GIAC Intrusion Detection Immersion Curriculum

SANS NS 2000, Monterey, California
- October 15 – 22, 2000 -

Practical Assignment (46 Pages)

Submitted by
Kyuchul Song, CISSP

*** Assignment 1 – Network Detects (30 Points) ***

Trace 1 – rpc.statd format string attack

```
Jan 21 19:38:11 victim rpc.statd[363]: gethostbyname error for
^X??X??Y??Y??Z??Z??[??[?\277bffff750 8049710
8052c20687465676274736f6d616e797265206520726f7220726f66
bffff718
bffff719
bffff71a
bffff71b\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
(truncated --)
```

Source of trace:
This trace was extracted from messages log on a compromised Linux server in South Korea (http://www.korea.net/ ↑↑).

Detect was generated by:
/var/adm/messages on a compromised Linux server
Format: timestamp | host | program [pid] | message | streams

Probability the source address was spoofed:
Low. This attack requires a 3-way handshake in order to compromise the target.

Description of attack:
This is an attack against rpc.statd format string vulnerability. The specific tool in this case is probably statd-toy.c/rpc-statd-xpl.c/statdx.c. The rpc.statd passes user-supplied data and without validation of this data, attacker may supply machine code to be executed with the privileges of the rpc.statd process, typically root.

CVE-2000-0666	rpc.statd in the nfs-utils package in various Linux distributions does not properly cleanse untrusted format strings, which allows remote attackers to gain root privileges
Bid 148	Multiple Linux Vendor rpc.statd Remote Format String Vulnerability

Attack mechanism:
Cited from Bid 148:
The rpc.statd server is an RPC server that implements the Network Status and Monitor RPC protocol. It's a component of the Network File System (NFS) architecture.
The logging code in rpc.statd uses the syslog() function passing it as the format string user supplied data. A malicious user can construct a format string that injects executable code into the process address space and overwrites a function's return address, thus forcing the program to execute the code. rpc.statd requires root privileges for opening its network socket, but fails to drop these privileges later on. Thus code executed by the malicious user will execute with root privileges. Debian, Red Hat and Connectiva have all released advisories on this matter. Presumably, any Linux distribution which runs the statd process is vulnerable, unless patched for the problem.
For more detailed information about format string attack – SANS: Format String Attacks: 101.

Correlations:

There are lots of correlated data and analysis reports. Here are just several samples: SANS GIAC Page search results ("rpc.statd+format+string"), George Bakos's GCIA practical, Joseph R. Rach's GCIA practical and Bid 148.

Evidence of active targeting:

This attack actively targeted at the specific system (Linux server) and succeeded in compromise.

Severity:

Severity = (Criticality + Lethality) - (System countermeasures + Network countermeasures)

Criticality	5	DNS server
Lethality	5	Attacker gained root access
System countermeasures	3	Older OS, some patches missing
Network countermeasures	2	Permissive Firewall

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

Defensive recommendation - Bid 148:

- Upgrade version of rpc.statd or disable the rpc.statd service if an update cannot be applied
- Block unneeded ports at firewall. Particularly, block port 111 (portmapper), as well as the port on which rpc.statd is running, which may vary

Multiple choice test question:

The above trace can be classified into?

- Configuration error
- Input validation error
- Race condition error
- Failure to handle exceptional conditions

Correct answer: 2

This is SecurityFocus classification.

Trace 2 – POP server buffer overflow attack

```
Jan 10 02:01:33 www 133>Jan 10 02:01:33 popper[16513]: @[attacker]:
-ERR Unknown authentication mechanism:
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
(truncated --)
```

Source of trace:

This trace was extracted from messages log on a compromised POP server in South Korea.

Detect was generated by:
Log messages on compromised POP server
Format: timestamp | host? | port? | > timestamp | program [pid] | @ attacker address | message | streams
I am not sure because enough information, such as POP server vendor and version, was not posted.

Probability the source address was spoofed:
Low. This attack requires a 3-way handshake in order to compromise the target.

Description of attack:
This is an attack against POP server buffer overflow vulnerability. The specific tool in this case is not clear, but probably linux-qpopper.c/bsd-qpopper.c. This attack makes it possible for a remote user to execute arbitrary commands on targets running a vulnerable version. Here are a number of known vulnerabilities (mainly buffer overflow) on POP server: CVE-1999-0006, CVE-1999-0042, CVE-1999-0494, CVE-1999-0759, CVE-1999-0920, CVE-1999-1004, CVE-2000-0091, CVE-2000-0442, CVE-2000-0989, CAN-1999-0673, CAN-2000-0016, CAN-2000-0060, CAN-2000-0592, CAN-2000-0840, CAN-2000-0841.

Attack mechanism:
Cited from Bid 133:
The vulnerability exists in the way POP daemon handles user supplied input for a number of pop commands, including, but not limited to, USER, PASS, as well as any line containing in excess of 1024 characters. This buffer overflow makes it possible for a remote user to execute arbitrary commands and gain root access on target.
From the messages log "-ERR Unknown authentication mechanism", it seems that attacker could gain root access using long username or password.

Correlations:
About 2 years ago, this attack against POP server was very popular in South Korea (and in the World??), but recently not. Therefore, it is difficult to find correlations. However, there are lots of correlated scannings for POP server, not buffer overflow! Here are just several samples: SANS GIAC Page search results ("pop3"), William Totten's GCIA practical, Joanne Treurniet's practical.

Evidence of active targeting:
This attack actively targeted at the specific system (POP server - victim) and succeeded in compromise.

Severity:
Severity = (Criticality + Lethality) - (System countermeasures + Network countermeasures)

Criticality	2	Unknown, assume 2: User UNIX desktop system
Lethality	5	Attacker gained root access
System countermeasures	3	Older OS, some patches missing
Network countermeasures	2	Permissive Firewall

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

Defensive recommendation:
- Upgrade version of POP server or disable the POP service if an update cannot be applied

- Deploy proxy-based packer filter devices which can filter specific command at proxy level

Multiple choice test question:
The above trace can be classified into?

- Boundary condition error
- Input validation error
- Race condition error
- Failure to handle exceptional conditions

Correct answer: 1
This is SecurityFocus classification.

Trace 3 – RPC Info Query

```
[**] RPC Info Query [**] 05/29-17:58:53.527261 209.27.200.129:986 ->
  nnn.n.nnn.130:111 TCP TTL:240 TOS:0x0 ID:28571 DF *****PA* Seq: 0xE95458DA
  Ack: 0xC901040F Win: 0x2238
[**] RPC Info Query [**] 05/29-17:59:15.029450 209.27.200.129:648 ->
  nnn.n.nnn.172:111 TCP TTL:240 TOS:0x0 ID:50061 DF *****PA* Seq: 0xE9D58B5F
  Ack: 0x47A7B659 Win: 0x2238
[**] RPC Info Query [**] 05/29-17:59:43.022267 209.27.200.129:761 ->
  nnn.n.nnn.229:111 TCP TTL:240 TOS:0x0 ID:12515 DF *****PA* Seq: 0xEA7C9968
    Ack: 0x1EF74F3F Win: 0x2238
```

Source of trace:
SANS GIAC Page: http://www.sans.org/y2k/053100-1100.htm.

Detect was generated by:
The data was collected by Snort. The rule that triggered this alert is:

```
alert tcp !$HOME_NET any ->$HOME_NET 111 (msg:"RPC Info Query"; content:"|00 01 86
A0 00 00 00 02 00 00 00 04|";)
```

Format: alert | timestamp | src ip:port -> dst ip:port | protocol | TTL | TOS | flags | sequence number | ACK number | window size

Probability the source address was spoofed:
Low. This attack is a reconnaissance, which can be succeeded only if a response is received. Attack address (209.27.200.129) is registered to Cable & Wireless, Inc.

Description of attack:
Attacker performed several scans for the same port within very short time interval (<1 second). This indicates that attacker used an automated scanning tool like Nmap. Attacker scans the network to order to find vulnerable systems running portmapper (or rpcbind) services on port 111 and query for a list of RPC services registered to portmapper. There are number of known vulnerabilities (mainly buffer overflow) with RPC services. With this information, attacker will launch a serious exploit against the running services. Here are a number of known vulnerabilities on RPC services: CVE-1999-0003, CVE-1999-0008, CVE-1999-0208, CVE-1999-0212, CVE-1999-0320, CVE-1999-0353, CVE-1999-0493, CVE-1999-0687, CVE-1999-0696,

CVE-1999-0900, CVE-1999-0969, CVE-1999-0974, CVE-2000-0508, CVE-2000-0771, CAN-1999-0078, CAN-1999-0195, CAN-1999-0568, CAN-1999-0613, CAN-1999-0625, CAN-1999-0632, CAN-1999-0795, CAN-2000-0114, CAN-2000-0544, CAN-2000-0800.

Attack mechanism:

Attacker queried a rpcinfo request to several targets on the network. *rpcinfo –p* lists RPC services registered to portmapper on port 111 and their associated version/protocol/port. There are a number of serious vulnerabilities associated with RPC services. Attacker then will attempt to launch more appropriate attack to compromise the running ports and gain unauthorized root access on the target. Here is sample *rpcinfo –p* on my network.

```
#rpcinfo -p my.net.host
program  vers  proto  port  service
100000    4   tcp    111  rpcbind
100000    3   tcp    111  rpcbind
100000    2   tcp    111  rpcbind
100000    4   udp     111  rpcbind
100002    3   tcp    32771 rusersd
   (truncated --)
```

Correlations:

There are lots of correlated data and analysis reports. Here are just several samples: SANS GIAC Page search results (“rpc+info+query”), Marc Bayerkohler’s GCIA practical.

Evidence of active targeting:

This attack is a general scan for several targets. However, I am not sure because enough information was not posted.

Severity:

Severity = (Criticality + Lethality) - (System countermeasures + Network countermeasures)

Criticality	2	Unknown, assume 2: User UNIX desktop system
Lethality	3	Reconnaissance, but could be used for a serious attack
System countermeasures	3	Unknown, assume 3: Older OS, some patches missing
Network countermeasures	2	Permissive Firewall

Severity =(2 + 3) - (3 + 2) = 0

Defensive recommendation:

- Patch the vulnerable RPC services and disable the unneeded RPC services
- Block inbound rpcinfo query at packet filter devices. Particularly, block port 111 (portmapper)

Multiple choice test question:

Which of the following command is used to list RPC services?

- rpcinfo –p
- rpcinfo –d
- rpcinfo –u
- rpcinfo –t

Correct answer: 1

For more detailed information: rpcinfo manual.

Trace 4 – OS fingerprinting

```
Feb 3 15:11:58 66.50.24.49:18245 -> a.b.c.44:21536 VECNA *******U
Feb 3 15:12:02 66.50.24.49:18245 -> a.b.c.44:21536 NOACK 2*SFRP*U RESERVEDBITS
Feb 3 15:12:02 66.50.24.49:18245 -> a.b.c.44:21536 VECNA 2****P*U RESERVEDBITS
Feb 3 15:12:02 66.50.24.49:18245 -> a.b.c.44:21536 XMAS 2**F*P*U RESERVEDBITS
Feb 3 15:12:05 66.50.24.49:18245 -> a.b.c.44:21536 INVALIDACK 2***R*AU
RESERVEDBITS
(truncated --)
```

Source of trace:
SANS GIAC Page: http://www.sans.org/y2k/013101-1200.htm.

Detect was generated by:
The data was collected by Snort.
Format: timestamp | src ip:port -> dst ip:port | alert | TCP flags

Probability the source address was spoofed:
Low. This attack is a reconnaissance (OS fingerprinting), which can be succeeded only if a response is received.
Attack address (66.50.25.19) is registered to Puerto Rico Telephone Company.

Description of attack:
This is a reconnaissance attack – TCP/IP stack fingerprinting to identify OS type of target. The close timestamp, illegal TCP flag combinations and unchanged source port number indicate that attacker used an automated scanning tool like Namp, Queso and hping.

Attack mechanism:
This attack is very popular OS fingerprinting. This is the scanning of sending intentionally-crafted illegal (mainly, illegal TCP flags combinations) packets to target and then examining the responses to identify OS type. This is possible because each developer of an operating system implements TCP/IP a bit differently than another developer of an operating system, different operating system's TCP/IP stack could respond differently given the same situation in a TCP/IP conversation, especially illegal packets. With this information the attacker can determine an appropriate attack against the target OS. Nmap and Queso are the most popular and powerful OS fingerprinting tools.
More detailed information can be found at SANS: ID FAQ - TCP/IP Stack Fingerprinting Principles.

Correlations:
There are lots of correlated data and analysis reports. Here are just several samples: SANS GIAC Page search results ("fingerprint"), Crist Clark - GCIA Practical Assignment, Todd Garrison's GCIA Practical.

Evidence of active targeting:
Yes. This attack actively targeted at the specific target (a.b.c.44).

Severity:
Severity = (Criticality + Lethality) - (System countermeasures + Network countermeasures)

Criticality	2	Unknown, assume 2: User UNIX desktop system

Lethality	2	Reconnaissance
System countermeasures	3	Unknown, assume 3: Older OS, some patches missing
Network countermeasures	2	Permissive Firewall

Severity =(2 + 2) - (3 + 2) = -1

Defensive recommendation:

- Drop illegal traffic, especially illegal combinations of TCP flags
- Include attack address into watchlist for further investigation

Multiple choice test question:

Which of the following tools is not used to identify OS type – OS fingerprinting?

- Queso
- Nmap
- hping
- Whisker

Correct answer: 4

Whisker is a popular and stealthy CGI scanner:

- http://www.wiretrip.net
- A look at Whisker's anti-IDS tactics

*** Assignment 2 – "Analyze This" Scenario (40 Points) ***

Introduction

This is a security analysis report about MY.NET network. MY.NEET network had been monitored with the Snort intrusion detection system for about 2 months. This report covers the followings:

- Data collection
- Overall analysis of Snort alert reports
- Detailed analysis of specific alerts
- Overall analysis of Snort scan reports
- Overall analysis of Snort alerts from internal network
- Overall analysis of Snort alerts from internal network
- Probably compromised hosts
- Summary and recommendations

Data collection

The most popular free IDS tool, Snort IDS, was used for monitoring suspicious traffics on MY.NET network. There are 3 types of Snort reports: Alerts, Scans and Raw data. The following table shows the collected Snort data.

[Table 1] Snort data

File name	Data type	Earliest	Latest	# Files	Total file size
SnortA*.txt	Alert report	09/26/00	11/22/00	54	14.9 MB+
SnortS*.txt	Scan report	09/27/00	11/23/00	42	21.2 MB+
SOOS*.txt	Raw Snort data	08/17/00	11/11/00	19	16.7 MB+

From the table above, it is apparent that there are not full data for all days - due to various reasons: power failure, disk full, etc. However, in my opinion, it is sufficient to suggest overall security picture of MY.NET network • .

Overall summary of Snort alert logs

The following table presents the overall summary of suspicious alerts on MY.NET network. This table clearly shows a huge amount of hostile traffic - 110457. There are probably lots of legitimate traffics - false positives. However, it is also apparent that there are lots of hostile activities needing further investigation. Specific alerts will be analyzed in the next section.

Total number of signatures: 20
Total number of alerts: 110457
Time interval: 09/26/00 – 11/22/00

[Table 2] Statistics of Snort alert signatures

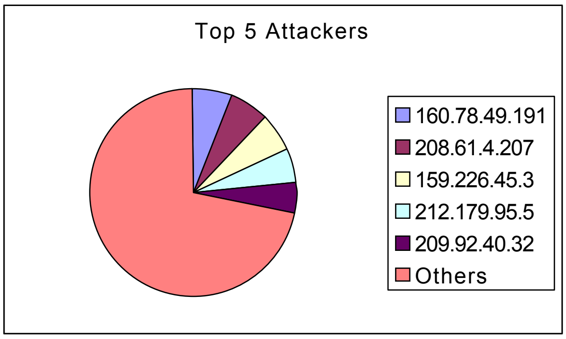
Signature	# Alerts	# Sources	# Destinations

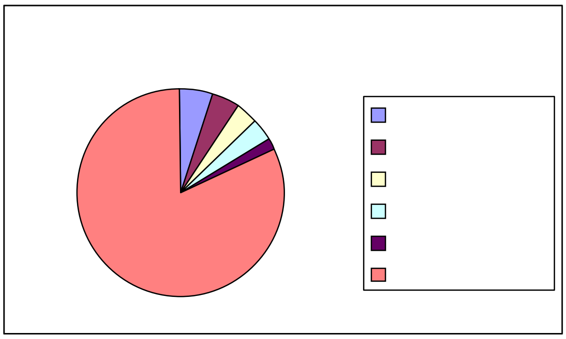
SYN-FIN scan!	56250	30	25751
Watchlist 000220 IL-ISDNNET-990517	30997	61	108
Watchlist 000222 NET-NCFC	8134	45	26
WinGate 1080 Attempt	4764	570	2655
TCP SMTP Source Port traffic	2893	4	2836
Attempted Sun RPC high port access	2542	20	33
Broadcast Ping to subnet 70	1813	216	1
Back Orifice	1697	40	932
SNMP public access	468	23	1
Null scan!	277	204	196
SMB Name Wildcard	218	33	33
Queso fingerprint	142	29	58
NMAP TCP ping!	96	21	20
SUNRPC highport access!	60	13	12
connect to 515 from inside	56	2	3
Probable NMAP fingerprint attempt	15	14	13
SITE EXEC – Possible wu-ftpd exploit – GIAC000623	13	4	7
External RPC call	13	8	3
Tiny Fragments – Possible Hostile Activity	7	5	6
Happy 99 Virus	2	2	2

The following table lists Top 5 Alert attacker and target addresses.

[Tabel 3] Top 5 Alert attackers and targets

Sources	Whois	# Alerts	Destinations	# Alerts
160.78.49.191	Centro di Calcolo di Ateneo	7199	MY.NET.6.7	5800
208.61.4.207	BellSouth.net Inc	6635	MY.NET.211.146	4814
159.226.45.3	The Computer Network Center Chinese Academy of Sciences	6295	MY.NET.223.98	3940
212.179.95.5	Cable-Modem-Experiment, IL	6117	MY.NET.206.90	3918
209.92.40.32	FASTNET Corporation	4967	MY.NET.70.255	1813





[Figure 1] Distributions of Top 5 Alert attackers and targets

A complete investigation into these prevalent attackers and targets needs to minimize the impact of the associated risks. Furthermore, it should be noted that 159.226.45.3 and 212.179.95.5 are included in the Watchlist address spaces (Watchlist 000222 NET-NCFC, Watchlist 000220 IL-ISDNNET-990517).

Detailed analysis of specific alerts

1. Reconnaissance alerts

The following table shows the overall summary of reconnaissance alerts. With the reconnaissance techniques, attackers could gather useful information about a network and sometimes evade IDS and Firewalls. This information could be used in the future serious attacks against the same target.

[Table 4] Statistics of Reconnaissance alert signatures

Signature	# Alerts	# Sources	# Destinations
SYN-FIN scan!	56250	30	25751
Null scan!	277	204	196
Queso fingerprint	142	29	58
NMAP TCP ping!	96	21	20
Probable NMAP fingerprint attempt	15	14	13

Reconnaissance techniques can be used for the following specific purposes:

- OS fingerprinting to identify OS type of the target
  - ■ Probable NMAP fingerprint attempt
  - ■ Queso fingerprint
  - ■ Null scan!
- Port scanning to find open ports of the target
  - ■ SYN-FIN scan! (to find open ports)
  - ■ NMAP TCP ping! (to determine the live computers and to find open ports)

1.1. OS fingerprinting alerts

Alert description: OS fingerprinting is the scanning of sending intentionally-crafted illegal (mainly, illegal TCP flags combinations) packets to target and then examining the responses to identify OS type OS. With this information the attacker can determine an appropriate attack against the target OS. Nmap and Queso are the most popular and powerful OS fingerprinting tools.

Statistics: There are 434 alerts from 243 sources to 252 destinations. The following table summarizes the most prevalent source and destination addresses:

[Table 5] Statistics of OS fingerprinting alerts

Sources	# Alerts	Destinations	# Alerts
24.3.161.193	45	MY.NET.145.9	43
195.115.7.2	22	MY.NET.217.26	23
129.242.219.27	19	MY.NET.227.10	12
64.80.63.121	15	MY.NET.130.116	9
128.253.247.116	13	MY.NET.105.120	9

Analysis: Probably, there are some false positives – legitimate ECN traffics. However, it is impossible to know for sure without further analysis. More detailed information can be found at Teri BidWell's GCIA practical and Toby Miller's report on ECN and it's impact on intrusion detection.

Sample signatures:

11/22-22:44:52.018936 [**] Probable NMAP fingerprint attempt [**] 24.69.214.58:2648 -> MY.NET.224.150:4999
11/22-16:10:36.268157 [**] Queso fingerprint [**] 193.251.42.11:18189 -> MY.NET.203.118:6346
11/22-20:33:10.371736 [**] Null scan! [**] 24.13.101.55:1742 -> MY.NET.130.91:20

Defensive recommendations:

- Immediately investigate all targets to see if system is compromised
- Block traffics from attack addresses and illegal traffic at the packet filter devices (Firewall, router, etc), if such traffics are indeed not required
- Include attack addresses into Watchlist for further investigation
- Deploy the latest security product – IDS, Firewall, etc.

1.2. Port scanning alerts

Alert description: Port scanning is the scanning of sending packets (usually illegal) to target and then examining the responses to determine the live computers or to find open ports. With this information the attacker can determine an appropriate attack against the open port. Especially, SYN-FIN scan and NMAP TCP ping techniques are very stealthy method that sometimes can evade IDS and Firewalls.

Statistics: There are 56346 alerts from 51 sources to 25756 destinations. The following table summarizes the most prevalent source and destination addresses:

[Table 6] Statistics of Port scanning alerts

Sources	# Alerts	Destinations	# Alerts
160.78.49.191	7199	MY.NET.1.8	51
208.61.4.207	6635	MY.NET.223.251	12
209.92.40.32	3897	MY.NET.201.126	8
63.195.56.20	3860	MY.NET.104.90	8

130.89.229.48	3572	MY.NET.1.88	8

Analysis: It should be noted that most attackers performed pretty heavy scanning on the wide range of hosts within short time interval. The below sample illustrates this point: (7199 scans / 23 minutes)

```
09/30-13:10:30.153412  [**] SYN-FIN scan! [**] 160.78.49.191:53 -> MY.NET.1.9:53
***
09/30-13:32:06.932517  [**] SYN-FIN scan! [**] 160.78.49.191:53 -> MY.NET.254.253:53
```

It indicates that most attackers used the automated scanning tool like Nmap.

Most attackers scanned for known vulnerable services (FTP, DNS, rpc.statd) or for already compromised ports (SubSeven, other Trojans/backdoors). The following table lists the most popular destination ports.

[Table 7] Top 5 Port scanning destination ports

Destination ports	# Alerts
21 (FTP control)	19613
53 (DNS)	18341
9704 (Linux rpc.statd)	14184
27374 (SubSeven Trojan)	3572
23 (telnet)	327

Through these ports, an attacker can gain root access by exploiting vulnerable or backdoor problem. Quite lethal! More detailed information about vulnerabilities in each port can be found at hyperlink (NetworkICE port knowledgebase).

Sample signatures:

```
09/30-13:10:30.153412  [**] SYN-FIN scan! [**] 160.78.49.191:53 -> MY.NET.1.9:53
11/22-22:05:59.996054  [**] NMAP TCP ping! [**] 63.119.91.2:80 -> MY.NET.1.3:53
```

Defensive recommendations:

- Immediately investigate all targets to see if system is compromised
- Disable the vulnerable service ports and backdoor port, if such ports are indeed not required
- Check that all systems are running with the latest patches
- Block traffics from attack addresses at the packet filter devices (Firewall, router, etc), if such traffics are indeed not required
- Include attack addresses into Watchlist for further investigation
- Deploy the latest security product – IDS, Firewall, etc

2. Happy 99 Virus

Alert description: Happy 99 virus (alias W32/Ska) is a worm that runs as an e-mail attachment, which displays a message "Happy New Year 1999!!" and displays "fireworks" graphics. The posting on the newsgroups has lead to its propagation. It can also spread on its own, as it can attach itself to a mail message and be sent unknowingly by a user. More detailed information can be found at http://vil.nai.com/villib/dispVirus.asp?virus_k=10144.

Statistics: There are 2 alerts from 2 sources to 2 destinations. The following table lists source and destination addresses:

[Table 8] Statistics of Happy 99 Virus alerts

Sources	# Alerts	Destinations	# Alerts
209.94.224.13	1	MY.NET.253.41	1
216.6.117.11	1	MY.NET.6.35	1

Analysis: MY.NET.253.41 and MY.NET.6.35 have possibly been compromised by Happy 99 virus. A complete recovery of these hosts needs to minimize the impact of the associated risks.

Sample signatures:

10/05-03:59:51.460766 [**] Happy 99 Virus [**] 216.6.117.11:41827 -> MY.NET.253.41:25
11/06-16:06:44.170359 [**] Happy 99 Virus [**] 209.94.224.13:2708 -> MY.NET.6.35:25

Defensive recommendations:

- Immediately virus-scan for Happy 99 on MY.NET.253.41 and MY.NET.6.35
- Install e-mail anti-virus software and update signatures continuously
- Block traffics from attack addresses at the packet filter devices (Firewall, router, etc), if such traffics are indeed not required
- Include attack addresses into Watchlist for further investigation

3. SITE EXEC – Possible wu-ftpd exploit – GIAC000623 & site exec – Possible wu-ftpd exploit – GIAC000623

Alert description: Due to a misconfiguration, some distributed binaries of wu-ftp version 2.4.1 and earlier allow an attacker with an FTP account on the system to gain root access by running a shell or other command using site exec. More detailed information can be found at SecurityFocus Bugtraq: 1995-11-30: wu-ftpd /bin SITE EXEC Misconfiguration Vulnerability.

Statistics: There are 13 alerts from 4 sources to 7 destinations. The following table lists the most prevalent source and destination addresses:

[Table 9] Statistics of Possible wu-ftpd exploit alerts

Sources	# Alerts	Destinations	# Alerts
208.61.44.215	9	MY.NET.205.94	4
24.31.88.99	2	MY.NET.130.242	3
63.202.13.20	1	MY.NET.221.82	2
202.9.188.89	1	MY.NET.100.209	1

Analysis: It should be noted that since a vulnerable server would allow the attacker to gain root access, this attack is quite lethal!

Sample signatures:

10/04-11:56:14.289566 [**] SITE EXEC - Possible wu-ftpd exploit - GIAC000623 [**] 63.202.13.20:1188 -> MY.NET.100.209:21
10/01-06:17:25.604955 [**] site exec - Possible wu-ftpd exploit - GIAC000623 [**] 208.61.44.215:3739 -> MY.NET.97.206:21

Defensive recommendations:

- Immediately investigate ftp log files of all target to see if system is compromised

- Check that all systems are running with the latest wu-ftpd patches
- Deploy proxy-based packer filter devices which can filter specific command at proxy level
- Block traffics from attack addresses at the packet filter devices (Firewall, router, etc), if such traffics are indeed not required
- Include attack addresses into Watchlist for further monitoring
- Deploy the latest security product – IDS, Firewall, etc

4. Tiny fragments
Alert description: Tiny fragmentation means that attackers intentionally craft shorter fragmented packets than the normal size (ex: Half-truncated TCP header packet). Tiny fragmentations can be used to launch denial of service or evade IDS and Firewalls. More detailed information can be found at SANS: IP Fragmentation and Fragrouter.

Statistics: There are 7 alerts from 5 sources to 6 destinations. The following table lists the most prevalent source and destination addresses:

[Table 10] Statistics of Tiny fragments alerts

Sources	# Alerts	Destinations	# Alerts
216.43.55.44	2	MY.NET.181.144	2
62.6.71.0	2	MY.NET.1.8	1

Sample signatures:
09/26-21:25:17.293957 [**] Tiny Fragments - Possible Hostile Activity [**] 172.157.126.93 -> MY.NET.201.2

Defensive recommendations:
- Immediately investigate all targets to see if system is compromised
- Deploy state-full IDS and Firewall which can reassembles fragmented packets or drop illegally-tiny fragmented packet
- Disable the vulnerable service ports and backdoor port, if such ports are indeed not required
- Check that all systems are running with the latest patches
- Block traffics from attack addresses at the packet filter devices (Firewall, router, etc), if such traffics are indeed not required
- Include attack addresses into Watchlist for further monitoring

5. External RPC Call
Alert description: External RPC call is an attempt to access the RPC service (rpcbind, portmapper) on port 111. External RPC call could list all the RPC programs that have a number of known vulnerabilities (mainly buffer overflow) and can be further exploited to grant root access. More detailed information can be found at NetworkICE: SUNRPC port probe.

Statistics: There are 13 alerts from 8 sources to 3 destinations. The following table lists the most prevalent source and destination addresses:

[Table 11] Statistics of External RPC Call alerts

Sources	# Alerts	Destinations	# Alerts
63.162.239.69	3	MY.NET.6.15	9

200.191.80.181	2	MY.NET.100.130	3
200.191.80.206	2	MY.NET.15.127	1

Analysis: It should be noted that since a vulnerable server would allow the attacker to gain root access, this attack is quite lethal!

Sample signatures:
10/10-20:23:36.018641 [**] External RPC call [**] 200.191.80.206:931 -> MY.NET.6.15:111

Defensive recommendations:

- Immediately investigate all targets to see if system is compromised
- Disable the portmapper service port (111), if this port are indeed not required
- Check that all systems are running with the latest patches
- Block traffics from attack addresses at the packet filter devices (Firewall, router, etc), if such traffics are indeed not required
- Include attack addresses into Watchlist for further monitoring
- Deploy the latest security product – IDS, Firewall, etc

6. Attempted Sun RPC high port access & SUNRPC high port access !

Alert description: This alert is similar to 'External RPC call' alert. Some SunOS machines listen at port 32771 (ghost portmapper) for portmapper in addition to the standard port 111. Since Firewalls frequently do not block high ports, it can allow the attacker access to portmapper even when port 111 is blocked. Fore more detailed information, refer to the 'External RPC call' section.

Statistics: There are 2062 alerts from 33 sources to 43 destinations. The following table lists the most prevalent source and destination addresses:

[Table 12] Statistics of Attempted Sun RPC high port access & SUNRPC high port access alerts

Sources	# Alerts	Source Port #	# Alerts	Destinations	# Alerts
205.188.153.0/24	2536	4000	2534	MY.NET.221.246	488
216.10.12.30	33	2078	33	MY.NET.225.210	435
216.148.218.160	6	5190	6	MY.NET.217.214	365
205.188.3.211	4	443	6	MY.NET.206.222	299
24.18.90.197	3	2089	4	MY.NET.222.98	187

Analysis: It should be noted that there are probably a large number of false positives. Most traffics from 205.188.153.0/24 (America Online, Inc) are legitimate AOL ICQ traffics. AOL runs ICQ usually on port 4000. The below sample illustrates this point:
09/26-08:34:21.306733 [**] Attempted Sun RPC high port access [**] 205.188.153.105:**4000** -> MY.NET.220.78:32771
This is correlated with Teri BidWell's GCIA practical.
Except possible false positives, there are 68 alerts from 19 sources to 17 destinations. The following table lists the most prevalent source and destination addresses:

[Table 13] Statistics of actual Sun RPC high port access alerts

Sources	# Alerts	Destinations	# Alerts
216.10.12.30	33	MY.NET.206.222	22

216.148.218.160	6	MY.NET.202.242	20
205.188.3.211	4	MY.NET.212.186	4
24.18.90.197	3	MY.NET.205.130	3
205.188.3.239	3	MY.NET.97.59	3

Analysis: It should be noted that since a vulnerable server would allow the attacker to gain root access, this attack is quite lethal!

Sample signatures:
10/04-05:49:29.920767 [**] Attempted Sun RPC high port access [**] 205.188.153.116:53 -> MY.NET.225.210:32771
09/28-13:28:03.304676 [**] SUNRPC highport access! [**] 24.18.90.197:4795 -> MY.NET.179.78:32771

Defensive recommendations:
- Immediately investigate all targets to see if system is compromised
- Disable the ghost portmapper service port (32771), if this port are indeed not required
- Uninstall ICQ, if this program are indeed not required
- Check that all systems are running with the latest patches
- Block traffics from attack addresses at the packet filter devices (Firewall, router, etc), if such traffics are indeed not required
- Include attack addresses into Watchlist for further monitoring
- Deploy the latest security product – IDS, Firewall, etc

7. Connect to 515 from inside
Alert description: This alert scans for LPD service on port 515. Most LPD services have several vulnerabilities such as buffer overflows or denial of service which the attack can execute arbitrary code as the root user. More detailed information can be found at CVE: CVE-2000-0232.

Statistics: There are 56 alerts from 2 sources to 3 destinations. The following table lists the most prevalent source and destination addresses:

[Table 14] Statistics of Connect to 515 from inside alerts

Sources	# Alerts	Destinations	# Alerts
MY.NET.101.142	54	MY.NET.100.3	54
MY.NET.179.78	2	64.244.202.66	1
		64.244.202.110	1

Analysis: It should be noted that MY.NET.101.142 performed fast scanning within short time interval. The below sample illustrates this point: (6 scans / 1 second).
11/19-13:56:31.876228 [**] connect to 515 from inside [**] MY.NET.101.142:1022 -> MY.NET.100.3:515
****
11/19-13:56:32.575642 [**] connect to 515 from inside [**] MY.NET.101.142:1022 -> MY.NET.100.3:515
However, if the LPD service was already patched, these traffics are false positives! Furthermore, it becomes more apparent because these sources triggered only this attack against the destination. High possibility of false positives!

Defensive recommendations:

- Immediately investigate the all targets to see if system is compromised
- Disable the LPD service port (515), if this port are indeed not required
- Check that all systems are running with the latest patches
- Block traffics from attack addresses at the packet filter devices (Firewall, router, etc), if such traffics are indeed not required
- Include attack addresses into Watchlist for further monitoring
- Deploy the latest security product – IDS, Firewall, etc

8. SMB Name Wildcard

Alert description: SMB Name Wildcard is a connection attempt to NetBIOS name service port 137. This traffic could be legitimate by Windows devices to find a hosts name. However, this traffic should be filtered at the perimeter because it can be used as a reconnaissance method to map out network and identify Windows devices such as shared directories and other services. More detailed information can be found at NetworkICE: 137.

Statistics: There are 218 alerts from 33 sources to 33 destinations. The following table lists the most prevalent source and destination addresses:

[Table 15] Statistics of SMB Name Wildcard alerts

Sources	# Alerts	Destinations	# Alerts
MY.NET.101.160	93	MY.NET.101.192	93
141.157.99.21	33	MY.NET.6.15	53
169.254.184.161	24	MY.NET.101.53	9
141.157.98.201	20	MY.NET.101.117	7
MY.NET.98.154	5	MY.NET.101.153	7

Analysis: It should be noted that there are probably lots of false positives - most alerts from internal network (MY.NET). Slow connections over long time interval could indicate false positives. The below sample illustrates this point: (10 connections / 6 hours)

10/10-11:40:04.616744 [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137
***
10/10-18:43:06.438109 [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137

Furthermore, it becomes more apparent because internal MY.NET addresses triggered only this attack against the destination. High possibility of false positives!
However, there are also actual alerts: NetBIOS traffics from external network. Such traffics should be blocked at perimeter.

Sample signatures:
11/20-01:14:27.821454 [**] SMB Name Wildcard [**] 141.157.99.21:137 -> MY.NET.6.15:137

Defensive recommendations:

- Immediately investigate the all targets to see if system is compromised
- Block the NetBIOS service port (137) at perimeter if this port are indeed not required
- Check that all systems are running with the latest patches
- Block traffics from attack addresses at the packet filter devices (Firewall, router, etc), if

such traffics are indeed not required
- Include attack addresses into Watchlist for further monitoring
- Deploy the latest security product – IDS, Firewall, etc

9. SNMP public access

Alert description: SNMP (Simple Network Management Protocol) is the protocol used to manage equipments in the Internet. However, if the default community string "public" is not changed, attacker can easily gather useful information such as system type and OS level, etc. More detailed information can be found at CVE: CVE-1999-0472, CAN-1999-0517.

Statistics: There are 218 alerts from 33 sources to 33 destinations. The following table summarizes the most prevalent source and destination addresses:

[Table 10] Statistics of SNMP public access alerts

Sources	# Alerts	Destinations	# Alerts	# Alerts (total)
MY.NET.98.106	58	MY.NET.101.192	468	561
MY.NET.98.174	49			
MY.NET.97.185	44			

Analysis: All sources addresses are internal network (MY.NET) and there is no clear evidence that these source addresses were compromised. Right??? Therefore, in my opinion, most alerts are false positives – system misconfiguration! Furthermore, it becomes more apparent because these sources triggered only this attack against the destination. High possibility of false positives!

Sample signatures:
11/11-10:35:48.256317 [**] SNMP public access [**] MY.NET.97.185:1322 -> MY.NET.101.192:161

Defensive recommendations:
- Immediately investigate the all targets to see if system is compromised
- Immediately change the default community string "public" to a more difficult string to guess

10. Back Orifice

Alert description: Back Orifice is a backdoor program commonly running at 31337 port. Scans on this port are usually searching for the target that has been already compromised by Back Orifice. More detailed information can be found at CVE: CAN-1999-0660.

Statistics: There are 1697 alerts from 40 sources to 932 destinations. The following table summarizes the most prevalent source and destination addresses:

[Table 17] Statistics of Back Orifice alerts

Sources	# Alerts	Destinations	# Alerts
62.136.90.120	306	MY.NET.98.150	7
63.46.46.143	291	MY.NET.97.208	7
203.148.182.108	111	MY.NET.98.81	6
213.43.69.72	99	MY.NET.98.82	6
203.155.130.111	79	MY.NET.98.77	6

<u>Analysis</u>: It should be noted that since a vulnerable server would allow the attacker full control of the system, this attack is quite lethal!

<u>Sample signatures</u>:
10/01-15:01:27.288758 [**] Back Orifice [**] 209.94.199.141:31338 -> MY.NET.60.34:31337

<u>Defensive recommendations</u>:

- Immediately investigate the all targets to see if system is compromised
- Disable Back Orifice backdoor port 31337
- Block the Back Orifice backdoor port 31337 at perimeter if this port are indeed not required
- Block traffics from attack addresses at the packet filter devices (Firewall, router, etc), if such traffics are indeed not required
- Include attack addresses into Watchlist for further monitoring
- Deploy the latest security product – IDS, Firewall, etc

11. Broadcast Ping to subnet 70

<u>Alert description</u>: If attacker pings the broadcast address, the live hosts on network will reply. This gives the attacker lists of the live hosts on the network. This also performs denial of service attack known as the Smurf against the spoofed victim. More detailed information can be found at CERT: http://www.cert.org/advisories/CA-98.01.smurf.html.

<u>Statistics</u>: There are 70 alerts from 40 sources to 932 destinations. The following table summarizes the most prevalent source and destination addresses:

[Table 18] Statistics of Broadcast Ping to subnet 70 alerts

Sources	# Alerts	Destinations	# Alerts
193.231.169.166	88	MY.NET.70.255	1813
193.226.60.179	55		
193.231.220.101	50		
213.154.131.131	49		
217.10.206.79	43		

<u>Sample signatures</u>:
10/03-14:48:07.021725 [**] Broadcast Ping to subnet 70 [**] 62.11.153.125 -> MY.NET.70.255

<u>Defensive recommendations</u>:

- Disable IP-directed broadcasts at perimeter
- Configure OS to prevent from responding to broadcast ICMP packets
- Include attack addresses into Watchlist for further monitoring
- Deploy the latest security product – IDS, Firewall, etc

12. TCP SMTP Source Port traffic

<u>Alert description</u>: This alert is suspicious because normal client/server program would initiate connection using a high source port (>1024). Therefore, it seems to be an attempt to evade the packet filter devices that allow port 25.

Statistics: There are 2893 alerts from 4 sources to 2836 destinations. The following table summarizes the most prevalent source and destination addresses:

[Table 19] Statistics of TCP SMTP Source Port traffic alerts

Source	# Alerts	Destinations	# Alerts
211.46.110.81	1789	MY.NET.145.98	2
24.7.227.215	1096	MY.NET.110.18	2
194.67.168.11	6	MY.NET.15.177	2
194.88.77.240	2	MY.NET.112.208	2

Analysis: It should be noted that most attackers performed pretty heavy scanning on the wide range of hosts within short time interval. The below sample illustrates this point: (1789 scans / 4.5 hours).

```
10/23-13:10:15.618101  [**] TCP SMTP Source Port traffic [**] 24.7.227.215:25 -> MY.NET.1.9:25
***
10/23-17:45:45.906329  [**] TCP SMTP Source Port traffic [**] 24.7.227.215:25 -> MY.NET.146.239:25
```

It indicates that most attackers used the automated scanning tool like Nmap.

Defensive recommendations:

- Immediately investigate the all targets to see if system is compromised
- Deploy packet filter devices that can analyze source port and destinations port
- Block traffics from attack addresses at the packet filter devices (Firewall, router, etc), if such traffics are indeed not required
- Include attack addresses into Watchlist for further monitoring
- Deploy the latest security product – IDS, Firewall, etc

13. WinGate 1080 Attempt

Alert description: WinGate or Socks is a popular proxy server for Windows running on 1080 or 8080 port. This alert indicates that attacker can possibly hide true source address as your address - surf anonymously. There are also several vulnerabilities with Wingate: CVE-1999-0290, CVE-1999-0291, CVE-1999-0441, CVE-1999-0494, CAN-1999-0657.

Statistics: There are 4764 alerts from 570 sources to 2655 destinations. The following table summarizes the most prevalent source and destination addresses:

[Table 20] Statistics of WinGate 1080 Attempt alerts

Source	# Alerts	Destinations	# Alerts
63.193.210.208	1883	MY.NET.206.118	372
208.194.161.155	220	MY.NET.225.154	126
198.63.2.192	179	MY.NET.60.11	67

Analysis: It is apparent that at least two destinations are running WinGate proxy servers: MY.NET.206.118:1080 and MY.NET.225.154:1080. There are sequential connections with the same source and destination addresses with incrementing source ports – legitimate WinGate traffics. The below sample illustrates this point:

```
10/04-02:51:49.554534  [**] WinGate 1080 Attempt [**] 24.214.18.65:2117 -> MY.NET.219.204:1080
```

```
10/04-02:51:50.233890 [**] WinGate 1080 Attempt [**] 24.214.18.65:2120 ->
MY.NET.219.204:1080
10/04-02:51:53.904376 [**] WinGate 1080 Attempt [**] 24.214.18.65:2138 ->
MY.NET.219.211:1080
10/04-02:51:55.762956 [**] WinGate 1080 Attempt [**] 24.214.18.65:2147 ->
MY.NET.219.212:1080
```

However, there are also actual scannings for WinGate proxy servers. The below sample illustrates this point: (1883 attempts / 5 minutes)

```
10/05-18:58:22.389439 [**] WinGate 1080 Attempt [**] 63.193.210.208:1605 ->
MY.NET.1.10:1080
***
10/05-19:03:42.376854 [**] WinGate 1080 Attempt [**] 63.193.210.208:2780 ->
MY.NET.254.249:1080
```

It indicates that most attackers used the automated scanning tool like Nmap.

Defensive recommendations:

- Immediately investigate all targets to see if system is compromised
- Disable the WinGate service port (1080 or 8080), if this port are indeed not required
- Check that all systems are running with the latest patches
- Block traffics from attack addresses at the packet filter devices (Firewall, router, etc), if such traffics are indeed not required
- Include attack addresses into Watchlist for further monitoring
- Deploy the latest security product – IDS, Firewall, etc

14. Watchlist connections

Alert description: This WatchList indicates past history of suspicious activities from Israel / China, and still needs to monitor suspicious activities from these addresses.

14.1. Watchlist 000220 IL-ISDNNET-990517

Statistics: There is a huge amount of traffics from Israel (ISDN Net Ltd., 212.179.0.0/17, hostmaster@isdn.net.il) - 30997 alerts from 61 sources to 108 destinations. The following table lists the most prevalent destination addresses and destination ports

[Table 21] Statistics of Watchlist 000220 IL-ISDNNET-990517 alerts

Destinations	# Alerts	Destination ports	# Alerts
MY.NET.211.146	4810	6699	9692
MY.NET.223.98	3938	4619	5733
MY.NET.206.90	3914	4922	4811

Analysis: lots of traffics are destined for port 6699 (Napster) which exchanges MP3 files. These could be legitimate Napster traffics, but sometimes hostile traffics looking for exploits.

Sample signatures:

```
10/05-16:56:00.844253 [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.66.2:7281 -> MY.NET.98.181:6699
```

This is correlated with Teri BidWell's GCIA practical.

14.2. Watchlist 000222 NET-NCFC

Statistics: There is also a huge amount of traffics from China (The Computer Network Center Chinese Academy of Sciences, 159.226.0.0/16, hlqian@NS.CNC.AC.CN) - 8134 alerts from 45 sources to 26 destinations. The following table lists the most prevalent destination addresses and ports:

[Table 22] Statistics of Watchlist 000222 NET-NCFC alerts

Destinations	# Alerts	Destination ports	# Alerts
MY.NET.6.7	5793	25	7823
MY.NET.100.230	1286	103	113
MY.NET.253.43	461	40627	70

Analysis: Lots of traffics are destined for port 25 (SMTP). These could be legitimate e-mail traffics, but sometimes actual attacks on the mail servers.

Sample signatures:

```
09/26-01:43:43.866602  [**] Watchlist 000222 NET-NCFC [**] 159.226.158.188:1249 ->
MY.NET.253.41:25
```

Defensive recommendations:

- Immediately investigate all targets to see if system is compromised
- Immediately investigate e-mail server
- Uninstall Napster, if this program are indeed not required
- Check that all systems are running with the latest patches
- Block traffics from attack addresses at the packet filter devices (Firewall, router, etc), if such traffics are indeed not required
- Deploy the latest security product – IDS, Firewall, etc

15. General port scans: spp_portscan

Alert description: The Snort portscan preprocessor raises alert if attacker would attempt a threshold number of connections within a given time interval. In this case, the threshold number of connections is 7 and the time interval is 2 seconds – but maybe not always. Attackers usually perform this general port scan for reconnaissance purposes.

Statistics: There are 27118 alerts from 1482 sources. The following table summarizes the most prevalent source and number of connections.

[Table 23] Statistics of General port scans alerts – spp_portscan

Sources	# Alerts
62.252.21.241	1761
63.248.55.245	1337
62.155.244.68	1054
63.88.175.201	973
216.191.162.145	925

Analysis: With this information the attacker can determine an appropriate attack against the open ports.

Sample signatures:

```
09/27-05:51:47.435678  [**] spp_portscan: PORTSCAN DETECTED from 24.28.2.123
(THRESHOLD 7 connections in 2 seconds) [**]
09/27-05:51:49.479475  [**] spp_portscan: portscan status from 24.28.2.123: 16
```

```
connections across 16 hosts: TCP(16), UDP(0) [**]
09/27-05:51:51.366990   [**] spp_portscan: End of portscan from 24.28.2.123 (TOTAL
HOSTS:17 TCP:16 UDP:0) [**]
```

Defensive recommendations:

- Immediately investigate the all targets to see if system is compromised
- Check that all systems are running with the latest patches
- Block traffics from attack addresses at the packet filter devices (Firewall, router, etc), if such traffics are indeed not required
- Include attack addresses into Watchlist for further monitoring
- Deploy the latest security product – IDS, Firewall, etc

Overall summary of Snort scan logs

The following table presents the overall summary of Top 10 suspicious scans on MY.NET network. This table clearly shows a huge amount of hostile traffic - 310477. There are probably lots of legitimate traffics -false positives. However, it is also apparent that there are lots of hostile activities needing further investigation.

Total number of signatures: 256
Total number of alerts: 310447
Time interval: 09/227/2000 – 11/23/2000
[Table 24] Statistics of Top 10 Snort Scan signatures

Signature	# Alerts	# Sources	# Destinations
TCP **S***** scan	235361	278	35788
TCP **SF**** scan	50523	26	24919
UDP scan	21585	84	1420
TCP ***F**** scan	454	28	369
TCP *****P** scan	351	4	349
TCP **S*R*A* scan	281	16	5
TCP ******** scan	221	166	160
TCP 21S***** scan	104	21	38
TCP ***FR*A* scan	57	36	40
TCP *1SF*P** scan	29	10	10

Analysis: Attackers scanned on most entire MY.NET network for every port with every combination of TCP flags. The following raw Snort data from SOOS* file illustrates this point:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+=+=+=+
11/04-01:32:25.296798 133.46.212.81:4940 -> MY.NET.211.146:4922
TCP TTL:110 TOS:0x0 ID:28253  DF
**SFRP*U Seq: 0xACD8F5   Ack: 0x11032B   Win: 0x5010
3E 2F 50 10 22 38 9C B2 00 00 00 00 00 00        >/P."8........
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+=+=+=+
11/04-01:47:25.326563 133.46.212.81:4940 -> MY.NET.211.146:4922
TCP TTL:110 TOS:0x0 ID:20804  DF
21SF*P*U Seq: 0x1100AE   Ack: 0x374503A2   Win: 0x5010
```

```
TCP Options => EOL EOL EOL EOL EOL EOL EOL NOP
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
+=+=+=+=+=+
11/04-01:59:05.250340 133.46.212.81:0 -> MY.NET.211.146:1738
TCP TTL:110 TOS:0x0 ID:6890  DF
21SF**** Seq: 0x133A00CB   Ack: 0x2020030B   Win: 0x218
TCP Options => EOL EOL EOL EOL EOL EOL WS: 1 NOP TS: 196608 0 EOL EOL EOL
EOL
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
+=+=+=+=+=+
11/04-02:10:05.588750 133.46.212.81:1867 -> MY.NET.211.146:4922
TCP TTL:110 TOS:0x0 ID:19543  DF
2*SF*PA* Seq: 0xD58F30   Ack: 0x50315   Win: 0x5B4
00 D5 8F 30 00 05 03 15 1A 5B 05 B4 6C 94 16 3A  ...0.....[..l..:
00 00 00 00 00 00                                ......
(truncated --)
```

The following table lists Top 5 Scan attacker and target addresses.

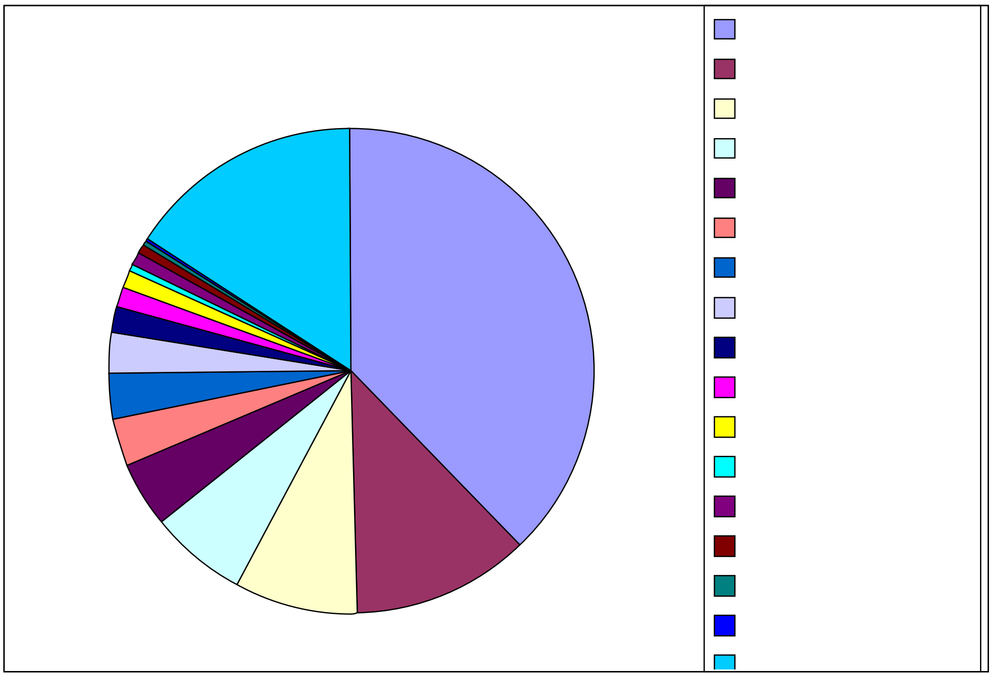
[Table 25] Top 5 Scan attackers and targets

Sources	Whois	# Alerts	Destinations	# Alerts
62.157.23.237	Deutsche Telekom AG, DE	9641	MY.NET.218.50	2359
63.248.55.245	Flashcom, Inc	9073	MY.NET.253.114	1976
62.96.169.86	neue mediengesellschaft ulm mbh	8939	MY.NET.206.94	1799
24.23.151.112	@Home Network	8763	MY.NET.162.77	1759
64.50.161.162	CapuNet, LLC	8635	MY.NET.120.36	1591

Analysis: A complete investigation into these prevalent attackers and targets needs to minimize the impact of the associated risks.
Most attackers scanned for known vulnerable services (FTP, DNS, rpc.statd) or for already compromised ports (SubSeven, Back Orifice, other Trojans/backdoors). The following table lists the most popular destination ports.

[Table 26] Top 16 Scan destination ports

Destination ports	# Alerts	Destination ports	# Alerts
21 (FTP control)	117678	139 (NetBIOS)	5648
27374 (SubSeven Trojan)	36214	113 (identd/auth)	4244
515 (Line printer)	25797	23 (telnet)	3044
53 (DNS)	19513	67 (Bootps)	2295
9704 (Linux rpc.statd)	14168	19000 (N/A)	2081
98 (linuxconf)	9467	1080 (WinGate)	1895
9088 (N/A)	8763	31337 (Back Orifice)	1217
110 (POP)	8685	5232 (N/A)	944



[Figure 2] Distribution of Top 16 Scan destination ports

Through these ports, an attacker can gain root access by exploiting vulnerable or backdoor problem. Quite lethal! More detailed information about vulnerabilities in each port can be found at hyperlink (NetworkICE port knowledgebase).

Overall summary of Snort alert logs from internal network

The following table presents the overall summary of suspicious alerts from internal network (MY.NET). Alerts from internal network indicate that internal attacker hosts have probably been compromised! Quite lethal!

Total number of signatures: 3
Total number of alerts: 646
Time interval: 10/01/2000 – 11/22/2000
[Table 27] Statistics of Snort Alert signatures from internal network

Signature	# Alerts	# Sources	# Destinations
Connect to 515 from inside	56	2	3
SMB Name Wildcard	122	17	18
SNMP public access	468	23	1

The following table lists the most prevalent source addresses.

[Table 28] Top 5 Alert - internal hosts

Internal sources	# Alerts
MY.NET.101.160	93
MY.NET.98.106	58
MY.NET.101.142	54

MY.NET.98.174	49
MY.NET.97.185	44

Analysis: In the previous specific alert sections, these alerts were already analyzed. Again, these internal hosts were not probably compromised because these alerts could be highly false positives. Refer to each specific alert section. However, it should be noted that a complete investigation into these internal hosts and targets needs to minimize the impact of the associated risks.
The following table lists all internal scanner - General Port Scan alerts (spp_portscan)

[Table 29] General port scan - internal hosts

Internal sources	# Alerts	Internal sources	# Alerts
MY.NET.5.25	116	MY.NET.110.105	2
MY.NET.1.3	59	MY.NET.109.41	2
MY.NET.221.82	21	MY.NET.109.40	2
MY.NET.1.4	5	MY.NET.109.38	2
MY.NET.152.165	3	MY.NET.99.120	1
MY.NET.101.1	3	MY.NET.19.10	1
MY.NET.110.16	2	MY.NET.110.108	1
MY.NET.110.111	2		

Analysis: It should be noted that most of these internal scanning hosts have been already scanned or attacked by other external hosts. The below sample illustrates this point: MY.NET.109.4
10/02-06:36:14.947776 [**] SYN-FIN scan! [**] 208.61.4.207:9704 -> MY.NET.109. 41:9704
10/23-16:25:38.423139 [**] TCP SMTP Source Port traffic [**] 24.7.227.215:25 -> MY.NET.109.41:25
Through these previous scanning or attacking techniques, external hosts probably succeeded in compromising internal hosts, and then tried to scan other internal hosts from the already compromised internal hosts. Quite lethal!

Overall summary of Snort scan logs from internal network

The following table presents the overall summary of suspicious alerts from internal network (MY.NET). Alerts from internal network indicate that internal attacker hosts have probably been compromised! Quite lethal!

Total number of signatures: 3
Total number of alerts: 10258
Time interval: 09/27/2000 – 11/23/2000
[Table 30] Statistics of Snort Scan signatures from internal network

Signature	# Alerts	# Sources	# Destinations
TCP 2**FR*A* scan	5	2	5
UDP scan	4511	14	574
TCP **S***** scan	5742	3	5511

Analysis: It should be noted that # sources are much smaller than # destinations. It indicates that each internal attack host performed heavy scanning on lots of other internal hosts. In other words, most internal scanning hosts have already been compromised! Quite lethal! Furthermore, most attackers performed pretty heavy scanning on the wide range of hosts within short time interval. The below sample illustrates this point: (2982 scans / 5 minutes).

```
Nov  2 16:13:52 MY.NET.224.150:2094 -> MY.NET.0.15:139 SYN **S*****
***
Nov  2 16:18:57 MY.NET.224.150:1883 -> MY.NET.255.205:139 SYN **S*****
```

It indicates that most attackers used the automated scanning tool like Nmap.

The following table lists all internal Scan hosts.

[Table 31] Scan Internal hosts

Internal sources	# Alerts	Internal sources	# Alerts
MY.NET.224.150	2981	MY.NET.110.109	120
MY.NET.221.82	2668	MY.NET.109.40	109
MY.NET.5.25	2311	MY.NET.110.110	100
MY.NET.1.3	577	MY.NET.213.58	94
MY.NET.110.111	270	MY.NET.109.38	93
MY.NET.110.16	267	MY.NET.1.4	22
MY.NET.109.41	252	MY.NET.152.165	14
MY.NET.110.105	215	MY.NET.101.1	4
MY.NET.110.108	160	MY.NET.19.10	1

Analysis: It should be noted that most of these internal scanning hosts have been already scanned or attacked by external hosts. The below sample illustrates this point: MY.NET.221.82

```
10/16-16:55:26.342617  [**] site exec - Possible wu-ftpd exploit - GIAC000623 [*
*] 24.31.88.99:62275 -> MY.NET.221.82:21
10/16-16:57:49.491247  [**] site exec - Possible wu-ftpd exploit - GIAC000623 [*
*] 24.31.88.99:62281 -> MY.NET.221.82:21
11/03-10:30:29.843211  [**] SYN-FIN scan! [**] 195.103.69.159:53 -> MY.NET.221.
82:53
```

Through these previous scanning or attacking techniques, external hosts probably succeeded in compromising internal hosts, and then tried to scan other internal hosts from the already compromised internal hosts.

Probably compromised hosts

The previous two sections show that the following hosts have possibly been compromised.

[Table 32] Probably compromised internal hosts – 19 hosts

Internal sources	Internal sources	Internal sources	Internal sources
MY.NET.224.150	MY.NET.110.16	MY.NET.109.40	MY.NET.152.165
MY.NET.221.82	MY.NET.109.41	MY.NET.110.110	MY.NET.101.1
MY.NET.5.25	MY.NET.110.105	MY.NET.213.58	MY.NET.19.10
MY.NET.1.3	MY.NET.110.108	MY.NET.109.38	MY.NET.99.120
MY.NET.110.111	MY.NET.110.109	MY.NET.1.4	

Analysis: It should be noted that the results of the previous two sections are very similar – only 3 hosts are different.

Defensive recommendations:

- Immediately investigate these hosts to see if system is compromised. Very important!
- Follow the "Incident Handling & Forensics procedures"

Summary and recommendations

This overall security analysis report shows that a huge amount of hostile activities happened in MY.NET network:

- Pretty heavy reconnaissance scans: OS fingerprinting and Port scanning
- Lots of attack attempts: known vulnerable program / services, backdoor, virus, etc.
- Probably several compromised host: quite lethal!
- Misconfigured system: default SNMP community string "public"
- Probably hostile program: ICQ, Napster

Therefore, I suggest the following defensive recommendations:

- Immediately investigate all targets (especially [Table 32]-Probably compromised hosts) to see if system is compromised
- Follow the "Incident Handling & Forensics procedures"
- Disable the service port (especially vulnerable service ports and known backdoor port), if this port are indeed not required
- Check that all systems are running with the latest patches
- Check that all systems are not misconfigured – Change default SNMP community string "public"
- Block traffics from attack addresses at the packet filter devices (Firewall, router, etc), if such traffics are indeed not required
- Uninstall probably hostile program, if this program are indeed not required
- Include attack addresses into Watchlist for further monitoring
- Deploy the latest security product – state-full IDS and Firewall, anti-virus program.

*** Assignment 3 – Analysis Process (30 Points) ***

First, I searched for the previous SANS GIAC practicals. Then, I downloaded them and referred to most previous reports - mainly honor reports. Especially, I referred to Teri BidWell's GCIA practical. Thanks ~ ↑↑.

I completed this assignment through the following steps (** Note – This is my final result. Of course, I tried many other techniques but have some problems with them).

Step 0. Download Snort data (huge amount of data!)
Step 1. Eliminate duplications (UNIX command: #diff file1 file 2)

```
SnortA14.txt - SnortA19.txt
SnortS20.txt - SnortS23.txt
OOSche4.txt - OOSche5.txt
```

Step 2. Combine Snort report files, eliminate duplication and sort

```
#cat SnortA* | uniq | sort > Alert.all
#cat SnortS* | uniq | sort > Scan.all
```

```
#cat O* | uniq | sort > Oos.all
```

Step 3. Change MY.NET to 255.255 for Snortsnarf analysis

```
vi command mode: %$ s/MY.NET/255.255/g
```

Step 4. Combine similar alerts

SITE EXEC – Possible wu-ftpd exploit - GIAC000623 + site exec – Possible wu-ftpd exploit - GIAC000623

SUNRPC highport access! + Attempted Sun RPC high port access

Step 5. Run Snort data analysis tool: SnortSnarf (http://www.silicondefense.com/snortsnarf/). For a long time – over 24 hours with my computer (550 MHz CPU, 128 MB RAM)

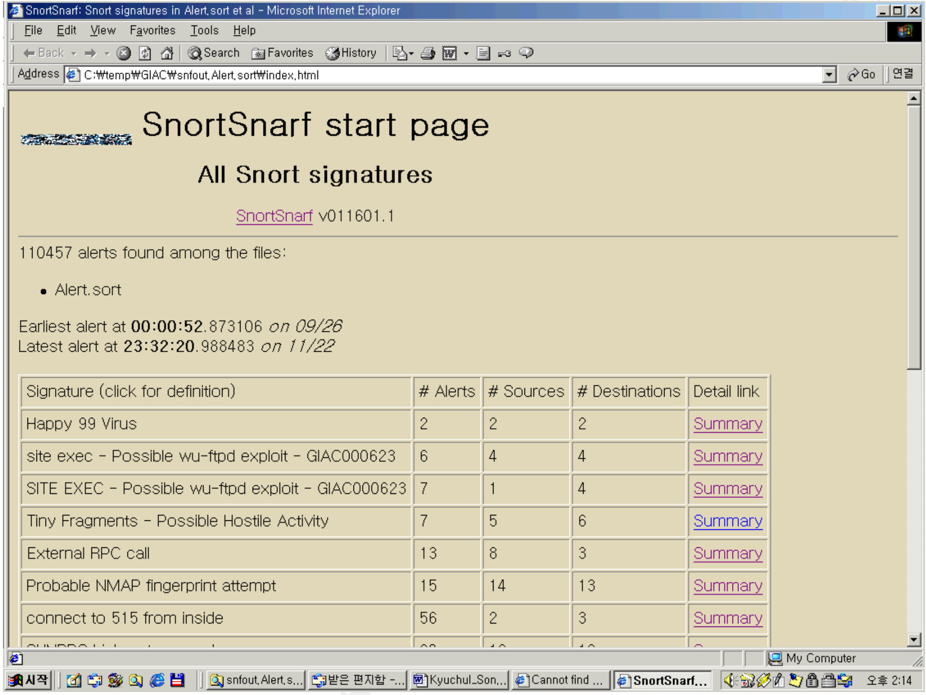
```
#perl snortsnarf.pl Alert.all
#perl snortsnarf.pl Scan.all
```

Step 6. Use SnortSnarf result (very useful information!)

Statistics of Snort alert reports

Statistics of Snort scan reports

Statistics of each specific alert



SnortSnarf: Snort signatures in Alert.sort et al - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:₩temp₩GIAC₩snfout.Alert.sort₩index.html
Go 연결
My Computer
시작 snfout.Alert.s... 받은 편지함 -... Kyuchul_Son... Cannot find ... SnortSnarf...
오후 2:14

SnortSnarf start page

All Snort signatures

SnortSnarf v011601.1

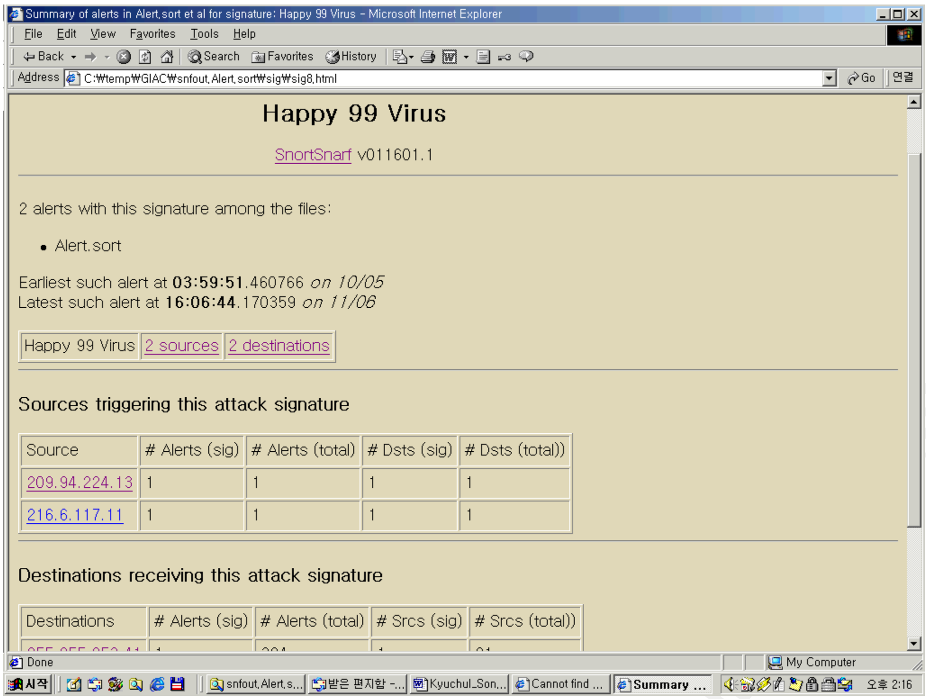
110457 alerts found among the files:

- Alert.sort

Earliest alert at **00:00:52**.873106 *on 09/26*
Latest alert at **23:32:20**.988483 *on 11/22*

Signature (click for definition)	# Alerts	# Sources	# Destinations	Detail link
Happy 99 Virus	2	2	2	Summary
site exec - Possible wu-ftpd exploit - GIAC000623	6	4	4	Summary
SITE EXEC - Possible wu-ftpd exploit - GIAC000623	7	1	4	Summary
Tiny Fragments - Possible Hostile Activity	7	5	6	Summary
External RPC call	13	8	3	Summary
Probable NMAP fingerprint attempt	15	14	13	Summary
connect to 515 from inside	56	2	3	Summary

[Figure 1] Snapshot of SnortSnarf – Overall statistics of all signatures



Happy 99 Virus

SnortSnarf v011601.1

2 alerts with this signature among the files:

- Alert.sort

Earliest such alert at **03:59:51**.460766 *on 10/05*
Latest such alert at **16:06:44**.170359 *on 11/06*

Happy 99 Virus	2 sources	2 destinations

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
209.94.224.13	1	1	1	1
216.6.117.11	1	1	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))

Summary of alerts in Alert.sort et al for signature: Happy 99 Virus - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:₩temp₩GIAC₩snfout.Alert.sort₩sig₩sig8.html
Go
Done
My Computer
시작
snfout.Alert.s...
받은 편지함 -...
Kyuchul_Son...
Cannot find ...
Summary ...
오후 2:16

[Figure 2] Snapshot of SnortSnarf – Overall statistics of specific signature

Step 7. Modify Terri BidWel scripts to find overall prevalent source and destination addresses and ports from Snort alerts

```
#alert.sh
#change [**] to &
cat $1 | sed s/"\[\*\*\]"/"\&"/g > $1.d
#change -> to &
cat $1.d | sed s/"\->"/"\&"/g > $1.del
# get source address and port
cat $1.del | awk -F"&" '{print $3}' > $1.src
# get destination address and port
cat $1.del | awk -F"&" '{print $4}' > $1.dst
# get source address and count the entries and sort
cat $1.src | awk -F":" '{print $1}' | sort | uniq -c | sort -r > $1.srci
# get source port and count the entries and sort
cat $1.src | awk -F":" '{print $2}' | sort | uniq -c | sort -r > $1.srcp
# get destination address and count the entries and sort
cat $1.dst | awk -F":" '{print $1}' | sort | uniq -c | sort -r > $1.dsti
# get destination port and count the entries and sort
cat $1.dst | awk -F":" '{print $2}' | sort | uniq -c | sort -r > $1.dstp
rm $1.d
rm $1.del
rm $1.src
rm $1.dst
#alert.sh Alert.all
==> Alert.all.srci + Alert.all.srcp + Alert.all.dsti + Alert.all.dstp
```

Step 8. Modify Teri Bidwell scripts to find overall prevalent source and destination addresses and ports from Snort scans

```
#scan.sh
```

```
# change -> to space
cat $1 | sed s/"\->"/" "/g > $1.del
# get source address and port
cat $1.del | awk -F" " '{print $4}' > $1.src
# get destination address and port
cat $1.del | awk -F" " '{print $5}' > $1.dst
# get source address and and count the entries and sort
cat $1.src | awk -F":" '{print $1}' | sort | uniq -c | sort -r > $1.srci
# get source port and and count the entries and sort
cat $1.src | awk -F":" '{print $2}' | sort | uniq -c | sort -r > $1.srcp
# get destination address and and count the entries and sort
cat $1.dst | awk -F":" '{print $1}' | sort | uniq -c | sort -r > $1.dsti
# get destination port and and count the entries and sort
cat $1.dst | awk -F":" '{print $2}' | sort | uniq -c | sort -r > $1.dstp
rm $1.del
rm $1.src
rm $1.dst
#scan.sh Scan.all
==> Scan.all.srci + Scan.all.srcp + Scan.all.dsti + Scan.all.dstp
```

Step 9. Combine OS fingerprinting alerts to find overall prevalent source and destination addresses and ports from OS fingerprinting alerts - Probable NMAP fingerprint attempt + Queso fingerprint + Null scan!

```
#grep Probable Alert.all > probable
#grep Queso Alert.all > queso
#grep Null Alert.all > null
#cat probable queso null > fingerprint
#alert.sh fingerprint
==> fingerprint.srci + fingerprint.srcp + fingerprint.dsti + fingerprint.dstp
```

Step 10. Combine Port scanning alerts to overall prevalent source and destination addresses and ports from Port scanning alerts: SYN-FIN scan! + NMAP TCP ping!

Similar to Step 9.

Step 11. Find SUN RPC alerts traffics from non-icq source port and to find overall prevalent source and destination addresses and ports from actual SUN RPC alerts

```
#grep –v :4000 rpc.all > rpc.no.icq
#alert.sh rpc.no.icq
==> rpc.no.icq.srci + rpc.no.icq.srcp + rpc.no.icq.dsti + rpc.no.icq.dstp
```

Step 12. Find source addresses in the General port scan – spp_portscan

```
#spp.sh
# get spp_portscan in Alert.all
grep status $1 > spp
# get source address and and count the entries and sort
cat port | awk -F" " '{print $7}' | sed s/":"/" "/g | sort | uniq -c | sort -r >
 spp.src
#spp.sh Alert.all
==> spp.src
```

Step 13. Find alerts from internal network (MY.NET) and find overall prevalent source and destination addresses and ports

```
#grep "] MY.NET" Alert.all > Alert.MY
#alert.sh Alert.MY
==> Alert.MY.srci + Alert.MY.srcp + Alert.MY.dsti + Alert.MY.dstp
```

Step 14. Find General Port Scan (spp_portscan) from internal network (MY.NET) and find overall prevalent source address

```
#grep MY.NET spp.src | sort | uniq -c | sort -r > spp.MY
```

Step 15. Find scans from internal network (MY.NET) and find overall prevalent source and destination addresses and ports

```
#scan.my.sh
# change “> “ to &
cat $1 | sed s/"> "/"\&"/g > $1.temp1
# get scans from internal network (MY.NET)
grep “ MY.NET” $1.temp1 > $1.temp2
# change & to space
cat $1.temp2 | sed s/"\&"/"> "/g > $1.MY
rm $1.temp1
rm $1.temp2
#scan.my.sh Scan.all
==> Scan.all.MY
```

Others

- whois, dns service in the Snortsnarf
- Excel – mainly for distribution graphic
- vi, cat, sort, grep, awk, egrep, sed, wc, uniq, and other UNIX commands
- Correlated data and analysis report - SANS: http://www.sans.org/search.htm
- Alerts analysis
  - CVE lists: http://mitre.cve.org
  - SecurityFocus: www.securityfocus.com
  - SANS : Information Security Reading Room and Intrusion Detection FAQ
- Vulnerable and backdoor ports lists
  - NetworkICE: http://advice.networkice.com/advice/Exploits/Ports/
  - SANS: ID FAQ - What port numbers do well-known trojan horses use?