Global Information Assurance Certification Paper

Below are the 5 detects for the SANS Security DC 2000 GIAC Intrusion Detection Curriculum Practical Assignment.
Steven T. Carey

Detect 1

[**] Descending TTL's [**]

TCP DUMP HEX

07:00:33.912522 0:10:7:17:38:c0 0:e0:fe:7c:30:c0 0800 60: 208.27.69.128.15358 > a.b.c.d.31933: R 0:0(0) ack 1 win 0
4500 0028 147c 0000 f606 d2ae d01b 4580
xxxx xxxx 3bfe 7cbd 0000 0000 ad85 b072
5014 0000 bb77 0000 d18b 9bc7 9ba1

** Break

07:00:33.953056 0:e0:fe:7c:30:c0 0:10:7:17:38:c0 0800 60: 208.27.69.128.15358 > a.b.c.d.31933: R 0:0(0) ack 1 win 0 [ttl 1]
4500 0028 147c 0000 0106 c7af d01b 4580
xxxx xxxx 3bfe 7cbd 0000 0000 ad85 b072
5014 0000 bb77 0000 d18b 9bc7 9ba1

TCP DUMP

Packet 2
IP Header

Version:	4
Header Length:	20 bytes
Service Type:	0x00
Datagram Length:	40 bytes
Identification:	0x147C
Flags:	MF=off, DF=off
Fragment Offset:	0
TTL:	246
Encapsulated Protocol:	TCP
Header Checksum:	0xD2AE
Source IP Address:	208.27.69.128
Destination IP Address:	a.b.c.d

TCP Header

Source Port:	15358 (<unknown>)
Destination Port:	31933 (<unknown>)
Sequence Number:	0000000000

Acknowledgement Number:	2911219826
Header Length:	20 bytes (data=0)
Flags:	URG=off, ACK=on, PSH=off RST=on, SYN=off, FIN=off
Window Advertisement:	0 bytes
Checksum:	0xBB77
Urgent Pointer:	0

TCP Data

<No data>

** Break

Packet 247

IP Header

Version:	4
Header Length:	20 bytes
Service Type:	0x00
Datagram Length:	40 bytes
Identification:	0x147C
Flags:	MF=off, DF=off
Fragment Offset:	0
TTL:	1
Encapsulated Protocol:	TCP
Header Checksum:	0xC7AF
Source IP Address:	208.27.69.128
Destination IP Address:	a.b.c.d

TCP Header

Source Port:	15358 (<unknown>)
Destination Port:	31933 (<unknown>)
Sequence Number:	0000000000
Acknowledgement Number:	2911219826
Header Length:	20 bytes (data=0)
Flags:	URG=off, ACK=on, PSH=off RST=on, SYN=off, FIN=off
Window Advertisement:	0 bytes
Checksum:	0xBB77
Urgent Pointer:	0

TCP Data

<No data>

---

1. **Source of Trace**:

My Network

2. **Detect was generated by**:

A. Shadow IDS

B. Format - TCP DUMP HEX:

```
07:00:33.911394  0:e0:fe:7c:30:c0 0:10:7:17:38:c0 0800 60:  208.27.69.128.15358 > a.b.c.d.31933: R      0:0     (0)      ack 2911219826 win 0
|    Timestamp      | MAC Address of last system                  |  Source IP       | Source |  Dest. | Dest. | TCP | Seq. | data  | ack seq. no.        | window |
                                                                                   |  Port    |  IP      | Port  | Flags|  No.  | field|                          |  size      |
                                                                                                                | set    |

                                              4500 0028 147c 0000 f706 d1ae d01b 4580
                                              |                      HEX                              |
```

3. **Probability the source address was spoofed** :

Most likely spoofed.   This detect came from a Reset / Ack trace.

4. **Description of attack**:

Attack is a false positive. Attack is actually a mis-configured router on my network.   There is no CVE number for a false positive.

5. **Attack mechanism**:

The Reset / Ack comes into the network to multiple addresses, when the Reset / Ack hits a non-existent IP address, on the internal mis-configured router, the router loops the packet decrementing the TTLs until zero.   The TCP dump shows all the data identical. The TCP dump hex shows the hex is the same until hex 40 and above, then the hex changes from packet one and packet two. Then the hex on the second through the last packets are the same.   This "attack" actually causes a Denial of Service, because of the resources the router uses to loop the packet.

6. **Correlation's**:

We had seen this on an increasing basis and was driving us crazy. Even posted it on the GIAC web page. Thanks to Donald McLachlan who pointed us in the right direction. To correlate the activity we took an IP Address that had descending TTLs. Then we ran a traceroute on it and saw the loop. Since only one router on my network looped the packets, it was difficult to determine that it was a false positive.

7. **Evidence of active targeting**:

N/A. False positive

8. **Severity**:

(2+1)-(2+2)= -1

9. **Defensive Recommendations**:

Defenses are marginal. Internal router is mis-configured.

10. **Multiple choice question**:

```
07:00:33.911394 lernd.to.wright.at.scool.org.15358 > a.b.c.d.31933: R 0:0(0) ack 2911219826 win 0
07:00:33.912522 lernd.to.wright.at.scool.org.15358 > a.b.c.d.31933: R 0:0(0) ack 2911219826 win 0
**Break**
07:00:33.953056 lernd.to.wright.at.scool.org.15358 > a.b.c.d.31933: R 0:0(0) ack 2911219826 win 0 [ttl 1]
```

The above trace is an example of

A. Port scan
B. Buffer Overflow attempt
C. Mis-configured router
D. ICQ session

Answer: c

Detect 2

[**] Ring0 Scan [**]

TCP DUMP HEX

```
01:57:30.316508 0:e0:fe:7c:30:c0 0:10:7:17:38:c0 0800 62: 195.222.17.214.1225 > a.b.c.98.80:
S 242593:242593(0) win 8192 <mss 1460,nop,nop,sackOK> (DF)
                  4500 0030 2303 4000 6d06 1a7d c3de 11d6
                  XXXX XX62 04c9 0050 0003 b3a1 0000 0000
                  7002 2000 da19 0000 0204 05b4 0101 0402
01:57:30.323012 0:10:7:17:38:c0 0:e0:fe:7c:30:c0 0800 60: a.b.c.98.80 > 195.222.17.214.1225:
S 3212993822:3212993822(0) ack 242594 win 33580 <mss 1460>
                  4500 002c 5adf 0000 3c06 53a5 XXXX XX62
                  c3de 11d6 0050 04c9 bf82 651e 0003 b3a2
                  6012 832c 6742 0000 0204 05b4 0000
01:57:30.836267 0:10:7:17:38:c0 0:e0:fe:7c:30:c0 0800 60: a.b.c.98.80 > 195.222.17.214.1225:
F 163:163(0) ack 358 win 33580
                  4500 0028 5af2 0000 3c06 5396 XXXX XX62
                  c3de 11d6 0050 04c9 bf82 65c1 0003 b507
                  5011 832c 7cf7 0000 0000 0000 0000
01:57:31.022349 0:e0:fe:7c:30:c0 0:10:7:17:38:c0 0800 60: 195.222.17.214.1225 > a.b.c.98.80:
F 358:358(0) ack 164 win 8598 (DF)
                  4500 0028 6803 4000 6d06 d584 c3de 11d6
                  XXXX XX62 04c9 0050 0003 b507 bf82 65c2
                  5011 2196 de8c 0000 696c 746f 3a42
01:57:31.025790 0:e0:fe:7c:30:c0 0:10:7:17:38:c0 0800 62: 195.222.17.214.1272 > a.b.c.98.8080:
S 243323:243323(0) win 8192 <mss 1460,nop,nop,sackOK> (DF)
                  4500 0030 6903 4000 6d06 d47c c3de 11d6
                  XXXX XX62 04f8 1f90 0003 b67b 0000 0000
                  7002 2000 b7d0 0000 0204 05b4 0101 0402
01:57:31.028547 0:10:7:17:38:c0 0:e0:fe:7c:30:c0 0800 60: a.b.c.98.8080 > 195.222.17.214.1272:
S 3213124511:3213124511(0) ack 243324 win 33580 <mss 1460>
                  4500 002c 5af4 0000 3c06 5390 XXXX XX62
                  c3de 11d6 1f90 04f8 bf84 639f 0003 b67c
                  6012 832c 4676 0000 0204 05b4 0000
01:57:31.293065 0:10:7:17:38:c0 0:e0:fe:7c:30:c0 0800 60: a.b.c.98.8080 > 195.222.17.214.1272:
F 1:1(0) ack 360 win 33580
                  4500 0028 5af5 0000 3c06 5393 XXXX XX62
                  c3de 11d6 1f90 04f8 bf84 63a0 0003 b7e3
                  5011 832c 5ccb 0000 0000 0000 0000
01:57:31.536206 0:e0:fe:7c:30:c0 0:10:7:17:38:c0 0800 60: 195.222.17.214.1272 > a.b.c.98.8080:
F 360:360(0) ack 2 win 8760 (DF)
                  4500 0028 8003 4000 6d06 bd84 c3de 11d6
```

```
                  XXXX XX62 04f8 1f90 0003 b7e3 bf84 63a1
                  5011 2238 bdbe 0000 3201 0000 4875
01:57:31.538133 0:e0:fe:7c:30:c0 0:10:7:17:38:c0 0800 62: 195.222.17.214.1273 > a.b.c.98.3128:
S 243835:243835(0) win 8192 <mss 1460,nop,nop,sackOK> (DF)
                  4500 0030 8103 4000 6d06 bc7c c3de 11d6
                  XXXX XX62 04f9 0c38 0003 b87b 0000 0000
                  7002 2000 c927 0000 0204 05b4 0101 0402
01:57:31.733803 0:e0:fe:7c:30:c0 0:10:7:17:38:c0 0800 60: 195.222.17.214.1272 > a.b.c.98.8080:
R 243684:243684(0) win 0
                  4500 0028 8203 0000 6d06 fb84 c3de 11d6
                  XXXX XX62 04f8 1f90 0003 b7e4 0000 0000
                  5004 0000 0329 0000 5655 382a 223a
01:57:34.502406 0:e0:fe:7c:30:c0 0:10:7:17:38:c0 0800 62: 195.222.17.214.1273 > a.b.c.3128:
S 243835:243835(0) win 8192 <mss 1460,nop,nop,sackOK> (DF)
                  4500 0030 2404 4000 6d06 197c c3de 11d6
                  XXXX XX62 04f9 0c38 0003 b87b 0000 0000
                  7002 2000 c927 0000 0204 05b4 0101 0402
01:57:40.531556 0:e0:fe:7c:30:c0 0:10:7:17:38:c0 0800 62: 195.222.17.214.1273 > a.b.c.98.3128:
S 243835:243835(0) win 8192 <mss 1460,nop,nop,sackOK> (DF)
                  4500 0030 6704 4000 6d06 d67b c3de 11d6
                  XXXX XX62 04f9 0c38 0003 b87b 0000 0000
                  7002 2000 c927 0000 0204 05b4 0101 0402
01:57:52.522063 0:e0:fe:7c:30:c0 0:10:7:17:38:c0 0800 62: 195.222.17.214.1273 > a.b.c.98.3128:
S 243835:243835(0) win 8192 <mss 1460,nop,nop,sackOK> (DF)
                  4500 0030 8404 4000 6d06 b97b c3de 11d6
                  XXXX XX62 04f9 0c38 0003 b87b 0000 0000
                  7002 2000 c927 0000 0204 05b4 0101 0402
```

1. **Source of Trace**:

   My Network

2. **Detect was generated by**:

   A. Shadow IDS

   B. Format - TCP DUMP HEX:

01:57:52.522063 0:e0:fe:7c:30:c0 0:10:7:17:38:c0: 195.222.17.214.1273 > a.b.c.98.31933: S 243835:243835(0) win 8192 <mss 1460,nop,nop,sackOK> (DF)

Timestamp	MAC Address of last system hop and destination system	Source IP	Source Port	Dest. IP	Dest. Port	TCP Flag set	Seq. No.	data field	window size	TCP Options	Don't Fragment

4500 0030 8404 4000 6d06 b97b c3de 11d6
HEX

3. **Probability the source address was spoofed** :

Unlikely the source address was spoofed.

4. **Description of attack**:

Scan for ports http (port 80), http proxy (port 8080), and squid proxy (port 3128), looking for proxies. There is no CVE number for Ring0.

5. **Attack mechanism**:

The source address sends a SYN packet to http (port 80), http proxy (port 8080), and squid proxy (port 3128) on a target system. If all three ports answer back with SYN/ACKs, then the target system most likely has the Ring0 Trojan installed. The Ring0 Trojan has two files (its.exe and pst.exe). The its.exe attempts to retrieve files from the webserver. The its.exe uses a its.dat file. This dat file is configurable and would allow an attacker to either scan or attack a target. An attacker could have the its program retrieve files such as financial or research and development. The pst.exe is an active scanner. It generates a 'random' range of IP addresses that it scans for ports 80, 8080, and 3128. When it discovers the proxies, the program send those IP addresses to www.rusftpsearch.net.

6. **Correlation's**:

This type of scan was first identified on 30 Sep 99. This type of scan has been reported numerous times to SANS GIAC (www.sans.org/giac.htm).

7. **Evidence of active targeting**:

Source address goes after specific host.

8. **Severity**:

(4+3)-(5+2)= 0

9. **Defensive Recommendations**:

Defense are fine, to date no evidence that port 3128 is currently being used. The Ring0 scans help us look for systems on My Network that are running proxies and can close them down.

10. **Multiple choice question**:

02:31:16.072407 ramax.spb.ru.62842 > a.b.c.14.3128: S 2668173:2668173(0) win 8192 (DF) [tos 0x60]

The above trace is an example of

A. Port scan
B. Buffer Overflow attempt
C. Mis-configured router
D. Ring0 Trojan scan

Answer: d

Detect 3

[**] Scan for FTP software [**]

TCP DUMP HEX

```
19:18:33.500899 0:e0:fe:7c:30:c0 0:10:7:17:38:c0 0800 60: 64.39.38.102.21 > a.b.c.125.21:
SF 1326956210:1326956210(0) win 1028
                  4500 0028 9a02 0000 2206 6d75 4027 2666
                  xxxx xx7d 0015 0015 4f17 beb2 79f2 f7c3
                  5003 0404 9ada 0000 0057 0000 0600
19:18:33.510567 0:10:7:17:38:c0 0:e0:fe:7c:30:c0 0800 60: a.b.c.125.21 > 64.39.38.102.21:
S 3445633806:3445633806(0) ack 1326956211 win 1014 <mss 536>
                  4500 002c 0081 0000 fc06 2cf2 xxxx xx7d
                  4027 2666 0015 0015 cd60 330e 4f17 beb3
```

```
                6012 03f6 f7ff 0000 0204 0218 7e7e
```

TCP DUMP

```
-----------------------------------------------------------------
Packet 12631
IP Header
      Version:                    4
      Header Length:              20 bytes
      Service Type:               0x00
      Datagram Length:            40 bytes
      Identification:             0x9A02
      Flags:                      MF=off, DF=off
      Fragment Offset:            0
      TTL:                        34
      Encapsulated Protocol:      TCP
      Header Checksum:            0x6D75
      Source IP Address:          64.39.38.102
      Destination IP Address:     a.b.c.125
TCP Header
      Source Port:                21 (ftp)
      Destination Port:           21 (ftp)
      Sequence Number:            1326956210
      Acknowledgement Number:     2045966275
      Header Length:              20 bytes (data=0)
      Flags:                      URG=off, ACK=off, PSH=off
                                  RST=off, SYN=on,  FIN=on
      Window Advertisement:       1028 bytes
      Checksum:                   0x9ADA
      Urgent Pointer:             0
TCP Data
      <No data>
-----------------------------------------------------------------
Packet 12807
IP Header
      Version:                    4
      Header Length:              20 bytes
      Service Type:               0x00
      Datagram Length:            75 bytes
      Identification:             0x0086
      Flags:                      MF=off, DF=off
      Fragment Offset:            0
```

```
      TTL:                          252
      Encapsulated Protocol:        TCP
      Header Checksum:              0x2CCE
      Source IP Address:            a.b.c.125
      Destination IP Address:       64.39.38.102
TCP Header
      Source Port:                  21 (ftp)
      Destination Port:             3970 (<unknown>)
      Sequence Number:              3446325390
      Acknowledgement Number:       0708997876
      Header Length:                20 bytes (data=35)
      Flags:                        URG=off, ACK=on,  PSH=on
                                    RST=off, SYN=off, FIN=on
      Window Advertisement:         1014 bytes
      Checksum:                     0x2994
      Urgent Pointer:               0
TCP Data
      421 Inactivity
      <*** Rest of data missing from packet dump ***>
```

---

1. **Source of Trace**:

   My Network

2. **Detect was generated by**:

   A. Shadow IDS

   B. Format - TCP DUMP HEX:

19:18:33.500899 0:e0:fe:7c:30:c0 0:10:7:17:38:c0: 64.39.38.102 .21> a.b.c.125. .31933: S 1326956210:1326956210 (0) win 1014 <mss 536>

Timestamp	MAC Address of last system hop and destination system	Source IP	Source Port	Dest. IP	Dest. Port	TCP Flag set	Seq. No.	data field	window size	TCP Options

4500 0028 9a02 0000 2206 6d75 4027 2666

HEX

3. **Probability the source address was spoofed** :

Unlikely the source address was spoofed.

4. **Description of attack**:

Scan for FTP software, using an automated tool.

The Common Vulnerabilities and Exposures (CVE) that pertain to vulnerable FTP software are:

CVE-1999-0219	CVE-1999-0914	CVE-1999-0707	CVE-1999-0185
CVE-1999-0097	CVE-1999-0079	CVE-1999-0349	CVE-1999-0368
CVE-1999-0777	CVE-1999-0878	CVE-1999-0201	CVE-1999-0202
CVE-1999-0302	CVE-1999-0017	CVE-1999-0035	CVE-1999-0080
CVE-1999-0054	CVE-1999-0351	CVE-1999-0082	CVE-1999-0083

5. **Attack mechanism**:

Source address sends SYN/FIN packets to every address in a subnet. Any FTP servers either send back a Reset packet or a SYN/Ack packet. The target systems, that answer with a SYN/Ack packet, will also provide their FTP software and version number. Then the source address has a list of FTP servers, their software and version number. This allows the source address to attack FTP servers using one or more of the vulnerabilities found in the scan.

6. **Correlation's**:

This type of scan has been reported numerous time to SANS GIAC (www.sans.org/giac.htm).

7. **Evidence of active targeting**:

General scan of entire network.

8. **Severity**:

(4+5)-(3+2)= 4

9. **Defensive Recommendations**:

Defenses are marginal. Numerous FTP servers sent a Reset packet back to the source address, however, some FTP servers sent their FTP software and version. A check of those FTP servers revealed that some of them had old unpatched FTP software.

10. **Multiple choice question**:

19:18:33.500899 0:e0:fe:7c:30:c0 0:10:7:17:38:c0 0800 60: 64.39.38.102.21 > a.b.c.1.21:
SF 1326956210:1326956210(0) win 1028
4500 0028 9a02 0000 2206 6d75 4027 2666
xxxx xx01 0015 0015 4f17 beb2 79f2 f7c3
5003 0404 9ada 0000 0057 0000 0600

19:18:33.500899 0:e0:fe:7c:30:c0 0:10:7:17:38:c0 0800 60: 64.39.38.102.21 > a.b.c.254.21:
SF 1326956210:1326956210(0) win 1028
4500 0028 9a02 0000 2206 6d75 4027 2666
xxxx xxFE 0015 0015 4f17 beb2 79f2 f7c3
5003 0404 9ada 0000 0057 0000 0600

The above trace is an example of

A. Port scan
B. Buffer Overflow attempt
C. Telnet connection
D. ICQ session

Answer: a

Detect 4

[**] HTTP UNIX PASSWORDS[**]
1st Trace
==========

```
Source      = 132.254.192.90 -- academ03.sin.itesm.mx
Destination =  a.b.c.84 -- (Unknown)
Start time  = Wed May 17 00:30:58 2000
Protocols   = [4603 80] (6)
Stream      = conn.000517:00.5024.stream.init
=============================================================

GET /cgi-bin/htsearch?exclude=%60/etc/passwd%60 HTTP/1.0
Host: a.b.c.84
User-Agent: Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)

2nd Trace
=============================================================
Source      =  a.b.c.84 -- (Unknown)
Destination = 132.254.192.90 -- academ03.sin.itesm.mx
Start time  = Wed May 17 00:30:58 2000
Protocols   = [80 4603] (6)
Stream      = conn.000517:00.5024.stream.dest
=============================================================

HTTP/1.0 404 Object Not Found (The system cannot find the path specified. )
Content-Type: text/html

<body><h1>HTTP/1.0 404 Object Not Found (The system cannot find the path specified. )
</h1></body>
 [***** End of stream *****]
```

---

1. **Source of Trace**:

   My Network

2. **Detect was generated by**:

   A. Network Intrusion Detection (NID)

   B. Format: 1st trace is from the target's point of view. 2nd trace is from the attacker's point of view.

3. **Probability the source address was spoofed** :

Unlikely the source address was spoofed.

4. **Description of attack**:

Attack against UNIX system password via web (HTTP) server.

The Common Vulnerabilities and Exposures (CVE) that pertain to HTTP vulnerabilities are

CVE-1999-0867	CVE-1999-0437	CVE-1999-0744	CVE-1999-0267
CVE-1999-0448	CVE-1999-0071	CVE-1999-0236	

5. **Attack mechanism**:

The attack works by completing the thee-way handshake, then sending a request for /etc/passwd file, via one of the CGI scripts installed on the web server. If the UNIX system finds the /etc/passwd file, it will provide the file to the attacker. Then the attacker can begin a brute force attack of all passwords on that system.

6. **Correlation's**:

Have seen scan's for FTP servers on the SANS GIAC page (www.sans.org/giac.htm).

7. **Evidence of active targeting**:

General scan of entire network.

8. **Severity**:

(3+4)-(5+2)= 0

9. **Defensive Recommendations**:

Defenses were fine. All public web servers are stand-a-lone systems, with TCPWrappers and host IDS. Plus the /etc/passwd file is not available on the UNIX systems.

10. **Multiple choice question**:

```
=================================================================
Source      = 132.254.192.90 -- academ03.sin.itesm.mx
Destination =  a.b.c.84 -- (Unknown)
Start time  = Wed May 17 00:30:58 2000
Protocols   = [4603 80] (6)
Stream      = conn.000517:00.5024.stream.init
=================================================================

GET /cgi-bin/htsearch?exclude=%60/etc/passwd%60 HTTP/1.0
Host: a.b.c.84
User-Agent: Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)
```

The above trace is an example of

A. HTTP UNIX Password attempt
B. Buffer Overflow attempt
C. Telnet connection
D. FTP connection

Answer: a

Detect 5

[**] Virus - ILOVEYOU or LOVE-LETTER-FOR-YOU.TXT.vbs[**]

High 5/15/00 8:19:33AM a.b.c.148 1150 206.101.197.226 80
High 5/15/00 8:19:36AM a.b.c.148 1150 206.101.197.226 80
High 5/15/00 8:19:42AM a.b.c.148 1150 206.101.197.226 80
High 5/15/00 8:19:54AM a.b.c.148 1150 206.101.197.226 80

1. **Source of Trace**:

My Network

2. **Detect was generated by**:

A. ISS Real Secure

B. FORMAT:

High 5/15/00 8:19:54AM a.b.c.148 1150 206.101.197.226 80

Threat level	Date/time group	Source IP	Source Port	Destination IP	Destination Port

3. **Probability the source address was spoofed** :

Unlikely the source address was spoofed, since it was one of MY Network IP addresses.

4. **Description of attack**:

Virus/worm spread through email. Specific tool was a .vbs script. Also a password stealing Trojan was downloaded unto a target system. There is no CVE number for viruses are Trojans at this time.

5. **Attack mechanism**:

The .vbs script comes through the email as an attachment. When opened the .vbs script executes. The script has a number of functions.

a. Sends the .vbs script to everyone in address book (for MS Outlook and Outlook Express only).

b. Checks for Winfat32.exe in /Windows/System directory, if not there, re-homes the home page in Internet Explorer to a web site, for MY NETWORK that site was http:// www.QZN.SKYINET.NET. The vbs script will attempt to access the site. Upon successful completion of the three-way handshake, the site downloads two files (WinFAT32.EXE and WIN-BUGSFIX.EXE).

c. Changes registry entries so that MSKernel32.vbs, WIN32DLL.vbs, and WIN-BUGFIX.EXE execute t startup.

d. For each drive, including network drives, the script searches for and replaces all .vbs, .vbe, .js, .jse, .css, .wsh, .sct, .hta, .jpg, .jpeg, .mp2, and .mp3 files and replaces their content with the virus .vbs script.

The WIN-BUGFIX.EXE is a password stealing program. The program will email all cached passwords found to "mailme@super.net.ph".

To remove the program Symantec created a software tool to remove the script.

The script will also spread via ,IRC by creating a script.ini file in the mIRC program directory, which then sends the dropped file Love-letter-for-you.htm to other users in the chatroom.

In addition, caused a Denial of Service on email servers, when hundreds or thousands of emails were generated.

MY NETWORK runs some ISS Real Secure intrusion detection systems. We made a filter to record and stop all web traffic to www.QZN.SKYINET.NET. That helped us determine who went to the web site for the Trojan and who did not. Not everyone who was infected with the virus/worm went to the site. Locally we attribute this to users who had preview open in the email programs. If preview was open, the virus/worm spread, but the system did not attempt to access the web site.

There are currently 29 versions of the LOVE-LETTER-FOR-YOU.TXT.vbs.

6. **Correlation's**:

This was reported numerous times on the SANS GIAC page (www.sans.org/giac.htm). There is also a write-up about it on www.sans.org/y2k/iloveyou_worm.htm.

7. **Evidence of active targeting**:

Attacker targeted Microsoft Exchange servers. Other email systems could deliver the virus/worm, but it was designed to flourish in a MS Exchange environment.

8. **Severity**:

(4+4)-(4+2)= 2

9. **Defensive Recommendations**:

Because there is no way to have Anti-virus software find viruses prior to being spread, it takes education of users to not open up attachments, or emails that seem suspicious, especially when the receive 10 or more with the same subject line.

10. **Multiple choice question**:

High 5/15/00 8:19:33AM xxx.xx.xxx.148 1150 206.101.197.226 80

The above trace is an example of

A. Raptor Firewall
B. ISS Real Secure IDS
C. SNORT IDS
D. Shadow IDS

Answer: b

Below is assignment 2, evaluation of an attack, for the SANS Security DC 2000 GIAC Intrusion Detection Curriculum Practical Assignment.
Steven T. Carey

EVALUATION OF AN ATTACK:

SUBSEVEN TROJAN (AKA BACKDOOR-G)

1. There are several sites that have the Trojan called SubSeven. One of the sites is www.networkice.com/Phauna/RATs/SubSeven/default.htm. They have a link to a site to download SubSeven.

2. SubSeven was written by an individual known as MobMan.

3. SubSeven is a Win32 Trojan. The current version is 2.1. When the infected file is run, SubSeven copies itself to \Windows\ directory with one of the following names:

"server.exe" (328kb)
"rundell6.exe" (328kb)
"systray.dl" (328kb)
"Task_bar.exe" (328kb)

4. Then it copies a file to \Windows\System\ directory with one of the following names:

"FAVPNMCFEE.dll" (35kb)
"MVOKH_32.dll (35kb)
"nodll.exe" (35kb)
"watching.dll" (35kb)

5. The server portion of SubSeven can be configured by the hacker to rerun itself everytime the system is rebooted due to an entry in one of four locations.

a. The first location is an entry on the "shell" line in the SYSTEM.INI file. (This is the location it normally would reside).

b. The second location is an entry on the "load=" or "run=" line in the WIN.INI file.

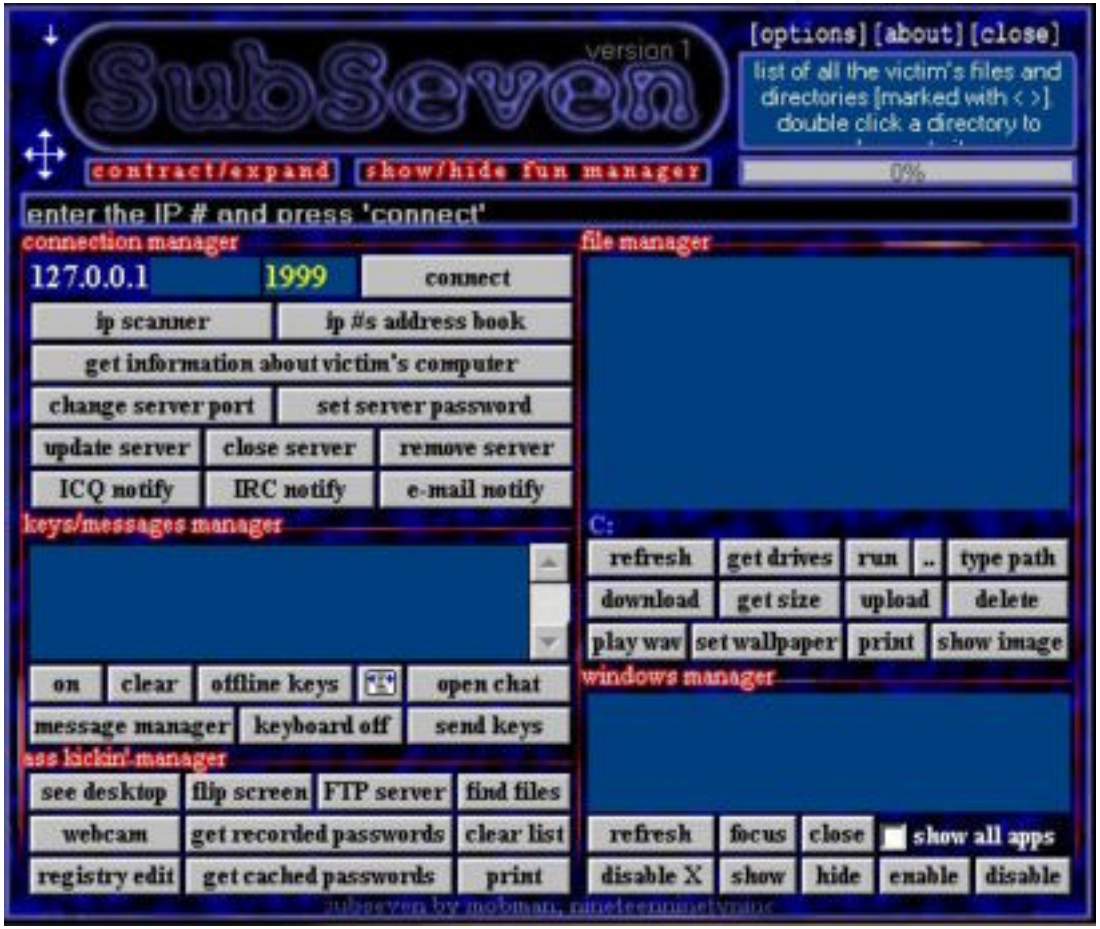
c. The third location is an entry under the "HKEY_LOCAL_MACHINE\software\Microsoft\Windows\CurrentVersion\Run".

d. The fourth location is an entry under the "HKEY_LOCAL_MACHINE\software\Microsoft\Windows\CurrentVersion\runServices".

6. Currently (as of 2000-05-20) the following TCP ports are used by SubSeven:

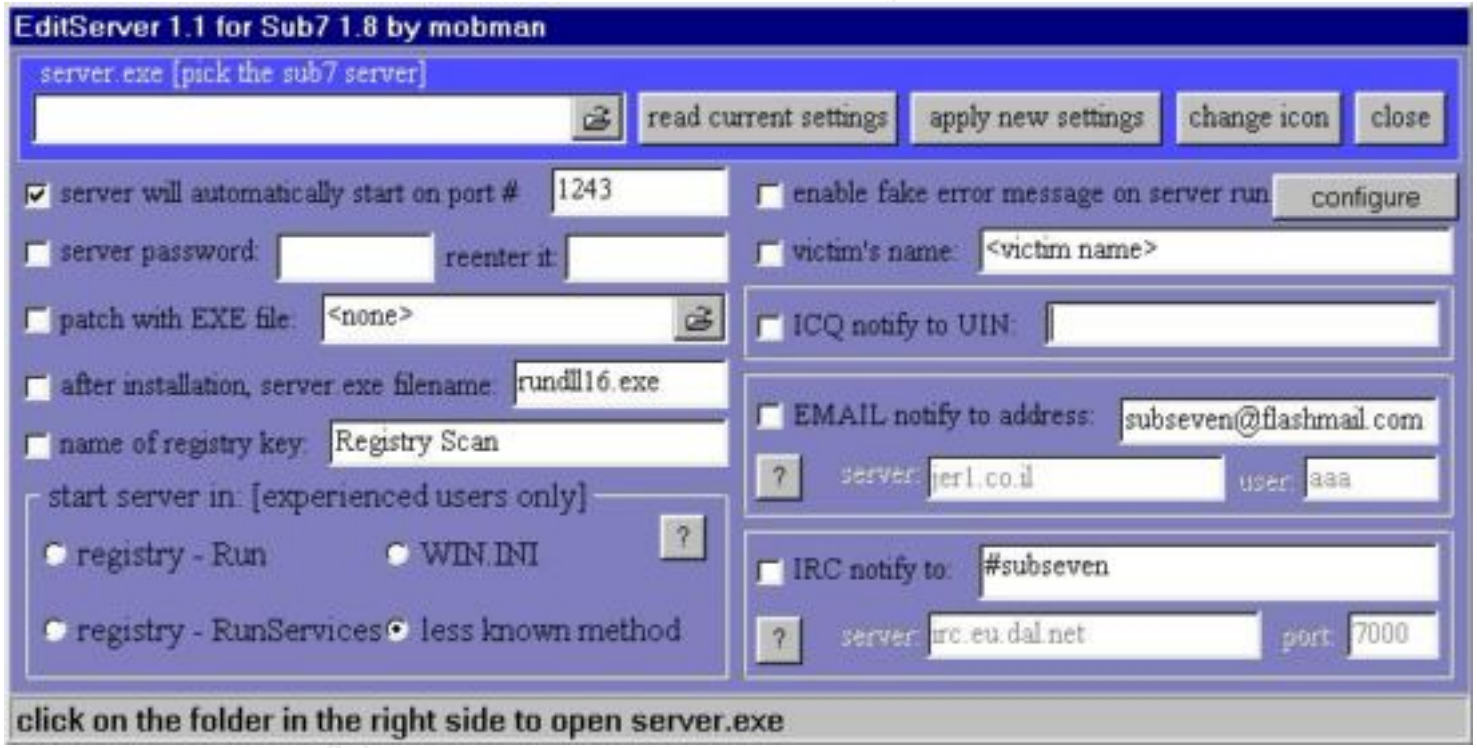
1243	6776	27374
2773	7000	27573
6711	7215	54283

7. Below is a screen shot of the "client" portion of SubSeven.



8. Below is a screenshot of information obtained by the client portion after it attached to a PC that was compromised with the server portion. This reflects information about the compromised system.

9. Below is a screenshot of the "EditServer" utility. This is the utility that allows the hacker to customize the "server" portion of SubSeven. After the server part of the SubSeven has been configured, it's sent to the victim...



10. Below are a list of some of the features for SubSeven.

    a. Show Folder/File names and navigate.

    b. Remote IP scanner (scan an IP range using the victim bandwidth and IP).

    c. Set/Remove Server Password.

    d. List recorded passwords.

    e. List cached passwords.

f. Registry Editor.

g. List drives.

h. Download files.

i. Upload files.

j. Edit files.

11. There are two ways to be infected by SubSeven.

a. One way is if you are already infected with a Trojan or the victim has FTP running. If infected with another Trojan, the attacker opens up the victim as a FTP server. Then using any FTP client, such as CuteFtp, upload the current version of SubSeven. After uploading SubSeven, the attacker attempts to execute SubSeven. If the victim is already infected with another Trojan, this is a simple process, however, if the victim is not infected with another Trojan, but has FTP running, then the following process takes place. The attacker copies the Subseven Server to the victim's C:\WINDOWS\START MENU\PROGRAMS\STARTUP folder. By doing this , the next time the victim starts up the windows, the SubSeven server will get executed , since it is in the startup folder. Then the attacker can access the victim's computer via the highly advanced SubSeven Client. Requires for the attacker to wait until the system is re-booted. In this case the attacker will not bind SubSeven to any other program and will use the "raw" server, which will execute silently and get installed without grabbing the victim's attention.

b. The second way is when the attacker binds the SubSeven "server" to a file or program using "The New Joiner" program. This is a program for binding the SubSeven "server" with any sort of program/file. The "server" can be bond to a .Wav , .BMP, .JPG , .EXE file etc. But the result will ALWAYS be storedin EXE format. Then changing the icon to something that looks innocent. Or, if f you have ever used Winzip Self Extractor then you may have seen a option "Run command after unzipping" in it while making a SFX archive. The attacker can place the SubSeven "server" inside a Winzip SFX Archive and configure the self extractor to run the "server file.exe" after the SFX unzips itself automatically. Then the attacker emails, or transfers through ICQ, the Trojanized program/file to the intended victim. When the victim opens the Trojanized file or runs the Trojanized program, Subseven will install itself. Then the attacker checks the victim's computer for the SubSeven "client".

12. How to remove the SubSeven Trojan. Because the server portion of the SubSeven Trojan can be configured to be loaded automatically from one of four locations, you'll need to look at all of the locations first. Keep in mind that several steps involve examining and possibly editing the registry. Although the steps are relatively easy, I cannot be held responsible if a mistake is made. Please use caution.

a. The first and second locations - The WIN.INI and SYSTEM.INI files

Step 1.
Click START | RUN
Type SYSEDIT and press ENTER

Step 2.
Click on the SYSTEM.INI file and look at the "shell=Explorere.exe" line under the [boot] section. There shouldn't be anything to the right of it. However, if yours looks like "shell=Explorer.exe Task_Bar.exe", then Task_Bar.exe is the server portion of
the Trojan.

Delete Task_Bar.exe from the line, save the change. Skip to the END.

Step 3.
Click on the WIN.INI file and look at the run= and load= lines under the [windows] section. Because it is common to have legitimate programs on either of these lines. You should look at the name of the file that appears on the line and compare it to those above.

If you find one, delete it from the line, save the change. Skip to the END

b. The third and fourth locations - The Registry

Step 1.
Click START | RUN
Type REGEDIT and press ENTER

Step 2.
In the left window, click the "+" (plus sign) to the left of the following:
HKEY_LOCAL_MACHINE
Software
Microsoft
Windows
CurrentVersion
Run

Step 3.
In the right window, look for a key that has a Value that loads one of the files listed above. If you don't find a file as listed above, it might mean that the server portion was renamed to something else. Note the names of any suspicious files.

What you will need to do, is open Windows Explorer and go to the WINDOWS directory. Locate each of the suspicious files that were referenced within the right window of regedit. When you find the file that's 328Kb in size. You've probably found the renamed server portion of SubSeven.

Step 4.
Return to the registry and in the right window, highlight the key that loads the file and hit the DELETE key. Answer YES to delete the entry.

Step 5.
Exit the Registry and reboot your computer.

Step 6.
After the computer has restarted, open Windows Explorer

Step 7.
Go to the WINDOWS directory and look for the suspicious file. Once you've found the file, DELETE it.

Step 8.
Exit Windows Explorer.

c. Congratulations! SubSeven has been removed.

13. Intrusion Detection (SHADOW) Trace for SubSeven.

00:10:51.788227 ph33r.g0d.co.uk.pop-3 > a.b.c.83.1243: S 755242890:755242890(0) ack 674711610 win 16384 (DF)
00:10:53.164849 ph33r.g0d.co.uk.pop-3 > a.b.c.83.1243: R 755242891:755242891(0) ack 674711610 win 16384 (DF)
00:20:46.311325 ph33r.g0d.co.uk.www > d.e.f.87.1142: S 1744258846:1744258846(0) ack 674711610 win 16384 (DF)
00:20:46.312019 d.e.f.87.1142 > ph33r.g0d.co.uk.www: R 674711610:674711610(0) win 0

Below is assignment 3, "Analysis This" Scenario, for the SANS Security DC 2000 GIAC Intrusion Detection Curriculum Practical Assignment.
Steven T. Carey

1. Intrusions Detection findings for MY NET.

A. The following systems may have been compromised with the Back Orifice Trojan and require additional investigation;

MY.NET.5.10
MY.NET.6.34
MY.NET.6.35
MY.NET.6.47
MY.NET.100.37
MY.NET.130.94
MY.NET.253.16
MY.NET.253.24
MY.NET.253.41
MY.NET.253.42
MY.NET.253.43
MY.NET.253.51
MY.NET.253.52
MY.NET.253.53

B. The following systems require additional investigation for possible compromise:

MY.NET.12.53
MY.NET.97.106
MY.NET.97.119
MY.NET.143.87
MY.NET.202.130
MY.NET.203.194

MY.NET.217.86
MY.NET.219.58
MY.NET.253.12

C. Network personnel need to change SMNP community string from "public" to something different. An easily guessed SMNP community string is a vulnerability. It allows an attacker to identify network configurations.

D. SMB Wildcard [**] from IP 166.90.30.149 to IP MY.NET.100.130 on Port 137 (NETBIOS Name Service).

E. Need to place an firewall on the perimeter and block unneeded ports and services into the site. Also to block IP addresses that attempt to intrude into the site.

F. Probable nmap fingerprint attempt to IP MY.NET.203.134, port 2857 (SimCtIP).

G. Queso fingerprint of IP MY.NET.20.10 on port 27005 (flex-lm).

Assignment #4

Below is the summary analysis of all attacks detected by laurie@edu and a summary characterization of the ISP Load Balancing detects, for the SANS Security DC 2000 GIAC Intrusion Detection Curriculum Practical Assignment. Steven T. Carey

1. Attack Summary - Laurie@edu has had a multitude of attempts against her site. She has also had a multitude of pre-attack probes and scans. It would have been possible to break down some of the areas, such as RPC Services, and list each individual service that was attacked, e.g., rpcbind dump (), RPC high port access attempt, portmap request, etc. However, to do that would not show the variety of the attacks against Laurie's site.

Also, a top ten list of attacks would not be complete without having a top ten list of pre-attack probes and scans directed against her site.

A list of the top ten attacker families is also included. It would have been more fun to do a top ten list of countries that have 'visited' Laurie. And then break down the U.S. addresses by state.

A. Top Ten Attacks

1. RPC Services

2. FTPd Service

3. Telnetd Service
4. MISC - WinGate - 8080 Attempt (Proxy Service)
5. DNS (Overflows, Zone Transfers, version-query, Iquery)
6. Sendmail Service
7. IMAP Service
8. POP/POP2/POP3 Services
9. NFS Services
10. Counterfiglet (web)

B. Top Ten Pre-Attacks

1. SYN-FIN Scan
2. PING Scan
3. Finger
4. Queso Fingerprint Attempt
5. Port Scan
6. Null Scan
7. NMAP fingerprint Attempt
8. SNMP

9. Network Scan

10. Trojan Scan

C. Top Ten Attacker Address Families

1. Exodus Communications Inc.
   Santa Clara, California

2. @Home Network
   California

3. TerraNet, Inc.
   Massachusetts

4. Bell Global Network Operations
   Canada

5. UUNET - US/ France/ Italy
   Virginia/ France/ Italy

6. Daewon elementary School
   South Korea

7. Chunghwa Telecom Co, Ltd
   Taiwan

8. Link Incorporated (Japan Network Information Center)
   Japan

9. NASK, Research and Academic Network
   Poland

10. Palace Garment Mfg.
    Malaysia

2. Summary Characterization of ISP Load Balancing or Latency Testing

In the 'now!!!' world of the Internet, a lot of ISP's and companies with web sites are conducting latency testing or load balancing to improve performance of their ISP or web sites. Their reasoning is that if they can find a quicker/shorter path for customers than anyone else, those customers will use their ISP or visit their web site, more than any competitor. The ISP or web site normally uses a product like 3dns (www.3dns.com). This product sends packets to port 53 of any Domain Name Server (DNS) that has customers. The product measures the round trip time to the target DNS server to determine how long it takes a packet to travel. It will normally send packets from three consecutive ports to port 53. The trick is...then the ISP or company tries different routes and/or from different servers they own/control. This creates a lot of traffic to DNS servers that look (to the Intrusion Detection world) like malicious activity. Have seen companies that are international conduct load balancing from as many as five different countries to one DNS server, at the same time, trying to find the shortest/fastest route for customers.

Unfortunately, since this looks like malicious activity, analysts tend to classify traffic to DNS servers, that fit the load balancing profile, as normal and ignore it. What happens then is that someone will use that load balancing profile and actually conduct malicious activity against DNS servers. Load balancing emits a tremendous amount of traffic. Then you add the factor that it is not a one time thing, but could be daily, weekly, monthly, and in some cases hourly. This amount of traffic can cause a big blind spot to a normally diligent analyst.

Lauie@edu receives a tremendous amount of load balancing traffic. How she can watch all that traffic and still have time to send all those reports to GIAC, is astounding. In the early part of 2000, Laurie provided a lot of reports to GIAC that ended up as load balancing. What is amazing is the number of ISP's and companies who have told her that it is 'just load balancing' when there is some doubt as to why her users are going to those sites in the first place. One user can generate mountains of load balancing traffic, just from one fleeting visit.

Personally, think that a lot of these ISP's and companies that are conducting load balancing are using these techniques to try and map the Internet. From past involvement with them Exodus Communications has been one of the biggest players in load balancing. Rumors are that Exodus has been mapping the Internet. Guess the first company that can provide an accurate map of the Internet, and keep it updated, could make a lot of money.

In the mean time, the Incident Intrusion Analyst must not develop a blind spot concerning load balancing. Assessments should be made as to how much traffic it is generating at your site and then develop a game plan to periodically look closer at repeated traffic and to look closer when load balancing traffic is first detected from a site. That is the only sane way to determine whether your site is being bombarded with load balancing traffic or malicious traffic (unless you are like me and have no life, then attempt to look at all of it.