Richard D. Salisko

Assignment One - Network Detects

Detect One:

Snort Output

```
[**] SCAN-SYN FIN [**]
08/29-05:26:54.909154 0:E0:39:80:42:6E -> x:x:x:x:x:x type:0x800 len:0x3C
small.fareast.isp:21 -> 172.16.10.2:21 TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0x7E4C527C   ACK: 0x520279BC   Win: 0x404

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] SCAN-SYN FIN [**]
08/29-05:26:54.930123 0:E0:39:80:42:6E -> x:x:x:x:x:x type:0x800 len:0x3C
small.fareast.isp:21 -> 172.16.10.3:21 TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0x7E4C527C   ACK: 0x520279BC   Win: 0x404

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] SCAN-SYN FIN [**]
08/29-05:26:55.168334 0:E0:39:80:42:6E -> FF:FF:FF:FF:FF:FF type:0x800 len:0x3C
small.fareast.isp:21 -> 172.16.10.95:21 TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0x7E4C527C   ACK: 0x520279BC   Win: 0x404

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

[**] SCAN-SYN FIN [**]
08/30-02:14:47.971325 0:E0:39:80:42:6E -> x:x:x:x:x:x type:0x800 len:0x3C
small.fareast.isp:109 -> 172.16.10.2:109 TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0x262180C   ACK: 0x5A762B7A   Win: 0x404

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] SCAN-SYN FIN [**]
08/30-02:14:47.986934 0:E0:39:80:42:6E -> x:x:x:x:x:x type:0x800 len:0x3C
small.fareast.isp:109 -> 172.16.10.3:109 TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0x262180C   ACK: 0x5A762B7A   Win: 0x404

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

[**] SCAN-SYN FIN [**]
08/30-02:14:48.227989 0:E0:39:80:42:6E -> FF:FF:FF:FF:FF:FF type:0x800 len:0x3C
small.fareast.isp:109 -> 172.16.10.95:109 TCP TTL:28 TOS:0x0 ID:39426
SF** Seq: 0x262180C   ACK: 0x5A762B7A   Win: 0x404

=+

Windump Output (Windows NT port of TCPDump)

05:26:54.908820 small.fareast.isp.21 > 172.16.10.2:21: SF 2118931068:2118931068(0) win 1028 (ttl 28, id 39426)
05:26:54.929811 small.fareast.isp.21 > 172.16.10.3.21: SF 2118931068:2118931068(0) win 1028 (ttl 28, id 39426)
05:26:54.930309 172.16.10.3.21 > small.fareast.isp.21: R 0:0(0) ACK 2118931070 win 0 (ttl 128, id 4864)
05:26:55.168002 small.fareast.isp.21 > 172.16.10.95.21: SF 2118931068:2118931068(0) win 1028 (ttl 28, id 39426)

02:14:47.970156 small.fareast.isp.109 > 172.16.10.2.109: SF 39983116:39983116(0) win 1028 (ttl 28, id 39426)
02:14:47.985755 small.fareast.isp.109 > 172.16.10.3.109: SF 39983116:39983116(0) win 1028 (ttl 28, id 39426)
02:14:47.986230 172.16.10.3.109 > small.fareast.isp.109: R 0:0(0) ACK 39983118 win 0 (ttl 128, id 5120)
02:14:48.226809 small.fareast.isp.109 > 172.16.10.95.109: SF 39983116:39983116(0) win 1028 (ttl 28, id 39426)

Firewall Log (all packets scanned log)

Note: fields: for firewall All packets Scanned log

1 = Date of intercept
2 = Local time of Intercept
3 = Action taken (D = Denied, F = Packet Forwarding Attack)
4 = src interface
5 = dst interface
6 = src IP
7 = dst IP
8 = protocol
9 = src port
10 = dst port

1	2	3	4	5	6	7	8	9	10
2000/08/29	05:26:50:	D	El90x2	lo0	small.fareast.isp	172.16.10.2	tcp	21	21
2000/08/29	05:26:50:	F	El90x2	all	172.16.10.3	small.fareast.isp	tcp	21	21
2000/08/30	02:13:51:	D	El90x2	lo0	small.fareast.isp	172.16.10.2	tcp	109	109
2000/08/30	02:13:51:	F	El90x2	all	172.16.10.3	small.fareast.isp	tcp	109	109

Firewall Packet Forwarding Attack log

Alert: forward_attack
Time: 2000/08/29 05:26:50
Src Interface: El90x2
Dst Interface: all
Src Address: 172.16.10.3
Dst Address: small.fareast.isp
Protocol: tcp
Src Port: 21
Dst Port: 21

Alert: forward_attack
Time: 2000/08/30 02:13:51
Src Interface: El90x2
Dst Interface: all
Src Address: 172.16.10.3
Dst Address: small.fareast.isp
Protocol: tcp
Src Port: 109
Dst Port: 109

1. Source of Trace

The Snort and WinDump traces were captured on a dual-function IDS system (172.16.10.3) outside the external firewall of our business network. This computer ran basic IDS software – a Windows NT Port of Snort using the 07272k ruleset, as well as WinDump – a Windows NT port of TCPDUMP - for additional analysis. The firewall logs were taken from the external firewall (172.16.10.2). Except for the router, these are the only two hosts on this subnet. The router is under the control of an ISP and the logs are not available for inclusion.

Richard D. Salisko

2. Detect Generated By

The Detect was generated by the Snort IDS and the external firewall. Firewall logs excerpts include breakdown of fields where required.

3. Probability of Source Address Spoofing

The speed and duration of the attack indicate that this is more likely to be a reconnaissance effort than a denial of service (DoS). In order for the information obtained by this scan to be of any use, the attacker must be able to see any response from the victim's system. Although the attacker could be using a compromised computer and spoofing his own address to get the responses, it is unlikely that the source address is spoofed.

4. Description of Attack

This is a SYN-FIN attack. A normal initial TCP sequence would consist of a SYN, a SYN-ACK, then an ACK. A normal session close would include a FIN and a corresponding ACK from each system. This system is attempting to open and close a session simultaneously – this makes this trace immediately suspect. The attacker sends a forged packet with both the SYN and FIN flags set. This combination does not exist in the TCP/IP standard, so the response from the victim computer is unpredictable, which could make it useful for OS Fingerprinting because different IP stacks will likely respond in different ways.

The attacker appears to be attempting to map the available hosts by directing a single packet at each host on the subnet and then to the broadcast address, although it is unlikely that the attacker knows the subnet mask. It is possible that the packets are addressed to the much larger Class B broadcast address and the external router, whose logs are not available, forwarded the packets for valid hosts to this subnet. ARP packets were not collected so this possibility cannot be confirmed nor denied, although the short break between scanning 172.16.10.3 and the broadcast address would indicate that the intermediate addresses were also scanned.

In this case the packet is obviously crafted since there are a number of oddities. Other than both the SYN and FIN flags being set, the ID number is identical in all six packets, regardless of the destination host or port. As well, the identical sequence number was used for the first three packets directed at port 21, then changed for port 109.

It should be noted that this detect consists of two scans, approximately a day apart. Therefore, the attacker is taking his time possibly to avoid attracting attention. No other traffic was noted from this address during the month or during the following few days.

CVE Number - There are numerous FTP vulnerabilities listed for FTP and at least one POP2 vulnerability - CVE 1999-0920

Richard D. Salisko

5. Attack Mechanism

This attack is likely a reconnaissance scan specifically targeting ports 21 (ftp) and 109 (pop2). The attacker is attempting to map the network by scanning host ports and, by observing any response from the target systems, may be able to fingerprint the operating systems. The attacker is also attempting to circumvent any packet filtering devices by using source ports often allowed through such devices. Also, by setting both the SYN and FIN Flags, the attacker may be trying to avoid detection since they are both opening and closing the connection with the same packet. This appears to be an automated or scripted process.

6. Correlation

Almost identical scans were reported in Detects Analyzed Feb 29, Mar 5, 20, 24, 25, and 29 2000, although the destination ports were changed in some cases (53, 109 primarily, 21 in my case) In all cases the attackers used the same ID number – 39426. And the same Window size – 1028. In all cases except one, the same sequence number was used for each scan, unless the port changed, in which case the sequence number changed as well. This probably means that it was not a continuation of the first scan, but a new one.

7. Evidence of Active Targeting

There is no evidence of active targeting. This appears to be a general reconnaissance scan of the entire network. The attack is probably automated or scripted and directed at a large number of networks.

8. Severity

Criticality	= 5 (firewall, IDS)
Lethality	= 2 (general reconnaissance scan only)
System Countermeasures	= 5 (both computers had hardened operating systems with all patches applied)
Network Countermeasures	= 4 (all appropriate rulesets in place, firewall properly dropped all packets silently rather than responding with reset. IDS was not protected by firewall but all available countermeasures in place)

Equation

Severity = (5 + 2) – (5 + 4) = -2

Richard D. Salisko

9. Defensive Recommendation

The current security posture of the firewall is correct. The ruleset properly detected and logged the attack although, if possible, the logs could contain more detail on the actual packets observed. More detailed logging is not available on this brand of firewall however.

The IDS computer responded to the SYN-FIN packet with a reset which could tell the attacker the system exists and possibly help to fingerprint it's operating system. However, because the default gateway of the IDS was set to the external interface of the firewall, the IDS computer attempted to forward the reset packet to the attacker via the firewall, as evidenced by the Packet Forwarding Attack log from the firewall. The firewall did not forward the packet. An improvement to this scenario could be made by not giving the network card in the IDS an IP address, thereby making it inaccessible.

The disadvantage to the above scenario is that there is a possibility of the response packets being sent back to the attacker. The advantage here is that the firewall logs all responses sent from the IDS system.

10. Multiple Choice Question

According to TCP/IP Standards, how should a properly implemented IP Stack respond to a tcp packet with both SYN and FIN Flags set?

a. With a reset indicating that the system is not listening on port 109.
b. With an ACK packet to acknowledge the sequence sent.
c. This sequence is not legal under TCP/IP standards.
d. With a FIN-ACK to properly close the IP session.

Answer: C: This is not a valid sequence in TCP/IP Standards. A packet with both the SYN and FIN flags set will not be observed in normal TCP sessions. The packet must be crafted.

Richard D. Salisko

Detect Two:

Firewall All Sessions Log

Note: fields: for firewall log

1 = Date of intercept
2 = Local time of Intercept
3 = Action taken (T = Terminated Session)
4 = src interface
5 = dst interface
6 = src IP
7 = dst IP
8 = protocol
9 = src port
10 = dst port
11 = number of source packets
12 = number of destination packets
13 = number of source bytes
14 = number of destination bytes

1	2	3	4	5	6	7	8	9	10	11	12	13	14
2000/04/10	04:22:26:	T	El90x2	lo0	142.35.245.45	172.16.10.1	tcp	8294	3517	1	0	40	0
2000/04/10	04:22:26:	T	El90x2	lo0	31.199.164.98	172.16.10.1	tcp	21364	16	1	0	40	0
2000/04/10	04:22:26:	T	El90x2	lo0	248.154.49.117	172.16.10.1	tcp	54154	2768	1	0	40	0
2000/04/10	04:22:26:	T	El90x2	lo0	50.49.10.122	172.16.10.1	tcp	34721	1480	1	0	40	0
2000/04/10	04:22:26:	T	El90x2	lo0	238.193.177.85	172.16.10.1	tcp	63104	2541	1	0	40	0
2000/04/10	04:22:26:	T	El90x2	lo0	157.205.211.107	172.16.10.1	tcp	13083	2803	1	0	40	0
2000/04/10	04:22:26:	T	El90x2	lo0	238.61.236.16	172.16.10.1	tcp	57599	3487	1	0	40	0
2000/04/10	04:22:26:	T	El90x2	lo0	26.135.177.91	172.16.10.1	tcp	1992	3327	1	0	40	0
2000/04/10	04:22:26:	T	El90x2	lo0	43.150.75.30	172.16.10.1	tcp	21801	446	2	0	80	0
2000/04/10	04:22:26:	T	El90x2	lo0	111.182.24.30	172.16.10.1	tcp	32740	579	2	0	80	0
2000/04/10	04:22:26:	T	El90x2	lo0	68.64.199.58	172.16.10.1	tcp	5348	244	1	0	40	0
2000/04/10	04:22:27:	T	El90x2	lo0	52.64.61.87	172.16.10.1	tcp	23226	12379	1	0	40	0
2000/04/10	04:22:27:	T	El90x2	lo0	204.23.37.7	172.16.10.1	tcp	50156	12005	1	0	40	0
2000/04/10	04:22:27:	T	El90x2	lo0	73.46.234.80	172.16.10.1	tcp	37708	11988	1	0	40	0
2000/04/10	04:22:27:	T	El90x2	lo0	235.68.220.5	172.16.10.1	tcp	44700	12497	1	0	40	0

2000/04/10 04:22:27: T EI90x2	lo0	242.177.126.11	172.16.10.1	tcp	37010	12175	1	0	40	0
2000/04/10 04:22:27: T EI90x2	lo0	137.254.80.36	172.16.10.1	tcp	38641	12264	1	0	40	0
2000/04/10 04:22:27: T EI90x2	lo0	126.200.104.45	172.16.10.1	tcp	57280	12384	1	0	40	0
2000/04/10 04:22:28: T EI90x2	lo0	151.85.154.66	172.16.10.1	tcp	21575	9901	1	0	40	0
2000/04/10 04:22:28: T EI90x2	lo0	12.157.156.39	172.16.10.1	tcp	2886	9839	1	0	40	0
2000/04/10 04:22:28: T EI90x2	lo0	24.98.125.114	172.16.10.1	tcp	54855	11383	1	0	40	0
2000/04/10 04:22:28: T EI90x2	lo0	202.109.71.86	172.16.10.1	tcp	6795	10794	1	0	40	0
2000/04/10 04:22:28: T EI90x2	lo0	207.77.160.119	172.16.10.1	tcp	12465	10915	1	0	40	0
2000/04/10 04:22:28: T EI90x2	lo0	73.252.74.65	172.16.10.1	tcp	1466	10396	1	0	40	0
2000/04/10 04:22:28: T EI90x2	lo0	194.108.223.62	172.16.10.1	tcp	63296	11308	2	0	80	0
2000/04/10 04:22:28: T EI90x2	lo0	17.25.65.31	172.16.10.1	tcp	51922	10633	2	0	80	0
2000/04/10 04:22:28: T EI90x2	lo0	51.128.215.54	172.16.10.1	tcp	40548	11365	1	0	40	0
2000/04/10 04:22:28: T EI90x2	lo0	78.125.65.13	172.16.10.1	tcp	21087	11136	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	147.106.54.122	172.16.10.1	tcp	34720	26252	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	225.56.13.10	172.16.10.1	tcp	29969	27891	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	41.157.139.27	172.16.10.1	tcp	30286	25709	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	237.164.184.30	172.16.10.1	tcp	20928	26047	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	27.199.73.94	172.16.10.1	tcp	54351	28226	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	136.179.64.7	172.16.10.1	tcp	62827	28085	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	37.98.83.19	172.16.10.1	tcp	4801	25207	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	85.92.235.96	172.16.10.1	tcp	60172	27135	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	203.129.38.71	172.16.10.1	tcp	30088	27029	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	104.89.73.103	172.16.10.1	tcp	46138	28222	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	74.75.196.51	172.16.10.1	tcp	33188	27966	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	16.137.172.10	172.16.10.1	tcp	57665	28237	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	130.136.98.100	172.16.10.1	tcp	44894	27717	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	188.14.101.120	172.16.10.1	tcp	47469	27911	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	72.110.12.39	172.16.10.1	tcp	42711	27687	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	17.167.195.121	172.16.10.1	tcp	27411	27663	1	0	40	0
2000/04/10 04:22:30: T EI90x2	lo0	81.158.166.43	172.16.10.1	tcp	17497	26663	1	0	40	0
2000/04/10 04:22:32: T EI90x2	lo0	46.245.221.38	172.16.10.1	tcp	14013	43267	1	0	40	0
2000/04/10 04:22:32: T EI90x2	lo0	219.84.205.3	172.16.10.1	tcp	9378	43426	1	0	40	0
2000/04/10 04:22:32: T EI90x2	lo0	61.71.188.96	172.16.10.1	tcp	36925	45373	1	0	40	0
2000/04/10 04:22:32: T EI90x2	lo0	138.51.244.111	172.16.10.1	tcp	19576	45227	1	0	40	0
2000/04/10 04:22:32: T EI90x2	lo0	202.93.2.13	172.16.10.1	tcp	20078	45120	1	0	40	0
2000/04/10 04:22:32: T EI90x2	lo0	229.147.80.44	172.16.10.1	tcp	40513	46300	1	0	40	0
2000/04/10 04:22:32: T EI90x2	lo0	83.24.204.21	172.16.10.1	tcp	10362	45048	1	0	40	0
2000/04/10 04:22:32: T EI90x2	lo0	180.25.178.31	172.16.10.1	tcp	48178	44819	1	0	40	0
2000/04/10 04:22:32: T EI90x2	lo0	199.99.179.103	172.16.10.1	tcp	3851	44725	1	0	40	0
2000/04/10 04:22:32: T EI90x2	lo0	137.96.5.61	172.16.10.1	tcp	38300	43756	2	0	80	0

2000/04/10	04:22:32:	T	EI90x2	lo0	34.181.102.111	172.16.10.1	tcp	26002	44061	1	0	40	0
2000/04/10	04:22:32:	T	EI90x2	lo0	28.110.234.59	172.16.10.1	tcp	46429	44678	1	0	40	0
2000/04/10	04:22:34:	T	EI90x2	lo0	175.169.49.92	172.16.10.1	tcp	49646	60314	1	0	40	0
2000/04/10	04:22:34:	T	EI90x2	lo0	119.66.73.60	172.16.10.1	tcp	26055	63567	1	0	40	0
2000/04/10	04:22:34:	T	EI90x2	lo0	201.13.204.79	172.16.10.1	tcp	4539	62599	1	0	40	0
2000/04/10	04:22:34:	T	EI90x2	lo0	47.103.248.113	172.16.10.1	tcp	22749	63466	1	0	40	0
2000/04/10	04:22:34:	T	EI90x2	lo0	114.178.5.7	172.16.10.1	tcp	49511	61445	1	0	40	0
2000/04/10	04:22:34:	T	EI90x2	lo0	202.167.178.117	172.16.10.1	tcp	46287	61420	1	0	40	0
2000/04/10	04:22:34:	T	EI90x2	lo0	221.160.177.34	172.16.10.1	tcp	12228	60130	1	0	40	0
2000/04/10	04:22:34:	T	EI90x2	lo0	232.125.224.69	172.16.10.1	tcp	42733	61681	2	0	80	0
2000/04/10	04:22:34:	T	EI90x2	lo0	13.95.247.38	172.16.10.1	tcp	28331	63590	1	0	40	0
2000/04/10	04:22:36:	T	EI90x2	lo0	214.80.116.59	172.16.10.1	tcp	50395	11760	1	0	40	0
2000/04/10	04:22:36:	T	EI90x2	lo0	136.64.246.102	172.16.10.1	tcp	64899	14000	1	0	40	0
2000/04/10	04:22:36:	T	EI90x2	lo0	245.88.95.125	172.16.10.1	tcp	8252	12463	1	0	40	0
2000/04/10	04:22:36:	T	EI90x2	lo0	30.166.147.89	172.16.10.1	tcp	54591	13895	1	0	40	0
2000/04/10	04:22:36:	T	EI90x2	lo0	134.195.135.110	172.16.10.1	tcp	3200	12939	2	0	80	0
2000/04/10	04:22:37:	T	EI90x2	lo0	167.102.133.83	172.16.10.1	tcp	4132	14799	1	0	40	0
2000/04/10	04:22:37:	T	EI90x2	lo0	27.73.226.112	172.16.10.1	tcp	39921	14488	1	0	40	0
2000/04/10	04:22:37:	T	EI90x2	lo0	153.10.56.11	172.16.10.1	tcp	13925	13380	2	0	80	0
2000/04/10	04:22:37:	T	EI90x2	lo0	150.210.185.38	172.16.10.1	tcp	39250	14541	1	0	40	0
2000/04/10	04:22:37:	T	EI90x2	lo0	207.80.222.102	172.16.10.1	tcp	51541	13428	1	0	40	0
2000/04/10	04:22:39:	T	EI90x2	lo0	249.166.13.113	172.16.10.1	tcp	65117	28757	1	0	40	0
2000/04/10	04:22:39:	T	EI90x2	lo0	107.66.153.69	172.16.10.1	tcp	39504	31536	1	0	40	0
2000/04/10	04:22:39:	T	EI90x2	lo0	249.129.64.83	172.16.10.1	tcp	8286	31121	1	0	40	0
2000/04/10	04:22:39:	T	EI90x2	lo0	219.227.64.87	172.16.10.1	tcp	12880	30562	1	0	40	0
2000/04/10	04:22:39:	T	EI90x2	lo0	197.50.167.4	172.16.10.1	tcp	48770	29358	1	0	40	0
2000/04/10	04:22:39:	T	EI90x2	lo0	57.206.150.114	172.16.10.1	tcp	7819	31040	1	0	40	0
2000/04/10	04:22:39:	T	EI90x2	lo0	89.165.45.85	172.16.10.1	tcp	65110	30560	1	0	40	0
2000/04/10	04:22:39:	T	EI90x2	lo0	130.28.58.70	172.16.10.1	tcp	25916	28994	1	0	40	0
2000/04/10	04:22:41:	T	EI90x2	lo0	80.91.95.54	172.16.10.1	tcp	39303	47063	1	0	40	0
2000/04/10	04:22:41:	T	EI90x2	lo0	221.119.218.8	172.16.10.1	tcp	10859	47037	1	0	40	0
2000/04/10	04:22:41:	T	EI90x2	lo0	42.41.132.55	172.16.10.1	tcp	61581	47028	1	0	40	0
2000/04/10	04:22:41:	T	EI90x2	lo0	239.113.97.7	172.16.10.1	tcp	40565	46357	1	0	40	0
2000/04/10	04:22:41:	T	EI90x2	lo0	157.61.129.120	172.16.10.1	tcp	21941	47779	1	0	40	0
2000/04/10	04:22:41:	T	EI90x2	lo0	26.57.4.24	172.16.10.1	tcp	24721	45828	1	0	40	0
2000/04/10	04:22:41:	T	EI90x2	lo0	44.61.164.3	172.16.10.1	tcp	56484	47698	1	0	40	0
2000/04/10	04:22:41:	T	EI90x2	lo0	76.7.118.93	172.16.10.1	tcp	27748	48051	1	0	40	0
2000/04/10	04:22:41:	T	EI90x2	lo0	58.36.152.40	172.16.10.1	tcp	27832	47407	1	0	40	0
2000/04/10	04:22:41:	T	EI90x2	lo0	79.100.181.48	172.16.10.1	tcp	10920	46807	1	0	40	0
2000/04/10	04:22:41:	T	EI90x2	lo0	56.193.114.44	172.16.10.1	tcp	55641	47746	1	0	40	0

Richard D. Salisko

2000/04/10 04:22:41: T EI90x2	lo0	101.187.247.84	172.16.10.1	tcp	18557	48202	1	0	40	0
2000/04/10 04:22:43: T EI90x2	lo0	189.246.34.44	172.16.10.1	tcp	23459	2090	1	0	40	0
2000/04/10 04:22:43: T EI90x2	lo0	92.64.162.20	172.16.10.1	tcp	3819	577	1	0	40	0
2000/04/10 04:22:43: T EI90x2	lo0	43.5.137.87	172.16.10.1	tcp	55199	64033	1	0	40	0
2000/04/10 04:22:43: T EI90x2	lo0	253.71.169.46	172.16.10.1	tcp	34492	650	1	0	40	0
2000/04/10 04:22:43: T EI90x2	lo0	15.166.14.121	172.16.10.1	tcp	7110	64664	1	0	40	0
2000/04/10 04:22:43: T EI90x2	lo0	210.163.102.25	172.16.10.1	tcp	51775	64315	1	0	40	0
2000/04/10 04:22:43: T EI90x2	lo0	85.248.198.26	172.16.10.1	tcp	23715	146	1	0	40	0
2000/04/10 04:22:43: T EI90x2	lo0	112.86.120.69	172.16.10.1	tcp	11576	487	1	0	40	0
2000/04/10 04:22:43: T EI90x2	lo0	70.93.7.105	172.16.10.1	tcp	39049	564	1	0	40	0
2000/04/10 04:22:43: T EI90x2	lo0	126.202.203.53	172.16.10.1	tcp	35043	96	1	0	40	0
2000/04/10 04:22:43: T EI90x2	lo0	71.135.193.0	172.16.10.1	tcp	51773	64005	1	0	40	0
2000/04/10 04:22:43: T EI90x2	lo0	214.116.83.114	172.16.10.1	tcp	47344	2078	1	0	40	0
2000/04/10 04:22:43: T EI90x2	lo0	231.221.187.115	172.16.10.1	tcp	30842	90	1	0	40	0
2000/04/10 04:22:43: T EI90x2	lo0	168.25.252.8	172.16.10.1	tcp	61049	64369	1	0	40	0
2000/04/10 04:22:43: T EI90x2	lo0	78.167.113.8	172.16.10.1	tcp	52381	2286	1	0	40	0
2000/04/10 04:22:43: T EI90x2	lo0	77.112.58.99	172.16.10.1	tcp	24320	2177	1	0	40	0
2000/04/10 04:22:45: T EI90x2	lo0	203.233.10.118	172.16.10.1	tcp	47753	21679	1	0	40	0
2000/04/10 04:22:45: T EI90x2	lo0	150.109.89.6	172.16.10.1	tcp	13703	22237	1	0	40	0
2000/04/10 04:22:45: T EI90x2	lo0	226.31.232.110	172.16.10.1	tcp	50750	21570	1	0	40	0
2000/04/10 04:22:48: T EI90x2	lo0	100.113.193.79	172.16.10.1	tcp	2211	38912	1	0	40	0
2000/04/10 04:22:48: T EI90x2	lo0	228.84.177.60	172.16.10.1	tcp	6124	38440	1	0	40	0
2000/04/10 04:22:48: T EI90x2	lo0	74.227.77.49	172.16.10.1	tcp	59543	38419	1	0	40	0
2000/04/10 04:22:48: T EI90x2	lo0	227.176.132.21	172.16.10.1	tcp	12117	34085	1	0	40	0
2000/04/10 04:22:48: T EI90x2	lo0	176.253.23.9	172.16.10.1	tcp	5488	32915	1	0	40	0
2000/04/10 04:22:48: T EI90x2	lo0	232.49.94.45	172.16.10.1	tcp	31899	32810	1	0	40	0
2000/04/10 04:22:48: T EI90x2	lo0	38.38.181.111	172.16.10.1	tcp	39913	32827	1	0	40	0
2000/04/10 04:22:48: T EI90x2	lo0	212.67.230.58	172.16.10.1	tcp	14368	37888	1	0	40	0
2000/04/10 04:22:48: T EI90x2	lo0	61.30.121.4	172.16.10.1	tcp	8451	34240	1	0	40	0
2000/04/10 04:22:48: T EI90x2	lo0	45.252.35.62	172.16.10.1	tcp	39371	37881	1	0	40	0
2000/04/10 04:22:48: T EI90x2	lo0	27.221.65.91	172.16.10.1	tcp	5183	38378	1	0	40	0
2000/04/10 04:22:50: T EI90x2	lo0	110.59.41.76	172.16.10.1	tcp	12380	55656	1	0	40	0
2000/04/10 04:22:50: T EI90x2	lo0	81.11.124.100	172.16.10.1	tcp	35768	56477	1	0	40	0
2000/04/10 04:22:50: T EI90x2	lo0	86.176.125.82	172.16.10.1	tcp	59949	56899	1	0	40	0
2000/04/10 04:22:50: T EI90x2	lo0	131.127.203.71	172.16.10.1	tcp	4649	56978	1	0	40	0
2000/04/10 04:22:50: T EI90x2	lo0	232.193.92.83	172.16.10.1	tcp	56792	56749	1	0	40	0
2000/04/10 04:22:50: T EI90x2	lo0	86.125.41.90	172.16.10.1	tcp	7280	55902	1	0	40	0
2000/04/10 04:22:50: T EI90x2	lo0	234.84.221.76	172.16.10.1	tcp	61392	56542	1	0	40	0
2000/04/10 04:22:50: T EI90x2	lo0	152.1.96.40	172.16.10.1	tcp	11203	55959	1	0	40	0
2000/04/10 04:22:50: T EI90x2	lo0	195.57.27.13	172.16.10.1	tcp	23827	56011	1	0	40	0

2000/04/10 04:22:50: T El90x2	lo0	171.143.254.48	172.16.10.1	tcp	11561	54899	1	0	40	0
2000/04/10 04:22:50: T El90x2	lo0	239.227.135.87	172.16.10.1	tcp	18328	56838	1	0	40	0
2000/04/10 04:22:50: T El90x2	lo0	84.176.132.9	172.16.10.1	tcp	16785	56971	1	0	40	0
2000/04/10 04:22:52: T El90x2	lo0	238.47.203.27	172.16.10.1	tcp	7228	8868	1	0	40	0
2000/04/10 04:22:52: T El90x2	lo0	214.145.100.123	172.16.10.1	tcp	34320	7142	1	0	40	0
2000/04/10 04:22:52: T El90x2	lo0	171.137.154.32	172.16.10.1	tcp	63287	7529	1	0	40	0
2000/04/10 04:22:52: T El90x2	lo0	51.141.210.19	172.16.10.1	tcp	13193	2356	1	0	40	0
2000/04/10 04:22:52: T El90x2	lo0	219.180.235.122	172.16.10.1	tcp	22610	8581	1	0	40	0
2000/04/10 04:22:52: T El90x2	lo0	8.31.150.86	172.16.10.1	tcp	62437	6691	1	0	40	0
2000/04/10 04:22:52: T El90x2	lo0	210.80.232.107	172.16.10.1	tcp	28590	8956	1	0	40	0
2000/04/10 04:22:54: T El90x2	lo0	117.168.81.27	172.16.10.1	tcp	23834	20802	1	0	40	0
2000/04/10 04:22:54: T El90x2	lo0	25.82.96.55	172.16.10.1	tcp	17990	20092	1	0	40	0
2000/04/10 04:22:54: T El90x2	lo0	43.126.201.123	172.16.10.1	tcp	59973	18955	1	0	40	0
2000/04/10 04:22:54: T El90x2	lo0	84.113.129.108	172.16.10.1	tcp	61067	24093	1	0	40	0
2000/04/10 04:22:54: T El90x2	lo0	41.112.108.118	172.16.10.1	tcp	21972	20490	1	0	40	0
2000/04/10 04:22:54: T El90x2	lo0	55.235.103.58	172.16.10.1	tcp	47680	21141	1	0	40	0
2000/04/10 04:22:54: T El90x2	lo0	175.156.90.70	172.16.10.1	tcp	33764	20837	1	0	40	0
2000/04/10 04:22:54: T El90x2	lo0	64.168.48.38	172.16.10.1	tcp	52576	19646	1	0	40	0
2000/04/10 04:22:54: T El90x2	lo0	177.64.22.29	172.16.10.1	tcp	39218	23691	1	0	40	0
2000/04/10 04:22:54: T El90x2	lo0	177.158.28.0	172.16.10.1	tcp	29680	20516	1	0	40	0
2000/04/10 04:22:54: T El90x2	lo0	88.88.165.27	172.16.10.1	tcp	9168	21020	1	0	40	0
2000/04/10 04:22:54: T El90x2	lo0	27.221.21.64	172.16.10.1	tcp	29441	20733	1	0	40	0
2000/04/10 04:22:54: T El90x2	lo0	50.59.22.27	172.16.10.1	tcp	9828	20540	1	0	40	0
2000/04/10 04:22:54: T El90x2	lo0	91.163.253.82	172.16.10.1	tcp	41986	19579	1	0	40	0
2000/04/10 04:22:55: T El90x2	lo0	144.253.194.65	172.16.10.1	tcp	44477	24725	1	0	40	0
2000/04/10 04:22:57: T El90x2	lo0	32.149.136.34	172.16.10.1	tcp	61699	39991	1	0	40	0
2000/04/10 04:22:57: T El90x2	lo0	177.96.149.18	172.16.10.1	tcp	59019	39255	1	0	40	0
2000/04/10 04:22:57: T El90x2	lo0	212.70.141.32	172.16.10.1	tcp	44742	39591	1	0	40	0
2000/04/10 04:22:57: T El90x2	lo0	171.1.255.96	172.16.10.1	tcp	11670	41604	1	0	40	0
2000/04/10 04:22:57: T El90x2	lo0	208.101.114.45	172.16.10.1	tcp	13022	40935	1	0	40	0
2000/04/10 04:22:57: T El90x2	lo0	9.30.217.44	172.16.10.1	tcp	13369	39655	1	0	40	0
2000/04/10 04:22:57: T El90x2	lo0	34.9.162.69	172.16.10.1	tcp	34414	40099	1	0	40	0
2000/04/10 04:22:57: T El90x2	lo0	208.31.203.51	172.16.10.1	tcp	58091	41587	1	0	40	0
2000/04/10 04:22:57: T El90x2	lo0	198.254.167.112	172.16.10.1	tcp	44338	39944	1	0	40	0
2000/04/10 04:22:59: T El90x2	lo0	162.239.166.48	172.16.10.1	tcp	24156	59710	1	0	40	0
2000/04/10 04:22:59: T El90x2	lo0	86.169.250.55	172.16.10.1	tcp	26037	58817	1	0	40	0
2000/04/10 04:22:59: T El90x2	lo0	159.38.60.113	172.16.10.1	tcp	4644	57419	1	0	40	0
2000/04/10 04:22:59: T El90x2	lo0	210.254.98.47	172.16.10.1	tcp	21642	55219	1	0	40	0
2000/04/10 04:22:59: T El90x2	lo0	78.47.6.54	172.16.10.1	tcp	65346	55248	1	0	40	0
2000/04/10 04:22:59: T El90x2	lo0	134.8.147.116	172.16.10.1	tcp	43750	59935	1	0	40	0

2000/04/10 04:22:59: T	El90x2	lo0	49.200.157.89	172.16.10.1	tcp	47622	56386	1	0	40	0
2000/04/10 04:22:59: T	El90x2	lo0	62.1.236.55	172.16.10.1	tcp	59789	56556	1	0	40	0
2000/04/10 04:22:59: T	El90x2	lo0	30.4.129.9	172.16.10.1	tcp	29723	57973	1	0	40	0
2000/04/10 04:22:59: T	El90x2	lo0	136.68.34.120	172.16.10.1	tcp	5196	55627	1	0	40	0
2000/04/10 04:22:59: T	El90x2	lo0	136.66.144.108	172.16.10.1	tcp	25810	59368	1	0	40	0
2000/04/10 04:22:59: T	El90x2	lo0	88.54.0.17	172.16.10.1	tcp	54060	55773	1	0	40	0
2000/04/10 04:22:59: T	El90x2	lo0	111.154.44.73	172.16.10.1	tcp	16862	59031	1	0	40	0
2000/04/10 04:22:59: T	El90x2	lo0	179.244.132.21	172.16.10.1	tcp	63581	54996	1	0	40	0
2000/04/10 04:22:59: T	El90x2	lo0	246.165.246.2	172.16.10.1	tcp	47378	59069	1	0	40	0
2000/04/10 04:22:59: T	El90x2	lo0	178.223.212.39	172.16.10.1	tcp	36876	58103	1	0	40	0
2000/04/10 04:23:01: T	El90x2	lo0	36.150.2.41	172.16.10.1	tcp	20324	6545	1	0	40	0
2000/04/10 04:23:01: T	El90x2	lo0	237.241.239.109	172.16.10.1	tcp	23796	8394	1	0	40	0
2000/04/10 04:23:01: T	El90x2	lo0	216.135.213.102	172.16.10.1	tcp	61341	10721	1	0	40	0
2000/04/10 04:23:01: T	El90x2	lo0	36.165.163.21	172.16.10.1	tcp	16168	8223	1	0	40	0
2000/04/10 04:23:01: T	El90x2	lo0	222.51.86.59	172.16.10.1	tcp	51057	6197	1	0	40	0

1. Source of Trace

The trace was taken from the ‘ Completed Sessions ‘ Log of the external firewall on our business network. At the time the external firewall was the only host on this segment except the Internet Service Provider’s router. The router logs are not available for inclusion.

2. Detect Generated By

The Detect was generated by the external firewall. The activity was first noted on the Firewall ‘ Denied Packets ‘ Logfile. Logs excerpts include breakdown of fields where required.

3. Probability of Source Address Spoofing

It is extremely likely that almost all addresses are spoofed. There are a total of 190 different but outwardly legitimate source addresses, although many of the addresses used are marked as reserved by the IANA. Also, no two source or destination ports are identical. This probably means that it is a single system selecting IP addresses, source and destination ports in a near random order.

Unfortunately, no TTLs are available to help gauge distance. It is very possible that the attackers real IP Address is one of the source addresses included.

Richard D. Salisko

4. Description of Attack

Without more information this appears to be denial of service attack – CVE 1999-0116. But it is possible that it is an attempt to Fingerprint the operating system of my firewall while sending lots of cover fire. All packets are 40 bytes in length. In all cases except 8, there was a just a single packet from each source IP address. In eight cases, the firewall logged two packets, for a total of 80 bytes. None of the 8 addresses were significant - some were listed as reserved by the IANA. Unfortunately, without additional information to analyze, it is impossible to determine what flags were set in the TCP packets. Although I could speculate that these were SYN or SYN-ACK packets typically used in a SYN Flood DoS, no confirmation is possible.

The timeline is broken and inconsistent, which implies that this site may not have been the only target of this attack. The intensity of the packets ranged from a low of 1 to a high of 17 packets per second and averaged 11-12 packets per second. This is consistent with the aim of a DoS Attack.

The attack lasted a total of 40 seconds. Source and destination ports used spanned almost the whole range of possibilities. There is insufficient information to determine if the attacker was targeting a single or range of ports.

5. Attack Mechanism

The aim of a Denial of Service (DoS) Attack of the SYN Flood variety is to keep the target system too busy or consume enough of its resources to prevent it from responding to legitimate service requests.

A typical attack works by sending a SYN packet to the target system. The target would respond with a SYN-ACK, reserve the appropriate buffer space and then wait for the next packet in the sequence (an ACK). In this case the next packet would never arrive. The attacking system would continue to send new SYN packets with different IDs and Sequence numbers to open more and more new sessions. The victim would eventually commit all of its available resources to these phantom sessions, leaving none for legitimate sessions. As little as five or six packets per second could overwhelm some older systems.

6. Correlation

There are numerous examples of this type of attack available although I have been unable to locate specific examples using this wide a range of apparently random source and destination ports.

7. Evidence of Active Targeting

It appears that this system was specifically targeted for this attack. DoS attacks are rarely random. Judging by the timeline, other systems were being simultaneously attacked. This is very obviously an automated or scripted attack.

8. Severity

Criticality	= 5 (firewall, IDS)
Lethality	= 5 (attempted Denial of Service)
System Countermeasures	= 5 (Very modern firewall - hardened operating systems with all patches applied)
Network Countermeasures	= 4 (all appropriate rulesets in place, firewall dropped all packets silently rather than responding with reset. Logging was adequate but missed the suspected DoS SYN Flood)

Equation

Severity = (5 + 5) – (5 + 4) = 1

9. Defensive Recommendation

The current security posture of the firewall is correct. No apparent loss of service was noted. The ruleset properly denied and adequately logged the attack although, if possible, the logs could contain more detail on the actual packets observed. According to manuals, the firewall has the ability to detect and report SYN Floods. It did not report this attack as a SYN Flood however. It is possible that as a modern firewall it has built-in defenses against such an attack and therefore the thresholds for detection are set quite high. To properly log this attack, I recommend that the thresholds be lowered to a level that allows firewall to detect this level of activity and alert the administrators. In addition, the settings on session time-outs should be verified and adjusted to lower the effects of this type of attack.

An IDS or some kind of packet logging utility is invaluable in analyzing this type of activity. At the time this attack occurred, no such facility was positioned on the external segment. That capability is now in position.

Richard D. Salisko

10. Multiple Choice Question

In the trace below, which is the TRUE source address of the attack?

Date	Time	A	Src if	Dst if	Src Address	DST Address	Protocol	Src port	Dst port				
2000/04/10	**04:22:41:**	**T**	**El90x2**	**lo0**	**58.36.152.40**	**172.16.10.1**	**tcp**	**27832**	**47407**	**1**	**0**	**40**	**0**
2000/04/10	**04:22:41:**	**T**	**El90x2**	**lo0**	**79.100.181.48**	**172.16.10.1**	**tcp**	**10920**	**46807**	**1**	**0**	**40**	**0**
2000/04/10	**04:22:41:**	**T**	**El90x2**	**lo0**	**56.193.114.44**	**172.16.10.1**	**tcp**	**514**	**47746**	**1**	**0**	**40**	**0**
2000/04/10	**04:22:41:**	**T**	**El90x2**	**lo0**	**101.187.247.84**	**172.16.10.1**	**tcp**	**18557**	**48202**	**1**	**0**	**40**	**0**
2000/04/10	**04:22:43:**	**T**	**El90x2**	**lo0**	**189.246.34.44**	**172.16.10.1**	**tcp**	**23459**	**2090**	**1**	**0**	**40**	**0**
2000/04/10	**04:22:43:**	**T**	**El90x2**	**lo0**	**92.64.162.20**	**172.16.10.1**	**tcp**	**3819**	**577**	**1**	**0**	**40**	**0**
2000/04/10	**04:22:43:**	**T**	**El90x2**	**lo0**	**43.5.137.87**	**172.16.10.1**	**tcp**	**55199**	**64033**	**1**	**0**	**40**	**0**

a. 56.193.114.44 because the source port is a well-known port < 1024.
b. 92.64.162.20 because the destination port is a well-known port < 1024.
c. Impossible to determine from the information shown.
d. 101.187.247.84 because it is the last packet sent at 04:22:41

Answer: C: There is nothing in this trace that can identify the true source. It is entirely possible that all addresses are spoofed.

Detect Three:

Firewall Log (all packets scanned log)

Note: fields: for firewall All packets Scanned log

1 = Date of intercept
2 = Local time of Intercept
3 = Action taken (D = Denied, F = Packet Forwarding Attack)
4 = src interface
5 = dst interface
6 = src IP
7 = dst IP
8 = protocol
9 = src port
10 = dst port

1	2	3	4	5	6	7	8	9	10
2000/09/03	04:42:57:	D	El90x2	lo0	61.141.205.154	172.16.10.2	tcp	1739	8080
2000/09/03	04:42:58:	F	El90x2	all	172.16.10.3	61.141.205.154	tcp	8080	2068
2000/09/03	04:43:10:	D	El90x2	lo0	61.141.205.154	172.16.10.2	tcp	2131	8080

Snort Output

```
[**] MISC-WinGate-8080-Attempt [**]
09/03-04:44:07.565782 0:E0:39:80:42:6E -> 0:10:4B:9C:3E:E9 type:0x800 len:0x4A
61.141.205.154:2068 -> 172.16.10.3:8080 TCP TTL:48 TOS:0x0 ID:17392  DF
**S***** Seq: 0x3A8F3931   ACK: 0x0   Win: 0x7C9C
TCP Options => MSS: 1450 SACKOK TS: 156089513 0 NOP WS: 0

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] MISC-WinGate-8080-Attempt [**]
09/03-04:44:06.566983 0:E0:39:80:42:6E -> 0:10:5A:19:3A:1C type:0x800 len:0x4A
61.141.205.154:1739 -> 172.16.10.2:8080 TCP TTL:48 TOS:0x0 ID:16522  DF
**S***** Seq: 0x3AB563D3   ACK: 0x0   Win: 0x7C9C
TCP Options => MSS: 1450 SACKOK TS: 156089413 0 NOP WS: 0

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
 [**] MISC-WinGate-8080-Attempt [**]
09/03-04:44:19.588097 0:E0:39:80:42:6E -> FF:FF:FF:FF:FF:FF type:0x800 len:0x4A
61.141.205.154:2131 -> 172.16.10.95:8080 TCP TTL:48 TOS:0x0 ID:27840  DF
**S***** Seq: 0x3B8B7C83   ACK: 0x0   Win: 0x7C9C
TCP Options => MSS: 1450 SACKOK TS: 156090713 0 NOP WS: 0

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

1. Source of Trace

The Snort trace was captured on a basic IDS system (172.16.10.3) outside the external firewall of our business network. This computer ran basic IDS software – a Windows NT Port of SNORT using the 07272k ruleset. The firewall logs were taken from the external firewall (172.16.10.2) Except for the router, these are the only two hosts on this subnet. The router is under the control of an ISP and the logs are not available for inclusion.

2. Detect Generated By

The Detect was generated by the SNORT IDS and the external firewall. Firewall logs excerpts include breakdown of fields where required.

3. Probability of Source Address Spoofing

In order for the information obtained by this scan to be of any use, the attacker must be able to see any response from the scanned systems. Although the attacker could be using a compromised computer and spoofing his own address to obtain the responses, it is unlikely that the source address is spoofed.

4. Description of Attack

The attacker appears to be checking all hosts on the subnet by directing a single SYN packet at each host and then to the broadcast address, although it is unlikely that the attacker knows the subnet mask. It is possible that the packets are addressed to a much larger broadcast address and the external router, whose logs are not available, forwarded the packets for valid hosts to this subnet. ARP packets were not collected so this possibility cannot be confirmed nor denied, although the break of a few seconds between the scan of 172.26.10.3 and the broadcast address of 172.16.10.95 seems to indicate that the intermediate addresses were also scanned.

The firewall log shows that 172.16.10.3 (the IDS) properly responded to the scan by sending a Reset, which tells the scanner the port is not open. The firewall logged this as a Packet Forwarding Attack because the IDS, to avoid sending just such responses back to the attacker, has it's default gateway address set to the firewall's external interface. When the IDS attempts to respond to any packet, it uses its default gateway, the firewall, to route such packets. The firewall rules forbid such behavior and log it as an attack. This little bit of extra log information helps to provide a better picture of the attack.

5. Attack Mechanism

Port 8080 scans are commonly associated with a trojan called Ringzero and proxies. If a trojan had infected an internal network system, this would be seen as outgoing traffic. However, the Ringzero trojan normally scans three ports in the following sequence – 80, 8080, and 3128. In this case, the subnet is being scanned only for port 8080 and from an external source, so it is reasonably safe to assume that this is a simple scan for open Wingate proxy servers using port 8080 – a fact confirmed by the Snort IDS Alert.

Wingate is a proxy server which allows multiple computers to use a single internet interface. According to CERT Vulnerability Note (VN) 98.03, the default configuration of the server will allows external connections without authentication (CVE 1999-0291). If an attacker can locate a server with this configuration, they could use that server to redirect their connections, thus hiding their true identity. This would provide an excellent stealth tool for attacking other systems.

6. Correlation

Wingate scans are common. Numerous similar examples can be seen in SANS Detects Analyzed reports by searching on the keywords Wingate and 8080.

7. Evidence of Active Targeting

There is no evidence of active targeting. This appears to be a general search for listening Wingate servers. The attack is probably automated or scripted and directed at a large number of networks.

8. Severity

Criticality	= 5 (firewall, IDS)
Lethality	= 1 (Very unlikely to succeed as no proxies are available for use. It it was successful however, it could prove costly in terms of reputation and legal responsibility if a local host was used to attack another organizations systems.)
System Countermeasures	= 5 (both computers hardened operating systems with all patches applied)
Network Countermeasures	= 4 (all appropriate rulesets in place, firewall properly dropped all packets silently rather than responding with reset. IDS was not protected by firewall but all available countermeasures in place)

Equation Severity = (5 + 1) – (5 + 4) = -3

9. Defensive Recommendation

The current security posture of the firewall is correct. The ruleset properly detected and logged the attack although, if possible, the logs could contain more detail on the actual packets observed. More detailed logging is not available on this brand of firewall however.

The IDS computer responded to the SYN-FIN packet with a reset which could tell the attacker the system exists. However, because the default gateway of the IDS was set to the external interface of the firewall, the IDS computer attempted to forward the reset packet to the attacker via the firewall, as evidenced by the Packet Forwarding Attack log from the firewall. The firewall did not forward the packet. An improvement to this scenario could be made by not giving the network card in the IDS an IP address, thereby making it inaccessible.

No packet logging facility, such as TCPDump was in place when these activities were logged. Such a facility would provide much more information to assist in analysis. WINDump, a Windows NT port of TCPDump, is now in place.

Richard D. Salisko

10. Multiple Choice Question

According to TCP/IP standards, how should a system running a properly implemented TCP/IP stack respond to a tcp SYN packet on any listening port ?

a. With a reset indicating that the system is available and listening on the port.
b. With an ACK packet to complete the two-way handshake.
c. With a SYN-ACK to Acknowledge the SYN and initiate it's part of the session.
d. With a FIN-ACK to properly close the IP session.

Answer: C: This is the second step in the three-way handshake. The Final part requires an ACK from the first system.

Detect Four:

Firewall Log (all packets scanned log)

Note: fields: for firewall All packets Scanned log

1 = Date of intercept
2 = Local time of Intercept
3 = Action taken (D = Denied, F = Packet Forwarding Attack)
4 = src interface
5 = dst interface
6 = src IP
7 = dst IP
8 = protocol
9 = src port
10 = dst port

1	2	3	4	5	6	7	8	9	10
2000/09/03	12:24:07:	F	El90x2	all	172.16.10.3	193.95.100.162	tcp	27374	1393
2000/09/03	12:24:07:	D	El90x2	lo0	193.95.100.162	172.16.10.2	tcp	1392	27374
2000/09/03	12:24:13:	F	El90x2	all	172.16.10.3	193.95.100.162	tcp	27374	1393
2000/09/03	12:24:25:	F	El90x2	all	172.16.10.3	193.95.100.162	tcp	27374	1393

Richard D. Salisko

WINDump Output (Windows NT Port of TCPDump)

```
12:25:16.086864 193.95.100.162.1393 > 172.16.10.3.27374: S 34321588:34321588(0) win 8192 <mss 1460> (DF) (ttl 107, id 33474)
                         4500 002c 82c2 4000 6b06 6185 c15f 64a2
                         d12f 3453 0571 6aee 020b b4b4 0000 0000
                         6002 2000 2583 0000 0204 05b4 b4b4
12:25:16.087378 172.16.10.3.27374 > 193.95.100.162.1393: R 0:0(0) ACK 34321589 win 0 (ttl 128, id 6656)
                         4500 0028 1a00 0000 8006 f54b d12f 3453
                         c15f 64a2 6aee 0571 0000 0000 020b b4b5
                         5014 0000 5d2c 0000
12:25:16.219248 193.95.100.162.1392 > 172.16.10.2.27374: S 34321588:34321588(0) win 8192 <mss 1460> (DF) (ttl 107, id 37570)
                         4500 002c 92c2 4000 6b06 5186 c15f 64a2
                         d12f 3452 0570 6aee 020b b4b4 0000 0000
                         6002 2000 2585 0000 0204 05b4 b4b4
12:25:22.206424 193.95.100.162.1393 > 172.16.10.3.27374: S 34321588:34321588(0) win 8192 <mss 1460> (DF) (ttl 107, id 16324)
                         4500 002c 3fc4 4000 6b06 a483 c15f 64a2
                         a14e 3453 0571 6aee 020b b4b4 0000 0000
                         6002 2000 2583 0000 0204 05b4 b4b4
12:25:22.206612 172.16.10.3.27374 > 193.95.100.162.1393: R 0:0(0) ACK 1 win 0 (ttl 128, id 6912)
                         4500 0028 1b00 0000 8006 f44b a14e 3453
                         c15f 64a2 6aee 0571 0000 0000 020b b4b5
                         5014 0000 5d2c 0000
12:25:34.157286 193.95.100.162.1393 > 172.16.10.3.27374: S 34321588:34321588(0) win 8192 <mss 1460> (DF) (ttl 107, id 57030)
                         4500 002c dec6 4000 6b06 0581 c15f 64a2
                         a14e 3453 0571 6aee 020b b4b4 0000 0000
                         6002 2000 2583 0000 0204 05b4 b4b4
12:25:34.157485 172.16.10.3.27374 > 193.95.100.162.1393: R 0:0(0) ACK 1 win 0 (ttl 128, id 7168)
                         4500 0028 1c00 0000 8006 f34b a14e 3453
                         c15f 64a2 6aee 0571 0000 0000 020b b4b5
                         5014 0000 5d2c 0000
12:25:34.269517 193.95.100.162.1392 > 172.16.10.2.27374: S 34321588:34321588(0) win 8192 <mss 1460> (DF) (ttl 107, id 61126)
                         4500 002c eec6 4000 6b06 f581 c15f 64a2
                         a14e 3452 0570 6aee 020b b4b4 0000 0000
                         6002 2000 2585 0000 0204 05b4 b4b4
```

1. Source of Trace

The WinDump trace was captured on a dual-function IDS system (172.16.10.3) outside the external firewall of our business network. This computer ran basic IDS software – a Windows NT Port of SNORT using the 07272k ruleset, as well as WINDump – a Windows NT port of TCPDump for additional analysis. The firewall logs were taken from the external firewall (172.16.10.2) Except for the router, these are the only two hosts on this subnet. The router is under the control of an ISP and the logs are not available for inclusion.

2. Detect Generated By

The Detect was generated by the External Firewall. Firewall logs excerpts include breakdown of fields where required. The Snort IDS system did not pick up this attack.

3. Probability of Source Address Spoofing

In order for the information obtained by this scan to be of any use, the attacker must be able to see any response from the scanned systems. Although the attacker could be using a compromised computer and spoofing his own address to get the responses, it is unlikely that the source address is spoofed.

4. Description of Attack

The attacker is checking all hosts on the subnet by directing a single normal SYN packet at each host, and then to the broadcast address, although it is unlikely that the attacker knows the subnet mask. It is possible that the packets are addressed to a much larger broadcast address and the external router, whose logs are not available, forwarded the packets for valid hosts to this subnet. ARP packets were not collected so this possibility cannot be confirmed nor denied, although the break of a few seconds between the scan of 172.26.10.3 and the broadcast address of 172.26.10.255 seems to indicate that the intermediate addresses were also scanned.

All packets are directed to port 27374, which is the signature port for this type of scan. The attacker is attempting to locate any system infected with and running the well-known SubSeven version 2 Trojan software.

The firewall log shows that 172.16.10.3 (the IDS) properly responded to the scan by sending a Reset, which tells the scanner the port is not open. The firewall logged this as a Packet Forwarding Attack because the IDS, to avoid sending just such responses back to the attacker, has it's default gateway address set to the firewall's external interface. When the IDS attempts to respond to any packet, it uses its default gateway, the firewall, to route such packets. The firewall rules forbid such behavior and log it as an attack. This little bit of extra log information helps to provide a better picture of the attack.

CVE Number – None allocated to this type of Scan

5. Attack Mechanism

The Sub-Seven Trojan, in this case version 2.0 or 2.1 depending on the document you read, is one of the most popular remote access trojans. The attacker runs the master software on their computer. The master software can scan the internet for systems running the slave software. Once a slave is located the master can connect to it, and through the slave software, remotely control the system on which it resides. This version of the trojan reportedly supports ' port redirection ' which means you can send the infected system software on a specific port and the computer will redirect the packet to another computer, on what ever port you designate. The slave software can also be used to scan for other infected computers systems.

This has very serious consequences. It means that, should you be able to converse with a single infected system on an organization's internal network, you can attack and/or compromise every other computer on that network.

6. Correlation

Trojan Scans are very common. Trojans are normally distributed via other means, including e-mail. Numerous similar examples can be seen in SANS Detects Analyzed reports by searching on the keywords Sub Seven and 27374.

7. Evidence of Active Targeting

There is no evidence of active targeting. This appears to be a general search for systems infected with the Sub-Seven Trojan. The attack is probably automated or scripted and directed at a large number of networks.

8. Severity

Criticality	= 5 (firewall, IDS)
Lethality	= 5 (If the Firewall were infected and could be remotely controlled by an attacker, the whole internal network could be compromised.
System Countermeasures	= 5 (both computers completely hardened operating systems with all patches applied)
Network Countermeasures	= 4 (all appropriate rulesets in place, firewall properly dropped all packets silently rather than responding with reset. IDS was not protected by firewall but all available countermeasures in place. In addition, there is a secondary firewall between the external firewall and the internal networks)

Equation Severity = (5 + 5) – (5 + 4) = 1

9. Defensive Recommendation

The current security posture of the firewall is correct. The firewall ruleset properly detected and logged the attack although, if possible, the logs could contain more detail on the actual packets observed. More detailed logging is not available on this brand of firewall however.

The IDS computer responded to the SYN-FIN packet with a reset which could tell the attacker the system exists. However, because the default gateway of the IDS was set to the external interface of the firewall, the IDS computer attempted to forward the reset packet to the attacker via the firewall, as evidenced by the Packet Forwarding Attack log from the firewall. The firewall did not forward the packet. An improvement to this scenario could be made by not giving the network card in the IDS an IP address, thereby making it inaccessible.

The Snort IDS System using the 07272k ruleset did not detect this scan. The rule set should be changed to include this signature and the signature of all other known trojans. The IDS system should also be running up-to-date anti-virus software to ensure that should a trojan be loaded on it, the trojan would be immediately detected.

10. Multiple Choice Question

Port 27374 is the signature port for what well-known Trojan?

a. BACK Orifice
b. Sub-Seven
c. HACK-A-TACK
d. Netbus

Answer: b – SubSeven – Version 2.1 (2.0?)

Richard D. Salisko

Assignment Two – Evaluate an Attack

1. Give the URL, location, or command that your acquired the attack from.

Http://packetstorm.securify.com/DoS/IGMPNukeV1_0.zip

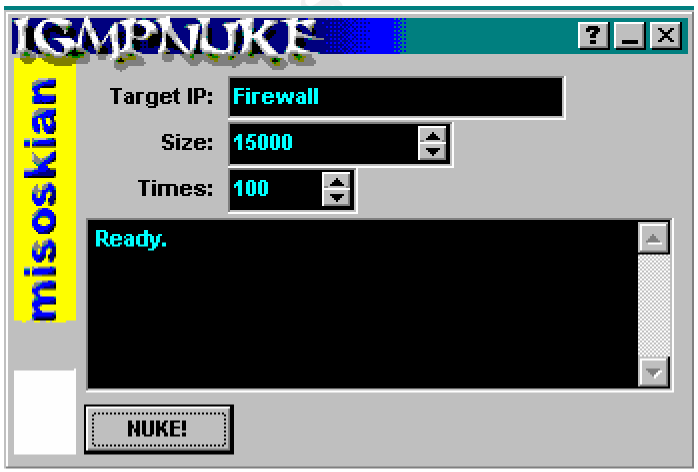
(Packet Storm Security Site – List of top 100 Files – Number 69)

2. Describe the attack, including how it works.

The program, IGMPNukeV1.0, is a small Windows executable that takes advantage of a vulnerability in the TCP/IP stack of most Microsoft Windows computers, including NT. The vulnerability, CVE 1999-0918, causes Windows 9X/NT (and reportedly 2000) computers to experience a variety of problems, from slowdowns to crashes (Blue Screen of Death) when they receive an oversized, fragmented IGMP packet. Patches for the vulnerability are available from Microsoft.

IGMP or Internet Group Multicast Protocol RFC966 is the extensions to IP to support multicasting groups. RFCs 988 (version 0), 1112 (version 1), and 2236 (version 2) refer to it as the Internet Group Management Protocol. None of the RFCs mention fragmentation, so presumably fragmented packets do not violate the standard.

The IGMPNuke Interface, shown below, allows the user to choose any size and number of IGMP packets to send. The program defaults, except for the ' Target IP ', are as shown. To send the packets the user has only to enter the IP Address of the target, change any desired parameters, and press the Nuke button.



Richard D. Salisko

The program breaks the IGMP packets into regular sized, but zero loaded, ethernet fragments (payload of 1480 bytes). The IGMP header is unusual. The IGMP header encapsulated in the IP Datagram should normally contain at least four bytes of data consisting of the IGMP type (1 byte), the Max Response time (1 byte), the Checksum (2 bytes) and possibly a Group Address (4+ bytes). The packets produced by this tool have no IGMP header information at all. No information is available as to whether or not this peculiarity contributes to the Windows vulnerability. Most sources refer to the problem being the fragmentation alone, with no mention of any particular packet attribute.

3. Provide an annotated network trace of the attack in action.

Attack using fragmented packets

Size of IGMP Packet: 3000 bytes
Number of packets 10 (only two shown for brevity)

The attack was directed at a Windows NT based firewall with all appropriate patches installed, using the parameters shown. Below is an excerpt from the firewall ' Sessions Completed ' log. The log fields legend is included below.

Firewall Log Fields:

1 = Date of intercept
2 = Local time of Intercept
3 = Action taken (T = Terminated Session)
4 = src interface
5 = dst interface
6 = src IP
7 = dst IP
8 = protocol
9 = src port
10 = dst port
11 = number of source packets
12 = number of destination packets
13 = number of source bytes
14 = number of destination bytes

1	2	3	4	5	6	7	8	9	10	11	12	13	14
2000/09/12	**18:17:54:**	**T**	**El90x2**	**lo0**	**172.16.10.14**	**172.16.10.2**	**igmp**	**0**	**0**	**10**	**0**	**3000**	**0**

The firewall log shows a total of 10 packets of length 3000.

Trace One:

Richard D. Salisko

The following Network trace was generated by Windump - a Windows NT port of TCPDump - using the command line shown. The command line parameters are identical to those of TCPDump

Windump –vv –n –s 300 –x (icmp and udp and tcp and igmp)

Note: Exclusions in filter not shown

```
18:18:03.886004 172.16.10.14 > 172.16.10.2: igmp-0 [v0][|igmp] (frag 3584:1480@0+) (ttl 128)
                         4500 05dc 0e00 2000 8002 fc10 a14e 345e
                         a14e 3452 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000
18:18:03.887235 172.16.10.14 > 172.16.10.2: (frag 3584:1480@1480+) (ttl 128)
                         4500 05dc 0e00 20b9 8002 fb57 a14e 345e
                         a14e 3452 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000 0000
```

```
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000
18:18:03.887291 172.16.10.14 > 172.16.10.2: (frag 3584:40@2960) (ttl 128)
                                      4500 003c 0e00 0172 8002 203f a14e 345e
                                      a14e 3452 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000
18:18:03.889176 172.16.10.14 > 172.16.10.2: igmp-0 [v0][|igmp] (frag 3840:1480@0+) (ttl 128)
                                      4500 05dc 0f00 2000 8002 fb10 a14e 345e
                                      a14e 3452 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000
18:18:03.890403 172.16.10.14 > 172.16.10.2: (frag 3840:1480@1480+) (ttl 128)
                                      4500 05dc 0f00 20b9 8002 fa57 a14e 345e
                                      a14e 3452 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
                                      0000 0000 0000 0000 0000 0000 0000 0000
```

```
                                0000 0000 0000 0000 0000 0000 0000 0000
                                0000 0000 0000 0000 0000 0000 0000 0000
                                0000 0000 0000 0000 0000 0000 0000 0000
                                0000 0000 0000 0000 0000 0000 0000 0000
                                0000 0000 0000 0000 0000 0000 0000 0000
                                0000 0000 0000 0000 0000 0000 0000 0000
                                0000 0000 0000 0000 0000 0000 0000 0000
                                0000 0000 0000 0000 0000 0000 0000 0000
                                0000 0000 0000 0000 0000 0000 0000 0000
                                0000 0000 0000 0000 0000 0000 0000
18:18:03.890457 172.16.10.14 > 172.16.10.2: (frag 3840:40@2960) (ttl 128)
                                4500 003c 0f00 0172 8002 1f3f a14e 345e
                                a14e 3452 0000 0000 0000 0000 0000 0000
                                0000 0000 0000 0000 0000 0000 0000 0000
                                0000 0000 0000 0000 0000 0000
```

The Trace shows a normally fragmented packet with no IGMP headers evident. The only parameter which leads us to believe this is IGMP is the IP Type 02 in byte nine of the IP Header.

Trace Two:

The following Network trace was generated by Windump - a Windows NT port of TCPDump - using the command line shown. The command line parameters are identical to those of TCPDump.

Windump –vv –n –s 300 (icmp and udp and tcp and igmp)

Note: Exclusions in filter not shown

```
18:18:03.882617 172.16.10.14 > 172.16.10.2: igmp-0 [v0][|igmp] (frag 3328:1480@0+) (ttl 128)
18:18:03.883803 172.16.10.14 > 172.16.10.2: (frag 3328:1480@1480+) (ttl 128)
18:18:03.883886 172.16.10.14 > 172.16.10.2: (frag 3328:40@2960) (ttl 128)
18:18:03.885748 172.16.10.14 > 172.16.10.2: igmp-0 [v0][|igmp] (frag 3584:1480@0+) (ttl 128)
18:18:03.886979 172.16.10.14 > 172.16.10.2: (frag 3584:1480@1480+) (ttl 128)
18:18:03.887030 172.16.10.14 > 172.16.10.2: (frag 3584:40@2960) (ttl 128)
```

Richard D. Salisko

Attack using packets sizes not requiring fragmentation

Size of IGMP Packet: 300 bytes

The tool's stated intent is to ' nuke ' windows based machines by sending large IGMP packets. It's interesting to note, however, that if you attempt to send packets so small they don't require fragmentation, you get some unexpected results.

Firewall ' Sessions Completed ' log

1	2	3	4	5	6	7	8	9	10	11	12	13	14
2000/09/12	**18:19:20:**	**T**	**EI90x2**	**lo0**	**172.16.10.14**	**172.16.10.2**	**igmp**	**0**	**0**	**10**	**0**	**300**	**0**

The firewall log appears identical, showing a total of 10 packets of length 300.

Trace Three:

Here's the Windump trace using the hex output option.

```
18:19:30.366048 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 5888)
                         4500 001e 1700 0000 8002 18cf a14e 345e
                         a14e 3452 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000
18:19:30.367368 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 6144)
                         4500 001e 1800 0000 8002 17cf a14e 345e
                         a14e 3452 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000
18:19:30.367882 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 6400)
                         4500 001e 1900 0000 8002 16cf a14e 345e
                         a14e 3452 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000
18:19:30.368419 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 6656)
                         4500 001e 1a00 0000 8002 15cf a14e 345e
                         a14e 3452 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000
18:19:30.368979 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 6912)
                         4500 001e 1b00 0000 8002 14cf a14e 345e
                         a14e 3452 0000 0000 0000 0000 0000 0000
                         0000 0000 0000 0000 0000 0000 0000
```

```
18:19:30.369986 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 7168)
                              4500 001e 1c00 0000 8002 13cf a14e 345e
                              a14e 3452 0000 0000 0000 0000 0000 0000
                              0000 0000 0000 0000 0000 0000 0000
18:19:30.371423 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 7424)
                              4500 001e 1d00 0000 8002 12cf a14e 345e
                              a14e 3452 0000 0000 0000 0000 0000 0000
                              0000 0000 0000 0000 0000 0000 0000
18:19:30.372505 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 7680)
                              4500 001e 1e00 0000 8002 11cf a14e 345e
                              a14e 3452 0000 0000 0000 0000 0000 3a3a
                              3a3a 3a3a 3a3a 3a3a 3a3a 3a3a 3a3a
18:19:30.373849 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 7936)
                              4500 001e 1f00 0000 8002 10cf a14e 345e
                              a14e 3452 0000 0000 0000 0000 0000 0000
                              0000 0000 0000 0000 0000 0000 0000
18:19:30.375351 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 8192)
                              4500 001e 2000 0000 8002 0fcf a14e 345e
                              a14e 3452 0000 0000 0000 0000 0000 0000
                              0000 0000 0000 0000 0000 0000 0000
```

When the whole IGMP datagram is contained in a single IP packet, a checksum error is generated. The firewall ignores the error because it ignores and silently discards the packet, but the WINDump program complains.

Trace Four:

Here's the Windump trace without the hex output option.

```
18:19:30.361664 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 5888)
18:19:30.362988 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 6144)
18:19:30.363552 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 6400)
18:19:30.364089 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 6656)
18:19:30.364649 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 6912)
18:19:30.365655 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 7168)
18:19:30.367093 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 7424)
18:19:30.368175 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 7680)
18:19:30.369518 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 7936)
18:19:30.371021 172.16.10.14 > 172.16.10.2: igmp-0 [v0] bad igmp cksum 0! (ttl 128, id 8192)
```

The program documentation, although extremely sparse, has no mention of this ‘ feature ‘, so I have to assume it’s a bug.

Assignment Three – Analyze This

In this scenario, a Snort IDS system was installed at an unspecified site for approximately one month. My organization was tasked with evaluating the Snort output files for any signs of compromised systems or network problems.

Trace One:

```
06/27-00:51:25.609698 [**] WinGate 1080 Attempt[**] 207.114.4.46:2546 -> MY.NET.97.235:1080
06/27-00:51:28.611857 [**] WinGate 1080 Attempt[**] 207.114.4.46:2546 -> MY.NET.97.235:1080
```

This appears to be a scan for Socks servers, but is likely a proxy server verifying that the host MY.NET.97.235, which attempted to connect to a server on it's network, is really who it claims to be. This can be verified by resolving the originator's address, which equates to ProxyScan.MD.US.Undernet.Org.

Trace Two:

```
06/28-06:35:13.540772 [**] Tiny Fragments - Possible Hostile Activity[**] 63.236.34.174 -> MY.NET.1.8
06/28-06:35:13.540827 [**] Tiny Fragments - Possible Hostile Activity[**] 63.236.34.174 -> MY.NET.1.8
06/28-06:35:13.540878 [**] Tiny Fragments - Possible Hostile Activity[**] 63.236.34.174 -> MY.NET.1.8
06/28-06:37:13.538078 [**] Tiny Fragments - Possible Hostile Activity[**] 63.236.34.174 -> MY.NET.1.8
06/28-06:37:13.538175 [**] Tiny Fragments - Possible Hostile Activity[**] 63.236.34.174 -> MY.NET.1.8
06/28-06:37:13.538272 [**] Tiny Fragments - Possible Hostile Activity[**] 63.236.34.174 -> MY.NET.1.8
```

Mutated fragmentation is often used in denial of service attacks. The Windows IGMP vulnerability, as well as the Teardrop attack both use fragmentation in their delivery, although in different ways. Fragmentation is a normal phenomena in network traffic, however, and this trace could very well be a false positive.

Richard D. Salisko

Trace Three:

```
06/28-06:52:48.291107  [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.1.1:53
06/28-06:52:48.330764  [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.1.3:53
06/28-06:52:48.518996  [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.1.11:53
06/28-06:52:48.522936  [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.1.12:53
...
06/28-07:14:23.646996  [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.254.249:53
06/28-07:14:23.649590  [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.254.248:53
06/28-07:14:23.702364  [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.254.251:53
06/28-07:14:23.732747  [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.254.255:53


06/28-07:17:46.987174 [**] spp_portscan: portscan status from 202.0.178.98: 47 connections across 47 hosts: TCP(46) UDP(1) STEALTH[**]
06/28-07:17:48.221891 [**] spp_portscan: End of portscan from 202.0.178.98 (TOTAL HOSTS:102 TCP:102 UDP:1)[**]
06/28-07:18:00.192353 [**] spp_portscan: PORTSCAN DETECTED from 202.0.178.98 (STEALTH)[**]
06/28-07:18:15.602578 [**] spp_portscan: End of portscan from 202.0.178.98 (TOTAL HOSTS:1338 TCP:1347 UDP:5)[**]
…
```

This trace shows a very fast sequential SYN-FIN scan, from a Hong Kong service provider, of almost all subnets in a Class B sized address range. Approximately 20000 hosts were scanned in less than 22 minutes. A SYN-FIN scan was used to help reduce the possibility of detection. Port 53 was probably used because it's primary purpose is for zone transfers and large DNS queries and therefore many firewalls leave this port open to allow traffic through. The primary purpose of this scan was likely to map the target network. It's interesting to note that Snort also reported some UDP traffic in the scan, although this cannot be verified. No other traffic was seen to or from this address in the traces although several other SYN-FIN scans were noted on other ports.

Richard D. Salisko

Trace Four:

```
06/29-09:44:27.347467 [**] SYN-FIN scan![**] 210.222.31.100:1524 -> MY.NET.1.3:1524
06/29-09:44:27.394689 [**] SYN-FIN scan![**] 210.222.31.100:1524 -> MY.NET.1.5:1524

07/11-19:10:54.498267 [**] SYN-FIN scan![**] 210.222.31.100:1524 -> MY.NET.1.3:1524
07/11-19:10:54.509349 [**] SYN-FIN scan![**] 210.222.31.100:1524 -> MY.NET.1.4:1524
07/11-19:10:54.518499 [**] SYN-FIN scan![**] 210.222.31.100:1524 -> MY.NET.1.5:1524

08/01-00:04:42.221090 [**] SYN-FIN scan![**] 207.0.62.254:1524 -> MY.NET.1.3:1524
08/01-00:04:42.239230 [**] SYN-FIN scan![**] 207.0.62.254:1524 -> MY.NET.1.4:1524
08/01-00:04:42.262533 [**] SYN-FIN scan![**] 207.0.62.254:1524 -> MY.NET.1.5:1524
```

These are SYN-FIN Scans searching for systems listening on port 1524. This port is typically associated with the Trinoo Distributed Denial of Service tool. When an attacker compromises a computer which he intends to use as a trinoo master for launching DoS attacks, part of the procedure is to install a listening shell, typically the Ingreslock Backdoor, which listens on this port. The open port is then used to install an additional attack tool kit.

Trace Five:

```
06/27-03:54:29.006819 [**] GIAC 000218 VA-CIRT port 34555[**] 192.101.175.131:25 -> MY.NET.100.230:34555
06/27-03:54:29.107374 [**] GIAC 000218 VA-CIRT port 34555[**] 192.101.175.131:25 -> MY.NET.100.230:34555
06/27-03:54:29.290881 [**] GIAC 000218 VA-CIRT port 34555[**] 192.101.175.131:25 -> MY.NET.100.230:34555
06/27-03:54:29.498544 [**] GIAC 000218 VA-CIRT port 34555[**] 192.101.175.131:25 -> MY.NET.100.230:34555
06/27-03:54:29.703712 [**] GIAC 000218 VA-CIRT port 34555[**] 192.101.175.131:25 -> MY.NET.100.230:34555
06/27-03:54:31.119839 [**] GIAC 000218 VA-CIRT port 34555[**] 192.101.175.131:25 -> MY.NET.100.230:34555
06/27-03:54:31.209413 [**] GIAC 000218 VA-CIRT port 34555[**] 192.101.175.131:25 -> MY.NET.100.230:34555
06/27-03:54:31.210076 [**] GIAC 000218 VA-CIRT port 34555[**] 192.101.175.131:25 -> MY.NET.100.230:34555
```

This is a Snort report of suspected WinTrinoo activity. Wintrinoo is a Windows version of the Trinoo Distributed DoS tool mentioned above. Wintrinoo differs from Trinoo in the ports on which it listens and sends (34555 and 35555). Activity was noted on both ports in this month's traces.

Trace Six:

```
06/30-01:59:39.580221 [**] External RPC call[**] 204.176.11.10:111 -> MY.NET.6.15:111
06/30-01:59:44.793285 [**] External RPC call[**] 204.176.11.10:1556 -> MY.NET.6.15:111
06/30-01:59:44.819258 [**] External RPC call[**] 204.176.11.10:1556 -> MY.NET.6.15:111
06/30-01:59:44.819365 [**] External RPC call[**] 204.176.11.10:1016 -> MY.NET.6.15:111
06/30-01:59:44.848794 [**] External RPC call[**] 204.176.11.10:1016 -> MY.NET.6.15:111
06/30-01:59:44.902365 [**] External RPC call[**] 204.176.11.10:1016 -> MY.NET.6.15:111
```

This is an attempt to access the portmapper service on host my.net.6.15 on tcp port 111. Portmapper keeps track of the various network services available on a host and the ports that each uses. A connection attempt to port 111 will be redirected to the desired service. By using certain commands (rpcinfo -p) on this port an attacker can view all available services on the victim. They can then direct their attack against services which have known vulnerabilities. In this case the attacker first tried port 111 as both source port and destination, then used normal ephemeral ports for the other scans.

Trace Seven:

```
Jun 30 04:23:51  195.132.120.31:4164 -> MY.NET.130.12:21  SYN **S*****
Jun 30 04:23:48  195.132.120.31:4161 -> MY.NET.130.9:21  SYN **S*****
Jun 30 04:23:48  195.132.120.31:4163 -> MY.NET.130.11:21  SYN **S*****
Jun 30 04:23:49  195.132.120.31:4165 -> MY.NET.130.13:21  SYN **S*****
...
Jun 30 04:47:14  195.132.120.31:3863 -> MY.NET.143.214:21  SYN **S*****
Jun 30 04:47:14  195.132.120.31:3865 -> MY.NET.143.216:21  SYN **S*****
Jun 30 04:47:14  195.132.120.31:3867 -> MY.NET.143.218:21  SYN **S*****
Jun 30 04:47:15  195.132.120.31:3869 -> MY.NET.143.220:21  SYN **S*****
```

This is a SYN scan from a cable provider in Paris. The scan covers six class C subnets on port 21 (FTP). The hosts on each subnet were scanned in a random order, presumably to defeat IDS systems looking for scan signatures. The subnets, however, were scanned in numerical order. The intent was probably to map available hosts. Many such scans were noted during the month, some scanning an entire class B address range. Most scans used port 21 (FTP). Other ports noted were 23 (Telnet), 53 (DNS), 98 (linuxconf), and 110 (Pop3)

Richard D. Salisko

Trace Eight:

```
06/30-16:33:57.773279 [**] site exec - Possible wu-ftpd exploit - GIAC000623[**] 151.164.223.206:4499 -> MY.NET.99.16:21
06/30-16:34:00.037398 [**] Possible wu-ftpd exploit - GIAC000623[**] 151.164.223.206:4499 -> MY.NET.99.16:21
06/30-16:35:11.406398 [**] site exec - Possible wu-ftpd exploit -

GIAC000623[**] 151.164.223.206:4500 -> MY.NET.144.59:21
06/30-16:35:13.560305 [**] site exec - Possible wu-ftpd exploit -

GIAC000623[**] 151.164.223.206:4500 -> MY.NET.144.59:21
06/30-16:35:13.626498 [**] Possible wu-ftpd exploit - GIAC000623[**] 151.164.223.206:4500 -> MY.NET.144.59:21
```

There are numerous CVEs listed for the WU-FTPD daemon. Most are related to buffer overflows which lead to root access. Although these alerts may prove to be false positives, traffic like this should be investigated.

Trace Nine:

```
Jul  9 08:13:32  212.29.71.87:3488 -> MY.NET.97.1:44767 UDP
Jul  9 08:13:32  212.29.71.87:3489 -> MY.NET.97.2:44767 UDP
Jul  9 08:13:32  212.29.71.87:3490 -> MY.NET.97.3:44767 UDP
Jul  9 08:13:32  212.29.71.87:3491 -> MY.NET.97.4:44767 UDP
Jul  9 08:13:32  212.29.71.87:3493 -> MY.NET.97.6:44767 UDP
...
Jul  9 08:13:37  212.29.71.87:3739 -> MY.NET.97.252:44767 UDP
Jul  9 08:13:37  212.29.71.87:3738 -> MY.NET.97.251:44767 UDP
Jul  9 08:13:37  212.29.71.87:3740 -> MY.NET.97.253:44767 UDP
Jul  9 08:13:37  212.29.71.87:3741 -> MY.NET.97.254:44767 UDP
Jul  9 08:13:37  212.29.71.87:3742 -> MY.NET.97.255:44767 UDP
```

Here's a scan from Turkey directed at UDP port 44767. It was a sequential scan of a full class C address range including the broadcast address .255. Speculations observed on the SANS website point to this port as possibly being used by an unidentified new trojan.

Richard D. Salisko

Trace Ten:

```
Jul  9 14:49:11  212.17.108.71:2877 -> MY.NET.219.154:8080 SYN **S*****
Jul  9 14:49:14  212.17.108.71:2875 -> MY.NET.219.154:3128 SYN **S*****
Jul  9 14:49:11  212.17.108.71:2871 -> MY.NET.97.238:8080 SYN **S*****
Jul  9 14:49:11  212.17.108.71:2874 -> MY.NET.20.10:8080 SYN **S*****
Jul  9 14:49:12  212.17.108.71:2872 -> MY.NET.20.10:3128 SYN **S*****
Jul  9 14:49:12  212.17.108.71:2863 -> MY.NET.98.135:3128 SYN **S*****
Jul  9 14:49:12  212.17.108.71:2865 -> MY.NET.98.135:8080 SYN **S*****
Jul  9 14:49:13  212.17.108.71:2878 -> MY.NET.97.239:3128 SYN **S*****
Jul  9 14:49:12  212.17.108.71:2880 -> MY.NET.97.239:8080 SYN **S*****
Jul  9 14:49:12  212.17.108.71:2866 -> MY.NET.98.93:3128 SYN **S*****
```

Over 5000 Proxy related scans were noted over the course of the month, originating from many different sources, including this one from Austria. These were primarily SYN scans on TCP ports 3128 (Squid) and 8080 (Wingate) looking for available proxy servers. There are at least four entries in the CVE for the Wingate server alone - (CVE 1999-0290, 1999-0291, 1999-0441, 1999-0494). Almost 50% of the scans noted were directed at a singlehost - MY.NET.253.105.

Richard D. Salisko

Trace Eleven:

```
Jul  9 20:46:24  193.173.174.119:53 -> MY.NET.1.1:53  SYNFIN **SF****
Jul  9 20:46:24  193.173.174.119:53 -> MY.NET.1.2:53  SYNFIN **SF****
Jul  9 20:46:24  193.173.174.119:53 -> MY.NET.1.7:53  SYNFIN **SF****
...
Jul  9 20:46:24  193.173.174.119:53 -> MY.NET.1.9:53  SYNFIN **SF****
Jul  9 20:46:25  193.173.174.119:4050 -> MY.NET.1.9:53  SYN **S*****
Jul  9 20:46:25  193.173.174.119:2032 -> MY.NET.1.9:53  UDP
...
Jul  9 20:46:25  193.173.174.119:53 -> MY.NET.1.48:53  SYNFIN **SF****
Jul  9 20:46:25  193.173.174.119:4048 -> MY.NET.1.4:53  SYN **S*****
Jul  9 20:46:26  193.173.174.119:2026 -> MY.NET.1.4:53  UDP
...
Jul  9 20:46:48  193.173.174.119:53 -> MY.NET.1.253:53  SYNFIN **SF****
Jul  9 20:46:48  193.173.174.119:53 -> MY.NET.1.254:53  SYNFIN **SF****
Jul  9 20:46:48  193.173.174.119:53 -> MY.NET.1.255:53  SYNFIN **SF****
```

This is a scan from an ISP in the Netherlands. It started with a SYN-FIN scan to the My.Net.1 subnet on port 53, then began interspersing SYN and UDP Scans on the same port number. This probably means that when a host responded with anything other than a reset to the first scan (SYN-FIN), a SYN scan and UDP 53 scan would follow to discover if additional responses could be obtained. This is obviously a scripted scan that uses responses from one scan as input to initiate another.

Trace Twelve:

```
Jul 11 21:22:42  198.62.155.106:53 -> MY.NET.1.5:53  UDP
Jul 11 21:22:42  198.62.155.106:37440 -> MY.NET.5.4:815  SYN **S*****
Jul 11 21:22:42  198.62.155.106:37462 -> MY.NET.5.4:1  SYN **S*****
Jul 11 21:22:42  198.62.155.106:37506 -> MY.NET.5.4:596  SYN **S*****
…
Jul 11 21:22:42  198.62.155.10:37431 -> MY.NET.5.4:665  SYN **S*****
Jul 11 21:22:42  198.62.155.10:37464 -> MY.NET.5.4:1353  SYN **S*****
Jul 11 21:22:42  198.62.155.10:37508 -> MY.NET.5.4:1667  SYN **S*****
…
Jul 11 21:22:42  198.62.155.11:37432 -> MY.NET.5.4:6143  SYN **S*****
```

```
Jul 11 21:22:42  198.62.155.11:37465 -> MY.NET.5.4:1462  SYN **S*****
Jul 11 21:22:42  198.62.155.11:37498 -> MY.NET.5.4:872  SYN **S*****
…
Jul 11 21:22:42  198.62.155.111:37433 -> MY.NET.5.4:703  SYN **S*****
Jul 11 21:22:42  198.62.155.111:37466 -> MY.NET.5.4:578  SYN **S*****
Jul 11 21:22:42  198.62.155.111:37499 -> MY.NET.5.4:931  SYN **S*****
…
Jul 11 21:22:42  198.62.155.102:37435 -> MY.NET.5.4:1381  SYN **S*****
Jul 11 21:22:42  198.62.155.102:37468 -> MY.NET.5.4:764  SYN **S*****
Jul 11 21:22:42  198.62.155.102:37501 -> MY.NET.5.4:1525  SYN **S*****
…
Jul 11 21:22:42  198.62.155.101:37434 -> MY.NET.5.4:632  SYN **S*****
Jul 11 21:22:42  198.62.155.101:37467 -> MY.NET.5.4:61  SYN **S*****
Jul 11 21:22:42  198.62.155.101:37500 -> MY.NET.5.4:6147  SYN **S*****
…
Jul 11 21:22:42  198.62.155.103:37436 -> MY.NET.5.4:670  SYN **S*****
Jul 11 21:22:42  198.62.155.103:37458 -> MY.NET.5.4:468  SYN **S*****
Jul 11 21:22:42  198.62.155.103:37469 -> MY.NET.5.4:1396  SYN **S*****
…
Jul 11 21:22:42  198.62.155.104:37437 -> MY.NET.5.4:32777  SYN **S*****
Jul 11 21:22:42  198.62.155.104:37459 -> MY.NET.5.4:2047  SYN **S*****
Jul 11 21:22:42  198.62.155.104:37470 -> MY.NET.5.4:462  SYN **S*****
…
Jul 11 21:22:42  198.62.155.109:37438 -> MY.NET.5.4:5801  SYN **S*****
Jul 11 21:22:42  198.62.155.109:37460 -> MY.NET.5.4:57  SYN **S*****
Jul 11 21:22:42  198.62.155.109:37471 -> MY.NET.5.4:1366  SYN **S*****
…
Jul 11 21:22:42  198.62.155.105:37439 -> MY.NET.5.4:611  SYN **S*****
Jul 11 21:22:42  198.62.155.105:37461 -> MY.NET.5.4:2004  SYN **S*****
Jul 11 21:22:42  198.62.155.105:37472 -> MY.NET.5.4:610  SYN **S*****
…
Jul 11 21:22:42  198.62.155.107:37463 -> MY.NET.5.4:1514  SYN **S*****
Jul 11 21:22:42  198.62.155.107:37507 -> MY.NET.5.4:1455  SYN **S*****
Jul 11 21:22:43  198.62.155.107:37551 -> MY.NET.5.4:1348  SYN **S*****
…
Jul 11 21:22:46  198.62.155.106:38254 -> MY.NET.5.4:479  SYN **S*****
Jul 11 21:22:46  198.62.155.106:38309 -> MY.NET.5.4:67  SYN **S*****
Jul 11 21:22:46  198.62.155.106:38375 -> MY.NET.5.4:1650  SYN **S*****
…
Jul 11 21:22:46  198.62.155.10:38223 -> MY.NET.5.4:170  SYN **S*****
```

```
Jul 11 21:22:46  198.62.155.10:38366 -> MY.NET.5.4:1547  SYN **S*****
Jul 11 21:22:46  198.62.155.10:38377 -> MY.NET.5.4:915  SYN **S*****
…
Jul 11 21:22:46  198.62.155.11:38224 -> MY.NET.5.4:870  SYN **S*****
Jul 11 21:22:46  198.62.155.11:38235 -> MY.NET.5.4:406  SYN **S*****
Jul 11 21:22:46  198.62.155.11:38367 -> MY.NET.5.4:1511  SYN **S*****
…
Jul 11 21:22:46  198.62.155.111:38236 -> MY.NET.5.4:2015  SYN **S*****
Jul 11 21:22:46  198.62.155.111:38368 -> MY.NET.5.4:165  SYN **S*****
Jul 11 21:22:46  198.62.155.111:38390 -> MY.NET.5.4:789  SYN **S*****
…
Jul 11 21:22:46  198.62.155.102:38238 -> MY.NET.5.4:379  SYN **S*****
Jul 11 21:22:46  198.62.155.102:38304 -> MY.NET.5.4:686  SYN **S*****
Jul 11 21:22:46  198.62.155.102:38337 -> MY.NET.5.4:1412  SYN **S*****
…
Jul 11 21:22:46  198.62.155.101:38237 -> MY.NET.5.4:979  SYN **S*****
Jul 11 21:22:46  198.62.155.101:38369 -> MY.NET.5.4:2034  SYN **S*****
Jul 11 21:22:46  198.62.155.101:38380 -> MY.NET.5.4:157  SYN **S*****
…
Jul 11 21:22:46  198.62.155.103:38239 -> MY.NET.5.4:1651  SYN **S*****
Jul 11 21:22:46  198.62.155.103:38261 -> MY.NET.5.4:20005  SYN **S*****
Jul 11 21:22:46  198.62.155.103:38305 -> MY.NET.5.4:107  SYN **S*****
…
Jul 11 21:22:46  198.62.155.104:38251 -> MY.NET.5.4:763  SYN **S*****
Jul 11 21:22:46  198.62.155.104:38306 -> MY.NET.5.4:1112  SYN **S*****
Jul 11 21:22:46  198.62.155.104:38372 -> MY.NET.5.4:2013  SYN **S*****
…
Jul 11 21:22:46  198.62.155.109:38307 -> MY.NET.5.4:698  SYN **S*****
Jul 11 21:22:46  198.62.155.109:38373 -> MY.NET.5.4:402  SYN **S*****
Jul 11 21:22:46  198.62.155.109:38384 -> MY.NET.5.4:6009  SYN **S*****
…
Jul 11 21:22:46  198.62.155.105:38374 -> MY.NET.5.4:420  SYN **S*****
Jul 11 21:22:46  198.62.155.105:38385 -> MY.NET.5.4:946  SYN **S*****
Jul 11 21:22:46  198.62.155.105:38451 -> MY.NET.5.4:1474  SYN **S*****
…
Jul 11 21:22:46  198.62.155.107:38233 -> MY.NET.5.4:34  SYN **S*****
Jul 11 21:22:46  198.62.155.107:38255 -> MY.NET.5.4:975  SYN **S*****
Jul 11 21:22:46  198.62.155.107:38365 -> MY.NET.5.4:291  SYN **S*****
…
Jul 11 21:22:49  198.62.155.106:38958 -> MY.NET.5.4:6008  SYN **S*****
```

```
Jul 11 21:22:50  198.62.155.10:38993 -> MY.NET.5.4:5800  SYN **S*****
Jul 11 21:22:49  198.62.155.11:38983 -> MY.NET.5.4:5800  SYN **S*****
Jul 11 21:22:49  198.62.155.111:38962 -> MY.NET.5.4:585  SYN **S*****
Jul 11 21:22:49  198.62.155.103:38932 -> MY.NET.5.4:4132  SYN **S*****
Jul 11 21:22:49  198.62.155.103:38965 -> MY.NET.5.4:3333  SYN **S*****
Jul 11 21:22:49  198.62.155.104:38933 -> MY.NET.5.4:463  SYN **S*****
Jul 11 21:22:49  198.62.155.104:38955 -> MY.NET.5.4:514  SYN **S*****
Jul 11 21:22:49  198.62.155.109:38934 -> MY.NET.5.4:470  SYN **S*****
Jul 11 21:22:49  198.62.155.109:38967 -> MY.NET.5.4:440  SYN **S*****
Jul 11 21:22:50  198.62.155.107:39003 -> MY.NET.5.4:5800  SYN **S*****
```

This is another very interesting SYN scan. At first glance, it appears to be eleven individual hosts from a consulting company in Maryland all scanning a single host simultaneously. None of the originating or destination ports seems to be repeated. However, if you look closely at the originating ports, you'll find that the hosts and ports are nearly sequential. For example, host 101 uses source port 37467, host 102 uses 37468, host 103 uses 37469, host 104 uses 37470, etc. This probably means that there is one host who is spoofing the other 10 addresses. Also, the scan begins with a single packet to and from UDP port 53 (DNS). This is obviously a reconnaissance probe on this computer to see what ports are open.

The true originating address is likely 192,62.155.106 - the host which originally sent the UDP port 53 packet. The probable intent of the attack is Denial of Service (DoS) by keeping the host too busy to respond to legitimate service requests.

Trace Thirteen:

```
Jul 14 21:44:44  198.211.16.69:2029 -> MY.NET.217.1:27374  SYN **S*****
Jul 14 21:44:44  198.211.16.69:2033 -> MY.NET.217.5:27374  SYN **S*****
...
Jul 14 21:45:07  198.211.16.69:2258 -> MY.NET.217.228:27374  SYN **S*****
Jul 14 21:45:07  198.211.16.69:2254 -> MY.NET.217.224:27374  SYN **S*****
Jul 14 21:45:19  198.211.16.69:2285 -> MY.NET.217.1:31337  SYN **S*****
Jul 14 21:45:19  198.211.16.69:2290 -> MY.NET.217.6:31337  SYN **S*****
...
Jul 14 21:45:42  198.211.16.69:2523 -> MY.NET.217.239:31337  SYN **S*****
Jul 14 21:45:42  198.211.16.69:2519 -> MY.NET.217.235:31337  SYN **S*****
```

Richard D. Salisko

There were numerous scans looking for individual trojan programs. Here's one looking for SubSeven V.2.1, followed by another for BO2. Similar scans were noted for port 6970 UDP (Gatecrasher), and 12345 (NetBus) Snort reported the Gatecrasher detect as a 'Large UDP Packet', shown below.

```
08/05-18:30:03.777730 [**] IDS247 - MISC - Large UDP Packet[**] 211.40.176.214:29536 -> MY.NET.98.179:6970
08/05-18:30:03.835886 [**] IDS247 - MISC - Large UDP Packet[**] 211.40.176.214:29536 -> MY.NET.98.179:6970
```

Trace Fourteen:

```
Jul 17 01:07:05  24.2.123.9:1693 -> MY.NET.1.7:53  UDP
Jul 17 01:07:05  24.2.123.9:1693 -> MY.NET.1.8:53  UDP
...
Jul 17 01:12:30  24.2.123.9:1693 -> MY.NET.254.253:53  UDP
Jul 17 01:12:30  24.2.123.9:1693 -> MY.NET.254.254:53  UDP
Jul 17 01:12:30  24.2.123.9:1693 -> MY.NET.254.255:53  UDP
```

Although most observed scans were TCP based There were some UDP. This is a UDP from California that scanned a whole Class B address range using the same source port and a destination port of UDP 53 (DNS) The whole scan took only five minutes.

Trace Fifteen:

```
Jul 17 12:37:42  213.8.203.144:47850 -> MY.NET.1.1:23  FIN ***F****
Jul 17 12:37:42  213.8.203.144:47850 -> MY.NET.1.2:23  FIN ***F****
Jul 17 12:37:42  213.8.203.144:47850 -> MY.NET.1.3:23  FIN ***F****
...
Jul 17 12:38:22  213.8.203.144:47850 -> MY.NET.1.252:23  FIN ***F****
Jul 17 12:38:22  213.8.203.144:47850 -> MY.NET.1.253:23  FIN ***F****
Jul 17 12:38:22  213.8.203.144:47850 -> MY.NET.1.254:23  FIN ***F****
```

Here's a scan from Isreal that used packets with the FIN flag set rather than the SYN, an obvious ploy to escape detection.

Richard D. Salisko

Quesno and Nmap are both very popular hacker tools. They can be used to map networks for available hosts, or to scan an individual host for available ports. Once a host is discovered, the tools can also be used to determine the operating system via OS Fingerprinting. To put it simply, packets with different characteristics will generate different responses from different TCP/IP stacks. By analyzing the response to a specific sequence of packets, the programs mentioned above can identify the TCP/IP stack in use and from that deduce the operating system. This information can then be used to compromise the host using any known vulnerability for that OS.

Trace Sixteen:

```
07/17-15:20:37.812781 [**] Queso Fingerprint[**] 193.233.7.254:3121 -> MY.NET.99.20:113
07/17-15:37:53.409978 [**] Queso Fingerprint[**] 193.233.7.65:3138 -> MY.NET.99.23:113
07/17-15:41:44.730499 [**] Queso Fingerprint[**] 193.233.7.65:3139 -> MY.NET.99.23:113
07/17-21:11:17.806127 [**] Queso Fingerprint[**] 192.203.80.142:3240 -> MY.NET.99.23:113


Jul 17 15:20:37  193.233.7.254:3121 -> MY.NET.99.20:113  SYN 21S***** RESERVEDBITS
Jul 17 15:37:53  193.233.7.65:3138 -> MY.NET.99.23:113  SYN 21S***** RESERVEDBITS
Jul 17 15:41:44  193.233.7.65:3139 -> MY.NET.99.23:113  SYN 21S***** RESERVEDBITS
Jul 17 21:11:17  192.203.80.142:3240 -> MY.NET.99.23:113  SYN 21S***** RESERVEDBITS
```

The above trace shows the Snort alert and corresponding packets. The pattern in the second trace is characteristic of the Quesno tool. Quesno typically sends a total of seven packets, each with different flags set. Each of the above alerts shows the seventh packet in the series, which has the SYN flag and the reserved (unused) flags set. The packets above are from two different hosts (both in Russia) but directed to port 113on the same host.

Trace Seventeen:

```
Jul 17 19:23:47  199.178.222.88:7777 -> MY.NET.153.112:1204  UDP
Jul 17 19:23:47  199.178.222.88:7777 -> MY.NET.153.109:1058  UDP
Jul 17 19:23:47  199.178.222.88:7777 -> MY.NET.153.111:2922  UDP
Jul 17 19:23:47  199.178.222.88:7777 -> MY.NET.153.111:2925  UDP
Jul 17 19:23:47  199.178.222.88:7777 -> MY.NET.153.111:2923  UDP
Jul 17 19:23:47  199.178.222.88:7777 -> MY.NET.153.111:2924  UDP
...
Jul 17 19:24:11  199.178.222.88:7777 -> MY.NET.153.111:2928  UDP
Jul 17 19:24:11  199.178.222.88:7777 -> MY.NET.153.109:1059  UDP
```

```
Jul 17 19:24:11  199.178.222.88:7778 -> MY.NET.153.109:2000  UDP
Jul 17 19:24:14  199.178.222.88:7777 -> MY.NET.153.109:1059  UDP
Jul 17 19:24:14  199.178.222.88:7777 -> MY.NET.153.111:2929  UDP
...
Jul 17 19:29:56  199.178.222.88:7777 -> MY.NET.153.112:1206  UDP
Jul 17 19:29:59  199.178.222.88:7777 -> MY.NET.153.111:2928  UDP
Jul 17 19:29:59  199.178.222.88:7777 -> MY.NET.153.111:2929  UDP
Jul 17 19:29:59  199.178.222.88:7777 -> MY.NET.153.109:1059  UDP
Jul 17 19:29:59  199.178.222.88:7777 -> MY.NET.153.112:1207  UDP
Jul 17 19:29:59  199.178.222.88:7777 -> MY.NET.153.112:1206  UDP
```

Here's another interesting trace. All traffic originates from the same address and same port, but is directed to three different hosts on the same subnet. Ports 7777 and 7778 UDP are used by some internet based online games (Unreal for one). Port 7777 is also used by Napster, a program for sharing MP3 files with other internet users. The source address equates to a hospital in Michigan. What we're seeing here is either three personnel from our network having a bit of fun or, more likely, indications of people sharing files via Napster. Notice that the three addresses are almost sequential - it's entirely possible that three adjoining cubicles are collaborating. Activity using the same ports was noted on July 8th and 9th. Instances of suspected online gaming activity were also noted on UDP ports in the 27000 range.

Other Napster related activity is shown below

```
08/05-18:34:09.147293 [**] Napster 7777 Data[**] 208.184.216.178:7777 -> MY.NET.97.229:49153
08/05-18:34:09.176848 [**] Napster 8888 Data[**] 208.184.216.189:8888 -> MY.NET.201.2:1472
08/05-18:45:04.216858  [**] Napster 8888 Data [**] 208.184.216.189:8888 -> MY.NET.201.2:1472
08/05-18:45:04.220832  [**] Napster 8888 Data [**] 208.184.216.189:8888 -> MY.NET.201.2:1472
08/05-18:42:12.834726  [**] Napster Client Data [**] 128.143.245.137:2298 -> MY.NET.97.230:6699
```

Trace Eighteen:

```
07/26-05:08:56.439380 [**] Watchlist 000220 IL-ISDNNET-990517[**] 212.179.54.69:6699 -> MY.NET.182.94:3661
07/26-05:08:56.722103 [**] Watchlist 000220 IL-ISDNNET-990517[**] 212.179.54.69:6699 -> MY.NET.182.94:3661
07/26-05:08:56.890582 [**] Watchlist 000220 IL-ISDNNET-990517[**] 212.179.54.69:6699 -> MY.NET.182.94:3661
07/26-05:08:57.296104 [**] Watchlist 000220 IL-ISDNNET-990517[**] 212.179.54.69:6699 -> MY.NET.182.94:3661
```

There were almost 19,000 Watchlist items like those above noted in the logfiles. Various source and destination ports were used, including 6688, 4110, 37733, and, once again, the well-known Napster on UDP port 6699.

Trace Nineteen:

```
07/11-19:28:57.652242 [**] Happy 99 Virus[**] 200.223.11.7:4836 -> MY.NET.110.150:25
...
07/19-04:28:40.867369 [**] Happy 99 Virus[**] 203.251.136.2:4985 -> MY.NET.253.42:25
...
07/26-07:50:56.700210 [**] Happy 99 Virus[**] 208.130.42.17:40221 -> MY.NET.6.34:25
```

Viruses are a normal hazard in internet mail. In this trace, Snort noted the Happy 99 virus signatures in e-mail sent to SMTP port 25. The destination addresses probably equate to internal mail servers.

Trace Twenty:

```
Jul 27 10:14:00  210.84.179.196:1054 -> MY.NET.60.8:1  SYN **S*****
Jul 27 10:14:00  210.84.179.196:1055 -> MY.NET.60.8:2  SYN **S*****
Jul 27 10:14:00  210.84.179.196:1056 -> MY.NET.60.8:3  SYN **S*****
...
Jul 27 10:14:03  210.84.179.196:1163 -> MY.NET.60.8:109  SYN **S*****
Jul 27 10:14:03  210.84.179.196:1166 -> MY.NET.60.8:112  SYN **S*****
Jul 27 10:14:03  210.84.179.196:1167 -> MY.NET.60.8:113  SYN **S*****
...
Jul 27 10:14:06  210.84.179.196:1252 -> MY.NET.60.8:198  SYN **S*****
Jul 27 10:14:06  210.84.179.196:1254 -> MY.NET.60.8:200  SYN **S*****
Jul 27 10:14:06  210.84.179.196:1253 -> MY.NET.60.8:199  SYN **S*****
Jul 27 10:15:14  210.84.179.196:15392 -> MY.NET.60.8:113  SYN **S*****
Jul 27 10:15:14  210.84.179.196:15396 -> MY.NET.60.8:113  SYNFIN **SF****
Jul 27 10:15:14  210.84.179.196:15398 -> MY.NET.60.8:113  SYN 21S***** RESERVEDBITS
Jul 27 10:15:14  210.84.179.196:15394 -> MY.NET.60.8:113  FIN ***F****
Jul 27 10:15:14  210.84.179.196:15397 -> MY.NET.60.8:113  VECNA *****P**
```

This Australian attacker sequentially SYN scanned the first 200 ports of a single host. They apparently found one port listening (port 113 authentication service), because they then begin to direct a number of crafted packets at this port. The packets contain a number of illegal flag combinations such as SYN-FINs and Reserved bits. The tool used to perform this attack was likely Nmap.

Trace Twenty-One

```
Jul 28 00:40:01 24.160.99.251:4225      ->    MY.NET.100.236:6346  UNKNOWN 2**F*PAU RESERVEDBITS
Jul 28 03:32:30 199.174.172.121:6399    ->    MY.NET.100.236:6346  NOACK ***FRP*U
Jul 28 03:38:06 199.174.172.121:6375    ->    MY.NET.100.236:53561SPAU **S**PAU
Jul 28 03:40:34 199.174.172.121:6399    ->    MY.NET.100.236:6346  INVALIDACK *1S*RPAU RESERVEDBITS
Jul 28 04:06:15 24.234.91.14:2742       ->    MY.NET.100.236:6346  NOACK **S*R***
Jul 28 04:14:12 24.234.91.14:2742       ->    MY.NET.100.236:6346  INVALIDACK 2*SF**A* RESERVEDBITS
Jul 28 04:55:12 24.234.91.14:2742       ->    MY.NET.100.236:6346  VECNA *****P**
Jul 28 05:18:00 24.234.91.14:2742       ->    MY.NET.100.236:6346  NOACK ****R**U
Jul 28 05:23:00 24.234.91.14:2742       ->    MY.NET.100.236:6346  UNKNOWN *1**R*** RESERVEDBITS
Jul 28 05:23:11 212.55.144.122:1356     ->    MY.NET.100.236:6346  INVALIDACK 2**FRPAU RESERVEDBITS
```

```
Jul 28 05:29:33 195.249.189.10:1035   ->    MY.NET.100.236:6346  INVALIDACK *1S**PA* RESERVEDBITS
Jul 28 06:45:11 212.4.207.26:1649     ->    MY.NET.100.236:6346  XMAS 21*F*P*U RESERVEDBITS
Jul 28 06:45:12 212.4.207.26:1649     ->    MY.NET.100.236:6346  UNKNOWN *1****AU RESERVEDBITS
Jul 28 06:45:17 212.4.207.26:1649     ->    MY.NET.100.236:6346  INVALIDACK **SFR*A*
Jul 28 06:45:33 212.4.207.26:1649     ->    MY.NET.100.236:6346  NULL ********
Jul 28 06:47:02 212.4.207.26:0        ->    MY.NET.100.236:1649  VECNA 21***P** RESERVEDBITS
Jul 28 06:47:25 212.4.207.26:1649     ->    MY.NET.100.236:6346  VECNA ***F*P**
Jul 28 06:58:11 24.234.91.14:1        ->    MY.NET.100.236:2742  INVALIDACK 21**RPAU RESERVEDBITS
Jul 28 07:24:44 24.234.91.14:2742     ->    MY.NET.100.236:6346  INVALIDACK 21**RPAU RESERVEDBITS
Jul 28 08:50:47 212.93.28.236:1200    ->    MY.NET.100.236:6346  NULL ********
Jul 28 09:01:45 212.93.28.236:1200    ->    MY.NET.100.236:6346  NULL ********
Jul 28 09:10:11 24.234.91.14:219      ->    MY.NET.100.236:2742  INVALIDACK 2***R*AU RESERVEDBITS
Jul 28 10:05:54 62.0.139.230:1440     ->    MY.NET.100.236:6346  NULL ********
Jul 28 11:23:27 24.147.11.125:203     ->    MY.NET.100.236:1766  XMAS 21*F*P*U RESERVEDBITS
Jul 28 11:23:40 24.147.11.125:203     ->    MY.NET.100.236:1766  INVALIDACK *1*FRPA* RESERVEDBITS
Jul 28 11:23:51 24.147.11.125:1766    ->    MY.NET.100.236:6346  NOACK **S****U
Jul 28 13:46:39 212.93.28.236:1200    ->    MY.NET.100.236:6346  NULL ********
Jul 28 16:31:26 142.166.205.247:2991  ->    MY.NET.100.236:6346  VECNA *1*F***U RESERVEDBITS
Jul 28 16:35:01 142.166.205.247:2991  ->    MY.NET.100.236:6346  NOACK **SF*P**
Jul 28 21:47:51 24.94.174.75:1749     ->    MY.NET.100.236:6346  INVALIDACK *1SF*PAU RESERVEDBITS
Jul 29 00:04:35 24.164.30.224:10529   ->    MY.NET.100.236:6346  FULLXMAS *1SFRPAU RESERVEDBITS
Jul 29 00:14:25 24.18.166.130:1963    ->    MY.NET.100.236:6346  NOACK 21*FR**U RESERVEDBITS
Jul 29 00:15:34 130.102.196.124:2354  ->    MY.NET.100.236:6346  NOACK *1SF*P** RESERVEDBITS
Jul 29 00:55:55 212.33.42.93:1055     ->    MY.NET.100.236:6346  UNKNOWN 2***RPA* RESERVEDBITS
Jul 29 01:03:31 24.18.166.130:0       ->    MY.NET.100.236:1963  NOACK *1*FRP*U RESERVEDBITS
Jul 29 01:03:43 24.18.166.130:1963    ->    MY.NET.100.236:6346  NOACK ***FR**U
Jul 29 01:10:45 212.33.42.93:1055     ->    MY.NET.100.236:6346  VECNA 2****P*U RESERVEDBITS
Jul 29 01:17:38 24.93.27.123:1302     ->    MY.NET.100.236:6346  UNKNOWN 2***RPA* RESERVEDBITS
Jul 29 03:05:51 212.4.207.26:218      ->    MY.NET.100.236:1462  NMAPID 2*SF*P*U RESERVEDBITS
Jul 29 03:07:17 212.4.207.26:1462     ->    MY.NET.100.236:6346  INVALIDACK ****R*AU
Jul 29 03:21:13 212.4.207.26:155      ->    MY.NET.100.236:1594  INVALIDACK **S*RPA*
Jul 29 03:25:33 212.4.207.26:255      ->    MY.NET.100.236:1594  INVALIDACK ***FRPA*
Jul 29 03:28:02 192.168.0.2:1798      ->    MY.NET.100.236:6346  INVALIDACK **S*RPA*
Jul 29 03:30:14 212.4.207.26:1594     ->    MY.NET.100.236:6346  FULLXMAS 21SFRPAU RESERVEDBITS
Jul 29 03:35:59 212.4.207.26:1594     ->    MY.NET.100.236:6346  XMAS 2**F*P*U RESERVEDBITS
Jul 29 03:36:41 161.184.167.9:1038    ->    MY.NET.100.236:6346  FIN ***F****
Jul 29 03:36:50 24.18.166.130:0             ->    MY.NET.100.236:1963  NOACK 2**FR**U RESERVEDBITS
Jul 29 03:44:14 24.18.166.130:1963    ->    MY.NET.100.236:6346  NULL ********
Jul 29 03:44:15 24.18.166.130:1963    ->    MY.NET.100.236:6346  INVALIDACK 2**FR*A* RESERVEDBITS
Jul 29 03:44:46 24.8.46.216:3673      ->    MY.NET.100.236:6346  SPAU 2*S**PAU RESERVEDBITS
```

Jul 29 03:51:52 212.4.207.26:0	->	MY.NET.100.236:1594	SYN 2*S***** RESERVEDBITS
Jul 29 04:00:05 212.4.207.26:1594	->	MY.NET.100.236:6346	SYN 2*S***** RESERVEDBITS
Jul 29 04:08:34 212.4.207.26:1594	->	MY.NET.100.236:6346	FIN ***F****
Jul 29 04:17:39 24.8.46.216:26	->	MY.NET.100.236:3673	INVALIDACK 2*S*R*A* RESERVEDBITS
Jul 29 04:17:40 24.8.46.216:118	->	MY.NET.100.236:3673	NOACK ****RP**
Jul 29 04:23:17 24.8.46.216:1	->	MY.NET.100.236:3673	UNKNOWN *1****A* RESERVEDBITS
Jul 29 06:13:33 207.171.37.127:62260	->	MY.NET.100.236:6346	UNKNOWN *1**R*A* RESERVEDBITS
Jul 29 06:15:44 207.171.37.127:62264	->	MY.NET.100.236:1068	NMAPID 2*SF*P*U RESERVEDBITS
Jul 29 06:16:38 207.171.37.127:62260	->	MY.NET.100.236:6346	UNKNOWN 2*****A* RESERVEDBITS
Jul 29 06:16:43 207.171.37.127:62260	->	MY.NET.100.236:6346	NMAPID 2*SF*P*U RESERVEDBITS
Jul 29 06:16:47 207.171.37.127:62260	->	MY.NET.100.236:6346	NOACK 2*SFRP** RESERVEDBITS
Jul 29 06:17:56 207.171.37.127:62260	->	MY.NET.100.236:6346	UNKNOWN *1S***A* RESERVEDBITS
Jul 29 06:20:31 207.171.37.127:62271	->	MY.NET.100.236:1068	INVALIDACK 21*FR*A* RESERVEDBITS
Jul 29 06:27:46 62.136.29.13:1418	->	MY.NET.100.236:6346	NULL ********
Jul 29 06:29:02 62.136.29.13:1418	->	MY.NET.100.236:6346	INVALIDACK 2*S*RPA* RESERVEDBITS
Jul 29 06:47:51 210.121.242.164:2929	->	MY.NET.100.236:6346	NULL ********
Jul 29 07:08:46 210.121.242.164:2929	->	MY.NET.100.236:6346	NULL ********
Jul 29 07:18:04 210.121.242.164:2929	->	MY.NET.100.236:6346	NOACK 21*FR**U RESERVEDBITS
Jul 29 07:18:10 210.121.242.164:123	->	MY.NET.100.236:2929	NOACK 21*FRP** RESERVEDBITS
Jul 29 07:29:19 210.121.242.164:2929	->	MY.NET.100.236:6346	INVALIDACK ***FR*AU
Jul 29 07:31:18 210.121.242.164:2929	->	MY.NET.100.236:6346	NOACK 21*FRP** RESERVEDBITS
Jul 29 08:07:10 210.121.242.164:2929	->	MY.NET.100.236:6346	NOACK 21*FRP** RESERVEDBITS
Jul 29 08:20:10 195.162.218.218:2494	->	MY.NET.100.236:6346	INVALIDACK ****RPAU
Jul 29 08:28:17 195.162.218.218:2494	->	MY.NET.100.236:6346	NULL ********
Jul 29 08:31:53 195.162.218.218:2494	->	MY.NET.100.236:6346	NULL ********
Jul 29 08:41:18 210.121.242.164:2929	->	MY.NET.100.236:6346	NOACK **S*R***
Jul 29 08:50:53 195.162.218.218:2494	->	MY.NET.100.236:6346	NULL ********
Jul 29 08:51:43 210.121.242.164:2929	->	MY.NET.100.236:6346	INVALIDACK *1*FR*AU RESERVEDBITS
Jul 29 08:52:32 210.121.242.164:2929	->	MY.NET.100.236:6346	NULL ********
Jul 29 09:08:38 210.121.242.164:2929	->	MY.NET.100.236:6346	UNKNOWN 21**R*A* RESERVEDBITS
Jul 29 09:13:30 210.121.242.164:2929	->	MY.NET.100.236:6346	UNKNOWN *1*F*PAU RESERVEDBITS
Jul 29 09:24:07 210.121.242.164:2929	->	MY.NET.100.236:6346	FIN *1*F**** RESERVEDBITS
Jul 29 09:25:42 24.132.25.229:1741	->	MY.NET.100.236:6346	INVALIDACK 2**FRPAU RESERVEDBITS
Jul 29 09:40:34 213.224.84.2:1078	->	MY.NET.100.236:6346	NMAPID 2*SF*P*U RESERVEDBITS
Jul 29 09:54:31 210.121.242.164:2929	->	MY.NET.100.236:6346	INVALIDACK 2*SF*PAU RESERVEDBITS
Jul 29 09:56:19 213.224.84.2:1078	->	MY.NET.100.236:6346	NOACK 21S*R*** RESERVEDBITS
Jul 29 09:56:37 213.224.84.2:1078	->	MY.NET.100.236:6346	NMAPID 2*SF*P*U RESERVEDBITS
Jul 29 10:11:03 210.121.242.164:0	->	MY.NET.100.236:2929	UNKNOWN 21*F**AU RESERVEDBITS
Jul 29 10:11:28 210.121.242.164:2929	->	MY.NET.100.236:6346	NULL ********
Jul 29 10:12:10 210.121.242.164:2929	->	MY.NET.100.236:6346	NULL ********

```
Jul 29 10:14:15 210.121.242.164:2929  ->      MY.NET.100.236:6346  NULL ********
Jul 29 10:21:54 210.121.242.164:2929  ->      MY.NET.100.236:6346  UNKNOWN 21**R*A* RESERVEDBITS
Jul 29 10:22:05 210.121.242.164:123   ->      MY.NET.100.236:2929  UNKNOWN 21*F**AU RESERVEDBITS
Jul 29 10:22:41 213.224.84.2:1078     ->      MY.NET.100.236:6346  NOACK **S**P**
Jul 29 10:34:15 210.121.242.164:2929  ->      MY.NET.100.236:6346  NOACK 21*FRP*U RESERVEDBITS
Jul 29 10:38:03 210.121.242.164:2929  ->      MY.NET.100.236:6346  NOACK 21*FRP** RESERVEDBITS
Jul 29 10:44:50 210.121.242.164:161   ->      MY.NET.100.236:2929  NOACK 21*FRP** RESERVEDBITS
Jul 29 10:45:40 210.121.242.164:2929  ->      MY.NET.100.236:6346  NOACK 21*FRP** RESERVEDBITS
Jul 29 11:11:02 210.121.242.164:123   ->      MY.NET.100.236:2929  NOACK 21*FRP** RESERVEDBITS
Jul 29 11:12:22 210.121.242.164:2929  ->      MY.NET.100.236:6346  VECNA ***F*P**
Jul 29 11:45:21 210.121.242.164:2929  ->      MY.NET.100.236:6346  XMAS 21*F*P*U RESERVEDBITS
Jul 29 11:46:28 210.121.242.164:2929  ->      MY.NET.100.236:6346  XMAS 2**F*P*U RESERVEDBITS
Jul 29 11:47:24 210.121.242.164:24    ->      MY.NET.100.236:2929  NOACK 21*FR*** RESERVEDBITS
Jul 29 11:51:58 210.121.242.164:2929  ->      MY.NET.100.236:6346  FIN *1*F**** RESERVEDBITS
Jul 29 11:52:20 210.121.242.164:2929  ->      MY.NET.100.236:6346  NOACK 2**FRP*U RESERVEDBITS
Jul 29 11:57:33 210.121.242.164:2929  ->      MY.NET.100.236:6346  FIN *1*F**** RESERVEDBITS
Jul 29 11:57:53 210.121.242.164:2929  ->      MY.NET.100.236:6346  INVALIDACK 2*S*R*A* RESERVEDBITS
Jul 29 12:02:49 210.121.242.164:2929  ->      MY.NET.100.236:6346  INVALIDACK ***FR*AU
Jul 29 12:10:55 210.121.242.164:2929  ->      MY.NET.100.236:6346  UNKNOWN 21*F*PAU RESERVEDBITS
Jul 29 12:25:07 210.121.242.164:2929  ->      MY.NET.100.236:6346  UNKNOWN 2**F**A* RESERVEDBITS
Jul 29 13:28:24 210.121.242.164:2929  ->      MY.NET.100.236:6346  NOACK 21*FRP** RESERVEDBITS
Jul 29 13:31:27 210.121.242.164:0     ->      MY.NET.100.236:2929  FIN *1*F**** RESERVEDBITS
Jul 29 13:48:55 210.121.242.164:2929  ->      MY.NET.100.236:6346  NOACK 21*FRP** RESERVEDBITS
```

This is probably the most interesting trace of the whole month. The packets are received over the space of two days from almost 30 different hosts all over the world, but are all directed at a single host and primarily the same port (TCP 6346). Almost all packets go against the TCP/IP standard by having reserved bits set or combinations of flags that would not normally be seen in TCP/IP traffic (such as FIN+RESET+PUSH). There are an exceptional number of variations here, although almost every combination can be seen coming from host 210.121.242.164 on various source ports, including 0. It is possible that many of the addresses seen here are spoofed as decoys while one or more real source addresses gather information. This is almost certainly an attempt to fingerprint the operating system of the target computer. The likely source of this scan is the Nmap Tool.

Trace Twenty-Two:

```
06/27-07:39:28.385752 [**] NMAP TCP ping![**] 209.218.228.46:80 -> MY.NET.1.8:53
06/27-07:39:28.388448 [**] NMAP TCP ping![**] 209.218.228.46:53 -> MY.NET.1.8:53
06/27-07:39:33.390475 [**] NMAP TCP ping![**] 209.218.228.46:80 -> MY.NET.1.8:53
06/27-07:39:33.390629 [**] NMAP TCP ping![**] 209.218.228.46:53 -> MY.NET.1.8:53
06/27-07:51:18.494768 [**] NMAP TCP ping![**] 195.54.105.6:80 -> MY.NET.1.9:53
06/27-07:51:18.494815 [**] NMAP TCP ping![**] 195.54.105.6:53 -> MY.NET.1.9:53
06/27-07:51:23.472464 [**] NMAP TCP ping![**] 195.54.105.6:80 -> MY.NET.1.9:53
06/27-07:51:23.472859 [**] NMAP TCP ping![**] 195.54.105.6:53 -> MY.NET.1.9:53
```

NMap can also be used in ping mode to map a network. Above are several Snort Alerts to an observed NMap ping from TCP ports 80 to 53 and 53 to 53. These ports are used for source and destination because they are often allowed through firewalls and thus can penetrate a network's outer perimeter to map the inside.

Trace Twenty-Three:

```
Jul 29 11:58:06  211.38.95.138:1132 -> MY.NET.1.0:21  SYN **S*****
Jul 29 11:58:06  211.38.95.138:1133 -> MY.NET.1.1:21  SYN **S*****
Jul 29 11:58:06  211.38.95.138:1134 -> MY.NET.1.2:21  SYN **S*****
Jul 29 11:58:06  211.38.95.138:1135 -> MY.NET.1.3:21  SYN **S*****
...
Jul 29 11:58:09  211.38.95.138:1602 -> MY.NET.2.215:21  SYN **S*****
Jul 29 11:58:10  211.38.95.138:1897 -> MY.NET.4.0:21  SYN **S*****
Jul 29 11:58:10  211.38.95.138:1898 -> MY.NET.4.1:21  SYN **S*****
...
Jul 29 12:01:59  211.38.95.138:4977 -> MY.NET.217.254:21  SYN **S*****
Jul 29 12:01:59  211.38.95.138:4978 -> MY.NET.218.0:21  SYN **S*****
Jul 29 12:01:59  211.38.95.138:4979 -> MY.NET.218.1:21  SYN **S*****
...
Jul 29 12:02:40  211.38.95.138:2229 -> MY.NET.253.238:21  SYN **S*****
Jul 29 12:02:40  211.38.95.138:2230 -> MY.NET.253.239:21  SYN **S*****
Jul 29 12:02:40  211.38.95.138:2231 -> MY.NET.253.240:21  SYN **S*****
```

This scan is unusual in that, on some subnets it scans, it starts at the .0 host rather than the .1 host, which was the case in all other scans. The .0 address on a subnet is interpreted by some hosts as a broadcast, and may elicit a responses from all hosts on the subnet.

Trace Twenty-Five:

```
Jul 29 18:32:47  194.165.162.132:4289 -> MY.NET.98.185:31336  UDP
Jul 29 18:32:48  194.165.162.132:1057 -> MY.NET.98.185:30029  SYN **S*****
Jul 29 18:32:49  194.165.162.132:1058 -> MY.NET.98.185:666  SYN **S*****
Jul 29 18:32:47  194.165.162.132:1060 -> MY.NET.98.185:1999  SYN **S*****
...
Jul 29 18:42:11  194.165.162.132:4433 -> MY.NET.98.185:23456  SYN **S*****
Jul 29 18:42:10  194.165.162.132:4427 -> MY.NET.98.185:34324  SYN **S*****
Jul 29 18:42:11  194.165.162.132:4434 -> MY.NET.98.185:5742  SYN **S*****
Jul 29 18:42:11  194.165.162.132:4435 -> MY.NET.98.185:2583  SYN **S*****
```

Here's a SYN scan looking for almost every imaginable trojan. Notice the solitary UDP packet mixed in with the others. Although most of the ports are known, some are not. The scan consists of 312 packets directed at 75 different ports, only one of which is UDP. A similar but shorter scan was noted originating from the same subnet, and directed at a different host, at 20:43 the same day. Trojan scans were common during the month.

Trace Twenty-Six:

```
Jul 29 21:17:07  24.3.39.44:7001 -> MY.NET.6.42:7000  UDP
Jul 29 21:17:08  24.3.39.44:7001 -> MY.NET.70.142:7000  UDP
Jul 29 21:17:08  24.3.39.44:7001 -> MY.NET.60.43:7000  UDP
Jul 29 21:17:08  24.3.39.44:7001 -> MY.NET.6.45:7000  UDP
Jul 29 21:17:08  24.3.39.44:7001 -> MY.NET.6.33:7003  UDP
Jul 29 21:17:08  24.3.39.44:7001 -> MY.NET.6.48:7000  UDP
Jul 29 21:17:08  24.3.39.44:7001 -> MY.NET.60.12:7003  UDP
Jul 29 21:17:08  24.3.39.44:7001 -> MY.NET.1.13:7003  UDP
```

From July 29th to August 10th there is quite a bit of activity from host 24.3.39.44 on udp port 7001 to eight hosts on the internal networks. It appears to be legitimate traffic (probably NFS related) however it does bear checking into.

Richard D. Salisko

Trace Twenty-Seven:

```
08/04-04:23:07.716948 [**] Attempted Sun RPC high port access[**] 205.188.153.111:4000 -> MY.NET.217.126:32771
08/04-04:24:07.603934 [**] Attempted Sun RPC high port access[**] 205.188.153.111:4000 -> MY.NET.217.126:32771
08/04-04:25:07.506764 [**] Attempted Sun RPC high port access[**] 205.188.153.111:4000 -> MY.NET.217.126:32771
08/04-04:26:07.375120 [**] Attempted Sun RPC high port access[**] 205.188.153.111:4000 -> MY.NET.217.126:32771
08/04-04:27:07.291395 [**] Attempted Sun RPC high port access[**] 205.188.153.111:4000 -> MY.NET.217.126:32771
```

Numerous attempts to connect to port 32771 were noted during the month, including this attempt which consisted of over 2200 individual packets, all from the same host. Because of the source port - 4000, and the identity of the 205.188.153.111 server - an AOL host, it's more probable that this is a ICQ server since they often send to high numbered ports.

Trace Twenty-Nine:

```
08/04-16:23:34.611008 [**] SMB Name Wildcard[**] 132.201.232.167:137 -> MY.NET.15.127:137
08/04-16:23:34.611091 [**] SMB Name Wildcard[**] 208.16.237.10:137 -> MY.NET.15.127:137
08/04-16:23:42.770412 [**] SMB Name Wildcard[**] 208.16.237.10:137 -> MY.NET.15.127:137
08/04-16:23:42.770527 [**] SMB Name Wildcard[**] 132.201.232.167:137 -> MY.NET.15.127:137
08/04-16:23:44.293564 [**] SMB Name Wildcard[**] 132.201.232.167:137 -> MY.NET.15.127:137
08/04-16:23:44.293784 [**] SMB Name Wildcard[**] 208.16.237.10:137 -> MY.NET.15.127:137
08/04-16:23:50.888340 [**] SMB Name Wildcard[**] 132.201.232.167:137 -> MY.NET.15.127:137
08/04-16:23:50.888445 [**] SMB Name Wildcard[**] 208.16.237.10:137 -> MY.NET.15.127:137
```

This trace shows two external hosts attempting to connect to a single internal host using the SMB (Server Message Block). This protocol is primarily used by Windows based hosts to locate other hosts. The UNIX Samba software also uses this protocol to communicate with Windows hosts. The fact that both the source and destination ports in this trace are port 137 (UDP) indicates that the source hosts are probably Windows systems.

Richard D. Salisko

Trace Thirty:

```
08/05-18:34:09.759975 [**] PING-ICMP Destination Unreachable[**] 216.68.192.50 -> MY.NET.70.121
08/05-18:34:09.786295 [**] PING-ICMP Destination Unreachable[**] 24.4.52.197 -> MY.NET.70.121
08/05-18:34:09.786345 [**] PING-ICMP Destination Unreachable[**] 24.4.52.197 -> MY.NET.70.121
08/05-18:34:09.792601 [**] PING-ICMP Destination Unreachable[**] 212.204.188.51 -> MY.NET.70.121
08/05-18:34:09.985369 [**] PING-ICMP Destination Unreachable[**] 216.68.192.50 -> MY.NET.70.121
08/05-18:34:10.274398 [**] PING-ICMP Destination Unreachable[**] 216.68.192.50 -> MY.NET.70.121
```

During the month's collection there were over 12,000 Destination unreachable messages recorded by Snort. Most of these (11,300) occurred between 18:30 and 18:42 on 05 August. They originated from a small number of hosts and were all directed at MY.NET.70.121. No evidence of a DoS attack originating from this host is apparent, so I can only assume that this address was being spoofed by someone outside launching a Denial of Service attack or very fast scan against the networks sending the above messages. The messages we are seeing are ICMP messages, probably from a router, to the source IP address saying that the destination cannot be reached.

Trace Thirty-One:

```
Aug  5 13:36:38  209.138.185.157:2606 -> MY.NET.253.114:8892  SYN **S*****
Aug  5 13:36:38  209.138.185.157:2608 -> MY.NET.253.114:1514  SYN **S*****
Aug  5 13:36:38  209.138.185.157:2607 -> MY.NET.253.114:340  SYN **S*****
...
Aug  5 13:36:54  209.138.185.157:3150 -> MY.NET.253.114:841  SYN **S*****
Aug  5 13:36:54  209.138.185.157:1666 -> MY.NET.253.114:53  UDP
Aug  5 13:36:54  209.138.185.157:3152 -> MY.NET.253.114:81  SYN **S*****
...
Aug  5 13:37:05  209.138.185.157:3552 -> MY.NET.253.114:8080  SYN **S*****
...
Aug  5 13:37:21  209.138.185.157:1666 -> MY.NET.253.114:53  UDP
...
Aug  5 13:37:25  209.138.185.157:4290 -> MY.NET.253.114:12345  SYN **S*****
...
Aug  5 13:37:26  209.138.185.157:4291 -> MY.NET.253.114:180  SYN **S*****
Aug  5 13:37:28  209.138.185.157:48682 -> MY.NET.253.114:22  SYN 2*S***** RESERVEDBITS
Aug  5 13:37:28  209.138.185.157:48675 -> MY.NET.253.114:1  UDP
```

Richard D. Salisko

This trace is part of a random scan for listening TCP and UDP ports on a single host. The scan consisted of 683 packets. Many of the TCP ports below 1024 were scanned along with several well known higher ports like 8080, and even a few trojan ports for good measure. The only UDP port scanned was 53(DNS). At the very end a SYN packet with reserved bits set and a single UDP packet to port 1 was sent.

Trace Thirty-Two:

```
08/05-18:57:30.891476 [**] PING-ICMP Source Quench[**] 209.245.5.158 -> MY.NET.70.121
```

This is an ICMP Source Quench packet alert. The ICMP source quench is used by one host to tell another to slow down! This is quite normal in network traffic and likely a false positive.

Trace Thirty-Three:

```
Aug  8 14:26:22  207.115.68.121:32769 -> MY.NET.140.51:33488  UDP
Aug  8 14:26:22  207.115.68.121:32769 -> MY.NET.140.51:33489  UDP
Aug  8 14:26:23  207.115.68.121:32769 -> MY.NET.140.51:33490  UDP
Aug  8 14:26:23  207.115.68.121:32769 -> MY.NET.140.51:33491  UDP
Aug  8 14:26:23  207.115.68.121:32769 -> MY.NET.140.51:33492  UDP
Aug  8 14:26:24  207.115.68.121:32769 -> MY.NET.140.51:33493  UDP
Aug  8 14:26:24  207.115.68.121:32769 -> MY.NET.140.51:33494  UDP
Aug  8 14:26:25  207.115.68.121:32769 -> MY.NET.140.51:33495  UDP
Aug  8 14:26:26  207.115.68.121:32769 -> MY.NET.140.51:33496  UDP
Aug  8 14:26:26  207.115.68.121:32769 -> MY.NET.140.51:33497  UDP
Aug  8 14:26:27  207.115.68.121:32769 -> MY.NET.140.51:33498  UDP
Aug  8 14:26:29  207.115.68.121:50000 -> MY.NET.140.51:50000  UDP
```

This trace is likely a traceroute directed at host MY.NET.140.51 since the UDP destination ports are all in the 33484+ range, which is the signature for traceroute.

Richard D. Salisko

Trace Thirty-Four:

```
Aug 10 06:15:03  MY.NET.5.37:2600 -> MY.NET.5.3:5632  UDP
Aug 10 06:15:03  MY.NET.5.37:2600 -> MY.NET.5.3:22  UDP
Aug 10 06:15:04  MY.NET.5.37:2600 -> MY.NET.5.13:5632  UDP
Aug 10 06:15:04  MY.NET.5.37:2600 -> MY.NET.5.13:22  UDP
Aug 10 06:15:04  MY.NET.5.37:2600 -> MY.NET.5.15:5632  UDP
Aug 10 06:15:04  MY.NET.5.37:2600 -> MY.NET.5.25:5632  UDP
Aug 10 06:15:04  MY.NET.5.37:2600 -> MY.NET.5.29:5632  UDP
Aug 10 06:15:04  MY.NET.5.37:2600 -> MY.NET.5.34:5632  UDP
Aug 10 06:15:05  MY.NET.5.37:2600 -> MY.NET.5.40:5632  UDP
Aug 10 06:15:05  MY.NET.5.37:2600 -> MY.NET.5.41:5632  UDP
```

Beginning August 10th, traffic is apparent from one internal host to others using destination ports of UDP ports 22 and 5632. These ports are related to PC Anywhere, a commercial program for remotely controlling other computers. This could be perfectly legitimate traffic from say, a help desk to a supported computer, on the other hand, it could be internal users who have found a new toy, or even traffic from a compromised host searching for exploitable systems.

Richard D. Salisko

Recommendations

First and foremost, a sound firewall architecture should be implemented to shield the internal network from external hosts although it is remotely possible that a firewall with extremely liberal policies is already in place. The rules should be set up to permit only traffic required to support the organization's functions. All incoming traffic should be proxied at the application level if possible.

Once the firewall is in place, the internal network should be extensively scanned for listening host services, particularly on trojan ports, and any suspicious findings noted and investigated. Policy should be implemented to ban or control all remote control and remote access programs, as well as ' network utilities ' like NMAP.

All hosts should have up-to-date anti-virus software installed and active. SMTP mail, along with HTTP and FTP traffic into and out of the network, should be scanned for virus signatures in realtime.

Internal and external IDS sensors should be setup, along with a good TCP/IP logging facility like TCPDump, to record network activity and provide data for offline analysis if required.

All internal hosts should be protected with TCP wrappers and similar products to help protect against future scans and vulnerabilities, especially those hosts which use a portmapper service. All externally visible hosts, including proxy servers such as WinGate, should be hardened and all Operating System and Application security patches applied.

Richard D. Salisko

Assignment Four – Analysis Process

Assignment three consisted of over 350,000 lines of text files to be analyzed. At first glance it is very easy to be overwhelmed with the shear size of the task at hand. However, sanity prevailed, and, using a few simple tools, I found the data could be comfortably manipulated. Although UNIX experts would probably use tools such as shell scripts and Perl to parse and strip irrelevant information, I work in a primarily Windows environment, and had to make do with tools from that environment.

The first task was to get the information into a machine-readable format. Most of the information is of variable length and cannot be easily parsed, at least not without the use of a script language such as Perl. Lacking that knowledge and the time to adequately obtain it, I opted for a word processor.

I developed macros in Microsoft Word which would take an input file and insert parsing characters (commas) in the appropriate places for importing the data into a spreadsheet format. After saving the parsed file in a straight text format, I then attempted to import the data into an Microsoft Excel spreadsheet, only to discover that there is, at least in my version, a limit of 65,000 records. I needed more.

I decided a database would be much more efficient for the task at hand. Microsoft Access proved more than equal. I imported the data from both the ‘ S ‘ and ‘ A ‘ series of text files into separate tables. This way, the data could be related by date and time as required. A simple change in date format allowed me to convert both dates to compatible values.

Once the data could be manipulated by machine, it was a matter of grouping similar scans and information in one file, documenting a representative data set, finding any correlating information in the other file, and building the report. I mostly used the information in the parsed text files– via Notepad - for pasting into the report because I ended up with unblemished data. When I wanted to re-arrange the data, as in the case of Trace Twelve where I needed to demonstrate a pattern, I simply cut and pasted the data directly from the database.

By systematically going through the files, analyzing the patterns, and then removing similar patterns from the data, I was able to work with a progressively smaller data set. When I deleted that final line from the table, the world was lifted from my shoulders…..