Intrusion Detection Practical Assignment
by Michael Semling

1 Assignment 1 - Network Detects

1.1 Introduction
This document contains 4 detects which were detected and logged at our monitored network.

We use several sensors which collect the data in tcpdump format. These are analysed with SNORT v1.7
(www.clark.net/~roesch/security.html) and sent periodically to a central server with various tools (snortstat, snortsnarf...) and
ACID on a MYSQL database. If there is data which needs more investigation, the tcpdump data is available for some time on the
sensors. We analyse dumped packets either directly with TCPDUMP or for a more readable format we use ethereal
(http://www.ethereal.com).

The time is MET. All IP addresses are obfuscated to 192.168.x.x or 10.x.x.x format.

1.2 Detection and analysis
The detection and analysis consists of following points from the guidelines
(http://www.sans.org/giac/b/ID_assignment_guidelines.htm):

1. The source of the trace
2. Detection trigger, type of event generator, the log format
3. Probability the source address was spoofed
4. Description of the attack
5. Attack mechanism
6. Correlation
7. Evidence of active targeting
8. Severity
9. Defensive recommendation
10. Multiple choice test
The format of SNORT is:

e.g.
[**] IDS159 - PING Microsoft Windows [**]
01/15-21:33:37.971780 0:E0:F7:4C:5C:D8 - 0:C0:4F:A9:BB:FA type:0x800 len:0x4A
192.168.11.11 - 192.168.11.110 ICMP TTL:22 TOS:0 ID:48640 IpLen:20 DgmLen:60
ID:512 Seq:1280 ECHO
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 qrsuvwabcdefghi

First line:
[**] Snort rule which triggers the log, [**], if there is an IDS number, explanations can be found under (http://whitehats.com/IDS/<number>) (in this case http://whitehats.com/IDS/159)

Second line:
Date (Month/Day) – time (hh:mm:ss:µµµµµ)
Source MAC (Media Access Control) address – Destination MAC address
type: Ethernet frame type field. 0x800 is IP.
len: is the total length of the Ethernet frame in (hex) Bytes without the final CRC

Third line:
Source IP address
direction ( - bi-directional, -> from left to right)
Destination IP address
IP protocol (ARP(Address Resolution Protocol), ICMP (Internet Control Message Protocol), UDP (User Datagram Protocol), TCP (Transfer Control Protocol),...)
TTL is the Time To Live counter, decremented by each hop from a router.
TOS is the Type Of Service.
ID is the IP IDentification value.
IP Len is the header Length in Bytes.
DgmLen is the Datagram Length.
The DF/MF (Don't Fragment / More Fragment) Flags are at the end of the third line.

Forth line:
This line depends on the protocol: For TCP
Flags (R1,R2, U,A,P,R,S,F)
Seq is the sequence number
Ack is the responding acknowledgement number
Win is the advertised window size
TCPLen is the Length of the TCP packet
TCP Options indicate the different TCP options.
For ICMP
ID is the ICMP identifier.
Seq is the ICMP sequence number.
ICMP type of packet (ECHO)
TCPDUMP format with / without Ethernet information:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>time (hh:mm:ss:mmm)</td>
<td>Source MAC address – Destination MAC address</td>
</tr>
<tr>
<td>Ethernet information</td>
<td>Time and length (minimum 60 / 0x3c)</td>
</tr>
<tr>
<td>source IP address, source port &gt; destination IP address, destination port</td>
<td>TCP flags</td>
</tr>
<tr>
<td>TCP start sequence number</td>
<td>TCP end sequence number &lt;br&gt; (Size)</td>
</tr>
<tr>
<td>ack and number which indicates the next block of data with this sequence number</td>
<td>win Advertised window size</td>
</tr>
<tr>
<td>IP Flags (DF)</td>
<td>TTL time to live value. See Snort.</td>
</tr>
<tr>
<td>ID IP identifier</td>
<td>Hex dump and ASCII representation.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1.3 Detects

1.3.1 Detect 1: BackOrifice 2000

1.3.1.1 The source of the trace

The source was found on the unprotected internet. The data was analysed with snort, then a special filter with tcpdump has been applied to the sensors data.

1.3.1.2 Detection trigger

The attacks were detected by snort with the filter rules.

```
alert tcp any 54321 <> $EXTERNAL_NET any (msg:"IDS189 - BACKDOOR ACTIVITY - Possible BackOrifice 2000"; flags:SA;)
```

Alert with the message if there was a TCP packet from any network address from and to IP addresses whose numbers belong to our range, called external network, using port 54321 with a successful second part of the handshake (SYN/ACK set).

These rules just look for ports, if there is an alert I apply independently a second check from SANS conference, book "3.2 Intrusion Detection and Packet Filing: How it works by Vicki Irwin and Hal Pomeranz, page 218/222". It will be increase the probability that this captured data stands for a BO2k session and helps to analyse the traffic flow. The filter is.

The UDP BO2k filter is: 

```
(udp[10:2] = 0) and ((ip[2:2] - (((ip[0] & 0x0f) * 4) - 8 - 4) = (udp[8:1] * 256 + udp[8:1]))
```

The TCP BO2k filter is (without TCP options):

```
(tcp[22:2] = 0) and ((ip[2:2] - (((ip[0] & 0x0f)*4) - 20 - 4) = (tcp[21]*256+tcp[20]))
```

1.3.1.3 Probability the source address was spoofed

As we can see in the dump, data was exchanged so the probability of a spoofed address is very low.

1.3.1.4 Description of the attack

This is probably BO2k (http://www.bo2k.com/) in the TCP mode. Snort reported the second part of the three way handshake.

```
[**] IDS189 - BACKDOOR ACTIVITY - Possible BackOrifice 2000 [**]
01/15 12:05:15.096608 0:20:AF:E3:7C:27 -> 0:20:AF:7C:7C:C8 type:0x800 len:0x3C
***A**S* Seq: 0x4A95E6 Ack: 0x4B26B9 W in: 0x2238 TcapLen: 24
```

Here the tcpdump output when the filter above was applied, italic bold is the tcp part.

For illustration purpose I just took two packets out of the sequence, which appeared after applying the filter. The first two refer to the snort alert with the default port 54321.

```
   4500 003e ea01 4000 8006 f4e c0a8 c80b
c0a8 c80b D431 0409 0043 8001 0044 11c0
5018 2141 684e 0000 1200 0000 a1fd 7a56
90f8 9a9b 990b b99b 990e
```

1.3.2 Key fingerprint

There are more data after filtering. The server seems to have changed the port to 21501, the IP addresses are the same (again just two packets for illustration, I dropped the rest).


1.3.1.5 Attack mechanism

Generally, BackOrifice 2000 is a Trojan horse with a lot of configuration possibilities. Remote control is possible when the ‘server’ is installed on a machine.

In this case, The attacker seems to know that this system has been compromised by Back Orifice 2000. It seems, the first contact occurs on the default port. Then the server might have been changed to another port. This can be done remotely.

1.3.1.6 Correlation


1.3.1.7 Evidence of active targeting

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.
These packets are coming from one source and going to one target. 100% active targeting.

### 1.3.1.8 Severity

\[
\text{Severity} = (\text{Critical} + \text{Lethal}) - (\text{System} + \text{Net countermeasures})
\]

- **Critical**: Desktop is compromised, 2
- **Lethal**: User access / remote administration / Root over the net, 5
- **System countermeasures**: Don't know but Trojan has been installed so none, 1
- **Net countermeasures**: There seems to be no firewall, 1

\[
\text{Severity} = (2 + 5) - (1 + 1) = 5
\]

### 1.3.1.9 Defensive recommendation

- Virus / E-mail scanner
- Desktop firewall
- No unprotected internet access

### 1.3.1.10 Multiple choice test

**BackOrifice 2000**

A) is simple to detect because it uses always the same port.
B) should not be used because everything is in plain text.
C) can also use the port 80 (HTTP) to pass through firewalls.
D) can be detected with the two filters above.

The correct answer is C).

Not correct because:

- A) Ports can be freely chosen.
- B) There are several plugins to do cryptography.
- D) Only if there is no TCP option set.

### 1.3.2 Detect 2: Slow FIN/ACK scan

#### 1.3.2.1 The source of the trace

The following data comes from our insecure network which is separated from the internet by a permissive firewall.

#### 1.3.2.2 Detection trigger

The trigger was snort sending a general alert reporting traffic to unused addresses from a specific source IP address. I analysed the sensor data with a simple tcpdump filter looking for this particular source IP address (IP and (host 192.168.141.27)) . I took just a window out of the data from which I think the scan is well recognisable.

#### 1.3.2.3 Probability the source address was spoofed
The source address was probably not spoofed or in the same routed network class to see the results.

1.3.2.4 Description of the attack

This is a slow scan to find machines. This is NOT a FIN scan. No information will be collected about the OS or if the port 53 (DNS) or 80 (HTTP) is open or not on the scanned machines. If a machine is up it will answer with Reset to this request.

```
19:09:51.482834 192.168.141.27.2084 > 192.168.140.10.53: F 55135905:55135905(0) ack 167535093 win 512 (DF) (¶ 53)
19:09:51.483426 192.168.140.10.53: > 192.168.141.27.2084: R 167535093:167535093(0) win 0 (DF) (¶ 60)
19:09:51.488515 192.168.141.27.2084 > 192.168.140.10.80: F 55135905:55135905(0) ack 167535093 win 512 (DF) (¶ 53)
19:09:51.489159 192.168.140.10.80 > 192.168.141.27.2084: R 167535093:167535093(0) win 0 (DF) (¶ 60)
19:10:41.502842 192.168.141.27.1808 > 192.168.140.11.53: F 1454240847:1454240847(0) ac k 1520589986 win 512 (¶ 53)
19:10:41.509029 192.168.140.11.53: > 192.168.141.27.1808: R 1520589986:1520589986(0) win 0 (¶ 56)
19:10:41.514027 192.168.141.27.1808 > 192.168.140.11.80: F 1454240847:1454240847(0) ack 1520589986 win 512 (¶ 53)
19:10:41.515075 192.168.141.27.1808 > 192.168.140.11.80: R 1520589986:1520589986(0) win 0 (¶ 56)
19:11:31.532897 192.168.141.27.1045 > 192.168.140.12.53: F 1777954818:1777954818(0) ack 1798268799 win 512 (¶ 53)
......
```

1.3.2.5 Attack mechanism

This is a slow FIN-ACK scan, the time for each host is about 50 seconds. For the same target IP address, the same source port, the same ACK number and the same sequence number is used. This are crafted packets.

For each target IP address the source port, the ACK and sequence number seem to be chosen randomly.

As we see host 192.168.140.12 is not up. Machines that are up will respond with a Reset. Not available host will be seen by the ICMP destination unreachable sent from the router or, if the router drops this message, because the machine does not answer.

Only two and low ports (53, 80) are tried. A possible explanation is, that a firewall would deny packets not on port 80 to a web server. And the only reason for this kind of FIN/ACK scan might be that this kind of packets pass a firewall that blocks incoming packets without an ACK. This doesn’t make any sense to let open the port 80 for a web server but blocking packets without any ACK so no incoming packets are allowed. It might be try not being detected by a sniffer which might have port 80 excluded.

1.3.2.6 Correlation

NMAP port scanner: (http://www.insecure.org/nmap/)

ICMP tools: (http://www.securityfocus.com/frames/?focus=ids&content=/focus/ids/articles/icmptools.html)

1.3.2.7 Evidence of active targeting

This attack (scan) was generated at this specific host that targets (actively) the range of the network. Because this is a network topology scan this is not an active targeting.

1.3.2.8 Severity

Severity = (Critical + Lethal) – (System + Net countermeasures)
Critical: Unix/Windows machines, 2
Lethal: The scan is successful, this is a pre-attack probe, 1
System countermeasures: There are systems with older OS, no patches, 3
Net countermeasures: permissive firewall, 2

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

1.3.2.9 Defensive recommendation
- Stateful firewall / one way firewall (not based on the ACK)

1.3.2.10 Multiple choice test

The trace above is a sign for
A) Known backdoors and Trojan horses.
B) Host mapping technique.
C) Can be detected by SNORT with the preprocessor rule
   * preprocessor ports can: 192.168.241.0/24 5 10 /var/log/snort_portscan.log *
   # | | | | | | |
   # | | | | | | |
   #Your IP address or Net work here ------ + | | | | | | |
   #Amount of ports being connected ------ + | | | | | | |
   # in this Interval (in seconds) ---------- + | | | | | | |
   # Log file (path/name) ------------------------------- +

D) This is the so called half open scan.

Correct Answer: B), this is a FIN -ACK scan, reachable machines will answer with a Reset.

Not correct because:
A) TCP port 53 (DNS) and TCP port 80 (HTTP) could be used by Trojans but DNS and HTTP uses them in most cases. A FIN -ACK is normally a sign for end of a tcp connection.
C) The interval in seconds is too small and the amount of ports was not reached.
D) The half open scan starts with a SYN and if a SYN/ACK returns a Reset is sent not completing the three way handshake.
### 1.3.3 Detect 3: Lot of alerts

#### 1.3.3.1 The source of the trace

Insecure network, protected with a permissive firewall.

#### 1.3.3.2 Detection trigger

The snort sensor shows a SYN/FIN Scan and then a lot of triggered rules.

#### 1.3.3.3 Probability the source address was spoofed

The attacks are coming from different addresses from all over the world. This could have several reasons:

a) The attackers exchange the information than they try from different location, we see several sources.

b) The attacker(s) uses distributed machines to launch the attack.

c) The attacker(s) spoof several addresses to hide their true address (decoy).

I think, some of the addresses are spoofed and it was a distributed attack, because they all appeared at the almost same time. But also information was exchanged, because days later other IP addresses still were trying.

#### 1.3.3.4 Description of the attack

Following messages are displayed by snort. There were no SYN/ACKs in the traces. I show only one though there were many of the same and different addresses.

First one address makes some SYN/FIN scans, then some connection trials. Later on other addresses are trying.

The attacks are coming from 31 different subnets. They are not coming all from the same subnet, this is just because of my obfuscation.

109 times with 79 different source addresses

```
[**] SCAN -SYN FIN [**]
01/09-02:18:29.904232 0:E0:F7:25:5C:D8 - 0:A0:36:0:8B:A1 type:0x800 len:0x3C
10.10.11.111 - 192.168.100.10:111 TCP TTL:26 TOS:0x0 ID:39426 ipLen:20 DgmLen:40
******SF Seq: 0x6C0D80CB Ack: 0x7BE718D7 Win: 0x404 TcpLen: 20

32 times with 13 different source addresses

[**] info = FTP into internal address space [**]
01/09-02:13:11.477293 0:E0:F7:25:5C:D8 - 0:A0:36:0:8B:A1 type:0x800 len:0x3E
******S* Seq: 0x913B661 Ack: 0x0 Win: 0x2238 TcpLen: 28
TCP Options = MSS: 1460 NOP NOP SackOK

11 times, 7 different source IP addresses

[**] MISC -WinGate -1080-Attempt [**]
01/10-07:39:17.122960 0:E0:F7:25:5C:D8 - 0:A0:36:0:8B:A1 type:0x800 len:0x3E
******S* Seq: 0x17BE98 Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options = MSS: 1460 NOP NOP SackOK
```
Two times, the same IP addresses as source

[**] Netbus/Gaba nBus [**]
01/07-08:52:26.944030 0:E0:F7:25:5C:D8 - 0:A0:36:0:8B:A1 type:0x800 len:0x3E
*****S* Seq: 0x71B9C0CE Ack: 0x0 Win: 0x4000 TcpLen: 28
TCP Options = MSS: 1360 NOP NOP SackOK

5 times from 3 source IP addresses

[**] Possible SubSeven access [**]
01/06-21:25:58.760032 0:E0:F7:25:5C:D8 - 0:A0:36:0:8B:A1 type:0x800 len:0x3E
10.10.15.15:4041 – 192.168.100.10:1243 TCP TTL:114 TOS:0x0 ID:54567 IpLen:20 DgmLen:44 DF
*****S* Seq: 0x4C654D Ack: 0x0 Win: 0x2000 TcpLen: 24
TCP Options = MSS: 1460

21 times from 3 source IP addresses

[**] info = HTTP PORT 80 into internal address space [**]
01/05-01:12:43.217648 0:E0:F7:25:5C:D8 - 0:A0:36:0:8B:A1 type:0x800 len:0x3C
*****S* Seq: 0x36ED205 Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options = MSS: 1436 NOP NOP SackOK

5 times from 2 source IP addresses

[**] info = HTTP PORT 8080 into internal address space [**]
01/04-10:02:957043 0:E0:F7:25:5C:D8 - 0:A0:36:0:8B:A1 type:0x800 len:0x3E
*****S* Seq: 0x4035210 Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options = MSS: 1436 NOP NOP SackOK

5 times from 3 source IP addresses

[**] IDS162 – PING Nmap2.36BETA [**]
01/03-09:11:14.321507 0:E0:F7:25:5C:D8 - 0:A0:36:0:8B:A1 type:0x800 len:0x3C
10.10.18.18:192.168.100.10 ICMP TTL:29 TOS:0x0 ID:668 IpLen:20 DgmLen:28 ID:10374 Seq:54275 ECHO

6 times, 3 addresses to different broadcast addresses .0

[**] ICMP Destination Unreachable [**]
01/03-04:41:48.465729 0:D0:FF:14:E0:38 - 8:0:2B.C3:6:32 type:0x800 len:0x3E
192.168.100.10 - 10.10.19.19 ICMP TTL:243 TO S:0x0 ID:38741 IpLen:20 DgmLen:28
** ORIGINAL DATAGRAM DUMP: 10.10.10.1:0.0 - 10.255.255.10:0 ICMP TTL:244 TO S:0x0 ID:38741 IpLen:20 DgmLen:40 **

END OF DUMP

Once and one address.
1.3.3.5 Attack mechanism

Generally: We see scan on a machine and a lot of snort rules triggered on different events.

This case: This seems to be a very weak machine which was probably hacked. I see many first part of tcp handshake but I could not see any responses. When I asked the responsible person he explained, this address belongs to an (old) ATM-Edge-Switch. When I do a NMAP port scan, all UDP ports are stated as filtered, no TCP ports are open. I suppose the switch is just dropping all the packets to its address and this is interpreted as ignored what normally means open port following the RFC 793. This seems to provoke a lot of trials.

1.3.3.6 Correlation

This is a switch. Somebody has been trying to hack actively this machine. It did not result in an actual compromise. There is no sign, that the network behind the switch is attacked. No correlation found.

1.3.3.7 Evidence of active targeting

This is only one machine as a target and many addresses were trying. Active targeting.

1.3.3.8 Severity

Severity = (Critical + Lethal) – (System + Net countermeasures)

Critical: System can only be administrated from a separate port, 1

Lethal: Unlikely successful, 1

System countermeasures: There are systems with older OS, no patches, 3

Net countermeasures: permissive firewall, 2

Severity = (1 + 1) – (5 + 2) = -5
1.3.3.9 Defensive recommendation

♦ Although this switch should forward packets from the Internet the machine itself must not be connectable. The router should deny packets from Internet directed to this machine but handle packets to this device destined to the network behind this switch.

1.3.3.10 Multiple choice test

In some of the previous alerts we see the SYN flag set. If the snort rules triggers only on the TCP SYN flag, you can be sure

A) That this is the first part of a TCP three way handshake.

B) To have just an incomplete TCP connection because you don’t see the SYN/ACK.

C) That this is a communication from a specific program assigned to this port

D) To see an established TCP connection

Answer correct is A).

Not correct because:

B) the SYN/ACK could be routed another way or you missed this packet because of heavy load.

C) ports could be used by other programs to tunnel or just by random assignment for higher ports.

D) You don’t know whether the TCP connection is completely established.
1.3.4 Detect 4: IP spoofing to access a trusted machine

1.3.4.1 The source of the trace
An internal local network lab with non restricted access for all worker.

1.3.4.2 Detection trigger
Detected by TCPDUMP data sniffed with following parameters:
- `e`: print link level header
- `n`: do not convert addresses
- `N`: do not print domain names
- `s`: 1514: Snap length
- `S`: show absolute tcp numbers (see everything)
- `vv`: verbose output
- `x`: show hex dump (I removed the hex dump manually for the most packets)

1.3.4.3 Probability the source address was spoofed
The probability is 100% that the attack used a spoofed IP address.

1.3.4.4 Description of the attack
First we see a normal ARP packet, the .8 is the friendly machine who is allowed to login to .6. The MAC addresses are bold. I cut out some packets to make it not too long.

<table>
<thead>
<tr>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Command</th>
<th>Details</th>
</tr>
</thead>
</table>

Secondly we see a successful login:

<table>
<thead>
<tr>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Command</th>
<th>Details</th>
</tr>
</thead>
</table>
Later on, in the evening, we see a normal ARP request from machine .7.

```
```

Next, some time later we see an ARP response to the .6 machine coming from the MAC of the machine with IP address .7 to- tending to be the machine with IP address .8. Have we missed the request?

```
```

Finally it is tested, whether the .6 machine has update d the cache and a login is performed again. The machine .6 will now talk only to the faked friend, the network card of the real .8 will not listen. To prevent that the real .8 is overrides the data in the ARP cache, ARP responses to the .6 containing the faked ARP response are repeated:

```
```

```
```

...... Some Packets


1.3.4.5 Attack mechanism

Generally: This is an ARP cache poisoning attack. Crafted ARP packets are sent to a host to map the attacker's MAC address to the IP address of the spoofed system. ARP implementations are generally stateless. The host just accepts this information. To
prevent the host from updating its cache with the real MAC address of the spoofed system, ARP transmissions are repeated frequently. All the hosts have to be on the same LAN.

In this case: The attacker uses a probably sniffed trust relationship. We see normal connects from this network as the attacker might have seen. These are older machines and some of these computers do have a trusted relationship. There are other machines on the same network not belonging to this group. The attacker uses the spoofed IP address to get a (successful) connect.

The attacker waits until there is no traffic on the network so the chance to have success is higher (less traffic from the real host). He then sends crafted packets with an ARP reply (gratuitous ARP reply). The target machine sees the packets and updates its cache. Along with IP spoofing, this ARP redirection allows the connection via rlogin to the server without disturbing original (spoofed) machine. The packets sent by the server are actually sent to the malicious machine. The spoofed machine doesn't see them, as its network card rejects the packets: they have the wrong MAC address. To be sure to keep the IP-MAC relationship in the cache these responses are sent from time to time.

1.3.4.6 Correlation


Article: ([http://www.insecure.org/sploits/arp.games.html](http://www.insecure.org/sploits/arp.games.html))

1.3.4.7 Evidence of active targeting

This attack was generated from a specific host to a specific machine. 100% of active targeting.

1.3.4.8 Severity

Severity = (Critical + Lethal) – (System + Net countermeasures)

Critical: Unix desktop, 2

Lethal: User access, 3

System countermeasures: This is an older OS, no patches, no password, 1

Net countermeasures: internal, no firewall, 1

Severity = (2 + 3) – ( 1 + 1 ) = 3

1.3.4.9 Defensive recommendation

- Do not allow or use rlogin (tcp port 512 – 514), use SSH (port 22) instead.
- Do not use trust relationships without strong authentication.
- Better physical security, use of intelligent switches instead of hubs. Makes it more difficult to sniff.
- Restrict access to the machines and periphery (e.g. the network cables)

1.3.4.10 Multiple choice test

You should build a trust relationship based on

A) The MAC address is unique and cannot be changed therefore you can trust it

B) The IP address must be unique in the network therefore you can trust it
C) If the IP address occurs in a packet with a specific MAC address you can trust it

D) Trust should not build on the IP or MAC address.

The correct answer is D), for building a trust relationship you should use a personal (or high-level) authentication e.g. session password and the lines should be encrypted.

Not correct because:

A) Also MAC addresses can be crafted. But the better argument is, that an attacker might just use the trusted machine to get access, then he has the correct MAC and IP. This trust would work only if all machines work on the same subnet. After the first router, this information is lost.

B) The IP address should better be unique in your network to avoid collisions! still, IP spoofing is possible (sometimes trivial): Not to use for trust.

C) The same arguments as in A and B: you can spoof both.
2 Assignment 2 - “Analyze This”

This is the answer to the security audit request from GIAC Enterprises (the customer), a e-business start-up enterprise. This answer has been provided by Michael Semling (the consultant, me).

The sent security data set consists of 3 parts (Snort, SnortA, OOSnort) which were made up of alerts generated by the Snort IDS during approximately one month. Due to power failures, disc space problems and the like, the set of security logs is incomplete, even missing days of data. For these reasons this assessment could only give an incomplete picture from the enterprise’s security situation.

This document is structured as follows:

A list of detects and the used tools or commands to get these are provided. The detects are split up into the same way as delivered, in SnortA (alert files), SnortS (scans), and OOSnort (data with partially dump). Provided is the snort rule the consultant thinks could be the trigger and the problems with this rule. Then follows a short description of the attack, the network name and known ports are written. Not all the hosts I don’t include the entire information about the hosts, the links to look for more detailed information are at the end of this document. This includes the sources and the registration information. The consultant used 4 colours:

<table>
<thead>
<tr>
<th>Colour</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RED</td>
<td>Definitely a compromise. This requires an immediate action. This could be a compromised host that has to be disconnected as soon as possible. This colour does not change, it disappears if the problem is solved.</td>
</tr>
<tr>
<td>YELLOW</td>
<td>This means there is a suspicious host or network problems. There is something “non-normal” going on. This might be scans, configuration problems, misconfigured hardware. YELLOW points needs further investigation and maybe (long-time) observation.</td>
</tr>
<tr>
<td>GREEN</td>
<td>This as a false positive. E.g. A Trojan detection rule depends only on the port number and we see a connection from port 80 to this port. The content is a normal html and the site is more or less identical. It could help to adapt the rule set to decrease the number of false positives.</td>
</tr>
<tr>
<td>BLUE</td>
<td>These are recommendation points. These are no attacks or compromise but his hint helps to make it more difficult for an attacker or easier for the analyst.</td>
</tr>
</tbody>
</table>

A list of “top talkers” is provided, including graphs and statistical numbers.

A list of recommendations and a summary analysis of the data is at the end of the document.

The IP source address is set to 193.168.x.x and is NOT the real home network address. This address was chosen to simplify the work with various tools. See Assignment 3 – Analysis Process. You can replace 193 with MY and 168 with NET.
### 2.1 Detects

**Alert summary:**

<table>
<thead>
<tr>
<th>Signature from alert</th>
<th># of alerts</th>
<th># of sources</th>
<th># of destinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Happy 99 Virus</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>site exec - Possible wu - ftpd exploit - GIAC000623</td>
<td>6</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>SITE EXEC - Possible wu - ftpd exploit - GIAC000623</td>
<td>7</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Tiny Fragments - Possible Hostile Activity</td>
<td>7</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>External RPC call</td>
<td>13</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Probable NMAP fingerprint attempt</td>
<td>15</td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>connect to 515 from inside</td>
<td>56</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>SUNRPC highport access!</td>
<td>60</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>NMAP TCP ping!</td>
<td>96</td>
<td>21</td>
<td>20</td>
</tr>
<tr>
<td>Queso fingerprint</td>
<td>149</td>
<td>29</td>
<td>58</td>
</tr>
<tr>
<td>SMB Name Wildcard</td>
<td>218</td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td>Null scan!</td>
<td>283</td>
<td>204</td>
<td>196</td>
</tr>
<tr>
<td>SNMP public access</td>
<td>468</td>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td>Back Orifice</td>
<td>1697</td>
<td>40</td>
<td>932</td>
</tr>
<tr>
<td>Broadcast Ping to subnet 70</td>
<td>1813</td>
<td>216</td>
<td>1</td>
</tr>
<tr>
<td>Attempted Sun RPC high port access</td>
<td>2542</td>
<td>20</td>
<td>33</td>
</tr>
<tr>
<td>TCP SMTP Source Port traffic</td>
<td>2893</td>
<td>4</td>
<td>2836</td>
</tr>
<tr>
<td>WinGate 1080 Attempt</td>
<td>4802</td>
<td>570</td>
<td>2655</td>
</tr>
<tr>
<td>Watchlist 000222.NET - NCFC</td>
<td>8166</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>Watchlist 000220 IL - ISDNNET-990517</td>
<td>30998</td>
<td>61</td>
<td>108</td>
</tr>
<tr>
<td>SYN - FIN scan!</td>
<td>56250</td>
<td>30</td>
<td>25751</td>
</tr>
</tbody>
</table>
2.1.1 Happy 99 virus

Alert trigger:

```
alert tcp any 110 -> $HOME_NET any (msg:"MCAFEE ID 10144 - Virus - Possible Incoming Happy99 Virus"; content:"X - Spanska:\Yes");
```

```
alert tcp $HOME_NET any -> any 25 (msg:"MCAFEE ID 10144 - Virus - Possible Outgoing Happy99 Virus"; content:"X - Spanska:\Yes");
```

No rules above would trigger the received alert. The first rule could do it, if the rule was changed e.g. “ any any -> $HOME_NET 25". I think in a normal E-mail a content string like :"X - Spanska:\Yes" does not occur. So I would say this is positive alert.

Both addresses occurred once. If these machines are mail server, this would indicate incoming mail containing the virus. If this are clients using external mail server this should be in your policy and the machines could be contaminated.

Recommendation: If I knew there was no scanner, this would be a RED ALERT. Scan all your machines with the latest virus scanner / signatures. Install a network / e-mail scanner.

For more information see: (http://vil.nai.com/vil/dispVirus.asp?virus_k=10144)

2.1.2 WU-FTP

Alert trigger:

```
alert tcp any any -> $HOME_NET 21 (msg:"site exec – Possible wu-ftp – exploit - GIAC000623"; content:"site exec");
```

```
alert tcp any any -> $HOME_NET 21 (msg:"SITE EXEC – Possible wu-ftp – exploit - GIAC000623"; content:"SITE EXEC");
```

This rule triggers everything that goes to a ftp server and contains an executable command. This rule could produce a lot of false positives.

4 sources, 7 destinations

<table>
<thead>
<tr>
<th>Source</th>
<th>Port</th>
<th>Destination</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>208.61.44.215</td>
<td>3746</td>
<td>BellSouth.net Inc</td>
<td></td>
</tr>
<tr>
<td>10/16-16:55:26.342617</td>
<td>62275-62281</td>
<td>ServiceCo LLC - Road Runner</td>
<td></td>
</tr>
</tbody>
</table>
There was an attempt to execute a command. See (http://whitehats.com/IDS/317).

Are these machines allowed to maintain a ftp server? If not, terminate the service immediately and close this port for all machines but the official ftp server by the firewall.

Recommendations: If these are ftp server, be sure they have the latest patches to close the known security holes. Consider restricting the use of specific features. This could be to not to accept site-exec commands. Limit the access through a TCP wrapper. The destination machines above should be analysed.

I would like to have the log files and the binary code from these transmissions. This should be further investigated.

### 2.1.3 Tiny Fragments

Alert trigger: preprocessor minfrag. I don’t know what size you gave. I think 128 is a good value.

5 sources, 6 destinations

<table>
<thead>
<tr>
<th>Source</th>
<th>Port</th>
<th>Source</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.157.126.93</td>
<td>9939</td>
<td>Toronto Star Newspapers</td>
<td></td>
</tr>
<tr>
<td>202.156.51.76</td>
<td>143</td>
<td>SINGAPORE CABLEVISION LTD</td>
<td></td>
</tr>
<tr>
<td>192.206.151.152</td>
<td>21</td>
<td>America Online, Inc</td>
<td></td>
</tr>
<tr>
<td>109.209.56.64</td>
<td>983</td>
<td>DISHNETDSL LTD</td>
<td></td>
</tr>
<tr>
<td>10/04-11:56:14.285666 [**]</td>
<td>Possible wu-ftpd exploit</td>
<td>GIA000623 [**]</td>
<td></td>
</tr>
</tbody>
</table>
I don’t know the value in your pre-processor settings, which was used to trigger the tiny fragments. But fragment technique was used to bypass IDS or firewalls and several (DoS) attacks based on fragments. I would like to see the size of the packets and the content. This could of course also be a trial for teardrop of ping of death. Hopefully these machines are well patched.

There is also a preprocessor called “defrag”, which is doing full IP defragmentation and reassembly. The data could then be logged and analysed.

### 2.1.4 External RPC call

Alert trigger:

```bash
alert tcp ! $HOME_NET any -> $HOME_NET 111 (msg: "External RPC call "; flags: S)
```

This rule relies only on the port number. This produces normally a lot of false positives. Still, because of the high number of known exploits, I think this rule is important.

8 sources, 3 destinations

<table>
<thead>
<tr>
<th>Source</th>
<th>Port</th>
<th>Source</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>63.162.239.69</td>
<td>975</td>
<td>BELZ INVESTCO</td>
<td>!low port</td>
</tr>
<tr>
<td>63.162.239.69</td>
<td>3655</td>
<td>BELZ INVESTCO</td>
<td>!low port</td>
</tr>
<tr>
<td>63.162.239.69</td>
<td>4496</td>
<td>BELZ INVESTCO</td>
<td>!low port</td>
</tr>
<tr>
<td>200.191.80.206</td>
<td>931</td>
<td>AcessoNet Ltda</td>
<td>!low port</td>
</tr>
<tr>
<td>200.191.80.181</td>
<td>634</td>
<td>AcessoNet Ltda</td>
<td>!low port</td>
</tr>
<tr>
<td>200.191.80.181</td>
<td>665</td>
<td>AcessoNet Ltda</td>
<td>!low port</td>
</tr>
<tr>
<td>38.200.223.8</td>
<td>2473</td>
<td>PSINETA</td>
<td>Aker-cdp</td>
</tr>
<tr>
<td>38.200.223.8</td>
<td>2473</td>
<td>PSINETA</td>
<td>Aker-cdp</td>
</tr>
<tr>
<td>38.200.223.8</td>
<td>2473</td>
<td>PSINETA</td>
<td>Aker-cdp</td>
</tr>
<tr>
<td>211.46.110.61</td>
<td>708</td>
<td>Korea Internet</td>
<td></td>
</tr>
<tr>
<td>211.46.110.61</td>
<td>4910</td>
<td>Korea Internet</td>
<td></td>
</tr>
<tr>
<td>24.23.151.112</td>
<td>3306</td>
<td>@Home Network</td>
<td>My SQL</td>
</tr>
<tr>
<td>24.23.151.112</td>
<td>3306</td>
<td>@Home Network</td>
<td>My SQL</td>
</tr>
<tr>
<td>24.7.227.215</td>
<td>5</td>
<td>@Home Network</td>
<td>Remote Job Entry</td>
</tr>
<tr>
<td>10/14-19:41:47.442871 [<em>] External RPC call [</em>] (1149 alerts from this range!!)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source</td>
<td>Port</td>
<td>Source</td>
<td>Port</td>
</tr>
<tr>
<td>12.34.21.196</td>
<td>700</td>
<td>CFS EUROPE LTD</td>
<td></td>
</tr>
</tbody>
</table>
A query was sent to the portmapper port (111). This could be an exploit (http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0078). There are many vulnerable services (tooltalk, cmsd). Are these destinations SUN/Linux/Windows machines? Are they well patched?

Personally: I think this is very dangerous.

Recommendation: These low ports must not enter generally into your enterprise, block it (port 5 from external?) and open only specific ports for specific server (e.g. port 80 for your webserver). I don't think, RPCs (e.g. NFS, Yellow Pages, NIS ...) should be globally accessible. Generally: Block incoming 111. Restrict the range, $HOME_NET only. Analyse these machines.

### 2.1.5 Probable NMAP fingerprint attempt

Alert trigger:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Probable NMAP fingerprint attempt";flags:SFU;)
```

This combination of flags is not normal. Either a misconfiguration or more probable a scan occurred. This signature is trustworthy.

14 sources, 13 destinations

<table>
<thead>
<tr>
<th>Source</th>
<th>Port</th>
<th>Source</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>24.95.192.51</td>
<td>1</td>
<td>ServiceCo LLC – RoadRunner</td>
<td>TCP Port Service Multiplexer</td>
</tr>
<tr>
<td>195.132.57.32</td>
<td>1</td>
<td>CYBERCABLE FR</td>
<td>Port</td>
</tr>
<tr>
<td>205.251.201.36</td>
<td>1490</td>
<td>Cable Atlantic Inc</td>
<td>insitu-conf</td>
</tr>
<tr>
<td>132.178.218.161</td>
<td>3448</td>
<td>Boise State University</td>
<td></td>
</tr>
<tr>
<td>24.6.151.155</td>
<td>62110</td>
<td>Deutsche Telekom AG</td>
<td>ODD Packet - RAT</td>
</tr>
<tr>
<td>24.6.180.134.156</td>
<td>57446</td>
<td>6699</td>
<td>0</td>
</tr>
<tr>
<td>24.9.64.57</td>
<td>1095</td>
<td>193.168.15.15 (9x)</td>
<td></td>
</tr>
<tr>
<td>193.168.15.127 (3x)</td>
<td></td>
<td>193.168.100.130 (1x)</td>
<td></td>
</tr>
</tbody>
</table>
NMAP tool was used to find out the network topology and probably the OS of the machines with OS fingerprinting. Also some low ports are not blocked. This is a pre-attack analysis. Not dangerous itself but with the gained knowledge an attacker can prepare the attack and win a possible time race.

Do not give away more information about your system more than necessary. Reject requests for unused addresses and machines which do not need to be connected from the outside.
2.1.6  NMAP TCP ping

A scan too. I will not list all sources and destinations only the ones I find interesting. Some statistical information can be found below in the TOP scan list.

Alert trigger:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"NMAP TCP PING";flags:A;ack:0;)
```

ACK flag set and acknumber = 0 can occur if the ack number count goes over the range of 2^32 and resets to 0. But more likely is a scan.

96 sources, 21 destinations

<table>
<thead>
<tr>
<th>Source</th>
<th>Port</th>
<th>Source</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.102.197.234</td>
<td>53</td>
<td>Intel Corporation</td>
<td>DNS</td>
</tr>
<tr>
<td>10/04-06:46:22</td>
<td>80</td>
<td>JARING-UNITAR2</td>
<td>HTTP</td>
</tr>
<tr>
<td>63.119.91.2</td>
<td>80</td>
<td>JUNET Technologies, Inc</td>
<td>HTTP</td>
</tr>
<tr>
<td>11/13-08:25:17</td>
<td>80</td>
<td>Teligent, Inc</td>
<td>HTTP</td>
</tr>
<tr>
<td>10/25-04:29:51</td>
<td>80</td>
<td>Southwire Company</td>
<td>HTTP</td>
</tr>
<tr>
<td>2.2.2.2</td>
<td>80</td>
<td>Reserved Range</td>
<td>HTTP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destination</th>
<th>Port</th>
<th>Destination</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.168.1.3</td>
<td>53</td>
<td>MY.NET</td>
<td>DNS</td>
</tr>
<tr>
<td>193.168.1.5</td>
<td>53</td>
<td>MY.NET</td>
<td>SMTP</td>
</tr>
<tr>
<td>193.168.1.7</td>
<td>25</td>
<td>MY.NET</td>
<td>SMTP</td>
</tr>
<tr>
<td>193.168.1.8</td>
<td>25</td>
<td>MY.NET</td>
<td>SMTP</td>
</tr>
<tr>
<td>193.168.1.9</td>
<td>25</td>
<td>MY.NET</td>
<td>SMTP</td>
</tr>
<tr>
<td>193.168.1.10</td>
<td>25</td>
<td>MY.NET</td>
<td>SMTP</td>
</tr>
<tr>
<td>193.168.253.43</td>
<td>25</td>
<td>MY.NET</td>
<td>SMTP</td>
</tr>
<tr>
<td>193.168.6.47</td>
<td>25</td>
<td>MY.NET</td>
<td>SMTP</td>
</tr>
<tr>
<td>193.168.6.35</td>
<td>25</td>
<td>MY.NET</td>
<td>SMTP</td>
</tr>
<tr>
<td>193.168.6.47</td>
<td>80</td>
<td>MY.NET</td>
<td>HTTP</td>
</tr>
<tr>
<td>193.168.253.42</td>
<td>80</td>
<td>MY.NET</td>
<td>HTTP</td>
</tr>
<tr>
<td>193.168.253.125</td>
<td>80</td>
<td>MY.NET</td>
<td>HTTP</td>
</tr>
<tr>
<td>193.168.6.14</td>
<td>80</td>
<td>MY.NET</td>
<td>HTTP</td>
</tr>
</tbody>
</table>
NMAP TCP ping was used to find out whether the host is alive and has an open port.

There were many scans from DNS and HTTP port which passed the firewall going to a DNS or HTTP or SMTP server. These are definitely scans because outgoing connections are normally from internal address highport to external address lowport and for incoming connections to a server on the internal network the internal address lowport and the external address highport. With DNS a connection could be established from source port 53 to destination port 53. But this should be for DNS exclusively. There is even a scan coming from NET – RESERVED –2 (reserved address space 2.0.0.0 – 2.255.255.255).

Recommendation: Block incoming low ports going to a server with low port, e.g. http, dns, smtp. Some attention is required with DNS and FTP. Be sure to have the latest patches on the DNS.

2.1.7 QUESO fingerprint attempt

There are 142 alerts with this rule. I just show you some examples.

Alert trigger:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Queso Fingerprint attempt";flags:S12;)
```

The reserved bits 1,2 are normally not used.

29 sources, 58 destinations, not all listed.

<table>
<thead>
<tr>
<th>Source</th>
<th>Port</th>
<th>Source</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>24.3.161.193</td>
<td>range</td>
<td>@Home Network</td>
<td></td>
</tr>
<tr>
<td>195.115.7.2</td>
<td>range</td>
<td>CEGETEL</td>
<td></td>
</tr>
<tr>
<td>129.242.219.27</td>
<td>range</td>
<td>University of Tromso</td>
<td></td>
</tr>
<tr>
<td>64.80.53.121</td>
<td>range</td>
<td>PaeTec Communications, Inc.</td>
<td></td>
</tr>
<tr>
<td>24.163.42.82</td>
<td>3347</td>
<td>ServiceCo LLC - Road Runner</td>
<td>Phoenix RPC</td>
</tr>
<tr>
<td>128.233.247.116</td>
<td>26</td>
<td>Cornell University</td>
<td>Unassigned</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destination</th>
<th>Port</th>
<th>Destination</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>MY.NET</td>
<td>POP3</td>
<td>HTTPS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>newacct</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>wkars</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>interserver</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GameGen1</td>
<td></td>
</tr>
</tbody>
</table>
This is a scan again. Some of the packets could be false positives because there are some VPN (Virtual Private Networks) using the reserved bits. If you are not using VPNs, this is definitively a scan going on over a long time. Of course it could be that not the same people use the machine to scan your network. But I think they are thoroughly inspecting your network. The discovered information could be used to plan or refine future attacks. These machines above should be analysed. This is the same as NMAP fingerprinting. Don’t give away too much information.

Recommendation: Enhance and improve your defence perimeter immediately!

2.1.8 null scan

Alert trigger:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"IDS04 - SCAN-NUL Scan",flags:0; seq:0; ack:0;)
```

No flags are normal. This signature can be trusted.

204 sources, 196 destinations. Only two machines with other alerts are listed.

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Source Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.253.247.116</td>
<td></td>
</tr>
<tr>
<td>24.180.134.156</td>
<td>KORNET</td>
</tr>
</tbody>
</table>

5 instances of Queso fingerprint  
8 instances of Null scan!

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Source Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.253.247.116</td>
<td></td>
</tr>
<tr>
<td>24.180.134.156</td>
<td>KORNET</td>
</tr>
</tbody>
</table>

1 instances of Null scan!  
1 instances of Probable NMAP fingerprint attempt  
1 instances of NMAP TCP ping!

These are not top scanners. Only 3 addresses (24.112.150.20, 24.113.148.32, 28.253.247.116) do have 8 times null scan alert.

2.1.9 SYN FIN scan

Alert trigger:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"IDS198 - SCAN - SYN FIN",flags:SF;)
```

SYN and FIN flags together are not normal. This signature can be trusted.

30 sources, 25751 destinations. Only the sources are shown:

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Source Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>160.78.49.191</td>
<td>PARMANET</td>
</tr>
<tr>
<td>209.92.40.32</td>
<td>FASTNET(tm)</td>
</tr>
<tr>
<td>130.89.229.48</td>
<td>University Twente</td>
</tr>
<tr>
<td>203.32.161.197</td>
<td>INTERNETPLUS1 -AU</td>
</tr>
<tr>
<td>193.64.114.10</td>
<td>PRINTEQ-TI-1</td>
</tr>
<tr>
<td>210.101.101.110</td>
<td>KORNET</td>
</tr>
<tr>
<td>143.89.13.3</td>
<td>Hong Kong University of Science &amp; Technlogy</td>
</tr>
<tr>
<td>63.167.58.13</td>
<td>Sprint</td>
</tr>
<tr>
<td>212.187.21.156</td>
<td>Telekabel Apeldoorn</td>
</tr>
<tr>
<td>139.130.61.206</td>
<td>Telstra Corporation Limited</td>
</tr>
<tr>
<td>24.7.222.215</td>
<td>@Home Network</td>
</tr>
<tr>
<td>129.101.18.16</td>
<td>University of Idaho</td>
</tr>
<tr>
<td>212.177.241.101</td>
<td>IT-UUNET-990512</td>
</tr>
<tr>
<td>24.200.140.155</td>
<td>Videoiron Lire</td>
</tr>
<tr>
<td>24.65.121.98</td>
<td>Shaw Fiberlink Ltd.</td>
</tr>
<tr>
<td>208.61.4.207</td>
<td>BellSouth.net Inc</td>
</tr>
<tr>
<td>63.195.56.20</td>
<td>Pacific Bell Internet Services,Inc</td>
</tr>
<tr>
<td>210.113.89.200</td>
<td>KORNET</td>
</tr>
<tr>
<td>128.2.81.133</td>
<td>Carnegie-Mellon University</td>
</tr>
<tr>
<td>195.103.69.159</td>
<td>TOPSOFT -NET</td>
</tr>
<tr>
<td>212.0.107.107</td>
<td>TELSON</td>
</tr>
<tr>
<td>213.41.69.52</td>
<td>FR-COLT-FRANCE-BESS-SHAREDHOSTING5</td>
</tr>
<tr>
<td>163.10.19.34</td>
<td>Universidad Nacional de La Plata</td>
</tr>
<tr>
<td>211.46.110.81</td>
<td>Korea Internet</td>
</tr>
<tr>
<td>202.153.112.222</td>
<td>POWERHOUSE</td>
</tr>
<tr>
<td>62.96.171.103</td>
<td>DE-COLT-NMG</td>
</tr>
<tr>
<td>129.130.98.92</td>
<td>Kansas State University</td>
</tr>
<tr>
<td>24.112.150.20</td>
<td>Rogers@Home Ontario (2 times)</td>
</tr>
<tr>
<td>129.93.206.170</td>
<td>University of Nebraska-Lincoln</td>
</tr>
</tbody>
</table>

© SANS Institute 2000 - 2002  
As part of GIAC practical repository.  
Author retains full rights.
There is a significant amount of scans. All scans came from outside. The top scanner did only scan.

Not only scanners were:

```
24.7.227.215  @Home Network  1 instances of External RPC call
24.112.150.20  Rogers@Home  2 instances of SYN -FIN scan!
24.200.140.155  Videotron Ltee  2 instances of SYN -FIN scan!
```

These attackers did scan the network but there is not substantial additional effort recognisable.

### 2.1.10 connect to 515 from inside

**Alert trigger:**

```
alert tcp $HOME_NET any -> any 515 (msg:“connect to 515 from inside ; flags:PA;“)
```

This rule only works with the port number. Therefore false positives may be frequent.

1 sources, 2 destinations

<table>
<thead>
<tr>
<th>Source</th>
<th>Port</th>
<th>Source</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.168.179.78</td>
<td>2274</td>
<td>MY.NET</td>
<td></td>
</tr>
<tr>
<td>193.168.179.78</td>
<td>2707</td>
<td>MY.NET</td>
<td>EMCSYMAPIPORT</td>
</tr>
</tbody>
</table>

Is 193.168.100.3 a printer spooler?

Port 515 is the line printer spooler port. Possible is remote file creation, deletion, and execution. ([http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0061](http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0061))

This could also be a lpd overflow. ([http://www.whitehats.com/info/IDS456](http://www.whitehats.com/info/IDS456), [http://www.whitehats.com/info/IDS457](http://www.whitehats.com/info/IDS457))

If these machines do act as printerserver (two of them outside and requested from inside??) this could be normal although it seems that they are hardly used. Could either be a successful hack so there is no more trial necessary or it was a misconfiguration.

Recommend: Look in the destinations' log files (probably not possible for the machines outside of your network). Hopefully there is something like tripwire to find differences. Block all unused ports on local machines.

This needs more investigation but I think this has to be done very soon. This is the reason why I give a red alert here.

### 2.1.11 SUN RPC HIGHPOR T access

**Alert trigger:**

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 32771 (msg:“Sun RPC high port access”;flags:A;)
```

This alert just triggers on the port. False positives, e.g. if a client uses this port as a source port, may be frequent.

13 sources, 12 destinations
<table>
<thead>
<tr>
<th>Source</th>
<th>Port</th>
<th>Source</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>216.10.12.30</td>
<td>2078</td>
<td>Virtual Development Inc</td>
<td>Always this port</td>
</tr>
<tr>
<td>216.148.218.200</td>
<td>443</td>
<td>TCG CERT/Net</td>
<td>HTTPS</td>
</tr>
<tr>
<td>193.168.206.180</td>
<td>3191</td>
<td>PTTNET-DIALUP3</td>
<td>Doing also 1080 scan?</td>
</tr>
<tr>
<td>216.10.12.2</td>
<td>4600</td>
<td>Virtual Development Inc</td>
<td>Piranha1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destination</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.168.206.222</td>
<td>32771</td>
</tr>
<tr>
<td>193.168.202.242</td>
<td></td>
</tr>
<tr>
<td>193.168.212.186</td>
<td></td>
</tr>
<tr>
<td>193.168.220.62</td>
<td></td>
</tr>
<tr>
<td>193.168.97.59</td>
<td></td>
</tr>
<tr>
<td>193.168.253.141</td>
<td></td>
</tr>
<tr>
<td>193.168.206.218</td>
<td></td>
</tr>
<tr>
<td>193.168.253.14</td>
<td></td>
</tr>
<tr>
<td>193.168.253.15</td>
<td></td>
</tr>
<tr>
<td>193.168.140.51</td>
<td></td>
</tr>
<tr>
<td>193.168.179.78</td>
<td></td>
</tr>
</tbody>
</table>

It is likely that most of them are false positives. If there are some requests from internally chosen port 32771.

Especially interesting because they don’t do only RPC highport access are address 211.46.110.81 with 2068 alerts, 195.34.28.117 with 9 alerts. 216.10.12.30 does alert 33 times, but only, highport access.

216.10.12.30 works with the machine 193.168.202.242 and later with 193.168.206.222. For 216.148.218.180 looks like a false positive with https port communicating also to 193.168.206.222. But because of data in the AlertOO file together with SCAN and Alerts this address is in the probably compromised list. But there are only two packets at a time, and it looks like the time is repeating more or less every 13 hours.
2.1.12  Attempt SUN RPC HIGHPORT access

Alert trigger:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 32771 (msg: "Attempt Sun RPC high port access"; flags:S;)
```

This alert just triggers on the port but also on the SYN. False positives, e.g. if a server responds on this port, may be frequent. Portscan triggers also this rule.

20 sources, 33 destinations

<table>
<thead>
<tr>
<th>Source</th>
<th>Port</th>
<th>Source</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>205.188.153.xxx</td>
<td>4000</td>
<td>America Online, Inc (2534 times)</td>
<td>53</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destination</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.168.1.8</td>
<td>32771</td>
</tr>
<tr>
<td>193.168.144.42</td>
<td></td>
</tr>
<tr>
<td>193.168.202.242</td>
<td></td>
</tr>
<tr>
<td>193.168.204.134</td>
<td></td>
</tr>
<tr>
<td>193.168.205.130</td>
<td></td>
</tr>
<tr>
<td>193.168.206.222</td>
<td></td>
</tr>
<tr>
<td>193.168.209.182</td>
<td></td>
</tr>
<tr>
<td>193.168.217.218</td>
<td></td>
</tr>
<tr>
<td>193.168.219.130</td>
<td></td>
</tr>
<tr>
<td>193.168.220.194</td>
<td></td>
</tr>
<tr>
<td>193.168.220.78</td>
<td></td>
</tr>
<tr>
<td>193.168.221.126</td>
<td></td>
</tr>
<tr>
<td>193.168.222.98</td>
<td></td>
</tr>
<tr>
<td>193.168.223.18</td>
<td></td>
</tr>
<tr>
<td>193.168.234.214</td>
<td></td>
</tr>
<tr>
<td>193.168.225.106</td>
<td></td>
</tr>
<tr>
<td>193.168.225.210</td>
<td></td>
</tr>
<tr>
<td>193.168.225.98</td>
<td></td>
</tr>
<tr>
<td>193.168.226.198</td>
<td></td>
</tr>
<tr>
<td>193.168.226.74</td>
<td></td>
</tr>
<tr>
<td>193.168.227.170</td>
<td></td>
</tr>
<tr>
<td>193.168.227.50</td>
<td></td>
</tr>
<tr>
<td>193.168.228.22</td>
<td></td>
</tr>
<tr>
<td>193.168.97.152</td>
<td></td>
</tr>
<tr>
<td>193.168.97.163</td>
<td></td>
</tr>
<tr>
<td>193.168.97.202</td>
<td></td>
</tr>
<tr>
<td>193.168.97.62</td>
<td></td>
</tr>
<tr>
<td>193.168.97.182</td>
<td></td>
</tr>
<tr>
<td>193.168.217.218</td>
<td></td>
</tr>
<tr>
<td>193.168.219.130</td>
<td></td>
</tr>
<tr>
<td>193.168.220.194</td>
<td></td>
</tr>
<tr>
<td>193.168.220.78</td>
<td></td>
</tr>
<tr>
<td>193.168.221.126</td>
<td></td>
</tr>
<tr>
<td>193.168.222.98</td>
<td></td>
</tr>
<tr>
<td>193.168.223.18</td>
<td></td>
</tr>
<tr>
<td>193.168.234.214</td>
<td></td>
</tr>
<tr>
<td>193.168.225.106</td>
<td></td>
</tr>
<tr>
<td>193.168.225.210</td>
<td></td>
</tr>
<tr>
<td>193.168.225.98</td>
<td></td>
</tr>
<tr>
<td>193.168.226.198</td>
<td></td>
</tr>
<tr>
<td>193.168.226.74</td>
<td></td>
</tr>
<tr>
<td>193.168.227.170</td>
<td></td>
</tr>
<tr>
<td>193.168.227.50</td>
<td></td>
</tr>
<tr>
<td>193.168.228.22</td>
<td></td>
</tr>
<tr>
<td>193.168.97.152</td>
<td></td>
</tr>
<tr>
<td>193.168.97.163</td>
<td></td>
</tr>
<tr>
<td>193.168.97.202</td>
<td></td>
</tr>
<tr>
<td>193.168.97.62</td>
<td></td>
</tr>
</tbody>
</table>

This seems to be a scan from one network 205.188.153.xxx with port 4000 or sometimes port 53.

Machines which already were in the RPC list e.g. 193.168.206.222 and 193.168.202.242.

Difficult to say whether they were just scanning. But the machine 193.168.206.222 appears in RPC and RPC attempt and could be compromised. 12 addresses are going to this IP address. The address 210.101.101.110 does a SYNFIN scan and then in the OOD datafile we see splitted SF with payload 00. This could be a part of the ttobser buffer overflow exploit though I never saw it in a SYNFIN packet (http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0003+)
2.1.13 SNMP public access

Alert trigger:

```
alert udp any any -> $HOME_NET 161 (msg:" SNMP public access"; content:"public");
```

This rule with the port and content analysis is rather reliable.

<table>
<thead>
<tr>
<th>Source</th>
<th>Port</th>
<th>Source</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.168.101.*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>193.168.97.*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>193.168.98.*</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destination</th>
<th>Port</th>
<th>Destination</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.168.101.192</td>
<td>161</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

There is one destination which is accessed from three subnets using the community string ‘public’ which is the default string. No one from outside connected this machine. Leaving a machine with default password makes it too easy for attackers. With the public community name a read-only access to the MIB (Management Information Base) is granted leading to a leakage problem.

Moreover, by inspecting other alerts where the destination machine appears, we can see that there are 23 sources accessing the destination 468 times with SNMP and 93 times with smb name wildcard. Is this machine a host doing also smb, and routing work? Or are these NetBIOS requests to get information. Is this a Windows machine maybe acting as printserver? This needs to be investigated!

This would be a candidate for blue colour if there was not too many sources. The smb rises a need for more information.

Recommendation: Block at the border (router, firewall) all SNMP queries entering and all SNMP traffic from leaving your network. Allow one management station to do SNMP settings.

2.1.14 SMB NAME WILDCARD

Alert trigger:

```
alert udp any any -> $HOME_NET 137 (msg:"SMB Name Wildcard"; content:"CKAAAAAAAAAAAAAAAAAAAAAAAAAAAAA|0000|";)
```

This rule with its content is rather reliable. I would, if allowed, accept SMB internally and set the source to !$HOME_NET.

<table>
<thead>
<tr>
<th>Internal Source</th>
<th>External Source</th>
<th>Name</th>
<th>Destinations</th>
</tr>
</thead>
</table>

33 sources, 33 destinations, not all listed
This is a standard netbios name table query. Windows machines use these queries in the file sharing protocol to determine NetBIOS names when they know only the IP addresses. An attacker uses the same query to extract information e.g. workstation name, domain, and currently logged in users. There are some vulnerabilities (http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=cve-1999-0225, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=cve-1999-0391, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=cain - 1999-0495). If we could see the content of the packets, we could look for ..

169.254.184.161 looks like scanning the machines for these entries. Bell Atlantic uses a server which seems to support smb. Recommendation: Do not use SMB to transfer files in the Internet. Block SMB queries, outgoing and incoming at the border.

2.1.15 Broadcast Ping to subnet 70

Alert trigger (might be):

```
alert icmp any any -> 193.168.70.255 any (msg: "Broadcast Ping to subnet"; itype: 8;)
```

This one is ping (icmp type 8) to broadcast to subnet 70. Reliable.

1813 sources, 1 destination, just the most "pinger"

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.231.169.166</td>
<td>213.154.131.131</td>
</tr>
<tr>
<td>193.226.60.179</td>
<td>193.231.220.71</td>
</tr>
<tr>
<td>193.226.60.179</td>
<td>193.231.220.71</td>
</tr>
<tr>
<td>193.231.169.166</td>
<td>213.154.131.131</td>
</tr>
<tr>
<td>193.231.220.71</td>
<td>193.231.220.71</td>
</tr>
<tr>
<td>193.231.220.71</td>
<td>193.231.220.71</td>
</tr>
<tr>
<td>193.231.220.101</td>
<td>217.10.206.79</td>
</tr>
</tbody>
</table>
193.168.70.255 was pinged. This could be a network topology recognition.
This could also be a DoS attack. All the machines on the other side will respond with a ICMP ECHO REPL. This can flood the router or firewall.
The SMURF (PAPASMURF, http://netscan.org/papasmurf.c, fraggle) attack uses a spoofed IP address which will in turn receive all the replies and become a DoS target.
For me this looks definitely like a smurf, some machines (e.g. 193.231.169.166) do this request 10 times during two minutes.
These addresses are spoofed with a high probability!
This makes your network looking like the source of the attacks to the target!
Recommendation: Block broadcast addresses (x.x.x.0 and x.x.x.255 and if you have smaller sub -networks these broadcasts too) immediately at the border (router, firewall), Routers have usually an option to deny forwarding of directed broadcast. E.g. "access-list xxx deny ip any host <internal address.0>" and "access-list xxx deny ip any host <internal address.255>"

2.1.16 Back Orifice
Alert trigger:
```plaintext
alert udp any any -> $HOME_NET 31337 (msg:"Back Orifice");
```
This rule only triggers on the port. There are a lot of false positives, especially if a server – client connections use this port (e.g. 80 -> 31337).
40 sources, 932 destinations. I just show some addresses with more than one connection for source or destination.

<table>
<thead>
<tr>
<th>Source Port</th>
<th>Source Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>62.136.90.120 1101</td>
<td>3327</td>
</tr>
<tr>
<td>62.136.90.120 3539</td>
<td>3599</td>
</tr>
<tr>
<td>62.136.90.120 1260</td>
<td>BBARS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source Port</th>
<th>Source Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>213.43.69.72 31338</td>
<td></td>
</tr>
<tr>
<td>213.43.69.72 291 times range</td>
<td></td>
</tr>
<tr>
<td>203.148.182.108 79 times</td>
<td></td>
</tr>
<tr>
<td>203.148.182.108 75 times 31338</td>
<td></td>
</tr>
<tr>
<td>209.94.199.186 78 times 31338</td>
<td></td>
</tr>
</tbody>
</table>
I just copy and pasted the ten most used MY.NET addresses:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Number</th>
<th>Source</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.168.98.150</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>193.168.97.208</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>193.168.98.82</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>193.168.98.119</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>193.168.98.151</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>193.168.98.77</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>193.168.97.142</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>193.168.98.81</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>193.168.98.195</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>193.168.98.118</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Normally, if I see a low port (e.g. 80) and the related client highport I look into the datastream and can tell whether it is HTTP or not.

What worries is, there are highport – highport connections. Most used Ports used are 31338, 21117, 14321, 4377, the address 62.136.90.120 likes to use 3327, 3539, 3599, 1280, 1101 are also frequent.

63.46.46.143 is doing a scan over the network MY.NET.219.* - MY.NET.231.* using various source ports.

203.148.182.108 is doing a scan over the range MY.NET.98.* using port 21117, 1041.

213.43.69.72 scans with port 31338 the MY.NET.97.*.

I would look at the machines above and the following list. These are machines which were scanned and accessed later directly.


There are just scans. But there are either direct trials or successful connections to machines inside. This is BackOrifice1 for windows95/98. If these machines use this OS an investigation is badly necessary!

Recommendation: Install a network virus scanner and a scanner individually on every machine. Do a network audit as soon as possible.

### 2.1.17 TCP SMTP source port traffic

Alert trigger:

```
alert tcp !$HOME_NET any -> $HOME_NET 25 (msg:"TCP SMTP Source Port traffic";flags:PA;)
```

This rule alerts for all not internal email addresses going on port 25 to the mail server.

4 sources, 2836 destinations

<table>
<thead>
<tr>
<th>Source</th>
<th>Port</th>
<th>Source</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>194.86.77.240</td>
<td>25</td>
<td>LONDON1-DIAL-POOL2</td>
<td>SMTP</td>
</tr>
<tr>
<td>194.67.168.11</td>
<td>25</td>
<td>RELSOFT</td>
<td></td>
</tr>
<tr>
<td>194.67.168.11</td>
<td>25</td>
<td>RELSOFT</td>
<td></td>
</tr>
</tbody>
</table>

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.
144.67.168.11 from Russia seems to know which addresses to access. This address with port 25 connects directly to the addresses in the destination field with ports 1001, 1004, 1013, 1014, 1018, 1019. Is one of these servers a mail server? Port 1001 is known for Trojans like Silencer, WebEx, Der Spaeher3, Insane Network.

Source and destination port is 25. Normally for mail transfer from internal client to external server, the internal machine uses a highport (>1023) and the external server is requested to port 25. Internal server on port 25 accepts incoming mail from any port. Maybe with source and destination port 25 the scanner knows that the inspected server accepts incoming and outgoing messages. Then this server could act simply as a mail relay.

These guys might be looking for an unprotected mail server. A machine could be used as a spam relay (http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0682). There exists also several exploits for sendmail or super mail transfer package for windows NT. (http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0404, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0203)

Under circumstances these machines can also be used to do DoS (see chameleon overflow (http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0261, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0582)

These machines should be investigated. The other activities look like scans. This means your are definitely a target. This could be yellow, but I think this is urgent to do, so I say red. Your mail server(s) could be used to send spam mail to others making your servers look like the sender.

Recommendation: Protect your mail server. Install the latest patches. Remove sendmail daemon if the machine is not a mail server. To send mails from a machine sendmail daemon is not required.

2.1.18 Wingate 1080 Attempt

Alert trigger:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 1080 (msg:"MISC -WinGate-1080-Attempt",flags:S) 
```

570 sources, 2655 destinations

These internal hosts do have the most traffic:

<table>
<thead>
<tr>
<th>Internal host</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.168.206.118</td>
<td>372</td>
</tr>
<tr>
<td>193.168.225.154</td>
<td>126</td>
</tr>
<tr>
<td>193.168.60.11</td>
<td>75</td>
</tr>
<tr>
<td>193.168.60.8</td>
<td>67</td>
</tr>
<tr>
<td>193.168.60.16</td>
<td>40</td>
</tr>
<tr>
<td>193.168.60.38</td>
<td>34</td>
</tr>
<tr>
<td>193.168.203.78</td>
<td>34</td>
</tr>
<tr>
<td>193.168.53.91</td>
<td>29</td>
</tr>
<tr>
<td>193.168.222.102</td>
<td>25</td>
</tr>
<tr>
<td>193.168.53.219</td>
<td>24</td>
</tr>
</tbody>
</table>

This could indicate that there are servers installed. Connection from an internal to an internal proxy was not found, if the rule above was used only external sources are triggered. There is a huge interest in these machines and there are too many exploits not to worry. This needs fast investigation.

Especially remarkable is the source IP address 127.0.0.1 going to 193.168.181.144. Is this machine also your sniffer? This machine should be worth special investigation and fast because it could be compromised.

Recommendation: If these proxies are legal, look for the latest patches. Maybe only internal traffic leaving the net and no external traffic is allowed to use this proxy. Set the rules!

### 2.2 Watchlists

<table>
<thead>
<tr>
<th>Signature from alertA</th>
<th># of alerts</th>
<th># of sources</th>
<th># of destinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Watchlist 000222 NET -NCFC</td>
<td>8166</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>Watchlist 000220 IL -ISDNNET -990517</td>
<td>30998</td>
<td>61</td>
<td>108</td>
</tr>
</tbody>
</table>

The Watchlist 000222 NET -NCFC registates suspicious traffic from/to "The Computer Network Center Chinese Academy of Sciences". There are many scans for telnet (23), or mail server (25) in this list.

The Watchlist 000220 IL -ISDNNET -990517 does the same for the "Cable-Modem-Experiment" network. There were a lot of alerts coming from this. Again some with port 0 or low ports.
2.3 TOP Talker

This section is divided into top talker incoming and top talker from MB.NET. This is repeated with the machines with the most exploits. A top talker could be a scanner. Then there is a link graph with the busiest machines and their partner.

<table>
<thead>
<tr>
<th>External Top Talker</th>
<th>Count</th>
<th>Internal Top Talker</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>160.78.49.191</td>
<td>7199</td>
<td>193.168.6.7</td>
<td>5808</td>
</tr>
<tr>
<td>208.61.4.207</td>
<td>6635</td>
<td>193.168.211.146</td>
<td>4814</td>
</tr>
<tr>
<td>159.226.45.3</td>
<td>6296</td>
<td>193.168.223.98</td>
<td>3938</td>
</tr>
<tr>
<td>212.179.96.5</td>
<td>6117</td>
<td>193.168.206.90</td>
<td>3918</td>
</tr>
<tr>
<td>209.92.40.32</td>
<td>4967</td>
<td>193.168.203.142</td>
<td>1640</td>
</tr>
<tr>
<td>212.179.27.6</td>
<td>4011</td>
<td>193.168.218.142</td>
<td>1463</td>
</tr>
<tr>
<td>212.179.79.2</td>
<td>3950</td>
<td>193.168.214.170</td>
<td>1371</td>
</tr>
<tr>
<td>212.179.44.115</td>
<td>3936</td>
<td>193.168.100.230</td>
<td>1302</td>
</tr>
<tr>
<td>63.195.56.20</td>
<td>3897</td>
<td>193.168.202.22</td>
<td>952</td>
</tr>
<tr>
<td>130.89.229.48</td>
<td>3860</td>
<td>193.168.201.174</td>
<td>803</td>
</tr>
</tbody>
</table>

Top Talker from machines which do have most traffic to each other:
### Most used ports:

<table>
<thead>
<tr>
<th>External IP addresses</th>
<th>Internal IP addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Count</td>
</tr>
<tr>
<td>21</td>
<td>19619</td>
</tr>
<tr>
<td>53</td>
<td>18307</td>
</tr>
<tr>
<td>9704</td>
<td>14184</td>
</tr>
<tr>
<td>27374</td>
<td>3572</td>
</tr>
<tr>
<td>25</td>
<td>2894</td>
</tr>
<tr>
<td>1067</td>
<td>2699</td>
</tr>
<tr>
<td>4000</td>
<td>2536</td>
</tr>
<tr>
<td>6699</td>
<td>1905</td>
</tr>
<tr>
<td>1498</td>
<td>1823</td>
</tr>
<tr>
<td>1574</td>
<td>1727</td>
</tr>
</tbody>
</table>

### Top Alerters:

<table>
<thead>
<tr>
<th>External</th>
<th>Internal</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRC IP</td>
<td>Count</td>
</tr>
<tr>
<td>160.75.49.191</td>
<td>7199</td>
</tr>
<tr>
<td>206.61.4.207</td>
<td>6635</td>
</tr>
<tr>
<td>159.226.45.3</td>
<td>6296</td>
</tr>
<tr>
<td>212.179.95.5</td>
<td>6117</td>
</tr>
<tr>
<td>209.92.40.32</td>
<td>4967</td>
</tr>
<tr>
<td>212.179.27.6</td>
<td>4011</td>
</tr>
<tr>
<td>212.179.79.2</td>
<td>3950</td>
</tr>
<tr>
<td>212.179.44.115</td>
<td>3936</td>
</tr>
<tr>
<td>63.195.56.20</td>
<td>3897</td>
</tr>
<tr>
<td>130.89.229.48</td>
<td>3860</td>
</tr>
</tbody>
</table>

### Most used Exploits:

<table>
<thead>
<tr>
<th>Internal – Internal Alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
</tr>
<tr>
<td>193.168.101.96</td>
</tr>
<tr>
<td>193.168.98.106</td>
</tr>
<tr>
<td>193.168.101.142</td>
</tr>
<tr>
<td>193.168.98.174</td>
</tr>
<tr>
<td>193.168.97.185</td>
</tr>
<tr>
<td>193.168.97.171</td>
</tr>
<tr>
<td>193.168.97.204</td>
</tr>
<tr>
<td>193.168.98.122</td>
</tr>
<tr>
<td>193.168.98.197</td>
</tr>
<tr>
<td>193.168.97.178</td>
</tr>
</tbody>
</table>
### Internal – External Alerts

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Count</th>
<th>Alert</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.168.179.78</td>
<td>64.244.202.66</td>
<td>1</td>
<td>[<em>] connect to 515 from inside [</em>]</td>
</tr>
<tr>
<td>193.168.179.78</td>
<td>64.244.202.110</td>
<td>1</td>
<td>[<em>] connect to 515 from inside [</em>]</td>
</tr>
</tbody>
</table>

### External – Internal Alerts

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Count</th>
<th>Alert</th>
</tr>
</thead>
<tbody>
<tr>
<td>159.226.45.3</td>
<td>193.168.6.7</td>
<td>5758</td>
<td>[<em>] Watchlist 000222 NET-NCFC [</em>]</td>
</tr>
<tr>
<td>212.179.95.5</td>
<td>193.168.211.146</td>
<td>4810</td>
<td>[<em>] Watchlist 000220 IL-SDNNET-990517 [</em>]</td>
</tr>
<tr>
<td>212.179.44.115</td>
<td>193.168.223.98</td>
<td>3936</td>
<td>[<em>] Watchlist 000220 IL-SDNNET-990517 [</em>]</td>
</tr>
<tr>
<td>212.179.27.6</td>
<td>193.168.206.90</td>
<td>3120</td>
<td>[<em>] Watchlist 000220 IL-SDNNET-990517 [</em>]</td>
</tr>
<tr>
<td>212.179.79.2</td>
<td>193.168.203.142</td>
<td>1638</td>
<td>[<em>] Watchlist 000220 IL-SDNNET-990517 [</em>]</td>
</tr>
<tr>
<td>212.179.79.2</td>
<td>193.168.218.142</td>
<td>1459</td>
<td>[<em>] Watchlist 000220 IL-SDNNET-990517 [</em>]</td>
</tr>
<tr>
<td>212.179.41.24</td>
<td>193.168.214.170</td>
<td>1353</td>
<td>[<em>] Watchlist 000220 IL-SDNNET-990517 [</em>]</td>
</tr>
<tr>
<td>159.226.91.20</td>
<td>193.168.100.230</td>
<td>1131</td>
<td>[<em>] Watchlist 000222 NET-NCFC [</em>]</td>
</tr>
<tr>
<td>205.188.153.108</td>
<td>193.168.221.246</td>
<td>488</td>
<td>[<em>] Attempted Sun RPC high port access [</em>]</td>
</tr>
</tbody>
</table>

**General Alert Chart:**

- [*] SYN-FIN scan! [*]
- [*] Watchlist 000220 IL-SDNNET-990517 [*]
- [*] Watchlist 000222 NET-NCFC [*]
- [*] WinGate 1080 Attempt [*]
- [*] TCP SMTP Source Port traffic [*]
- [*] Attempted Sun RPC high port access [*]
- [*] Back Orifice [*]
- [*] SNMP public access [*]
- [*] Null scan! [*]
- [*] SMB Name Wildcard [*]
- [*] Queso Fingerprint [*]
- [*] NMAP TCP ping! [*]
- [*] SUNRPC highport access [*]
- [*] connect to 515 from inside [*]
- [*] Probable NMAP Fingerprint attempt [*]
- [*] External RPC call [*]
- [*] site exec - Possible wu-ftpd exploit - GIAC000623 [*]
- [*] Tiny Fragments - Possible Hostile Activity [*]
- [*] Happy 99 Virus [*]
2.4 TOP Scans

In the TOP Scans the machines acting either as source or as destination with most of scan are listed. There are a lot of SYN scans. But also SYN/FIN scans and UDP scans. You are scanned intensively! This could be a SYN attack or just machines which establish a lot of connections depending of the settings of your rule to trigger SYN scans.

<table>
<thead>
<tr>
<th>Scan Type</th>
<th>Counts</th>
<th>Scan Type</th>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYN <strong>S</strong>***</td>
<td>235386</td>
<td>FULLXMAS 21SFPAU RESERVE DBITS</td>
<td>28</td>
</tr>
<tr>
<td>SYNFIN <strong>SF</strong>**</td>
<td>51628</td>
<td>SYNFIN 21SF**** RESERVE DBITS</td>
<td>16</td>
</tr>
<tr>
<td>UDP</td>
<td>23954</td>
<td>FIN <em>1F</em>*** RESERVE DBITS</td>
<td>16</td>
</tr>
<tr>
<td>FIN <strong><strong>E</strong></strong></td>
<td>454</td>
<td>NOACK 2<em>SF</em>*** RESERVE DBITS</td>
<td>14</td>
</tr>
<tr>
<td>VECNA **<strong>R</strong>*A</td>
<td>351</td>
<td>UNKNOWN 2<em>S</em>**A'RESERVE DBITS</td>
<td>14</td>
</tr>
<tr>
<td>INVALIDACK <strong>S<em>R</em></strong></td>
<td>281</td>
<td>NOACK <strong>S<em>R</em></strong></td>
<td>14</td>
</tr>
<tr>
<td>NULL <strong><strong>E</strong></strong></td>
<td>16</td>
<td>NOACK*1SFPA** RESERVE DBITS</td>
<td>14</td>
</tr>
<tr>
<td>SYNF 2<em>S</em>*** RESERVEDBITS</td>
<td>104</td>
<td>NOACK*1<strong>RP</strong> RESERVE DBITS</td>
<td>14</td>
</tr>
<tr>
<td>INVALIDACK <em><strong>FR</strong></em></td>
<td>60</td>
<td>NOACK<strong>S</strong>P**</td>
<td>13</td>
</tr>
<tr>
<td>NOACK*1SFPA** RESERVEDBITS</td>
<td>29</td>
<td>INVALIDACK*1SFPA** RESERVE DBITS</td>
<td>13</td>
</tr>
</tbody>
</table>

Addresses with the most scans:

<table>
<thead>
<tr>
<th>External Source addresses doing the most scans</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>66.9.27.254</td>
<td>20649</td>
</tr>
<tr>
<td>62.252.21.241</td>
<td>13057</td>
</tr>
<tr>
<td>194.244.78.145</td>
<td>11904</td>
</tr>
<tr>
<td>63.88.175.201</td>
<td>11718</td>
</tr>
<tr>
<td>62.157.23.237</td>
<td>9641</td>
</tr>
</tbody>
</table>
### Internal Source addresses doing the most scans

<table>
<thead>
<tr>
<th>Source Address</th>
<th>Count</th>
<th>Types of Scans</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.168.224.150</td>
<td>2981</td>
<td>SYN <strong>S</strong>*** , UDP</td>
</tr>
<tr>
<td>193.168.221.82</td>
<td>2668</td>
<td>SYN <strong>S</strong>***</td>
</tr>
<tr>
<td>193.168.5.25</td>
<td>2311</td>
<td>UDP</td>
</tr>
<tr>
<td>193.168.1.3</td>
<td>577</td>
<td>UDP</td>
</tr>
<tr>
<td>193.168.110.111</td>
<td>270</td>
<td>UDP</td>
</tr>
<tr>
<td>193.168.110.16</td>
<td>267</td>
<td>UDP</td>
</tr>
<tr>
<td>193.168.109.41</td>
<td>262</td>
<td>UDP</td>
</tr>
<tr>
<td>193.168.110.105</td>
<td>215</td>
<td>UDP</td>
</tr>
<tr>
<td>193.168.110.108</td>
<td>160</td>
<td>UDP</td>
</tr>
<tr>
<td>193.168.110.109</td>
<td>120</td>
<td>UDP</td>
</tr>
<tr>
<td>193.168.109.40</td>
<td>109</td>
<td>UDP</td>
</tr>
<tr>
<td>193.168.110.110</td>
<td>100</td>
<td>UDP</td>
</tr>
<tr>
<td>193.168.213.58</td>
<td>94</td>
<td>SYN <strong>S</strong>***</td>
</tr>
<tr>
<td>193.168.109.38</td>
<td>93</td>
<td>UDP</td>
</tr>
<tr>
<td>193.168.1.4</td>
<td>22</td>
<td>UDP</td>
</tr>
<tr>
<td>193.168.152.165</td>
<td>14</td>
<td>UDP</td>
</tr>
<tr>
<td>193.168.101.1</td>
<td>117</td>
<td>INVALIDACK 2**FR<em>A</em> RESERVEDBITS</td>
</tr>
<tr>
<td>193.168.19.10</td>
<td>117</td>
<td>INVALIDACK 2**FR<em>A</em> RESERVEDBITS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source Address</th>
<th>Count</th>
<th>Types of Scans</th>
</tr>
</thead>
<tbody>
<tr>
<td>63.248.55.245</td>
<td>9073</td>
<td></td>
</tr>
<tr>
<td>62.96.169.86</td>
<td>8939</td>
<td></td>
</tr>
<tr>
<td>24.23.151.112</td>
<td>8763</td>
<td></td>
</tr>
<tr>
<td>64.50.161.162</td>
<td>8635</td>
<td></td>
</tr>
<tr>
<td>160.78.49.191</td>
<td>7192</td>
<td></td>
</tr>
<tr>
<td>128.211.237.11</td>
<td>7003</td>
<td></td>
</tr>
<tr>
<td>211.49.165.9</td>
<td>6919</td>
<td></td>
</tr>
<tr>
<td>208.61.4.207</td>
<td>6634</td>
<td></td>
</tr>
<tr>
<td>62.155.244.68</td>
<td>6401</td>
<td></td>
</tr>
<tr>
<td>216.191.162.145</td>
<td>6093</td>
<td></td>
</tr>
<tr>
<td>63.198.207.51</td>
<td>6033</td>
<td></td>
</tr>
<tr>
<td>209.92.40.32</td>
<td>4956</td>
<td></td>
</tr>
<tr>
<td>208.214.247.60</td>
<td>4881</td>
<td></td>
</tr>
<tr>
<td>146.101.14.7.251</td>
<td>4855</td>
<td></td>
</tr>
<tr>
<td>199.239.94.98</td>
<td>4836</td>
<td></td>
</tr>
</tbody>
</table>

© SANS Institute 2000 - 2002
As part of GIAC practical repository. Author retains full rights.
<table>
<thead>
<tr>
<th>Most scanned targets</th>
<th>Count</th>
<th>Most scanned targets</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.168.220.2</td>
<td>11926</td>
<td>193.168.70.121</td>
<td>1206</td>
</tr>
<tr>
<td>193.168.218.50</td>
<td>4710</td>
<td>193.168.204.26</td>
<td>1169</td>
</tr>
<tr>
<td>193.168.206.94</td>
<td>1799</td>
<td>193.168.212.114</td>
<td>1083</td>
</tr>
<tr>
<td>193.168.162.77</td>
<td>1759</td>
<td>193.168.205.246</td>
<td>1076</td>
</tr>
<tr>
<td>193.168.120.36</td>
<td>1591</td>
<td>193.168.97.59</td>
<td>996</td>
</tr>
<tr>
<td>193.168.205.214</td>
<td>1589</td>
<td>193.168.98.168</td>
<td>973</td>
</tr>
<tr>
<td>193.168.215.210</td>
<td>1367</td>
<td>193.168.98.171</td>
<td>900</td>
</tr>
<tr>
<td>193.168.60.16</td>
<td>1306</td>
<td>193.168.162.36</td>
<td>867</td>
</tr>
<tr>
<td>193.168.140.57</td>
<td>1220</td>
<td>193.168.211.254</td>
<td>719</td>
</tr>
</tbody>
</table>
### 2.5 Possibly compromised machines

The following list is a collection of alerts and scans including the red alert ones:

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Scan Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.168.1.2</td>
<td><strong>Watchlist 000222 NET-NCFC [</strong>] <strong>TCP SMTP Source Port traffic [</strong>] <strong>SYN-FIN scan [</strong>]</td>
</tr>
<tr>
<td>193.168.6.15</td>
<td><strong>SMB Name Wildcard [</strong>] <strong>External RPC call [</strong>] <strong>SUNRPC highport access! [</strong>]</td>
</tr>
<tr>
<td>193.168.6.35</td>
<td><strong>Watchlist 000222 NET-NCFC [</strong>] <strong>Dueso fingerprint [</strong>] <strong>NMAP TCP ping! [</strong>] <strong>Happy 99 Virus [</strong>]</td>
</tr>
<tr>
<td>193.168.19.10</td>
<td>Scanning external network</td>
</tr>
<tr>
<td>193.168.100.3</td>
<td><strong>connect to 515 from inside [</strong>] <strong>SYN-FIN scan [</strong>]</td>
</tr>
<tr>
<td>193.168.100.130</td>
<td><strong>SMB Name Wildcard [</strong>] <strong>External RPC call [</strong>] <strong>SYN-FIN scan [</strong>] <strong>External RPC call [</strong>]</td>
</tr>
<tr>
<td>193.168.101.1</td>
<td>Scanning external network</td>
</tr>
<tr>
<td>193.168.101.192</td>
<td><strong>SNMP public access [</strong>] <strong>SMB Name Wildcard [</strong>]</td>
</tr>
<tr>
<td>193.168.203.198</td>
<td>Connection with machines 205.188.3.*</td>
</tr>
<tr>
<td>193.168.206.222</td>
<td><strong>Attempted Sun RPC high port access [</strong>] <strong>SUNRPC highport access! [</strong>] <strong>SYN-FIN scan [</strong>] Possible Exploits w. tooltalk</td>
</tr>
<tr>
<td>193.168.218.50</td>
<td><strong>SYN-FIN scan [</strong>] <strong>WinGate 1080 Attempt [</strong>] <strong>Watchlist 000220 IL-ISDN-990517 [</strong>] <strong>Back Orifice [</strong>]</td>
</tr>
<tr>
<td>193.168.218.106</td>
<td>Connection with 207.172.3.46, several low ports (e.g. 0,1,10,...), making OS detection</td>
</tr>
<tr>
<td>193.168.225.98</td>
<td><strong>Attempted Sun RPC high port access [</strong>] <strong>SYN-FIN scan [</strong>] <strong>Back Orifice [</strong>]</td>
</tr>
</tbody>
</table>

### 2.6 Recommendations


I recommend several changes to increase the security and provide a better IDS.

- I did not receive the enterprise policy. There must be an up-to-date policy.
- All the machines should be audited.
- The customer should implement a 24x7 days intrusion detection with worst case space available and UPS (uninterruptible power supply). Do install several sensors if you have a distributed network and collect the data on a central analysing machine. Log the content of packets causing an alert at least as long as your examination time lasts.
- There should be a perimeter defence, a stateful firewall or better a proxy filter. If there are load problems put the filter between two packet filter firewalls. The packet filter firewalls will keep away a lot of traffic and relieve the proxy or stateful firewall.
- Network defences. E.g. Virusscanner on the mail server, router settings should make it hard to attack.
- This snort is not up to date. (The TCP flags are still in wrong order!) Install the latest snort and ruleset. Include also the plugins. Manage your ruleset for your network in a different set so you can include it easily.
- Install host security software. E.g. Virusscanner, Desktop firewalls.
- Do network management from a well defined hosts.

Our team specialist would be pleased to help to improve the security of your network.

2.7 Summary

There was a lot of data to analyse. Make smaller pieces and send them regularly to the analyst. This amount of data almost leads to Denial of Analyst and increases the response time!

Normally I analyse networks which do have at least a minimum of protection. The data shows that your network is weak. There are several red alerts where reaction is required immediately. Also I suppose your network was abused to attack other machines (e.g. smurf attack). Some of your machines have been used to OS scan other machines too.

There was no green alert (false positive) because I could not verify the false positive status. Surely many of the attacks are false positives. So most of the status are yellow, requiring further investigation.

To improve the security you should filter the unsolicited traffic at the router by blocking incoming access to non-public servers. A proxy firewall could help to keep away a lot of traffic. All your machines should be audited regularly (e.g. with Nessus).
3 Assignment 3 - Analysis Process

The data I downloaded consisted of 3 parts:

2. SnortS - 42 files, 21 748 kB. The file with the scans.

After receiving the data I decided to analyse them in three blocks as they arrived – I wanted to use tools to process the large logs. First I combined the files:

```
for files in SnortA*.txt ;  do  cat $files >> alertA1.txt ; done
for files in SnortS*.txt;  do  cat $files >> alertS1.txt ; done
for files in OOSche*.txt;  do  cat $files >> alertO1.txt ; done
```

For the alert file:

Each file has some information in its header, I just want to have the lines with an alert with 
```
\`
```
in:
```
grep '^[a-zA-Z]*\[**\][a-zA-Z]*\]' alertA1.txt > alertA2.txt
```

To see if I removed too much I looked at the lines I threw away:

```
grep -v '^[a-zA-Z]*\[**\][a-zA-Z]*\]' alertA1.txt > dust.txt
```

I wanted the `n` to change the "MY.NET" into a number because the available tools would work then. I looked for an unused address "192.168." with the command:

```
grep -i '192.168.' alertA2.txt
```

This address was found so I just tried with a next one "193.168.", which was not found. This address was not the original network! Next was replacing all the "MY.NET." by "193.168." Again: **This is NOT the real home network address**! This address belongs to EU-960208 or LU-RESTENA-193-168-64-127 or M.ETRONET.

```
cat alertA2.txt | sed 's/MY.NET./193.168./g' > alertA.txt
```

For the scan files I could not just look for the alert.

For the purpose to remove lines with just a few letter (e.g. '\n') I just wrote a `quick and dirty` perl script. No error handling.

```
#!/usr/bin/perl
open(DATA,"<$ARGV[0]") || die "Can't read from $ARGV[0]"

($dev, $ino, $mode, $nlink, $uid, $gid, $rdev, $size, $atime, $mtime, $ctime, $blksize, $blocks) = stat($ARGV[0]);

# read all bytes of the file into $s
read(DATA,$s,$size);

# we split the whole file into lines separated by '\n'
@line = split('/\n/',$s);

foreach $elem (@line)
{
    # if the length of the line is smaller then 10 Bytes discard it
```
I then run tools like snortsnarf, snortstat on this data set. If you request I send you a CD with the data on.

The alertA file was no problem, 10 minutes of processing time. The alertS file needed a little bit more memory. Therefore I added swap to my Linux machine. I used the commands:

1. `dd if=/dev/zero of=swapfile bs=1024 count= 524288`
2. `mkswap swapfile 524288`
3. `sync`
4. `swapon swapfile`
5. at the end I deactivated it with “swapoff swapfile” and deleted the file.

The result of snortsnarf on alertA is the table and all the links at the beginning to assignment 2.

To do statistics I decided (because of the large amount of data) to use ACCESS instead of EXCEL. So I changed the format into semicolon delimited format. I changed the ‘:’ into spaces with the command:

`cat alertA.txt | tr : ' ' > alertDBA1.txt`

Because of the format I excluded the broadcast ping and the portscan alerts, these do not have the same format as all the other alerts:

`cat AlertDBA1.txt | grep -v 'spp_portscan' | grep -v 'Broadcast Ping to subnet 70' > alertDBA2.txt`

The I used a perl program to separate. I kept the IP addresses as one single field, I can do a (slow) search with text and wildcards.

`./changeA2DB.pl alertDBA2.txt > alertDBA.txt`

I imported then the alertDBA.txt into ACCESS using its import functions. No Errors occurred.

Screenshots of Access with the queries (Back Orifice alert) in the upper window and a result in the window below.
And the same thing again for alertS.txt:

To avoid errors with multiple spaces I run the following line before the perl line:

```
cat alertS.txt \| awk '{print $1, print $2, print $3, print $4, print $5, print $6, print $7, print $8, print $9}' > alertDBS1.txt
```

```
#!/usr/bin/perl
$_ = `cat $ARGV[0]`;
LOOP : while(<>)  {
    /(.*?) (.*?) (.*?) (.*?)
      : (.*?)-
      : (.*?) (.*?
        : n/sgc;
print $1, " ; ", $2, " ; ", $3, " ; ", $4, " ; ", $5, " ; ", $6, " ; ", $7, "; ", $8,
};
```

And the import worked without any error messages. With the query wizards, SQL queries are easily made.
### 3.1 References:

<table>
<thead>
<tr>
<th>Tool</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNORT</td>
<td><a href="http://www.snort.org">http://www.snort.org</a></td>
</tr>
<tr>
<td>ACID</td>
<td><a href="http://www.cert.org/kb/acid/">http://www.cert.org/kb/acid/</a></td>
</tr>
<tr>
<td>Correlation for the Analysis</td>
<td><a href="http://www.sans.org/y2k/practical/Guy_Bruneau.doc">http://www.sans.org/y2k/practical/Guy_Bruneau.doc</a></td>
</tr>
<tr>
<td>RIPE</td>
<td><a href="http://www.ripe.net">http://www.ripe.net</a></td>
</tr>
<tr>
<td>ARIN</td>
<td><a href="http://www.arin.net">http://www.arin.net</a></td>
</tr>
<tr>
<td>SANS</td>
<td><a href="http://www.sans.org">http://www.sans.org</a></td>
</tr>
<tr>
<td>APNIC</td>
<td><a href="http://www.apnic.net">http://www.apnic.net</a></td>
</tr>
</tbody>
</table>
## Upcoming Training

<table>
<thead>
<tr>
<th>Event</th>
<th>Location</th>
<th>Dates</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mentor Session - SEC503</td>
<td>Houston, TX</td>
<td>Mar 21, 2020 - Apr 25, 2020</td>
<td>Mentor</td>
</tr>
<tr>
<td>SANS 2020</td>
<td>Orlando, FL</td>
<td>Apr 03, 2020 - Apr 10, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS London April 2020</td>
<td>London, United Kingdom</td>
<td>Apr 20, 2020 - Apr 25, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Baltimore Spring 2020</td>
<td>Baltimore, MD</td>
<td>Apr 27, 2020 - May 02, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Security West 2020</td>
<td>San Diego, CA</td>
<td>May 06, 2020 - May 13, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Amsterdam May 2020</td>
<td>Amsterdam, Netherlands</td>
<td>May 11, 2020 - May 18, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS San Antonio 2020</td>
<td>San Antonio, TX</td>
<td>May 17, 2020 - May 22, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Las Vegas Summer 2020</td>
<td>Las Vegas, NV</td>
<td>Jun 08, 2020 - Jun 13, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Munich July 2020</td>
<td>Munich, Germany</td>
<td>Jul 06, 2020 - Jul 11, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>Rocky Mountain Summer 2020 - SEC503: Intrusion Detection In-Depth</td>
<td>Denver, CO</td>
<td>Jul 20, 2020 - Jul 25, 2020</td>
<td>vLive</td>
</tr>
<tr>
<td>SANS Columbia 2020</td>
<td>Columbia, MD</td>
<td>Jul 20, 2020 - Jul 25, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Rocky Mountain Summer 2020</td>
<td>Denver, CO</td>
<td>Jul 20, 2020 - Jul 25, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS vLive - SEC503: Intrusion Detection In-Depth</td>
<td>SEC503 - 2020008,</td>
<td>Aug 10, 2020 - Sep 16, 2020</td>
<td>vLive</td>
</tr>
<tr>
<td>SANS Amsterdam August 2020</td>
<td>Amsterdam, Netherlands</td>
<td>Aug 17, 2020 - Aug 22, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Melbourne 2020</td>
<td>Melbourne, Australia</td>
<td>Aug 17, 2020 - Aug 22, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Virginia Beach 2020</td>
<td>Virginia Beach, VA</td>
<td>Aug 24, 2020 - Sep 04, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS New York City Summer 2020</td>
<td>New York City, NY</td>
<td>Aug 30, 2020 - Sep 04, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS London September 2020</td>
<td>London, United Kingdom</td>
<td>Sep 07, 2020 - Sep 12, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Munich September 2020</td>
<td>Munich, Germany</td>
<td>Sep 14, 2020 - Sep 19, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Network Security 2020</td>
<td>Las Vegas, NV</td>
<td>Sep 20, 2020 - Sep 27, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Northern VA - Reston Fall 2020</td>
<td>Reston, VA</td>
<td>Sep 28, 2020 - Oct 03, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS OnDemand</td>
<td>Online</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
<tr>
<td>SANS SelfStudy</td>
<td>Books &amp; MP3s Only</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
</tbody>
</table>