Global Information Assurance Certification Paper

GCIA *PRACTICAL*
WASHINGTON DC 2000

By: Victor H. Maseda

Assignment 1 - Network Detects

Detect No. 1: Network News Transfer Protocol (NNTP) Scan

Type	Date	Time	Source	Destination	Transport
FWIN	12/23/00	14:57:32 -5:00 GMT	24.X.X.X:47277	My.Host.X.X:119	TCP
FWIN	12/23/00	14:57:32 -5:00 GMT	24.X.X.X:47572	My.Host.X.X:119	TCP
FWIN	12/23/00	18:54:36 -5:00 GMT	24.X.X.X:44083	My.Host.X.X:119	TCP
FWIN	12/23/00	18:54:36 -5:00 GMT	24.X.X.X:44623	My.Host.X.X:119	TCP
FWIN	12/23/00	23:02:52 -5:00 GMT	24.X.X.X:60529	My.Host.X.X:119	TCP
FWIN	12/23/00	23:02:52 -5:00 GMT	24.X.X.X:60890	My.Host.X.X:119	TCP
FWIN	12/24/00	10:52:30 -5:00 GMT	24.X.X.X:55891	My.Host.X.X:119	TCP
FWIN	12/24/00	10:52:30 -5:00 GMT	24.X.X.X:56227	My.Host.X.X:119	TCP
FWIN	12/24/00	15:01:00 -5:00 GMT	24.X.X.X:38407	My.Host.X.X:119	TCP
FWIN	12/24/00	15:01:00 -5:00 GMT	24.X.X.X:38871	My.Host.X.X:119	TCP
FWIN	12/24/00	19:05:30 -5:00 GMT	24.X.X.X:56842	My.Host.X.X:119	TCP
FWIN	12/24/00	19:05:32 -5:00 GMT	24.X.X.X:57304	My.Host.X.X:119	TCP
FWIN	12/24/00	23:06:42 -5:00 GMT	24.X.X.X:47262	My.Host.X.X:119	TCP
FWIN	12/24/00	23:06:42 -5:00 GMT	24.X.X.X:47686	My.Host.X.X:119	TCP
FWIN	12/25/00	03:00:26 -5:00 GMT	24.X.X.X:54607	My.Host.X.X:119	TCP
FWIN	12/25/00	03:00:26 -5:00 GMT	24.X.X.X:55068	My.Host.X.X:119	TCP
FWIN	12/25/00	06:56:50 -5:00 GMT	24.X.X.X:50467	My.Host.X.X:119	TCP
FWIN	12/25/00	06:56:50 -5:00 GMT	24.X.X.X:50940	My.Host.X.X:119	TCP
FWIN	12/26/00	11:02:12 -5:00 GMT	24.X.X.X:53475	My.Host.X.X:119	TCP
FWIN	12/26/00	11:02:14 -5:00 GMT	24.X.X.X:54196	My.Host.X.X:119	TCP
FWIN	12/27/00	15:17:46 -5:00 GMT	24.X.X.X:33410	My.Host.X.X:119	TCP
FWIN	12/27/00	15:17:46 -5:00 GMT	24.X.X.X:33990	My.Host.X.X:119	TCP
FWIN	12/27/00	19:24:04 -5:00 GMT	24.X.X.X:49294	My.Host.X.X:119	TCP
FWIN	12/27/00	19:24:04 -5:00 GMT	24.X.X.X:49997	My.Host.X.X:119	TCP
FWIN	12/29/00	11:59:38 -5:00 GMT	24.X.X.X:49550	My.Host.X.X:119	TCP
FWIN	12/29/00	11:59:38 -5:00 GMT	24.X.X.X:49895	My.Host.X.X:119	TCP

ZoneAlarm Basic Logging Client v2.1.44
Windows NT-4.0.1381-Service Pack 6-SP

1. Source of Trace:

My @HOME Internet connection.

2. Detect was Generated by:

ZoneAlarm Logs, a free personal firewall **(**www.zonelabs.com**).**

3. Probability the Source Address was Spoofed:

This looked like part of a reconnaissance process that the individual was taking to determine hosts that respond to the Network News Transfer Protocol (nntp). It is not likely that the source address was spoofed; the attacker would need to see the responses from those hosts running this service. This IP is the source initiator of this reconnaissance/scanning process.

4. Description of Attack:

The source is scanning to locate hosts that are running nntp. After a host running this service is defined, the attacker can exploit all know vulnerabilities or perform denial of service attacks on host using this service (references will be provided in section 6 "Correlations").

5. Attack Mechanism:

The source (24.X.X.X) is scanning my computer and probably the whole entire @HOME network, twice every four hours on a daily base, to track any @HOME computers that might have NNTP running. NNTP is the protocol that runs interference between newsreaders and news servers, it is the means by which USENET news articles are accessed and posted. NNTP web sites allow users to share and access information anonymously, what a great thing to a hacker.

NNTP appeals to hackers not only due to its anonymous access, but also because of the types of vulnerabilities this service has (e.g. allow a remote user to execute commands at root level). Hosts running nntp are very easily comprised making them very attractive hosts to be used as backdoors into protected networks and as zombies or daemons for DDoS attacks. There are

well-documented denial of service attacks that can be rendered on nntp server. References will be provided in section 6 "Correlations".

While doing more investigation on the source address, nslookup return this name, "authorized-scan1.security.home.net". This name triggered more my curiosity, so I place a call to my ISP to find out whether this node was legitimate. This is how they responded:

E-Mail: "Thank you for your report. The IP address 24.X.X.X is a machine we have here in our corporate domain. It is actively scanning our network for security problems in our customers' configuration which could be exploited. We have been seeing a significant number of customers installing networking software which is leaving their computers open to attack, this is why we must respond in this way. You may see it scanning 2-3 times a day. This proactive scanning is allowed by the Acceptable Use Policy, which can be reviewed at the following URL …"

6. Correlations:

CERT(*) Summary CS-97.02 INN (InterNetNews) is a commonly used software program for serving and managing news according to the NNTP protocol. In versions of INN prior to 1.5.1, this vulnerability allows remote users to execute arbitrary commands on the news server with the same privileges as the user-id that manages the news server

CERT® Advisory CA-1997-08 INN vulnerabilities allow unauthorized users to execute arbitrary commands on the machine running INN by sending a maliciously formed news control message. Because the problem is with the content of news control messages, attacks can be launched remotely and may reach news servers located behind Internet firewalls.

SecurityFocus Advisory 2186 Remote DoS attack in CASSANDRA NNTPServer v1.10

7. Evidence of Active Targeting:

Yes, but this is a legitimate scan of the whole network performed by my ISP in an effort to detect nntp hosts which might be running the risk of being comprised and compromising the security status of the @HOME network.

8. Severity: [Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)]

Criticality: 3	My personal computer which has some critical data.
Lethality: 1	This is a legit scan performed by my ISP and my node is not running this service
System Countermeasures: 5	OS has latest patches and is running a personal firewall
Network Countermeasures: 4	Firewall protected and ISP currently scanning for existing nntp servers

Severity = (3 + 1) – (5 + 4) = -5
Low: Non-Hostile scan.

9. Defense Recommendation:

A good defense mechanism would be what my ISP is doing; keeping a close look at the users that might be installing/configuring NNTP server, and educating the users of this protocol.

In the case of a company's networks, if you must run a news group (NNTP server), I would recommend to proxy NNTP by setting up a bastion host on a DMZ that talks to external sites, sanitizing information in the process and relaying news to an internal NNTP server. Your internal newsreaders should only be able to read news at the internal NNTP server. A host IDS on the external NNTP server should be considered to keep a closer watch on activities using this protocol.

Use the latest versions and patches available of software's that use this protocol.

10. Multiple Choice Test Question:

Which port does the Network News Protocol use:
(A) 111
(B) 112
(C) 119
(D) 143
Answer is: C

Detect No. 2 : Subseven (S7S) Scans

Type	Date	Time	Source	Destination	Transport
FWIN	12/23/00	22:46:14 -5:00 GMT	65.34.36.78:2357	My.Host.X.X:27374	TCP
FWIN	12/29/00	00:28:42 -5:00 GMT	Bad.Guy.X.X:21899*	My.Host.X.X:6667	TCP
FWIN	12/29/00	00:50:44 -5:00 GMT	Bad.Guy.X.X:23387*	My.Host.X.X:1243	TCP
FWIN	12/30/00	16:59:52 -5:00 GMT	Bad.Guy.X.X:25030*	My.Host.X.X:27374	TCP
FWIN	1/1/01	14:06:58 -5:00 GMT	24.161.100.162:3788	My.Host.X.X:27374	TCP
FWIN	1/1/01	14:46:46 -5:00 GMT	194.230.137.17:1633	My.Host.X.X:1243	TCP
FWIN	1/4/01	19:48:04 -5:00 GMT	24.6.254.168:1555*	My.Host.X.X:27374	TCP
FWIN	1/4/01	21:00:52 -5:00 GMT	24.66.114.184:2055*	My.Host.X.X:27374	TCP
FWIN	1/4/01	23:19:46 -5:00 GMT	24.17.14.109:3074*	My.Host.X.X:27374	TCP
FWIN	1/8/01	14:31:24 -5:00 GMT	216.77.215.26:1581	My.Host.X.X:27374	TCP
FWIN	1/8/01	16:12:32 -5:00 GMT	209.214.83.170:1113	My.Host.X.X:27374	TCP
FWIN	1/8/01	20:36:06 -5:00 GMT	142.163.72.48:3765	My.Host.X.X:1243	TCP
FWIN	1/11/01	02:04:06 -5:00 GMT	24.68.143.133:2297	My.Host.X.X:27374	TCP
FWIN	1/14/01	15:42:34 -5:00 GMT	194.230.171.66:1092	My.Host.X.X:1243	TCP

ZoneAlarm Basic Logging Client v2.1.44
Windows NT-4.0.1381-Service Pack 6-SP

*** @HOME User**

1. **Source of Trace:**

   My @HOME Internet connection.

2. **Detect was Generated by:**

   ZoneAlarm Logs, a free personal firewall **(**www.zonelabs.com**).**

3. Probability the Source Address was Spoofed:

No, it is unlikely that the source address(es) were spoofed. The sources have initiated a scan in order to find any/all hosts that have been compromised by the subseven Trojan. This is one possibility, but due to Subseven's updated capabilities, there is now a slim probability that the scans (performed by the source(s)) could be responses triggered by Subseven client's stimulus. In another words, these nodes that are scanning for Subseven servers could themselves be compromised nodes (S7S).

4. Description of Attack:

Subseven Trojan ports scan.

5. Attack Mechanism:

Developed in the Netherlands by "Mobman", the Subseven trojan (also known as Backdoor-G) has become more popular than any other trojan alike: Netbus, Back Orifice, and BO2k. This client/server application's popularity is attributed to its stability, capabilities, and configurability. The Subseven server can be installed on a victim's node (default port 27374, but configurable), via a disguised email attachment (file, jpg, bmp), or hidden within software that is downloaded from the Internet. Hosts comprised with the Subseven trojan (Windows 9x/NT node only) can be remotely controlled via a Subseven client (GUI interface).

To illustrate the popularity of the Subseven scans, I've included all source IPs, logged by my firewall, which were scanning for S7Ss. I wanted to specifically point out **Bad.Guy.X.X** (another @HOME subscriber**)** because this source is probing for various Subseven servers/commonly used S7S ports (version 1.X (port 1243), version 2.X (default port 27374), and Internet Relay Chat Servers (port TCP 6667) which are used to share S7Ss among attackers). S7S's Internet Relay Chat (IRC) connection feature can be used to specify an IRC server to channel S7S's data (IP address, listening port, and password) to participants in the channel. S7S can act as a standard IRC robot ("bot"), which issue channel commands.

After the scans are performed and S7Ss are identified/discovered, the Subseven client gives an attacker fairly complete control of the compromised host (Windows 9x/NT). Some of the capabilities provided by the Subseven server (S7S) to the client are the ability to: launch port scans from the compromised system (S7S), orchestrate DDoS, start and FTP server rooted at c:\ with full read/write permissions, remotely edit the registry, retrieve passwords (RAS, ICQ), application and port redirection, capture screenshots, restart the system, log keystrokes, run remote terminal (listens on port 7215 by default), hijack the mouse,

remotely spy on applications/Web cams (ICQ, AOL, Instant, MSN, and Yahoo Messenger), open a browser, accesses user-define sites, obtain cached passwords, print, and notify attackers of successful compromises via ICQ and e-mail.

I reported my analysis to my ISP. About a month later, from my submitted report, I was notified that "appropiate actions were taken against this account" (**Bad.Guy.X.X)**.

6. Correlations:

http://subseven.slak.org Subseven Homepage.

http://vil.nai.com/villib/dispVirus.asp?virus_k=10566 Trojan which has been consistently updated by the author.

http://vil.mcafee.com/dispVirus.asp?virus_k=10171& McAfee Antivirus software

http://advice.networkice.com/Advice/Phauna/RATs/SubSeven/default.htm The most popular Trojan (as measured in scans for this trojan).

7. Evidence of Active Targeting:

Yes, the @HOME network is the target of the scan(s).

8. Severity: [Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)]

Criticality: 3	My personal computer which has some critical data
Lethality: 1	Attack is very unlikely to succeed
System Countermeasures: 4	OS is running an anti-virus software with the latest signatures, and a personal firewall
Network Countermeasures: 5	IDSs with subseven filters are in place

Severity = (3 + 1) – (4 + 5) = -5
Low: It's a scan. The destination host has not been compromised.

9. Defense Recommendation:

Continue to update NIDS filters and firewall rules to detect and block, respectively, subseven port trends/newly configured S7S ports, update anti-virus signatures, and do not allow users to download applications from the web (educate users on the risks of downloading shareware from the web).

10. Multiple Choice Test Question:

Which of the following scan is a Subseven scan?

(1) 18:12:42.455414 172.100.15.2:2346 -> X.X.X.X:27374 SYN **S*****
18:12:42.455414 172.100.15.2:2346 -> X.X.X.X:27374 SYN **S*****

(2) 18:12:42.455414 172.100.15.2:2346 -> X.X.X.X:1243 SYN **S*****
18:12:42.455414 172.100.15.2:2346 -> X.X.X.X:1243 SYN **S*****

(3) 18:12:42.455414 172.100.15.2:2346 -> X.X.X.X:1234 SYN **S*****
18:12:42.455414 172.100.15.2:2346 -> X.X.X.X:1234 SYN **S*****

(4) 18:12:42.455414 172.100.15.2:2346 -> X.X.X.X:27473 SYN **S*****
18:12:42.455414 172.100.15.2:2346 -> X.X.X.X:27473 SYN **S*****

(A) 1 and 2
(B) 1 and 3
(C) 1 and 4
(D) 4 and 2
(E) All of the Above
Answer is: A

Detect No. 3: FTP to FTP SYN-FIN Scan

```
[**] IDS198/SYN FIN Scan [**]
01/17-22:25:39.269443 130.192.10.133:21 -> Internal.Net.Host.5:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0xDDC5D16   Ack: 0x40D8C654   Win: 0x404

[**] IDS198/SYN FIN Scan [**]
01/17-22:25:39.368817 130.192.10.133:21 -> Internal.Net.Host.6:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0xDDC5D16   Ack: 0x40D8C654   Win: 0x404

[**] IDS198/SYN FIN Scan [**]
01/17-22:25:39.468093 130.192.10.133:21 -> Internal.Net.Host.7:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0xDDC5D16   Ack: 0x40D8C654   Win: 0x404

[**] IDS198/SYN FIN Scan [**]
01/17-22:25:39.675330 130.192.10.133:21 -> Internal.Net.Host.8:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0xDDC5D16   Ack: 0x40D8C654   Win: 0x404

[**] IDS198/SYN FIN Scan [**]
01/17-22:25:39.769687 130.192.10.133:21 -> Internal.Net.Host.9:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0xDDC5D16   Ack: 0x40D8C654   Win: 0x404

[**] IDS198/SYN FIN Scan [**]
01/17-22:25:39.770192 130.192.10.133:21 -> Internal.Net.Host.10:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0xDDC5D16   Ack: 0x40D8C654   Win: 0x404

[**] IDS198/SYN FIN Scan [**]
01/17-22:25:39.868210 130.192.10.133:21 -> Internal.Net.Host.11:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0xDDC5D16   Ack: 0x40D8C654   Win: 0x404

[**] IDS198/SYN FIN Scan [**]
01/17-22:25:39.980361 130.192.10.133:21 -> Internal.Net.Host.12:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0xDDC5D16   Ack: 0x40D8C654   Win: 0x404
```

1. Source of Trace:

This trace was captured from a frien's network.

2. Detect was Generated by:

SNORT Intrusion Detection System located in the DMZ.

3. Probability the Source Address was Spoofed:

The probability that the source address is spoofed is high. Tools like Idlescan (to perform IP ID Spoof scanning), S7S (that provide the means for an attacker to use a compromised node to perform the actual scan on his/her behave), or a mediator technique (to perform a masqueraded scan), could have been utilized to disguise the legitimate initiator of the scan.

4. Description of Attack:

This is a scan looking for FTP servers.

5. Attack Mechanism:

Logged on January 17th, 2001, at 10:25pm this scan targeted 8 nodes within the "Internal.Network.IPBlock" for FTP server services (**TCP port 21**). These nodes were scanned using SYN-FIN flag combinations (non-SYN-ACK-RST or "impossible packets"). Due to the presence of the FIN flag (connection termination) in this TCP flag combination, some logging devices will not record the targeted service, making this scan stealth in nature (this could be the case in networks that do not have an IDS watching). At the same time, the presence of the FIN flag will evade "Established" ACL on filtering devices. Hardwired source/destination port, and the same Sequence (**Seq: 0xDDC5D16)**, ACK (**Ack: 0x40D8C654)**, and IP ID (**ID:39426)** numbers, are signatures that this scan was performed with crafted packets. The source of the scan **130.192.10.133** is an IP of a node from a polytechnic institute in Torino, Italy. Knowing how easy it is to access university nodes, it would not be far fetched to presume that the source (**130.192.10.133)** has been either compromised and is being "remote controlled" via a Trojan to perform various devious activities (scans, DoS, DDoS, etc) or is being locally accessed/utilized to test downloaded scanning and other tools. Scenario number three would be that the node is being accessed (locally) to collect data as part of a reconnaissance process.

Although this scan could have been part of the reconnaissance process to identify and compromise FTP servers (for scanning and penetrating protected networks (i.e. FTP Bounce Back)), it is unlikely that this was the intent. The IP addressed of the existing FTP servers are out of the range that was scanned, and no other IPs were probed.

6. Correlations:

http://www.securityfocus.com/tools/679 Idlescan, a portscanning tool that uses IP ID spoofing scanning techniques.

Idelscan References Other Idlescan resources

http://subseven.slak.org Subseven Home Page.

http://www.cert.org/tech_tips/ftp_port_attacks.html#3 FTP Bounce Attack

http://www.whitehats.com/info/IDS198 SYN-FIN an "impossible" packet used for probing

7. Evidence of Active Targeting:

The block of IPs that was scanned did not include any of the FTP servers, nor did we see any other traffic activity targeting our network with the abovementioned signatures. I would not perceive this traffic as "active targeting" our network, the reconnaissance process is neither extensive nor detailed/precise enough to categorize it as such. This is probably a script kiddy or a student trying out a downloaded scanning tool.

8. Severity: [Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)]

Criticality: 2	Regular network users not running the FTP server service.
Lethality: 1	An FTP attack is very unlikely to succeed on these nodes.
System Countermeasures: 5	OS that were scanned do not have the FTP server service running.
Network Countermeasures: 5	Network has an IDS with filters that flag any bogus TCP flag combination.

Severity = (2 + 1) – (5 + 5) = -7
Low: It's a scan that covers a small range of nodes that are not running FTP server services.

9. Defense Recommendation:

There is really no need to take any action for this scan because an IDS is in place to detect "impossible packets" and because there is no real targeting done by this scan.

10. Multiple Choice Test Question:

What TCP flag combinations are considered non-SYN-ACK-RST packets

(A) F (Only)
(B) No Flags set
(C) SF and SR
(D) RF
(E) A and C
(F) All of the above
Answer is: F

Detect No. 4: Stacheldraht Scan

```
[**] IDS194 - DDoS - Stacheldraht client-check-gag [**]
01/16-18:49:45.216574 0:50:73:43:F9:80 -> 8:0:20:79:4A:50 type:0x800 len:0x3C
193.166.141.176 -> Prot.Net.Host.175 ICMP TTL:110 TOS:0x0 ID:62273
ID:39938   Seq:0  ECHO REPLY

[**] IDS194 - DDoS - Stacheldraht client-check-gag [**]
01/16-18:49:45.218958 0:50:73:43:F9:80 -> 8:0:20:79:4A:50 type:0x800 len:0x3C
193.166.141.176 -> Prot.Net.Host.176 ICMP TTL:110 TOS:0x0 ID:63041
ID:39938   Seq:0  ECHO REPLY

[**] IDS194 - DDoS - Stacheldraht client-check-gag [**]
01/16-18:49:45.221082 0:50:73:43:F9:80 -> 8:0:20:79:4A:50 type:0x800 len:0x3C
193.166.141.176 -> Prot.Net.Host.177 ICMP TTL:110 TOS:0x0 ID:63809
ID:39938   Seq:0  ECHO REPLY

[**] IDS194 - DDoS - Stacheldraht client-check-gag [**]
01/16-18:49:45.223116 0:50:73:43:F9:80 -> 8:0:20:79:4A:50 type:0x800 len:0x3C
193.166.141.176 -> Prot.Net.Host.178 ICMP TTL:110 TOS:0x0 ID:64577
ID:39938   Seq:0  ECHO REPLY

[**] spp_portscan: PORTSCAN DETECTED from 193.166.141.176 (THRESHOLD 3 connections exceeded in 0 seconds)
```

1. **Source of Trace:**

   This trace was captured from a friend's network
2. **Detect was Generated by:**

   SNORT Intrusion Detection System located in the DMZ.

3. Probability the Source Address was Spoofed:

There is a possibility that the source IP is not the actual originator of the Stacheldraht scan. This source could very well be a compromised host. More investigation is required on this host.

4. Description of Attack:

This is another scan, this time looking for compromised systems (agents).

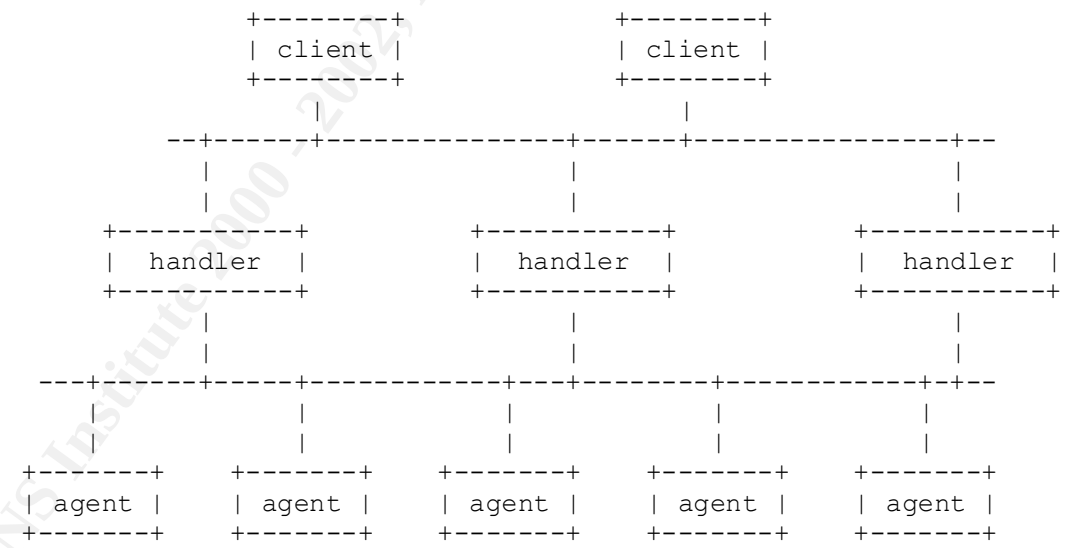
5. Attack Mechanism:

The following is an analysis of "stacheldraht", a distributed denial of service attack tool, based on source code from the "Tribe Flood Network" distributed denial of service attack tool. Stacheldraht (German for "barbed wire") combines features of the "trinoo" distributed denial of service tool, with those of the original TFN, and adds encryption of communication between the attacker and stacheldraht handler and automated update of the agents (http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt). Similar to TFN, Stacheldraht attacks with ICMP-. UDP-, SYN-, and Smurf-type attacks. To communicate between the handler and the agents, Stacheldraht uses a combination of TCP and ICMP (ECHO reply) packets.

Make special note to the sequence and IP ID of these Stacheldraht packets. Stacheldraht uses the ID fields and data fields to communicate between its participants (client(s),handler(s), and agent(s)).

This is the only activity that we saw from this host in the logs that were provided. There is a possibility that this source is scanning 5 hosts at a time for Stacheldraht agents. Although the source **193.166.141.176** is just scanning for agents this activity should be treated with great concern due to its final intended purpose (DdoS).

The stacheldraht network is made up of one or more handler programs("mserv.c") and a large set of agents ("leaf/td.c"). The attacker uses an encrypting "telnet alike" program to connect to and communicate with the handlers ("telnetc/client.c"). A stacheldraht network would look like this:



The network: client(s)-->handler(s)-->agent(s)-->victim(s)

The attacker(s) control one or more handlers using encrypting clients. Each handler can control many agents. (There is an internal limit in the "mserv.c" code to 1000 agents. This is most likely to ensure the number of open file handles, commonly 1024, is not exceeded by the program. Thanks to Adam C. Greenfield for pointing this out. Besides, the code says that "1000 sockets are leet0.") The agents are all instructed to coordinate a packet based attack against one or more victim systems by the handler (referred to as an "mserver" or "master server" in the code.)
http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt

6. Correlations:

http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt *Stacheldraht*

http://www.cert.org/incident_notes/IN-99-04.html CERT® Incident Note IN-99-04

http://staff.washington.edu/dittrich/misc/trinoo.analysis Trinoo

http://staff.washington.edu/dittrich/misc/tfn.analysis TFN

7. Evidence of Active Targeting:

Not enough data was provided to conclude such activity/intent.

8. Severity: [Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)]

Criticality: 3	Only 5 Hosts were scanned.
Lethality: 2	Users have the latest OS Hot Fixes and Patches.
System Countermeasures: 3	OS that were scanned apparently have not been compromised
Network Countermeasures: 1	Network has an IDS that detects some of Stacheldraht's signatures. Firewall has High TCP-Ports open.

Severity = (3 + 2) – (3 + 1) = 1

Low: Even though it's a scan and because of the lack of data to confirm whether this is on going reconnaissance, I would categorize its severity Low-Medium. Stacheldraht is not a weapon to be taken lightly.

9. Defense Recommendation:

Tighten up the network's perimeter and create more filters to detect new Stacheldraht signatures.

If possible block **93.166.141.176** and other sources, performing such scans, on your network (via packet filtering device/firewall).

Disallow all ICMP traffic inbound to your network.

Keep nodes up to date (install latest patches and Hot Fixes).

Set properly file/directory permissions.

To protect from *Stacheldraht agens (*zombie) attacks employ a sort of rate filtering at your border routers, such as ICMP rate filtering to limit ICMP and Smurf attacks.

10. Multiple Choice Test Question:

Which DDoS tool provides encryption between client and server communications:

(A) Stacheldraht
(B) TFN
(C) Trinoo
(D) SubSeven
(E) A and D
(F) All of the Above
Answer: (E) For now ;-)

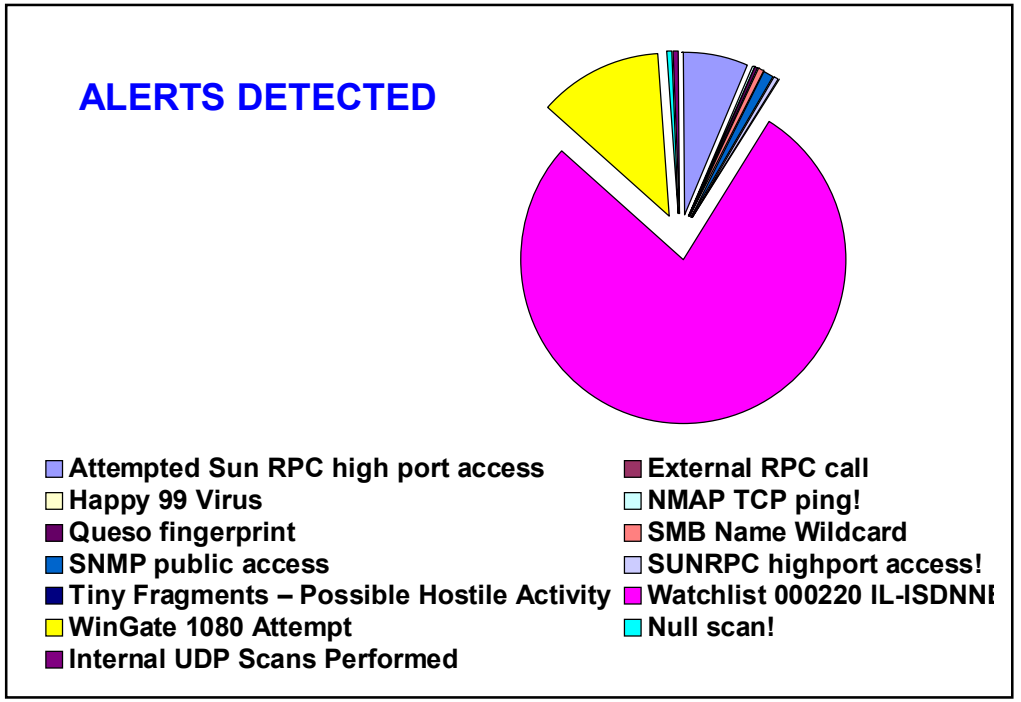
Assignment 2 – "Analyze This" Scenario

SCENARIO

Our organization has been asked by GIAC Enterprises to put together a bid for security services. Almost one-month worth of data from a Snort IDS has been provided by GIAC for our analysis. After processing the SNORT logs and analyzing the provided traffic we were able to draw the following conclusion about the "Fortune Message" private network's (MY.NET) security posture.

Analysis Performed:

Alerts Type	Alerts Count
Attempted Sun RPC high port access	2,542
External RPC call	13
Happy 99 Virus	2
NMAP TCP ping!	96
Queso fingerprint	142
SMB Name Wildcard	218
SNMP public access	468
SUNRPC highport access!	60
Tiny Fragments – Possible Hostile Activity	7
Watchlist 000220 IL-ISDNNET-990517	30,998
WinGate 1080 Attempt	4,802
Null scan!	283
Internal UDP Scans Performed	200
MY.NET Scan Summary	Summany



Files provided for analysis:

Alert Files = 54
Scan Files = 42
OOS Files = 19

Specific Traffic Activity:

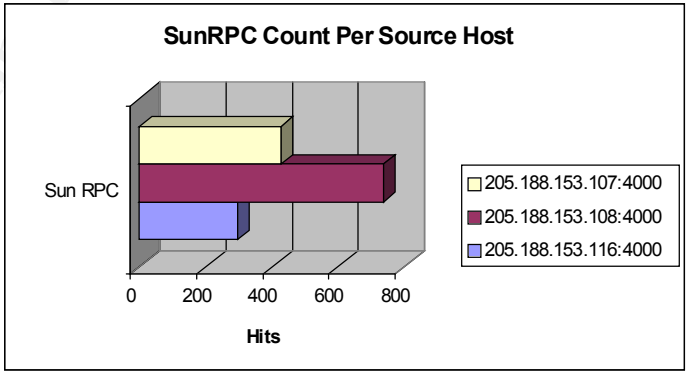
Attempted Sun RPC high port access – Emphasizing the importance of this traffic, here is a quote from the System Administration, Networking, and Security (SANS) web site. "Remote procedure calls (RPC) allow programs on one computer to execute programs on a second computer. They are widely used to access network services such as shared files in NFS. Multiple vulnerabilities caused by flaws in RPC, are being actively exploited. There is compelling evidence that the vast majority of the distributed denial of service attacks launched during 1999 and early 2000 were executed by systems that had been victimized because they had the RPC vulnerabilities. The broadly successful attack on U.S. military systems during the Solar Sunrise incident also exploited an RPC flaw found on hundreds of Department of Defense systems".

These are the "Top 10" IP Sources that attempted Sun RPC high port access between the observed period; September 26 through November 22, 2000.

Date	Source	Target	Number of Sun RPC hits
10/03	205.188.153.116:4000	MY.NET.225.210:32771	117
10/04	205.188.153.116:4000	MY.NET.225.210:32771	182
10/12	205.188.153.107:4000	MY.NET.226.74:32771	120
10/24	205.188.153.107:4000	MY.NET.217.214:32771	178
10/25	205.188.153.107:4000	MY.NET.217.214:32771	127
11/04	205.188.153.108:4000	MY.NET.221.246:32771	189
11/05	205.188.153.108:4000	MY.NET.221.246:32771	156
11/06	205.188.153.108:4000	MY.NET.221.246:32771	112
11/07	205.188.153.109:4000	MY.NET.222.98:32771	153
11/14	205.188.153.109:4000	MY.NET.206.222:32771	128

ATTEMPTED SUNRPC High Port Access Source IP:

America Online, Inc (NETBLK-AOL-DTC)
22080 Pacific Blvd
Sterling, VA 20166
US
Netblock: 205.188.0.0 - 205.188.255.255



The **IP range 205.188.0.0 – 205.188.255.255 belongs to America Online Internet Service Providers**. These IP addresses have been seen throughout the log attempting High port access to MY.NET. AOL runs **ICQ usually on port 4000 or higher**, therefore this **data may actually be a false positive** and may be a result of employees chatting on ICQ servers. Further investigation on the targets is required to determine traffic legitimacy.

External Procedure Call – As described before there are vulnerabilities involved with the RPC services for Unix operating Systems. On Solaris 2.x operating systems, rpcbind listens on TCP port 111, and UDP port 111. Rpcbind permits a remote attacker to insert and delete entries without super user status by spoofing a source address. Ironically, it inserts the entries as being owned by super user. Among the IP address listed below, **200.191.80.206** was **able to connect to port 111**, this source could have been able to compromise the targeted system.

Date	Source	Target	Number of External Proc hits
10/10	200.191.80.181:634	MY.NET.6.15:111	1
10/10	200.191.80.181:665	MY.NET.6.15:111	1
10/10	200.191.80.206:931	MY.NET.6.15:111	2
10/11	63.162.239.69:3655	MY.NET.15.127:111	1
10/11	63.162.239.69:4496	MY.NET.100.130:111	1
10/11	63.162.239.69:975	MY.NET.6.15:111	1
10/14	24.23.151.112:3306	MY.NET.100.130:111	1
10/23	24.7.227.215:5	MY.NET.6.15:111	1
10/28	12.34.21.196:700	MY.NET.6.15:111	1
11/01	38.200.223.8:2473	MY.NET.6.15:111	1
11/10	211.46.110.81:4910	MY.NET.100.130:111	1
11/10	211.46.110.81:708	MY.NET.6.15:111	1

EXTERNAL PROCEDURE CALL Source IP:
RNP (Brazilian Research Network) (NETBLK-BRAZIL-BLK2). These addresses have been further assigned to Brazilian users.

Netname: BRAZIL-BLK2
Netblock: 200.128.0.0 - 200.255.255.255
Maintainer: RNP

Happy99 Virus – The Happy99 Virus is a Virus that is attached to email and if the recipient opens the email the Happy99 virus infects the computer system. Then if the recipient sends additional email to other, the virus spreads. There are a number of infected computers sending unsuspecting people an email attachment, which bears the name Happy99.exe or Trojan.exe. If the hopeless recipient opens the .exe file they will see a brief fireworks display heralding the infection of their computer; Symantec calls a worm/virus. And this is bad news for the owner of the computer and bad for those he emails for they will be sent the virus file without knowledge of the sender. This results in more infections. One of the several things this insidious worm is doing is using the WSOCK32.DLL to spread the infection. One must go to Windows, System, and delete SKA.EXE, and SKA.DLL and replace WSOCK32.DLL with SOCK32.SKA. Further instructions may be had via the manufacturer of your virus program.

You can see below that the snort log picked up on the computer virus and alerted on it, but the connections were probably made to **MY.NET.253.41** and **MY.NET.6.35 on port 25** (SMTP – Simple Mail Transport Protocol). If the virus was sent and the recipient opened the email, the system is now infected with the virus.

Date	Source	Target	Number of Happy99 hits
10/05	216.6.117.11:41827	MY.NET.253.41:25	1
11/06	209.94.224.13:2708	MY.NET.6.35:25	1

HAPPY99 Source IP:
Hostname: NS2.HYPERIA.COM
System: ? running ?
(This is an African (Nigerian) ISP)
Record last updated on 07-Oct-2000.
Database last updated on 19-Feb-2001 18:27:46 EDT.

Address: 216.6.117.11

HAPPY99 Source IP:
Reaction Systems, Inc. (NETBLK-REACTION97)
Village Online Div.
1210 Hamblen Rd.
Kingwood, TX 77339
US

Netblock: 209.94.224.0 - 209.94.255.255

NMAP TCP Ping – NMAP is a network-scanning tool that allows the attacker to craft TCP packets to bypass firewalls and make the scanning stealthy. By crafting TCP packets the attacker initiates a TCP communication using Flags set that are unusual (i.e. SYN–FIN flags set together, or all the flag bits sets (Reserved 2, Reserves 1,URG, ACK, PSH, RST, SYN, and FIN (non-SYN-ACK-RST packets)). This flexibility allows an attacker not only scan the network (for existing hosts) or hosts (for existing services), in a stealth matter (bypassing firewalls that are looking for certain TCP flags set), but also provides the means to perform OS finger printing.

Traffic from **192.102.197.234** should be further analyzed and/or blocked (if this is not a recognized source). There should be no reason why an external user is trying to access the MY.NET internal DNS server. This source appears to be performing reconnaissance on the internal DNS, a quick and dirty way to map the whole "MY.NET" network.

Date	Source	Target	Number of NMAP TCP hits
10/23	205.128.11.157:53	MY.NET.1.9:53	2
10/29	192.102.197.234:53	MY.NET.1.8:53	2
10/30	192.102.197.234:80	MY.NET.1.8:53	3
10/31	192.102.197.234:53	MY.NET.1.8:53	2
11/06	192.102.197.234:80	MY.NET.1.8:53	2
11/07	192.102.197.234:80	MY.NET.1.8:53	2
11/08	192.102.197.234:80	MY.NET.1.8:53	2
11/09	192.102.197.234:53	MY.NET.1.8:53	4
11/16	192.102.197.234:80	MY.NET.1.8:53	3
11/19	192.102.197.234:53	MY.NET.1.8:53	3
11/20	192.102.197.234:53	MY.NET.1.8:53	2
11/20	192.102.197.234:80	MY.NET.1.8:53	2

NMAP TCP Ping:
Intel Corporation (NET-LOCALNET16)
Corporate Information Services
1900 Prairie City Road, FM1-56
Folsom,CA 95670

Netblock: 192.102.197.0 - 192.102.197.255

QUESO Finger PRINTING – Most hit hosts.

Date	Source	Target	Number of Queso hits
09/27	128.253.247.116:2720	MY.NET.227.10:4053	3
10/21	24.3.161.193:33236	MY.NET.145.9:110	2

QUESO FINGER PRINTING:
Cornell University (NET-CCS-NET)
Cornell Information Technologies
Network Resources
143 Caldwell Hall
Ithaca, NY 14853

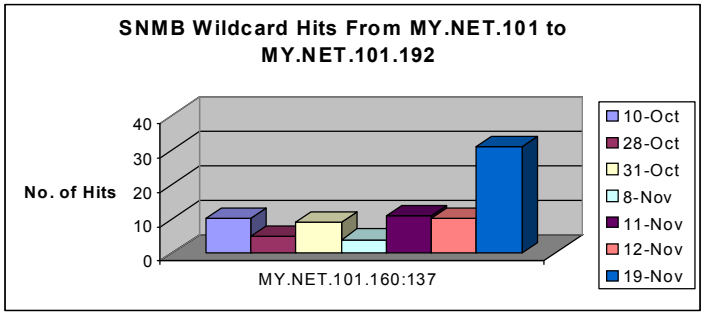
Netblock: 128.253.0.0 - 128.253.255.255

SMB Name Wildcard – This SMB Wildcard is a Netbios Name query and this probe is a prelude to an SMB connection. Packets sent to UDP port 137 from port 137 are extremely common and rarely indicate an attack. Within a windows network there is a definite pattern to these connections, especially as some are accompanied by other scans. As one can see, the SMB Netbios Name Query comes from inside, **MY.NET.101.160** (this source created the most SMB Name Wildcard alerts), and **could be either a compromised system or an employee with bad intentions**. In either case this would require further investigation.

Date	Source	Target	Number of SMB Name hits
10/01	MY.NET.101.160:137	MY.NET.101.192:137	7
10/08	129.37.159.177:137	MY.NET.100.130:137	4
10/10	MY.NET.101.160:137	MY.NET.101.192:137	10
10/28	MY.NET.101.160:137	MY.NET.101.192:137	5
10/31	MY.NET.101.160:137	MY.NET.101.192:137	9
11/08	MY.NET.101.160:137	MY.NET.101.192:137	4
11/11	MY.NET.101.160:137	MY.NET.101.192:137	11
11/12	MY.NET.101.160:137	MY.NET.101.192:137	10
11/19	MY.NET.101.160:137	MY.NET.101.192:137	31
11/20	141.157.99.21:137	MY.NET.6.15:137	33
11/22	141.157.98.201:137	MY.NET.6.15:137	20

SMB Name Wildcard Source IP:
Bell Atlantic (NETBLK-BELL-ATLANTIC)
1880 Campus Commons Drive
Reston, VA 20191 US

Netblock: 141.149.0.0 - 141.158.255.255



SNMP Public Access – Simple Network Management Protocol allows connections to SNMP with the default string of public. Though malicious SNMP scanning does exist (it can identify "open" HP hubs and printers for one thing) there are many cases of software sending out SNMP probes in the natural course of events (programs which use SNMP as one tool to attempt to map out a network via SNMP, printer drivers attempting to browse and probe for HP printers to list for users wishing to select a printer, network management stations 'discovering' managed objects with SNMP agents and associated MIBs, etc). Browsing remote SNMP MIBs you can often determine the remote system type, OS level and other useful information when managing and doing an inventory of your network (of course in the wrong hands that info can be used against you). If the attacker can get the "write" community string he/she can take control of the box.
Here is some traffic that requires investigation (**SNMP connections coming from MY.NET destine to MY.NET**). Further analyses needs to be performed on this nodes to determine whether or not these MY.NET sources have been compromised. Most likely some have been compromised.

Date	Source	Target	Number of SNMP Pulbic hits
10/01	MY.NET.97.171:1172	MY.NET.101.192:161	2
10/01	MY.NET.97.171:1173	MY.NET.101.192:161	2
10/01	MY.NET.97.171:1217	MY.NET.101.192:161	2
10/01	MY.NET.97.171:1240	MY.NET.101.192:161	2
10/01	MY.NET.97.171:1247	MY.NET.101.192:161	2
10/28	MY.NET.97.178:1173	MY.NET.101.192:161	2
10/28	MY.NET.97.178:1309	MY.NET.101.192:161	2
10/31	MY.NET.98.174:1514	MY.NET.101.192:161	2
10/31	MY.NET.98.174:1627	MY.NET.101.192:161	2
11/08	MY.NET.97.130:1210	MY.NET.101.192:161	2

SUNRPC high port access – These alerts should be of concern. The listed IP addresses below (**216.10.12.30** and **216.148.218.160**) running **an unknown service (port 2078)** and **http protocol over TLS SSL (443)** **respectively**, were able to connect to a high port and depending on what services may have been listening, may have been able to compromise the MY.NET computer system with known vulnerabilities. On some OSs, for example, Solaris 2.x operating systems, rpcbind listens not only on TCP port 111, and UDP port 111, but also on a port greater than 32770. This results in a large number of packet filters, which intend to block access to rpcbind/portmapper, being ineffective. Instead of sending requests to TCP or UDP port 111, the attacker simply sends them to a UDP port greater than 32770 on which rpcbind is listening.

Date	Source	Target	Number of SUNRPC Access hits
10/03	205.188.3.211:2089	MY.NET.212.186:32771	4
10/03	216.10.12.30:2078	MY.NET.202.242:32771	3
10/14	195.34.28.117:3364	MY.NET.97.59:32771	2
10/16	216.10.12.30:2078	MY.NET.202.242:32771	5
10/19	205.188.3.239:5190	MY.NET.228.62:32771	3
10/19	216.10.12.30:2078	MY.NET.202.242:32771	10
10/26	216.148.218.160:443	MY.NET.206.222:32771	2
11/01	216.148.218.160:443	MY.NET.206.222:32771	2
11/10	216.148.218.160:443	MY.NET.206.222:32771	2
11/17	205.188.4.2:5190	MY.NET.53.23:32771	2
11/20	216.10.12.30:2078	MY.NET.206.222:32771	7
11/21	216.10.12.30:2078	MY.NET.206.222:32771	7

SUNRPC High Port Access Source IP:
Virtual Development Inc (NETBLK-VDI)
1373 Broad Street, Suite 306
Clifton, NJ 07013 US

Netblock: 216.10.0.0 - 216.10.31.255

SUNRPC High Port Access Source IP:
TCG CERFnet (NETBLK-CERFNET-BLK-4)
P.O. Box 919014
San Diego, CA 92191-9014 US

Netblock: 216.148.0.0 - 216.148.255.255

Tiny Fragments – TCP fragmenting is a method to obscure scanning implementations (network or host) by splitting the TCP header into smaller fragments. IP reassembly on the server or host side can often lead to unpredictable/abnormal results. Many hosts are unable to parse and reassemble the tiny packets and thus may cause crashes, reboots, or even network device monitoring dumps. Fragmentation is able to bypass firewalls and intrusion detection systems, and is used for reconnaissance.

A normal TCP header is 20 bytes or greater in length. "Tiny Fragments" are fragmented TPC headers that are less than 20 bytes so that these packets will bypass firewalls or intrusion detection systems un-inspected.

Date	Source	Target	Number of Tiny Frag hits
09/26	172.157.126.93	MY.NET.201.2	1
09/28	216.43.55.44	MY.NET.211.2	1
09/30	216.43.55.44	MY.NET.202.102	1
10/08	62.6.71.0	MY.NET.181.144	2
10/19	192.206.151.152	MY.NET.1.8	1
11/16	202.156.51.76	MY.NET.201.198	1

TINY FRAGMENTS Source IP:

McLeodUSA Incorporated
6400 C Street SW, PO Box 3177
Cedar Rapids, IA 52406 US

Netblock: 216.43.0.0 - 216.43.255.255

TINY FRAGMENTS Source IP:

BT Public Internet Service
descr: BT-IMS-net
country: GB

inetnum: 62.6.64.0 - 62.6.95.255

Watchlist – The Watchlist contains certain IP addresses that may be of questionable character (based on past experience of other Intrusion Detection personnel experiences). There were **228 alerts found on Watchlist IP addresses** between September 26 and November 22, 2000. All Top 10 Watchlist **IP addresses listed below were all from Israel**. The assortment of scans from China and Israel were extremely extensive, probing most all well-known ports. There is a possibility that a connection was established with the Israel host throughout the monitored period. As can be seen in the dimensions listed below, various hosts on MY.NET have been targeted. A more thorough examination of the targeted hosts is required for verification.

Date	Source	Target	Number of Watchlist hits
10/08	212.179.44.115:1057	MY.NET.223.98:6699	1239
10/08	212.179.44.115:1067	MY.NET.223.98:6699	2699
10/13	212.179.41.24:1031	MY.NET.214.170:6699	1353
10/14	212.179.45.81:1167	MY.NET.202.22:6699	950
11/05	212.179.95.5:1192	MY.NET.211.146:4922	1344
11/05	212.179.95.5:1574	MY.NET.211.146:4922	1725
11/11	212.179.27.6:1498	MY.NET.206.90:4619	1819
11/11	212.179.27.6:2078	MY.NET.206.90:4619	1087
11/13	212.179.79.2:32685	MY.NET.203.142:4619	1409
11/16	212.179.79.2:13270	MY.NET.218.142:4990	1459

WATCHLIST Source IP:

inetnum: 212.179.44.64 - 212.179.44.127
address: Bezeq International
address: 40 Hashacham St.
address: Petach Tikvah Israel
descr: GIVAT-BRENER-LAN
country: IL

WATCHLIST Source IP:

inetnum: 212.179.27.4 - 212.179.27.7
address: Bezeq International
address: 40 Hashacham St.
address: Petach Tikvah Israel
descr: ADI-ASSOCIATION-SERIAL
country: IL

WinGate Attempts – A Wingate or Socks proxy server commonly operate on ports 1080 and 8080. A Wingate proxy can be utilize to surf anonymously on the web. There are also vulnerabilities with certain versions of Wingate that allows intruders access to the Wingate server hard drive. There were a large number of scans to MY.NET apparently in search of Wingate servers. It is unclear from the logs to ascertain if any have been compromised. The Wingate access attempts occurred continuously between September 26 and November 22, 2000. There were **4802 hits** on MY.NET **in search of a Wingate Proxy**. The following are a list of the most intrusive sources.

Date	Source	Target	Number of WinGate Attmps hits
09/26	203.164.23.11:1710	MY.NET.219.118:1080	4
09/26	208.194.161.155:3797	MY.NET.207.54:1080	5
09/27	208.194.161.155:2209	MY.NET.210.170:1080	4
09/27	210.9.19.185:3047	MY.NET.53.45:1080	4
10/09	194.87.13.86:2074	MY.NET.209.34:1080	4
10/09	194.87.13.86:2363	MY.NET.209.34:1080	4
10/09	216.67.50.18:1424	MY.NET.60.11:1080	4
11/08	24.169.61.162:2731	MY.NET.98.226:1080	4

WINGATE ATTEMPTS Source IP:

First Internet Alliance (NETBLK-UU-208-194-160) UU-208-194-160 208.194.160.0 - 208.194.167.255

NULL Scans – MY.NET has been probed by a type of TCP packet in which no flag bits are set. This type of scan is utilized to map out a network topology. This is considered to be a reconnaissance of MY.NET, which is usually the prelude to a more directed attack. MY.NET received many NULL scans between September 26 and November 22, 2000. There were **283 Nulls Scans performed** on MY.NET. These are the targets that were scanned the most.

Date	Source	Target	Number of Null Scan hits
09/27	128.253.247.116:2720	MY.NET.227.10:4053	3
09/27	128.253.247.116:3932	MY.NET.227.10:3516	4
10/18	128.195.229.11:1498	MY.NET.205.2:1263	3
10/30	24.113.148.32:5501	MY.NET.214.166:3874	5
10/30	24.113.148.32:5501	MY.NET.214.166:3882	3
11/01	134.88.222.41:8311	MY.NET.212.142:1662	3
11/08	24.112.150.20:1030	MY.NET.105.120:1944	4

NULL SCAN Source IP:

Cornell University (NET-CCS-NET)
Cornell Information Technologies
Netblock: 128.253.0.0 - 128.253.255.255

NULL SCAN Source IP:

@Home
Netblock: 24.113.148.0 - 24.113.151.255

MY.NET Source and Destination Scans – host has also scanned multiple addresses inside MY.NET. These were **UDP scans** and they are hostile**.** In order for a host from MY.NET to generate this type of data, one of two possibilities exists. Either an employee is performing network activities that he is not authorized to or the sources have been compromised and are being controlled by an intruder. In either case, further investigation should be considered. These scans took place between September 26 and November 22, 2000**. MY.NET.5.25** has scanned many MY.NET. destinations on various days looking for Bootstrap Protocol Server service.

Source	Target	Number of UDP Scans hits
MY.NET.5.25:67	MY.NET.217.241:67	15
MY.NET.5.25:67	MY.NET.217.93:67	11
MY.NET.5.25:67	MY.NET.220.61:67	39
MY.NET.5.25:67	MY.NET.220.65:67	11
MY.NET.5.25:67	MY.NET.221.81:67	14
MY.NET.5.25:67	MY.NET.222.101:67	11
MY.NET.5.25:67	MY.NET.222.109:67	12
MY.NET.5.25:67	MY.NET.222.73:67	12
MY.NET.5.25:67	MY.NET.223.238:68	28
MY.NET.5.25:67	MY.NET.223.29:67	17
MY.NET.5.25:67	MY.NET.225.165:67	18
MY.NET.5.25:67	MY.NET.225.21:67	12

MY.NET SUMMARY OF SCANS - Due to time constraints, I've provided a summary of the "Top Ten" scans that were performed against MY.NET. These were…

Flags Used on Scan	Number of Scans Performed
SYN**S*****	235,386
SYNFIN**SF****	51,628
UDP	23,954
FIN***F****	454
VECNA*****P**	351
INVALIDACK**S*R*A*	281
NULL********	226
SYN21S*****RESERVEDBITS	104
INVALIDACK***FR*A*	60
NOACK*1SF*P**RESERVEDBITS	29

The following are a few ***OOS analysis*** that I had time to analyze.

SYN-FIN Scan

```
10/03  59:14.0209.92.40.32:9704 MY.NET.209.255:9704  TCP TTL:28 TOS:0x0 ID:39426**SF**** Seq:0x68DEADF9 Ack:0x790A34F6 Win:0x404
10/03  59:14.2209.92.40.32:9704 MY.NET.210.7:9704    TCP TTL:28 TOS:0x0 ID:39426**SF**** Seq:0x68DEADF9 Ack:0x790A34F6 Win:0x404
10/03  59:14.2209.92.40.32:9704 MY.NET.210.9:9704    TCP TTL:28 TOS:0x0 ID:39426**SF**** Seq:0x68DEADF9 Ack:0x790A34F6 Win:0x404
10/03  59:14.6209.92.40.32:9704 MY.NET.210.29:9704   TCP TTL:28 TOS:0x0 ID:39426**SF**** Seq:0x68DEADF9 Ack:0x790A34F6 Win:0x404
```

OS Fingerprinting.

DATE	TIME	SOURCE	DESTINATION	HEADER INFO	FLAGS	SEQ #	ACK #	Win Size
10/14	07:02.5	128.175.127.68:2784	MY.NET.202.250:8311	TCP TTL:122 TOS:0x0 ID:3216 DF	21SFRPAU	Seq:0x264300E	Ack:0x17A0227	Win:0x5010
10/14	50:49.0	MY.NET.218.106:1086	207.172.3.46:119	TCP TTL:126 TOS:0x0 ID:481 DF	21SFRPAU	Seq:0x18EC04	Ack:0x3B79BA23	Win:0x5010
11/07	01:47.5	130.234.183.108:1080	MY.NET.214.90:6688	TCP TTL:116 TOS:0x0 ID:10733 DF	21SFRPAU	Seq:0x2513DC	Ack:0x22BE471D	Win:0x5010
11/07	36:31.0	MY.NET.219.2:1621	207.172.3.46:119	TCP TTL:126 TOS:0x0 ID:44350 DF	21SFRPAU	Seq:0x4AEA924	Ack:0xCB93E90A	Win:0x5010
11/07	59:14.6	24.112.185.15:3107	MY.NET.205.34:6699	TCP TTL:117 TOS:0x0 ID:39957 DF	21SFRPAU	Seq:0x3E7040CC	Ack:0xF57FC3	Win:0x5018
10/03	34:45.4	206.158.102.30:1	MY.NET.224.206:1930	TCP TTL:117 TOS:0x0 ID:6605 DF	21SFRPAU	Seq:0x1A2B1171	Ack:0x52A4003A	Win:0x5010
10/07	26:05.3	MY.NET.220.142:1185	207.172.3.46:119	TCP TTL:126 TOS:0x0 ID:27242 DF	21SFRPAU	Seq:0xFE00B	Ack:0x31B3E25F	Win:0x5010
10/10	34:33.5	MY.NET.218.106:1226	207.172.3.46:119	TCP TTL:126 TOS:0x0 ID:48081 DF	21SFRPAU	Seq:0xA39EF	Ack:0xDFBA8D48	Win:0x5010
10/10	41:34.3	MY.NET.218.106:1226	207.172.3.46:119	TCP TTL:126 TOS:0x0 ID:28383 DF	21SFRPAU	Seq:0xA498A	Ack:0xE7DFD501	Win:0x5010
10/10	56:21.4	MY.NET.218.106:1226	207.172.3.46:119	TCP TTL:126 TOS:0x0 ID:41062 DF	21SFRPAU	Seq:0xA6D71	Ack:0xF9E3EFF2	Win:0x5010
10/10	57:50.4	MY.NET.218.106:1226	207.172.3.46:119	TCP TTL:126 TOS:0x0 ID:51376 DF	21SFRPAU	Seq:0xA70E5	Ack:0xFBB974FF	Win:0x5010
10/10	01:04.0	MY.NET.218.106:2238	207.172.3.46:119	TCP TTL:126 TOS:0x0 ID:59653 DF	21SFRPAU	Seq:0x1428B9D	Ack:0x88D93EA7	Win:0x5010
10/10	51:26.0	140.254.110.71:1482	MY.NET.212.118:6699	TCP TTL:119 TOS:0x0 ID:52470 DF	21SFRPAU	Seq:0xAAE1A49	Ack:0x3CD18AAA	Win:0x5018

Summary/Recommendations: After analyzing one-month worth of MY.NET traffic, it is recommended that further investigation of the (possible compromised) abovementioned hosts be perform to ensure these in fact have not been compromised. It is apparent that MY.NET suffers from continually being probed and scanned, both randomly seeking available services as well as being directly targeted in search of Trojan Horses and other vulnerabilities. Also observed was the transfer of the Happy 99 virus via email. Several networking security measures need to be examined and hardened such as the Windows File & Print sharing and the SNMP public & private strings. It is highly recommended that all latest patches and hot fixes be installed on all operating systems and software utilized in the MY.NET network to prevent exploitation of the same. By taking additional security measures, MY.NET will be able to avoid the loss of data as well as being responsible for innocently attacking others machines (being used as an amplified network or any of the MY.NET hosts be used as zombies/slaves). A benefit of eliminating this excessive amount of random network traffic is an increase in network performance and security.

Assignment 3 – Analysis Process

Procedures Executed:

Due to the lack of access to a descent data mining tools, a good amount of time and effort was spent in putting together and tailoring all Snort log files that were provided (Alert, Scans, and Snort Data Dump files). As a means to provide the best analysis possible, I felt the need to merge together all similar files (i.e. all alert files, all Snort Packet dump, and all the scan files).

- First similar files format were move to separate directories and merged via DOS command (Prompt> copy SnortA*.* AllAlertsFiles.txt). The same was done with the Scans and OOS files.
- Then Perl scripts were created and used to separate (for Alerts Only) and parse.

ALERT FILES:

This script was created to extract all alerts from within the "AllAlertsFiles".

```
#!user/bin/perl
opendir(BIN, "/Perl/bin/alert/");
@files = readdir(BIN);
closedir BIN;
foreach $name (@files) {
    open (ALERT,
"/perl/bin/alert/$name");
    @file = <ALERT>;
    foreach $line(@file) {
        if ($line =~ /-\>/) {
            push (@type, $line);
        }
    }
}
open(BIG,
">>/Perl/bin/alert/record1.txt")
|| die;
print BIG @type;
```

This script was created to parse relevant data from all alerts for analysis.

```
#!user/bin/perl
open(RECORDS,
"/perl/bin/alert/record1.txt") ||
die;
open(RECORD,
">>/perl/bin/alert/parsed1.txt")
|| die;
@file = <RECORDS>;
foreach $line (@file) {
    @row = split(/\[\*\*\]/,
$line);
    $first = $row[0];
    $second = $row[1];
    $third = $row[2];
    @one = split (/-/, $first);
    @three = split (/-\>/,
$third);
    print RECORD $one[0] . "|";
    print RECORD $one[1] . "|";
    print RECORD $second . "|";
    print RECORD $three[0] . "|";
    print RECORD $three[1] . "|";
}
```

This script was created to extract all summarized one-liners from within the "AllAlertsFiles".

```
#!user/bin/perl
open(RECORD,
"/perl/bin/alert/parsed2.txt") ||
die;
open(RECORD2,
">>/perl/bin/alert/parsed21.txt")
|| die;
@file = <RECORD>;
foreach $line (@file) {
        $line =~ s/\(TOTAL
HOSTS\|/\|\(TOTAL HOSTS/g;
        $line =~
s/\(STEALTH/\|\(STEALTH/g;
        $line =~ s/\|\|/\|/g;
        $line =~
s/\(THRESHOLD/\|\(THRESHOLD/g;
        print RECORD2 $line;
}
close RECORD;
```

This script was created to parse relevant data from all summarized one-liners for analysis.

```
#!user/bin/perl
open(RECORDS,
"/perl/bin/alert/record2.txt") ||
die;
open(RECORD,
">>/perl/bin/alert/parsed2.txt")
|| die;
@file = <RECORDS>;
foreach $line (@file) {
    @row = split(/\[\*\*\]/,
$line);
    $first = $row[0];
    $second = $row[1];
    @one = split (/-/, $first);
    @two = split (/from/,
$second);
    $column = $two[1];
    @three = split (/\:/,
$column);
    print RECORD $one[0] . "|";
    print RECORD $one[1] . "|";
    print RECORD $two[0] . "from"
. "|";
    print RECORD $three[0] . "|";
    print RECORD $three[1] . "|" .
"\n";
}
```

SCAN FILES:

This script was created to parse the AllScans file.txt

```
#!user/bin/perl
open(RECORDS,
"/perl/bin/scan/record1.txt") ||
die;
open(RECORD,
">>/perl/bin/scan/parsed1.txt") ||
die;
@file = <RECORDS>;
foreach $line (@file) {
    @row = split(/->/, $line);
    $first = $row[0];
    $second = $row[1];
    @one = split (/\s/, $first);
    @two = split (/\s/, $second);
    print RECORD $one[0] . $one[1]
. "|";
    print RECORD $one[2] . "|";
    print RECORD $one[3] . "|";
    print RECORD $two[1] . "|";
    print RECORD $two[2] . $two[3]
. $two[4] . $two[5]
     . $two[6] . $two[7] . "|" .
"\n";
}
```

OOS FILES:

This script was created to parse the AllOOS file.txt

```
#!user/bin/perl
open(RECORDS,
"/perl/bin/oos/record12.txt") ||
die;
open(RECORD,
">>/perl/bin/oos/parsed1.txt") ||
die;
@file = <RECORDS>;
foreach $line (@file) {
    @row = split(/\|\|\|/, $line);
    $first = $row[1];
    $second = $row[2];
    $third = $row[3];
    $four = $row[4] . $row[5];

    @one = split (/->/, $first);
    $firstone = $one[0];
    @lineone = split (/\s/,
$firstone);
    $firsttwo = $lineone[0];
    @lineonetwo = split (/-/,
$firsttwo);

    @three = split (/\s/, $third);


    print $lineonetwo[0] . "\|";
   print  $lineonetwo[1] . "\|";
    print $lineone[1] . "\|";
   print $one[1] . "\|" ;
    print $second . "\|";
    print $three[0] . "\|";
    print $three[1] . $three[2] .
"\|";
     print  $three[5] . $three[6]
. "\|";
    print $three[9] . $three[10] .
"\|";
     print $four . "\n";
}
```

- After parsing the files, they were imported into Access as tables.
- Data was then analyzed using Pivot Tables.

Last Updated: May 23rd, 2017

Upcoming Training

Mentor Session - SEC503	Oceanside, CA	May 29, 2017 - Jun 29, 2017	Mentor
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC503: Intrusion Detection In-Depth	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
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 September 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 Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS San Diego 2017	San Diego, CA	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
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZ	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced