Global Information Assurance Certification Paper

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/giactc/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 [**]
01/15-21:53:37.971780 0:E0:F7:25:5C:D8 - 0:C0:4F:A9:BB:FA type:0x800 len:0x4A
192.168.11.11 - 192.168.11.110 ICMP TTL:22 TOS:0x0 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 qrstuvwabcdefghi

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.
IPLen 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:

time (hh:mm:ss: μμμμμ)
Source MAC address – Destination MAC address
Ethernet information :type and length (minimum 60 / 0x3c)
source IP address. source port > destination IP address. destination port
TCP flags
TCP start sequence number : TCP end sequence number
(Size)
ack and number which indicates the next block of data with this sequence nu mber
win Advertised window size
IP Flags (DF)
TTL time to live value. See Snort.
ID IP identifier
Hex dump and ASCII representation.

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 not 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 Intr u-sion Detection and Packet Filt ering: How it works by Vicki Irwin and Hal Pomeranz, page 218/222". It will be increase the pro b-ability 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[0:1] & 0x0f) * 4) -8 -4) = (udp[9:1] * 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 se e 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 – BACKDOO R 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
192.168.200.11:54321  -> 192.168.200.12:1037 TCP TTL:128 TOS:0x0 ID:29698 IpLen:20 DgmLen:44 DF
***A**S* Seq: 0x4A95E6  Ack: 0x4B26B9  W  in: 0x2238  TcpLen: 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 snor t alert with the default port 54321.

```
11:57:30.706901 192.168.200.11.54321 > 192.168.200.12.1034: P 4423681:4423703(22) ack 4460992 win 8513 (DF) (ttl 128, id
59905)
                        4500 003e ea01 4000 8006 ff4e c0a8 c80b
                        c0a8 c80c  D431 040a 0043 8001 0044 11c0
                        5018 2141 684e 0000 1200 0000 ae1d 7a56
                        908e fa9b 990e b99b 990e b99b 990e
```

11:57:30.716278 192.168.200.12.1034 > 192.168.200.11.54321: P 4460992:4461036(44) ack 4423703 win 8738 (DF) (ttl 128, id 55041)

4500 0054 d701 4000 8006 1239 c0a8 c80c
c0a8 c80b ***040a D431 0044 11c0 0043 8017***
5018 2222 e0eb 0000 2800 0000 ae1d 7a56
7201 fd9b b90e b99b 980e b99b 980e b99b
6c01 fd9b 990e b99b 990e b99b 990e b99b
990e b99b

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

14:30:48.475773 192.168.200.12.1040 > 192.168.200.11.21501: P 13657856:13657900(44) ack 13620843 win 8563 (DF) (ttl 128, id 37635)

4500 0 054 9303 4000 8006 5637 c0a8 c80c
c0a8 c80b ***0410 53fd 00d0 6700 00cf d66b***
5018 2173 de15 0000 2800 0000 ae1d 7a56
1f6b 699b b90e b99b 980e b99b 980e b99b
3d6b 699b 990e b99b 990e b99b 990e b99b
990e b99b

14:30:48.482888 192.168.200.11.2 1501 > 192.168.200.12.1040: P 13620843:13620932(89) ack 13657900 win 8570 (DF) (ttl 128, id 3331)

4500 0081 0d03 4000 8006 dc0a c0a8 c80b
c0a8 c80c ***53fd 0410 00cf d66b 00d0 672c***
5018 217a 7d17 0000 5500 0000 ae1d 7a56
20db 769b d40e b99b 9b 0e b99b 66f1 4664
3d6b 699b 980e b99b b40e b99b 990e b99b
b423 87bb cf6b cbe8 f061 d7a1 b94c d8f8
f22e f6e9 f068 d0f8 fc2e 8bab a93e 99b3
db41 8bd0 b02e cfaa b73f b391 990e b99b
99

1.3.1.5 Attack mechanism

Generally: BackOrifice 2000 is a Tro jan horse with a lot of configuration possibilities. Remote control is poss ible 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 r emotely.

1.3.1.6 Correlation

Candidate for the CVE list CAN -1999-0660 (http://www.c ve.mitre.org/cgi -bin/cvename.cgi?name=CAN -1999-0660). See also http://www.bo2k.com/ .

1.3.1.7 Evidence of active targeting

These packets are coming from one source and going to one target. 100% active targeting.

1.3.1.8 Severity

Severity = (Critical + Lethal) – (System + 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 countermeasur es: There seems to be no firewall, 1

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 sa me 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 t rigger

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 reco gnisable.

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 r esults.

1.3.2.4 Description of the at tack

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 (ttl 53)
19:09:51.483426 192.168.140.10.53: > 192.168.141.27.2084: R 167535093:167535093(0) win 0 (DF) (ttl 60)
19:09:51.488515 192.168.141.27.2084 > 192.16 8.140.10.80: F 55135905:55135905(0) ack 167535093 win 512 (ttl 53)
19:09:51.489159 192.168.140.10.80 > 192.168.141.27.2084: R 167535093:167535093(0) win 0 (DF) (ttl 60)

19:10:41.502842 192.168.141.27.1808 > 192.168.140.11.53: F 1454240847:1454240847(0) ac k 1520589986 win 512 (ttl 53)
19:10:41.509029 192.168.140.11.53: > 192.168.141.27.1808: R 1520589986:1520589986(0) win 0 (ttl 56)
19:10:41.514027 192.168.141.27.1808 > 192.168.140.11.80: F 1454240847:1454240847(0) ack 1520589986 win 512 (ttl 53)
19:10:41.5 15075 192.168.140.11.80 > 192.168.141.27.1808: R 1520589986:1520589986(0) win 0 (ttl 56)

19:11:31.532897 192.168.141.27.1045 > 192.168.140.12.53: F 1777954818:1777954818(0) ack 1798268799 win 512 (ttl 53)
19:11:31.548795 192.168.141.27.1045 > 192.168.140. 12.80: F 1777954818:1777954818(0) ack 1798268799 win 512 (ttl 53)

19:12:21.562848 192.168.141.27.1903 > 192.168.140.13.53: F 2113850935:2113850935(0) ack 1014478355 win 512 (ttl 53)
.....
```

1.3.2.5 Attack mechanism

This is a slow FIN -ACK scan, the time for each hos t 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 a crafted packets.

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

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 a nswer.

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 bloc ks 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 exclu ded.

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 at tack (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 : U nix/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 recommen dation

- 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 a 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
10.10.11.11:111 - 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
10.10.12.12:2000 - 192.168.100.10:21 TCP TTL:110 TOS:0x0 ID:28801 IpLen:20 DgmLen:48 DF
******S* Seq: 0x613B661 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
10.10.13.13:1674 - 192.168.100.10:1080 TCP TTL:108 TOS:0x0 ID:9378 IpLen:20 DgmLen:48 DF
******S* Seq: 0xF17BE98 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
10.10.14.14:2155 - 192.168.100.10:12345 TCP TTL:113 TOS:0x0 ID:157 IpLen:20 DgmLen:48 DF
******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
10.10.15.15:3712 – 192.168.100.10:1243 TCP TTL:114 TOS:0x0 ID:54567 IpLen:20 DgmLen: 48 DF
******S* Seq: 0x4C654D Ack: 0x0 Win: 0x2000 TcpLen: 28 TCP Options = MSS: 536 NOP NOP SackOK
```

22 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
10.10.16.16:3783 – 192.168.100.10:80 TCP TTL:126 TOS:0x0 ID:47502 IpLen:20 DgmLen:44 DF
******S* Seq: 0x36ED205 Ack: 0x0 Win: 0x2000 TcpLen: 24
TCP Options = MSS: 1460
```

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
10.10.17.17:2849 – 192.168.100.10:8080 TCP TTL:112 TOS:0x0 ID:34298 IpLen:20 DgmLen:48 DF ******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:0x46
192.168.100.10 - 10.10.19.19 ICMP TTL:243 TO S:0x0 ID:38741 IpLen:20
DgmLen:56 DESTINATION UNREACHABLE: NET UNREACHABLE
** ORIGINAL DATAGRAM DUMP: 10.1.0.10:0 - 10.255.255.10:0 TCP TTL:244 TOS:0x0 ID:38741 IpLen:20 DgmLen:40 **
END OF DUMP
```

Once and one address.

```
[**] info = INCOMING SMTP into in  ternal address space [**]
01/04-22:09:58.609627 0:E0:F7:25:5C:D8  - 0:A0:36:0:8B:A1 type:0x800 len:0x4A
10.10.20.20:1855  - 192.168.100.10:25 TCP TTL:52 TOS:0x0 ID:6671 IpLen:20 DgmLen:60 DF
******S* Seq: 0xB380F29F Ack: 0x0 Win: 0x7D78 TcpLen: 40
TCP Options = MSS: 1460 SackOK TS: 100711305 0 NOP WS: 0
```

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 p art 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 fo llowing 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 acti vely this machine. It did not result in an actual co mpromise. There is no sign, that the network behind the switch is attacked. No corr elation 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 connec table. The router should deny packets from Interne t 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 se 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 corr ect 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 a ddresses are bold. I cut out some packets to make it not too long.

16:10:27.629537 **8:0:20:18:9b:f1 Broadcast** arp 60: arp who -has 192.168.241.8 (Broadcast) tell 192.168.241.6
16:10:27.630143 **8:0:2b:e5:63:9d Broadcast** arp 60: arp who -has 192.168.241.6 tell 192.16 8.241.8

Secondly we see a successful login:

16:10:35.204928 **8:0:2b:e5:63:9d 8:0:20:18:9b:f1** ip 60: 192.168.241.8.1023 > 192.168.241.6.512: S 2491359445:2491359445(0) win 512 <mss 1460> (ttl 64, id 57757)

16:10:35.205792 **8:0:20:18:9b:f1 8:0:2b:e5:63:9d** ip 60: 192.168.241.6.512 > 192.168.241.8.1023: S 396005587:396005587(0) ack 2491359446 win 8760 <mss 1460> (DF) (ttl 255, id 22623)

16:10:35.206023 **8:0:2b:e5:63:9d 8:0:20:18:9b:f1** ip 60: 192.168.241.8.1023 > 192.168.241.6.512: . 1:1(0) ack 1 win 32120 (DF) (ttl 64, id 57758)

16:10:35.206174 **8:0:2b:e5:63:9d 8:0:20:18:9b:f1** ip 60: 192.168.241.8.1023 > 192.168.241.6.512: P 1:2(1) ack 1 win 32120 (DF) (ttl 64, id 57759)

16:10:35.257885 **8:0:20:18:9b:f1 8:0:2b:e5:63:9d** ip 60: 192.168.241.6.512 > 192.168.241.8.1 023: . 1:1(0) ack 2 win 8760 (DF) (ttl 255, id 22624)

16:10:35.258054 **8:0:2b:e5:63:9d 8:0:20:18:9b:f1** ip 74: 192.168.241.8.1023 > 192.168.241.6.512: P 2:22(20) ack 1 win 32120 (DF) (ttl 64, id 57760)

16:10:35.308639 **8:0:20:18:9b:f1 8:0:2b:e5:63:9d** ip 60: 192.168.241.6.512 > 192.168.241.8.1023: . 1:1(0) ack 22 win 8760

```
(DF) (ttl 255, id 22625)
        4500 0028 5861 4000 ff06 c00d c0a8 f106      E..)Xb@. ........
        c0a8 f108 0201 03ff 179a 90d4 947f 20eb      ........ ........
        5018 2238 c659 0000 0063 6861 6e67           P."8.Y.. .chang

16:10:35.375791 8:0:20:18:9b:f1 8:0:2b:e5:63:9d ip 60: 192.168.241.6.512 > 192.168.241.8.1023: P 1:2(1) ack 22 win 8760
(DF) (ttl 255, id 22626)

........ Some more Packets

16:10:35.710763 8:0:20:18:9b:f1 8:0:2b:e5:63:9d ip 106: 192.168.241 .6.512 > 192.168.241.8.1023: P 3:55(52) ack 34 win 8760
(DF) (ttl 255, id 22629)
        4500 005c 5865 4000 ff06 bfd5 c0a8 f106      E.\Xe@.. ........
        c0a8 f108 0201 03ff 179a 90d6 947f 20f7      ........ ........
        5018 2238 fd0a 0000 4c61 7374 206c 6f67      P."8.... Last log
        696e 3a20 5475 6520 4a61 6e20 2039 2031      in:.Tue. Jan..9.1
        343a 3037 3a33 3920 6672 6f6d 2063 6861      4:07:39. from.cha
        6e67 2e73 686f 772e 6368 0d0a                ng.show. ch..


16:10:35.729992 8:0:2b:e5:63:9d 8:0:20:18:9b:f1 ip 60: 192.16 8.241.8.1023 > 192.168.241.6.512: . 34:34(0) ack 55 win 32120
(DF) [tos 0x10] (ttl 64, id 57763)
```

Later on, in the evening, we see a normal ARP request from machine .7.

```
18:15:11.685973 0:c0:4f:d3:42:a4 Broadcast arp 60: arp who -has 192.168.241.8 tell 19 2.168.241.7

18:15:11.924973 8:0:2b:e5:63:9d 0:c0:4f:d3:42:a4 arp 60: arp reply 192.168.241.8 is -at 8:0:2b:e5:63:9d (0:c0:4f:d3:42:a4 )
```

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

```
19:16:12.395335 0:c0:4f:d3:42:a4 8:0:20:18:9b:f1 arp 60: arp reply 192.168.241.8 is -at 0:c0:4f:d3:42:a4 (8:0:20:18:9b:f1 )
```

Finally it is tested, whether the .6 machine has update d the cache and a login is performed again. The m achine .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 f aked ARP response are repeated:

```
19:16:37.290023 0:c0:4f:d3:42:a4 8:0:20:18:9b:f1 ip 60: 192.168.241.8.1023 > 192.168.241.6.512: S
2644391733:2644391733(0) win 512 <mss 1460> (ttl 64, id 879)

19:16:37.290792 0:c0:4f:d3:42:a4 8:0:20:18:9b:f1 arp 60: arp rep ly 192.168.241.8 is -at 0:c0:4f:d3:42:a4 ( 8:0:20:18:9b:f1 )

19:16:37.291485 8:0:20:18:9b:f1 0:c0:4f:d3:42:a4 ip 60: 192.168.241.6.512 > 192.168.241.8.1023: S
1791708142:1791708142(0) ack 2644391734 win 8760 <mss 1460> (DF) (ttl 255, id 53651)

19:16:37.2917 00 0:c0:4f:d3:42:a4 8:0:20:18:9b:f1 ip 60: 192.168.241.8.1023 > 192.168.241.6.512: . 1:1(0) ack 1 win 32120
(DF) (ttl 64, id 880)
```

19:16:37.291880 **0:c0:4f:d3:42:a4 8:0:20:18:9b:f1** ip 60: 192.168.241.8.1023 > 192.168.241.6.512: P 1:2(1) ack 1 win 32120 (DF) (ttl 64, id 881)

19:16:37.309006 **0:c0:4f:d3:42:a4 8:0:20:18:9b:f1** arp 60: arp reply 192.168.241.8 is -at 0:c0:4f:d3:42:a4 (**8:0:20:18:9b:f1**)

19:16:37.319835 **0:c0:4f:d3:42:a4 8:0:20:18:9b:f1** arp 60: arp reply 192.168.241.8 is -at 0:c0:4f:d3:42:a4 (**8:0:20:18 :9b:f1**)

19:16:37.342098 **8:0:20:18:9b:f1 0:c0:4f:d3:42:a4** ip 60: 192.168.241.6.512 > 192.168.241.8.1023: . 1:1(0) ack 2 win 8760 (DF) (ttl 255, id 53652)

Formatted

19:16:37.342334 **0:c0:4f:d3:42:a4 8:0:20:18:9b:f1** ip 74: 192.168.241.8.1023 > 192.168.241.6.512: P 2:2 2(20) ack 1 win 32120 (DF) (ttl 64, id 882)

19:16:37.563419 **0:c0:4f:d3:42:a4 8:0:20:18:9b:f1** arp 60: arp reply 192.168.241.8 is -at 0:c0:4f:d3:42:a4 (**8:0:20:18:9b:f1**)

19:16:37.690129 **8:0:20:18:9b:f1 0:c0:4f:d3:42:a4** ip 60: 192.168.241.6.512 > 192.168.241. 8.1023: . 2:3(1) ack 22 win 8760 urg 1 (DF) (ttl 255, id 53655)

```
4500 0029 d197 4000 ff06 46d6 c0a8 f106    E..)..@. ..F.....
c0a8 f108 0201 03ff 6acb 4bf0 9d9e 374b    ........ j.K...7K
5030 2238 1874 0001 8063 6861 6e67         P0"8.t.. .chang
```

19:16:37.69 0725 **0:c0:4f:d3:42:a4 8:0:20:18:9b:f1** ip 66: 192.168.241.8.1023 > 192.168.241.6.512: P 22:34(12) ack 3 win 32120 (DF) [tos 0x10] (ttl 64, id 884)

..... Some Packets

19:16:37.789210 **8:0:20:18:9b:f1 0 0:c0:4f:d3:42:a4** ip 106: 192.168.241.6.512 > 192.1 68.241.8.1023: P 3:55(52) ack 34 win 8760 (DF) (ttl 255, id 53657)

```
4500 005c d199 4000 ff06 46a1 c0a8 f106    E..\..@. ..F.....
c0a8 f108 0201 03ff 6acb 4bf1 9d9e 3757    ........ j.K...7W
5018 2238 d140 0000 4c61 7374 206c 6f67    P."8.@.. Last log
696e 3a20 5475 6520 4a61 6e20 2039 2031    in:.Tue. Jan..9.1
343a 3436 3a33 3320 6672 6f6d 2063 6861    4:46:33. from.cha
6e67 2e73 686f 772e 6368 0d0a              ng.show. ch..
```

19:16:37.800865 **0:c0:4f:d3:42:a4 8:0:20:18:9b:f1** ip 60: 192.168.241.8.1023 > 19 2.168.241.6.512: . 34:34(0) ack 55 win 32120 (DF) [tos 0x10] (ttl 64, id 885)

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 s ystem. 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 fr e-quently. All the hosts have to be o n the same LAN.

In this case: The attacker uses a probably sniffed trust relationship. We see normal connects from this ne twork as the attacker might have seen. These are older machines and some of these computers do have a trusted relationship. There ar e other m a-chines 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 .8). 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 disturbin g the original (spoofed) machine. The packets sent by the server are a ctually sent to the malicious machine. The spoofed machine doesn't see them, as it's network card rejects the packets: they have the wrong MAC address. To be sure to keep the IP -MAC rela tionship in the cache these responses are sent from time to time.

1.3.4.6 Correlation

Candidate for CVE CAN -1999-0667 (http://www.cve.mitre.org/cgi -bin/cvename.cgi?name=CAN -1999-0667)

Article: (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 m ore 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 can not 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 a n-swer 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 gene rated 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 sec u-rity 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 deli v-ered, in SnortA (alert files), SnortS (scans), and OOSnort (data with partially dump). Provided is the snort rule the c onsultant 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 i nclude the entire information about the hosts, the links to loo k for more detailed information are at the end of this document. This includes the sources and the registration information. The consultant used 4 co l-ours:

Colour	Explanation
RED	Definitively 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.
YELLOW	This means there is a suspicious host or network problems. There is something "non -normal" going on. This migh t be scans, configuration problems, misconfigurated hardware. YELLOW points needs further investigation and maybe (long -time) observation.
GREEN	This as a false positive. E.g. A Trojan detection rule depends only on the port number and we see a connectio n 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.
BLUE	These are recommendation points. These are no attacks or compromise but t his hint helps to make it more difficult for an attacker or easier for the analyst.

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 documen t.

The IP source address is set to 193.168.x.x and is **NOT** the real home network address. This address was ch osen 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

Attack alert summary:

Signature from alertA	# of alerts	# of sources	# of destinations
Happy 99 Virus	2	2	2
site exec - Possible wu-ftpd exploit - GIAC000623	6	4	4
SITE EXEC - Possible wu-ftpd exploit - GIAC000623	7	1	4
Tiny Fragments - Possible Hostile Activity	7	5	6
External RPC call	13	8	3
Probable NMAP fingerprint attempt	15	14	13
connect to 515 from inside	56	2	3
SUNRPC highport access!	60	13	12
NMAP TCP ping!	96	21	20
Queso fingerprint	149	29	58
SMB Name Wildcard	218	33	33
Null scan!	283	204	196
SNMP public access	468	23	1
Back Orifice	1697	40	932
Broadcast Ping to subnet 70	1813	216	1
Attempted Sun RPC high port access	2542	20	33
TCP SMTP Source Port traffic	2893	4	2836
WinGate 1080 Attempt	4802	570	2655
Watchlist 000222 NET -NCFC	8166	45	26
Watchlist 000220 IL -ISDNNET-990517	30998	61	108
SYN-FIN scan!	56250	30	25751

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 $HOM E_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 i n a normal E -mail a content string like :"X -Spanska\:Yes" does not occur. So I would say this is pos itive alert.

10/05-03:59:51.460766 [**] Happy 99 Virus [**]			
Source	Port	Source	Port
216.6.117.11	41827	MOTOPHONE LTD	
11/06-16:06:44.170359 [**] Happy 99 Virus [**]			
Source	Port	Source	Port
209.94.224.13	2708	Village Online Div	Banyan -Net

Destination	Port	Destination	Port
193.168.253.41 193.168.6.35	25	MY.NET	SMTP

Both addresses occurred once. If these machines are mail server, this would ind icate incoming mail contai ning 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 machine s with the latest v i-russcanner / sign atures. 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 -ftpd exploit - GIAC000623"; content:"site exec";)
alert tcp any any -> $HOME_NET 21 (msg:"SITE EXEC – Possible wu -ftpd exploit - GIAC000623"; co ntent:"SITE EXEC";)
```

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

4 sources, 7 destinations

10/01-06:17:23.004770 [**]site exec - Possible wu -ftpd exploit – GIAC000623 [**]			
Source	Port	Source	Port
208.61.44.215	3746-3820	BellSouth.net Inc	
10/16-16:55:26.342617 [**] site exec - Possible wu -ftpd exploit – GIAC000623 [**]			
Source	Port	Source	Port
24.31.88.99	62275-62281	ServiceCo LLC - Road Ru n-ner	
10/07-07:00:03.744205 [**] site exec - Possible wu -ftpd exploit – GIAC000623 [**]			

Source	Port	Source	Port
202.9.188.89	61693	DISHNETDSL LTD	
10/04-11:56:14.289566 [**] site exec - Possible wu -ftpd exploit – GIAC000623 [**]			
Source	Port	Source	Port
63.202.13.20	1188	Pacific Bell Internet Se r-vices,Inc	HP Web Admin

Destination	Port	Destination	Port
193.168.205.94 193.168.97.206 193.168.99.130 193.168.130.81 193.168.130.242 193.168.205.94 193.168.221.82	21	MY.NET	ftp

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 r e-stricting the use of specific features. This could be to not to accept site -exec commands. Limit the access through a TCP wra p-per. The dest ination machines above should be analysed.

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

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 destina tions

10/08-11:06:57.334675 [**] Tiny Fragments - Possible Hostile Activity [**]			
Source	Port	Source	Port
62.6.71.0		IMSNET	
09/28-22:02:54.922273 [**] Tiny Fragments – Possible Hostile Activity [**]			
Source	Port	Source	Port
216.43.55.44		McLeodUSA Inc orporated	
11/16-14:39:19.160234 [**] Tiny Fragments – Possible Hostile Activity [**]			
Source	Port	Source	Port
202.156.51.76		SINGAPORE CABLE V ISION LTD	
10/19-09:48:03.549003 [**] Tiny Fragments - Possible Hostile Activity [**]			
Source	Port	Source	Port
192.206.151.152		Toronto Star Newspapers	
09/26-21:25:17.293957 [**] Tiny Fragments – Possible Hostile Activity [**]			
Source	Port	Source	Port
172.157.126.93		America Online, Inc	

Destination	Port	Destination	Port
193.168.1.8 193.168.181.144 193.168.201.2 193.168.201.198 193.168.202.102 193.168.211.2		MY.NET	

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 ba sed 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:

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

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

8 sources, 3 destinations

Source	Port	Source	Port
63.162.239.69	975 3655 4496	BELZ INVESTCO	!!low port
10/10-20:23:36.018641 [**] Externa l RPC call [**]			
Source	Port	Source	Port
200.191.80.206 200.191.80.181	931 634 665	AcessoNet Ltda	!!low port ginad Sun DR
11/10-17:45:06.672352 [**] External RPC call [**] (2068 alerts from this machine!!)			
Source	Port	Source	Port
211.46.110.81	708 4910	Korea Internet	
11/01-12:11:00.474172 [**] External RPC call [**]			
Source 38.200.223.8	Port 2473	Source PSINETA	Port Aker-cdp
10/14-19:41:47.442871 [**] External RPC call [**] (1149 alerts from this range!!)			
Source	Port	Source	Port
24.23.151.112 24.7.227.215	3306 **5**	@Home Network	MySQL Remote Job Entry
10/28-19:41:44.513820 [**] External RPC call [**]			
Source 12.34.21.196	Port 700	Source CFS EUROPE LTD	Port

Destination	Port	Destination	Port

193.168.6.15 (9x) 193.168.15.127 (3x) 193.168.100.130 (1x)	111	MY.NET	Portmapper

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:SFPU;)
```

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

14 sources, 13 destinations

11/03-06:30:31.521549 [**] Probable NMAP fingerprint attempt [**]			
Source	Port	Source	Port
24.95.192.51	1 10	ServiceCo LLC – RoadRunner	TCP Port Service Multiplexer
10/27-09:57:08.723116 [**] Probable NMAP fingerprint attempt [**]			
Source	Port	Source	Port
195.132.57.32	0	CYBERCABLE FR	This is a reserved port!!!
11/22-22:44:52.018936 [**] Probable NMAP fingerprint attempt [**]			
Source	Port	Source	Port
24.69.214.58	2648	Shaw Fiberlink ltd	Upsnotifyprot
10/22-16:11:47.728191 [**] Probable NMAP fingerprint attempt [**]			
Source	Port	Source	Port
205.251.201.36	1490	Cable Atlantic Inc	insitu-conf
10/08-18:15:35.119165 [**] Probable NMAP fingerprint attempt [**]			
Source	Port	Source	Port
132.178.218.181	3449	Boise State University	
11/07-19:50:33.189009 [**] Probable NMAP fingerprint attempt [**]			
Source	Port	Source	Port
24.6.151.155 24.180.134.156 24.108.140.159 24.9.64.57	62110 57446 6699 0	@Home Network	
10/20-14:53:22.737915 [**] Probable NMAP fingerprint attempt [**]			
Source	Port	Source	Port
62.226.88.88	1095	Deutsche Telekom AG	ODD Packet - RAT
11/01-18:12:25.450405 [**] Probable NMAP fingerprint attempt [**]			
Source	Port	Source	Port

169.233.14.204 128.54.203.218	1092 6699	University of California	
11/09-18:31:50.987024 [**] Probable NMAP fingerprint attempt [**]			
Source	Port	Source	Port
193.231.207.72	48806	ICAFE-NET	
10/06-13:38:00.767581 [**] Probable NMAP fingerprint attempt [**]			
Source	Port	Source	Port
128.194.79.228	195	Texas A&M University	DNSIX Network Level Module Audit

Destination	Port	Destination	Port
193.168.211.94	1270	MY.NET	
193.168.219.146	2529		UTS FTP
193.168.224.150	4999		
193.168.70.93	6699		
193.168.204.170	1632		PAMMRATC
193.168.162.39	21		FTP
193.168.202.134	5000		Sockets de Troie
193.168.204.202	1078		
193.168.207.14	4389		
193.168.201.126	6688		
193.168.207.14	4968		
193.168.60.38	21		FTP
193.168.218.162	2803		
193.168.206.50	80		HTTP

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 for 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 on ly the ones I find interesting. Some statistical inform ation can be found b e-low 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 num ber count goes over the range of 2^(32) and r estarts with 0. But more likely is a scan.

96 sources, 21 destinations

10/27-10:40:20.889310 [**] NMAP TCP ping! [**]			
Source	Port	Source	Port
192.102.197.234	53 80	Intel Corporation	DNS HTTP
10/04-06:46:22.064365 [**] NMAP TCP ping! [**]			
Source	Port	Source	Port
202.187.24.3	80	JARING-UNITAR2	HTTP
Source	Port	Source	Port
63.119.91.2	80	UUNET Technologies, Inc	HTTP
11/13-06:25:17.371413 [**] NMAP TCP ping! [**]			
Source	Port	Source	Port
64.64.226.2	80	Teligent, Inc	HTTP
10/25-04:29:51.770991 [**] NMAP TCP ping! [**]			
Source	Port	Source	Port
204.155.48.3	80	Southwire Company	HTTP
09/26-05:40:00.709907 [**] NMAP TCP ping! [**]			
Source	Port	Source	Port
2.2.2.2	80	Reserved Range	HTTP

Destination	Port	Destination	Port
193.168.1.3	53	MY.NET	DNS
193.168.1.5	53		
193.168.1.7 (5808 alerts!)			
193.168.1.8	25		SMTP
193.168.1.9	25		SMTP
193.168.1.10			
193.168.253.43 (589 alerts)			
193.168.6.47			
193.168.6.35	25		
193.168.100.230 (1320 alerts)	25		
	25		
193.168.6.47			
193.168.253.42	25		
193.168.253.125	80		HTTP
193.168.6.14	80		HTTP

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 definitively 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.

09/26-04:27:59.343599 [**] Queso fingerprint [**] (During almost 3 months)			
Source	Port	Source	Port
24.3.161.193	range	@Home Network	
10/30-05:42:15.161598 [**] Queso fingerprint [**]			
Source	Port	Source	Port
195.115.7.2	range	CEGETEL	
10/28-17:08:22.282509 [**] Queso fingerprint [**] (during a month), this one is a fix IP and scans the network for port 23 (telnet), 113(authentication/verification),1080 (socks).			
Source	Port	Source	Port
129.242.219.27	range	University of Tromso	
09/26-08:38:11.102016 [**] Queso fingerprint [**]			
Source	Port	Source	Port
64.80.63.121	range	PaeTec Communications, Inc.	
10/28-12:28:18.800360 [**] Queso fingerprint [**]			
Source	Port	Source	Port
24.163.42.82	3347	ServiceCo LLC - Road Runner	Phoenix RPC
09/27-07:31:17.992797 [**] Queso fingerprint [**]			
Source	Port	Source	Port
128.253.247.116	26	Cornell University	Unassigned

Destination	Port	Destination	Port
193.168.145.9	110	MY.NET	POP3
193.168.253.112	443		HTTPS
193.168.217.26	6346		???
193.168.227.194	100		newacct
193.168.227.10	2720,		wkars,
193.168.130.190	3060		interserver
193.168.211.146	1738		GameGen1

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-NULL Scan";flags:0; seq:0; ack:0;)
```

No flags are not normal. This signature can be trusted.

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

128.253.247.116		5 instances of Queso fingerprint 8 instances of Null scan!
24.180.134.156		1 instances of Null scan! 1 instances of Probable NMAP fingerprint attempt 1 instances of NMAP TCP ping!

These are not top scanner. 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:

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

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 51 instances of SYN -FIN scan! 1096 instances of TCP SMTP Source Port traffic
24.112.150.20	Rogers@Home	1 instances of SYN -FIN scan! 8 instances of Null scan!
24.200.140.155	Videotron Ltee	1 instances of SYN -FIN scan! 4 instances of Null scan!

These attackers did scan the net work 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 po sitives may be frequent.

1 sources, 2 destinations

11/22-11:24:06.406682 [**] connect to 515 from inside [**]			
Source	Port	Source	Port
193.168.179.78	2274	MY.NET	
11/22-11:33:56.296324 [**] connect to 515 from inside [**]			
Source	Port	Source	Port
193.168.179.78	2707	MY.NET	EMCSYMAPIPORT

Destination	Port	Destination	Port
64.244.202.110 64.244.202.66 193.168.100.3	515	MY.NET	line printer spooler BSD lpd(8)

Is 193.168.100.3 a printer spooler?

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

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

If these machines do act as printerserver (two of them outside and requested from inside??) this could b e 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 misconfigur a-tion.

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 diffe rences. 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 HIGHPORT access

Alert trigger:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 32771 (msg:"Sun RPC high port a ccess";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 fr equent.

13 sources, 12 destin ations

10/03-22:15:56.551396 [**] SUNRPC highport access! [**]			
Source	Port	Source	Port
216.10.12.30	2078	Virtual Development Inc	Always this port
10/26-00:28:54.652275 [**] SUNRPC highport access! [**]			
Source	Port	Source	Port
216.148.218.160	443	TCG CERFnet	HTTPS
10/14-12:29:16.379139 [**] SUNRPC highport access! [**]			
Source	Port	Source	Port
195.34.28.117	3191	PTTNET-DIALUP3	Doing also 1080 scan!?
10/19-09:37:08.111361 [**] SUNRPC highport access! [**]			
Source	Port	Source	Port
205.188.3.239	5190	America Online, Inc.	America-Online
11/11-11:08:56.576798 [**] SUNRPC highport access! [**]			
Source	Port	Source	Port
211.46.110.81	690	Korea Internet Inform ation Service	1 SUNRPC highport! 2 External RPC call 276 of SYN-FIN scan! 1789 of TCP SMTP Source Port traffic
10/05-23:44:23.183592 [**] SUNRPC highport access! [**]			
Source	Port	Source	Port
212.86.129.227	888	TOOLNINE-NET	CD Database Protocol Login and environment pas s-ing
10/16-20:58:51.128761 [**] SUNRPC highport access! [**]			
Source	Port	Source	Port
216.10.12.2	4600	Virtual Development Inc	Piranha1

Destination	Port	Destination	Port
193.168.206.222	32771		
193.168.202.242			
193.168.212.186			
193.168.228.62			
193.168.97.59			
193.168.53.23			
193.168.253.114			
193.168.206.218			
193.168.53.14			
193.168.6. 15			
193.168.140.51			
193.168.179.78			

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 216.148.218.160 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 a ccess";flags:S;)
```

This alert just triggers on the port too 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

11/01-14:00:49.592878 [**] Attempted Sun RPC high port access [**]			
Source	Port	Source	Port
205.188.153.xxx	4000 53	America Online, Inc (2534 times)	Terabase

Destination	Port	Destination	Port
193.168.1.8	32771		
193.168.144.42			
193.168.202.242			
193.168.204.134			
193.168.205.130			
193.168.206.222			
193.168.209.182			
193.168.217.218			
193.168.219.130			
193.168.220.194			
193.168.220.78			
193.168.221.126			
193.168.222.98			
193.168.223.18			
193.168.224.214			
193.168.225.106			
193.168.225.210			
193.168.225.98			
193.168.226.198			
193.168.226.74			
193.168.227.170			
193.168.227.50			
193.168.228.22			
193.168.97.152			
193.168.97.163			
193.168.97.202			
193.168.97.62			

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 OO datafile we see splitted SF with payload 00. This could be a part of the ttdbserv 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"; co ntent:"public";)
```

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

23 sources, 1 destinations

Source	Port	Source	Port
193.168.101 .* 193.168.97.* 193.168.98.*			

Destination	Port	Destination	Port
193.168.101.192	161		

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 th e public community name a read -only access to the MIB (Management I nformation 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 m achine 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 inve stigated!

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

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"; co n-
tent:"CKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA|0000|";)
```

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

33 sources, 33 destinations, not all listed

Internal Source 193.168.101.160	External Source	Name	Destinations 193.168.101.192
	141.157.99.21 141.157.98.201	Bell Atlantic	193.168.6.15

	169.254.184.161	LINKLOCAL	193.168.101.53 193.168.101.113 193.168.101.117 193.168.101.153 193.168.101.147 193.168.101.145 193.168.101.89 193.168.101.152 193.168.101.158
193.168.98.154			193.168.101.53 193.168.101.89 193.168.101.145 193.168.101.158
	129.37.159.177	IBM-RSCH-NET	193.168.100.130
193.168.97.207			193.168.101.53 193.168.101.99 193.168.101.145 193.168.101.153
	130.227.195.57	DENET-227	193.168.152.110
193.168.97.120			193.168.101.99 193.168.101.117 193.168.101.153

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 vulnerabilites (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=can-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 destinatio n, just the most "pinger"

88 times, 12 days (with interrupts because of missing data) up to 5 times per minute	49 time, interrupts, 3 days
Source	Source
193.231.169.166	213.154.131.131
55 times, one day	43 times, 30 minutes
Source	Source
193.226.60.17 9	193.231.220.71
50 times, one day within 46 minutes	43, two days
Source	Source
193.231.220.101	217.10.206.79

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 REPLY. 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 definitively 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 forward 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:

```
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, 932destinations. I just show some addresses with more than one connection for source or destination.

396 times			
Source	Port	Source	Port
62.136.90.120	1101 3327 3539 3599 1280		BBARS
99 times			
Source	Port	Source	Port
213.43.69.72	31338		
291 times			
Source	Port	Source	Port
63.46.46.143	range		
111 times			
Source	Port	Source	Port
203.148.182.108			
79 times			
Source	Port	Source	Port
203.155.130.111	31338		
78 times			
Source	Port	Source	Port
209.94.199.186	31338		
75 times			
Source	Port	Source	Port
213.43.69.126	31338		

70 times			
Source	Port	Source	Port
168.120.12.33	31338		

I just copy and pasted the ten m ost used MY.NET addresses			
Destination	Number	Destination	Port
193.168.98.150	7	MY.NET	
193.168.97.208	7		
193.168.98.82	6		
193.168.98.119	6		
193.168.98.151	6		
193.168.98.77	6		
193.168.97.142	6		
193.168.98.81	6		
193.168.98.195	5		
193.168.98.118	5		

Normally, if I see a low port (e.g. 80) and and the related client highport I look into the datastream an 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.
193.168.162.36, 193.168.60.14, 193.168.98.149, 193.168.98.194, 193.168.218.50.

There are just scans. But there are either direct trials or successful connections to machines inside. This is BackOrifice1 for wi n-dows95/98. If these machines uses this OS an investigation is badly necessary!

Recommendation: Install a network virus sca nner 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 int ernal addresses going on port 25 to the mail server.

4 sources, 2836 destinations

2 times			
Source	Port	Source	Port
194.88.77.240	25	LONDON1 -DIAL-POOL2	SMTP
6times			
Source	Port	Source	Port
194.67.168.11	25	RELSOFT	
1879 times			
Source	Port	Source	Port

211.46.110.81	25	Korea Internet Inform ation Service	
1096 times (scanning)			
Source	Port	Source	Port
24.7.227.215	25	@Home Network	

Destination	Port	Destination	Port
193.168.160.89 193.168.184.36 193.168.185.80 193.168.2.82 193.168.211.56 193.168.25.5 193.168.1.2	25		

194.67.168.11 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 definitively 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 numbers show a large scan activity. These scans represent a security risk. There are exploits which relay attacks to other machines. (http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0290, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0291, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0441, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0494, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-1048)

These internal hosts do have the most traffic:

Internal host	count
193.168.206.118	372
193.168.225.154	126
193.168.60.11	75
193.168.60.8	67
193.168.60.16	40
193.168.60.38	34
193.168.203.78	34
193.168.53.91	29
193.168.222.102	25
193.168.53.219	24

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

Signature from alertA	# of alerts	# of sources	# of destinations
Watchlist 000222 NET-NCFC	8166	45	26
Watchlist 000220 IL-ISDNNET-990517	30998	61	108

The Watchlist 000222 NET-NCFC registrates 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.

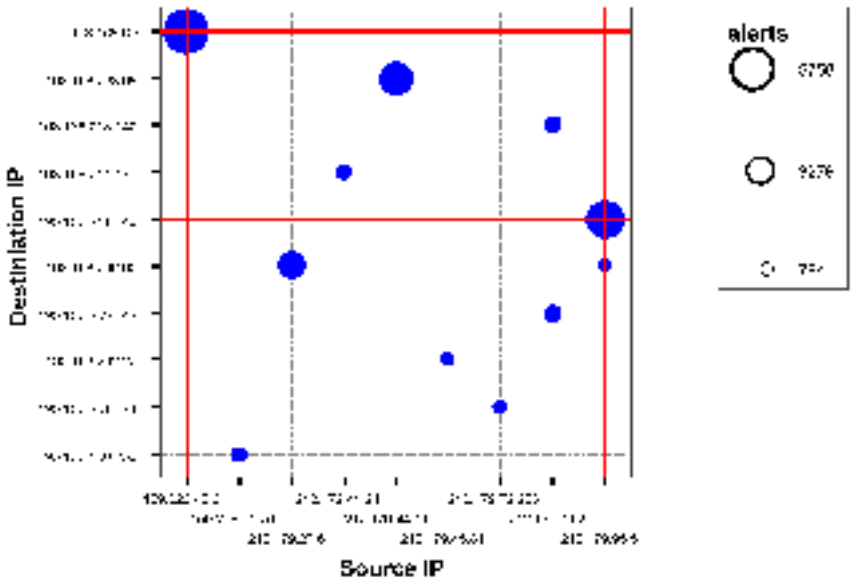
The Watchlist 000220 IL-ISDNNET-990517 does the same for the "Cable-Modem-Experiment" network. There were a lot of alerts coming from this. Again some with port 0 or low ports.

2.3 TOP Talker

This section is divided into top talker incoming and top talker f rom MY.NET. This is repeated with the m achines with the most e x-ploits. A top talker could be a scanner. Then there is a link graph with the busiest m achines and their partner.

External Top Talker	Count	Internal Top Talker	Count
160.78.49.191	7199	193.168.6.7	5808
208.61.4.20 7	6635	193.168.211.146	4814
159.226.45.3	6296	193.168.223.98	3938
212.179.95.5	6117	193.168.206.90	3918
209.92.40.32	4967	193.168.203.142	1640
212.179.27.6	4011	193.168.218.142	1463
212.179.79.2	3950	193.168 .214.170	1371
212.179.44.115	3936	193.168.100.230	1302
63.195.56.20	3897	193.168.202.22	952
130.89.229.48	3860	193.168.201.174	803

Top Talker from machines which do have most traffic to each other:



Most used ports:

External IP addresses		Internal IP addresses	
Port	Count	Port	Count
21	19619	21	19639
53	18307	53	18341
9704	14184	9704	14184
27374	3572	25	11052
25	2894	6699	9762
1067	2699	4619	5734
4000	2536	4922	4813
6699	1905	1080	4809
1498	1823	27374	3577
1574	1727	6688	3295

Top Alerter:

External		Internal	
SRC IP	Count	SRC IP	Count
160.78.49.191	7199	193.168.101.160	93
208.61.4.207	6635	193.168.98.106	58
159.226.45.3	6296	193.168.101.142	54
212.179.95.5	6117	193.168.98.174	49
209.92.40.32	4967	193.168.97.185	44
212.179.27.6	4011	193.168.97.171	40
212.179.79.2	3950	193.168.97.204	37
212.179.44.115	3936	193.168.98.122	36
63.195.56.20	3897	193.168.98.197	32
130.89.229.48	3860	193.168.97.178	31

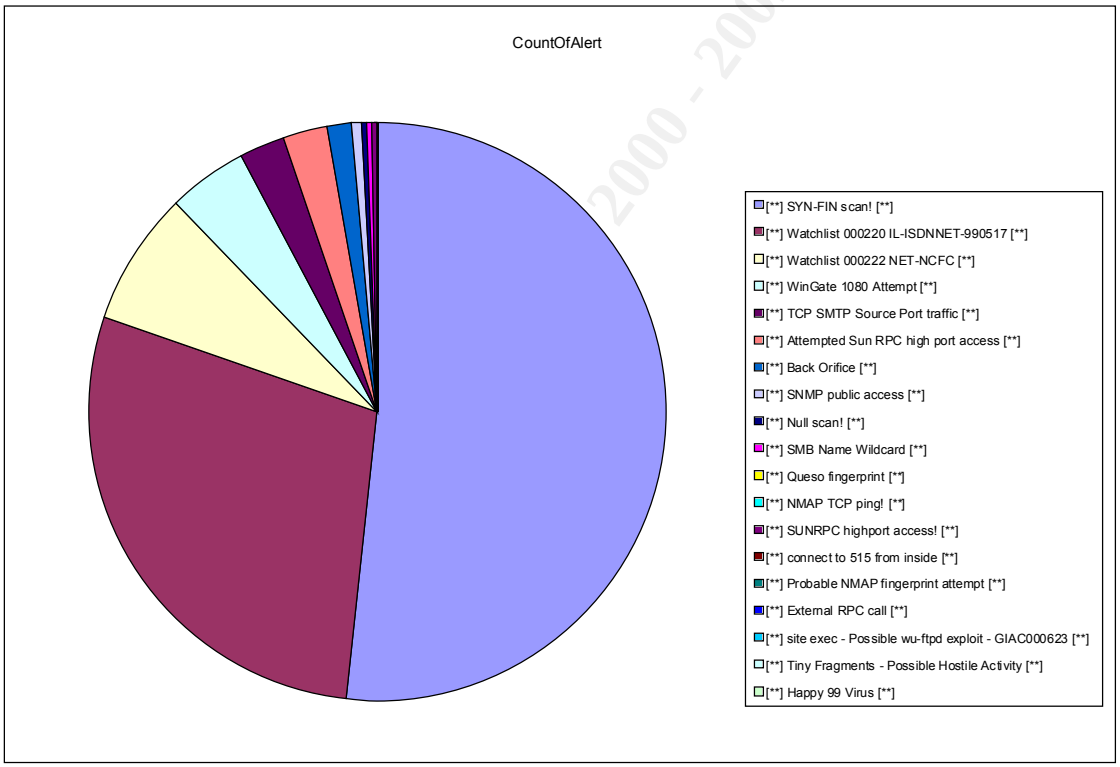
Most used Exploits:

Internal – Internal Alerts			
Source	Destination	Count	Alert
193.168.101.160	193.168.101.192	93	[**] SMB Name Wildcard [**]
193.168.98.106	193.168.101.192	58	[**] SNMP public access [**]
193.168.101.142	193.168.100.3	54	[**] connect to 515 from inside [**]
193.168.98 .174	193.168.101.192	49	[**] SNMP public access [**]
193.168.97.185	193.168.101.192	44	[**] SNMP public access [**]
193.168.97.171	193.168.101.192	40	[**] SNMP public access [**]
193.168.97.204	193.168.101.192	37	[**] SNMP public access [**]
193.168.98.122	193.168.101.192	36	[**] SNMP public ac cess [**]
193.168.98.197	193.168.101.192	32	[**] SNMP public access [**]
193.168.97.178	193.168.101.192	31	[**] SNMP public access [**]

Internal – External Alerts			
Source	Destination	Count	Alert
193.168.179.78	64.244.202.66	1	[**] connect to 515 from inside [**]
193.168.179.78	64.244.202.110	1	[**] connect to 515 from i nside [**]

External – Internal Alerts			
Source	Destination	Count	Alert
159.226.45.3	193.168.6.7	5758	[**] Watchlist 000222 NET -NCFC [**]
212.179.95.5	193.168.211.146	4810	[**] Watchlist 000220 IL -ISDNNET-990517 [**]
212.179.44.115	193.168.223.98	3936	[**] Watchlist 000220 IL -ISDNNET-990517 [**]
212.179.27.6	193.168.206 .90	3120	[**] Watchlist 000220 IL -ISDNNET-990517 [**]
212.179.79.2	193.168.203.142	1638	[**] Watchlist 000220 IL -ISDNNET-990517 [**]
212.179.79.2	193.168.218.142	1459	[**] Watchlist 000220 IL -ISDNNET-990517 [**]
212.179.41.24	193.168.214.170	1353	[**] Watchlist 000220 IL -ISDNNET-990517 [**]
159.226.91.20	193.168.100.230	1131	[**] Watchlist 000222 NET -NCFC [**]
…	……	….	….
205.188.153.108	193.168.221.246	488	[**] Attempted Sun RPC high port access [**]

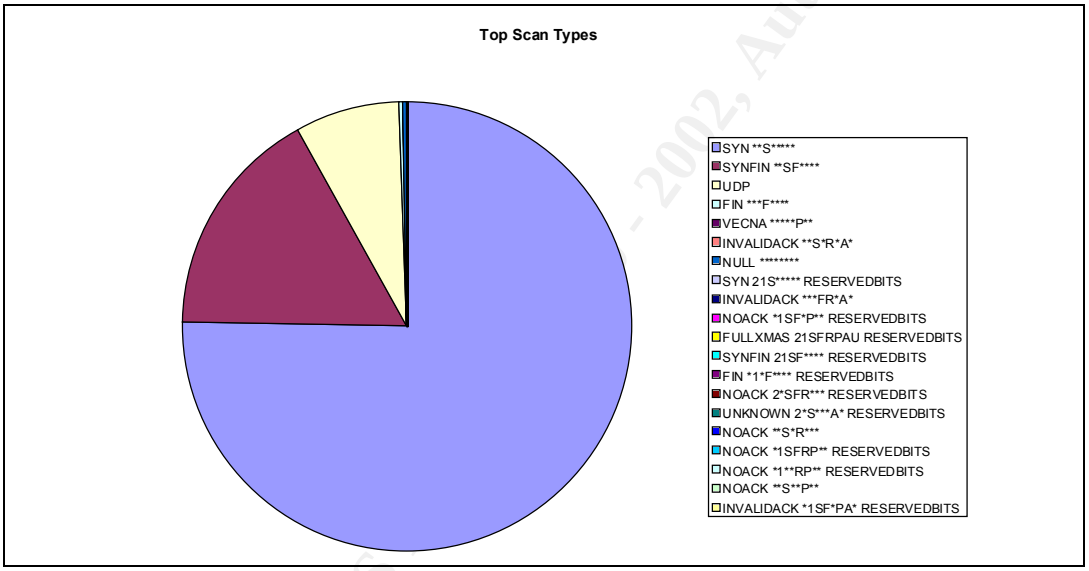
General Alert Chart:



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.

Scan Type	Counts	Scan Type	Counts
SYN **S*****	235386	FULLXMAS 21SFRPAU RESERVE DBITS	28
SYNFIN **SF****	51628	SYNFIN 21SF**** RESERVE DBITS	16
UDP	23954	FIN *1*F**** RESERVE DBITS	16
FIN ***F****	454	NOACK 2*SFR*** RESERVE DBITS	14
VECNA *****P**	351	UNKNOWN 2*S***A* R ESERVE DBITS	14
INVALIDACK **S*R*A*	281	NOACK **S*R***	14
NULL **** ****	226	NOACK *1SFRP** RESERVE DBITS	14
SYN 21S***** RESERVEDBITS	104	NOACK *1**RP** RESERVE DBITS	14
INVALIDACK ***FR*A*	60	NOACK **S**P**	13
NOACK *1SF*P** RESERVEDBITS	29	INVALIDACK *1SF*PA* RESERVE DBITS	13



Addresses with the most scans:

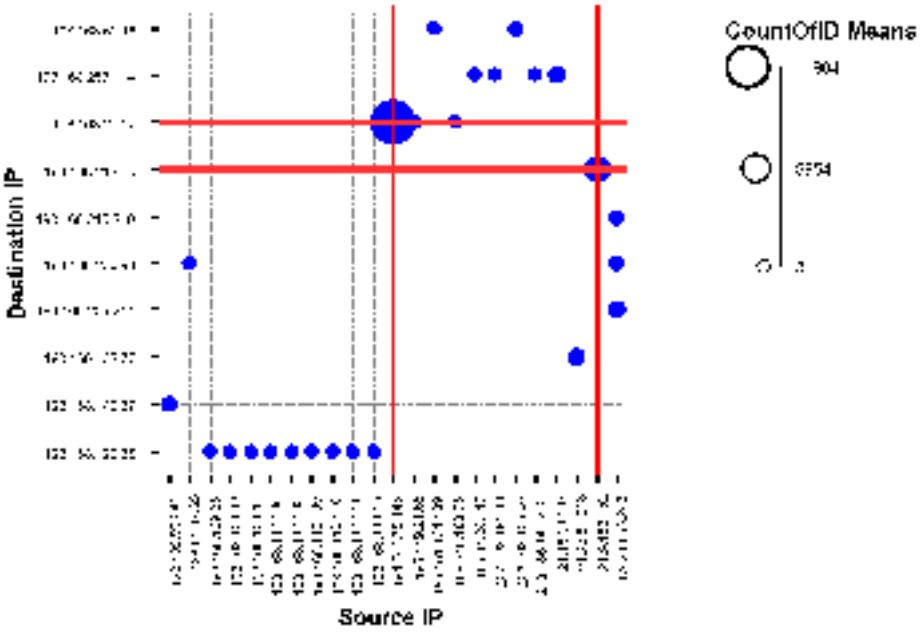
External Source addresses doing the most scans	Count
66.9.27.254	20649
62.252.21.241	13057
194.244.78.145	11904
63.88.175.201	11718
62.157.23.237	9641

63.248.55.245	9073
62.96.169.86	8939
24.23.151.112	8763
64.50.161.162	8635
160.78.49.191	7192
128.211.237.11	7003
211.49.165.9	6919
208.61.4.207	6634
62.155.244.68	6401
216.191.162.145	6093
63.198.207.51	6033
209.92.40.32	4956
208.214.247.60	4881
146.101.14 7.251	4855
199.239.94.98	4836

Internal Source addresses doing the most scans	Count	Types of Scans
193.168.224.150	**2981**	**SYN **S*****, UDP**
193.168.221.82	**2668**	**SYN **S*******
193.168.5.25	**2311**	**UDP**
193.168.1.3	577	UDP
193.168.110.111	270	UDP
193.168.110.16	267	UDP
193.168.109.41	252	UDP
193.168.110.105	215	UDP
193.168.110.108	160	UDP
193.168.110.109	120	UDP
193.168.109.40	109	UDP
193.168.110.110	100	UDP
193.168.213.58	94	SYN **S*****
193.168.109.38	93	UDP
193.168.1.4	22	UDP
193.168.152.165	14	UDP
193.168.101.1	**4**	**INVALIDACK 2**FR*A* RESERVEDBITS**
193.168.19.10	**1**	**INVALIDACK 2**FR*A* RESERVEDBITS**

Most scanned targets

Most scanned targets	Count	Most scanned targets	Count
193.168.220.2	11926	193.168.70.121	1206
193.168.218.50	4710	193.168.204.26	1169
193.168.253.114	1976	193.168.204.218	1127
193.168.206.94	1799	193.168.212.114	1083
193.168.162.77	1759	193.168.205.246	1076
193.168.120.36	1591	193.168.97.59	996
193.168.205.214	1589	193.168.98.168	973
193.168.215.210	1367	193.168.98.171	900
193.168.60.16	1306	193.168.162.36	867
193.168.140.57	1220	193.168.211.254	719



2.5 Possibly compromised machines

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

193.168.1.2	[**] Watchlist 000222 NET -NCFC [**] [**] TCP SMTP Source Port traffic [**] [**] SYN -FIN scan! [**]	8 2 1
193.168.6.15	[**] SMB Name Wildcard [**] [**] External RPC call [**] [**] SUNRPC highport access! [**]	53 9 1
193.168.6.35	[**] Watchlist 000222 NET -NCFC [**] [**] Queso fingerprint [**] [**] NMAP TCP ping! [**] [**] Happy 99 Virus [**]	2 1 1 1
193.168.19.10	Scanning external network	
193.168.100.3	[**] connect to 515 from inside [**] [**] SYN -FIN scan! [**]	54 4
193.168.100. 130	[**] SMB Name Wildcard [**] [**] External RPC call [**] [**] SYN -FIN scan! [**] [**] External RPC call [**] [**] External RPC call [**]	4 1 1 1 1
193.168.101.1	Scanning external network	
193.168.101.192	[**] SNMP public access [**] [**] SMB Name Wild card [**]	468 93
193.168.203.198	Connection with machines 205.188.3.*	10
193.168.206.222	[**] Attempted Sun RPC high port access [**] [**] SUNRPC highport access! [**] [**] SYN -FIN scan! [**] Possible Exploits w. tooltalk	298 21 3
193.168.218.50	[**] SYN -FIN scan! [**] [**] WinGate 1080 Attempt [**] [**] Watchlist 000220 IL -ISDNNET -990517 [**] [**] Back Orifice [**]	3 2 2 2
193.168.218.106	Connection with 207.172.3.46, several low ports (e.g. 0,1,10,....), making OS detection	93
193.168.225.98	[**] Attem pted Sun RPC high port access [**] [**] SYN -FIN scan! [**] [**] Back Orifice [**]	37 1 1

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 policies.
- All the machines should be audited.
- The customer should implement a 24hx7days 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 m a-chine. Log the content of packets causing an alert at least as long as your examin ation time lasts.
- There should be a perimeter defence, a stateful firewall or better a proxyfilter. If there are load problems put the filter b e-tween two packet filter firewalls. The packet filter firewalls will keep away a lot of traffic and relieve the proxy or stateful fir e-wall.
- 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. I nclude also the plugins. Manage your ruleset for your network in a different set so you can include it ea sily.
- Install host security software. E.g. Virusscann er, 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 ne twork is weak. There are several red alerts where react ion is required immediately. Also I suppose your ne twork 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 a ttacks 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:

1. SnortA – 54 Files, 15 335 kB. The SNORT alert file.
2. SnortS - 42 files, 21 748 kB. The file with the scans.
3. OOs – OS-check file partially with dump.

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 '\[**\]' alertA1.txt > alertA2.txt"

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

"grep –v '\[**\]' alertA1.txt > dust.txt"

I wanted then 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-ZZ-960208 or LU-RESTENA-193-168-64-127 or METRONET.

"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
```

```
    if ( length($elem) < 10 )
    {
    }
    else
    {
       # unless it is smaller print it out (could have used     this command...)
       print $elem, " \n";
    }
};
```

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 l ittle bit more me mory. Therefore I added swap to my linux m achine 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 wil d-cards.

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

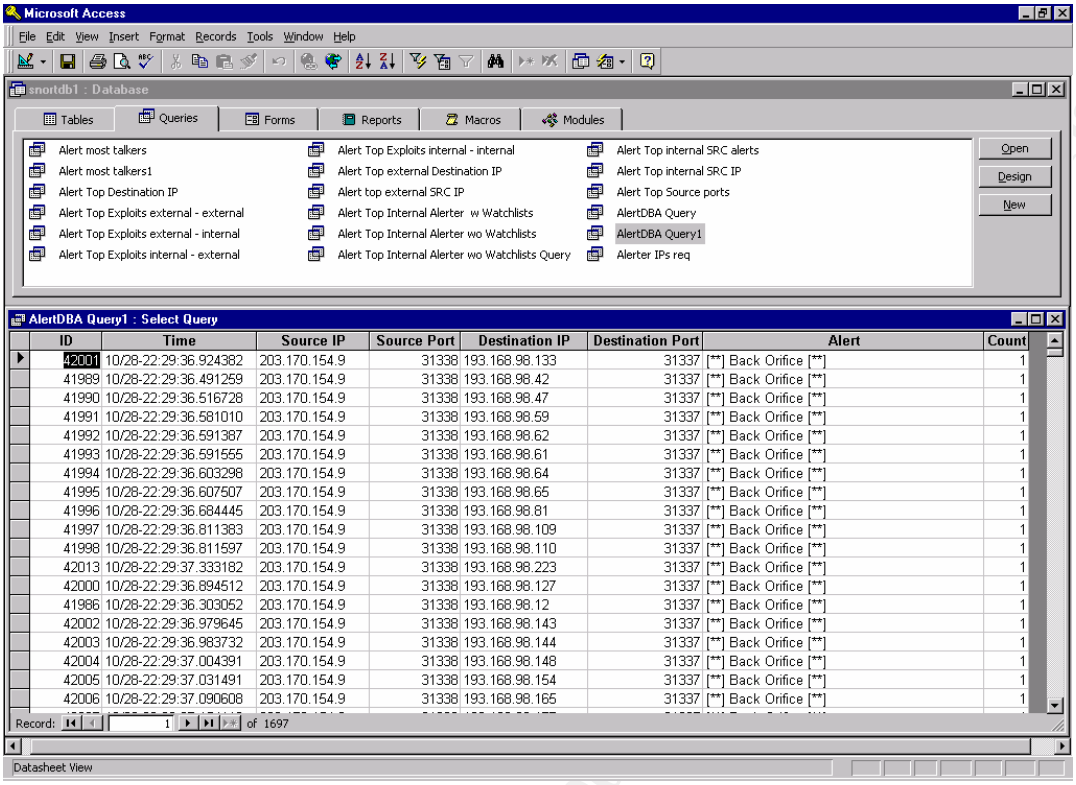
```
#!/usr/bin/perl

$_ = `cat $ARGV[0]`;

LOOP :while(/(.*?) ( \[\*\*\].*?\[\*\*\])(.*?)\:(.*?)->(.*?)\:(.*?)\n/sgc) {
    print $1, "; ", $2, " ;", $3, " ; ", $4, ";", $5, " ; ", $6, "    \n";
};
```

I imported then the alertDBA.txt into ACCESS using its import functio ns. 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 p erl 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:

SNORT	http://www.snort.org
SNORTSNARF	http://www.silicondefense.com/
ACID	http://www.cert.org/kb/acid/
GEEKTOOLS	http://www.geektools.com/
Correlation for the Analysis	http://www.sans.org/y2k/practical/Guy_Bruneau.doc
Nessus RIPE	http://www.nessus.org/ http://www.ripe.net
ARIN	http://www.arin.net
SANS	http://www.sans.org
Whitehats IDS	http://www.whitehats.com/ids
CVE	http://www.cve.mitre.org/
APNIC	http://www.apnic.net
Back Orifice	http://www.cultdeadcow.com/tools/
tripwire	http://www.tripwiresecurity.com/ http://www.tripwire.org/

Last Updated: July 28th, 2017

Upcoming Training

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