the password to his favourite password he can change a lot of administrative settings. (i.e. shutdown the Phorum tool, set a redirect URL, …)
One interesting part of the scan is the jumping source port because normally the source port increases by one for each new connection. There are two possibilities how this could be evoked. One possibility is that the attacker's system has a bad and non -standard TCP/IP stack implemen - tation, which chooses the source ports randomly. The other and more plausible possibility is that the attacker's scanning tool used random source ports for the att empts.

4.5 Attack mechanism

The attacker (195.3.96.70) attempts to start the admin.php3 script at port 80 of the server (associated to the web service). Admin.php3 is a script for administrative tasks for the Phorum tool. He tried to connect 6 times and the source port is, instead of increasing as usual by one, jumping from 22215 to 4524 to 13483. Every source port is used for two connection attempts.

A WHOIS search in the RIPE database for the attackers IP results following:
inetnum: 195.3.86.72 - 195.3.97.255
netname: TA-HIGHWAY
descr: Telekom Austria Aktiengesellschaft
country: AT
admin-c: HMH25-RIPE
tech-c: AAH12-RIPE
tech-c: DAH12-RIPE
tech-c: HMH25-RIPE
status: ASSIGNED PA
mnt-by: AS8447-MNT

mnt-lower: AS8447-MNT
changed: hostmaster@aon.at 20000310
changed: hostmaster@aon.at 20010129
source: RIPE

This are a pool for Dial-In Users at the ISP Telekom Austria.

4.6 Correlations

The attack was detected by me on our external IDS. It was triggered by a SNORT rule explained above. This was a known exploit and was published by arachNIDS (reference 5)

4.7 Evidence of active targeting

Indeed we have evidence of active targeting, because this is not a scan for this vulnerability but an exact and single attempt to this server, which is a UNIX server running apache with php3 -support. The three-way handshake is completed, too. But th e tool Phorum isn't installed on this system, so the exploit wasn't successful.

4.8 Severity

GIAC's approach to determining severity is to apply the following formula:
(Criticality + Lethality) – (System + Net Countermeasures) = Severity
Each metric is assigned on a five-point scale (1 as the lowest and 5 as the highest).

Criticality –This is a stand-alone web-server only for one customer and so if this system crashed or is hacked only the web appearance of one customer is disturbed, so the criticality of this system is 2. (I would set the criticality to 1 only for test systems)
Lethality – If this exploit is successfully done, the attacker can shut down the Phorum service and after reading and cracking the /etc/passwd he have accounts of this machine. And so everything even a Denial of Service attack can be started, so the lethality will be 5.
System Countermeasure - The system countermeasure is 4 because this is a good maintained web server without running the Phorum tool (which is a prerequisite).
Network Countermeasure -..will be 3 because this attack can't be filtered on the packet filters in front of this network. But the attacker can only access the web -port (80/tcp) and can't access to a virtual terminal after cracking the passwd -file because the pac ket filter block all other connections but to port 80 and 443 (SHTTP).

(Criticality	+	Lethality)	–	(System Countermeasure	+	Network Countermeasure)	=	**Severity**
(	2	+	5	) – (	4	+	3	) =	**0**

4.9 Defense recommendation

First a possibility is that we install a content filtering device on the firewall to secure the network and increase the network countermeasure by 1 or 2 points. Also I recommended that if the Phorum tool will be installed, the newe st version (which fix this exploit) should be used.

4.10 Multiple choice test question

How can you stop a PHP3 script attack against a web server?

a. With a packet filtering device
b. With a content filtering device
c. With a stateful inspection device
d. With a NAT firewall

Answer: b. With a content filtering device – only a content filter can block this attack.

5. whisker attack

data:

a. SNORT alerts

[2001-08-07 15:23:26] 192.109.27.168:3131 -> target.host:80 [arachNIDS/296] WEB -MISC whisker splice attack
[2001-08-07 15:23:27] 192.109.27.168:3130 -> target.host:80 [arachNIDS/296] WEB -MISC whisker splice attack
[2001-08-07 15:25:02] 192.109.27.168:3139 -> target.host:80 [arachNIDS/296] WEB -MISC whisker splice attack
[2001-08-07 15:25:02] 192.109.27.168:3140 -> target.host:80 [arachNIDS/296] WEB -MISC whisker splice attack
[2001-08-07 15:25:44] 192.109.27.168:3138 -> target.host:80 [arachNIDS/296] WEB -MISC whisker splice attack
[2001-08-07 15:28:22] 192.109.27.168:3138 -> target.host:80 [arachNIDS/296] WE B-MISC whisker splice attack
[2001-08-07 15:31:00] 192.109.27.168:3138 -> target.host:80 [arachNIDS/296] WEB -MISC whisker splice attack

b. packet #1

Generated by ACID v0.9.6b12 on Sat September 08, 2001 12:33:35
---------- ----------
#(3 - 237801) [2001-08-07 15:23:26] [arachNIDS/296] WEB -MISC whisker splice attack
IPv4: 192.109.27.168 -> target.host
hlen=5 TOS=0 dlen=41 ID=597 flags=0 offset=0 TTL=21 chksum=36577
TCP: port=3131 -> dport: 80 flags=***A**** seq=17825973
ack=1421780511 off=5 res=0 win=3072 urp=0 chksum=26124
Payload: length = 1
000 : 20

5.1 Source of Trace

This logs were collected on our company's external network.

5.2 Detect was generated by

The detect was generated by the SNORT Intrusion detection system and the entries was triggered by the following rule (this is a default SNORT rule):
alert tcp $EXTERNAL any -> $HTTP_SERVERS 80 (msg:"WEB -MISC whisker splice attack"; content: "|20|"; flags: A+; dsize: 1;referen ce:arachnids,296; classtype:attempted -recon; sid:1104; rev:1;)

5.3 Probability the source address was spoofed

The detected packet is a part of an established TCP connection. If a stateful inspection firewall is used and the target systems are not vulnera ble to sequence number prediction attacks I'm pretty certain that the source address of this attack isn't spoofed.

5.4 Description of attack

Whisker is a known CGI scanner with special options to obfuscate Intrusion Detection System without packet reasse mbling. Because this feature is turned on on our IDS SNORT we detect this splicing attack. This stealthy attack based on small tcp packets that will be reassembled at the target host.
The attacker tried to hide his web CGI requests and this is a warning si gnal to the analyst. But we don't see any other traffic few days before and after this alert from the attacker's source address to our network (this information is from our 7 -day full network dump) and so I think someone was just experimenting with the whi sker tool by using one of our web servers.

5.5 Attack mechanism

First the attacker established a TCP connection to port 80 (web service) that means that the three-way handshake was successfully done. The 'content: "|20|"' in the SNORT rule means that the content is a binary bytecode, so the binary content is: "0010 0000". The payload of the packet is 1 byte (dsize:1, that's a very unusual payload size), and so the packet was triggered by the IDS.

Source IP address: 192.109.27.168
Domain name: mpih-1-168.mpih-frankfurt.mpg.de

A WHOIS query on the RIPE home results:

```
inetnum:      192.109.27.0  - 192.109.27.255

netname:      MPIH-LAN

descr:        Max-Planck-Institut fuer Hirnforschung

descr:        Frankfurt/Main

country:      DE

admin-c:      WL21-RIPE
```

```
tech-c:         WL21-RIPE

rev-srv:        gwdu01 .GWDG.DE

rev-srv:        deneb.DFN.DE

rev-srv:        gwdu04.GWDG.DE

status:         ASSIGNED PI

mnt-by:         DFN-NTFY

changed:        cp@deins.Informatik.Uni  -Dortmund.DE 19920902

changed:        rv@ Informatik.Uni -Dortmund.DE 19930920

source:         RIPE
```

The Max-Planck institute is a university like location and maybe there are a lot of students who try security tools like whisker.

5.6 Correlations

The attack was detected by me on our external IDS. It was triggered by a SNORT rule explained above. Whisker is a known CGI script scanner and can be downloaded from the wiretrip homepage (reference 7) to check your own web servers or Intrusion Detection Systems.

5.7 Evidence of active targeting

Indeed we have evidence of active targeting because this attack is only to one of our servers (and this is a web server). The attacker doesn't scan the network with the whisker tool he only tried to find out if one specific server has CGI troubles. Next the splici ng option is set to bypass Intrusion Detection Systems. These are all indications of active targeting.

5.8 Severity

GIAC's approach to determining severity is to apply the following formula:
(Criticality + Lethality) – (System + Net Countermeasures) = S everity
Each metric is assigned on a five -point scale (1 as the lowest and 5 as the highest).

Criticality –This is a stand-alone web-server only for one customer and so if this system crashed or is hacked only the web appearance of one customer is disturb ed, so the criticality of this system is 2. (I would set the criticality to 1 only for test systems)

Lethality – If this exploit is successfully done and the attacker can run some forbidden cgi scripts, it depends on the possibilities of the CGI script, s o my rating for lethality will be 3.

System Countermeasure - The system countermeasure is 3 because this is a medium maintained web server with CGI scripts running.

Network Countermeasure -..will be 3 because this attack can't be filtered on the packet filte rs in front of this network. But the attacker can only access the web -port (80/tcp) and can't access to a virtual terminal after potentially getting a user/password combination because the packet filter block all other connections but to port 80 and 443 (S HTTP).

(Criticality	+ Lethality)	– (System Countermeasure	+ Network Countermeasure) =	**Severity**
(2	+ 3)	– (3	+ 3) =	**-1**

5.9 Defence recommendation

First a possibility is that we install a content filtering device on the firewall to secure the network and increase the network countermeasure by 1 or 2 points. Also the web server admin of this server should be informed that he should check his CGI scripts and search for possible changes on his system.

5.10 Multiple choice test question

What is whisker?

a. a programming language
b. a port scanner
c. a CGI scanner
d. a intrusion detection system

Assignment 2 - Describe the State of Intrusion Detection

Bypassing masquerading firewalls

1. Introduction

The first time I read about such a vulnerability to bypass a masquerading firewall was in April 2000. (www.securityfocus.om/bid/1078). The vulnerability was released at March 27, 2000 to securityfocus. On the website they use this vulnerability for obtainin g a tunnel from outside to the secured masqueraded network. But the real danger for my opinion doesn't get attention at this site. It's possible to start additional reconnassaince or to run attacks against some internal network services. The other thing of interest is, that a update to this bug was released on November 10, 2000. So it was still possible more than half a year later to obtain illegal access because of this vulnerability. My question is if it's still possible today to use this bug (in newer releases). In the next few pages I will describe how to use the masquerading vulnerability to get unallowed access to a secured network and I also will show some theoretical approaches to use UDP masquerading on "not vulnerable" systems for some dangerous p roblems. At the end I will give some methods and techniques to recognize and defend crafted packets for bypassing masquerading firewalls.

2. How masquerading works

There are a lot of masquerading description in the internet and I used about 10 to 15 sour ces for this topic and so this is a very short but technical description of masquerading
Masquerading is a special case of Network Address Translation (NAT). Masquerading firewalls uses a mixture of NAT and PAT (port address translation) to handle packets. Masquerading is a feature of firewalls and layer three gateways to change the IP address of packets to and from a local address. The next table and picture will show two packets on the way through the firewall.

1 original packet from internal system (1. 1.1.1, port 1076) to the destination IP and port of the external system (5.5.5.6, port 23)

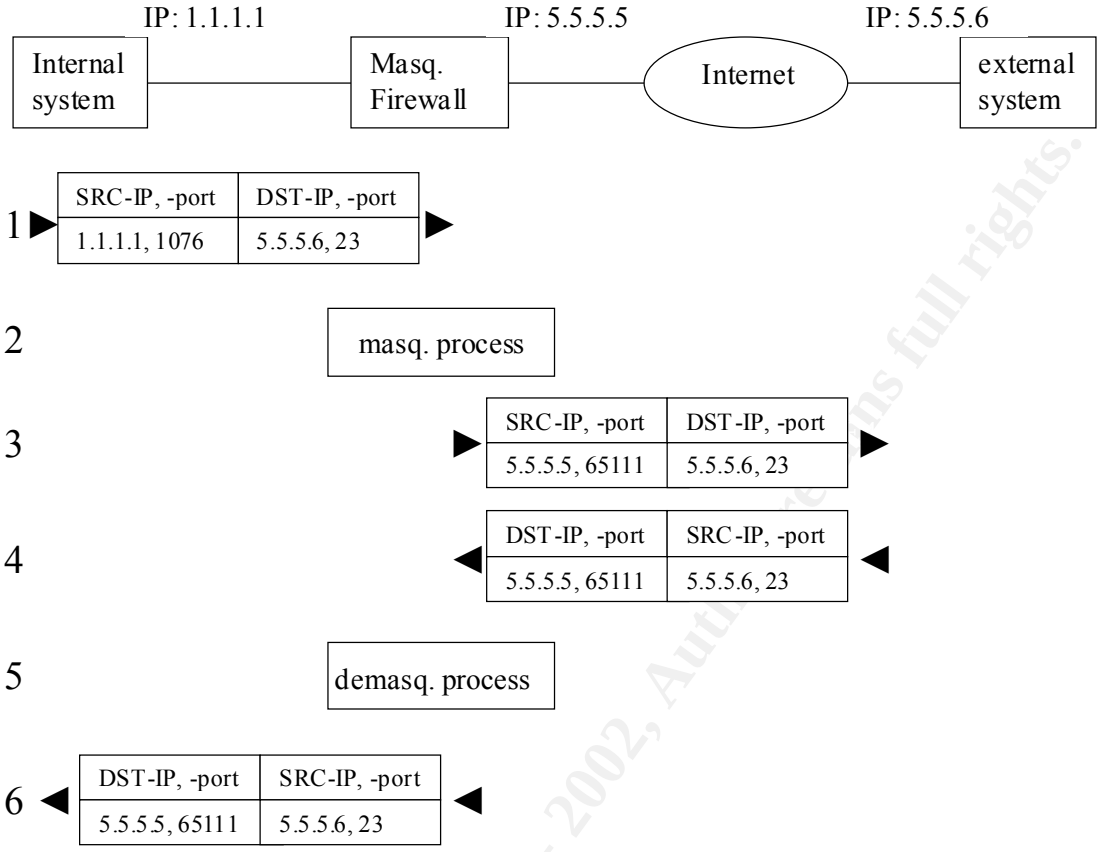
2 masquerading process (described below)

3 packet changed Source IP to outside firewall IP address (5.5.5.5) and a generic port (65111)

4 reply packet to the IP a ddress and port of the firewall (5.5.5.5, 65111) leaves external system

5 demasquerading process (described below)

6 demasqueraded packet: destination IP address and port changed to the internal system's IP address and original port (1.1.1.1, port 1076)



◄ ,► …… direction of packet flow

This show the principals of masquerading, now we look closer to the masquerading engine (step 2 and 5).

masquerading process:
When a packet comes to the firewall form the internal network it will first go through the input and forwarding rule of the firewall. Then the packet is changed in the masquerading process. In this process the source address of the IP packet is changed to the IP address of the outside interface of the firewall (or the interface, where the packets goes out, i.e. this could also be a demilitarized zone). Also the source port in the UDP or TCP field is changed to a gener ic port from 61000 to 65096 (per default). This port range is reserved in the Linux kernel or masquerading. But this can easily be changed by editing "linux -version/include/net/ip_masq.h". That means that a default Linux masquerading implementation can han dle a maximum of 4096 concurrent connections. After these changes of source IP address and port, Linux will create an entry in the masquerading table for this connection.

internal IP, port	masquer. port
1.1.1.1, 1076	65111
....	....

With Linux you can masquerade packets by inserting firewall rules in the following way: (in this example all TCP connection fro m the internal address 1.1.1.1 to the port 23, which means normally telnet service, will be masqueraded)

ipfwadm (2.0.x): ipfwadm -A -M -p tcp -S 1.1.1.1 -D 0/0 23

ipfwadm	firewalling tool
-A	append new rule
-M	masquerade packet after accept ing
-p tcp	TCP is the specified protocol
-S 1.1.1.1	accepted source address of internal network
-D 0/0 23	destination address: any, destination port:23

ipchains (2.2.x): ipchains -i forward -j MASQ -p tcp -s 1.1.1.1 -d 0/0 23

ipchains	firewalling tool
-i forward	insert rule in forward chain
-j MASQ	if rule matches jump to masquerading process
-p tcp	TCP is the specified protocol
-s 1.1.1.1	accepted source address of internal network
-d 0/0 23	destination address: any, destinat ion port:23

iptables (2.4.x): iptables –A FORWARD –t nat –to-source *fw.out.ip* –p tcp –s 1.1.1.1 –d 0/0 –dport 23 –j SNAT

iptables	firewalling tool
-A FORWARD	append to forward chain
-t nat	rule for table: nat
--to-source fw.out.ip	translate to outside firewall IP (masquerading)
-p tcp	TCP is the specified protocol
-s 1.1.1.1	accepted source address of internal network
-d 0/0	destination address: any
-dport 23	destination port: 23
-j SNAT	if rule matches jump to Source N AT process (masquerading)

Normally you can change the time how long a masquerading entry will exist in the table. In ipfwadm and ipchains there are 3 parameters to modify:
How long a masquerading entry exists after ....

1. ... a TCP FIN packet was sent (normally a very low value)
2. ... "normal" packet was sent (idle timout, default 15 minutes)
3. ... UDP packet matches the specified entry

After reaching one of the timeout values, the entry in the masquerading table will b deleted. After sending a new packet without a related masquerading entry, a new one will be created with a new random port (61000 -65096).

demasquerading process:
The demasquerading process works by changing the destination IP address and port of the incoming packet.
After receiving a packet the demasquerading process will look into the masquerading table, if a masqueraded port matches the destination port in the incoming packet. After this the destination IP address and port is changed to the internal machines IP address and UDP or TCP port. Exactly this behaviour (not checking for the source IP and port of the incoming packet) will make the attack possible.

3. The original vulnerability

The original attack is used for establishing a tunnel through a masquerading firewall from outside (reference 8).
Citation of bugtraq:

> *A serious vulnerability exists in the IP Masquerading code present in, but not necessarily limited to, the 2.2.x Linux kernel. Due to poor checking of connections in the kernel code, an attacker can potentially rewrite the UDP masquerading entries, making it possible for UDP packets to be routed back to the internal machine.*
> *The IP masquerading code only uses destination ports to determine if a packet from the external network is to be forwarded to the internal network. It then sets the remote host and port in its tables to the source address and port of the incoming packet. The attacker needs to determine the local port on the masq gateway to be able to rewrite the table with their own address and port. As the range of ports used to masquerade connections is small, from 61000 to 65096 for both UDP and TCP, it becomes fairly easy for an external host to determine the ports in use.*

As it is written in the text above it's possible that these attack technique functions as well on other Linux kernels and other masquerading firewalls.

Now I will show the original trace of the example shown at securityfocus:

```
ipchains -L -M -n on the masq gateway BEFORE the probes
> UDP 03:39.21 192.168.1.100 10.0.0.25 1035 (63767)   -> 53
[ tcpdump from attacker's machine ]
( we picked source port 12345 for our packets just so the trace would be
easier to follow)
[ snip -- this starts at port 61000 ]
10.0.0.1 > 10.10.187.13: icmp: 10.0.0.1 udp port 63762 unreachable [tos 0xd8] (ttl 245,    id
13135)
10.10.187.13.12345 > 10.0.0.1.63763: udp 0 (DF) [tos 0x18] (ttl 254, id 23069)
10.0.0.1 > 10.10.187.13: icmp: 10.0.0.1 udp port 63763 unreachable [tos 0xd8] (ttl 245, id
13136)
10.10.187.13.12345 > 10.0.0.1.63764: udp 0 (DF) [tos 0x18] (ttl 254,    id 23070)
10.0.0.1 > 10.10.187.13: icmp: 10.0.0.1 udp port 63764 unreachable [tos 0xd8] (ttl 245, id
13137)
10.10.187.13.12345 > 10.0.0.1.63765: udp 0 (DF) [tos 0x18] (ttl 254, id 23071)
10.0.0.1 > 10.10.187.13: icmp: 10.0.0.1 udp port 63765 unreachable [   tos 0xd8] (ttl 245, id
13138)
```

10.10.187.13.12345 > 10.0.0.1.63766: udp 0 (DF) [tos 0x18] (ttl 254, id 23074)
10.0.0.1 > 10.10.187.13: icmp: 10.0.0.1 udp port 63766 unreachable [tos 0xd8] (ttl 245, id 13139)
10.10.187.13.12345 > 10.0.0.1.63767: udp 0 (DF) [tos 0x18] (ttl 254, id 23083)

10.0.0.1 > 10.10.187.13: icmp: 10.0.0.1 udp port 63767 unreachable [tos 0xd8] (ttl 244, id 17205)
^^
The above packet's I D is substantially different, we may have found a masq'd connection !!!

10.10.187.13.12345 > 10.0.0.1.63768: udp 0 (DF) [tos 0x18] (ttl 254, id 23084)
10.0.0.1 > 10.10.187.13: icmp: 10.0.0.1 udp port 63768 unreachable [tos 0xd8] (ttl 245, id 13140)
10.10.187.13.12345 > 10.0.0.1.63769: udp 0 (DF) [tos 0x18] (ttl 254, id 23088)
10.0.0.1 > 10.10.187.13: icmp: 10.0.0.1 udp port 63769 unreachable [tos 0xd8] (ttl 245, id 13141)
10.10.187.13.12345 > 10.0.0.1.63770: udp 0 (DF) [tos 0x18] (ttl 254, id 23090)
10.0.0.1 > 10.10.187.13: icmp: 10.0.0.1 udp port 63770 unreachable [tos 0xd8] (ttl 245, id 13142)
10.10.187.13.12345 > 10.0.0.1.63771: udp 0 (DF) [tos 0x18] (ttl 254, id 23091)
10.0.0.1 > 10.10.187.13: icmp: 10.0.0.1 udp port 63771 unreachable [tos 0xd8] (ttl 2 45, id 13143)
10.10.187.13.12345 > 10.0.0.1.63771: udp 0 (DF) [tos 0x18] (ttl 254, id 23092)
10.0.0.1 > 10.10.187.13: icmp: 10.0.0.1 udp port 63772 unreachable [tos 0xd8] (ttl 245, id 13144)

[snip -- all the way to the upper end of our masq ports]
ipchains -L -M -n on the masq gateway AFTER the probes
> UDP 04:35.12 192.168.1.100 10.10.187.13 1035 (63767) -> 12345

Explanation of trace:
First the output of "ipchains –L –M –n shows the masquerading entries and there is a entry for the internal IP addre ss of 192.168.1.100 and source port of 1035, the masqueraded source port (after passing the firewall and getting the firewall IP address) is 63767. The destination host is 10.10.187.13 with the destination port of 12345. The attacker scans the UDP ports o f the firewall from 61000 to 65096. For each port checked he gets a UDP port unreachable message. All of them are from the firewall, except the one for port 63767 which is originated by the masqueraded host. Normally the UDP port unreachable packet's IP ID increases by one every time and the TTL (time to live) is normally set to 245, that means the firewall is 10 hops away. But the UDP port unreachable message for port 63767 is different, it has a completely different IP ID and a TTL of 244 (11 hops away), so we know that this packet couldn't be sent by the firewall but by the internal server 192.168.1.100.
After we have seen a packet like this, we look once again in the masquerading table with "ipchains –L –M –n". We will see that the destination host were changed to the attackers IP address 10.10.187.13 without a stimulus packet from the internal network. Now a channel from this attacking system to the internal system 192.168.1.100 to port 1035 exists.

4. Requirements

To be able to use the exploit as des cribed, a Linux firewall with one of the following distributions and/or kernels is needed:

Debian Linux 2.2pre potato
Debian Linux 2.2
Debian Linux 2.1
Linux kernel 2.2.14
    +Caldera eServer 2.3.1
    +Caldera eDesktop 2.4
Linux kernel 2.2.12
Linux kernel 2.2.10
    +Caldera OpenLinux 2.3
RedHat Linux 6.2 i386
RedHat Linux 6.1 sparc
RedHat Linux 6.1 i386
RedHat Linux 6.1 alpha
RedHat Linux 6.0 sparc
RedHat Linux 6.0 i386
RedHat Linux 6.0 alpha

The next thing which is required is a masquerading entry for the internal target. But in case of a server as the target of attack (DNS, NTP, ...) this requirement is easily fulfilled, and in most cases servers are the attackers primary targets. This vulnerability is tested with UDP masquerading entries. I think it is also possible to bypass the TCP masquerading process, but this would be much more difficult to demonstrate or exploit.
Next I also think that some other (not so popular) firewalls can be vulnerable in case of a similar behaviour.

5. Modification of attack to worst case scenario

To create a tunnel it's imaginable that the attacker send a Trojan horse to one of the internal clients (i.e. per E-mail). This Trojan may send packets to a popular external server with a well-known service from the victim with a high source port (grater than 1024 and the "listening" port of the Trojan horse, normal behaviour of IP stack). Because this is a normal and common connection, neither the firewall nor an Intrusion Detection system will detect this connection as a forbidden one. Now the attacker is able to connect to the victim by using the described vulnerability of obfuscating the masquerading process. So a connection from the Internet to the internal and "secured" network is possible and nobody would realize it because it seems that this is a normal UDP connection.

The bigger problem and more dangerous thing are internal UDP servers which a source port lower than 1024. For example a lot of domain name systems will connect via UDP from source port 53 to destination port 53. So the masquerading table is filled with the following entry: "*internal.server, 53, 63201* " where "*internal.server*" is the internal system, " *53*" is the source port of the internal server" and " *63201*" is the masqueraded source port on the firewall. If this server starts a request to an external DNS server (i.e. the Root Servers) it will create such an

entry in the masquerading table. After that an attacker can send DNS packets to the UDP port 63201 on the firewall's outside interfac e. This packet will be forwarded from the firewall to the internal.server on the destination port of 53. So it is possible that an outside attacker can resolve internal domain names. Also exploits could be started to the domain name server. Nowadays most of domain name servers use a high port for it's source port of requests, so this attack isn't possible to these servers (there is still a option to set source port to 53), but I'm sure that there are a lot of old implementations for DNS servers.

This is the same trace as shown in the original attack but the ports and addresses changed. We used a RedHat Linux system with kernel 2.2.14. The scan was generated by NMAP with a normal UDP scan: *nmap –sU –p 61000-65096 –P0 fw_out*

-sU	UDP scan
-p 61000-65096	scanned port range
-P0	don't ping destination host
fw_out	outside firewall interface

```
Fwall:~# ipchains -ML                          # on our masquerading firewall before
                                                 nmap scan was started
UDP  04:54.48 internal_DNS      external_DNS     53 (64180)   -> 53

[ tcpdump from attacker's machine ]
fw_out > attacker : icmp: fw_out udp port 61176 unreachable [tos 0xc0] (ttl 253, id 9364)
attacker.1073 > fw_out.61177: udp 0 (DF) [tos 0x18] (ttl 254, id 7096)
fw_out > attacker: icmp: fw_out udp port 61177 unreachable [to  s 0xc0] (ttl 253, id 9365)
attacker. 1073 > fw_out.61178: udp 0 (DF) [tos 0x18] (ttl 254, id 7099)
fw_out > attacker: icmp: fw_out udp port 61178 unreachable [tos 0xc0] (ttl 253, id 9366)
attacker. 1073 > fw_out.61179: udp 0 (DF) [tos 0x18] (ttl 254, id 7  101)
fw_out > attacker: icmp: fw_out udp port 61179 unreachable [tos 0xc0] (ttl 253, id 9368)
attacker. 1073 > fw_out. 61180: udp 0 (DF) [tos 0x18] (ttl 254, id 7102)
attacker. 1073 > fw_out.61181: udp 0 (DF) [tos 0x18] (ttl 254, id 7105)
fw_out > attacker : icmp: fw_out udp port 61181 unreachable [tos 0xc0] (ttl 253, id 9372)
attacker. 1073 > fw_out.61182: udp 0 (DF) [tos 0x18] (ttl 254, id 7116)
fw_out > attacker: icmp: fw_out udp port 61182 unreachable [tos 0xc0] (ttl 253, id 9373)
attacker. 1073 > fw_out .61183: udp 0 (DF) [tos 0x18] (ttl 254, id 7119)
fw_out > attacker: icmp: fw_out udp port 61183 unreachable [tos 0xc0] (ttl 253, id 9375)

Fwall:~# ipchains -ML                          # on our masquerading firewall after port
                                                 scan
UDP  04:52.55 internal_DNS      atta cker        53 (64180)  -> 53
```

Internal_DNS	this is our internal name server
External_DNS	this is our external name server
Attacker	attacking system outside the firewall
Fw_out	outside firewall interface

First we see the normal UDP masquerading entr y for the outgoing DNS request. Then I started the nmap UDP scan to the masquerading port range to the outside firewall IP. For all ports

except 61180 I get UDP port unreachable messages, that means that this port could be open (and indeed it is).

Another affected service is NTP (network time protocol) that is associated with the well -known UDP port 123. Most of NTP servers also use source port=destination port, that means that a connection from a NTP server are done from the local source port 123/UDP. So the "bypass masquerading firewall" vulnerability can also be targeted to NTP servers.

6. Detection and Defence

A good choice to reduce the possibility of an attack is to change the masquerading ports (61000 to 65096) to another range. Also you can check for port probes on the masquerading ports. If you change the range you can look at the default and the new port range, because in my mind an attacker will first test the original ports. In most cases after getting no response for the original probe he will stop checking to these vulnerability if he isn't really sure, that the system is a masquerading firewall. The exploit is only published for Linux firewalls but I recommend that all existing firewalls should be checked for the masquerading vulnerability, b ecause in my opinion this is one of the most dangerous bugs for masquerading firewalls if the system is vulnerable, because every experienced hacker could exploit this and then have access to internal systems. Another good defence is that you never use sou rce ports of known services (lower than 1024 and all special high port services). On Linux systems the value for source ports can be set in the at /proc/sys/net/ipv4/ip_local_port_range. In this file there are two values separated by a tab, the first value indicates the low end of the source port range, the other value indicates the high end of the source port range. This range can only be set for TCP and UDP simultaneously. So this values should be set to a range where no running service have it's default port. Another good defence possibility is that a stateful inspection firewall is being installed because the firewall can check if before a incoming UDP packet is received a outgoing one was passed.

That's a very good example how important egress filterin g (outgoing filters) can be. If you don't create and maintain outgoing rules you won't be aware of such attacks, furthermore outgoing port unreachable messages should be blocked because than the attacker don't know if a port is in listening state or closed

I've written a Snort rule for the bypassing firewall vulnerability for DNS, this rule should check if a incoming UDP DNS packet is a DNS request or a response. A request from outside to our internal nameserver should never occur.

```
alert udp !$INTERNAL any -> $INTERNAL 53 (msg:"DNS bypassing firewall attempt";
content: "|0*|"; regex; offset: 2; depth: 1; reference:cve,CVE -2000-0289;
reference:bugtraq,1078; classtype: successful-recon-largescale; rev:1;)
```

7. Conclusion

This is an older exploit (about one y ear old) , although I think it works on a lot of systems today, because I'm sure that many systems are still running Linux kernel 2.2.14 or lower. As noticed above there are a high possibility that also other systems are exploitable and so this one of few vulnerabilities that an attacker can send packets to the internal systems "directly" (not via other hacked systems in for example a demilitarised zone. After showing a lot of defence techniques it should be possible to stop bypassing the masquerading firew all.

Assignment 3 – "Analyze This" Scenario

GIAC Enterprises has provided us with one week of SNORT data from their network. The data was analysed using various tools and scripts. (Main tool was SnortSnarf from www.silicondefense.com). Some data are missing because of various troubles (full disk, power outages, etc. The resulting data and analysis are presented below:

1. Introduction

The data provided for this analysis was from alert logs generated by the Sno rt intrusion detection system. The analyses based on the file from July 23 rd ot July 29 th because in this week I've a lot of security incidents myself, so I decided to look what other networks „say" about this week.

Number of analysed Alerts: 1582163

Number of analysed Network Scans: 463444

Number of analysed Out of Spec packets: 1076, all are TCP packets
(data from July 27 th are missing)

Statistical analysis was used on these alerts, scans and OoS packets to identify:

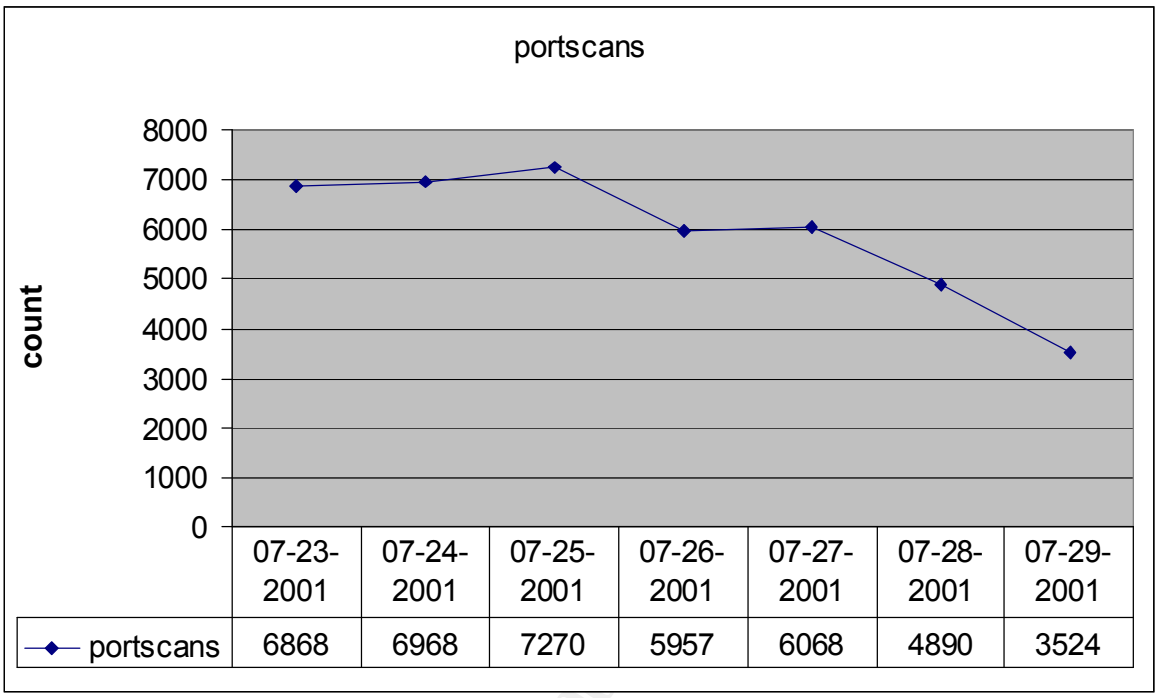
Most scans by source host
Most scans by destination host
Most scans by destination port .... done with scan log files and sometimes compared with data from alert and OOS logs.

Most crafted packets by source host .... done with OOS (Out of Spec) logs

All other analyses are based on resu lts of the above analyses (if a more exact research is necessary) or on the snort_stat.pl output of Appendix A.

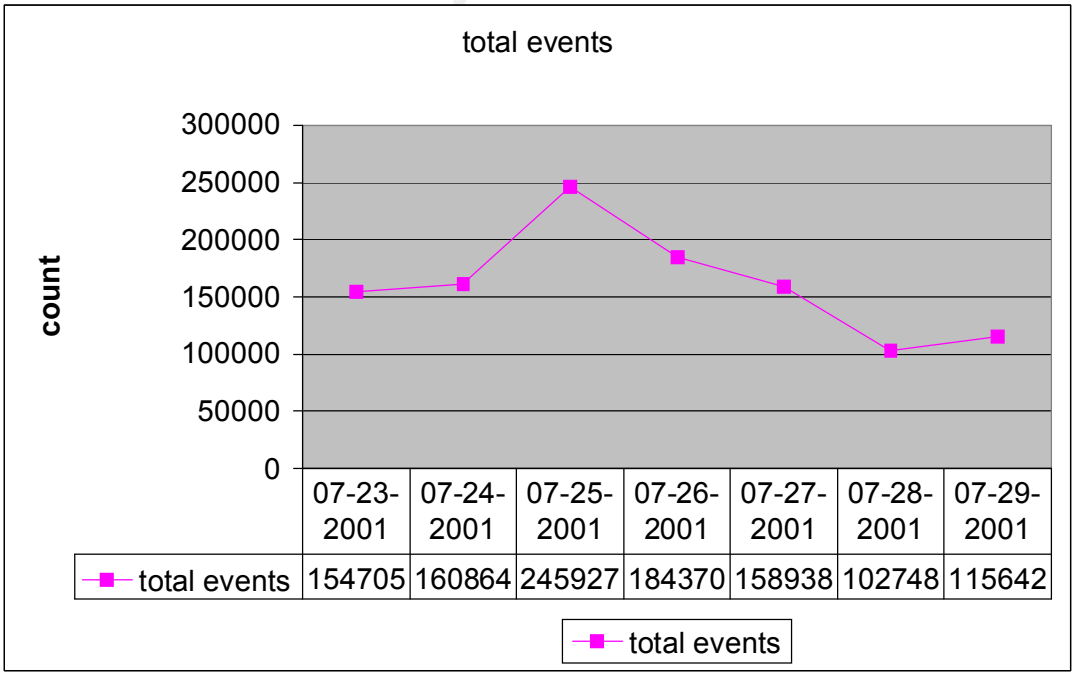
I always show as much results as needed for research, my default value are the top 10 counts.

2. Portscan statistics



In the graph above there are no special statistic anomalies detected. This is only the sum of all portscans, detailed analyses for portscans will be done later in this document.

3. statistical analysis on alert data

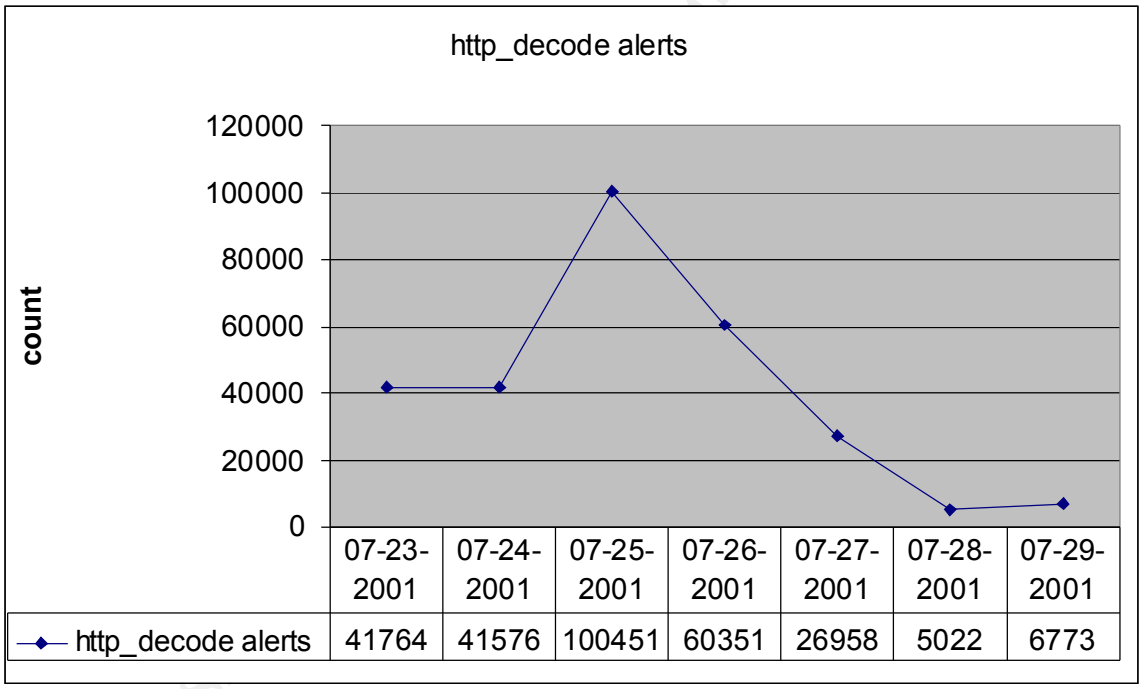


On July 25th, 2001 we have a increase of total events by about 60 percents, so this day should be analysed a little bit closer. So after analysing the snort_stat – data (appendix A) in depth I saw a huge amount of http_decode, the following tables presented this result:

distribution of attack methods on July 25 th, 2001:

%	count	attack method
28.80	70820	spp_http_decode
12.05	29631	spp_http_decode
9.77	24039	Possible trojan server activity
7.75	19070	MISC traceroute
7.04	17309	ICMP Destination Unreachable (Communication Administratively Prohibited)
6.03	14839	WEB-MISC prefix-get //
3.90	9603	INFO MSN IM Chat data
3.83	9424	ICMP Ec ho Request Nmap or HPING2
3.21	7886	SMB Name Wildcard
3.18	7824	ICMP Echo Request L3retriever Ping

So the http_decode data are taken to Excel to see a trend for this alerts:



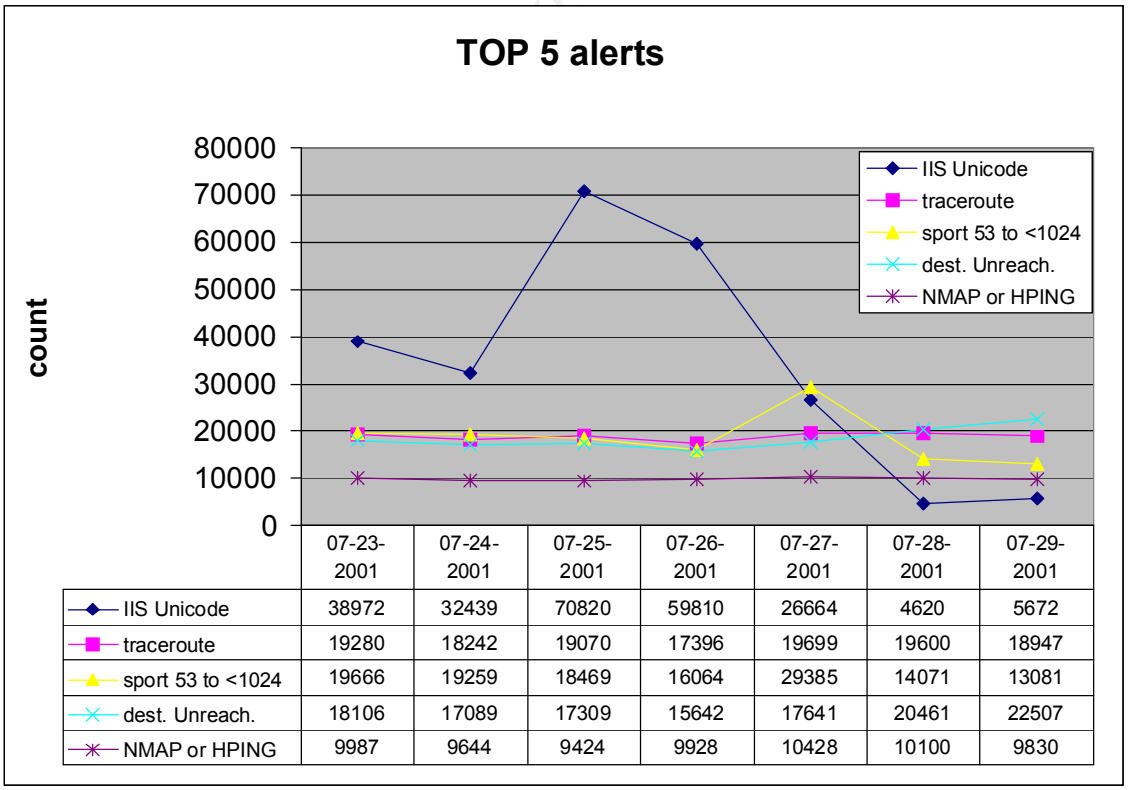
On July 25th, 2001 the amount of http_decode alerts increase by about 120 percent, so some resources were searched for similar activities on this date. (SANS home, incidents.org, securityfocus.com, caida.org, …). The possibility that the high increase of http ale rts is attributed to the spread of the famous code red worm, a exact analysis of this worm can be found on the CAIDA homepage (reference 21). This is a worm spread by himself and are distributed based on a Microsoft Internet Information Server vulnerabilit y.
The following table shows most occurred attacks by source and destination host to see which servers are probable infected.

# of attacks	from	to	method

24163	**10.10.75.150**	216.241.219.14	spp_http_decode: CGI Null Byte attack detected
10484	**10.10.157.108**	207.200.89.193	spp_http_decode: IIS Unicode attack detected
9177	10.10.162.228	204.152.190.70	ICMP Ech o Request Nmap or HPING2
7532	**10.10.157.108**	207.200.89.225	spp_http_decode: IIS Unicode attack detected
5664	**10.10.104.47**	207.200.86.66	spp_http_decode: IIS Unicode attack detected
5618	**10.10.85.74**	207.200.86.97	spp_http_decode: IIS Unicode attack detected
4426	**10.10.85.74**	207.200.86.66	spp_http_decode: IIS Unicode attack detected
4401	**10.10.104.47**	207.200.86.97	spp_http_decode: IIS Unicode attack detected
4291	10.10.16.5	10.10.162.228	ICMP Destination Unreachable (Communication Administratively Prohibited)
4236	216.150.152.145	10.10.5.44	SMB Name Wildcard

All http_decode entries in the top 10 table above are starte d from internal systems, this is an indication that internal servers are infected by CodeRed. It's strongly recommended that the highlighted systems are checked immediately. Also the Incidents Response Team should be alarmed because internal systems are m ost likely infected an this indicates an critical security vulnerability in the companies network.
The other statistical data from snort_stat.pl don't show gross anomalies and so only the top 5 alarms of this week are analysed a little bit more.

4. TOP 5 alerts



	07-23-2001	07-24-2001	07-25-2001	07-26-2001	07-27-2001	07-28-2001	07-29-2001
IIS Unicode	38972	32439	70820	59810	26664	4620	5672
traceroute	19280	18242	19070	17396	19699	19600	18947
sport 53 to <1024	19666	19259	18469	16064	29385	14071	13081
dest. Unreach.	18106	17089	17309	15642	17641	20461	22507
NMAP or HPING	9987	9644	9424	9928	10428	10100	9830

4.1 IIS Unicode attack detected

(reference 25)
There are some vulnerabilities in the Microsoft Internet Information Server based to craft the URL with UNICODE characters. Then resources which normally be inaccessi ble can be accessed. A lot of these exploits are fixed in the IIS code, nevertheless exploits being published ever and anon. Because of this the IIS servers should be checked for intrusions more than other services and be updated as soon as security patche s exists. Normally this is a noisy alert (which happens often) but in this analysis it occurs more often than the noise because of the Code Red worm in these days (shown above).

4.2 ICMP traceroute

Traceroute is a tool to show the route to the specified h ost, by sending packets with increasing Time-To-Live by one every packet sent (beginning by 1). The first gateway gets the packet, decrease Time-To-Live by one to zero then the gateway send back a TTL exceeded to the origin. The second packet will be repli ed by the second gateway, and so on ….
The UNIX traceroute implementation works by sending UDP packets to the destination, the Windows implementation uses ICMP packets, so this traceroute is most likely originated from a Windows system. Probability that th ese are false positives are minimal, because packets normally don't have a TTL of 1, but this traffic is normally not hostile traffic.
The source address can be easily spoofed, because ICMP is a connectionless protocol, but I challenge if source IP spoofin g makes sense in this case, the only reason could be to fill up the IDS logs to hide some attacks.

4.3 source port 53 to <1024

arachNIDS (reference 22)
This alert indicates that one is making a connection to a privileged port using the source port 53. Some years ago this was normal to DNS servers which start requests from port 53 to port 53, also nowadays server can be justified to use destination port 53. Today this is used to bypass packet filters, because some old packet filters let incoming UDP packets from port 53 pass because of having no connection state table. This traffic should be analysed more closer to see if this is a stimulus or a response. If it is a stimulus the IP address should be checked if the origin is an old DNS server or some attacker .
It's recommended that stateful inspection filters are installed to block such packets.
The source address can be easily spoofed, because this event was caused by a TCP packet, but the packet isn't thought to be a part of an existing connection. So exploi t packets can be sent to victims without getting the attacker's identity.

4.4 ICMP Destination Unreachable (Communication Administratively Prohibited)

This is a ICMP message of type 3 (Destination Unreachable Message) and code 13 (prohibited communication). If a packet is sent to a destination and the destination doesn't prohibit this packet some systems send a ICMP message back to the originator. The problem of this message is that the destination host send back information of prohibited services and coul d be used for backward host scan (look for denied services). Normally this messages should be blocked by the egress filters (outgoing filters) to stop sending information back to the origin.

4.5 ICMP Echo Request Nmap or HPING2

arachnids (reference 23)
Normally ICMP echo requests are used for mapping networks to get additional information of destination networks. Many implementations of the PING program (which uses ICMP echo requests and replies) create unique echo -request packets so these particular aler ts were associated with the pattern of the HPING2 or NMAP ping. These are two tools to scan networks by permitting to set a lot of protocol parameters by command line options. But the special pattern of the packet can be easily self -maded by network progra mmers and so these packets could also originated by some unknown tools.
The source address can be easily spoofed, because ICMP is a connectionless protocol, but I challenge if source IP spoofing makes sense in this case, the only reason could be to fill up the IDS logs to hide some attacks.

5. Most scans by source host

The following table shows the count of scans per source host, this is to detect possible attackers who constantly scan your systems. (used script: scan.sourcehost.pl)

Count	source host
171812	10.10.160.114
40999	205.188.233.153
35962	205.188.233.185
27210	205.188.246.121
24343	209.11.66.3
16697	10.10.217.114
15871	205.150.237.250
14311	205.188.244.121
12170	205.188.244.249
11540	10.10.70.242
10917	213.51.204.55
10669	210.15.29.17

The top source host is an internal system, it is recommended to check this system immediately, because either their is a busy employee who scan different networks or the system had been compromised and is now used as a “anonymizer” (attacker scan from a source host other then his own). After I saw this I decide to write a script to check the destination addresses of the scans from 10.10.160.114.

5.1 more exact analyses to scanning source host 10.10.160.114

Now I want to get more information of network scans from the internal system 10.10.160.114. First I want to show which hosts are the most targeted system.
(used script: scans.dadr.sourcehost.pl)

Count	destination scanned by 10.10.160.114
14058	206.206.48.64
10821	66.66.130.148
10221	166.84.159.101
8959	66.92.70.235
6754	24.30.5.24
6731	24.43.12.34
4836	24.19.234.22
4161	64.108.11.238
2482	64.91.29.66
2340	63.62.0.96
2296	24.23.61.159
2238	24.22.167.43
2004	24.101.53.229

Because there is no decisive indicator in the analyse above another analyse is started to check a ll destination ports scanned by 10.10.160.114.

Count	destination ports scanned by 10.10.160.114
124637	27005
8945	27500
1594	27243
1513	1027
976	1051

Top 5 destination ports are enough because we can yet see what the problem is. Most of the connections go to port 27005 which looks like a Trojan horse. The problem is that there are no known Trojan associated with this port (also port 27500 and 27243 are not known, reference 1) so it is possible that this is a new Trojan **. Now I strongly recommended that the system should be checked in depth.** If a integrity checker is installed the logs on this checker should be scanned carefully. The Incidents Response Team should be alarmed to handle this case. The team should consist in at least one security expert and one system administrator for 10.10.160.114.

Commentary after complete analyse: As shown some section below this is probably only a online game (Half-Life), for more details jump to section 4.2! Before alarming the Incident Response Team read defence recommendation.

6. Most scans by destination host

The following table shows the count of scans per destination host, this is to detect possible victim systems which are constantly scanned.
Used script: scans.targethost.pl

Count	destination host
14058	206.206.48.64
10821	66.66.130.148
10221	166.84.159.101
9940	10.10.110.33
9458	10.10.109.62
8959	66.92.70.235
8374	10.10.178.222
8285	10.10.110.169
7503	10.10.145.166
7418	10.10.178.188

There are no statistical anomalies detected by this analyses.

7. Most scans by destination port

The following table shows the count of scans per destination port, the usage of this analyse is for detecting increased scanning of a special service/Trojan.
Used script: scans.targetport.pl

Count	destionation port
134843	6970
124647	27005
30412	21
29903	23
29795	53
10169	6699
8950	27500
7684	13139
7132	1214
6535	4665

As seen above there are two destination ports were many more scans occurred than by the other ones. About half of the scans are targeted to these two ports, that is an indication to analyse this in depth. First I want to check associations with the two ports.

7.1 more exact analyses to scanned destination port 6970

The port 6970 is a high port which is associated to the GateCrasher Trojan horse. This is a Trojan based on Windows systems and is spread by a Macro in a "Microsoft Word97 documents". (Macro started as soon as the victim opens the document) A more specific description of this Trojan is given by the "Privacy Software Corporation" (reference 15).

To download the GateCrasher software for security researches go to reference 16 (handle with care). Then you can check your network security devices and your IDS to handle this detect and block GateCrasher attempts.

Now we should look which systems are scanned by this Trojan, so I want to create a table how often one target host are scanned on this port.
Used script: scans.targethost.target port6970.pl

Count	destination hosts scanned on port 6970
9940	10.10.110.33
9455	10.10.109.62
8327	10.10.178.222
8244	10.10.110.169
7501	10.10.145.166
7416	10.10.178.188
7389	10.10.180.76
7307	10.10.108.15
7253	10.10.130.132
6534	10.10.108.13

There could be some reasons, one is that a lot of systems are infected by this Trojan and so no statistical anomalies are detected. The other possibility is a new invasion of this Trojan horse so we should look closer to the source hosts scanning to port 6970.
After getting the result (with the script scans.sourcehost.targettport6970.pl) we can realize that all source hosts are from external systems (look at the list below), and most of scans come from one /16-range (exactly 3 class C networks). So this is indeed an indication of deeper research.

Count	source hosts scanning destination port 6970
40571	205.188.233.153
35952	205.188.233.185
26321	205.188.246.121
14167	205.188.244.121
12042	205.188.244.249
5789	205.188.233.121
1	213.67.144.6

The WHOIS query at ARIN (American Registry for Internet Numbers) results:
America Online, Inc (NETBLK-AOL-DTC)
22080 Pacific Blvd
Sterling, VA 20166
US

Netname: AOL-DTC
Netblock: 205.188.0.0 - 205.188.255.255

Coordinator:
America Online, Inc. (AOL-NOC-ARIN) domains@AOL.NET
703-265-4670

Domain System inverse mapping provided by:

DNS-01.NS.AOL.COM 152.163.159.232
DNS-02.NS.AOL.COM 205.188.157.232

Record last updated on 27 -Apr-1998.
Database last updated on 7 -Sep-2001 23:28:25 EDT.

All scans are launched from IP addresses in the AOL address space. This could be some dial -in users who tried to spread the GateCrasher Trojan or it is possible that some American Online systems are infected with this Trojan.

Correlation:
GateCrasher attempts are also logged on March 11 th, 2000 (reference 17). Another alert was published by Arrigo Triulzi on May 28 th, 2000 (reference 18), but he identified the attempt as a false positive because of a NAT firewall. Neither on whitehats.com nor on inc idents.org I found more correlations to GateCrasher attempts (I also searched our own SNORT logs).

Defence recommendation:
Citation of reference 15 – Privacy Software Corporation Security Advisory:
The GateCrasher server will install its program in the re gistry under the HKEY_LOCAL_MACHINE \SOFTWARE\Microsoft\Windows\CurrentVersion\Run key with a string value of „Explore" with data pointing to the file which was installed, which will usually be Explore.exe.
It is necessary to remove the registry subkey fir st. It will not be possible to remove the program file while the server is running and you may also be prevented from shutting down the computer. A reboot will be required in order to restart the machine without the GateCrasher" server being reloaded at wh ich time the file pointed to in the registry can be removed without further risk.
As a result, care should be taken to back up your registry first as well as your programs and files in the event that removal of the registry entry results in damage to your system. Use of Privacy Software Corporation's BOClean 3.01 program will safeguard against this possibility by removing the program and its registry entries automatically without risk of damage, or the nee d to disconnect the infected machine or reboot.

Next the SNORT ruleset should be updated because of non -alarming this in the alert logs, so a rule for possible GateCrasher Trojan access (filter on destination port 6969 and 6970) should be installed.

7.2 more exact analyses to scanned destination 27005

As shown above the count of scanned port 27005 are 124647, this is many more than other scanned ports, I realized this again in section 2.1, and so I do some additional analysis to that port. The only known program associated to this port is Half Life (Game) and so I decided to check which internal users use this.
We need two tables, one for the scanned target hosts and one for the scanning source hosts.

Used script: scans.sourcehost.dport27005.pl

Count	source hosts scanning destination port 27005
124637	10.10.160.114
6	10.10.160.169
3	10.10.98.195
1	10.10.97.242

Used script: scans.targethost.dport27005.pl

Count	destination hosts scanned on port 27005
14017	206.206.48.64
10808	66.66.130.148
10133	166.84.159.101
6750	24.30.5.24
6661	24.43.12.34
4830	24.19.234.22
4157	64.108.11.238
2481	64.91.29.66
2339	63.62.0.96
2239	24.23.61.159

After doing the correlations to this port there was no other reference to this port than the one from Lars Hansen (reference 19), so it's recommended to check the internal systems if anyone is playing Half-Life from the internal network. Also incidents.org associate this port (27005) to Half-Life. (reference 20)

Defense Recommendation:
If Half-Life can be found on the four source hosts, only a deinstallation of this game (to decrease the amount of network traffic) is recommended, if Half-Life isn't detected on the systems, it is strongly recommended that the Incident Response Team is called, because it's possible that this is a new trojan. Then new analyses based on port 27005 must be started.

Correlation:
The scan to port 27005 correlates to the previous analyses for host 10.10.160.114. The Global Incidents Analyses Center has a lot of analysed detects to port 27005 on their pages:

03/22/2000 Handler on Duty: Jeff Stutzman
03/18/2000 Handler on Duty: Jeff Stutzman
05/28/2000 by Andy Johnston

On February 26th, 2001 Lars Hansen wrote that port 27005 is a default port for Half-Life (Game), so before starting new and deeper analyses the systems should be checked for this game.

8. TOP 5 crafted packet origins

Used script: **oos.sourcehost.pl**

Count	source host
231	199.183.24.194
185	141.157.88.27
91	198.186.202.147
51	62.41.32.27
49	193.231.15.194

8.1 "199.183.24.194"

domain name: vger.kernel.org
WHOIS query:

Red Hat Software (NET-REDHAT)
P.O. Box 4325
Chapel Hill, NC 27515
US
Netname: REDHAT
Netblock: 199.183.24.0 - 199.183.24.255
Coordinator:
Taylor, Stacy (ST452-ARIN) abuse@icgcom.com
408-579-5000
Record last updated on 01 -Mar-2001.
Database last updated on 7 -Sep-2001 23:28:25 EDT.

Correlations:

Another website could be found which logs Out of Specs data for this IP address, the data was from the SnortSnarf alert page of nothing-on.tv, they log data from this source on July 26th and 27th, 2001 – exactly in the same timeframe of our logs (data of July 27 th are missing in Out of Spec logs). It's possible that the server on kernel.org was being compromised these days . We should inform the admin of this server on kernel.org (normally a trustable organization) to check the systems for signs of compromise.

8.2 "141.157.88.27"

domain name: pool-141-157-88-27.balt.east.verizon.net
WHOIS query:

Bell Atlantic (NETBLK-BELL-ATLANTIC)
1880 Campus Commons Drive
Reston, VA 20191
US
Netname: BELL-ATLANTIC
Netblock: 141.149.0.0 - 141.158.255.255
Maintainer: BAIS
Coordinator:
Verizon Global Networks Inc. (ZV20-ARIN) noc@gnilink.net
(703) 295-4583
Domain System inverse mapping provided by:
NSDC.BA-DSG.NET 199.45.45.14
GTEPH.BA-DSG.NET 141.151.0.68
Record last updated on 19 -Jul-2001.
Database last updated on 7 -Sep-2001 23:28:25 EDT.

No correlations for this IP address was found.

8.3 “198.186.202.147”

domain name: panoramix.v alinux.com
WHOIS query:

Dandelion Digital (NETBLK -DANDELION -C)
930 Tahoe Blvd. #802 -546
Incline Village, NV 89451
US
Netname: NETBLK -DANDELION -C
Netblock: 198.186.200.0 - 198.186.203.255
Coordinator:
Zubkoff, Leonard N. (LNZ-ARIN) lnz@DANDELION.COM
775.832.1068
Domain System inverse mapping provided by:
NS1.VALINUX.COM 198.186.202.135
NS2.VALINUX.COM 198.186.202.136
Record last updated on 18 -Jul-2001.
Database last updated on 7 -Sep-2001 23:28:25 EDT.

No correlations for this IP address was found .

8.4 “62.41.32.27”

domain name: proxy.sharif.ac.ir
WHOIS query:

inetnum: **62.41.32.0 - 62.41.32.255**

netname: CYBERROUTE

descr: Cyberroute ASP

country: NL

admin-c: AP8093 -RIPE

tech-c: AP8093 -RIPE

status: ASSIGNED PA

notify: pols@cyberroute.com

mnt-by: RIPE-NCC-NONE -MNT

changed: beri@EU.net 20000612

source: RIPE

route: **62.41.0.0/16**

descr: KPNQwest

origin: AS286

mnt-by: AS286-MNT

changed: beri@EU.net 20000529

source: RIPE

No correlations for this IP address was found.

8.5 "193.231.15.194"

domain name: hal.cs.tuiasi.ro
WHOIS query:

inetnum: 193.231.15.0 - 193.231.15.255

netname: UTI-NET

descr: "Gh. A sachi" Technical University of Iasi - ROMANIA

country: RO

admin-c: BC4-RIPE

tech-c: FA27-RIPE

tech-c: CC28-RIPE

status: ASSIGNED PA

notify: domain-admin@rnc.ro

mnt-by: AS3233-MNT

changed: irina@u1.ici.ro 19960405

changed: estaicut@rnc.ro 19981123

changed: cristih@rnc.ro 20010215

source: RIPE

route: 193.231.0.0/19

descr: Romanian Education Network

origin: AS2614

mnt-by: PUB-MNT

changed: george@roedu.net 20000923

source: RIPE

No correlations for this IP address was found.

9. Analysis process:

The analyses was done on a 800MHz CPU with 256MB RAM and 512 MB Swap space, the hard drive has about 55GB of free space.
First I downloaded the sources from www.research.umbc.edu/~andy (resource given by SANS). This include alerts, scans and OOS files.

9.1 general process

After I changed the pattern "MY.NET" to "10.10" because of readability to SnortSnarf. I do this with the following script:

```
#!/bin/bash
for file in `ls ./*`
do
cat $file | sed 's/MY.NET/10.10/g' >./$file.vers1
done
```

(modified script from practical assignment of Paul Asadoorian)

This was verified by: *grep „MY.NET" *.vers1* and there was no output, so there is no „MY.NET" in the log file s any more.

Next I combined the files into one

*ls –l *.vers1 | awk ‚{print „cat „$9 „ >> merged.alerts"}' | sh*
(The above code was present in many previous practicals.)

Then I sorted it by date and time

cat merged.alerts |sort > sorted.merged.alerts

after this I deleted the header lines with VI and do all previous tasks with the scan log files.

I tried to start SnortSnarf for the merged file, but my system ran out of memory and the process was killed. Then I tried SnortSnarf with the single day logs, but the system ran out of memory again. So I wrote some scripts to get the desired tables for port scans (modification of a self - written DENY-Log-Analyser).

After a lot of tries with different log analysers I used **snort_stat.pl** to generated analyses bas ed on daily logs and correlate the data by hand. This perl -script could be downloaded from the SNORT website. (reference 27)

*cat ./alert.0107**.B.gz.vers1 |./snort_stat.pl > alerts -0107**.stat*

"**" was substituted by date

Then all relevant data are bein g cut out, only the top 10 of every statistical overview were saved. (data in appendix A). Then I took some data to Excel to look for some trends respectively some anomalies. All Excel -created data are described in the related chapters.

9.2 Scripts for network scan logs

Most scans by source host
Scriptname: **scan.sourcehost.pl**
Activation: ./scans.sourcehost.pl |sort –r –k1 > result.scan.sourcehost
(the sort command is used to sort the count downward)

```
#!/usr/bin/perl –w

use strict;
use FileHandle;
my $inplog=”scans/sorted.merged.scans”;                    #input file
my ($sadr);                                                #declaration of some variables
my $sum;
my (%user,@usert,@keys,$key);
# Jul 23 23:44:34 10.10.160.114:777   -> 172.172.124.105:3046 UDP   #sample scan log line
open(LOGREAD,”$inplog”) or die “open   nicht ok”;
while (<LOGREAD>) {
     ($sadr) = /.* \:\d\d\s(.+?) \:.+? \s->\s.+? \:.+? \s.+? \s.*/i;   #regular expression to filter out diff.   Sources
     $user{“$sadr”}+=1;
}
foreach $key (sort(keys %user)) {
$_=$key;
($sadr)=/(.+)/;
printf “%10s % -16s \n”,$user{$key},$sadr;
$sum=$sum+$user{$key};
}
printf $sum;
close LOGREAD;
exit;
```

This script is a little bit long for this use, because this is a modified script of a DENY -Log analyser, which is a little bit more complex, but so it can easy be modified f or my other analyses.

Additional research scripts for 10.10.160.114:

Script name: **scans.dadr.sourcehost.pl**
Activation: ./scans.dadr.sourcehost.pl |sort –r –k1 |less

```
#!/usr/bin/perl –w

use strict;
use FileHandle;
my $inplog=”scans/sorted.merged.scans”;
#my $inplog=”test”;
my ($dadr);
my $sum;
my (%user,@usert,@keys,$key);
# Jul 23 23:44:34 10.10.160.114:777   -> 172.172.124.105:3046 UDP

open(LOGREAD,”$inplog”) or die “open nicht ok”;
while (<LOGREAD>) {
   if (/.*\:\d\d\s10\.10\.160\.114\:.+?\s->\s.+?\:.+?\s.+?\s.*/i) {
     ($dadr) = /.* \:\d\d\s.+?\:.+?\s->\s(.+?)\:.+?\s.+?\s.*/i;
     $user{“$dadr”}+=1;
   }
```

```
}
foreach $key (sort(keys %user)) {
$_=$key;
($dadr)=/(.+)/;
printf "%10s % -16s\n",$user{$key},$dadr;
$sum=$sum+$user{$key};
}
printf $sum;
close LOGREAD;
exit;
```

script name: **scans.dport.sourcehost.pl**
activation: ./scans.dport.sourcehost.pl |sort –r –k1 |less

```
#!/usr/bin/perl –w

use strict;
use FileHandle;
my $inplog="scans/sorted.merged.scans";
#my $inplog="test";
my ($dport);
my $sum;
my (%user,@usert,@keys,$key);
# Jul 23 23:44:34 10.10.160.114:777   -> 172.172.124.105:3046 UDP

open(LOGREAD,"$inplog") or die "open nicht ok";
while (<LOGREAD>) {
   if (/.*\:\d\d\s10\.10\.160\.114\:.+?\s->\s.+?\:.+?\s.+?\s.*/i) {
      ($dport) = /.* \:\d\d\s.+?\:.+?\s->\s.+?\☹.+?)\s.+?\s.*/i;
      $user{"$dport"}+=1;
   }
}
foreach $key (sort(keys %user)) {
$_=$key;
($dport)=/(.+)/;
printf "%10s % -16s\n",$user{$key},$dport;
$sum=$sum+$user{$key};
}
printf $sum;
close LOGREAD;
exit;
```

script name: **scans.targethost.pl**
activation: ./scans.targethost.pl |sort –r –k1 |less

```
#!/usr/bin/perl –w

use strict;
use FileHandle;
my $inplog="scans/sorted.merged.scans";
#my $inplog="test";
my ($dadr);
my $sum;
my (%user,@usert,@keys,$key);
# Jul 23 23:44:34 10.10.160.  114:777  -> 172.172.124.105:3046 UDP

open(LOGREAD,"$inplog") or die "open nicht ok";
while (<LOGREAD>) {
```

```
    ($dadr) = /.* \:\d\d\s.+?\:.+?\s->\s(.+?)\:.+?\s.+?\s.*/i;
    $user{"$dadr"}+=1;
}
foreach $key (sort(keys %user)) {
$_=$key;
($dadr)=/(.+)/;
printf "%10s % -16s\n",$user{$key},$dadr;
$sum=$sum+$user{$key};
}
printf $sum;
close LOGREAD;
exit;
```

script name: **scans.targetport.pl**
activation: ./scans.targetport.pl |sort –r –k1 |less

```
#!/usr/bin/perl –w

use strict;
use FileHandle;
my $inplog="s cans/sorted.merged.scans";
#my $inplog="test";
my ($dport);
my $sum;
my (%user,@usert,@keys,$key);
# Jul 23 23:44:34 10.10.160.114:777   -> 172.172.124.105:3046 UDP

open(LOGREAD,"$inplog") or die "open nicht ok";
while (<LOGREAD>) {
    ($dport) = /.* \:\d\d\s.+?\:.+?\s->\s.+?\☹.+?)\s.+?\s.*/i;
    $user{"$dport"}+=1;
}
foreach $key (sort(keys %user)) {
$_=$key;
($dport)=/(.+)/;
printf "%10s % -16s\n",$user{$key},$dport;
$sum=$sum+$user{$key};
}
printf $sum;
close LOGREAD;
exit;
```

script name: **scans.targethost.dport6970.pl**
Activation: ./scans.targethost.dport6970.pl |sort –r –k1 |less

```
#!/usr/bin/perl –w

use strict;
use FileHandle;
my $inplog="scans/sorted.merged.scans";
#my $inplog="test";
my ($dadr);
my $sum;
my (%user,@usert,@keys,$key);
# Jul 23 23:44:34 10.10.160.114:777   -> 172.172.124.105:3046 UDP

open(LOGREAD,"$inplog") or die "open nicht ok";
while (<LOGREAD>) {
```

```
   if (/.*\:\d\d\s.+?\:.+?\s->\s.+?\:6970\s.+?\s.*/i) {
     ($dadr) = /.* \:\d\d\s.+?\:.+?\s->\s(.+?)\:6970\s.+?\s.*/i;
     $user{"$dadr"}+=1;
   }
}
foreach $key (sort(keys %user)) {
$_=$key;
($dadr)=/(.+)/;
printf "%10s % -16s\n",$user{$key},$dadr;
$sum=$sum+$user{$key};
}
printf $sum;
close LOGREAD;
exit;
```

script name: **scans.sourcehost.dport6970.pl**
activation: ./scans.sourcehos t.dport6970.pl |sort –r –k1 |less

```
#!/usr/bin/perl  -w

use strict;
use FileHandle;
my $inplog="scans/sorted.merged.scans";
#my $inplog="test";
my ($sadr);
my $sum;
my (%user,@usert,@keys,$key);
# Jul 23 23:44:34 10.10.160.114:777   -> 172.172.124.105:3046 UDP

open(LOGREAD,"$inplog") or die "open nicht ok";
while (<LOGREAD>) {
   if (/.*\:\d\d\s.+?\:.+?\s->\s.+?\:6970\s.+?\s.*/i) {
     ($sadr) = /.* \:\d\d\s(.+?)\:.+?\s->\s.+?\:6970\s.+?\s.*/i;
     $user{"$sadr"}+=1;
   }
}
foreach $key (sort(keys %user  )) {
$_=$key;
($sadr)=/(.+)/;
printf "%10s % -16s\n",$user{$key},$sadr;
$sum=$sum+$user{$key};
}
printf $sum;
close LOGREAD;
exit;
```

The scripts **scans.targethost.dport27005.pl** and **scans.sourcehost.dport27005.pl** are similar to the two scripts for port 69 70 above, only the destination port are changed in the regular expression.

script name: **oos.sourcehost.pl**
activation: ./scans.sourcehost.pl |sort –r –k1 |less

```
#!/usr/bin/perl  -w

use strict;
use FileHandle;
my $inplog="oos/merged.oos";
```

```
my ($sadr);
my (%us er,@usert,@keys,$key);
# 07/23-10:01:07.985411 198.186.202.147:40726   -> 10.10.70.113:113

open(LOGREAD,"$inplog") or die "open nicht ok";
while (<LOGREAD>) {
   if (/.* \s.+?\:.+?\s->\s.+?\:.+?/i) {
     ($sadr) = /.* \s(.+?)\:.+?\s->\s.+?\:.+?/i;
     $user{"$sadr"}+=1;
   }
}
foreach $key (sort(keys %user)) {
#print $key, "=", $user{$key}, "  \n";
$_=$key;
($sadr)=/(.+)/;
printf "%10s % -16s\n",$user{$key},$sadr;
}
close LOGREAD;
exit;
```

Appendix A – Consolidated data from snort_stat.pl output

1. July 23rd, 2001

Subject: snort daily report

The log begins from: 07 23 00:00:06
The log ends at: 07 23 23:50:23
Total events: 154705
Signatures recorded: 128
Source IP recorded: 7992
Destination IP recorded: 2070
Portscan recorded: 6868

The number of attacks from same host to same destination using same method

# of attacks	from	to	method
9273	10.10.162.228	204.152.190.70	ICMP Echo Request Nmap or HPING2
4301	10.10.16.5	10.10.162.228	ICMP Destination Unreachable (Communication Administratively Prohibited)
3508	216.158.21.42	10.10.137.7	ICMP Destination Unreachable (Communication Administratively Prohibited)
2427	10.10.14.1	10.10.162.228	ICMP Destination Unreachable (Communication Administratively Prohibited)
2310	10.10.153.196	211.111.214.125	spp _http_decode: IIS Unicode attack detected
2297	10.10.14.1	10.10.5.4	ICMP Destination Unreachable (Communication Administratively Prohibited)
2292	198.32.224.31	10.10.140.9	ICMP Destination Unreachable (Host Unre achable)
2060	216.158.21.226	10.10.137.7	ICMP Destination Unreachable (Communication Administratively Prohibited)
2030	10.10.97.209	64.12.151.216	spp_http_decode: IIS Unicode attack detected
1850	10.10.85.74	207.200.86.66	spp_http_decode: IIS Unicode attack detected

Percentage and number of attacks from a host to a destination

%	# of attacks	from	to
5.99	9273	10.10.162.228	204.152.190.70
2.78	4301	10.10.16.5	10.10.162.228

2.27	3508	216.158.21.42	10.10.137.7
1.57	2427	10.10.14.1	10.10.162.228
1.49	2310	10.10.153.196	211.111.214.125
1.48	2297	10.10.14.1	10.10.5.4
1.48	2292	198.32.224.31	10.10.140.9
1.33	2060	216.158.21.226	10.10.137.7
1.31	2030	10.10.97.209	64.12.151.216
1.20	1850	10.10.85.74	207.200.86.66

Percentage and number of attacks from one host to any with same method

%	# of attacks	from	method
5.99	9273	10.10.162.228	ICMP Echo Request Nmap or HPING2
3.84	5937	10.10.14.1	ICMP Destination Unreachable (Communication Administratively Prohibited)
3.81	5901	10.10.16.5	ICMP Destination Unreachable (Communication Administratively Prohibited)
2.97	4592	10.10.97.209	spp_http_decode: IIS Unicode attack detected
2.27	3508	216.158.21.42	ICMP Destination Unreachable (Communication Administratively Prohibited)
1.74	2686	10.10.71.19	spp_http_decode: IIS Unicode attack detected
1.73	2675	10.10.85.74	spp_http_decode: IIS Unicode attack detected
1.69	2620	10.10.153.196	spp_http_decode: IIS Unicode attack detected
1.48	2292	198.32.224.31	ICMP Destination Unreachable (Host Unreachable)
1.34	2066	10.10.153.203	spp_http_decode: IIS Unicode attack detected

The distribution of attack methods

%	# of attacks	method
25.19	38972	spp_http_decode
12.71	19666	MISC source port 53 to <1024
	1247	134.93.19.12 -> 10.10.130.122
	160	195.161.192.249 -> 10.10.1.4
	72	63.108.210.10 -> 10.10.1.3
	50	209.182.195.135 -> 10.10.1.3
	49	130.113.45.26 -> 10.10.1.5
	48	130.113.45.26 -> 10.10.1.3
	46	207.25.71.78 -> 10.10.1.5
	41	63.236.72.138 -> 10.10.1.5
	41	130.113.45.26 -> 10.10.1.4
	39	207.25.71.78 -> 10.10.1.4

12.46 19280 MISC traceroute
361 192.5.110.20 -> 10.10.140.9
361 130.191.18.80 -> 10.10.140.9
361 35.8.2.229 -> 10.10.140.9
356 128.182.61.50 -> 10.10.140.9
354 128.82.254.69 -> 10.10.140.9
354 128.114.129.62 -> 10.10.140.9
353 18.201.0.122 -> 10.10.140.9
353 206.220.240.230 -> 10.10.140.9
353 128.103.209.74 -> 10.10.140.9
352 198.202.74.41 -> 10.10.140.9

11.70 18106 ICMP Destination Unreachable (Communication Administratively Prohibited)
4301 10.10.16.5 -> 10.10.162.228
3508 216.158.21.42 -> 10.10.137.7
2427 10.10.14.1 -> 10.10.162.228
2297 10.10.14.1 -> 10.10.5.4
2060 216.158.21.226 -> 10.10.137.7
428 192.80.53.46 -> 10.10.140.9
308 10.10.16.5 -> 10.10.110.56
237 10.10.16.5 -> 10.10.140.196
227 10.10.14.1 -> 10.10.10.56
189 10.10.16.5 -> 10.10.97.35

10.09 15610 WEB-MISC prefix-get //
798 63.48.140.86 -> 10.10.253.114
495 208.11.51.239 -> 10.10.253.114
285 137.187.30.199 -> 10.10.253.114
268 134.192.172.168 -> 10.10.253.114
260 208.58.202.152 -> 10.10.253.114
211 12.42.37.200 -> 10.10.253.114
196 63.82.104.195 -> 10.10.253.114
194 140.90.233.216 -> 10.10.253.114
194 63.210.219.181 -> 10.10.253.114
185 172.159.14.190 -> 10.10.253.114

6.46 9987 ICMP Echo Request Nmap or HPING2
9273 10.10.162.228 -> 204.152.190.70
61 10.10.98.159 -> 10.0.3.3
36 10.10.137.7 -> 207.245.122.125
35 10.10.153.171 -> 24.147.249.241
34 10.10.115.115 -> 141.151.19.217
34 10.10.115.115 -> 12.154.10.63
27 10.10.137.7 -> 216.158.50.240
24 10.10.185.35 -> 149.1.1.1
21 10.10.97.244 -> 172.160.206.40

5.72 8846 INFO MSN IM Chat data
242 10.10.153.195 -> 64.4.13.138
177 64.4.13.192 -> 10.10.97.197
172 10.10.97.199 -> 64.4.13.137

149 10.10.97.241 -> 64.4.13.117
140 64.4.13.165 -> 10.10.97.180
137 10.10.98.164 -> 64.4.13.196
134 64.4.13.195 -> 10.10.98.242
121 10.10.15.185 -> 64.4.13.165
117 10.10.153.201 -> 64.4.13.123
109 10.10.218.230 -> 64.4.13.123

2.06 3180 INFO - ICQ Access
84 10.10.97.187 -> 205.188.248.25
83 10.10.97.187 -> 64.12.184.121
72 10.10.97.159 -> 205.188.248.89
68 10.10.97.187 -> 205.188.140.249
61 10.10.98.129 -> 64.12.174.153
51 10.10.97.187 -> 205.188.248.89
50 10.10.98.129 -> 205.188.248.57
47 10.10.97.159 -> 205.188.140.249
47 10.10.97.187 -> 205.188.140.185
44 10.10.98.129 -> 205.188.140.185

1.94 2999 MISC Large UDP Packet
984 204.248.36.85 -> 10.10.115.74
782 211.35.66.166 -> 10.10.153.143
607 216.106.166.145 -> 10.10.115.74
380 216.106.166.147 -> 10.10.115.74
237 202.109.129.46 -> 10.10.111.188
7 128.2.121.218 -> 10.10.7.10
1 207.25.79.227 -> 10.10.178.155
1 207.25.79.227 -> 10.10.177.16

1.87 2890 ICMP Destination Unreachable (Host Unreachable)
2292 198.32.224.31 -> 10.10.140.9
61 204.177.252.34 -> 10.10.70.161
43 63.146.1.33 -> 10.10.70.97
14 63.146.1.33 -> 10.10.70.42
12 213.65.232.247 -> 10.10.70.97
11 63.146.1.33 -> 10.10.135.3
9 205.171.4.10 -> 10.10.70.97
8 63.146.1.33 -> 10.10.70.72

Portscans performed to/from HOME_NET

# of attacks	from
620	10.10.70.242
564	10.10.217.114
355	10.10.100.230
167	10.10.253.24
141	10.10.160.114
106	10.10.150.220
61	10.10.218.126
54	10.10.111.157
54	10.10.70.97
52	10.10.53.220

2. July 24th, 2001

Subject: snort daily report

The log begins from: 07 24 00:00:06
The log ends at: 07 24 23:51:54
Total events: 160864
Signatures recorded: 115
Source IP recorded: 7697
Destination IP recorded: 2184
Portscan recorded: 6968

The number of attacks from same host to same destination using same method

# of attacks	from	to	method
9042	10.10.162.228	204.152.190.70	ICMP Echo Request Nmap or HPING2
5105	61.153.19.95	10.10.162.164	MISC Large UDP Packet
4185	10.10.16.5	10.10.162.228	ICMP Destination Unreachable (Communication Administratively Prohibited)
3273	10.10.130.127	216.241.219.28	spp_http_decode: CGI Null Byte attack detected
3001	10.10.85.74	207.200.86.66	spp_http_decode: IIS Unicode attack detected
2871	10.10.85.74	20 7.200.86.97	spp_http_decode: IIS Unicode attack detected
2764	216.158.21.226	10.10.137.7	ICMP Destination Unreachable (Communication Administratively Prohibited)
2508	216.158.21.42	10.10.137.7	ICMP Destination U nreachable (Communication Administratively Prohibited)
2332	10.10.14.1	10.10.162.228	ICMP Destination Unreachable (Communication Administratively Prohibited)
2251	198.32.224.31	10.10.140.9	ICMP Destination Unreachab le (Host Unreachable)

Percentage and number of attacks from a host to a destination

%	# of attacks	from	to
5.62	9042	10.10.162.228	204.152.190.70
3.17	5106	61.153.19.95	10.10.162.164
2.60	4185	10.10.16.5	10.10.162.228
2.03	3273	10.10.130.127	216.241.219.28
1.87	3001	10.10.85.74	207.200.86.66
1.83	2949	216.150.152.145	10.10.5.44
1.78	2871	10.10.85.74	207.200.86.97
1.72	2764	216.158.21.226	10.10.137.7
1.56	2508	216.158.21.42	10.10.137.7
1.45	2332	10.10.14.1	10.10.162.228

Percentage and number of attacks from one host to any with same method

%	# of attacks	from	method
5.62	9042	10.10.162.228	ICMP Echo Request Nmap or HPING2
3.66	5889	10.10.85.74	spp_http_decode: IIS Unicode attack detected
3.55	5704	10.10.16.5	ICMP Destination Unreac hable (Communication Administratively Prohibited)
3.37	5425	10.10.14.1	ICMP Destination Unreachable (Communication Administratively Prohibited)
3.17	5105	61.153.19.95	MISC Large UDP Packet
2.03	3273	10.10.130.127	spp_http_decode: CGI Null Byte attack detected
1.72	2764	216.158.21.226	ICMP Destination Unreachable (Communication Administratively Prohibited)
1.59	2552	10.10.157.108	spp_http_decode: IIS Unicode attack detected
1.58	2545	216.150.152.145	ICMP Echo Request L3retriever Ping
1.58	2534	216.150.152.145	SMB Name Wildcard

Percentage and number of attacks to one certain host

%	# of attacks	to	method
11.32	18216	10.10.140.9	MISC traceroute
9.26	14890	10.10.253.114	WEB -MISC prefix-get //
5.62	9042	204.152.190.70	ICMP Echo Request Nmap or HPING2
4.45	7166	10.10.1.3	MISC source port 53 to <1024
4.05	6517	10.10.162.228	ICMP Destination Unreachable (Communication Administratively Prohibited)
3.50	5625	10.10.1.5	MISC source port 53 to <1024
3.35	5392	64.12.151.216	spp_http_decode: IIS Unicode attack detected
3.28	5272	10.10.137.7	ICMP Destination Unreachable (Communication Administratively Prohibited)
3.22	5178	216.241.219.28	spp_http_decode: CGI Null Byte attack detected
3.17	5105	10.10.162.164	MISC Large UDP Packet

The distribution of attack methods

%	# of attacks	method		
20.17	32439	spp_http_decode		
11.97	19259	MISC source port 53 to <1024		
	1020	134.93.19.12	->	10.10.130.122
	56	208.48.26.15	->	10.10.1.3
	54	63.108.210.10	->	10.10.1.3
	54	207.25.71.78	->	10.10.1.5
	53	149.173.1.4	->	10.10.1.5
	46	169.207.0.9	->	10.10.1.4
	42	205.197.182.100	->	10.10.1.5
	37	216.32.74.10	->	10.10.1.2
	37	207.25.71.78	->	10.10.1.4
11.34	18242	MISC traceroute		
	392	129.137.254.5	->	10.10.140.9
	353	128.103.209.74	->	10.10.140.9
	347	129.119.224.250	->	10.10.140.9
	343	128.195.186.5	->	10.10.140.9
	343	129.89.70.20	->	10.10.140.9
	342	129.116.218.196	->	10.10.140.9
	341	128.118.58.5	->	10.10.140.9
	341	199.249.169.82	->	10.10.140.9
	340	131.215.48.15	->	10.10.140.9
	339	130.191.18.80	->	10.10.140.9
9.27	14914	WEB-MISC prefix-get //		
	479	199.196.144.13	->	10.10.253.114
	321	209.218.232.100	->	10.10.253.114

267 207.86.98.202 -> 10.10.253.114
232 169.253.4.1 -> 10.10.253.114
215 208.146.240.164 -> 10.10.253.114
212 24.249.229.91 -> 10.10.253.114
197 65.201.178.34 -> 10.10.253.114
197 198.139.249.94 -> 10.10.253.114
195 63.253.99.158 -> 10.10.253.114
194 63.44.135.162 -> 10.10.253.114

6.97 11206 INFO MSN IM Chat data
589 64.4.13.132 -> 10.10.218.186
220 10.10.98.204 -> 64.4.13.195
211 64.4.13.164 -> 10.10.218.230
187 10.10.98.204 -> 64.4.13.191
160 10.10.98.106 -> 64.4.13.126
151 64.4.13.123 -> 10.10.97.43
130 10.10.153.241 -> 64.4.13.126
122 10.10.97.78 -> 64.4.13.133
120 10.10.98.189 -> 64.4.13.168
112 10.10.98.109 -> 64.4.13.191

6.00 9644 ICMP Ec ho Request Nmap or HPING2
9042 10.10.162.228 -> 204.152.190.70
85 10.10.17.47 -> 65.229.226.249
70 10.10.98.154 -> 64.58.77.28
55 10.10.137.7 -> 207.245.122.125
53 10.10.98.235 -> 64.231.214.213
43 10.10.137.7 -> 216.158.50.240
32 10.10.98.235 -> 24.88.213.8
28 10.10.98.192 -> 65.15.57.92
27 10.10.98.235 -> 66.31.59.224
18 10.10.153.141 -> 24.21.224.230

5.68 9137 spp_http_decode

3.42 5500 MISC Large UDP Packet
5105 61.153.19.95 -> 10.10.162.164
316 209.132.218.11 -> 10.10.153.147
73 63.250.209.74 -> 10.10.178.141
5 208.48.88.11 -> 10.10.98.153
1 207.25.79.227 -> 10.10.97.219
1.96 3157 ICMP Destination Unreachable (Host Unreachable)
2251 198.32.224.31 -> 10.10.140.9
273 63.146.1.33 -> 10.10.70.11
57 142.169.13.225 -> 10.10.70.174
20 207.136.109.3 -> 10.10.70.11
18 195.249.6.93 -> 10.10.70.97
1.87 3007 INFO - ICQ Access
109 10.10.217.18 -> 205.188.248.89
97 10.10.217.18 -> 205.188.248.25
89 10.10.98.164 -> 64.12.184.89

88	10.10.98.164	-> 205.188.248.25
87	10.10.98.122	-> 152.163.180.24
72	10.10.217.18	-> 205.188.248.57
52	10.10.98.225	-> 205.188.248.25
46	10.10.98.152	-> 64.12.174.153
46	10.10.97.184	-> 152.163.180.24
45	10.10.97.207	-> 64.12.184.57

Portscans performed to/from HOME_NET

# of attacks	from
637	10.10.217.114
609	10.10.70.242
315	10.10.100.230
275	10.10.253.24
174	10.10.150.220
162	10.10.160.114
141	10.10.218.106
107	10.10.60.43
69	10.10.218.206

3. July 25th, 2001

Subject: snort daily report

The log begins from: 07 25 00:00:05
The log ends at: 0 7 25 23:52:57
Total events: 245927
Signatures recorded: 111
Source IP recorded: 8602
Destination IP recorded: 7811
Portscan recorded: 7270

The number of attacks from same host to same destination using same method

# of attacks	from	to	method
24163	10.10.75.150	216.241.219.14	spp_http_decode: CGI Null Byte attack detecte d
10484	10.10.157.108	207.200.89.193	spp_http_decode: IIS Unicode attack detected
9177	10.10.162.228	204.152.190.70	ICMP Echo Request Nmap or HPING2
7532	10.10.157.108	207.200.89.225	spp_http_decode: IIS U nicode attack detected
5664	10.10.104.47	207.200.86.66	spp_http_decode: IIS Unicode attack detected
5618	10.10.85.74	207.200.86.97	spp_http_decode: IIS Unicode attack detected

4426	10.10.85.74	207.200.86 .66	spp_http_decode: IIS Unicode attack detected
4401	10.10.104.47	207.200.86.97	spp_http_decode: IIS Unicode attack detected
4291	10.10.16.5	10.10.162.228	ICMP Destination Unreachable (Communication Administra tively Prohibited)
4236	216.150.152.145	10.10.5.44	SMB Name Wildcard

Percentage and number of attacks from a host to a destination

%	# of attacks	from	to
9.83	24163	10.10.75.150	216.241.219.14
4.26	10484	10.10.157.108	207.200.89.193
3.73	9177	10.10.162.228	204.152.190.70
3.44	8448	216.150.152.145	10.10.5.44
3.06	7532	10.10.157.108	207.200.89.225
2.81	6906	216.150.152.145	10.10.5.45
2.30	5664	10.10.104.47	207.200.86.66
2.28	5618	10.10.85.74	207.200.86.97
1.80	4426	10.10.85.74	207.200.86.66
1.79	4401	10.10.104.47	207.200.86.97

Percentage and number of attacks from one host to any with same method

%	# of attacks	from	method
9.83	24163	10.10.75.150	spp_http_decode: CGI Null Byte attack detected
8.80	21644	10.10.157.108	spp_http_decode: IIS Un icode attack detected
4.09	10065	10.10.104.47	spp_http_decode: IIS Unicode attack detected
4.08	10044	10.10.85.74	spp_http_decode: IIS Unicode attack detected
3.73	9177	10.10.162.228	ICMP Echo Request Nmap or HP ING2
3.13	7692	216.150.152.145	SMB Name Wildcard
3.12	7662	216.150.152.145	ICMP Echo Request L3retriever Ping
2.84	6995	10.10.16.5	ICMP Destination Unreachable (Communication Administratively Prohibited)
2.51	6165	10.10.14.1	ICMP Destination Unreachable (Communication Administratively Prohibited)
1.61	3961	10.10.15.71	spp_http_decode: IIS Unicode attack detected

Percentage and number of attacks to one certain host

%	# of attacks	to	method
9.83	24163	216.241.219.14	spp_http_decode: CGI Null Byte attack d etected

7.73	19021	10.10.140.9	MISC traceroute
6.00	14747	10.10.253.114	WEB -MISC prefix-get //
5.11	12564	207.200.89.193	spp_http_decode: IIS Unicode attack detected
4.46	10970	207.200.86.66	spp_http_decode: IIS Unicode attack detected
4.35	10710	207.200.86.97	spp_http_decode: IIS Unicode attack detected
4.08	10032	207.200.89.225	spp_http_decode: IIS Unicode attack detected
3.73	9177	204.152.190.70	ICMP Ec ho Request Nmap or HPING2
2.71	6658	10.10.162.228	ICMP Destination Unreachable (Communication Administratively Prohibited)
2.59	6364	10.10.1.4	MISC source port 53 to <1024

The distribution of attack methods

%	# of attacks	method	
28.80	70820	spp_http_decode	
12.05	29631	spp_http_decode	
9.77	24039	Possible trojan server activity	
	6	137.216.4.10	-> 10.10.4.3
	6	64.229.69.60	-> 10.10.218.106
	6	10.10.53.39	-> 64.231.133.240
	5	10.10.217.70	-> 64.245.58.28
	4	10.10.226.177	-> 24.226.114.182
	4	24.141.140.131	-> 10.10.18.9
	4	24.169.166.164	-> 10.10.153.72
	4	24.36.83.12	-> 10.10.68.32
	4	64.229.20.127	-> 10.10.204.78
	4	10.10.226.245	-> 24.226.114.182
7.75	19070	MISC traceroute	
	362	131.247.254.97	-> 10.10.140.9
	361	199.165.80.10	-> 10.10.140.9
	359	128.182.61.50	-> 10.10.140.9
	357	146.229.127.200	-> 10.10.140.9
	356	137.145.206.101	-> 10.10.140.9
	356	128.82.254.69	-> 10.10.140.9
	356	129.79.20.239	-> 10.10.140.9
	353	131.215.48.15	-> 10.10.140.9
	352	216.79.60.26	-> 10.10.140.9
	352	128.118.58.5	-> 10.10.140.9
7.04	17309	ICMP Destination Unreachable (Communication Adminis tratively Prohibited)	
	4291	10.10.16.5	-> 10.10.162.228
	2367	10.10.14.1	-> 10.10.162.228
	2225	10.10.14.1	-> 10.10.5.4
	2186	216.158.21.226	-> 10.10.137.7
	920	216.158.21.42	-> 10.10.137.7
	513	192.80.53.46	-> 10.10.140.9
	444	10.10.16.5	-> 10.10.111.131

375 10.10.16.5 -> 10.10.98.145
303 10.10.16.5 -> 10.10.110.56
247 10.10.14.1 -> 10.10.98.160

6.03 14839 WEB-MISC prefix-get //
354 66.44.15.104 -> 10.10.253.114
317 137.187.144.220 -> 10.10.253.114
292 207.144.254.249 -> 10.10.253.114
270 198.26.130.37 -> 10.10.253.114
238 128.8.23.129 -> 10.10.253.114
219 141.161.104.140 -> 10.10.253.114
208 198.50.63.15 -> 10.10.253.114
206 66.44.17.127 -> 10.10.253.114
199 208.210.197.2 -> 10.10.253.114
195 63.16.79.64 -> 10.10.253.114

3.90 9603 INFO MSN IM Chat data
226 10.10.98.113 -> 64.4.13.133
177 10.10.98.128 -> 64.4.13.192
165 10.10.98.250 -> 64.4.13.117
152 10.10.98.202 -> 64.4.13.125
149 64.4.13.168 -> 10.10.219.10
146 64.4.13.194 -> 10.10.160.121
145 10.10.98.112 -> 64.4.13.138
129 64.4.13.165 -> 10.10.97.220
119 10.10.160.179 -> 64.4.13.164
113 64.4.13.113 -> 10.10.98.162

3.83 9424 ICMP Echo Request Nmap or HPING2
9177 10.10.162.228 -> 204.152.190.70
60 10.10.97.209 -> 204.192.47.24
48 10.10.137.7 -> 207.245.122.125
21 10.10.137.7 -> 216.158.50.240
14 10.10.97.192 -> 64.65.45.117
8 10.10.99.53 -> 149.1.1.1
8 10.10.110.33 -> 149.1.1.1
7 10.10.150.220 -> 193.145.124.6
6 10.10.10.21 -> 149.1.1.1
5 10.10.99.225 -> 149.1.1.1

3.21 7886 SMB Name Wildcard
4236 216.150.152.145 -> 10.10.5.44
3456 216.150.152.145 -> 10.10.5.45
12 130.13.103.236 -> 10.10.137.5
11 211.192.158.74 -> 10.10.132.1
10 216.158.50.7 -> 10.10.137.7

3.18 7824 ICMP Echo Request L3retriever Ping
4212 216.150.152.145 -> 10.10.5.44
3450 216.150.152.145 -> 10.10.5.45
155 10.10.98.172 -> 64.14.116.72
4 10.10.137.7 -> 216.158.50.240
3 207.245.122.125 -> 10.10.137.7

Portscans performed to/from HOME_NET

# of attacks	from
680	10.10.70.242
326	10.10.100.230
307	10.10.253.24
286	10.10.217.114
233	10.10.150.220
135	10.10.160.114
89	10.10.218.206
75	10.10.253.53
75	24.180.237.142
51	10.10.98.147

4. July 26th, 2001

Subject: snort daily report

The log begins from: 07 26 00:00:06
The log ends at: 07 26 23:51:35
Total events: 184370
Signatures recorded: 122
Source IP recorded: 7042
Destination IP recorded: 3187
Portscan recorded: 5957

The number of attacks from same host to same destination using same method

# of attacks	from	to	method
13131	216.166.219.72	10.10.160.114	High port 65535 udp - possible Red Worm - traffic
12136	10.10.157.108	207.200.89.225	spp_http_decode: IIS Unic ode attack detected
9360	10.10.157.108	207.200.89.193	spp_http_decode: IIS Unicode attack detected
8615	10.10.162.228	204.152.190.70	ICMP Echo Request Nmap or HPING2
6096	10.10.157.108	64.12.151.216	spp_h ttp_decode: IIS Unicode attack detected
4021	10.10.16.5	10.10.162.228	ICMP Destination Unreachable (Communication Administratively Prohibited)
3217	10.10.85.74	207.200.86.97	spp_http_decode: IIS Unicode attack detec ted
2860	10.10.71.19	64.12.151.216	spp_http_decode: IIS Unicode attack detected
2549	10.10.85.74	207.200.86.66	spp_http_decode: IIS Unicode attack detected

2526 216.158.21.42 10.10.137.7 ICMP Dest ination Unreachable (Communication Administratively Prohibited)

Percentage and number of attacks from a host to a destination

%	# of attacks	from	to
7.12	13131	216.166.219.72	10.10.160.114
6.58	12136	10.10.157.108	207.200.89.225
5.08	9360	10.10.157.108	207.200.89.193
4.67	8615	10.10.162.228	204.152.190.70
3.31	6096	10.10.157.108	64.12.151.216
2.68	4947	216.150.152.145	10.10.5.45
2.45	4518	216.150.152.145	10.10.5.44
2.18	4021	10.10.16.5	10.10.162.228
1.74	3217	10.1 0.85.74	207.200.86.97
1.55	2860	10.10.71.19	64.12.151.216

Percentage and number of attacks from one host to any with same method

%	# of attacks	from	method
15.11	27856	10.10.157.108	spp_http_decode: IIS Unicode attack detected
7.12	13131	216.166.219.72	High port 65535 udp - possible Red Worm - traffic
4.67	8615	10.10.162.228	ICMP Echo Request Nmap or HPING2
3.13	5770	10.10.85.74	spp_http_decode: IIS Unicode attack detected
3.10	5720	10.10.16.5	ICMP Destination Unreachable (Communication Administratively Prohi bited)
2.89	5336	10.10.14.1	ICMP Destination Unreachable (Communication Administratively Prohibited)
2.59	4780	216.150.152.145	SMB Name Wildcard
2.54	4680	216.150.152.145	ICMP Echo Request L3retriever Ping
2.53	4668	10.10.71.19	spp_http_decode: IIS Unicode attack detected
1.43	2630	10.10.153.146	spp_http_decode: IIS Unicode attack detected

Percentage and number of attacks to one certain host

%	# of attacks	to	method
9.41	17344	10.10.140.9	MISC traceroute
7.78	14350	207.200.89.225	s pp_http_decode: IIS Unicode attack detected
7.12	13131	10.10.160.114	High port 65535 udp - possible Red Worm - traffic
5.87	10814	64.12.151.216	spp_http_decode: IIS Unicode attack detected

5.52	10174	207.200.89.193	spp _http_decode: IIS Unicode attack detected
5.06	9325	10.10.253.114	WEB -MISC prefix -get //
4.67	8615	204.152.190.70	ICMP Echo Request Nmap or HPING2
3.42	6299	10.10.162.228	ICMP Destination Unreachable (Communication Administratively Prohibited)
2.78	5134	10.10.1.3	MISC source port 53 to <1024
2.49	4595	10.10.1.5	MISC source port 53 to <1024

The distribution of attack methods

%	# of attacks	method	
32.44	59810	spp_http_decode	
9.44	17396	MISC traceroute	
	336	128.252.120.3	-> 10.10.140.9
	334	129.137.254.5	-> 10.10.140.9
	330	128.114.129.62	-> 10.10.140.9
	327	170.140.127.97	-> 10.10.140.9
	327	128.249.1.199	-> 10.10.140.9
	327	146.229.127.200	-> 10.10.140.9
	325	137.145.206.116	-> 10.10.140.9
	325	129.237.15.1	-> 10.10.140.9
	324	18.201.0.122	-> 10.10.140.9
	322	137.78.21.22	-> 10.10.140.9
8.48	15642	ICMP Destination Unreachable (Communication Administratively Prohibited)	
	4021	10.10.16.5	-> 10.10.162.228
	2526	216.158.21.42	-> 10.10.137.7
	2278	10.10.14.1	-> 10.10.162.228
	2082	10.10.14.1	-> 10.10.5.4
	1486	216.158.21.226	-> 10.10.137.7
	366	192.80.53.46	-> 10.10.140.9
	264	10.10.16.5	-> 10.10.110.56
	189	10.10.16.5	-> 10.10.218.114
	166	10.10.16.5	-> 10.10.60.11
	158	10.10.16.5	-> 10.10.98.200
7.16	13207	High port 65535 udp - possible Red Worm - traffic	
	13131	216.166.219.72	-> 10.10.160.114
	16	217.59.83.45	-> 10.10.163.54
	14	209.10.56.35	-> 10.10.177.22
	11	217.59.83.44	-> 10.10.163.54
	8	62.158.28.160	-> 10.10.70.242
	8	209.10.56.38	-> 10.10.177.22
	4	62.227.62.212	-> 10.10.70.242
	3	64.182.96.150	-> 10.10.160.169
	3	199.34.53.73	-> 10.10.183.12
5.61	10334	INFO MSN IM Chat data	

196 64.4.13.191 -> 10.10.98.129
184 10.10.97.21 -> 64.4.13.121
157 10.10.160.179 -> 64.4.13.161
152 64.4.13.121 -> 10.10.97.21
150 64.4.13.135 -> 10.10.98.124
144 10.10.98.183 -> 64.4.13.115
142 10.10.160.179 -> 64.4.13.197
122 10.10.97.169 -> 64.4.13.121
121 10.10.98.226 -> 64.4.13.123
110 10.10.98.124 -> 64.4.13.135

5.38 9928 ICMP Echo Request Nmap or HPING2
8615 10.10.162.228 -> 204.152.190.70
144 10.10.97.69 -> 10.0.3.3
51 10.10.98.109 -> 206.251.6.192
38 10.10.98.211 -> 24.182.118.68
38 10.10.98.211 -> 24.222.133.138
37 10.10.115.115 -> 66.41.33.78
32 10.10.137.7 -> 216.158.50.240
31 10.10.97.162 -> 24.116.72.121
30 10.10.98.211 -> 66.56.140.64
28 10.10.137.7 -> 207.245.122.125

5.07 9347 WEB-MISC prefix-get //
250 206.71.96.3 -> 10.10.253.114
246 63.22.71.141 -> 10.10.253.114
194 64.91.149.214 -> 10.10.253.114
185 24.4.252.26 -> 10.10.253.114
162 63.21.84.171 -> 10.10.253.114
159 198.26.119.86 -> 10.10.253.114
129 216.76.144.73 -> 10.10.253.114
126 141.157.90.35 -> 10.10.253.114
121 24.4.252.28 -> 10.10.253.114
116 131.118.250.205 -> 10.10.253.114

2.69 4968 SMB Name Wildcard
2490 216.150.152.145 -> 10.10.5.45
2290 216.150.152.145 -> 10.10.5.44
11 204.245.190.131 -> 10.10.132.10
10 64.12.34.214 -> 10.10.137.7

2.55 4704 ICMP Echo Request L3retriever Ping
2452 216.150.152.145 -> 10.10.5.45
2228 216.150.152.145 -> 10.10.5.44
11 207.245.122.245 -> 10.10.137.7
4 10.10.137.7 -> 216.158.50.240
3 10.10.97.157 -> 146.225.100.239
2 216.158.50.245 -> 10.10.137.7
2 216.158.50.240 -> 10.10.137.7
2 10.10.98.201 -> 209.202.149.202
1.75 3218 Watchlist 000222 NET -NCFC

1048 159.226.125.129 -> 10.10.6.47
571 159.226.39.4 -> 10.10.6.7
357 159.226.125.129 -> 10.10.6.34
342 159.226.5.83 -> 10.10.100.165
289 159.226.39.4 -> 10.10.253.43
207 159.226.8.67 -> 10.10.100.165
175 159.226.39.4 -> 10.10.253.41
124 159.226.158.174 -> 10.10.253.114
98 159.226.39.4 -> 10.10.253.42
4 159.226.5.222 -> 10.10.100.230
3 159.226.6.2 -> 10.10.111.197

Portscans performed to/from HOME_NET

# of attacks	from
692	10.10.217.114
531	10.10.70.242
315	10.10.150.220
312	10.10.100.230
234	10.10.253.24
96	10.10.6.45
93	10.10.160.114
78	141.157.88.27
71	10.10.218.206
51	10.10.253.53

5. July 27th, 2001

Subject: snort daily report

The log begins from: 07 27 00:00:08
The log ends at: 07 27 23:52:24
Total events: 158938
Signatures recorded: 118
Source IP recorded: 7275
Destination IP recorded: 2139
Portscan recorded: 6068

The number of attacks from same host to same destination using same method

# of attacks	from	to	met hod
9528	10.10.162.228	204.152.190.70	ICMP Echo Request Nmap or HPING2

8733	204.178.125.65	10.10.163.111	BACKDOOR NetMetro Incoming Traffic
5556	10.10.157.108	207.200.89.193	spp_http_decode: IIS Unicode attack detected
4419	10.10.16.5	10.10.162.228	ICMP Destination Unreachable (Communication Administratively Prohibited)
3867	10.10.98.181	207.200.86.66	spp_http_decode: IIS Unicode attack detected
3804	10.10.157.108	207.200.89.225	spp_http_decode: IIS Unicode attack detected
3090	216.158.21.42	10.10.137.7	ICMP Destination Unreachable (Communication Administratively Prohibited)
2927	216.150.152.145	10.10.5.45	SMB Name Wildcard
2912	216.150.152.145	10.10.5.45	ICMP Echo Request L3retriever Ping
2732	216.158.21.226	10.10.137.7	ICMP Destination Unreachable (Communication Administratively Prohibited)

Percentage and number of attacks from a host to a destination

%	# of attacks	from	to
5.99	9528	10.10.162.228	204.152.190.70
5.49	8733	204.178.125.65	10.10.163.111
3.67	5839	216.150.152.145	10.10.5.45
3.50	5556	10.10.157.108	207.200.89.193
2.78	4419	10.10.16.5	10.10.162.228
2.44	3874	216.150.152.145	10.10.5.44
2.43	3867	10.10.98.181	207.200.86.66
2.39	3804	10.10.157.108	207.200.89.225
1.94	3090	216.158.21.42	10.10.137.7
1.72	2732	216.158.21.226	10.10.137.7

Percentage and number of attacks from one host to any with same method

%	# of attacks	from	method
6.55	10404	10.10.157.108	spp_http_decode: IIS Unicode attack detected
5.99	9528	10.10.162.228	ICMP Echo Request Nmap or HPING2
5.49	8733	204.178.125.65	BACKDOOR NetMe tro Incoming Traffic
3.57	5670	10.10.14.1	ICMP Destination Unreachable (Communication Administratively Prohibited)
3.52	5589	10.10.98.181	spp_http_decode: IIS Unicode attack detected
3.41	5420	10.10.16.5	ICM P Destination Unreachable (Communication Administratively Prohibited)
3.07	4887	216.150.152.145	SMB Name Wildcard
3.04	4826	216.150.152.145	ICMP Echo Request L3retriever Ping
1.94	3090	216.158.21.42	ICMP Destin ation Unreachable (Communication Administratively Prohibited)

1.72	2732	216.158.21.226	ICMP Destination Unreachable (Communication Administratively Prohibited)

Percentage and number of attacks to one certain host

%	# of attacks	to	method
12.38	19673	10.10.140.9	MISC traceroute
11.17	17752	10.10.1.3	MISC source port 53 to <1024
6.19	9832	10.10.253.114	WEB -MISC prefix -get //
5.99	9528	204.152.190.70	ICMP Echo Request Nmap or HPING2
5.49	8733	10.10.163.111	BACKDOOR NetMetro Incoming Traffic
4.39	6970	10.10.162.228	ICMP Destination Unreachable (Communication Administratively Prohibited)
3.92	6223	207.200.89.193	spp_http_decode: IIS Unicode attack detected
3.66	5822	10.10.137.7	ICMP Destination Unreachable (Communicatio n Administratively Prohibited)
3.42	5430	10.10.1.5	MISC source port 53 to <1024
2.92	4646	10.10.1.4	MISC source port 53 to <1024

The distribution of attack methods

%	# of attacks	method
18.49	29385	MISC source port 53 to <1024
	1699	206.132.75.247 -> 10.10.1.3
	1394	206.132.75.249 -> 10.10.1.3
	1249	134.93.19.12 -> 10.10.130.122
	943	209.184.192.51 -> 10.10.1.3
	941	209.184.192.50 -> 10.10.1.3
	550	207.127.157.135 -> 10.10.1.3
	480	205.232.111.3 -> 10.10.1.3
	471	205.232.111.2 -> 10.10.1.3
	361	207.127.157.10 -> 10.10.1.3
	288	207.171.178.7 -> 10.10.1.3
16.78	26664	spp_http_decode
12.39	19699	MISC traceroute
	379	128.82.254.69 -> 10.10.140.9
	375	192.5.110.20 -> 10.10.140.9
	373	206.220.240.230 -> 10.10.140.9
	371	139.78.100.102 -> 10.10.140.9
	370	152.3.2.244 -> 10.10.140.9
	368	138.26.220.46 -> 10.10.140.9
	368	18.201.0.122 -> 10.10.140.9
	367	132.198.101.254 -> 10.10.140.9
	367	129.237.15.1 -> 10.10.140.9
	367	128.182.61.50 -> 10.10.140.9

6.56 10428 ICMP Echo Request Nmap or HPING2
9528 10.10.162.228 -> 204.152.190.70
461 10.10.156.125 -> 207.46.131.30
69 10.10.97.224 -> 10.0.3.3
62 10.10.137.7 -> 216.158.50.240
39 10.10.98.134 -> 206.45.220.176
19 10.10.99.53 -> 149.1.1.1
18 10.10.97.222 -> 146.83.19.208
18 10.10.98.134 -> 4.43.165.85
16 10.10.97.205 -> 216.35.217.28
10 10.10.110.33 -> 149.1.1.1

6.19 9835 WEB-MISC prefix-get //
282 63.44.198.248 -> 10.10.253.114
275 61.168.11.14 -> 10.10.253.114
242 63.74.185.126 -> 10.10.253.114
236 24.6.134.211 -> 10.10.253.114
202 24.4.161.134 -> 10.10.253.114
186 206.99.186.21 -> 10.10.253.114
179 24.13.120.48 -> 10.10.253.114
165 134.113.185.96 -> 10.10.253.114
159 206.128.71.76 -> 10.10.253.114
159 216.10.178.218 -> 10.10.253.114

5.49 8733 BACKDOOR NetMetro Incoming Traffic
8733 204.178.125.65 -> 10.10.163.111
4.46 7083 INFO MSN IM Chat data
246 10.10.98.221 -> 64.4.13.167
204 10.10.139.10 -> 64.4.13.117
182 64.4.13.135 -> 10.10.98.126
177 64.4.13.167 -> 10.10.98.221
173 64.4.13.161 -> 10.10.97.16
170 10.10.97.178 -> 64.4.13.196
161 64.4.13.137 -> 10.10.99.39
140 64.4.13.165 -> 10.10.53.40
98 10.10.97.199 -> 64.4.13.193
97 64.4.13.197 -> 10.10.97.161

3.18 5054 SMB Name Wildcard
2927 216.150.152.145 -> 10.10.5.45
1960 216.150.152.145 -> 10.10.5.44
6 130.13.91.62 -> 10.10.133.115
6 130.13.149.59 -> 10.10.137.235
6 65.2.38.115 -> 10.10.132.119
6 211.220.82.215 -> 10.10.137.11
6 211.236.46.105 -> 10.10.135.119
6 130.13.163.244 -> 10.10.133.14
5 130.13.148.165 -> 10.10.135.146
3.08 4900 ICMP Echo Request L3retriever Ping
2912 216.150.152.145 -> 10.10.5.45

	1914	216.150.152.145 -> 10.10.5.44
	52	216.158.50.245 -> 10.10.137.7
	6	207.245.122.245 -> 10.10.137.7
	4	10.10.137.7 -> 216.158.50.240
	4	207.245.122.125 -> 10.10.137.7
	4	216.158.50.240 -> 10.10.137.7
	2	209.205.91.18 -> 10.10.60.16
	2	209.205.91.18 -> 10.10.60.11
2.12	3375	ICMP Destination Unreachable (Host Unreachable)
	2338	198.32.224.31 -> 10.10.140.9
	63	63.146.1.33 -> 10.10.70.11
	63	131.118.255.17 -> 10.10.70.225
	50	131.118.255.17 -> 10.10.140.9
	24	209.165.57.5 -> 10.10.70.11
	24	131.118.255.17 -> 10.10.70.161
	21	193.152.5.161 -> 10.10.70.161
	20	131.118.255.17 -> 10.10.137.7
	18	213.65.232.247 -> 10.10.70.11
	15	63.146.1.33 -> 10.10.135.3

Portscans performed to/from HOME_NET

# of attacks	from
598	10.10.70.242
396	10.10.217.26
285	10.10.100.230
259	10.10.150.220
254	10.10.253.24
142	10.10.160.114
88	10.10.186.17
78	199.183.24.194
77	10.10.111.89
69	10.10.53.197

6. July 28th, 2001

Subject: snort daily report

The log begins from: 07 28 00:00:07
The log ends at: 07 28 23:56:46
Total events: 102748
Signatures recorded: 109
Source IP recorded: 4847
Destination IP recorded: 1752
Portscan recorded: 4890

The number of attacks from same host to same destination using same method

# of attacks	from	to	method
9723	10.10.162.228	204.152.190.70	ICMP Echo Request Nmap or HPING2
4664	10.10 .16.5	10.10.162.228	ICMP Destination Unreachable (Communication Administratively Prohibited)
3412	216.150.152.145	10.10.5.45	SMB Name Wildcard
3391	216.150.152.145	10.10.5.45	ICMP Echo Request L3retriever Ping
2782	216.158.21.42	10.10.137.7	ICMP Destination Unreachable (Communication Administratively Prohibited)
2702	10.10.14.1	10.10.162.228	ICMP Destination Unreachable (Communication Administratively Prohibited)
2611	10.10.14.1	10.10.5.4	ICMP Destination Unreachable (Communication Administratively Prohibited)
2527	216.158.21.226	10.10.137.7	ICMP Destination Unreachable (Communication Administratively Prohibited)
2443	198.32.224.31	10.10.140.9	ICMP Destination Unreachable (Host Unreachable)
2429	10.10.16.5	10.10.115.155	ICMP Destination Unreachable (Communication Administratively Prohibited)

Percentage and number of attacks from a h ost to a destination

%	# of attacks	from	to
9.46	9723	10.10.162.228	204.152. 190.70
6.62	6803	216.150.152.145	10.10.5.45
4.64	4768	216.150.152.145	10.10.5.44
4.54	4664	10.10.16.5	10.10.162.228
2.71	2782	216.158.21.42	10.10.137.7
2.63	2702	10.10.14.1	10.10.162.228
2.54	2611	10.10.14.1	10.10.5.4
2.46	2527	216.158.21.226	10.10.137.7
2.38	2443	198.32.224.31	10.10.140.9
2.36	2429	10.10.16.5	10.10.115.155

Percentage and number of attacks from one host to any with same method

%	# of attacks	from	method
9.46	9723	10.10.162.2 28	ICMP Echo Request Nmap or HPING2

8.04	8261	10.10.16.5	ICMP Destination Unreachable (Communication Administratively Prohibited)
5.94	6108	10.10.14.1	ICMP Destination Unreachable (Communication Administratively Proh ibited)
5.63	5786	216.150.152.145	SMB Name Wildcard
5.63	5785	216.150.152.145	ICMP Echo Request L3retriever Ping
2.71	2782	216.158.21.42	ICMP Destination Unreachable (Communication Administratively Prohibited)
2.46	2527	216.158.21.226	ICMP Destination Unreachable (Communication Administratively Prohibited)
2.38	2443	198.32.224.31	ICMP Destination Unreachable (Host Unreachable)
1.54	1583	207.46.230.189	MISC Large UDP Packet
1.13	1163	134.93.19.12	MISC source port 53 to <1024

Percentage and number of attacks to one certain host

%	# of attacks	to	method
19.05	19573	10.10.140.9	MISC traceroute
9.46	9723	204.152.190.70	ICMP Echo Request Nmap or HPING2
7.17	7366	10.10.162.228	ICMP Destination Unreacha ble (Communication Administratively Prohibited)
5.17	5309	10.10.137.7	ICMP Destination Unreachable (Communication Administratively Prohibited)
4.38	4497	10.10.253.114	WEB -MISC prefix -get //
4.21	4328	10.10.1.5	MISC source port 53 to <1024
4.04	4151	10.10.1.4	MISC source port 53 to <1024
3.97	4074	10.10.1.3	MISC source port 53 to <1024
3.32	3412	10.10.5.45	SMB Name Wildcard
3.30	3391	10.10.5.45	ICMP Echo Request L3retriever Ping

The distribution of attack methods

%	# of attacks	method
19.91	20461	ICMP Destination Unrea chable (Communication Administratively Prohibited)
	4664	10.10.16.5 -> 10.10.162.228
	2782	216.158.21.42 -> 10.10.137.7
	2702	10.10.14.1 -> 10.10.162.228
	2611	10.10.14.1 -> 10.10.5.4
	2527	216.158.21.226 -> 10.10.137.7
	2429	10.10.16.5 -> 10.10.115.155
	475	192.80.53.46 -> 10.10.140.9
	355	10.10.16.5 -> 10.10.110.56
	297	10.10.16.5 -> 10.10.115.178
	129	151.159.97.254 -> 10.10.140.9

19.08 19600 MISC traceroute
380 205.166.205.222 -> 10.10.140.9
379 134.79.196.42 -> 10.10.140.9
378 128.103.209.74 -> 10.10.140.9
374 137.78.21.22 -> 10.10.140.9
373 128.182.61.50 -> 10.10.140.9
373 137.145.206.116 -> 10.10.140.9
373 35.8.2.229 -> 10.10.140.9
373 199.249.169.82 -> 10.10.140.9
371 130.191.18.80 -> 10.10.140.9
371 137.145.206.101 -> 10.10.140.9

13.69 14071 MISC source port 53 to <1024
1163 134.93.19.12 -> 10.10.130.122
453 204.34.211.86 -> 10.10.1.4
449 204.34.211.86 -> 10.10.1.5
175 204.34.211.86 -> 10.10.1.3
159 204.34.211.87 -> 10.10.1.5
155 204.34.211.81 -> 10.10.1.4
148 204.34.211.81 -> 10.10.1.5
144 204.34.211.82 -> 10.10.1.5
119 204.34.211.87 -> 10.10.1.4
107 204.34.211.82 -> 10.10.1.4

5.78 5934 SMB Name Wildcard
3412 216.150.152.145 -> 10.10.5.45
2374 216.150.152.145 -> 10.10.5.44
8 130.13.120.130 -> 10.10.135.127
8 62.163.146.66 -> 10.10.137.203
6 64.12.34.214 -> 10.10.137.7
6 169.254.187.207 -> 10.10.134.41
5 61.216.5.150 -> 10.10.135.227
5 203.134.61.111 -> 10.10.134.41
5 130.74.106.81 -> 10.10.134.197
4 61.217.73.71 -> 10.10.137.153

4.50 4620 spp_http_decode
4.38 4503 WEB-MISC prefix-get //
315 131.118.250.190 -> 10.10.253.114
279 199.196.144.12 -> 10.10.253.114
231 209.36.53.212 -> 10.10.253.114
152 64.20.3.194 -> 10.10.253.114
150 140.198.37.149 -> 10.10.253.114
141 12.78.125.143 -> 10.10.253.114
126 63.48.100.106 -> 10.10.253.114
113 64.198.132.69 -> 10.10.253.114
111 63.44.145.111 -> 10.10.253.114
108 61.169.127.73 -> 10.10.253.114
3.90 4011 INFO MSN IM Chat data
131 10.10.97.227 -> 64.4.13.191
116 64.4.13.166 -> 10.10.98.237

105 10.10.98.218 -> 64.4.13.124
104 64.4.13.122 -> 10.10.97.236
101 10.10.97.231 -> 64.4.13.139
99 10.10.98.237 -> 64.4.13.166
99 10.10.97.236 -> 64.4.13.122
90 64.4.13.191 -> 10.10.97.227
85 64.4.13.124 -> 10.10.98.218
85 64.4.13.168 -> 10.10.97.217

3.61 3712 ICMP Destination Unreachable (Host Unreachable)
2443 198.32.224.31 -> 10.10.140.9
284 63.146.1.33 -> 10.10.140.9
44 63.146.1.33 -> 10.10.70.11
37 216.135.4.21 -> 10.10.70.174
13 206.115.152.194 -> 10.10.70.161
9 212.185.251.69 -> 10.10.70.11
9 195.162.200.1 -> 10.10.70.11
8 192.169.40.139 -> 10.10.70.11
8 62.225.254.9 -> 10.10.70.11
8 206.115.220.248 -> 10.10.70.11
1.71 1753 MISC Large UDP Packet
1583 207.46.230.189 -> 10.10.182.13
170 216.54.223.195 -> 10.10.153.159
1.24 1276 INFO - ICQ Access
55 10.10.98.153 -> 205.188.248.89
40 10.10.98.153 -> 64.12.174.185
40 10.10.98.120 -> 205.188.248.25
36 10.10.98.153 -> 205.188.248.25
34 10.10.97.223 -> 205.188.248.89
31 10.10.98.153 -> 64.12.174.153
29 10.10.98.132 -> 205.188.248.57
28 10.10.162.117 -> 205.188.248.57
27 10.10.98.132 -> 205.188.248.89
25 10.10.98.121 -> 205.188.248.25

Portscans performed to/from HOME_NET

# of attacks	from
595	10.10.217.26
531	10.10.70.242
521	10.10.150.220
215	10.10.160.114
207	10.10.100.230
156	10.10.253.24
79	207.109.34.25
67	10.10.98.164
65	10.10.186.17
59	10.10.98.159

7. July 29th, 2001

Subject: snort daily report

The log begins from: 07 29 00:00:02
The log ends at: 07 29 23:53:13
Total events: 115642
Signatures recorded: 111
Source IP recorded: 5047
Destination IP recorded: 1733
Portscan recorded: 3524

The number of attacks from same host to same destination using same method

# of attacks	from	to	method
9316	10.10.162.228	204.152.190.70	ICMP Echo Re quest Nmap or HPING2
5958	10.10.16.5	10.10.115.155	ICMP Destination Unreachable (Communication Administratively Prohibited)
4611	216.150.152.145	10.10.5.44	ICMP Echo Request L3retriever Ping
4607	216.150.152. 145	10.10.5.44	SMB Name Wildcard
4379	216.150.152.145	10.10.5.45	ICMP Echo Request L3retriever Ping
4314	216.150.152.145	10.10.5.45	SMB Name Wildcard
4304	10.10.16.5	10.10.162.228	ICMP D estination Unreachable (Communication Administratively Prohibited)
2952	216.158.21.226	10.10.137.7	ICMP Destination Unreachable (Communication Administratively Prohibited)
2420	10.10.14.1	10.10.162.228	ICMP Destinati on Unreachable (Communication Administratively Prohibited)
2313	198.32.224.31	10.10.140.9	ICMP Destination Unreachable (Host Unreachable)

Percentage and number of attacks from a host to a destination

%	# of attacks	from	to
8.06	9316	10.10.162.228	204.152.190.70
7.97	9218	216.150.152.145	10.10.5.4 4
7.52	8693	216.150.152.145	10.10.5.45
5.15	5958	10.10.16.5	10.10.115.155
3.72	4304	10.10.16.5	10.10.162.228
2.55	2952	216.158.21.226	10.10.137.7
2.09	2420	10.10.14.1	10.10.162.228

2.00	2313	198.32.224.31	10.10.140.9
1.92	2217	10.10.14.1	10.10.5.4
1.48	1717	216.158.21.42	10.10.137.7

Percentage and number of attacks from one host to any
with same method

%	# of attacks	from	method
10.31	11920	10.10.16.5	ICMP Destination Unreachable (Communication Administratively Prohibited)
8.06	9316	10.10.162.228	ICMP Echo Request Nmap or HPING2
7.77	8990	216.150.152.145	ICMP Echo Request L3retriever Ping
7.71	8921	216.150.152.145	SMB Name Wildcard
4.60	5318	10.10.14.1	ICMP Destination Unreachable (Communication Administratively Prohibited)
2.55	2952	216.158.21.226	ICMP Destination Unreachable (Communication Administratively Prohibited)
2.00	2313	198.32.224.31	ICMP Destination Unreachable (Host Unreachable)
1.48	1717	216.158.21.42	ICMP Destination Unreachable (Communication Administratively Prohibited)
1.31	1516	205.188.160.165	MISC Large UDP Packet
0.85	979	134.93.19.12	MISC source port 53 to <1024

Percentage and number of attacks to one certain host

%	# of attacks	to	method
16.35	18905	10.10.140.9	MISC traceroute
8.06	9316	204.152.190.70	ICMP Echo Request Nmap or HPING2
5.81	6724	10.10.162.228	ICMP Destination Unreachable (Communication Administratively Prohibited)
5.15	5958	10.10.115.155	ICMP Destination Unreachable (Communication Administratively Prohibited)
4.61	5335	10.10.253.114	WEB -MISC prefix -get //
4.04	4669	10.10.137.7	ICMP Destination Unreachable (Communication Administratively Prohibited)
3.99	4611	10.10.5.44	ICMP Echo Request L3retriever Ping
3.98	4607	10.10.5.44	SMB Name Wildcard
3.79	4379	10.10.5.45	ICMP Echo Request L3retriever Ping
3.74	4322	10.10.1.4	MISC source port 53 to <1024

The distribution of attack methods

%	# of attacks	method

19.46 22507 ICMP Destination Unreachab le (Communication Administratively Prohibited)
5958 10.10.16.5 -> 10.10.115.155
4304 10.10.16.5 -> 10.10.162.228
2952 216.158.21.226 -> 10.10.137.7
2420 10.10.14.1 -> 10.10.162.228
2217 10.10.14.1 -> 10.10.5.4
1717 216.158.21.42 -> 10.10.137.7
652 10.10.16.5 -> 10.10.115.178
483 192.80.53.46 -> 10.10.140.9
269 10.10.16.5 -> 10.10.110.56
159 10.10.16.5 -> 10.10.98.158

16.38 18947 MI SC traceroute
400 129.137.254.5 -> 10.10.140.9
372 129.116.218.196 -> 10.10.140.9
369 134.79.196.42 -> 10.10.140.9
366 128.182.61.50 -> 10.10.140.9
365 205.166.205.222 -> 10.10.140.9
363 128.227.0.107 -> 10.10.140.9
361 198.202.74.41 -> 10.10.140.9
360 198.119.6.13 -> 10.10.140.9
360 18.201.0.122 -> 10.10.140.9
360 138.26.220.46 -> 10.10.140.9

11.31 13081 MISC source port 53 t o <1024
979 134.93.19.12 -> 10.10.130.122
485 204.34.211.86 -> 10.10.1.4
356 204.34.211.86 -> 10.10.1.5
157 204.34.211.81 -> 10.10.1.4
149 204.34.211.82 -> 10.10.1.4
137 204.34.211.87 -> 10.10.1.4
116 204.34.211.81 -> 10.10.1.5
107 204.34.211.82 -> 10.10.1.5
103 204.34.211.87 -> 10.10.1.5
87 204.34.211.86 -> 10.10.1.3

8.50 9830 ICMP Echo Request Nmap or HPING2
9316 10.10.162.228 -> 204.152.190.70
63 10.10.98.183 -> 10.0.3.3
40 10.10.137.7 -> 216.158.50.240
39 10.10.98.162 -> 24.12.12.182
32 10.10.137.7 -> 207.245.122.125
26 10.10.184.37 -> 65.42.135.63
19 10.10.98.239 -> 66.21.168.61
15 10.10.98.205 -> 206.251.6.192
13 10.10.97.242 -> 206.251.6.192
12 10.10.98.162 -> 172.155.124.227

7.87 9102 SMB Name Wildcard
4607 216.150.152.145 -> 10.10.5.44
4314 216.150.152.145 -> 10.10.5.45

12 144.132.19.157 -> 10.10.132.92
11 203.198.242.31 -> 10.10.133.186
10 165.247.240.201 -> 10.10.134.79
7 130.67.134.135 -> 10.10.137.138
7 63.70.217.146 -> 10.10.133.46
6 12.44.129.42 -> 10.10.134.82
6 130.13.149.59 -> 10.10.133.3
6 192.115.218.254 -> 10.10.132.102

7.77 8990 ICMP Echo Request L3retriever Ping
4611 216.150.152.145 -> 10.10.5.44
4379 216.150.152.145 -> 10.10.5.45
6.55 7574 INFO MSN IM Chat data
443 10.10.98.166 -> 64.4.13.122
338 64.4.13.165 -> 10.10.98.158
229 10.10.97.18 -> 64.4.13.194
193 64.4.13.194 -> 10.10.97.18
177 64.4.13.167 -> 10.10.98.143
147 64.4.13.122 -> 10.10.98.166
144 10.10.98.112 -> 64.4.13.169
143 10.10.97.241 -> 64.4.13.116
134 64.4.13.136 -> 10.10.97.52
119 10.10.97.225 -> 64.4.13.118

4.90 5672 spp_http_decode
4.62 5344 WEB-MISC prefix-get //
270 172.164.232.120 -> 10.10.253.114
241 12.78.236.101 -> 10.10.253.114
188 203.197.187.204 -> 10.10.253.114
174 63.22.24.41 -> 10.10.253.114
172 64.105.100.88 -> 10.10.253.114
155 65.80.148.149 -> 10.10.253.114
150 144.59.13.2 -> 10.10.253.114
150 65.165.88.155 -> 10.10.253.114
126 66.44.113.19 -> 10.10.253.114
126 24.4.252.27 -> 10.10.253.114

3.54 4088 ICMP Destination Unreachable (Host Unreachable)
2313 198.32.224.31 -> 10.10.140.9
299 205.171.1.26 -> 10.10.70.174
87 63.146.1.33 -> 10.10.70.11
57 203.63.176.52 -> 10.10.70.161
35 205.171.1.26 -> 10.10.70.11
34 206.47.121.162 -> 10.10.70.11
20 213.65.88.247 -> 10.10.70.11
14 196.3.153.6 -> 10.10.70.11
13 206.115.151.29 -> 10.10.70.11
12 62.161.0.33 -> 10.10.70.11

Portscans performed to/from HOME_NET

# of attacks	from
343	10.10.100.230
165	10.10.218.94
152	10.10.150.220
126	10.10.253.24
113	10.10.217.26
107	10.10.151.79
72	10.10.97.209
66	10.10.98.183
60	10.10.160.114
51	10.10.70.11

Appendix B – Out of Spec files

There are daily log files for Out of Spec packets, these are packets which aren't created by specification, so this packets must be crafted. These files are for additional research to correlate data.

Sample Out of Spec data:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+= +=+=+=+=+
07/23-09:45:34.125077 194.138.17.98:2000 -> 10.10.60.14:80
TCP TTL:47 TOS:0x0 ID:57419 DF
21S***** Seq: 0x79FF149E   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 2356437 0 EOL EOL EOL EOL
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+= +=+=+=+=+=+=+=+=+=+=+=+=+
07/23-09:54:49.955667 198.186.202.147:40481 -> 10.10.70.113:113
TCP TTL:47 TOS:0x0 ID:38055 DF
21S***** Seq: 0x9EC7A5F0   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 99692887 0 EOL EOL EOL EOL
=+=+=+=+=+=+=+=+=+=+=+= +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
07/23-09:54:59.430238 198.186.202.147:40494 -> 10.10.70.113:25
TCP TTL:47 TOS:0x0 ID:23766 DF
21S***** Seq: 0x9F7969DC   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 99693834 0 EOL EOL EOL EOL
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
07/23-10:01:07.985411 198.186.202.147:40726 -> 10.10.70.113:113
TCP TTL:47 TOS:0x0 ID:45209 DF
[root@ids oos]# tail -n20 oos_Jul.23.2001.gz.vers1
20 00 4E 3B 00 00 02 04 05 B4 01 01             .N;........
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Following anomalies can be detected are:

- SF – SYN FIN flags set
- Static IP ID
- Same TCP sequence number
- Same Source and Destination port

Appendix C - References

Assignment 1 – Network Detects

Detect 1:

Ref. 1: What port numbers do well-known Trojan horses use? – SANS Institute
http://www.sans.org/newlook/resources/IDFAQ/oddports.ht m

Ref. 2: NetBus - Chris A. Hayden December 17, 2000
http://www.sans.org/infosecFAQ/malicious/netbus.htm

Ref. 3: Subseven Trojan Summary - Kelly Kester December 19, 2000
http://www.sans.org/infosecFAQ/malicious/subseven3.htm

Detect 2:

Ref. 4: arachNIDS – SGI TELNETD FORMAT BUG
http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids304

Detect 4:

Ref. 5: arachNIDS –HTTP-Phorum-Admin
http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids205

Detect 5:

Ref. 6: atachNIDS – HTTP Whisker splicing attack space
http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids296

Ref. 7: Wiretrip – Whisker CGI scanner
http://www.wiretrip.net/rfp/p/doc.asp?id=21&iface=3

Assignment 2 – Describe the state of Intrusion Detection

Ref. 8: securityfocus.com – Multiple Linux Vendor 2.2.x Kernel IP Masquerading Vuln.
http://www.securityfocus.com/vdb/bottom.html?vid=1078

Ref. 9: Release Notes for Linux kernel 2.2.15
http://www.linux.org.uk/VERSION/relnotes.2215.html

Ref. 10: CVE-2000-0289
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0289

Ref. 11: TCP/IP Illustrated, Volume 1 – The Protocols by W. Richard Stevens
(Addison Wesley, ISBN: 0-201-63346-9)

Ref. 12: IPFWADM:Linux firewall facilities for kernel-level packet screening
http://www.kulichki.com/moshkow/SECURITY/ipfwadm/paper.txt

Ref. 13: Diplomarbeit – IP NETWORK ADRESS TRANSLATION, Michael Hasenstein, 1997
http://www.suse.de/~mha/linux-ip-nat/diplom/nat.html

Ref. 14: Linux IP Masquerading HOWTO, Version 1.95c
http://www.e-infomax.com/ipmasq/howto/ipmasq-HOWTO-1.95c.html

Assignment 3 – „Analyse This“ Scenario

Ref. 15: Privacy Software Corporation Security Advisory
http://www.nsclean.com/psc-gc.html

Ref. 16: anonymous Trojan download page (should be handled carefully!!)
http://www.multimania.com/cdcorg/trojans.html

Ref. 17: Global Incident Analysis Center - Detects Analyzed 3/11/00
http://www.sans.org/y2k/031100.htm

Ref. 18: Global Incident Analysis Center - Detects Analyzed 5/28/00
http://www.incidents.org/archives/y2k/052800-1130.htm

Ref. 19: Global Incident Analysis Center - Detects Analyzed 02/26/01
http://www.incidents.org/archives/y2k/022601-1400.htm

Ref. 20: What Are Some Of The Signs Of Internet Gaming by analyst Matt Scarborough
http://www.incidents.org/detect/gaming.php

Ref. 21: CAIDA Analysis of Code Red
http://www.caida.org/analysis/security/code-red/

Ref. 22: arachNIDS – traffic from source port 53 to <1024
http://www.whitehats.com/info/ids07

Ref. 23: arachNIDS – HPING2 or NMAP echo request
http://www.whitehats.com/info/ids162

Ref. 24: RFC 792
http://rfc.fh-koeln.de/rfc/html/rfc0792.html

Ref. 25: Serious flaw in Microsoft IIS UNICODE translation
http://www.infowar.com/iwftp/xforce/advise68.shtml

Ref. 26: SnortSnarf alert page of nothing-on.tv
http://www.nothing-on.tv/snarf/20010727/199/183/24/src199.183.24.194.html

Ref. 27: snort_stat.pl download page
http://www.snort.org/downloads.html

Last Updated: July 23rd, 2017

Upcoming Training

SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS vLive - SEC503: Intrusion Detection In-Depth	SEC503 - 201709,	Sep 11, 2017 - Oct 18, 2017	vLive
Baltimore Fall 2017 - SEC503: Intrusion Detection In-Depth	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Scottsdale SEC503	Scottsdale, AZ	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
Community SANS Ottawa SEC503	Ottawa, ON	Oct 16, 2017 - Oct 21, 2017	Community SANS
SANS Berlin 2017	Berlin, Germany	Oct 23, 2017 - Oct 28, 2017	Live Event
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
Community SANS Pensacola SEC503	Pensacola, FL	Nov 27, 2017 - Dec 02, 2017	Community SANS
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZ	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event
SANS Dallas 2018	Dallas, TX	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced