Global Information Assurance Certification Paper

Submission of Practical assignment using SANS Parliament Hill 2000 format
by Pamela Pirkle
10/14/00

DETECT #1 All addresses have been sanitized

1) 17:18:27.120679 212.11.222.33.1195 > 123.38.123.123.**80**: S 4216451:4216451(0) win 8192 (DF)
2) 17:18:29.522093 212.11.222.33.1195 > 123.38.123.123.**80**: S 4216451:4216451(0) win 8192 (DF)
3) 17:18:36.071192 212.11.222.33.1195 > 123.38.123.123.**80**: S 4216451:4216451(0) win 8192 (DF)
4) 17:18:48.305689 212.11.222.33.1195 > 123.38.123.123.**80**: S 4216451:4216**451**(0) win 8192 (DF)
5) 17:19:12.350985 212.11.222.33.1276 > 123.38.123.123.**8080**: S 4261**790**:4261790(0) win 8192 (DF)
6) 17:19:15.296817 212.11.222.33.1276 > 123.38.123.123.**8080**: S 4261790:4261790(0) win 8192 (DF)
7) 17:19:21.509566 212.11.222.33.1276 > 123.38.123.123.**8080**: S 4261790:4261790(0) win 8192 (DF)
8) 17:19:33.577708 212.11.222.33.1276 > 123.38.123.123.**8080**: S 4261790:4**261790**(0) win 8192 (DF)
9) 17:19:58.077006 212.11.222.33.1319 > 123.38.123.123.**3128**: S 4**307013**:4307013(0) win 8192 (DF)
10) 17:20:01.065642 212.11.222.33.1319 > 123.38.123.123.**3128**: S 4307013:4307013(0) win 8192 (DF)
11) 17:20:07.152031 212.11.222.33.1319 > 123.38.123.123.**3128**: S 4307013:4307013(0) win 8192 (DF)
12) 17:20:19.562780 212.11.222.33.1319 > 123.38.123.123.**3128**: S 4307013:4307013(0) win 8192 (DF)

1. **Source of trace**: Trace pulled from incident report from a military command shadow

17:18:27.120679	212.11.222.33 1195	>	123.38.123.123.**80**	:	S	4216451:4216451(0)	win 8192	(DF)
A	B C		D	E	F	G	H	I J

   A. Timestamp of detect
   B. Source IP
   C. Source Port
   D. Target IP
   E. Target Port
   F. Flag set: SYN
   G. Sequence number
   H. Data
   I. Window size
   J. Do not Fragment

2. **Detect was generated by**: Shadow detection software in TCPdump format.

3. **Probability the source address was spoofed**: At first glance, I suspected the IP could be spoofed because the unchanging sequence number for each port being scanned might be indicative of a crafted packet. Why craft a packet and not spoof the IP? But then I realized that the sequence number would only change with each new attempted connection. A retry would not generate a new sequence number. (book II page 155 SAN Institute, GIAC Certified Intrusion Analyst) There is no evidence that the three way handshake has been acknowledged: Only the SYN flag is set. When a *new* connection is being attempted the source port and sequence numbers change as seen on trace lines five and nine. If this trace had ttl information or the packet ID it would be easier to verify. A tcpshow could give this information. Also, since the perpetrating IP is seeking information, it would need to receive a response. Therefore, it is not likely this IP is spoofed.

4. **Description of Attack:** Scan for TCP ports 80, 8080, 3128 is a signature for the trojan RingZero or it could be a simple scan for web, proxy and squid proxy servers. The source IP 212.11.222.33* (whois info indicates a foreign source) sends a packet with the SYN flag set, waiting for the SYN ACK in the three way handshake. The targeted system (a workstation) does not acknowledge the packet. Since there is not an acknowledgement, there is a retry. The time stamp shows the rate at which the probe is occurring. This appears to follow the time delta described in book II page 155 SAN Institute, GIAC Certified Intrusion Analyst. Note that twenty four seconds pass as the new destination port is targeted.(time stamp data from line four, five, eight and nine) A total of three minutes elapse during this attack. (again, ttl information would be great) Also of note is the big jump in sequence numbers between each new connection attempt. The perpetrator could be interleaving addresses scanning several activities just looking for an opportunity to run an exploit. This attack is most likely a scripted attack The targeted IP is a workstation.(nslookup information on IP)

5. **Attack Mechanism**: The Trojan RingZero runs as a hidden process on the target system. It sends and retrieves data (ip addresses) over an Internet connection to a central server. There are three versions of this trojan horse: ITS.EXE, PST.EXE andTELNET23.EXE. The attacker is probing for the availability of these ports. A system infected with this Trojan would initiate a random scan for ports 80/8080/ 3128 looking for a match to the IP address(or home web address) imbedded in the script. The traffic of interest to the analyst in this case would be outbound . This should all occur within one minute of the initial packet being sent. As noted above, the time span is well above that timeframe. TCP ports 80 (common port for world wide web), 8080 (common location for proxy), and 3128 (squid proxy) are all common exploitable ports. The attacker would attempt to connect to the port and retrieve the data it desires. An attacker scanning for these ports is likely searching for a proxy server they can use to surf the Internet anonymously. Another cause of scans at this port, for a similar reason, is when users enter chatrooms. Other users or the servers themselves will attempt to check this port to see if the user's machines supports proxying.

6. **Correlations**: Wason Han wrote the write up for this detect for the Symantec web page URL: http://www.symantec.com/avcenter/venc/data/ringzero.trojan.html

   CVE-1999-0158 Description Cisco PIX firewall manager (PFM) on Windows NT allows attackers to connect to port 8080 on the PFM server and retrieve any file whose name and location is known.
   http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0158
   http://www.sans.org/newlook/resources/ringzero.htm
   "Hunt for RingZero"

7. **Evidence of Active targeting**: This detect is targeting one specific IP for the following locations: world wide web, proxy, and 3128.

8. **Severity**: (Critical + Lethal) - (System Countermeasures + Network Countermeasure) = Severity
   4 would be assigned for Criticality. Though this is just a scan, the web servers and proxy servers are the target for this scan.
   1 is assigned for lethality. Do not believe this to be a RingZero trojan, though the information gained from this scan could lead to more exploits, access is not very like gained from this scan.

   There are no signs that this has been a successful scan. The system does not send a response. 5 and 4 will be assigned respectively

   (4+1)-(5+4)= Severity. 5-9= *4*

9. **Defensive Recommendation**: Defenses are fine. The scan appears to be blocked at the firewall. If this were RingZero, the signature for this Trojan could be down loaded from either the Symantec web page or Norton's Antivirus software. Once detected, the files should be quarantined and then deleted. If the targeted command does not utilize proxy servers, the activity could be blocked at the firewall using an expanded ACL.

10. **Multiple choice test question**:

    1. What information would help the analyst confirm this detect did indeed contain a crafted packet?
       A. IP ID number increments with each SYN sent.
       B. IP ID number remains the same with each retry.
       C. IP ID number changes with each reconnect.
       D. IP ID number remains the same with each reconnect.

    The answer is: D

Detect # 2 ALL Addresses have been sanitized

Attempted socks.

09:47:36.251345 mybrainhurts.org.54613 > ABC.DEFG.HIJK.LMN.domain: 60014 (**44**)
09:47:36.257799 ABC.DEFG.HIJK.LMN.domain > mybrainhurts.org.54613: 60014 NXDomain* 0/1/0 (123) (DF)
09:47:36.673606 mybrainhurts.org.56843 > 123.45.123.64.**1080**: S 2795607899:2795607899(0) win 8192 (DF)
09:47:36.673900 123.45.123.64.**1080** >mybrainhurts.org.56843: S 3975082800:3975082800(0) ack 2795607900 win 10136 (DF)
09:47:36.751114 mybrainhurts.org.56843 > 123.45.123.64.**1080**: . ack 3975082801 win 8760 (DF)
09:47:36.754319 123.45.123.64.**1080** >mybrainhurts.org.56843: F 3975082801:3975082801(0) ack 2795607900 win 10136 (DF)
09:47:36.836949 mybrainhurts.org.56843 > 123.45.123.64.**1080**: . ack 3975082802 win 8760 (DF)
09:47:37.667455 mybrainhurts.org.56843 > 123.45.123.64.**1080**: F 2795607900:2795607900(0) ack 3975082802 win 8760 (DF)

09:47:37.683062 123.45.123.64.**1080** > mybrainhurts.org.56843: . ack 2795607901 win 10136 (DF)
09:50:41.065697 mybrainhurts.org.56865 > 123.45.123.64.**1080**: S 2846093463:2846093463(0) win 8192 (DF)
09:50:41.066035 123.45.123.64.**1080** > mybrainhurts.org.56865: S 242928986:242928986(0) ack 2846093464 win 10136 (DF)
09:50:41.143996 mybrainhurts.org.56865 > 123.45.123.64.**1080**: . ack 242928987 win 8760 (DF)
09:50:41.147840 123.45.123.64.**1080** > mybrainhurts.org.56865: F 242928987:242928987(0) ack 2846093464 win 10136 (DF)
09:50:41.228736 mybrainhurts.org.56865 > 123.45.123.64.**1080**: . ack 242928988 win 8760 (DF)
09:50:42.062178 mybrainhurts.org.56865 > 123.45.123.64.**1080**: F 2846093464:2846093464(0) ack 242928988 win 8760 (DF)
09:50:42.063646 123.45.123.64.**1080** > mybrainhurts.org.56865: . ack 2846093465 win 10136 (DF)

1. **Source of Trace**: Detect pulled from incident report generated from a navy command

09:50:42.063646 123.45.123.64. **1080** > mybrainhurts.org.56865: . ack 2846093465 win
A B C D E F G H
10136 (DF)
H I

A. Timestamp of trace
B. Target IP
C. Target Port
D. Source domain name
E. Source Port
F. Set Flag (acknowledgment)
G. Sequence number
H. Window size
I. Do not fragment

2. **Detect was generated by**: Shadow detection system which formats in a tcpdump
3. **Probability detect was spoofed**: The source IP is probably not spoofed because the three way hand shake has been completed. The end user will see the results.
4. **Description of Attack**: 09:47:36.251345 source mybrainhurts.org initiates a link with the domain server ABC.DEFG.HIJK.LMN.domain with 64 bytes data in the header. ABC.DEFG.HIJK.LMN.domain replies back with 60014 NXDomain* 0/1/0 (123) (DF). The source then sends a packet with a SYN flag set. Target command responds with a SYN/ACK, Source sends an ACK, the Target command responds with a FIN then the source command replies back with a FIN/ACK and finally the Target com mand sends an ACK. The three way hand shake has been completed. Note there was not a PUSH flag sent for data. This is a SOCKS exploit.
5. **Attack mechanism**: Port 1080 is used by the SOCKS networking proxy protocol. It is designed to allow a host outside of a firewall to connect transparently and securely through the firewall. As a consequence, some sites may have port 1080 opened for incoming connections to a system running a socks daemon. One of the more common uses of SOCKS seems to be to allow ICQ traffic to hosts that are behind a firewall.

Also, if a system is found to be running WinGate, remote attackers can perform a denial of service in machines using a buffer overflow in the Winsock Redirector Service. This protocol tunnels traffic through firewalls, allowing many people behind the firewall access to the Internet through a single IP address. In theory, it should only tunnel inside traffic out towards the Internet. However, it is frequently misconfigured and allows hackers/crackers to tunnel their attacks inwards, or simply bounce through the system to other Internet machines, masking their attacks as if they were coming from within.

6. **Correlations**: http://www.sans.org/y2k/socks.htm, SANS Institute GIAC Certified Intrusion Analyst Read Ahead Information Page 12
   http://www.securityfocus.com/bid/509.html
   http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0441
   http://www.simovits.com/nyheter9902.html
7. **Evidence of active targeting**: The specific IP has been targeted. The handshake is completed.
8. **Severity**: (Critical + Lethal) - (S + N Countermeasure) = Severity
   (4 + 4) - (2+1) = **8-3 = *5***
   Not able to confirm what box was attacked. The first line is evidence that this could be the Domain Server. The fact that the handshake was completed is not good. The attacker has gained valuable information. This activity is apparently permitted by the firewall.
9. **Defensive recommendation**: If system is not being utilized as a proxy server, configure a firewall with an extended ACL that reads
   access-list 112 deny tcp any any eq 1080
   access-list 112 deny udp any any eq 1080
10 **Multiple choice test question**:
   1. What common port is associated with the SOCKS exploit?
      A. port 80
      B. port 143
      C. port 111
      D. port 1080

The answer is D port 1080

DETECT # 3

00:42:05.153286 146.xyz.xyz.z > SKINNI.IWANNABE.WIL: icmp: echo request
00:42:05.153775 146.xyz.xyz.z > SKINNI.IWANNABE.WIL: icmp: echo request
00:42:05.154606 146.xyz.xyz.z > SKINNI.IWANNABE.WIL: icmp: echo request
00:42:05.156941 2XX.XX.XXX.33 > 146.xyz.xyz.z: icmp: host SKINNI.IWANNABE.WILunreachable - admin prohibited filter
00:42:09.693603 146.xyz.xyz.z.2300 > SKINNI.IWANNABE.WIL.domain: S 471142303:471142367(**64**) win **2048**
00:42:09.694177 146.xyz.xyz.z.2301 > SKINNI.IWANNEBE.WILdomain: S 1148491736:1148491800(**64**) win **2048**
00:42:09.694735 146.xyz.xyz.z 2302 > SKINNI.IWANNABE.WIL.domain: S 1921068335:1921068399(**64**) win **2048**

00:42:09.697383 2XX.XX.XXX.33 > 146.xyz.xyz.z: icmp: host SKINNI.IWANNABE.WIL unreachable - admin prohibited filter
01:19:16.250767 146.xyz.xyz.z > SKINNI.IWANNABE.WIL: icmp: echo request
01:19:16.251280 146.xyz.xyz.z > SKINNI.IWANNABE.WIL: icmp: echo request
01:19:16.251781 146.xyz.xyz.z > SKINNI.IWANNABE.WIL: icmp: echo request
01:19:16.254378 2XX.XX.XXX.33 > 146.xyz.xyz.z: icmp: host SKINNI.IWANNABE.WIL unreachable - admin prohibited filter

1. **Source of Trace**: Detect pulled from incident report generated from a navy command.

00:42:05.153286 146.xyz.xyz.z > SKINNI.IWANNABE.WIL: icmp: echo request
A. B. C. D. E.

A. Date time stamp. Great source of information used to decipher whether attack is scripted or if addresses might be interleaved.

B. Source Address: The IP where detect originated

C. Destination: This is the actual name of the system/domain being scanned.
D. Protocol Field: Protocol being used is ICMP
E. Type of service: This field contains the type of service being used. This instance is Ping request (type 8 code 0)

00:42:09.693603 146.xyz.xyz.z.2300 > SKINNI.IWANNABE.WIL.domain: S
A. B. C. D. E.
471142303:471142367(**64**) win **2048**
F. G. H.

A. Time stamp: Time detect occurred
B. Source IP Address
C. Source Port
D. Destination Domain name
E. SYN Flag set
F. Sequence number
G. Bytes of data being sent with packet
H. Window size

2. **Detect was generated by**: Shadow detection system. TCPdump format

3. **Probability detect was spoofed**: Although it is not very likely the source address is spoofed, the 64 bytes of data is indicative of a crafted packet; possibly a covert -channel. The nature of the probe would indicate that they would want to receive information back. With a spoofed IP address, all responses would go to the spoofed IP and not to the person attempting to gain the information.

4. **Description of Attack**: ICMP (echo) & Domain (unauth zone xfer) probe. The source IP 146.xyz.xyz.z sends out an ICMP echo request probing for a response This request happens very rapidly indicating an automated/scripted attack. Note the time frame. At **00:42:05.156941,** the 2XX.XX.XXX.33 responds back with SKINNI.IWANNABE.WIL unreachable - admin prohibited filter. Four seconds later at **00:42:09.693603**, the source initiates a TCP connection by sending a packet with the SYN flag set**.** Four reconnect attempts occur within split seconds of each other. Note that the source ports change with each attempt, incrementing by one. Also, the sequence numbers increment at an extremely high rate, with only 64 bytes of data in the field. The normal pattern should be one up with a SYN ACK response bumping the number (the ACK flag =1 byte) and or the bytes of data pushed in the packet would also increment the sequence number by that amount.

   The target IP sends a reply back that the domain is unreachable with an admin prohibited filter. Approximately four seconds later, the source IP initiates a probe of the domain by sending a TCP packet with the SYN flag set and 64 bits of data in the packet.. The target IP does not send a response. 146.xyz.xyz.x then sends another ICMP echo request, resulting with yet another host unreachable admin prohibited filter.

5. **Attack mechanism**: The intruder launches a scripted probe using ICMP requests. Since the targeted system is suppose to echo the identifier and sequence number fields, the attacker is able to determine if its target is "alive" and possible estimate its distance away. Also, any optional data sent by the information gathering machine must be echoed. Note, that even though the name of the destination host was used, the fourth line of output contains the IP address. This indicates that a name resolver, possibly the DNS server, is in its path. The attacker utilizes the information gained to launch another attack. Reconnaissance is the main mission of this probe, but it should also be remembered, that ICMP ping responses are often used as a covert-channel (the 64 bytes of data with the SYN packet could be an indication) The massive DDoS attacks against Internet portals used this as a covert channel. This could be a precursor to a Smurf attack, possible intent to use as an intermediary spot.

6. **Correlations**: ICMP echo reply
   ICMP type 8 ICMP type 0 CA-98.01, "smurf" IP Denial-of-Service Attacks
   http://www.sans.org/y2k/CVE.htm - ICMP

CVE-1999-0214	Reference: XF:icmp-unreachable Denial of service by sending forged ICMP unreachable packets.
CVE-1999-0513	Reference: CERT:CA-98.01.smurf Reference: FreeBSD:FreeBSD-SA-98:06 Reference: XF:smurf ICMP messages to broadcast addresses are allowed, allowing for a Smurf attack that can cause a denial of service.
CVE-1999-0128	Reference: XF:ping-death Reference: CERT:CA-96.26.ping

Oversized ICMP ping packets can result in a denial of service, aka Ping o' Death.

7. **Evidence of active targeting**: There is evidence of active targeting. The attacker is probing the specific domain for a response.
8.. **Severity**: : (Critical + Lethal) - (System Countermeasures + Network Countermeasure) = Severity
   (4+2)-(4+4)= **6-8= *-2***
9. **Defensive recommendation**: Filters are in place to prohibit this activity. Edit Extended ACL so that the echo requests are dropped silently without giving any possible information. Damaging denial of service attacks led to the writing of [2] on Ingress Filtering. Many network providers and corporate networks have endorsed the use of these methods to ensure their networks are not the source of such attacks.

10 **Multiple choice test question**:

1. What type of service is an ICMP Echo Request?
   A. 5
   B. 4
   C. 11
   D. 8

The answer is D.
5. is Redirect
4 is source quench
11 time exceeded

DETECT # 4

Oct 2 09:40:10 deny TCP 208.xx.xxx.61:4724 64.yy.yy.4:23
Oct 2 09:40:10 deny TCP 208.xx.xxx.61:4726 64.yy.yy.5:23
Oct 2 09:40:10 deny TCP 208.xx.xxx.61:4729 64.yy.yy.7:23
Oct 2 09:40:13 deny TCP 208.xx.xxx.61:4726 64.yy.yy.5:23
Oct 2 09:40:18 deny TCP 208.xx.xxx.61:1935 64.yy.yy.7:143
Oct 2 09:40:18 deny TCP 208.xx.xxx.61:1937 64.yy.yy.5:143
Oct 2 09:40:18 deny TCP 208.xx.xxx.61:1938 64.yy.yy.4:143
Oct 3 04:26:59 deny TCP 63.zzz.zz.90:2432 64.yy.yy.4:23
Oct 3 04:26:59 deny TCP 63.zzz.zz.90:2436 64.yy.yy.5:23
Oct 3 04:26:59 deny TCP 63.zzz.zz.90:2450 64.yy.yy.7:23
Oct 3 04:27:02 deny TCP 63.zzz.zz.90:2432 64.yy.yy.4:23
Oct 3 04:27:02 deny TCP 63.zzz.zz.90:2436 64.yy.yy.5:23
Oct 3 04:27:02 deny TCP 63.zzz.zz.90:2450 64.yy.yy.7:23
Oct 3 04:27:04 deny TCP 63.zzz.zz.90:1177 64.yy.yy.5:143
Oct 3 04:27:04 deny TCP 63.zzz.zz.90:1180 64.yy.yy.4:143
Oct 3 04:27:04 deny TCP 63.zzz.zz.90:1190 64.yy.yy.7:143

Oct 3 04:27:07 deny TCP 63.zzz.zz.90:1177 64.yy.yy.5:143
Oct 3 04:27:07 deny TCP 63.zzz.zz.90:1180 64.yy.yy.4:143
Oct 3 04:27:07 deny TCP 63.zzz.zz.90:1190 64.yy.yy.7:143

1. **Source of trace**: http://www.sans.org/y2k/100400.htm

Oct 2 09:40:18	deny TCP	208.xx.xxx.61:1937	64.yy.yy.5:143		
A	B	C	D	E	F

A. Date time stamp
B. ACL rule deny TCP
C. Source IP
D. Source Port
E. Target IP
F. Target Port

2. **Detect was generated by:** Firewall log

3. **Probability the source was spoofed:** This IP would not be spoofed. There would be nothing gained from attempting an imap/telnet session if the source would not receive the information it is seeking. Also, with each new connection attempt, the source port changes.
4. **Description of attack**: Source IP 208.xx.xxx.61 (4724) initiates a telnet session with the target IP 64.yy.yy.4 port (23).When the packet reaches the firewall the ACL rule denies this transaction. Source IP 208.xx.xxx.61:4726 then attempts a new telnet session with 64.yy.yy.5:23 Note that the source port changes as it should with a new connection and that the target IP has changed slightly(host). The firewall again stops this transaction. After three seconds of attempting to connect to three different hosts, another probe is initiated by the attacker. An imap is conducted on the same hosts to port TCP port 143. Again the attempts are blocked by the firewall.
   The next day, a different source IP targets the same destination hosts/ports. Again the attempts are blocked at the firewall. Also of note is that only .4, .7, .5 have been targeted. Both of the source IPs resolve into companies based out of the same state. The first IP resolves seems to be based from a corporation, while the other is a common internet service provider previously associated with other exploits and attacks. If I had access to the systems log ports and databases, I would look to see what other activity has been seen going to this destination IP. Could be one of these scans is a decoy or is being initiated from an already compromised box.
5. **Attack mechanism**: This appears to be a scripted attack. Every three seconds a request to telnet occurs, 2-3 more attempts, then on to the IMAP probe. Each host is hit at least once. The intruder is looking for a remote login to UNIX. Most of the time intruders scan for this port simply to find out more about what operating system is being used. In addition, if the intruder finds passwords using some other technique, they will try the passwords here. Same security idea as POP3 , numerous IMAP servers have buffer overflows that allow compromise during the login. Note that for awhile, there was a Linux worm (admw0rm) that would spread by

compromising port 143, so a lot of scans on this port are actually from innocent people who have already been compromised. IMAP exploits became popular when RedHat enabled the service by default on its distributions.

.

The port identifies which protocol the Telnet session is trying to emulate:

23 FTP (File Transfer Protocol)
25 SMTP (Simple Message Transfer Protocol)
79 Finger
80 HTTP (Hyper-Text Transfer Protocol)
110 POP (Post Office Protocol)
143 IMAP (Internet Mail Access Protocol)

6. **Correlations**: Activitiy from these IP's have been correlated through the JCD2 Data Base on 30SEP00 targeting different IP's using the same telnet/imap sequence. There are at least 25 cve's and 15 candidates pertaining to the vulnerabilities associated with imap and telnet.

http://www.robertgraham.com/pubs/firewall-seen.html#1.1
http://www.cve.mitre.org/cgi-bin/cvekey.cgi?keyword=imap%2Ftelnet
CVE-1999-0073

http://www.sans.org/y2k/CVE.htm - IMAP
IMAP
CVE-1999-0005 Reference: CERT:CA-98.09.imapd
Reference: XF:imap-authenticate-bo
Reference: SUN:00177
Arbitrary command execution via IMAP buffer overflow in authenticate command.
CVE-1999-0042 Reference: NAI:NAI-21
Reference: CERT:CA-97.09.imap_pop
Reference: XF:popimap-bo
Buffer overflow in University of Washington's implementation of IMAP and POP servers.

7. **Evidence of active targeting**: The Target IP has activity on two separate days.

8. **Severity**: (Critical + Lethal) - (System Countermeasures + Network Countermeasure) = Severity

I do not know the type box that was targeted, but will presume the worse case scenario. If these IPs were Firewalls, DNS servers or core routers the Criticality assigned would be 5 A successful telnet session could eventually result in a root compromise. If they guessed the login password the rest is a matter of time. Lethality would therefore be a 5.

All system countermeasures and network countermeasures appear to be in place. Not sure if there any external connections from this system. 4 points would therefore be assigned for

System and 5 for Network countermeasures since the attack was not successful. The formula would look like this:

(5+5) - (4+5) = Severity **10-9 = *1*** The Severity of this attack is 1

9. **Defensive recommendation**: Firewall logs show evidence that the access control lists are working. Do not believe any information has been gleaned by the prober/attacker.

10. **Multiple choice test question**:
What port is the telnet service located?
A. port 21
B. port 20
C. port 143
D. Port 23

The correct answer is **D**. Port 23

Port 21 is FTP (control)
20 is FTP data
143 is IMAP

Assignment 2 Evaluate an Attack

1. **Give the URL, location, or command that attack was acquired from**:
This attack was taken from our local network and has been sanitized. The tools were downloaded from http://www.insecure.org/nmap/

2. **Describe the attack including how it works**:
nmap is a widely available scanner tool. It is one of the most powerful information-gathering tools used by both the aggressor in and attack and the defender. (Network Intrusion Detection: An Analyst's Handbook, Stephen Northcutt) Nmap supports the following:
Vanilla TCP connect() scanning,
TCP SYN (half open) scanning,
TCP FIN, Xmas, or NULL (stealth) scanning,
TCP ftp proxy (bounce attack) scanning
SYN/FIN scanning using IP fragments (bypasses some packet filters),
TCP ACK and Window scanning,

UDP raw ICMP port unreachable scanning,
ICMP scanning (ping-sweep)
TCP Ping scanning
Direct (non portmapper) RPC scanning
Remote OS Identification by TCP/IP Fingerprinting, and
Reverse-ident scanning.

Nmap also offers flexible target and port specification, decoy scanning, determination of TCP sequence predictability characteristics, and output to machine parseable or human readable log files. This same tool can be used to test your own system for vulnerabilities.

How NMAP works: This is an automated scripted file that over the course of designated time span will map the entire network searching for vulnerabilities.
The following NMAP scan was generated on my network. Only snapshots are included for brevity's sake.
The following is a break out of the script used and what each option is:

- nmap -v -P0 -sS -p 80 nmap.got.you.org –O

options: -v verbose

-PO Do not try and ping hosts at all before scanning them. This allows the scanning of networks that don't allow ICMP echo requests (or responses) through their firewall.

-sS TCP SYN scan: This technique is often referred to as "half-open" scanning, because you don't open a full TCP connection. You send a SYN packet, as if you are going to open a real connection and you wait for a response. A SYN|ACK indicates the port is listening. A RST is indicative of a non-listener. If a SYN|ACK is received, a RST is immediately sent to tear down the connection The primary advantage to this scanning technique is that fewer sites will log it. Stealth scanning.

-p <port ranges> This option specifies what ports you want to Scan. The default is to scan all ports between 1 and 1024 as well as any ports listed in the services file which comes with nmap.

Acquired definitions of options by viewing the man pages then cutting and pasting

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

command used to generate this trace: `nmap -v -v -P0 -sS -p 80 nmap.got.you.victim`

This is a snapshot from the source

```
Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Initiating SYN half-open stealth scan against  (nmap.got.you.org)
Adding TCP port 80 (state Open).
The SYN scan took 0 seconds to scan 1 ports.
Interesting ports on  (nmap.got.you.org):
Port    State       Protocol  Service
80      open        tcp       http
```

```
Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
nmap scan port 80, 80 open
```

And this is a copy of the trace in the victim's logs. Of note is how quickly the scan completed it's reconnaissance and gathered the information it desired.

TCP DUMP of NMAP.GOT.YOU.VICTIM LOGS

16:35:36.435030 < nmap.gonna.get.you.ooh.59930 > nmap.got.you.victim.www: S 2226199272:2226199272(0) win 4096
16:35:36.435132 > nmap.got.you.victim.www > nmap.gonna.get.you.ooh.59930: S 3983754609:3983754609(0) ack 2226199273 win 32696 <mss 536> (DF)
16:35:36.435652 < nmap.gonna.get.you.ooh.59930 > nmap.got.you.victim.www: R 2226199273:2226199273(0) win 0
16:35:41.430818 > arp who-has nmap.gonna.get.you.ooh tell nmap.got.you.victim (0:60:97:3c:af:1)
16:35:41.431176 < arp reply nmap.gonna.get.you.ooh is-at 0:0:c0:58:8f:f4 (0:60:97:3c:af:1)

1. At 16:35:36.4345030 nmap.gonna.get.you.ooh sent a packet with the SYN flag set to nmap.got.you.victim.www (port 80) This is a reconnaissance packet sent to the victim in hopes of eliciting response.

2. 16:35:36.435132 > nmap.got.you.victim.www > nmap.gonna.get.you.ooh.59930: S 3983754609:3983754609(0) ack 2226199273 win 32696 <mss 536> (DF) BINGO! nmap.got.you.victim. responds with a SYN/ACK, the second part of the three way handshake. This lets the aggressor know that this port is open.

3. The aggressor nmap.gonna.get.you.ooh then resets the connection. It has almost completed this part of his mission.

4. The final snipet of this trace is perhaps the most significant of this simple trace because the MAC addresses have been exchanged. This not only opens the door for the attacker to exploit, but also may provide the victim command with a means of tracing the attacker to it's box.

Assignment 3 - "Analyze This" Scenario

Analysis of Snort detects for approximately one month using a standard rule base.
Snort Alert logs start 29 June 00 and end 06 Aug 00.
Snort Scan logs start 30 June 00 and end 10 Aug 00.
Unfortunately data is incomplete due to power failures and/or the back up disk was full.
Therefore there are lapses in alert and scan logs resulting in days not accounted for.

Utilized SnortSnarf v10094001.1 to analyze date.
(available at www.silicondefense.com/snortsnarf/mai n.html- created by (Jim Hoagland and Stuart Staniford))

Merged alert logs into single file SnortMergA.txt. Merged scan logs into single SnortMergS.txt.
Substituted 255.254 for MY.NET in order for the perl script to run properly and then generated the following with Snortsnarf.

Utilized SnortSnarf to analyze data. (available at www.silicondefense.com/snortsnarf/main.html) After compilation was complete, cut and pasted log files, then replaced 255.254 with MY.NET for easy recognition.

362199 alerts processed. Table below lists Snort Signatures.

Signature	# Alerts	# Sources	# Destinations
FTP-bad-login	1	1	1
Telnet daemon-active	1	1	1
PING-ICMP Source Quench	1	1	1
Back Orifice	1	1	1
Possible wu-ftpd exploit	2	1	2
Queso fingerprint	3	3	3
Happy 99 Virus	4	4	4
wu-ftpd exploit	5	3	4
large ICMP Packet	5	5	1
TELNET - Login Incorrect	7	3	6
External RPC call	8	2	1
Tiny Fragments-Possible Hostile activity	9	3	3
Napster Client Data	12	8	7
SUNRPC highport access!	18	3	3
Null scan!	30	20	19
NMAP TCP ping!	45	6	5
Napster 7777 Data	170	14	13
GIAC 000218 VA-CIRT port 35555	182	28	9
GIAC 000218 VA-CIRT port 34555	196	25	9
SMB Name Wildcard	229	5	4
Napster 8888 Data	323	8	8
SNMP public access	1147	28	1
MISC - Large UDP Packet	1170	1	1
WinGate 1080 Attempt	2042	353	305
Attempted Sun RPC high port access	2241	10	8
WinGate 8080 Attempt	3222	89	16
Watchlist 000222 NET-NCFC	4711	40	12
PING-ICMP Time Exceeded	6689	299	117
PING-ICMP Destination Unreachable	12313	133	144
Watchlist 000220 IL-ISDNNET-990517	13962	19	17
SYN-FIN scan!	19844	11	19801

Reviewed results for Snortsnarf and found following items of interest:

Most active sources:

# of alerts	IP address	type of alarm/exploit	whois
1	MY.NET.99.51	ACTIVE TELNET DAEMON	
1	209.245.5.158	ICMP SOURCE QUENCH	
1	202.159.46.234	BACKORRIFICE	INDONET, INDONESIA
2	151.164.223.206	WU-FTPD	SOUTHWESTERN BELL,TX
1	24.3.29.155	QUESO FINGERPRINT	@ HOME , MD
1	210.84179.196	QUESO FINGERPRINT	OZEMAIL2-AU
1	192.203.80.142	QUESO FINGERPRINT	RUSSIAN ACADEMY OF SCI
1	203.251.136.2	HAPPY 99 VIRUS	KOREA TELECOM
1	200.223.11.7	HAPPY 99 VIRUS	RNP BRAZIL
1	206.67.51.242	HAPPY 99 VIRUS	MEDIA 3 TECH
1	208.130.42.17	HAPPY 99 VIRUS	LOGON AMERICA
6	63.236.34.174	TINY FRAGMENTS	QUOKA SPORTS
14	205.188.3.205	SUNRPC HIGH PORT ACCESS	AOL
3	210.121.242.164	NULL SCAN	KOREA TELECOM
5	149.225.111.69	NULL SCAN	AUNET, DE
23	205.128.11.157	NMAP TCP PING	HEADHUNTER NET
90	208.184.216.183	NAPSTER 7777 DATA	ABOVENET
14	207.217.120.29	GAIC PORT 35555	EARTHLINK
63	152.163.224.100	GIAC 000218 VA-CIRT PORT 34555	AOL
219	MY.NET.101.160	SMB NAME WILDCARD	
205	208.184.216.189	NAPSTER 8888 DATA	ABOVENET
131	MY.NET.97.237	SNMP PUBLIC ACCESS	
159	MY.NET.97.80	SNMP PUBLIC ACCESS	
208	MY.NET.97.186	SNMP PUBLIC ACCESS	
1170	211.40.176.214	large UDP packet	BORANET KOREA
155	168.120.16.250	WINGATE 1080	ASSUMPTION UNIVERSITY, TH
104	208.240.218.220	WINGATE 1080	PROF. COMPUTER SERVICES
2166	205.188.153.111	SUNRPC	AOL, VIRGINIA
1145	128.231.171.123	WINGATE 8080	National Inst of Health (1 DEST)
275	24.3.26.53	WINGATE	@ HOME MD, CATV (1 Dest)
222	216.0.124.26	WINGATE	DIGEX INC, MD
19818	202.0.178.98	SYN/FIN	China Motion Telcom Holdings Ltd.
4923	24.23.96.119 (PA)	dest unreachable	@Home Network
2346	24.4.52.197 (TX)	dest unreachable	@Home Network
801	MY.NET.14.2	ICMP time exceeded	MY.NET (112 destinations)

Most active destinations:

# of alerts	IP address	type of alarm/exploit	whois
1	24.25.111.117	TIME WARNER ROADRUNNER MN	
1	MY.NET.70.121	ICMP SOURCE QUENCH	
1	MY.NET.100.100	BACKORRIFICE	
1	MY.NET.99.16	WU-FTPD	
1	MY.NET.144.59	WU-FTPD	
1	MY.NET.60.8	QUESO	
1	MY.NET.6.44	QUESO	
1	MY.NET.99.23	QUESO	
1	MY.NET.110.150	HAPPY 99	
1	MY.NET.253.42	HAPPY 99	
1	MY.NET.6.47	HAPPY 99	
1	MY.NET.6.34	HAPPY 99	

6	MY.NET.1.8	TINY FRAGMENTS
14	MY.NET.98.145	SUNRPC HIGH PORT ACCESS
5	MY.NET.60.14	NULL SCAN
4	MY.NET.100.236	NULL SCAN
34	MY.NET.1.8	NMAP
90	MY.NET.97.204	NAPSTER 7777
74	MY.NET.253.24	GIAC 000218 35555
115	MY.NET.253.24	GIAC 000218 34555
219	MY.NET.101.192	SMB NAME WILDCARD
249	MY.NET.201.2	NAPSTER 8888
1147	MY.NET.101.192	SNMP PUBLIC ACCESS
1170	MY.NET.98.179	LARGE UDP PACKET
150	MY.NET.60.16	WINGATE 1080
241	MY.NET.60.8	WINGATE 1080
285	MY.NET.60.11	WINGATE 1080
2166	MY.NET.217.126	SUNRPC
2854	MY.NET.253.105	WINGATE (51 SOURCES)
11305	MY.NET.70.121	dest unreachable
271	MY.NET.140.9	dest unreachable
5830	MY.NET.140.9	ICMP Time exceeded

Total number of alerts from Israel: 13962 (watch list 000220 IL-ISDNET-990517)
Total number of alerts from China: 4711 (watch list 000222 NET-NCFC)

Watchlist 000220 IL-ISDNNET-990517

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
212.179.38.141	4320	4320	1	1
212.179.19.134	3231	3231	1	1
212.179.41.218	1970	1970	1	1
212.179.54.69	1805	1805	1	1
212.179.23.4	1702	1702	1	1
212.179.4.238	730	730	1	1
212.179.101.218	85	85	1	1
212.179.123.13	64	64	1	1
212.179.69.68	10	10	1	1
212.179.27.6	9	9	1	1
212.179.126.2	7	7	1	1
212.179.125.114	6	6	1	1
212.179.126.8	4	4	1	1
212.179.29.132	4	4	1	1
212.179.103.179	4	4	1	1
212.179.5.131	4	4	2	2
212.179.103.232	4	4	1	1
212.179.30.29	2	2	1	1
212.179.58.2	1	1	1	1

**********NOTEWORTHY EVENTS**********

The first alarm that is of interest is the **SYN–FIN scan**. The majority of the alarms were generated by IP address 202.0.178.98 (19818).
19844 alerts (11 sources - 202.0.178.98 foreign – China triggered 19818 of those alerts)A SYN-FYN scan is also an intelligence gathering tool. The hacker, by crafting the FYN into the packet is sometimes able to sneak through the firewall undetected and gain knowledge from within. If a box were to respond to this packet with a SYN-FIN-ACK, one might be able to gather that the box is linux(Network Intrusion Detection, An analyst's Handbook Stephen Northcutt pg. 98)

07/14-16:09:40.239312 [**] Watchlist 000222 NET-NCFC [**] 159.226.49.23:4552 -> MY.NET.145.9:25
07/17-11:51:24.229184 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.4.238:1072 -> MY.NET.53.28:4110

.
6/28/00 IP address **202.0.178.98** a scripted automated SYN-FINscan of nearly all subnets from 225.254.1.3-255.254.254.255 against port 53 (DNS). The duration of the attack was from 0652:28-0714:23. The duration of the attack on each subnet lasted for exactly 5 seconds per subnet scanned (very fast!). Source IP resolves to:
inetnum: 202.0.160.0 - 202.0.179.255
netname: CMNET-HK
descr: China Motion Telcom Holdings Ltd.
descr: Roaming Paging Services Provider
descr: Roaming Trunking Services Provider
descr: Hong Kong

6/29/00 IP address **210.222.31.100** conducted probes to ports 1524 (ingreslock) and 2222 (Allen-Bradley unregistered port) on IP addresses 255.254.1.4 and 255.254.1.5 respectively. Source IP resolves to:
IP Address : 210.222.31.96-210.222.31.127
Connect ISP Name : KORNET
Connect Date : 1999.09.17
Registration Date: 19991027
Network Name : KRJD-GAME

IP address **207.236.111.226** conducted activity against 255.254.1.4 source/dest port 21 (FTP).
Bell Global Network Operations (NETBLK-BELLGLOBAL-2)
160 Elgin Street, Floor 12
Ottawa, Ontario K2P 2C4
Ca
7/11/00 1910:54 IP address **210.222.31.100** (Network Name : KRJD-GAME)
conducts SYN-FIN against 255.254.1.4-1.5 on ports 1524 (ingreslock).
IP Address : 210.222.31.96-210.222.31.127
Connect ISP Name : KORNET
Connect Date : 1999.09.17
Registration Date: 19991027
Network Name : KRJD-GAME

7/17/00 0304:29 IP address **200.255.45.3**7 conducted 0304:29 IP address 200.255.45.37 conducted a SYN-FIN against 255.254.1.4-1.5 with a source and destination port of 25 (smtp).
RNP (Brazilian Research Network) (NETBLK-BRAZIL-BLK2)
Rua Pio XI, 1500
Sao Paulo, 05468-901

BR

7/29/00 IP address **208.50.27.150** conducted probes to ports 53 on addresses 255.254.1.3-1.5. The times were 1306:49 and 1751:09 (2 sets).
Source IP resolves to:
UB Networks (NETBLK-FGC-REQ000000004806-1)
624 S Grand 1 Wilshire BLDG
Los Angeles, CA 90007
US
IP address **212.177.241.139** at 1752:43 conducted SYN-FIN activity against port 109 (pop-2)
8/1/00 IP address 207.0.62.254 conducted probes to port 1524 and 9704 to addressed 255.254.1.4-1.5 at 0442:24 and 1432:06 respectively.
inetnum: 212.177.0.0 - 212.177.255.255
netname: IT-UUNET-990512
descr: PROVIDER
country: IT

8/3/00 IP address **206.78.1.18** conducted SYN-FIN scan to 255.254.1.4-1.5, source/destination port of 21(FTP).
Tulare County Office of Education (NETBLK-TCOENET-0-31)
2637 West Burrel
Visalia, CA 93278-5091
US

IP address **63.69.63.2** conducted activity against 255.254.1.4 source/dest port 21 (FTP).
Guthrie & Assoc. & Realty (NETBLK-UU-63-69-63)
1357 Washington Street
Clarkesville, GA 30523
US
8/5/00 IP address **63.16.52.48** conducted SYN-FIN activity against 255.254.1.4-1.5 port 53 (DNS).
UUNET Technologies, Inc. (NETBLK-NETBLK-UUNET97DU)
3060 Williams Drive, Suite 601
Fairfax, va 22031
US
The SYN-FIN scan is an information gathering mission on the part of the attacker. This flag combination is the result of a crafted packet and set to elicit a response from the
victim.

07/29-13:06:49.354956 [**] SYN-FIN scan! [**] 208.50.27.150:21 -> MY.NET.1.**3**:21
07/29-13:06:49.369729 [**] SYN-FIN scan! [**] 208.50.27.150:21 -> MY.NET.1.4:21
07/29-13:06:49.400032 [**] SYN-FIN scan! [**] 208.50.27.150:21 -> MY.NET.1.5:21
07/29-15:30:46.393217 [**] SYN-FIN scan! [**] 212.177.241.139:80 -> MY.NET.1.5:80
07/29-17:51:09.959388 [**] SYN-FIN scan! [**] 208.50.27.150:21 -> MY.NET.**1.3**:21
07/29-17:51:09.973574 [**] SYN-FIN scan! [**] 208.50.27.150:21 -> MY.NET.1.4:21
07/29-17:51:09.993383 [**] SYN-FIN scan! [**] 208.50.27.150:21 -> MY.NET.1.5:21
07/29-17:52:43.151531 [**] SYN-FIN scan! [**] 212.177.241.139:109 -> MY.NET.1.3:109

07/29-13:06:49.354956 [**] SYN-FIN scan! [**] 208.50.27.150:21 -> MY.NET.1.**3**:21
07/29-13:06:49.369729 [**] SYN-FIN scan! [**] 208.50.27.150:21 -> MY.NET.1.4:21
07/29-13:06:49.400032 [**] SYN-FIN scan! [**] 208.50.27.150:21 -> MY.NET.1.5:21
07/29-15:30:46.393217 [**] SYN-FIN scan! [**] 212.177.241.139:80 -> MY.NET.1.5:80
07/29-17:51:09.959388 [**] SYN-FIN scan! [**] 208.50.27.150:21 -> MY.NET.1.3:21
07/29-17:51:09.973574 [**] SYN-FIN scan! [**] 208.50.27.150:21 -> MY.NET.1.4:21
07/29-17:51:09.993383 [**] SYN-FIN scan! [**] 208.50.27.150:21 -> MY.NET.1.5:21
07/29-17:52:43.151531 [**] SYN-FIN scan! [**] 212.177.241.139:109 -> MY.NET.1.3:109

06/28-06:52:55.149077 [**] SYN-FIN scan! [**] **202.0.178.98:53** -> MY.NET.2.88:53
06/28-06:52:55.263743 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.2.93:53

06/28-06:52:55.268354 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.2.95:53
06/28-06:52:55.319495 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.2.98:53

NAPSTER (port 6699)

07/26-05:04:31.407020 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.54.69:6699 -> MY.NET.182.94:3661
07/26-05:04:31.861892 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.54.69:6699 -> MY.NET.182.94:3661

Jul 27 12:45:02 24.112.193.183:6699 -> MY.NET.182.71:2334 NOACK 2*S*R*** RESERVEDBITS
Jul 27 12:45:05 24.112.193.183:6699 -> MY.NET.182.71:2334 NOACK 2*S*R*** RESERVEDBITS
Jul 27 13:32:23 24.166.184.108:2116 -> MY.NET.98.107:6699 INVALIDACK ****R*AU
Jul 27 13:58:30 24.166.184.108:2116 -> MY.NET.98.107:6699 INVALIDACK ****R*AU

08/05-18:30:07.112277 [**] Napster 8888 Data [**] MY.NET.201.2:1463 -> 208.184.216.191:8888
08/05-18:30:07.201812 [**] Napster 8888 Data [**] MY.NET.201.2:1463 -> 208.184.216.191:8888

Aug 4 12:35:38 193.150.235.135:52547 -> MY.NET.20.10:8888 SYN **S*****
Aug 10 17:34:03 64.244.202.66:62949 -> MY.NET.179.86:8888 SYN **S*****
Jul 24 21:56:58 209.123.109.175:1706 -> MY.NET.98.118:8888 SYN **S*****

Jul 9 21:26:06 165.138.228.4:7777 -> MY.NET.97.68:2077 UDP
Jul 9 21:26:06 165.138.228.4:7777 -> MY.NET.97.68:2079 UDP
Jul 9 21:26:06 165.138.228.4:7777 -> MY.NET.97.68:2081 UDP

Jul 17 19:24:59 199.178.222.88:7777 -> MY.NET.153.111:2928 UDP
Jul 17 19:25:02 199.178.222.88:7777 -> MY.NET.153.109:1059 UDP
Jul 17 19:25:02 199.178.222.88:7777 -> MY.NET.153.111:2929 UDP

This is a very common signature with false positives triggered by people accessing the popular Naptser.com site and downloading audio files (MP3 files). The significance with these traces is that the whois look up for 212.1799.54.69 resolves to country: IL.(noted on theWatch list) There would not be a valid reason for this foreign IP to attempt to access this. They are most likely looking to exploit this potential vulnerability. Since these files can be quite large and consume a lot of bandwidth, this detect may be a denial of service against the client computer. Two CVE candidates are relevant to napster:

** CANDIDATE (under review) CAN-2000-0281 ** Buffer overflow in the Napster client beta 5 allows remote attackers to cause a denial of service via a long message.

** CANDIDATE (under review) CAN-2000-0412 ** The gnapster and knapster clients for Napster do not properly restrict access only to MP3 files, which allows remote attackers to read arbitrary files from the client by specifying the full pathname for the file.

WINTRINOO Trojan (Ddos)

Dest port 34555 – 196 hits 25 sources – 9 destinations

The following communications to ports 34555 and 35555 are indicative of Wintrinoo, a distributed denial of service (DDos) tool. If the network is compromised by Wintrinoo, you may unwittingly become a participant in a DDos attack against a large organization, similar to recent attacks against eBay and Yahoo.

The latest control patch 660 with engine version 5.000-1119 can detect this Trojan. Upon execution this Trojan virus becomes resident in memory wherein it waits for the master server file it needs to function and gain control. In doing so, it opens port number "34555" specific to another client program.

Any client that knows the IP Address of the computer where this Server Trojan is executed, could gain access known to the client program.

```
[navcirt@thematrix ~]$ more SnortMerg*.txt |grep 34555 | more
07/14-17:24:07.822935  [**] GIAC 000218 VA-CIRT port 34555 [**] 165.251.8.74:25
-> 255.254.253.24:34555
07/14-17:24:07.836480  [**] GIAC 000218 VA-CIRT port 34555 [**] 165.251.8.74:25
-> 255.254.253.24:34555
07/14-17:24:08.041518  [**] GIAC 000218 VA-CIRT port 34555 [**] 165.251.8.74:25
-> 255.254.253.24:34555
07/14-17:24:08.095022  [**] GIAC 000218 VA-CIRT port 34555 [**] 165.251.8.74:25
-> 255.254.253.24:34555
07/14-17:24:08.217332  [**] GIAC 000218 VA-CIRT port 34555 [**] 165.251.8.74:25
-> 255.254.253.24:34555
07/14-17:24:08.311098  [**] GIAC 000218 VA-CIRT port 34555 [**] 165.251.8.74:25
-> 255.254.253.24:34555
07/14-17:24:08.386778  [**] GIAC 000218 VA-CIRT port 34555 [**] 165.251.8.74:25
-> 255.254.253.24:34555
07/14-17:24:09.818616  [**] GIAC 000218 VA-CIRT port 34555 [**] 165.251.8.74:25
-> 255.254.253.24:34555
07/14-17:24:09.820191  [**] GIAC 000218 VA-CIRT port 34555 [**] 165.251.8.74:25
 ->255.254.253.24:34555

7/27-02:24:55.894950  [**] GIAC 000218 VA-CIRT port 34555 [**] 165.166.0.25:25
-> 255.254.253.24:34555
07/27-02:24:56.098479  [**] GIAC 000218 VA-CIRT port 34555 [**] 165.166.0.25:25
-> 255.254.253.24:34555
07/27-02:25:24.196893  [**] GIAC 000218 VA-CIRT port 34555 [**] 165.166.0.25:25
-> 255.254.253.24:34555
07/27-02:25:24.197032  [**] GIAC 000218 VA-CIRT port 34555 [**] 165.166.0.25:2
```

Relevent CVE candidates:

CAN-2000-0138

** CANDIDATE (under review) ** A system has a distributed denial of service (DDOS) attack master, agent, or zombie installed, such as (1) Trinoo, (2) Tribe Flood Network (TFN), (3) Tribe Flood Network 2000 (TFN2K), (4) stacheldraht, (5) mstream, or (6) shaft.

Back Orifice

Foreign source - Indonesia.

```
07/12-17:16:32.897041  [**] Back Orifice [**] 202.159.46.234:31338 -> MY.NET.100.130:31337
```

BackOrifice is a program that allows hackers to access and even control someone else's PC, over the Internet. It was released in August of 1998 by a group of hackers calling themselves The Cult of the Dead Cow (they call it a "remote administration tool). BackOrifice can only affect a machine on which it's been deliberately installed, and it works only on computers running Windows 95 or 98. Once you detect BackOrifice, you can neutralize it fairly quickly. To find out whether or not BackOrifice is installed on your machine, you can search your hard drive for a file called "windll.dll," which BackOrifice creates whenever it runs

Relevent CVE candidates:

CAN-1999-0660

** CANDIDATE (under review) ** A hacker utility or Trojan Horse is installed on a system, e.g. NetBus, Back Orifice, Rootkit, etc.

CAN-2000-0562

** CANDIDATE (under review) ** BlackIce Defender 2.1 and earlier, and BlackIce Pro 2.0.23 and earlier, do not properly block Back Orifice traffic when the security setting is Nervous or lower.

Happy 99 Virus

This signature demonstrates how a virus can infiltrate the system. This worm, when executed changes the date and time stamp. Seems more of a nuisance than anything else. But it does indicate poor system security. The systems antivirus software needs to be updated to prohibite this activity from occurring.

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
08/05-11:22:48.017066 [**] Happy 99 Virus [**] 206.67.51.242:4889 -> MY.NET.6.47:25
07/11-19:28:57.652242 [**] Happy 99 Virus [**] 200.223.11.7:4836 -> MY.NET.110.150:25

Wingate 1080 SOCKS

155 Wingate scans from foreign source. This is just a small snipet of the activity

07/14-00:03:20.138859 [**] WinGate 1080 Attempt [**] 168.120.16.250:55067 -> MY.NET.97.135:1080
07/14-00:04:04.529242 [**] WinGate 1080 Attempt [**] 203.155.129.248:4387 -> MY.NET.97.135:1080

Most scans for port 1080 are actually looking for WinGate, a popular firewall/proxy for Windows
Relevent CVE's:

CVE-1999-0290

The WinGate telnet proxy allows remote attackers to cause a denial of service via a large number of connections to localhost.

CVE-1999-0291

The WinGate proxy is installed without a password, which allows remote attackers to redirect connections without authentication.

CVE-1999-0441

Remote attackers can perform a denial of service in WinGate machines using a buffer overflow in the Winsock Redirector Service.

CVE-1999-0494

Denial of service in WinGate proxy through a buffer overflow in POP3.

NMAP ping

This is a major intelligence gathering tool. The variety of scanning modes available as well as TCP fingerprinting and TCP sequence number prediction difficulty makes this tool one of the most powerful.

07/28-23:32:23.408944 [**] NMAP TCP ping! [**] 216.127.150.136:57882 -> MY.NET.253.114:1
08/04-08:01:02.191197 [**] NMAP TCP ping! [**] 195.25.86.2:80 -> MY.NET.179.77:80
08/04-10:49:10.811041 [**] NMAP TCP ping! [**] 205.128.11.157:80 -> MY.NET.1.8:53
08/04-10:49:10.811088 [**] NMAP TCP ping! [**] 205.128.11.157:53 -> MY.NET.1.8:53
08/04-11:18:28.348261 [**] NMAP TCP ping! [**] 205.128.11.157:80 -> MY.NET.1.8:53
08/04-11:18:28.348302 [**] NMAP TCP ping! [**] 205.128.11.157:53 -> MY.NET.1.8:53

07/12-12:46:34.921774 [**] Probable NMAP fingerprint attempt [**] 24.200.160.45:1548 -> MY.NET.70.241:8899
www.insecure.org

SunRPC Port 32771

07/19-14:26:12.632395 [**] Attempted Sun RPC high port access [**] 24.4.129.16:407 -> MY.NET.115.91:32771
07/19-14:26:12.632451 [**] Attempted Sun RPC high port access [**] 24.4.129.16:1419 -> MY.NET.115.91:**32771**

Connection attempts are being made to port **32771**. Under Solaris, the Rpcbind service listens on port 32771 in addition to the standard port 111. It is very likely that the attackers are attempting to connect to this service in order to find out what RPC services are being offered. There are several known buffer overflow vulnerabilities with RPC services that can be exploited to grant root access.

Ghost Portmapper: Some SunOS machines listen at this port for portmapper. Since firewalls frequently don't filter at high ports, it can allow the attacker access to portmapper even when port 111 is blocked.

NULL SCAN

07/14-12:28:25.838842 [**] Null scan! [**] 24.232.51.137:1152 -> MY.NET.110.57:6688
07/14-12:28:29.384871 [**] Null scan! [**] 24.232.51.137:1152 -> MY.NET.110.57:6688
08/03-19:59:23.823304 [**] Null scan! [**] 149.225.111.69:7904 -> MY.NET.60.14: 37
08/03-19:59:23.877719 [**] Null scan! [**] 149.225.111.69:7904 -> MY.NET.60.14:137
08/03-19:59:23.971660 [**] Null scan! [**] 149.225.111.69:7904 -> MY.NET.60.14:513
06/30-08:26:02.939759 [**] NMAP TCP ping! [**] 195.25.86.2:80 -> MY.NET.60.14:80

The NULL scan can collect a lot of information about the Windows system if queries are allowed through the firewall on port 137. It is logging into the system as a nobody user.

Ports with known vulnerabilities are scanned in the above session:
1302- unassigned
7-echo
22-SSH Remote Login Protocol
37-Time
137-NETBIOS Name Service
513-remote login via telnet
80-World Wide Web HTTP
53-Domain Name Server

Following Systems possibly compromised:

MY.NET.1.3

07/19-09:49:16.702569 [**] Queso fingerprint [**] 212.171.169.46:24122 -> MY.NET.1.3:21

Jul 29 13:06:49 208.50.27.150:21 -> MY.NET.1.3:21 SYNFIN **SF****

07/14-13:48:58.394814 [**] spp_portscan: portscan status from MY.NET.1.3: 10 connections across 2 hosts: TCP(0), UDP(10) [**]
07/14-13:49:00.680576 [**] spp_portscan: End of portscan from MY.NET.1.3 (TOTAL HOSTS:2 TCP:0 UDP:10) [**]

Aug 5 10:32:29 MY.NET.1.3:53 -> MY.NET.101.89:41909 UDP
Aug 5 10:32:29 MY.NET.1.3:53 -> MY.NET.101.89:41910 UDP
Aug 5 10:32:29 MY.NET.1.3:53 -> MY.NET.101.89:41911 UDP
Aug 5 10:32:29 MY.NET.1.3:53 -> MY.NET.101.89:41912 UDP

Aug 5 10:32:29 MY.NET.1.3:53 -> MY.NET.101.89:41913 UDP
Aug 5 10:32:29 MY.NET.1.3:53 -> MY.NET.101.89:41914 UDP
Aug 5 10:32:29 MY.NET.1.3:53 -> MY.NET.101.89:41916 UDP

MY.NET.1.3 performed scans against it's own network after being scanned and fingerprinted by outside sources. The fact that these scans occurred points to a compromised system. Not a good thing.

MY.NET.1.8

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
07/08-07:21:32.145547 [**] Attempted Sun RPC high port access [**] 64.27.29.2:2385 -> MY.NET.1.8:32771
07/08-07:33:06.203162 [**] Attempted Sun RPC high port access [**] 207.230.26.34:1295 -> MY.NET.1.8:32771
07/08-20:02:37.444826 [**] NMAP TCP ping! [**] 209.218.228.46:80 -> MY.NET.1.8:53

MY.NET.1.8 was victim of reconnaissance thru NMAP (as stated previously, this is a very powerful tool During this time frame, SunRPC high port access was achieved.
There were 42 alerts going to MY.NET.1.8

MY.NET.99.51

07/26-02:46:25.820700 [**] WinGate 1080 Attempt [**] 207.114.4.46:3875 -> MY.NET.99.51:1080
07/28-05:44:51.442479 [**] WinGate 1080 Attempt [**] 207.114.4.46:1272 -> MY.NET.99.51:1080
08/05-19:03:45.522918 [**] IDS08 - TELNET - daemon-active [**] MY.NET.99.51:23-> 24.25.111.117:1029
06/29-04:40:46.546586 [**] WinGate 1080 Attempt [**] 207.114.4.46:3816 -> MY.NET.99.51:1080
06/30-05:54:34.091505 [**] WinGate 1080 Attempt [**] 207.114.4.46:4360 -> MY.NET.99.51:1080

MY.NET.99.51 recieved numerous WinGate proxy scans. **Telnet daemon indicates successful telnet connection** has been established from outside local network. Telnet is a very insecure protocol and should be replaced with SSH.

DISTRIBUTED DENIAL OF SERVICE

On 08/05 a massive denial of service was initiated against MY.NET. Note that the above mentioned Telnet Daemon occurred during this activity. Ran a grep script against the merged files to see what activity IP 24.25.111.117 had previously initiated. No data resulted from the query. Makes me question if this activity were a decoy to cover the activity of the one connection.

18. 03 -> MY.NET.70.121
08/05-18:30:02.462952 [**] PING-ICMP Destination Unreachable [**] 209.86.165.10
5 -> MY.NET.70.121
08/05-18:30:02.467568 [**] PING-ICMP Destination Unreachable [**] 209.86.165.10
5 -> MY.NET.70.121
08/05-18:30:02.619108 [**] PING-ICMP Destination Unreachable [**] 209.178.160.2
03 -> MY.NET.70.121
08/05-18:30:02.683382 [**] PING-ICMP Destination Unreachable [**] 209.86.165.10
5 -> MY.NET.70.121
08/05-18:30:02.805540 [**] PING-ICMP Destination Unreachable [**] 216.127.194.3
7 -> MY.NET.70.121
08/05-18:30:03.032120 [**] PING-ICMP Destination Unreachable [**] 216.127.194.3

```
7 -> MY.NET.70.121
08/05-18:30:03.264610  [**] PING-ICMP Destination Unreachable [**] 216.127.194.3
7 -> MY.NET.70.121
08/05-18:30:03.268228  [**] PING-ICMP Destination Unreachable [**] 209.178.160.2
03 -> MY.NET.70.121
08/05-18:30:08.577356  [**] PING-ICMP Time Exceeded [**] 198.32.8.29 -> MY.NET.1
40.9
08/05-18:30:08.603764  [**] PING-ICMP Time Exceeded [**] 198.32.8.29 -> MY.NET.1
40.9
08/05-18:30:08.626256  [**] PING-ICMP Time Exceeded [**] 198.32.8.29 -> MY.NET.1
40.9
08/05-18:30:08.649502  [**] PING-ICMP Time Exceeded [**] 192.88.115.122 -> MY.NE
T.140.9
08/05-18:30:08.668185  [**] PING-ICMP Time Exceeded [**] 192.88.115.122 -> MY.NE
T.140.9
08/05-18:30:08.686245  [**] PING-ICMP Destination Unreachable [**] 216.127.194.3
7 -> MY.NET.70.121
08/05-18:30:08.691255  [**] PING-ICMP Time Exceeded [**] 192.88.115.122 -> MY.NE
T.140.9
08/05-18:30:08.714035  [**] PING-ICMP Time Exceeded [**] 198.32.224.66 -> MY.NET
.140.9
08/05-18:30:08.725620  [**] PING-ICMP Destination Unreachable [**] 209.178.160.2
03 -> MY.NET.70.121
08/05-18:30:08.743617  [**] PING-ICMP Time Exceeded [**] 198.32.224.66 -> MY.NET
.140.9
08/05-18:30:08.775434  [**] PING-ICMP Time Exceeded [**] 198.32.224.66 -> MY.NET
.140.9
08/05-18:30:09.495411  [**] PING-ICMP Destination Unreachable [**] MY.NET.70.121
 -> 209.49.106.28
08/05-18:30:09.501664  [**] PING-ICMP Destination Unreachable [**] MY.NET.70.121
 -> 64.252.35.162
08/05-18:30:09.502259  [**] PING-ICMP Destination Unreachable [**] MY.NET.70.121
 -> 63.205.40.169
08/05-18:30:09.502311  [**] PING-ICMP Destination Unreachable [**] MY.NET.70.121
 -> 24.112.94.71
08/05-18:30:09.503262  [**] PING-ICMP Destination Unreachable [**] MY.NET.70.121
 -> 24.168.8.137
08/05-18:30:09.523701  [**] PING-ICMP Destination Unreachable [**] MY.NET.70.121
 -> 213.200.186.173
08/05-18:30:09.559708  [**] PING-ICMP Destination Unreachable [**] MY.NET.70.121
 -> 24.4.52.197
08/05-18:30:09.576522  [**] PING-ICMP Destination Unreachable [**] MY.NET.70.121
 -> 24.129.222.8
08/05-18:30:09.598026  [**] PING-ICMP Destination Unreachable [**] MY.NET.70.121
24.17.201.70
08/05-18:30:13.755720  [**] IDS247 - MISC - Large UDP Packet [**] 211.40.176.214
:29536 -> MY.NET.98.179:6970
08/05-18:30:13.757070  [**] IDS247 - MISC - Large UDP Packet [**] 211.40.176.214
:29536 -> MY.NET.98.179:6970
08/05-18:30:13.760533  [**] IDS247 - MISC - Large UDP Packet [**] 211.40.176.214
:29536 -> MY.NET.98.179:6970
08/05-18:30:24.096348  [**] IDS247 - MISC - Large UDP Packet [**] 211.40.176.214
:29536 -> MY.NET.98.179:6970
08/05-18:30:24.098000  [**] IDS247 - MISC - Large UDP Packet [**] 211.40.176.214
:29536 -> MY.NET.98.179:6970
08/05-18:30:24.100646  [**] IDS247 - MISC - Large UDP Packet [**] 211.40.176.214
```

```
:29536 -> MY.NET.98.179:6970
08/05-18:30:29.402872  [**] IDS247 - MISC - Large UDP Packet [**] 211.40.176.214
:29536 -> MY.NET.98.179:6970
[navcirt@thematrix ~]$ more SnortMergA.txt | grep 08/05-19:03:45 | more
08/05-19:03:45.028697  [**] PING-ICMP Time Exceeded [**] 198.32.248.61 -> MY.NET
.140.9
08/05-19:03:45.283695  [**] PING-ICMP Time Exceeded [**] 206.196.178.5 -> MY.NET
.140.9
08/05-19:03:45.283891  [**] PING-ICMP Time Exceeded [**] 206.196.178.5 -> MY.NET
.140.9
08/05-19:03:45.284581  [**] PING-ICMP Time Exceeded [**] 206.196.178.5 -> MY.NET
.140.9
08/05-19:03:45.290169  [**] PING-ICMP Time Exceeded [**] 206.196.177.9 -> MY.NET
.140.9
08/05-19:03:45.297401  [**] PING-ICMP Time Exceeded [**] 206.196.177.9 -> MY.NET
.140.9
08/05-19:03:45.327004  [**] PING-ICMP Time Exceeded [**] 198.32.8.45 -> MY.NET.1
40.9
08/05-19:03:45.338953  [**] PING-ICMP Time Exceeded [**] 198.32.8.45 -> MY.NET.1
40.9
08/05-19:03:45.372818  [**] PING-ICMP Time Exceeded [**] 198.32.8.65 -> MY.NET.1
40.9
08/05-19:03:45.409750  [**] PING-ICMP Time Exceeded [**] 198.32.8.65 -> MY.NET.1
40.9
08/05-19:03:45.522918  [**] IDS08 - TELNET - daemon-active [**] MY.NET.99.51:23
-> 24.25.111.117:1029
08/05-19:03:45.633509  [**] PING-ICMP Time Exceeded [**] 206.196.178.5 -> MY.NET
[navcirt@thematrix ~]$ more SnortMergA.txt | grep 08/05-19:03:45 | more
08/05-19:03:45.028697  [**] PING-ICMP Time Exceeded [**] 198.32.248.61 -> MY.NET
.140.9
```

CONCLUSION and *RECOMMENDATIONS*

MY.NET network was the target of numerous reconnaissance efforts and distributed denial of service attacks. The loss of data within the logs made it difficult to confirm all the activity. For example, the activity going to or from 24.25.111.117 may have been contained in the data during the power loss. Also of significance is that the intruder may have acquired access via the telnet session and wiped his fingerprints clean.

Though SYN floods seem to be a very common tactic, they can still be effective with out the proper defense mechanisms in place. Also a Firewall or personal computer that will allow the Happy99 worm in is very susceptible to other more dangerous activity. For example, the Trinoo virus could launch a denial of service within it's own network. I ran a grep against the data base to see if the systems targeted with wintrinoo also had activity during the time span. Nothing correlated within the logs. Updated antivirus software will help malicious activity from virus and Trojan signatures.

The presence of WatchList indicates past history of suspicious activity from Israel and China but we still see successful reconnaissance activity thru firewall.

Site needs to ensure

1. Update ACLs on Firewall to silently drop unauthorized activity,i.e. WatchList IP's.
2. ICMP disabled when not needed.
3. Disable port 1080 unless actually a proxy server.
4. Antivirus software is current.
5. Deinstall Napster and not authorized it for usage.

Assignment 4
Analysis Process

The amount of data that needed to be analyzed appeared to be overwhelming at first. I personally did not have the resources to compile the data. A team of us worked together with the system administrator so the files could be downloaded on a system large enough to handle the data.

We downloaded the files on one computer then accessed the web site www.silicondefense.com/snortsnarf/main.html. From there the necessary tools, snortsnart, was downloaded so each could partition and analyze the data as they seemed fit. The scan files and alert files were merged and then FTP'd to a Unix box, which I am more familiar with.

I then analyzed the data using snortsnarf, looking for common IP's and ports, ect. Checked out the Watchlist and any other alerts I played close attention to.

Once my suspicions list was set, I then searched the data base using the grep command searching for any correlating activity. Crossed referenced date time stamps and IP's. That is how I noticed the telnet Daemon occurring during the Dos attack. That is also how I was able to identify that no other activity from the IP initiating the telnet daemon had been logged into the data base.

Last Updated: July 24th, 2017

Upcoming Training

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