Global Information Assurance Certification Paper

GIAC Level Two: Intrusion Detection In Depth
SANS Parliament Square, London
June 20-23, 2001

Assignments for GCIA (v2.9)

Robert Turner

Table of Contents

1 Assignment 1 – Network Detects

1.1 ICMP Time Exceeded Packet to Network Address

1.1.1 Source of Trace

The event was alerted between the 6th and 9th November 2000 through an ISS RealSecure IDS system, from which the following information was obtained:

Date	08/11/2000
Time	16:39:08
Detect name	Trace_Route
Source Address	B.B.197.0
Destination Address	202.104.139.195

This was noticed immediately as the B.B.197.0 address is a /24 network address, and the only time that B.B.197.0 could be a valid address would be if the subnet was /23 or below (with the node name taking nine or more bits from the IP address).

TCPDump [TCPDump] was logging traffic data on a machine on this subnet, and the following trace was extracted using the B.B.197.0 address as a key:

```
tcpdump -r 16.35 -nX 'host B.B.197.0'
16:37:03.573766 210.77.146.1 > B.B.197.0: icmp: time exceeded in-transit [tos 0xc0]
0x0000   45c0 0038 b34c 0000 f601 0e45 d24d 9201        E..8.L.....E....
0x0010   BBBB c500 0b00 9fa3 0000 0000 4500 003c        BB..........E..<
0x0020   8545 0000 0101 018a BBBB c500 ca68 8bc3        .E......BB...h..
0x0030   0000 3d5c 0100 1700                            ..=\....
```

From this information extracted from this packet [RFC0777], an examination of the firewall logs produced the following information:

Date	08/11/2000
Time	16:37:08
Firewall Name	fw1
Protocol	ICMP
Source Address	210.77.146.1
Destination Address	B.B.197.0
Description	i-timxceed-intrans
ICMP Code	11+0
Permitted/Denied	p

Keying a search on the source address (210.77.146.1) above gave over 200,000 attacks. From the further information extracted from the firewall logs, some packets were of particular interest. The next section contains a sample of these packets:

Date	Time	Protocol	Source	Port	Destination	Port/Type	Status
07/11/2000	09:32:49	icmp	B.B1.77.197	-	210.77.146.1	echo	permit
07/11/2000	09:33:00	icmp	210.77.146.1	-	B.B1.77.197	echo-reply	permit
07/11/2000	09:34:20	udp	B.B1.77.197	137[1]	210.77.146.1	137	deny
07/11/2000	10:13:49	udp	B.B1.77.197	137	210.77.146.1	137	deny
07/11/2000	10:14:19	icmp	B.B1.77.197	-	210.77.146.1	echo	permit
07/11/2000	10:15:19	icmp	B.B1.77.197	-	210.77.146.1	echo	permit
07/11/2000	10:15:53	icmp	210.77.146.1	-	B.B1.77.197	echo-reply	permit
07/11/2000	10:15:55	udp	B.B1.77.197	137	210.77.146.1	137	deny
08/11/2000	12:06:47	udp	B.B2.41.95	137	210.77.146.1	137	deny
08/11/2000	12:06:52	icmp	B.B2.41.95	-	210.77.146.1	echo	permit
08/11/2000	12:06:54	icmp	210.77.146.1	-	B.B2.41.95	echo-reply	permit
08/11/2000	12:07:53	icmp	B.B2.41.95	-	210.77.146.1	echo	permit

Finally, a search for the addresses mentioned in the above traces revealed that neither had reverse lookups configured, but searching on APNIC [APNIC] revealed the following information:

202.104.139.195 Topearch Printed Circuits (Chinanet - Guangdong province network)
210.77.146.1 A3Dial-Net (Beijing, China – An ISP offering dialup service)

1.1.2 Detect was generated by

The initial detect was generated by ISS RealSecure [ISS], with correlating information from TCPDump [TCPDump] and a corporate firewall log analysis/correlation tool.

1.1.3 Probability the source address was spoofed

It is not clear from the packet whether the external (source) address was spoofed, but either the packet was crafted, or is the result of a crafted packet. The destination address was certainly spoofed, as otherwise the packet seen would have to be in reply to a packet from a machine with the network address as a host IP address. This is, of course, extremely unlikely.

Examination of firewall logs provides no evidence of an outward packet, and confirms this assumption that the packet was indeed crafted to slip through firewall perimeters.

1.1.4 Description of attack

This attack would appear to be an attempt at mapping the network. Further examination of the extended firewall logs show that well over 200,000 probes of the six class B subnets being monitored took place over 20 days between the 6th and 26th November 2000, with each /24 network being probed around 1000 times. The sole anomaly in this data is that there was no traffic recorded between 23:59 on the 13th November and 23:59 on the 14th November.[2]

[1] Port 137 is the NetBIOS name service [NeoPort] [SnorPort].
[2] This was later found to be due to an error in the logging scripts. Firewall logs for 14th November had been deleted through operator error.

However, the mapping incident would seem to be destined to fail, as the definition of an ICMP error packet [RFC0777] such as time-exceeded states that it should not be replied to under any circumstances. Therefore, the purpose of the probe cannot be fully determined.

The data at the end of section §1.1.1 above shows that not all hosts abide by the requirements of the RFC, with at least two hosts attempting to communicate with the supposed sender of the packet. These communications are through both ping requests and NetBIOS name-service requests. The former requests were passed through the firewall, whilst the latter were blocked.

1.1.5 Attack mechanism

There are two possible scenarios for the probe:

1) The attacker crafted the ICMP Time Exceeded packet, and sent it directly to the machine being probed.
2) The attacker crafted an ICMP ping packet and sent it to a known router, with the guarantee that it would expire and generate a time-exceeded packet.

Examination of firewall log data shows that every one of the 200,000 packets destined for x.y.z.0 (or x.y.0.1) addresses came from the same router. If scenario two above is correct then it would be expected that at least a percentage would come from different routers. Therefore, the most likely scenario is the first and hence this is the most probable attack mechanism in use.

Of interest is the behaviour of the network under attack when the packet arrives. The firewall allows the ICMP Time Exceeded packet through as this is a common response to a packet sourced by the 'traceroute' command under UNIX or the 'tracert' command under 32-bit Windows Operating Systems. The packet is routed as normal to the subnet concerned. At this point, if the x.y.z.0 address is indeed a network address the router for the subnet automatically converts the destination of the packet to the broadcast address 255.255.255.255.

The reason for this conversion is that certain systems have a TCP stack which considers that an address in which the host portion of the IP address is 0 is a broadcast address. Cisco routers have this behaviour, and convert the ICMP packet to an ICMP broadcast. Any host that is configured to reply to these packets will then originate a packet which may be used to map the network. An example of such a system may be software that attempts name resolution. Under Windows, this resolution will be attempted through local hosts file lookups, WINS, DNS and finally by a directed NetBIOS call to the machine in question. This is indeed the behaviour observed.

1.1.6 Correlations

As mentioned above, correlation for the attack came from three sources:

1) ISS RealSecure
2) Tcp-Dump analysis
3) Firewall logs

Further correlation came from the firewall logs, which indicated a large number of similar probes.

1.1.7 Evidence of active targeting

There is no evidence of active targeting. This scan would appear from evidence to be a randomly generated scan of all IP addresses of the format A.B.C.0 or A.B.0.1.

1.1.8 Severity

Severity is defined by using the pseudo-equation:
(Criticality + Lethality) – (System Security + Network Countermeasures)

In this case, Severity could be outlined as being 0, or ((4 + 1) – (2 + 3)). Criticality is high at four out of a maximum possible score of five as entire networks have been probed. Lethality, on the other hand, is low at one out of four as there is no known vulnerability which could be invoked with this behaviour.

System Security is also low at two as, although firewalls and IDS systems are in place, the probe travelled straight through the systems. Finally, network countermeasures are mediocre at three for much the same reasons. The probe travelled through the network but what responses were sent were successfully stopped from reaching the pseudo-attacker.

1.1.9 Defensive recommendation

In order to stop this type of faked time-exceeded packet from penetrating the firewalls, the option to make the firewalls stateful should be investigated. This would allow ICMP reply packets to penetrate the firewalls if, and only if, an originating packet from the internal host had been seen. However, such a stateful system would have to examine the contents of packets. This is because, even under normal situations, an incoming time-exceeded packet would not match any outgoing packet if the ICMP headers are compared.

It should also be possible to change the firewalls ruleset so that any packet destined for a typical network address (x.y.z.0) is dropped. This may affect a few systems if they have been subnetted to have more than 254 hosts in a subnet, but this would be a relatively few situations and problems could be easily avoided.

Note also that the various systems in §1.1.1 above reported the alert with different timestamps:

RealSecure: 16:39:08
TCPDump: 16:37:03.573766
Firewall: 16:34:08

The firewalls are already synchronised with a NTP server. It is recommended that any other machine which is used for forensic data collection purposes is also synchronised in this manner.

1.1.10 Multiple choice test question

In what situation would 10.172.211.0 be a valid host IP address

a) It is always a valid IP address
b) If the subnet mask is 255.255.255.128
c) If the subnet is 10.172.210.0/23
d) It is never a valid IP address

Answer: (c)

An address of the format x.y.z.0 is usually a network address, but in some circumstances can be an IP address, so (a) and (d) are not correct. For the subnet with mask 255.255.255.128, the subnet range in question would be 10.172.211.1 to 10.172.211.126, with 10.172.211.0 being the network address and 10.172.211.127 being the broadcast address.

In case (c), the subnet range is from 10.172.210.1 to 10.172.211.254 with 10.172.210.1 being the network address and 10.172.211.255 the broadcast address. Thus 10.172.211.0 is a valid address in this range, addressing the machine with hostid 256 in the subnet.

1.2 Loki Backdoor

1.2.1 Source of Trace

On the 7th July 2001, a Loki attack was revealed by an ISS RealSecure IDS system. The information as supplied by this system was:

Date	07/07/2001
Time	14:18:51
Detect name	Loki
Source Address	B.B1.142.62
Destination Address	B.B2.159.218

```
tcpdump -r 14.15 -nX 'host B.B1.142.62'
14:18:51.507735 B.B1.142.62 > B.B2.159.218: icmp: request

0x0000   4500 02d8 3d17 0000 7b01 924b BBBB 8e3e        E...=...{..KBB.>
0x0010   BBBB 9fda 0800 6a80 0100 f001 a147 1842        BB....j......G.B
0x0020   2c49 3a4d 233f 5f51 337c cd4a b156 1e5d        ,I:M#?_Q3|.J.V.]
0x0030   fc57 9e57 f877 5e29 3e4d 1c2b f71f 3315        .W.W.w^)>M.+..3.
0x0040   877c 266a 620c 9322 c86e 7f0d 4458 6730        .|&jb.."n..DXg0
................
0x02b0   fb2f c00b f172 130f 0042 f05d e94b cf5c        ./...r...B.].K.\
0x02c0   1979 f658 2076 8c17 fc7a 4522 8e7d 7266        .y.X.v...zE".}rf
0x02d0   fb30 7221 c400 df2d cf6d e81e ce24 b943        .0r!...-.m...$.C
0x02e0   473f 78a5                                      G?x.
```

1.2.2 Detect was generated by

The initial detect was generated by ISS RealSecure [ISS], with correlating information from TCPDump [TCPDump].

1.2.3 Probability the source address was spoofed

This packet is very unlikely to have been spoofed as both the source and destination addresses are internal to the corporate network.

1.2.4 Description of attack

Loki is a backdoor trojan horse program that was published in Phrack 51, Article 6 [Phra51§6]. It uses ICMP echo request|reply packets as a carrier for the data payload, which is used to remotely control the infected computer. The ICMP header contains a sequence number (offset 0x001A in

the tcpdump output in section §1.2.1 above), which is set to 0xf001 (or 0x01f0 in little-endian notation) for a Loki attack. It is therefore to be expected that, assuming a regular distribution of sequence numbers, two in every 65,536 ICMP packets will contain one of these sequence numbers, and therefore trigger a Loki false positive.

The event here was accepted as a false positive. The full dump of ICMP packets from TCPDump showed four ping requests and replies, each containing a payload of 700 random bytes. These ping requests had consecutive sequence numbers (0xefe1, 0xeff1, 0xf001, 0xf011), and only one packet contained the 'trigger' sequence number.

In addition, the machine B.B1.142.62 was found to be a network monitoring machine, using large ICMP packets to test network resilience. If the target machine B.B2.159.218 had, indeed, been compromised then a larger number of ICMP packets would have been expected, and they would all have had the same sequence number.

1.2.5 Attack mechanism

There is no attack in this incidence, but the ping probe was determined to be a heartbeat monitor coming from HP Openview [HewlOV], a network monitoring tool.

1.2.6 Correlations

The RealSecure alert was correlated to the TCPDump output. No further correlations are indicated or required.

1.2.7 Evidence of active targeting

There is no evidence of active targeting and this alert has been accepted as a false positive.

1.2.8 Severity

Severity is defined by using the pseudo-equation:
(Criticality + Lethality) – (System Security + Network Countermeasures)

In this case, Severity could be outlined as being -2, or ((4 + 2) – (4 + 4)). Criticality is high as the machine supposedly under attack is a main database server. Lethality has a medium score of two as the machine could, feasibly, have been infected with Loki although this is unlikely.

Security is high as firewalls and IDS systems are in place and ICMP traffic is not allowed out of this secured subnet onto the general network or the internet. Finally, network countermeasures are also high at four for much the same reasons. Even if the machine had been infected there was no risk of the penetration proving fruitful for the hacker.

1.2.9 Defensive recommendation

No direct defensive recommendation can be made to improve protection against this false positive. However, it is recommended that a threshold be set on Loki alerts so that alerts of this type are only fully investigated if more than one event has been seen in an agreed period, for example a day, on a single host. This should reduce the risk of the analyst spending time on spurious investigations.

1.2.10 Multiple choice test question

For what purpose do ICMP packets have a sequence number?

a) It is the ICMP equivalent to port numbers in TCP or UDP
b) It has no relevance at all
c) It is the process id of the calling program
d) It is used to correlate ICMP requests and replies

Answer: (d)

ICMP has no port numbers, and so the sequence number cannot be a port number as claimed in answer (a). However, ICMP does have a type and code reference, indicating whether the packet is a request, reply or error packet etc. These references are located at 0x0014 and 0x0015 respectively in the IP packet.

The process ID of the calling program has a different meaning depending on the Operating System, and indeed has no relevance to an ICMP packet so (c) is just plain wrong. The sequence number is used to correlate an ICMP reply packet to its originating request packet, and so (d) is correct, simultaneously indicating that (b) is not the right answer.

1.3 Scan for web server

1.3.1 Source of Trace

The following source was obtained [Inci0803] from the incidents.org [Incidents] website. Note that all packets were received on August 3rd 2001, between 07:17:00 and 07:17:04 (Timezone unknown), and that dates and times have been removed from the listing below for clarity.

```
Aug  3 07:17:00 hosth snort: WEB-MISC http directory traversal [Classification:
Attempted Information Leak   Priority: 3]: 216.4.30.25:3167 -> a.b.c.62:80

Aug  3 07:17:00 216.4.30.25:3109 -> a.b.c.4:80 SYN ******S*
216.4.30.25:3119 -> a.b.c.14:80 SYN ******S*    216.4.30.25:3195 -> a.b.c.90:80 SYN ******S*
216.4.30.25:3120 -> a.b.c.15:80 SYN ******S*    216.4.30.25:3197 -> a.b.c.92:80 SYN ******S*
216.4.30.25:3122 -> a.b.c.17:80 SYN ******S*    216.4.30.25:3202 -> a.b.c.97:80 SYN ******S*
216.4.30.25:3125 -> a.b.c.20:80 SYN ******S*    216.4.30.25:3204 -> a.b.c.99:80 SYN ******S*
216.4.30.25:3131 -> a.b.c.26:80 SYN ******S*    216.4.30.25:3206 -> a.b.c.101:80 SYN ******S*
216.4.30.25:3132 -> a.b.c.27:80 SYN ******S*    216.4.30.25:3208 -> a.b.c.103:80 SYN ******S*
216.4.30.25:3134 -> a.b.c.29:80 SYN ******S*    216.4.30.25:3214 -> a.b.c.109:80 SYN ******S*
216.4.30.25:3138 -> a.b.c.33:80 SYN ******S*    216.4.30.25:3219 -> a.b.c.114:80 SYN ******S*
216.4.30.25:3141 -> a.b.c.36:80 SYN ******S*    216.4.30.25:3220 -> a.b.c.115:80 SYN ******S*
216.4.30.25:3148 -> a.b.c.43:80 SYN ******S*    216.4.30.25:3221 -> a.b.c.116:80 SYN ******S*
216.4.30.25:3149 -> a.b.c.44:80 SYN ******S*    216.4.30.25:3230 -> a.b.c.125:80 SYN ******S*
216.4.30.25:3151 -> a.b.c.46:80 SYN ******S*    216.4.30.25:3232 -> a.b.c.127:80 SYN ******S*
216.4.30.25:3152 -> a.b.c.47:80 SYN ******S*    216.4.30.25:3235 -> a.b.c.130:80 SYN ******S*
216.4.30.25:3156 -> a.b.c.51:80 SYN ******S*    216.4.30.25:3240 -> a.b.c.135:80 SYN ******S*
216.4.30.25:3167 -> a.b.c.62:80 SYN ******S*    216.4.30.25:3241 -> a.b.c.136:80 SYN ******S*
216.4.30.25:3169 -> a.b.c.64:80 SYN ******S*    216.4.30.25:3247 -> a.b.c.142:80 SYN ******S*
216.4.30.25:3170 -> a.b.c.65:80 SYN ******S*    216.4.30.25:3248 -> a.b.c.143:80 SYN ******S*
216.4.30.25:3174 -> a.b.c.69:80 SYN ******S*    216.4.30.25:3254 -> a.b.c.149:80 SYN ******S*
216.4.30.25:3175 -> a.b.c.70:80 SYN ******S*    216.4.30.25:3256 -> a.b.c.151:80 SYN ******S*
216.4.30.25:3176 -> a.b.c.71:80 SYN ******S*    216.4.30.25:3258 -> a.b.c.153:80 SYN ******S*
216.4.30.25:3181 -> a.b.c.76:80 SYN ******S*    216.4.30.25:3261 -> a.b.c.156:80 SYN ******S*
216.4.30.25:3182 -> a.b.c.77:80 SYN ******S*    216.4.30.25:3262 -> a.b.c.157:80 SYN ******S*
216.4.30.25:3185 -> a.b.c.80:80 SYN ******S*    216.4.30.25:3271 -> a.b.c.166:80 SYN ******S*
216.4.30.25:3190 -> a.b.c.85:80 SYN ******S*    216.4.30.25:3272 -> a.b.c.167:80 SYN ******S*
216.4.30.25:3193 -> a.b.c.88:80 SYN ******S*    216.4.30.25:3273 -> a.b.c.168:80 SYN ******S*
```

```
216.4.30.25:3274 -> a.b.c.169:80 SYN ******S*
216.4.30.25:3276 -> a.b.c.171:80 SYN ******S*
216.4.30.25:3285 -> a.b.c.180:80 SYN ******S*
216.4.30.25:3286 -> a.b.c.181:80 SYN ******S*
216.4.30.25:3287 -> a.b.c.182:80 SYN ******S*
216.4.30.25:3289 -> a.b.c.184:80 SYN ******S*
216.4.30.25:3290 -> a.b.c.185:80 SYN ******S*
216.4.30.25:3293 -> a.b.c.188:80 SYN ******S*
216.4.30.25:3294 -> a.b.c.189:80 SYN ******S*
216.4.30.25:3296 -> a.b.c.191:80 SYN ******S*
216.4.30.25:3299 -> a.b.c.194:80 SYN ******S*
216.4.30.25:3304 -> a.b.c.199:80 SYN ******S*
216.4.30.25:3305 -> a.b.c.200:80 SYN ******S*
216.4.30.25:3310 -> a.b.c.205:80 SYN ******S*
216.4.30.25:3311 -> a.b.c.206:80 SYN ******S*
216.4.30.25:3317 -> a.b.c.212:80 SYN ******S*
216.4.30.25:3329 -> a.b.c.224:80 SYN ******S*
216.4.30.25:3334 -> a.b.c.229:80 SYN ******S*
216.4.30.25:3342 -> a.b.c.237:80 SYN ******S*
216.4.30.25:3350 -> a.b.c.245:80 SYN ******S*
216.4.30.25:3352 -> a.b.c.247:80 SYN ******S*
216.4.30.25:3354 -> a.b.c.249:80 SYN ******S*
216.4.30.25:3355 -> a.b.c.250:80 SYN ******S*
216.4.30.25:3356 -> a.b.c.251:80 SYN ******S*
216.4.30.25:3410 -> a.b.d.51:80 SYN ******S*
216.4.30.25:3411 -> a.b.d.52:80 SYN ******S*
216.4.30.25:3558 -> a.b.d.197:80 SYN ******S*
216.4.30.25:3559 -> a.b.d.198:80 SYN ******S*
216.4.30.25:3560 -> a.b.d.199:80 SYN ******S*
216.4.30.25:3567 -> a.b.d.206:80 SYN ******S*
216.4.30.25:3569 -> a.b.d.208:80 SYN ******S*
216.4.30.25:3574 -> a.b.d.213:80 SYN ******S*
216.4.30.25:3576 -> a.b.d.215:80 SYN ******S*
216.4.30.25:3578 -> a.b.d.217:80 SYN ******S*
216.4.30.25:3581 -> a.b.d.220:80 SYN ******S*
216.4.30.25:3583 -> a.b.d.222:80 SYN ******S*
216.4.30.25:3584 -> a.b.d.223:80 SYN ******S*
216.4.30.25:3586 -> a.b.d.225:80 SYN ******S*
216.4.30.25:3590 -> a.b.d.229:80 SYN ******S*
216.4.30.25:3595 -> a.b.d.234:80 SYN ******S*
216.4.30.25:3598 -> a.b.d.237:80 SYN ******S*
216.4.30.25:3602 -> a.b.d.241:80 SYN ******S*
216.4.30.25:3606 -> a.b.d.245:80 SYN ******S*
216.4.30.25:3609 -> a.b.d.248:80 SYN ******S*
216.4.30.25:3611 -> a.b.d.250:80 SYN ******S*
216.4.30.25:3615 -> a.b.d.254:80 SYN ******S*
216.4.30.25:3620 -> a.b.e.5:80 SYN ******S*
216.4.30.25:3627 -> a.b.e.12:80 SYN ******S*
216.4.30.25:3628 -> a.b.e.13:80 SYN ******S*
216.4.30.25:3629 -> a.b.e.14:80 SYN ******S*
216.4.30.25:3630 -> a.b.e.15:80 SYN ******S*
216.4.30.25:3631 -> a.b.e.16:80 SYN ******S*
216.4.30.25:3632 -> a.b.e.17:80 SYN ******S*
216.4.30.25:3633 -> a.b.e.18:80 SYN ******S*
216.4.30.25:3634 -> a.b.e.19:80 SYN ******S*
216.4.30.25:3635 -> a.b.e.20:80 SYN ******S*
216.4.30.25:3636 -> a.b.e.21:80 SYN ******S*
216.4.30.25:3637 -> a.b.e.22:80 SYN ******S*
216.4.30.25:3640 -> a.b.e.25:80 SYN ******S*
216.4.30.25:3642 -> a.b.e.27:80 SYN ******S*
216.4.30.25:3643 -> a.b.e.28:80 SYN ******S*
216.4.30.25:3645 -> a.b.e.30:80 SYN ******S*
216.4.30.25:3647 -> a.b.e.32:80 SYN ******S*
216.4.30.25:3656 -> a.b.e.41:80 SYN ******S*
216.4.30.25:3663 -> a.b.e.48:80 SYN ******S*
216.4.30.25:3673 -> a.b.e.58:80 SYN ******S*
216.4.30.25:3675 -> a.b.e.60:80 SYN ******S*
216.4.30.25:3678 -> a.b.e.63:80 SYN ******S*
216.4.30.25:3680 -> a.b.e.65:80 SYN ******S*
216.4.30.25:3683 -> a.b.e.68:80 SYN ******S*
216.4.30.25:3688 -> a.b.e.73:80 SYN ******S*
216.4.30.25:3691 -> a.b.e.76:80 SYN ******S*
216.4.30.25:3692 -> a.b.e.77:80 SYN ******S*
216.4.30.25:3693 -> a.b.e.78:80 SYN ******S*
216.4.30.25:3694 -> a.b.e.79:80 SYN ******S*
216.4.30.25:3699 -> a.b.e.84:80 SYN ******S*
216.4.30.25:3700 -> a.b.e.85:80 SYN ******S*
216.4.30.25:3702 -> a.b.e.87:80 SYN ******S*
216.4.30.25:3703 -> a.b.e.88:80 SYN ******S*
216.4.30.25:3710 -> a.b.e.95:80 SYN ******S*
216.4.30.25:3716 -> a.b.e.101:80 SYN ******S*
216.4.30.25:3719 -> a.b.e.104:80 SYN ******S*
216.4.30.25:3720 -> a.b.e.105:80 SYN ******S*
216.4.30.25:3723 -> a.b.e.108:80 SYN ******S*
216.4.30.25:3726 -> a.b.e.111:80 SYN ******S*
216.4.30.25:3727 -> a.b.e.112:80 SYN ******S*
216.4.30.25:3731 -> a.b.e.116:80 SYN ******S*
216.4.30.25:3734 -> a.b.e.119:80 SYN ******S*
216.4.30.25:3740 -> a.b.e.125:80 SYN ******S*
216.4.30.25:3742 -> a.b.e.127:80 SYN ******S*
216.4.30.25:3743 -> a.b.e.128:80 SYN ******S*
216.4.30.25:3748 -> a.b.e.133:80 SYN ******S*
216.4.30.25:3751 -> a.b.e.136:80 SYN ******S*
216.4.30.25:3755 -> a.b.e.140:80 SYN ******S*
216.4.30.25:3775 -> a.b.e.160:80 SYN ******S*
216.4.30.25:3782 -> a.b.e.167:80 SYN ******S*
216.4.30.25:3784 -> a.b.e.169:80 SYN ******S*
216.4.30.25:3786 -> a.b.e.171:80 SYN ******S*
216.4.30.25:3791 -> a.b.e.176:80 SYN ******S*
216.4.30.25:3792 -> a.b.e.177:80 SYN ******S*
216.4.30.25:3794 -> a.b.e.179:80 SYN ******S*
216.4.30.25:3799 -> a.b.e.184:80 SYN ******S*
216.4.30.25:3801 -> a.b.e.186:80 SYN ******S*
216.4.30.25:3802 -> a.b.e.187:80 SYN ******S*
216.4.30.25:3807 -> a.b.e.192:80 SYN ******S*
216.4.30.25:3809 -> a.b.e.194:80 SYN ******S*
216.4.30.25:3815 -> a.b.e.200:80 SYN ******S*
216.4.30.25:3831 -> a.b.e.215:80 SYN ******S*
216.4.30.25:3833 -> a.b.e.217:80 SYN ******S*
216.4.30.25:3840 -> a.b.e.224:80 SYN ******S*
216.4.30.25:3841 -> a.b.e.225:80 SYN ******S*
216.4.30.25:3845 -> a.b.e.229:80 SYN ******S*
216.4.30.25:3848 -> a.b.e.232:80 SYN ******S*
216.4.30.25:3849 -> a.b.e.233:80 SYN ******S*
216.4.30.25:3852 -> a.b.e.236:80 SYN ******S*
216.4.30.25:3857 -> a.b.e.241:80 SYN ******S*
216.4.30.25:3863 -> a.b.e.247:80 SYN ******S*
216.4.30.25:3864 -> a.b.e.248:80 SYN ******S*
216.4.30.25:3872 -> a.b.f.2:80 SYN ******S*
216.4.30.25:3879 -> a.b.f.9:80 SYN ******S*
216.4.30.25:3880 -> a.b.f.10:80 SYN ******S*
216.4.30.25:3884 -> a.b.f.14:80 SYN ******S*
216.4.30.25:3891 -> a.b.f.21:80 SYN ******S*
216.4.30.25:3897 -> a.b.f.27:80 SYN ******S*
216.4.30.25:3898 -> a.b.f.28:80 SYN ******S*
216.4.30.25:3902 -> a.b.f.32:80 SYN ******S*
216.4.30.25:3904 -> a.b.f.34:80 SYN ******S*
216.4.30.25:3906 -> a.b.f.36:80 SYN ******S*
216.4.30.25:3907 -> a.b.f.37:80 SYN ******S*
216.4.30.25:3909 -> a.b.f.39:80 SYN ******S*
216.4.30.25:3911 -> a.b.f.41:80 SYN ******S*
216.4.30.25:3922 -> a.b.f.52:80 SYN ******S*
216.4.30.25:3925 -> a.b.f.55:80 SYN ******S*
216.4.30.25:3927 -> a.b.f.57:80 SYN ******S*
216.4.30.25:3928 -> a.b.f.58:80 SYN ******S*
216.4.30.25:3930 -> a.b.f.60:80 SYN ******S*
216.4.30.25:3937 -> a.b.f.67:80 SYN ******S*
216.4.30.25:3939 -> a.b.f.69:80 SYN ******S*
216.4.30.25:3944 -> a.b.f.74:80 SYN ******S*
216.4.30.25:3949 -> a.b.f.79:80 SYN ******S*
216.4.30.25:3954 -> a.b.f.84:80 SYN ******S*
216.4.30.25:3955 -> a.b.f.85:80 SYN ******S*
```

```
216.4.30.25:3956 -> a.b.f.86:80 SYN ******S*     216.4.30.25:4015 -> a.b.f.145:80 SYN ******S*
216.4.30.25:3957 -> a.b.f.87:80 SYN ******S*     216.4.30.25:4016 -> a.b.f.146:80 SYN ******S*
216.4.30.25:3958 -> a.b.f.88:80 SYN ******S*     216.4.30.25:4019 -> a.b.f.149:80 SYN ******S*
216.4.30.25:3959 -> a.b.f.89:80 SYN ******S*     216.4.30.25:4020 -> a.b.f.150:80 SYN ******S*
216.4.30.25:3961 -> a.b.f.91:80 SYN ******S*     216.4.30.25:4022 -> a.b.f.152:80 SYN ******S*
216.4.30.25:3974 -> a.b.f.104:80 SYN ******S*    216.4.30.25:4025 -> a.b.f.155:80 SYN ******S*
216.4.30.25:3983 -> a.b.f.113:80 SYN ******S*    216.4.30.25:4026 -> a.b.f.156:80 SYN ******S*
216.4.30.25:3986 -> a.b.f.116:80 SYN ******S*    216.4.30.25:4034 -> a.b.f.164:80 SYN ******S*
216.4.30.25:3995 -> a.b.f.125:80 SYN ******S*    216.4.30.25:4038 -> a.b.f.168:80 SYN ******S*
216.4.30.25:4005 -> a.b.f.135:80 SYN ******S*    216.4.30.25:4053 -> a.b.f.183:80 SYN ******S*
216.4.30.25:4006 -> a.b.f.136:80 SYN ******S*    216.4.30.25:4060 -> a.b.f.190:80 SYN ******S*
216.4.30.25:4013 -> a.b.f.143:80 SYN ******S*    216.4.30.25:4115 -> a.b.f.245:80 SYN ******S*
216.4.30.25:4014 -> a.b.f.144:80 SYN ******S*    216.4.30.25:4117 -> a.b.f.247:80 SYN ******S*
```

1.3.2 Detect was generated by

The source was obtained from the incidents.org website [Inci0803], and as such details about the software used to generate the alert is limited. From the format of the log files, it can be safely assumed that the data was generated by Snort [Snort]. As the "WEB-MISC http directory traversal" was not an alert present in the standard Snort v1.7 release, it is likely that the system was running version 1.8 or above. Snort version 1.8.1 was not publicly released until the 13th August 2001, so it is most probable that the alerts came from Snort version 1.8.

The actual alert that generated the dump was generated by the "WEB-MISC http directory traversal" trigger. This is given here.

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC http directory
   traversal"; flags: A+; content: "..\\";reference:arachnids,298; classtype:
   attempted-recon; sid:1112; rev:1;)
```

The rule is activated if any external network, as defined by Snort, tries to connect to one of a given list of webservers with a request including the pattern "..\" (the second backslash is part of the format of the request). This technique is used by web-hackers to try to bypass the filesystem security of a webserver.

1.3.3 Probability the source address was spoofed

The source address of this scan (216.4.30.25) is given in the incidents.org mailing as:

Business Internet, Inc. (NET-ICIX-MD-BLK17)
3625 Queen Palm Drive Tampa, FL 33619 US
Netname: ICIX-MD-BLK17
Netblock: 216.0.0.0 - 216.5.255.255
Maintainer: IMBI

This information has been confirmed with reference to the ARIN whois database [ARIN]. The address itself is not resolvable to a hostname, but is accessible through http://216.4.30.25/ and produces a defamatory message (as at 23/07/2001). This suggests that the machine has been compromised, which would indicate that the source address is not spoofed.

1.3.4 Description of attack

This sequence of alerts has every hallmark of a simple probe for web servers in order to carry out some nefarious activity. The packets are all valid TCP packets, with correct flags and increasing port numbers. Across the span of the report there are 219 scans covering four class C subnets. The port numbers increase with the IP addresses being scanned (254 addresses in a class C excluding a.b.c.0 and a.b.c.255, the network and broadcast addresses respectively) there only being three extra port numbers introduced into the sequence.

An alert, "WEB-MISC http directory traversal", was generated by at least one packet. This is interesting as there is no evidence of data in the packets listed, but the alert (as described in §1.3.2 above) would only alert if the string "..\" appeared in the packet. This would indicate that the attacked machine, a.b.c.62, was indeed a webserver and some further communication took place in response to the initial protocol probe.

1.3.5 Attack mechanism

The extremely short timescale over which the probe took place (219 reported probes in under four seconds) indicate that the probe was scripted. This is also borne out by the sequential nature of the originating port numbers. It was noted that two gaps appeared in the port numbers (after a.b.d.52 and after a.b.e.200). This would indicate that the port numbers were not forged, and that the software concerned automatically generated the packet numbers, carrying out two or three other operations during the scan.

Note that there are only 219 probes across the four class C networks, a possible 1016 addresses (assuming normal subnet masks). The port numbers would seem to indicate that all of the possible addresses were probed, so it is the assumption that either the other 797 addresses were not scanned, or that the IDS software is missing a significant number of packets. Investigations into the cause of these 'missing' packets should be carried out as a matter of urgency.

1.3.6 Correlations

There are no direct correlations for a probe against these networks. Correlation that could be used includes firewall logs and further analysis of the Snort and webserver logs. These logs are not available to the analyst.

1.3.7 Evidence of active targeting

The initial evidence indicates that this is a general scan of the internet for webservers, possibly as a pre-attack probe. However, analysis of the author's own networks provided no correlating attack probes. This would indicate that the probe took place against a small section of the IP address space, possibly targeting this particular section. Therefore, further analysis to ensure the security of servers is recommended.

A rule of the format given above in §1.3.2 does have false positives. If a website designer uses relative URLs in their pages (as is often recommended) this will cause the rule to generate an alert when normal usage is present. Awareness of these quirks of the rulesets is important when analysing alerts of this type.

1.3.8 Severity

Severity is defined by using the pseudo-equation:
(Criticality + Lethality) – (System Security + Network Countermeasures)

In this case, Severity could be given as being 2, or ((4 + 3) – (3 + 2)). Criticality is relatively high as the entire network is being probed, which will almost certainly find external webservers. The fact that the probing machine has been compromised also increases this value. Lethality has a medium score of three out of five as although the probe is not lethal in itself it does act as a warning that further attacks are likely against these servers.

Security is medium as, although IDS and logging systems are in place, there is no knowledge of firewall status or the security levels on the monitored machines. Finally, network counter-measures are given a low figure of two for much the same reasons. On a known network, the severity figure is likely to be lower but the unknown status of the systems increases the risk significantly.

1.3.9 Defensive recommendation

Nothing can stop an attacker from probing systems prior to launching an attack. However, measures can be taken to stop such an attack from being successful. These measures include examining firewall rulesets to ensure that traffic to port 80 (http) is only allowed to 'real' webservers. This simple change would ensure that other machines are not compromised by 'mistake', i.e. if webserving software has been installed or enabled when this is against corporate policy. Protection can then be targeted against the known webservers and effort not wasted.

1.3.10 Multiple choice test question

What software can use port 80 to communicate

a) BackOrifice
b) Netscape Communicator
c) Microsoft Telnet
d) All of the above

Answer: (d)

The fact that port 80 is normally associated with web (http) traffic does not mean that only http traffic can travel across port 80. If a telnet daemon has been configured to listen on port 80 then this can be used to bypass firewalls, a fact that is often used by backdoor software such as BackOrifice. Therefore, the answer to the question is (d), and indeed the answer would have been (d) for almost any given list of software packages.

1.4 Email Relay Attempt

1.4.1 Source of Trace

On the 16th April 2001, an Email_Relay_Spam attack was revealed by an ISS RealSecure IDS system. The information as supplied by this system was:

Date	16/04/2001
Time	04:15:00
Detect name	Email_Relay_Spam
Auxiliary Information	username%dataforce.co.uk@mailhost.mydomain.com
Source Address	B.B1.159.220
Destination Address	B.B2.207.111

Packet logs from TCPDump revealed the following smtp communication.

B.B1.159.220:37578-B.B2.207.111:25[3]
HELO host.mydomain.com
B.B2.207.111:25- B.B1.159.220:37578
250 mailhost.mydomain.com Hello host.mydomain.com ([B.B1.159.220]), pleased to meet you
B.B1.159.220:37578-B.B2.207.111:25
MAIL From:<username@host.mydomain.com>
B.B2.207.111:25- B.B1.159.220:37578
250 <username@host.mydomain.com>... Sender ok
B.B1.159.220:37578-B.B2.207.111:25
RCPT To:< username%dataforce.co.uk@mailhost.mydomain.com>
B.B2.207.111:25- B.B1.159.220:37578
250 <username%dataforce.co.uk@mailhost.mydomain.com>... Recipient ok

Further investigation revealed that the same alert had been occurring at, or around, the same time for twelve months. Alert was revealed by a change in IDS policy.

1.4.2 Detect was generated by

The initial detect was generated by ISS RealSecure [ISS], with correlating information from TCPDump [TCPDump].

1.4.3 Probability the source address was spoofed

This packet is very unlikely to have been spoofed as both the source and destination addresses are internal to the corporate network.

[3] Port 25 is the standard smtp (simple mail transport protocol) port number [NeoPort]

1.4.4 Description of attack

When electronic mail was first being used heavily there was no guarantee that a two computers had a direct connection in order to send mail between them. Therefore, a mechanism was introduced whereby a connection could be made via a third computer which had connectivity to both of the interested parties.

In practice, Adam (on HostA) wanted to send Eve (on HostB) a message at her mail address 'Eve@hostb'. He would send the message to HostC, telling it to forward the message on to Eve. The nomenclature used for this is 'Eve%hostb@hostc'. The mail program on HostC would strip out its own name from the address, recognise that the destination is not local, and forward on, replacing the last '%' sign with an '@' symbol. There could theoretically be any number of these redirection stages in order to correctly transmit electronic mail.

This procedure has been depreciated in recent years (at least since the mid-1990's) when it has become safer to assume that all mail servers can communicate directly, and is now most often used when the sender of an e-mail wants to hide the origin of his message. Therefore, this is a good reason to examine alerts of this type for validity as the hosts concerned may be used for spamming purposes.

In this particular case, there was an automated script on a host which was designed to contact a courier company. However, the host on the internal network does not have permission to directly contact the internet, and therefore needs to use mail redirection in order to send its automated messages.

1.4.5 Attack mechanism

In this particular case, the use of mail redirection is perfectly correct, and indeed is the use for which this style of operation was originally designed. Therefore, there is no attack to be analysed. However, the mail message was determined to be from an automated application located on an internal server.

1.4.6 Correlations

The RealSecure alert was correlated to the TCPDump output. No further correlations are indicated or required.

1.4.7 Evidence of active targeting

There is no evidence of active targeting and this alert has been accepted as a false positive.

1.4.8 Severity

Severity is defined by using the pseudo-equation:
(Criticality + Lethality) – (System Security + Network Countermeasures)

In this case, Severity could be outlined as being -3, or ((3 + 2) – (4 + 4)). Criticality is medium as although the mail server is vulnerable to these attacks, it is only vulnerable to internal hosts. Lethality has a low score as there is no direct harm that can come from this style of attack.

Security is high as firewalls and IDS systems are in place. Finally, network countermeasures are also high at four for much the same reasons.

However, note that should this attack ever become reality, the effect would be in contrast to the severity listed as it may well affect company reputation. This shows the limitations of pseudo-equations such as the one used here as the sole analysis of severity.

1.4.9 Defensive recommendation

It is recommended that this particular alert is accepted as a false positive, but that any other alerts of this type are investigated thoroughly.

1.4.10 Multiple choice test question

Which of the statements below is not a valid e-mail address

a) Joe.Bloggs@mailhost.yourcompany.co.za
b) Joe.Bloggs@mailhost.yourcompany.co.za@mailhost.mycompany.com
c) mypc!mailhost!mailhost.yourcompany.co.za!bloggsj
d) Joe.Bloggs%host.yourcompany.co.za@host.mycompany.com

Answer: (b)

The normal form of mail address is given here in section (a), whilst (d) is the original format of mail redirection where the originating machine does not have direct connection to the recipient machine and an intermediary is required. The UUCP standard for mail transmission [RFC0976] is shown here in (c)

1.5 Duplicate IP Addresses

1.5.1 Source of Trace

During a four month period, the ISS RealSecure NetworkSensor on a corporate network produced alerts similar to the one here:

Date	03/04/2001
Time	11:17:26
Detect name	IPDuplicate
Source Address	B.B.46.99
Destination Address	B.B.145.52
Additional Info	MAC1:AA:0:4:0:26:B4; MAC2:AA:0:4:0:25:B4

There were a very large number of events, with the top four source addresses given here:

Source	Destination	MAC Address 1	MAC Address 2	Count
B.B.128.34	B.B.145.52	aa:0:4:0:26:b4	aa:0:4:0:25:b4	1422
B.B.128.35	B.B.145.52	aa:0:4:0:26:b4	aa:0:4:0:25:b4	5435
B.B.159.77	B.B.145.52	aa:0:4:0:26:b4	aa:0:4:0:25:b4	111
B.B.159.200	B.B.145.52	aa:0:4:0:26:b4	aa:0:4:0:25:b4	21

1.5.2 Detect was generated by

The initial detects were generated by ISS RealSecure [ISS].

1.5.3 Probability the source address was spoofed

This packet is very unlikely to have been spoofed as both the source and destination addresses are internal to the corporate network.

1.5.4 Description of attack

These alerts are caused by two MAC addresses sending data on the local network with a single IP address. This attack can be used as an insertion technique to confuse servers and/or clients into malicious operations. A common false positive for this event is load-balancing routers which split traffic evenly, and add their own MAC address to the IP packet as they retransmit.

The cause of the IPDuplicate events was found to be the destination machine, B.B.145.52. This machine was sending out incorrect ARP requests for IP addresses that are not on the current subnet. The two routers concerned were configured with Proxy ARP enabled which led to them replying with their own hardware addresses (or at least the DEC-NET pseudo addresses relevant to the subnet) and the requested destination address. As two machines replied with the same IP address, but different hardware addresses, an IPDuplicate event was raised.

This is related to, but not directly similar to, the false positive as outlined above. A host which wants to convert an IP address on its own network to a MAC address for insertion into the IP address header sends an ARP (Address Resolution Protocol) request. If the destination host is not on the local network then the host is expected to use the address of a suitable (or default) router.

1.5.5 Attack mechanism

The attack in this situation was found to be a server with an incorrect subnet mask. The network in question was a /24 network, therefore with subnet mask 255.255.255.0. This means that hosts on this subnet should only send ARP packets to machines with IP addresses that start B.B.145. The host B.B.145.52 was configured incorrectly with a subnet mask of 255.255.0.0 and therefore did not have the correct understanding of what comprised a local host.

Under normal circumstances, the ARP request for a non-local host should have failed at which time the host would default to sending to a local router. However, the two routers were configured with ARP Proxy enabled, allowing them to reply to non-local addresses. Whilst this is not correct as per the corporate network design policy it is not an attack and so can be safely accepted as a false positive.

Note that the alert as given above has the source and destination addresses the ‘wrong’ way around compared to what intuition would imply. This is because ISS RealSecure alerts on ARP replies, not ARP requests. Therefore, in the first example, the ARP request is sent from B.B.145.52 to B.B.46.99. B.B.145.1 and B.B.145.2 would both reply with ARP responses pretending to be B.B.46.99, which caused ISS RealSecure to raise an alert. B.B.145.1 and B.B.145.2 are the two load-balanced routers on the edge of the B.B.145 network.

1.5.6 Correlations

This alert was self-correlating due to the number and extensive nature of the alerts. However, it was also correlated with communications between the analyst and the host owner.

1.5.7 Evidence of active targeting

There is no evidence of active targeting and this alert has been accepted as a false positive.

1.5.8 Severity

Severity is defined by using the pseudo-equation:
(Criticality + Lethality) – (System Security + Network Countermeasures)

In this case, Severity could be outlined as being -2, or ((2 + 2) – (4 + 2)). Criticality and Lethality are both low at 2, as although there is no attack any network mis-configuration can lead to system downtime and unexpected problems.

Security is high at 4 as IDS systems and data analysis are in place, whilst Network Counter-measures are low at 2 as the problems should have been discovered before IDS systems were installed, and raises the issue of other, more serious, network problems yet to be discovered.

1.5.9 Defensive recommendation

Investigations with the system manager confirmed the mis-configuration as evaluated above. The server was corrected and the alert generation was successfully removed. The solution was simple, and much more satisfying to the analyst than simply accepting the false positive without further investigation. It is recommended that comparable alerts are followed up in a similar manner to a satisfactory conclusion.

1.5.10 Multiple choice test question

The current host is B.B.1.5, sitting on a subnet with network address B.B.1.0/27. An ARP request to which host is most likely to be seen by a network analyser?

a) B.B.1.1
b) B.B.1.5
c) B.B.1.51
d) B.B.51.1

Answer: (a)

The host is on subnet B.B.1.0/27, which contains addresses B.B.1.1 to B.B.1.31 inclusive. Therefore, the addresses in (c) and (d) are outside the current subnet, and should not be the subject of normal ARP requests. A host should not ARP for itself, removing option (b) from the running and so B.B.1.1 (option (a)) is the correct answer.

Note that routers are often given the first address in a subnet range (B.B.1.1 in this example) and if this is the case, then all packets to non-local addresses could be preceded by ARP requests to this address. Whilst this results in the correct answer, it is reached by incorrect assumptions and therefore should be appreciated.

2 Assignment 2 – The State of Intrusion Detection

2.1 What are False Positives anyway?

Anyone who has been working in the field of computer security for more than a few days will have heard more experienced colleagues cursing Intrusion Detection software for not being more accurate and for producing too many 'false positives'.

This paper aims to give a detailed background to the issue of false positives by outlining the background to the phenomenon, in the process revisiting some basic statistics that owe more to Psychology than Computer Science.

It will then address the reasons that false positives occur and why software manufacturers allow them into their software. Finally, the paper will make some recommendations as to techniques that can be used to minimise false positives, whilst outlining some of the more common pitfalls.

There are a number of papers on the Internet covering false positives, most notably on the SANS website [Deba2000] and the presentation by Klaus Julisch [Juli2000] which are good overviews of the problem. There are also product specific papers such as the one for Cisco Secure [Cisc2001], but these do not go into details about false positives.

2.2 Background

In statistics, any given hypothesis can be tested and represented by a simple matrix. In this matrix (shown right), the top line represents the hypothesis, and the left hand side represents which hypothesis is accepted after due testing and investigation. Obviously, the desired solution is that H_0 is accepted if H_0 is true and similarly that H_0 is rejected if H_0 is false. If the H_0 hypothesis is rejected when it is actually correct then a Type I error has been made. Similarly, if the H_0 hypothesis has been accepted when it is not correct a Type II error has been made. This terminology is often used in psychological and econometric statistics to test the division of a population into two discrete sections [Hays1981§7].

		Hypothesis	
		H_0	not H_0
Accept Hypothesis?	Yes	✓	✗
	No	✗	✓

In order to map this behaviour to Intrusion Detection techniques, it is first necessary to construct a suitable hypothesis that needs to be tested. An obvious hypothesis is whether the system being evaluated is actually under attack or not. There is a clean split between the possible situations, with the system either being under attack or not being under attack. The acceptance of the hypothesis depends upon the results generated by the Intrusion Detection System in operation.

In this situation, the Type I error occurs when an attack is occurring but the Intrusion Detection System (IDS) has not alerted, whilst a Type II error occurs when the IDS alerts when no attack is in progress. It is generally accepted that accepting the hypothesis is considered a positive result, whilst rejecting the hypothesis is a negative result.

		Attack Happening?	
		Yes	No
Event Detected?	Yes	**OK**	**False Positive (Type II)**
	No	**False Negative (Type I)**	**OK**

If the IDS alerts, it is considered a positive result. The situation where the system is not under attack but such a positive result is observed is classified as a 'False Positive'. Using a similar argument, the situation where the system is under attack but the IDS does not alert is classified as a 'False Negative'.

The diagram to the left has the same structure as the standard Hypothesis table shown above. It shows graphically the link between 'False Positives' and 'False Negatives'.

2.3 Why They Occur

As outlined above, false positives are a natural extension of trying to split a community (in our case, packets in a network stream) into two distinct halves. They are part of the difference between the two observed halves (alerts and non-alerts) and the two real halves (attacks and non-attacks). However, false positives are only a part of the story.

Where false positives occur, false negatives can also arise. These can, in a security field, be much more dangerous. A false positive can get the analyst out of bed needlessly. A false negative could get her sacked! It is therefore the attitude of most network security experts, and accordingly of most software companies, that the false negative rate of IDS software must be kept as low as possible. There are many reasons why this is done and is mainly due to the desire to improve software packages to be 'best of breed'.

However, there is also the understanding that bad publicity could destroy the reputation of a network security company almost overnight. This could be attached to a break-in taking place even though a properly configured and operating security system was in place An example of this could be the recent discussions on personal firewalls that seriously (if temporarily) affected the reputations of a number of products [Gibs2001, SecP2001].

This minimisation of the false negative rate does imply that the system will tend to take a rather loose view on the definition of 'an attack' and cry wolf at the slightest sight of brown fur. This will obviously increase the number of positive reports but, whilst including almost all real attacks in the reports, will also indicate that the number of false positives will increase.

2.4 Solving the Problem

In real life a large proportion of all alerts generated by an IDS will be false positives. Some analysts put this figure as high as 99%, some go even further. Somewhere in the mass of alerts generated is the real attacker, the problem comes in trying to differentiate between the false positives and the real attacks.

2.4.1 Risky Business

The most important decision that needs to be made is what is the corporate attitude to risk where computers are concerned. This information should be in the corporate security policy, and needs to outline the direction of investigations. Put simply, is the company interested whenever it is

under attack, or just when it is an attack that is likely to be successful. The workload of the security analyst can increase tenfold or even more due to the number of alerts arriving at the detection station if the corporate attitude falls into the former camp. The latter attitude is more realistic, but can be seen by some as being fatalistic.

The difference between the two attitudes is epitomised by the placement of the main IDS machines. If these are on the inside of the firewall, between it and the main routers/switches, then the corporate attitude is towards 'real' risk. If the main IDS machines are placed outside the firewall, then the corporate attitude will lean towards 'perceived' risk. A number of corporations have IDSs in both positions, but there will usually be one which is actively monitored and one that is for logs and information. The monitored IDS will be the 'main' IDS for the purposes of the above scenario.

2.4.2 Solutions

Solving this problem is a good percentage of the work undertaken by the Security Analyst, and may take up the vast majority of her working day. A good first step is to reduce the number of alerts seen. This may mean modifying the rulebase operating on the IDS. For example, do you really need to monitor for SQL buffer overloads if you only run Oracle, or for IIS alerts if you run Apache? Reducing the number of active alerts will reduce the false positives; any alert that falls into these categories will, by definition, be a false positive. This general rule can be extended to considering the location of the IDS. However, this style of fine tuning is only a small part of the modifications that can be made to the rulebase.

Another good technique is to examine the collated logs for a month or two and see which events have the highest count. These events will be the ones that are most likely to be obscuring the situation, but this is not certain so all events must be treated with the same caution. However, confirming a false positive that is producing a large number of alerts will be more beneficial than doing the same for an event that only occurs once or twice in the detection period.

Other likely areas for investigation could be events caused by, or destined for, a common host. Such hosts could include network monitoring hosts, web servers and workstations used by network managers or system operators. Web servers are, of course, the targets of a large percentage of all attacks on systems. Because of this they are some of the most heavily monitored hosts, and therefore also the cause of a good number of false positives. The skill is in working out the difference between the false positives and the real hacks.

2.4.3 Into the Pit

Once the number of false positives has been reduced, then the real hacks should become more obvious. However, it must be emphasised that any automated acceptance of false positives should be revisited on a regular basis. This will be necessary if server addresses or functions changes, or if the systems being monitored undergo a software revision.

The lack of a regular revision of alert signatures and 'accepted' false positives can be very dangerous. Just because the last fifteen 'Loki' (see §1.2 above) alerts have been false positives, does not mean that the next one is not the real thing. Signatures should be updated on a regular basis, as should the analyst's understanding of signatures. During the recent CodeRed [EEyeCR],

[EEyeCRII] incident, a number of analysts complained that signatures were not available for the worm for their own IDS system. Many had not linked the alerts reporting problems with the IIS Index Server with the CodeRed worm. .

This means that the automatic acceptance of false positives, either by disabling the relevant signature from the IDS engine, or by more esoteric methods at the management level, must be treated with extreme care. It is to be recommended that all such decisions are evaluated on a regular basis, with the review period being directly related to the calculated risk of a real attack being automatically deleted. A review period of six to twelve months should be the absolute maximum in use, the exact timescale being dependant on local factors.

2.5 Conclusions

Any system that tries to divide the population into two halves (e.g. Male/Female, Left Handed/ Right Handed, Attack/Safe etc) will have errors involved in the process of division. Therefore, part of the process in designing the procedures used in the dividing should deal with how to cope with these errors.

An IDS system by its very nature creates errors of these types. The successful management of these alerts leads to a successful Intrusion Detection System, whilst failure to control false positives results in an unmanageable solution.

However, the management of false positives is a continuous process and must be regularly reviewed in order to keep the systems up to date and relevant. This can be due to new and updated signatures from the IDS vendor (or indeed from the user), but existing rulesets should be evaluated on a regular basis.

Once these basic rules have been followed, many months and years of successful system protection should follow. Happy hunting![4,5]

[4] After this article was finished, an extremely good document was published by Chris Klaus of ISS Atlanta on the focus-ids@securityfocus.com [SecuFocu] mailing list. This broke down the problem into False Positives and False Alerts and is available for reference at:
http://www.securityfocus.com/templates/archive.pike?fromthread=0&list=96&threads=0&mid=212463&end=2001-09-08&start=2001-09-02&

[5] Another excellent article by Kevin Timm was published on the Security Focus website which focuses on methods designed to reduce false positives. http://www.securityfocus.com/focus/ids/articles/falsealarm1.html

3 Assignment 3 – "Analyse This" Scenario

The following network analysis for Wearside University, England[6], was carried out during the months of June and July 2001. Initially, a management summary is outlined and a list of our recommendations for further defensive measures is given. Summary analyses are given for the full period of analysis but a more detailed analysis is given for the week of the 9th July 2001. Following the specification agreed at our meeting of the 22nd June 2001, the analyses will include a breakdown of attacks by alert type, information such as the most aggressive attackers and detailed information about the top talkers. Finally, as requested by your Network Manager, a detailed breakdown of Out-of-Spec (OOS) packets is also included.

3.1 Management Summary

The network is under attack. This is a simplified statement – every network on the internet is under attack. However, the evidence that has been outlined above gives the impression that there is very little protection on this network against trojan horses, backdoor applications and misuse of the systems provided for student and staff academic use.

The majority of alerts were caused by traffic passing through the university network, not actually destined for the network itself (§3.6.7, §3.6.17 and §3.6.19 above). Of the traffic actually concerning the network, most was concerned with external systems scanning the internal network (§3.6.14, §3.6.22, §0, §3.6.4, §3.6.11, §3.6.12 and §3.6.16 above) for either standard protocols, trojan horses or other backdoors.

There is, however, a serious concern with file sharing applications such as gnutella [Gnutella] and kazaa [Kazaa]. There are a number of connections to and from the ports concerned with these applications (6346 and 1214 respectively). Due to the number of connections used by these applications it must be recommended that an official decision be made about the use of the products on the campus network.

3.1.1 Top Talkers – Source Addresses

The source IP addresses which caused the most alerts are given here. Criteria for inclusion in this list is that the host was the source address which generated at least 200 alerts. Full details about the IP addresses (where available) is given in Appendix B.

IP Address	Count	Reference
130.160.4.60	40,520	§3.6.19
24.189.216.251	5,527	§3.6.14
216.150.152.145	4,015	§3.6.14
134.129.71.56	3,738	§3.6.19
134.129.125.158	960	§3.6.19
212.179.15.28	740	§3.6.20
212.179.31.180	512	§3.6.20

[6] Assumptions about academic practices made in this document will be based on the authors experience of UK Academic institutions.

133.25.193.54	504	§3.6.18
211.217.77.163	446	§3.6.4
210.223.52.151	398	§3.6.4
61.10.19.164	394	§3.6.4
203.75.48.252	347	§3.6.3
144.111.85.130	234	§3.6.17
172.143.247.100	224	§3.6.19
64.158.160.82	208	§3.6.3

3.1.2 Top Talkers – Destination Addresses

The destination IP addresses which caused the most alerts are given here. Criteria for inclusion in this list is that the host was the destination address which generated at least 100 alerts. Full details about the IP addresses (where available) is given in Appendix B.

IP Address	Count	Reference
225.130.160.2	36,387	§3.6.19
MY.NET.5.45	7,398	§3.6.14
233.24.119.155	4,816	§3.6.19
225.130.160.3	4,329	§3.6.19
MY.NET.5.44	2,150	§3.6.14
MY.NET.218.90	740	§3.6.20
MY.NET.218.214	698	§3.6.18, §3.6.20
MY.NET.98.178	512	§3.6.20
MY.NET.150.143	202	§3.6.20
MY.NET.70.97	189	§3.6.9, §3.6.20
MY.NET.150.225	165	§3.6.20
MY.NET.100.37	155	§3.6.21
130.132.143.42	142	§3.6.19
130.132.143.43	133	§3.6.19
MY.NET.253.43	129	§3.6.12, §3.6.21
MY.NET.253.41	121	§3.6.10, §3.6.12, §3.6.21
MY.NET.253.42	116	§3.6.5, §3.6.12, §3.6.21

3.1.3 Infected machines

There is little evidence of extensive infection of internal machines, however there are a few highlights. MY.NET.60.11 is almost certainly infected with SubSeven, and there is evidence that the hosts MY.NET.6.53, MY.NET.60.16, MY.NET.75.82, MY.NET.98.121, MY.NET.98.142, MY.NET.98.201, MY.NET.100.87, MY.NET.110.69, MY.NET.110.157, MY.NET.111.155, MY.NET.146.26, MY.NET.153.175, MY.NET.232.69 and MY.NET.253.115 may also be infected with this trojan horse.

There is also considerable evidence that MY.NET.218.214 has been compromised with gnutella [Gnutella], a file sharing application (§3.6.5, §3.6.22).

3.2 Defensive Recommendations

A review of firewall provision is strongly recommended. This should bar all incoming connections except those that are expressly allowed, for example to publicly facing web, ftp, telnet servers etc. Connections which originate internally should be allowed out of the network as this will allow academic usage to be kept as free as possible whilst limiting risk.

There will inevitably be some negative feedback to this recommendation, probably on the grounds of academic necessity or freedom of information. Therefore, a formal method for the authorisation of internal servers should be implemented with a regular review date at six or twelve monthly intervals. These servers should be tied down to IP address and port number(s) and the process should be made difficult, but not impossible, probably requiring head of department or dean of faculty authorisation. This should deter the casual service provider from putting a server up 'just for a week', and should stop undergraduate student provisioning completely.

The final recommendation is for a review of the current Acceptable Use Policy and Security Policy with regard to Computer Use. Recent developments in computer use, especially in the sharing of files between users, mean that most previous AUPs and SecPols are now obsolete, as they usually assume that a user requires some ability or authorisation before installing a server application. This is no longer the case, and the various security documents should be updated to reflect these changes. Such a review should not be a one-off exercise, with constant reviews being agreed on at least an annual basis.

3.3 Alert Collection

In order to collect the alerts a Snort [Snort] Network IDS sensor was added into the network by your Network Manager. Summaries of the output from this sensor were supplied to be analysed as agreed. The rulebase used was kept unchanged from the version available from the Snort website [SnorRule].

3.4 Files Analysed

All of the files listed below were downloaded on 1st August 2001.

File Name	Date	Time	Orig. Size	Raw Size
alert.010709.gz	10-Jul-2001	00:05	108k	1,270,921
alert.010710.gz	11-Jul-2001	00:11	347k	5,963,728
alert.010711.gz	12-Jul-2001	00:05	96k	1,002,341
alert.010712.gz	13-Jul-2001	00:05	87k	980,160
alert.010713.gz	14-Jul-2001	00:05	89k	1,092,763
alert.010714.gz	15-Jul-2001	00:04	80k	923,747
alert.010715.gz	16-Jul-2001	00:05	106k	1,237,692
scans.010709.gz	10-Jul-2001	00:13	215k	2,869,901
scans.010710.gz	11-Jul-2001	00:12	185k	2,252,909
scans.010711.gz	12-Jul-2001	00:14	313k	3,359,813
scans.010712.gz	13-Jul-2001	00:16	291k	3,535,915
scans.010713.gz	14-Jul-2001	00:15	243k	3,269,349
scans.010714.gz	15-Jul-2001	00:12	320k	3,588,330
scans.010715.gz	16-Jul-2001	00:14	251k	2,959,219

The OOS files analysed were the 61 files which contained traffic between the 1st June 2001 and the 31st July 2001 inclusive. These files were also downloaded on the 1st August.

3.5 Analysis Technique

Analyses were split into three concurrent sections. Analysis of the Out-of-Spec files was carried out using Microsoft Excel, as were the logs of scanning attempts. The actual alert files were analysed using SnortSnarf version 010821.1 [SnorSnar] as well as Microsoft Excel. For reference purposes, the analysis was carried out on a PC running Windows 2000 Perl v5.6.1 [Perl561]. The PC was configured with an AMD Thunderbird 1.2GHz processor and 512Mb memory. Most of the daily analyses were carried out in under ninety minutes, but the traffic for one particular day, 14th July 2001, took over two hundred hours of analysis before the process was stopped. Particular attention was paid to this file, with details below in §3.5.1. It was also discovered that the number of source and destination addresses allocated to each alert was incorrect. The numbers quoted in §3.6 below are extracted from Microsoft Excel.

3.5.1 Logs from 14th July 2001

The reason for the problems analysing the logs from 14th July 2001 is that a very large number of alerts were received from address 0.0.0.0, and these caused snortsnarf problems when detailing source addresses. However, it was appreciated that the logs causing problems were evaluation logs and therefore were not used in the final analysis below.

3.5.2 Comments on Log Files

There were a few quirks in the log files. Two files were available for both the scans and alerts for most of the dates. After consultation with your network manager, it was agreed that the 'B' files were development files which could be ignored for the purposes of this evaluation.

There are no indications in the log files as to the timezone used. As Wearside University is in GMT (UTC) it has been assumed throughout the analysis below that the logs are to be read as if they are in GMT and not BST[7]. If this is not the case, or for some reason the logs are in some other timezone (i.e. PST[8], EDT[9] etc) then some of the assumptions about working hours or operation may be incorrect.

3.6 Alert Analysis

The seven days between the 9th and 15th July 2001 inclusive produced 63,296 alerts. Over 70% of these were classified as "UDP SRC and DST outside network" (§3.6.19) with 58% of the remainder being "SMB Name Wildcard" (§3.6.14). A complete list of alerts is given here sorted by the total number of alerts seen. The final two columns give the number of unique source and destination addresses observed.

Signature Name	Number of Alerts 09/07 – 15/07/2001		Src Addr	Dest Addr
UDP SRC and DST outside network	46,157	72.92%	1	1
SMB Name Wildcard	10,039	15.86%	1	1
Watchlist 000220 IL-ISDNNET-990517	1,981	3.13%	6	539
External RPC call	1,357	2.14%	7	831
connect to 515 from outside	826	1.30%	17	19
Tiny Fragments - Possible Hostile Activity	677	1.07%	12	6
WinGate 1080 Attempt	528	0.83%	5	5
Watchlist 000222 NET-NCFC	385	0.61%	21	15
Possible trojan server activity	383	0.61%	30	16
TCP SRC and DST outside network	277	0.44%	24	24
Queso fingerprint	233	0.37%	123	156
SUNRPC highport access!	150	0.24%	27	30
Port 55850 tcp - Possible myserver activity - ref. 010313-1	112	0.18%	1	2
High port 65535 tcp - possible Red Worm - traffic	48	0.08%	239	180
NMAP TCP ping!	46	0.07%	1	1
Attempted Sun RPC high port access	32	0.05%	8	5
Null scan!	32	0.05%	25	92
High port 65535 udp - possible Red Worm - traffic	16	0.03%	2	1
ICMP SRC and DST outside network	7	0.01%	49	101
site exec - Possible wu-ftpd exploit - GIAC000623	7	0.01%	97	28
connect to 515 from inside	2	0.00%	12	8
STATDX UDP attack	1	0.00%	79	199

[7] BST – British Summer Time (GMT0BST), runs from the last Sunday in March to the last Sunday in October.
[8] PST – Pacific Standard Time (PST-8).
[9] EDT – Eastern Daylight Time (EST-5EDT), runs from the first Sunday in April to the last Sunday in October.

3.6.1 Attempted Sun RPC high port access

The snort rule for this alert is given in the 1.7 version of snort (misc-lib) as:

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

In other words, the alert is triggered when an external host tries to access a local host on one of the common Sun RPC port numbers (32771). This particular alert was caused by the host 205.188.153.99 attempting to contact MY.NET.98.186 from port 4000 to port 32771. The alerts all took place between 00:46:36 and 01:17:29 on the 15th July 2001.

Port 4000 is a common IRC port [GCIAGB§39], and the most likely scenario for this alert is a student computer being used in the early hours of the morning for an IRC discussion, with the computer having randomly chosen the port 32771 for communications. However, this assumption should be verified with firewall and application logs before the possibility that it is a clever backdoor masquerading as an IRC session.

See the 'SUNRPC highport access!' section (§3.6.16 below) for further information on this trigger.

3.6.2 connect to 515 from inside

These two alerts and the 826 alerts for "connect to 515 from outside" are both concerned with printer communication, which usually uses this port. The two "connect to 515 from outside" alerts are both between MY.NET.179.78 and 24.13.123.8. The two connections were on 9th and 11th July 2001 at 20:51 and 10:41 respectively. The source ports being used were 49685 and 33263.

As there is no correlation between these alerts apart from the IP addresses being used, it can be surmised that either the user concerned has access to an external printer or they have an incorrectly configured printer setting on their computer. Further examination of the extended logs revealed similar connections between these two devices on the 5th, 8th and 17th July 2001. The behaviour observed ceased on the 17th July 2001, so investigations into the equipment concerned may not be advantageous, but interviewing the user could prove beneficial.

3.6.3 connect to 515 from outside

The two alerts comprising the "connect to 515 from inside" and the 826 alerts for this alert are both concerned with printer communication, which usually uses this port. The latter alerts fall into six categories:

Source Host	Destination Host	Date	Time	Count
64.158.160.82	MY.NET.132.*, MY.NET.133.*, MY.NET.135.*, MY.NET.137.*	09/07/2001	03:18:04-03:18:19	208
200.206.165.19	MY.NET.133.*, MY.NET.137.*	12/07/2001	08:05:31-08:05:40	135
202.224.218.44	MY.NET.132.*, MY.NET.133.*, MY.NET.135.*, MY.NET.137.*	14/07/2001	12:43:40-12:44:04	91

203.75.48.252	MY.NET.132.*, MY.NET.133.*, MY.NET.134.*, MY.NET.135.*, MY.NET.137.*	15/07/2001	22:39:14-22:39:35	347
210.103.58.65	MY.NET.132.*, MY.NET.134.*, MY.NET.137.*	11/07/2001	04:14:00-04:14:15	44
255.255.255.255	MY.NET.133.44	15/07/2001	05:43:47	1

Most of these alerts follow a standard pattern, with no source port below 1219 or above 4937. The target addresses are in the subnets listed above, but no subnet was completely scanned. The scans from 202.224.218.44, 203.75.48.252 and 210.103.58.63 were all preceded a few minutes before the intense scan by a probe to either the MY.NET.5.* or MY.NET.15.* subnets. Packets arrived in a very short interval, indicating an automated scan for open ports.

The only entry that differs from this pattern is the last entry in the table above. This was a packet ostensibly from 255.255.255.255 on port 31337. This combination gives an almost certainty that the packet was crafted. The 31337 port is often used as a backdoor for trojan horses such as BackOrifice, and the broadcast port 255.255.255.255 can only be used as a source address in a crafted packet. The purpose of this packet is unclear, but further investigation should be undertaken with reference to complete external firewall logs.

3.6.4 External RPC call

There were 1,357 connections from external hosts to the portmapper address, port 111. This port is used to discover which RPC programs are running on a host, and for translating information about a package into an active port number. This information is often used for targeting attacks against computers more precisely.

Source Host	Destination Host	Date	Time	Count
24.18.229.6	MY.NET.13?.*	11/07/2001	20:49:37-20:49:40	18
61.10.19.164	MY.NET.13?.*	11/07/2001	18:25:41-19:22:49	394
66.74.208.214	MY.NET.13?.*	10/07/2001	18:27:19-18:27:23	20
210.223.52.151	MY.NET.13?.*	13/07/2001	03:48:10-03:54:57	397
211.79.76.65	MY.NET.13?.*	09/07/2001	14:15:35-14:15:43	22
211.100.112.190	MY.NET.13?.*	11/07/2001	12:42:55-12:43:01	60
211.217.77.163	MY.NET.13?.*	15/07/2001	08:13:01-08:15:28	446

The probes were against the MY.NET.132, MY.NET.133, MY.NET.134, MY.NET.135 and MY.NET.137 subnets. Investigations should be carried out into whether the machines concerned replied to the information requests, and as to whether the machines are vulnerable to attacks against RPC-registered applications.

This scan, in conjunction with the "connect to 515 from outside" in §3.6.3 above leads to a question about the network structure. These scans are only being observed on the MY.NET.132, MY.NET.133, MY.NET.134, MY.NET.135 and MY.NET.137 subnets. If this is an extensive scan then it could be assumed that at least the MY.NET.136 subnet would be scanned, if not the rest of the class B subnet. The analysis below of Out Of Spec packets (§3.8) shows that more than just these subnets can be seen by the Snort logging machine.

This leads to a number of possibilities. The simplest, but least likely, is that the scanner or scanners only scanned the hosts observed. This is unlikely as a methodical approach usually produces more certain results. Another simple solution may be that the Snort monitor producing the alerts data is different from the monitor producing Out Of Spec logs. Once these have been ruled out, the options of firewalls with limited, but open, holes should be investigated.
It is to be strongly recommended that further investigation be carried out into the limited number of alerts received with respect to firewall and perimeter log analysis.

3.6.5 High port 65535 tcp - possible Red Worm – traffic

There were 64 events covering this one (48) and the udp alternative (16). These alerts are raised when a packet is observed originating, or destined for, port 65535. This is the highest possible port number, and is completely legal. However, it is used by a number of malicious applications as a fixed port number to use for data transfer. This alert is reported as being 'Red Worm'[10], but there are many other alternatives such as the RC1 Trojan.

Thirteen of the alerts appear to be a part of three 'normal' TCP or UDP traffic, with request and reply packets being sent in either or both directions. These three communications were:

First Host	Second Host	Protocol	Date	Time	Count
MY.NET.218.214:6436	212.209.158.149:65535	TCP	10/07/2001	08:47	4
MY.NET.98.143:3658	24.182.2.226:65535	TCP	13/07/2001	08:09	7
195.179.0.28:65535	MY.NET.70.242:27963	UDP	15/07/2001	08:37	2

Note that in each case the worrisome port is on the external host. Therefore, these are likely to be false positives, although investigations of the internal hosts concerned would not be unwarranted. However, correlations between this alert and other events such as "Port 55850 tcp - Possible myserver activity - ref. 010313-1" (§3.6.10 below) indicate that there is a high probability that MY.NET.218.214 is compromised with gnutella [Gnutella] or some other application using port 6346.

Of the remaining 51 packets concerning port 65535, 33 were with e-mail ports (smtp – port 25 and pop-3 – port 110), one on port 53 (dns), and one with the auth port (113). The remaining 16 connections were on apparently random higher ports, from 1107 to 27963.

Note that this latter port is mentioned above. All three connections to this port (and one to port 27961) were to MY.NET.70.242. This implies that this host runs a service or services on port 27963 and/or 27961, although the latter may be a typo. Port 27963 is not a known trojan port, although 27960 can be used as a Quake server and this may be related. Further investigation of this machine is warranted and recommended.

There is no obvious correlation between the other events of this type, and it is recommended that they be accepted as normal TCP and UDP traffic.

[10] This worm is not the CodeRed worm, which this alert predates and which uses a different transport mechanism.

3.6.6 High port 65535 udp - possible Red Worm – traffic

For an analysis of this alert, see §3.6.5 above.

3.6.7 ICMP SRC and DST outside network

For an analysis of this alert, see §3.6.19 below

3.6.8 NMAP TCP ping!

The 46 alerts of this type were generated by the snort rule:

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

In other words, any TCP packet destined for the internal network with the ACK flag set and an acknowledgement number of 0. Whilst this ack number is legitimate, it is unlikely to occur regularly and so is used as one of the signatures for an NMAP scan, often the first stage for a Operating System or Protocol scan.

The only real correlation between the source addresses used is that a third (18) of the packets were from 204.167.220.253, occurring across the week, mainly aimed at MY.NET.1.8 and MY.NET.1.8. On the destination address side, 21 of the alerts were aimed at MY.NET.1.0/20.

Another interesting correlation was that 10 of the packets were aimed at MY.NET.6.7, MY.NET.6.14, MY.NET.60.14 and MY.NET.60.17. This sequence of numbers does appear to be an attacker probing for a machine for which they know the general address but are not sure of the exact address..

All of the packets were designed to bypass firewalls, with source and/or destination address being a commonly used port number. For example, six packets were to and from port 53 (DNS) and all of the rest of the packets involved web ports (80 or 443). Both of these protocols are often allowed through firewalls. The indication that only these ports have been seen could be due to the fact that only these scans took place, or that the Snort machine is positioned behind a firewall.

The action taken on these alerts depends on the institutions attitude to external scans. If such scans are not a concern then no action needs to be taken, otherwise firewalls should be configured to block all packets with an acknowledgement number of 0.

3.6.9 Null scan!

This alert is generated by scan tools, which use a sequence of NULL entries in the Flags, Sequence and Acknowledgement fields of the TCP headers. The snort rule is:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Null Scan!";flags:0; seq:0; ack:0;)
```

There is no particular correlation between the packets causing these 32 alerts. However, half of the alerts were destined for port 1214 (kazaa [Kazaa]) and half were from the Class A network 24.0.0.0/8. There was no direct correlation between these hosts, but see §3.6.20 below for a fuller

description of traffic on port 1214. As mentioned above in §3.6.8, the action taken on these alerts depends on the attitude to external scans. Firewalls should be configured to block all packets with the relevant flags set to 0, or no action could be taken at all if such scans are not a concern.

3.6.10 Port 55850 tcp - Possible myserver activity - ref. 010313-1

No information about this (presumed) trojan activity, nor any reference to the port number can be found in any of the standard port lists [NeoPort], [SnorPort], [OntPort], nor in the records for SANS or some of the more common mailing lists. Therefore, it must be assumed that it is an internal rule, which was probably entered into the Snort Database on the 13th March 2001.

Full details about risks involved in these 112 alerts be given only with full disclosure of the reasons behind the addition of the rule to the configuration files. However, 82 of the alerts were to e-mail ports (25 – smtp, 110 – pop3 and 143 – imap) and a further fourteen were to port 443 (https – secure http). These may be communications that were designed to bypass firewalls, but they may just be valid activity.

More worrying is the connections between MY.NET.218.214 and three external hosts (132.199.101.19, 193.251.10.16 and 211.135.120.218). All of these communications were between port 55850 on the external host and port 6346 on the internal host. This would indicate, especially when correlated with other alerts (such as §3.6.5 above), that the internal host is compromised by gnutella [Gnutella] or some other application using port 6346.

3.6.11 Possible trojan server activity

This alert is named because it is designed to analyse activity on port 27374, which is usually associated with the SubSeven trojan. There are 383 alerts of this type, of which no single host dominated the alerts. The most interesting alerts in the logs are outlined below.

195.222.189.75 communicated with two internal hosts on the 11th July 2001, both sessions lasted for a number of request-reply pairs. MY.NET.110.157 was active around 04:43 and MY.NET.232.69 at 06:41. This does indicate that the local machines are running programs which are listening on the SubSeven port. Similar evidence can be presented for the local hosts MY.NET.6.53, MY.NET.60.16, MY.NET.75.82, MY.NET.98.121, MY.NET.98.142, MY.NET.98.201, MY.NET.100.87, MY.NET.110.69, MY.NET.111.155, MY.NET.146.26, MY.NET.153.175 and MY.NET.253.115. All of these hosts should be investigated for possible SubSeven compromises.

MY.NET.60.11 is a special case. This host has been probed by (and replied to) 64.228.84.102, 64.229.68.232, 202.63.219.126 and 209.181.206.99. It has also scanned a number of hosts using 27374 as a source port. This latter fact may be a red herring, but the number of hosts which have contacted MY.NET.60.11 indicates that it is on a list of vulnerable hosts and should be investigated with some urgency.

MY.NET.100.230 originated a number of communication attempts with 63.97.226.2 between the imap port (143) and 27374. These took place on the 9th and 12th July 2001, some of which were answered. This indicates more that 63.97.226.2 is infected than MY.NET.100.230, but the host

should be investigated for SubSeven client software. Alternatively, this could be a 'noisy' imap connection between these hosts which re-used the high port more than once.

This alert has also increased the evidence that MY.NET.218.214 is running the gnutella client, with a communication using ports 27374 and 6346 on the 12th July 2001 at 07:33.

3.6.12 Queso fingerprint

There are 233 alerts which match the Snort rule for the Queso scanning tool.

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

This alert triggers when the SYN flag is set along with the two reserved flags in the TCP header. However, as is discussed in §3.8.3.1 below, it is believed that the network being monitored has modern network devices which utilise these bits for congestion purposes. This assumption leads to the recommendation that these alerts are treated as a false positive.

However, if this is not the case then, as mentioned above in §3.6.8, the action taken on these alerts depends on the corporate attitude to external scans. If such scans are not a concern then no action needs to be taken, otherwise firewalls should be configured to block all packets with the reserved bits set.

3.6.13 site exec - Possible wu-ftpd exploit - GIAC000623

There were seven alerts for this exploit, all of which occurred on the 15th July 2001 between 09:37 and 09:45. The two Snort rules for this exploit are given here:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP EXPLOIT wu-ftpd 2.6.0 site exec
overflow"; content: "SITE EXEC %p"; nocase; flags: A+; depth: 16;
reference:arachnids,285; classtype:attempted-admin; sid:345; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP EXPLOIT wu-ftpd 2.6.0 site exec
overflow"; content: "|66 25 2E 66 25 2E 66 25 2E 66 25 2E 66 25 2E|"; flags: A+;
depth: 32; reference:arachnids,286; classtype:attempted-admin; sid:346; rev:1;)
```

All of the alerts originated from 211.46.39.194 and were targeted at MY.NET.144.59 (five alerts at 09:37) and MY.NET.99.85 (two alerts at 09:45). It is strongly recommended that the two servers concerned are examined for the presence of wu-ftpd and if the application is found an updated version is installed.

3.6.14 SMB Name Wildcard

This alert is caused by the command '`nbtstat -A <IP address>`' [SMBWild] or a comparable command implemented in a program. It is used to gain a list of NetBIOS information for a host, which can then be used for further probes or attacks.

Of the 10,039 alerts reported during the week in question, there are only a few hosts involved in more than 10 scans. 24.189.216.251 scanned MY.NET.5.45 (5,527 packets), whilst 216.150.152.145 scanned MY.NET.5.44 and MY.NET.5.45 (4,015 packets). The interesting factor here is the number of scans against two hosts. This would imply that these two hosts have

been heavily targeted by one or two attackers, and therefore should be thoroughly investigated for compromises in their NetBIOS systems.

3.6.15 STATDX UDP attack

This alert references a specific attack against a Linux Statd vulnerability [Statdx]. The snort trigger given here is copied from the Arachnids database [Arachnid].

```
alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS442/rpc_rpc-statdx-exploit"; flags: A+; rpc:
100024,*,*; content: "/bin|c74604|/sh"; classtype: system-attempt; reference: arachnids,442;)
```

There is only one alert for this trigger, from 210.223.52.151 to MY.NET.6.15. The direct targeting of the event indicates that the attacker had prior knowledge of the host MY.NET.6.15. Therefore, investigation of the host is recommended to ensure that it has not been compromised.

3.6.16 SUNRPC highport access!

There is no Snort rule available for this alert, but observational techniques indicate that this alert is almost identical to §0 above. However, this trigger resulted in 150 alerts, whereas the trigger used in the 'Attempted Sun RPC high port access' section gave 32 alerts. This indicates that either the trigger used in the previous rule is incorrect, or that the data comes from two different Snort engines.

Almost half (69) of the alerts comprised traffic between 24.9.158.233 and MY.NET.163.17 on the port 22 (ssh). If MY.NET.163.17 can run an ssh client, then this is most likely to be a false positive. Of the remaining alerts, 59 were to MY.NET.218.146. This host should be checked for validity of external hosts contacting portmapped applications, and if correct then the alerts can be accepted.

The remaining 20 alerts were split evenly between MY.NET.217.10, MY.NET.217.214 and MY.NET.253.53 with communications to the latter host being on port 25 (smtp). These communications could be investigated if required, but do look to be normal tcp client-server traffic.

3.6.17 TCP SRC and DST outside network

For an analysis of this alert, see §3.6.19 below

3.6.18 Tiny Fragments - Possible Hostile Activity

This alert is designed to trigger when packets with a data payload less than 25 bytes are observed on the snooped network. The snort rule for this alert is:

```
alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"Tiny Fragments – Possible Hostile
Activity"; fragbits:M; dsize: < 25; classtype:bad-unknown; sid:522; rev:1;)
```

A packet is naturally fragmented when it is larger than the maximum size allowed on a network across which it needs to travel. However, the normal value for this size is 1480 bytes for the first and subsequent packets, with only the final packet in the sequence being the size required to complete sending the packet. The alert does not trigger when the last packet is received, reducing

the risk of false positives. This is done through the '`fragbits:M`' entry in the alert, which indicates that more fragments are to follow.

In a fragmentation attack such as this one the malicious packet is split into small fragments. The intention of this is to bypass signature-based IDS tools which use offsets within packets as an index to the location of the signature. There are few occasions where traffic such as this is legitimate, mainly happening when routers or other network devices have been incorrectly configured with a small MTU.
There were 677 instances of this alert, split into two attacks. The first attack was between 133.25.193.54 and MY.NET.218.214 and took place on the 9th July 2001 between 04:55 and 06:12. The second attack was from 133.25.193.234 against the same host, and took place on the 10th July 2001 between 04:46 and 04:59. The similarity between the attacking IP addresses indicate that this may be the same attacker using a dial-up service. Data from Appendix B shows that this is probably the case, and that the attacking host is in Japan.

Further analysis is not possible with the information provided in Snort logs, but there is definitely enough evidence to examine firewall logs in detail to discover the extent of the communication, and if necessary to examine the host MY.NET.218.214 for possible compromises.

3.6.19 UDP SRC and DST outside network

46,441 packets were noticed by the Snort filter which had both source and destination address outside the known internal network. This can indicate a poor routing configuration, a source-routed attack using the monitored network as a jumping off station or a valid routing of packets to a third party through the monitored network.

Of these packets, 7 were ICMP, 277 were TCP and 46,157 were UDP. 58 of the packets, mainly UDP packets involving port 137 (netbios) were using the non-routable addresses 10.0.0.0/24 or 192.168.0.0/16. These packets have been removed from further analysis but routers should be examined for correct behaviour with respect to non-routable addresses.

Hosts on the networks 169.254.0.0/24 and 144.111.0.0/16 are very heavily represented, totalling 168 of 236 unique source addresses. Therefore, it would be worth investigating whether these networks are being routed through the monitored network and whether this is normal behaviour or due to a mis-configuration of a piece of network equipment.

The vast majority of the alerts, 40,716, were destined for 225.130.160.2 and 225.130.160.3 on ports 4446 to 4449, with over 36,000 being destined for 225.130.160.2 on port 4446, all from 130.160.4.60 on ports 1041 or 1042. All other connections were also from hosts on the 130.160.4.0/24 network. These ports do not have an application in common, so if access to the hosts is available clarification of the application in use would be beneficial.

All non-local packets should be treated with suspicion as they should not be passing through the local network under normal operating situations. If the local network is acting as a gateway for other users then this should be filtered at an early stage, otherwise problems such as the ones given here may occur. Other reasons for packets such as these include internal networks using non-local addresses and source-routed packets. Actions taken on the latter depends on the universities' attitude to these packets, but the former should either be renumbered or be added to the lists of 'acceptable' internal hosts.

3.6.20 Watchlist 000220 IL-ISDNNET-990517

Watchlists are lists of suspect address ranges which have been observed carrying out malicious activity at some stage in the past. Watchlist 000220 refers to any traffic from the network 212.179.0.0/16, whilst Watchlist 000222 refers to network 159.226.0.0/16. In this case, the former trigger produced 1,981 alerts whilst the latter gave 385 alerts.

Alerts caused by traffic from 212.179.0.0/16 mainly comprised three groups. The largest group is of traffic between 212.179.15.28 on port 2224 and MY.NET.218.90 on port 41186 which took place on the 12th July 2001 between 23:23 and 23:47. There are no known applications or trojans etc which use either of these ports, so investigation of the local host is recommended.

The second group of alerts is made up of 660 packets to local hosts on port 1214 (kazaa [Kazaa]). This file sharing application uses port 1214 as a lightweight http server for communicating files between users. Internal hosts used in these communications include MY.NET.150.143 (200 packets), MY.NET.70.97 (178 packets) and MY.NET.150.225 (162 packets). If the use of such file sharing applications is discouraged on the network then these hosts should be investigated.

Communication between 212.179.31.180 on port 3471 and MY.NET.98.178 on port 4189 makes up the 512 packets of the third group. Port 3471 is listed as jt400-ssl, whilst port 4189 does not have a listing in the common reference sites. Therefore, it may be assumed that either this traffic is a secure communication originated from the internal site, or there is an application using port 4189 that is not known about. Either way, investigation is recommended.

Connections from 159.226.0.0/16 just fall into two categories, connections to port 25 on hosts on MY.NET.253.0/24 and a session between MY.NET.100.37 and 159.226.41.166. The latter session occurred on the 11th July 2001 and comprised 155 packets between port 33243 on the internal host and 23 (telnet) on the remote host. There are a number of trojan applications that use port 23 as this is commonly allowed through firewalls, but this particular connection 'feels' as if it is a valid telnet session.

The 208 packets to the smtp port (25) on internal hosts were destined for MY.NET.253.41 (63 packets), MY.NET.253.42 (54 packets) and MY.NET.253.43 (85 packets), with the remainder aimed at MY.NET.6.47. 159.226.152.1 was the main source address with 124 packets in four sessions on the 10th and 15th July 2001. If these internal hosts are, indeed, mail servers then this may well be valid mail transactions from remote uses. However, the mail servers may be being used in order to assist mail-bombing and spamming. If external access to the mail servers is not required then it is recommended to alter firewall rules to restrict access.

3.6.21 Watchlist 000222 NET-NCFC

For an analysis of this alert, see §3.6.20 above.

3.6.22 WinGate 1080 Attempt

This alert was repeated 528 times, and was triggered by the Snort rule:

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

WinGate [WinGate] is an application that allows multiple computers to use a single internet connection. This alert is triggered by any incoming connections to port 1080, one of the ports used by the applications. This port is within the acceptable list of ports for connections to services, but the alert only triggers on packets incoming to the port number. Therefore, it is most likely to be either a probe for WinGate, a connection to some other application (for example a web server) using port 1080 or normal TCP communications.

213.23.45.252 was one address that definitely scanned for servers on this port, being the cause of 165 of the alerts, scanning quite widely across the MY.NET network. 168.70.145.77 was more targeted, only scanning MY.NET.70 and MY.NET.71 addresses in its 52 attempts. On the internal side, MY.NET.60.11 (43 packets) and MY.NET.217.142 (40 packets) were the prime suspects. The former host has already been mentioned, in §3.6.11 above, as a possibly compromised host, whilst the latter was the subject of a number of repeated scans. This may indicate that the host was compromised at some stage in the past and is on a list of vulnerable servers somewhere on the internet.

3.7 Scan Analysis

During the week being analysed in detail, 324,684 packets were listed in the scan files given in §3.4 above. The quietest day was Tuesday 10th July 2001 with 33,846 packets, whilst the noisiest was Saturday 14th July 2001 with 52,398 packets.

Almost 300,000 of the packets in the scan logs can be broken down into a relatively small number of similar scans, either by source or destination address or by protocol. These are given here with a relatively simple explanation of the scan. Further analysis and explanation is available if required.

3.7.1 Scan from MY.NET.160.114

During the week, MY.NET.160.114 was constantly scanning other hosts from port 777. Of the total of 92,008 packets, over 66,000 were destined for port 27005, with the remainder being spread relatively randomly across the packet spectrum. The analyst can find no trace of a specific tool designed to use port 777 as a source port (the common 'trojan' lists such as [NeoPort], [SnorPort] and [OntPort] only list destination ports) on the internet. Port 27005 is usually given as a Flex-LM port but this operates as TCP and the scans observed here are UDP. Therefore, a thorough examination of MY.NET.160.114 for evidence of compromises is to be recommended. Additionally, of course, it would need examining for network management tools that have been incorrectly configured.

3.7.2 Scans to port 21 (ftp)

Throughout the week a large number (53,107) of packets destined for the ftp port (port 21) were observed. Over 23,000 of these came from 24.101.17.26, with 211.120.40.2 generating just over 10,000 packets and 63.23.174.61, 213.46.30.84 and 217.57.19.30 all raising between 4,800 and 8,500 packets each. Normal FTP communication uses port 21 as a control port, with port 20 being used to transmit data. However, passive ftp can be used which transfers all data across port 21. This may be the cause of this traffic but this is unlikely as the packets all contained the 'SYN' flag set, which indicates that they are the start of a session, not data being transmitted during its operation.

Other traffic on port 21 could include backdoor applications such as BackConstruction, which use the port as it is often allowed through firewalls. However, it is the opinion of the analyst that these are simply scans for open ftp servers with the intention to collect information prior to a more extensive attack at a later date.

The decision as to what action to take about these scans depends considerably on the Universities' attitude to the risk involved. Firewalls could be tightened to only allow traffic in to known FTP servers but this would restrict use and academic staff would almost certainly complain.

3.7.3 Scans to port 6970 (GateCrasher)

There were a total of 46,269 scans to the MY.NET class B network, from five source addresses, from 205.188.224/27. Port 6970 for TCP is listed in the backdoor lists as the GateCrasher trojan, and these packets have the earmarks of a scan for this trojan. The only problem with this solution is that the packets are UDP. Therefore some other solution needs to be discovered. The complete list of hosts is

Source Address	Count
205.188.233.121	6,872
205.188.233.153	16,985
205.188.233.185	6,168
205.188.244.121	13,515
205.188.246.121	2,729

Note the similarity to three of the addresses – this may indicate that the source addresses are spoofed. However, this would negate the purposes of the scan as the 'real' machine would get any replies rather than the attacking PC, therefore it is more likely to be a coincidence. Until such time as the real reason for the scans has been identified, it is to be recommended that this port be blocked to incoming traffic at the external firewall.

3.7.4 Scans to port 53 (dns)

There are 32,301 packets destined to port 53 in the logs, of which 2,065 were outgoing DNS packets from MY.NET.100.230 to various hosts. It is a reasonable assumption that this host is a DNS server on the local network, if this is not the case then further investigation may be required at a local level.

On the incoming packets, the probing hosts were 212.227.251.13 (13,960 packets), 213.39.73.250 (10,851 packets) and 210.241.238.230 (5,425 packets). There was no correlation for target host addresses, and this is believed to be extensive scanning for DNS servers. Whether this is for nefarious or acceptable purposes depends on the relationship between the University and the owners of the IP addresses.

3.7.5 Scan from 211.120.40.2

Between the 13th and 15th July 2001, the host 211.120.40.2 performed two TCP scans of the University system. The first scan (of 10,682 packets) was for the ftp port, and is outlined above in §3.7.2. The second scan, on the 15th July 2001, comprised 12,099 packets to the internal network on port 1. This is an interesting port to scan as no application should be actively using it. This must, therefore, be a scan to find active hosts which would reply with reset packets and may be a part of a mapping exercise. As with much of the activity in this section of the report, any action should be in keeping with the universities' security policy and may include alterations to the firewall for either the source address or the destination port.

3.7.6 Telnet scan from 62.110.146.3

Between 02:59 and 03:19 on the 11th July 2001, 62.110.146.3 probed the internal network, performing a protocol scan for port 23 (telnet) with 10,391 packets. There was no correlation between either the destination hosts or source addresses and this is believed to be a simple scan for active telnet daemons or some other application listening on this port.

3.7.7 WebServer Scans

A number of packets incoming to the network had the appearance of being scans for web servers running on non-standard ports. 61.142.200.4 sent 5,313 packets between 09:42 and 09:54 on the 13th July 2001 to ports 8080 and 3128, respectively a common alternative httpd port and one of the common squid http proxy ports. This may well be a simple scan for these ports. However, it is more likely to be indicative of a RingZero trojan [SANSID08]. This program scans actively ports 80, 3128 and 8080. Packets destined to port 80 may well be removed from these logs for purposes of clarity.

The host 168.70.145.77 sent 3,366 packets between 21:25 and 21:33 on the 11th July 2001 to a large number of internal hosts, mainly on the MY.NET.70 and MY.NET.71 subnets. These packets were consistently scanning for a fixed number of 72 ports, all of which had the same approximate format as common http ports, for example in the middle of the range, the scanning software probed for ports 2000, 2020, 2080, 2128, 3000, 3030, 3080, 3128, 4000, 4040, 4080 and 4128. This compares to the 'standard' ports of 80, 8000, 8080 and the squid proxy port, 3128. This is definitely a system scan, any action will have to be with referral to the university security policy.

3.7.8 Scan from MY.NET.217.242

MY.NET.217.242 performed three extensive scans on the 12th and 14th July 2001. Firstly, it scanned 1,999 ports on 167.206.254.176, and then scanned 2,376 ports on 209.14.208.91. Both of these scans took place on the 12th July 2001. Finally, 212.46.64.180 was scanned 3,439 times on the 14th July 2001. These scans were almost certainly port scans of the hosts concerned.

Whether any action is taken against the user and/or machine concerned is a matter for the university to consider.

3.7.9 Scans to port 1214 (Kazaa/Morpheus)

During the entire week, a large number of source machines scanned different sections of the internal network for port 1214. In total, there were 7,724 packets received in the snort logs. There was no correlation between either the source or destination hosts, nor the source ports. This situation is believed to be a number of probes for the Morpheus file-sharing program, which uses port 1214 as a lightweight http daemon to aid file sharing. If this application is not operational then no further action is required, however it is recommended that incoming connections to this port be stopped at the firewall.

3.7.10 Scans to port 4665

Four internal hosts carried out a number of scans on the internet, scanning for applications running on port 4665. There were 4,506 scans in total, from MY.NET.217.38 (580 packets), MY.NET.218.154 (1,115 packets), MY.NET.218.18 (1,871 packets) and MY.NET.218.218 (940 packets). There is no application registered to port 4665 in any of the standard application lists [NeoPort], [SnorPort], [OntPort]. Therefore the reason for the scan is concerning, and the matter ought to be investigated with the owners of the systems concerned.

3.7.11 Scans to port 6346

A large number of external hosts scanned internal hosts on port 6346. This is the gnutella [Gnutella] server port, and the 2,601 probes are almost certainly external gnutella users attempting to discover other users of the system. Whether this is an area to be concerned about depends on the university security policy and the users concerned. Further investigation is not considered necessary but alterations to the firewall rulesets are to be recommended.

3.7.12 Scans to and from port 6112

There were a large number (1,780) of packets which were both to and from port 6112. The majority of these were 1,556 packets from MY.NET.98.159 to 65.1.205.108, with the remaining packets being from MY.NET.98.219 to miscellaneous external hosts. The port lists give UDP port 6112 as being either the Common Desktop Environment (CDE) process control daemon or a game called fsgs. With experience, the latter is more likely to be the cause of this behaviour. Actions to be taken depend, as usual, on the university security policy.

3.7.13 Miscellaneous Scans

A number of other scans took place from internal hosts which do not fit elsewhere in this report. MY.NET.160.169 probed the internet 2,872 times. There was no correlation between the destination hosts or ports, but the source port remained constant for a large number of consecutive requests. This is not normal behaviour, and further investigation is warranted. Similarly to this scan, MY.NET.217.142 scanned a large number of external addresses. The 1,491 packets were mainly sent from ports 2000-2010 and destined for ports from 7000-9000. The short ranges of these probes indicate a programmed solution, and investigation of the computer concerned is recommended.

MY.NET.179.78 probed 24.13.123.8 on a wide number of ports with 2,626 packets. This is almost certainly a 'simple' port scan, and action depends on the security policy. Finally, MY.NET.70.242 scanned the internet 682 times. What was strange about this UDP scan was that all packets were sourced from port 27963. This indicates a scripted scan and further investigation should be undertaken.

3.8 Out-Of-Spec Packets

During the two months of the scanning period, 38,229 out of spec packets were captured. These are packets that do not meet the normally accepted behaviour of TCP packets with respect to their flag status (byte 13 in the TCP Header). The decision as to which options were out of spec were defined by your network manager, a full list is available from him on request. Twenty-nine of the packets were flagged as packet fragments, and are not analysed here.

3.8.1 Packets sourced internally

Out of the total number of packets with invalid flags, only eighteen were sourced internally. Of these, twelve were between MY.NET.100.153 and 132.229.131.40 on the 15th June 2001 between 10:01 and 10:08. Snort reported that all of these packets had the TCP flags "`21S*****`". This indicates that the SYN flag has been set along with the two reserved bits. See the description in §3.8.3.1 below for details of this entry. Other packets sourced internally could be accepted as false positives or irrelevant.

3.8.2 Packets with no local address

Of the packets recorded, 173 did not have a local address as either the source or destination address. All of these were observed on the 22nd June 2001, and matched one of the following three criterion. 40 packets were from 192.168.1.1 to 216.235.163.151, and 21 from the same address to 216.235.163.163. All of these packets were received between 18:30 and 18:32, and all had both the source and destination port set to 0 (zero). The remaining 112 packets were received between 22:47 and 22:48, and were transmitted from 111.111.111.111 to 216.235.163.151. There was no observable pattern in the port numbers of the packets.

In all of the packets, the TCP flags were crafted, with a wide range of options configured, leading to the conclusion that the attacks are being used by an automated application to 'fingerprint' a host, and try to discover the operating system being used by way of the result of incorrect packets. The 111.111.111.111 address is almost certainly spoofed. The only other address of interest is 192.168.1.1 which is a non-routable address as according to RFC1918 [RFC1918]. Therefore either some router is incorrectly routing these addresses, or a local host has been configured with this address. Either way, investigations should discover which solution is the truth.

3.8.3 Flags Analysis

There were 104 unique flag settings mentioned in the 38,200 OOS packets captured. Highlights of the settings are the 25,321 packets with the SYN and FIN flags set and the 10,918 packets with SYN and the two reserved bits set. The former are likely to be scanning attempts and are discussed in detail in §3.8.4 below, whilst the latter are discussed in more detail in §3.8.3.1 following.

Ignoring the state of the reserved bits, 535 packets had at least five of the six flags set. These were broken down into "SFRP*U" (158), "SF*PAU" (102), "SFR*AU" (83), "SFRPA*" (77), "SFRPAU" (72), "*FRPAU" (23) and "S*RPAU" (20). These packets, along with most of the others, are probably crafted packets used for scanning purposes.

3.8.3.1 Reserved Bits

In a large percentage (26%) of the packets captured, the TCP flags were given as "21S*****". In recent common practice, the two highest bits (given as "21" here) are used as indications of congestion on the network in the Explicit Congestion Notification (ECN) model [RFC2481]. As the traces were captured on a university, it could be assumed that up to date hardware is being utilised.

This assumption should be checked for validity and if found to be correct, then the snort rule for determining OOS packets should be revised. However, as shown above in §3.6.12, this is also a possible fingerprint for the Queso scan tool, so any assumptions outlined below should be carefully examined with further investigation.

Note that for clarity, in the remainder of this analysis the two reserved bits will be referred to as 'ECT' (ECN-Capable Transport) and 'CE' (Congestion Experienced). Where a Snort report would give all flags as "21SFRPAU", in this document it would be shown as "ECSFRPAU".

3.8.4 Time Analysis

There was no direct correlation by time of packet capture, but 35% (13,602) of the packets arrived on the 22nd July 2001, 24% (9,152) on the 18th June 2001 and 5% (2,057) on the 11th June 2001. No other date achieved more than 1000 OOS packets per day.

3.8.4.1 11th June 2001

2,050 of the OOS packets captured on the 11th June 2001 were effectively identical – packets from 211.240.28.66 to hosts on the local network from port 21 (FTP) on the attacking host to port 21 (FTP) on the internal host. All packets had the SYN and FIN TCP flags set. There is also evidence that the Sequence and Acknowledgement fields in the TCP header are also forged.

3.8.4.2 18th June 2001

Out of the OOS packets captured on the 18th June, 97% (8,877) were identical to those observed on the 11th June with the exception that the source address is 61.13.106.35.

3.8.4.3 22nd July 2001

99% (13,457) of the OOS packets captured on the 22nd July 2001 comprised what appears to be a portion of a complete scan of the University Class B address. Packets had the TCP flags SYN and FIN set and are all to and from the POP2 port (109), and the scan originated from external address 195.82.167.59.

3.8.5 Source Address Analysis

1,365 source addresses were referenced in the OOS packet lists. Given here are the top dozen talkers by number of packets sent. Full descriptions of the hosts are given in Appendix B.

IP Address	Count
61.13.106.35	8,877
158.75.57.4	1,586
193.226.113.248	352
195.82.167.59	13,457
199.183.24.194	2,585
210.77.146.33	1,165
211.114.44.2	364
211.180.236.194	557
211.240.28.66	2,050
213.116.114.212	424
213.116.168.124	250
213.117.6.207	259

The top talker (195.82.167.59) is referenced in §3.8.4.3 above, whilst 61.13.106.35 and 211.240.28.66 are referenced in §3.8.4.1 and §3.8.4.2. IP addresses 211.114.44.2 and 211.180.236.194 are both involved in a SYN-FIN scan, to the ftp (21) and sunrpc ports respectively. All of the other top talkers are involved in ECN traffic, with SYN flag set along with the two reserved bits. The only anomaly is that 193.226.113.248 sent 246 of its packets to MY.NET.70.97, all on port 1214 (kazaa [Kazaa]).

3.8.6 Source Port Analysis

Of the 38,200 OOS packets, the top ten source ports by quantity are given here.

Port Number	Service Name	Count	Usual Flags
0	-	181	No correlation
20	ftp-data	215	ECT,CE,SYN
21	ftp	11,291	SYN,FIN
109	Pop2	13,460	SYN,FIN
111	sunrpc	557	SYN,FIN
1214	kazaa	583	No correlation
1341	qubes	66	No correlation
6346	gnutella-svc	149	No correlation
18245	?	108	SYN,FIN,RST,PSH,URG
60020	?	165	ECT,CE,SYN

The packets from the ftp (21), pop2 (109), sunrpc (111) and 18245 ports are likely to be part of Operating System scans. For descriptions of these see §3.8.4 above. The packets with the ECT, CE and SYN flags set that comprise the ftp-data (20) and 60020 details may be due to congestion and should be investigated as mentioned in §3.8.3.1 above. The other items could be from almost any scan and could be investigated further if required.

3.8.7 Destination Address Analysis

The distribution of destination addresses is much more even than that for the source addresses in §3.8.5 above. The table does, however, give the top ten receivers of packets.

IP Address	Count
MY.NET.70.97	643
MY.NET.98.139	706
MY.NET.100.165	2,486
MY.NET.109.234	715
MY.NET.202.54	209
MY.NET.253.41	931
MY.NET.253.42	828
MY.NET.253.43	860
MY.NET.253.114	1,093
MY.NET.253.125	408

All of these addresses were the target of packets which just had the SYN and ECN bytes set. With the exception of the first address, referenced in §3.8.5 above, the remainder were the recipient of packets destined for ports 25 (smtp), 80 (http), 443 (https) and 6346 (gnutella). There was no particular correlation between the other destination addresses.

3.8.8 Destination Port Analysis

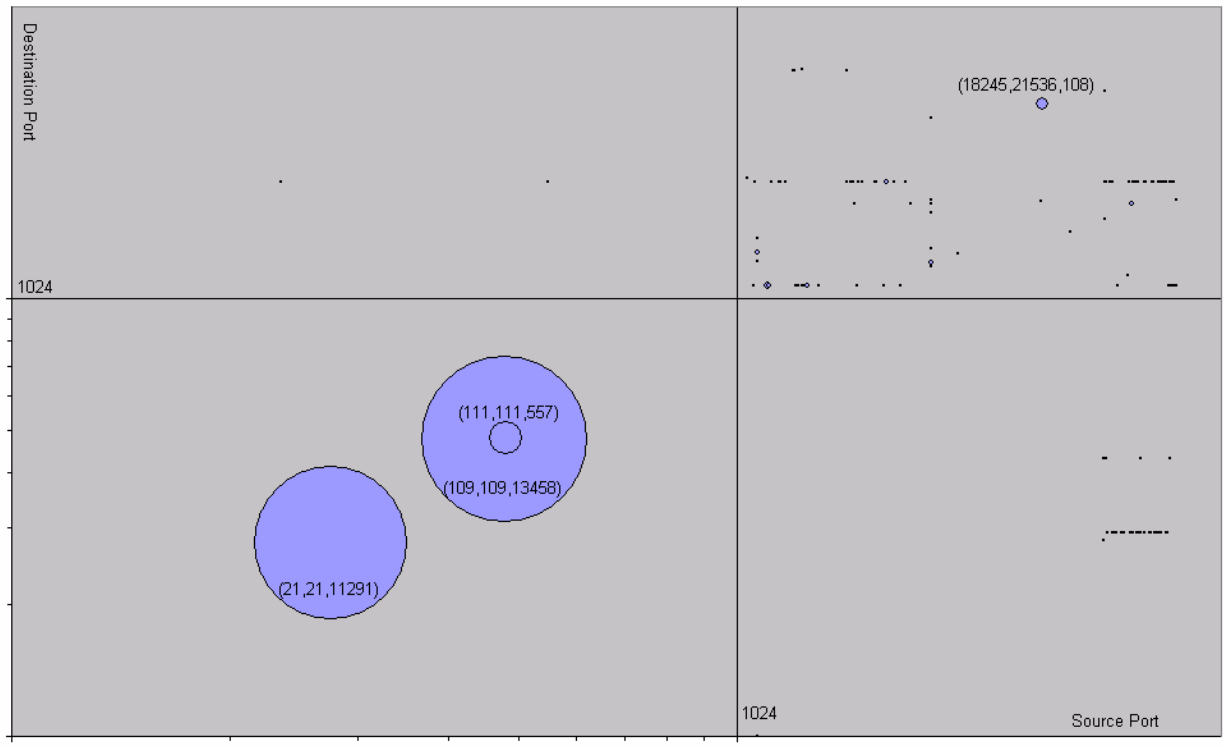
Of the 38,200 OOS packets, the top fifteen by quantity are given here.

Port Number	Service Name	Count	Usual Flags
0	-	129	No correlation
21	ftp	11,307	SYN,FIN
22	ssh	66	ECT,CE,SYN
23	telnet	44	ECT,CE,SYN
25	smtp	2,765	ECT,CE,SYN
53	dns	45	ECT,CE,SYN
80	http	4,385	ECT,CE,SYN
109	pop2	13,458	SYN,FIN
111	sunrpc	557	SYN,FIN
113	auth	267	ECT,CE,SYN
443	https	58	ECT,CE,SYN
1214	kazaa	583	ECT,CE,SYN
6346	gnutella-svc	2,493	ECT,CE,SYN
6347	gnutella-rtr	115	ECT,CE,SYN
21536	?	108	SYN,FIN,RST,ACK

The packets destined for the ftp (21), pop2 (109) and sunrpc (111) ports are likely to be part of Operating System scans. For descriptions of these see §3.8.4 above. The packets with the ECT, CE and SYN flags set that comprise most of the other packets may be due to congestion and

should be investigated as mentioned in §3.8.3.1 above. Further investigations into this could be carried out if required.

3.8.9 Link Graph



The above graph gives a representation of OOS packets. The X axis is a logarithmic representation of the source ports, with the Y access similarly being a logarithmic representation of the destination ports. The size of the 'dots' indicate the number of hits. The two grid lines represent port 1024. Six lines can be seen clearly, four making up a 'box' just above the grid lines, one just to the right of this box and one in the lower right quadrant.

The four horizontal lines represent destination ports 25 (smtp) in the lower right quadrant, 1214 (gnutella) as the lower edge of the box, and 6346 (kazaa) as the top of the box and the line to the right hand side. The left and right hand edges of the box are similarly ports 1214 and 6346, but as source addresses rather than destination addresses.

References

[APNIC] Asia Pacific Network Information Centre; http://www.apnic.net/

[Arachnid] Arachnids (Advanced Reference Archive of Current Heuristics for Network Intrusion Detection Systems); http://whitehats.com/ids/index.html

[ARIN] American Registry for Internet Numbers; http://www.arin.net/whois/index.html

[Cisc2001] Cisco; Cisco Secure IDS – Excluding False Positive Alarms; http://www.cisco.com/warp/public/707/f_pos.html

[Deba2000] Debar, Hans; Why does my intrusion-detection system generate false alarms/no alarms. SANS Intrusion Detection FAQ; http://www.sans.org/newlook/resources/IDFAQ/false_alarms.htm

[EEyeCR] Electronic-Eye CodeRed Analysis; http://www.eeye.com/html/Research/Advisories/AL20010717.html

[EEyeCRII] Electronic-Eye CodeRed II Analysis; http://www.eeye.com/html/Research/Advisories/AL20010804.html

[GCIAGB] GCIA Dissertation, Guy Bruneau; http://www.sans.org/y2k/practical/guy_bruneau_gcia.doc

[GCIAMB] GCIA Dissertation, Marc Bayerkohler; http://www.sans.org/y2k/practical/Marc_Bayerkohler_GCIA.html

[GCIAPA] GCIA Dissertation, Paul Asadoorian; http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc

[GCIAPG] GCIA Dissertation, PJ Goodwin; http://www.sans.org/y2k/practical/PJ_Goodwin_GCIA.doc

[GCIAPrac] Index of GCIA Practicals; http://www.sans.org/giactc/gcia.htm

[GCIARC] GCIA Dissertation, Robert Currie; http://www.sans.org/y2k/practical/Robert_Currie.doc

[GCIATB] GCIA Dissertation, Teri Bidwell; http://www.sans.org/y2k/practical/Teri_Bidwell_GCIA.doc

[Gibs2001] Steve Gibson; Personal Firewall Scorecard; http://grc.com/lt/scoreboard.htm

[Gnutella] Gnutella, file sharing applicaion; http://gnutella.wego.com/; http://www.gnutellanews.com/

[Hays1981] Hayslett, H.T.; Statistics Made Simple, Heinemann, 2nd ed, 1981

[HewlOV] Hewlett-Packard OpenView; http://www.managementsoftware.hp.com/products/express/index.asp

[Inci0803] Incidents.org, Source for trace in §1.3; http://www.incidents.org/archives/intrusions/msg01295.html

[Incidents] Incidents.org; http://www.incidents.org/

[ISS] ISS Website; http://www.iss.net/

[JPNIC] Japan NIC; http://www.nic.ad.jp/en/db/index.html

[Juli2000] Julisch, Klaus; Dealing with False Positives in Intrusion Detection; http://www.raid-symposium.org/raid2000/Materials/Abstracts/50/Julisch_foils_RAID2000.pdf

[Kazaa] Description of Kazaa/Morpheus file sharing application; http://www.openp2p.com/pub/a/p2p/2001/07/02/morpheus.html?page=2

[KRNIC] Korean whois database; http://whois.nic.or.kr/english/index.html

[NeoPort] Neohapsis Ports List; http://www.neohapsis.com/neolabs/neo-ports/

[Nort01] Northcutt, Stephen et al; Intrusion Signatures and Analysis, New Riders, 1st ed, 2001

[Nort99] Northcutt, Stephen; Network Intrusion Detection – An Analyst's Handbook, New Riders, 1st ed, 1999

[ONCTPort] ONCTek List of Trojan/Backdoor port activity; http://www.onctek.com/trojanports.html

[Perl561] ActivePerl v5.6.1 for Win32; http://aspn.activestate.com/ASPN/Downloads/ActivePerl/index/

[Phra51] Phrack, Issue 51, September 1997. Article 6 – Loki2 (the Implementation); http://www.phrack.com/search.phtml?view&article=p51-6 mirrored at http://packetstormsecurity.org/mag/phrack/phrack51/P51-06

[Pott2000] Potter, Bruce; Intrusion Detection System for Fun and Profit; http://www.shmoo.com/wp/ids/

[RIPE] Réseaux IP Européens; http://www.ripe.net/

[RFC0760] RFC for the Internet Protocol; http://rfc.net/rfc0760.html

[RFC0761] RFC for TCP Packets; http://rfc.net/rfc0761.html

[RFC0768] RFC for UDP Packets; http://rfc.net/rfc0768.html

[RFC0777] RFC for ICMP Packets; http://rfc.net/rfc0777.html

[RFC0976] RFC for UUCP Mail; http://rfc.net/rfc0976.html

[RFC1918] RFC for Address Allocation for Private Intranets; http://rfc.net/rfc1918.html

[RFC2481] RFC for Explicit Congestion Notification (ECN); http://rfc.net/rfc2481.html

[SANS] SANS Website; http://www.sans.org/

[SANSID08] SANS GCIA training, ID_08_tcpdump_sbs.pdf

[SecP2001] SecurityPortal; Report on Personal Firewalls; http://securityportal.com/cover/coverstory20000717.html

[SecuFocu] SecurityFocus mailing list, Focus-IDS; http://www.securityfocus.com/ids

[SMBWild] Description of SMB WildCard probe; http://www.sans.org/newlook/resources/IDFAQ/port_137.htm

[Snort] Snort – The Open Source Network IDS; http://www.snort.org/ or http://snort.sourcefire.com/.

[SnorPort] Snort – Ports Database; http://www.snort.org/ports.html

[SnorRule] Snort, Standard Rulebase (snort.rules); http://snort.sourcefire.com/downloads/snortrules.tar.gz

[SnorSnar] SnortSnarf, Snort Log file analysis, Silicon Defense; http://www.silicondefense.com/software/snortsnarf/

[Statdx] Statdx Linux exploit. Description; http://whitehats.com/info/IDS442 Exploit Information; http://neworder.box.sk/showme.php3?id=2440

[TCPDump] TCPDump network monitoring software; http://www.tcpdump.org/

[WinGate] WinGate network connection sharing software; http://wingate.deerfield.com/

Appendix A – Conventions

A.1 IP Addresses

In this paper, some IP addresses have been sanitised. These have used the convention A.x.y.z for a class A address, B.B.y.z for class B addresses and C.C.C.z for class C addresses. x, y and z may be given numbers or may be left algebraically. Where two addresses of the same class need to be differentiated, a numeric suffix will be attached, i.e. B.B1.y.z.

A.2 Hostnames

Hostnames related to addresses are given in sanitised format, indicating machine purpose (i.e. fw indicates a firewall, ws a workstation). External domain and/or host names, where used, are given as of the date of the detect. Note that names and assignments may have altered since the detect was first logged.

A.3 Dates and Times

All dates used in this document are given in UK format, DD/MM/YYYY. Where possible dates have been spelled out (i.e. 9th July 2001) but in some instances this is not possible. All times in the paper are given as GMT (UTC).

Appendix B – IP Addresses

All external IP addresses referenced in this document are given here, with whois data from one of the global whois database lookup engines, [APNIC], [ARIN], [JPNIC], [KRNIC] and [RIPE]. Information given here is as accurate as possible, but is correct as of the 17th September 2001 and should not be taken as accurate without reference to the relevant originating database.

IP Address	Name/Location
024.000.000.000/24	@home network, 450 Broadway Street, Redwood City, CA 94063, US
024.009.158.233	cc916074-a.catv1.md.home.com, @home network, 450 Broadway Street, Redwood City, CA 94063, US
024.013.123.008	cc61691-a.abdn1.md.home.com, @home network, 450 Broadway Street, Redwood City, CA 94063, US
024.018.229.006	cc279722-b.ebnsk1.nj.home.com, @home network, 450 Broadway Street, Redwood City, CA 94063, US
024.101.017.026	cr1031953-a.etob1.on.wave.home.com, @home network, 450 Broadway Street, Redwood City, CA 94063, US
024.182.002.226	c1093128-a.peoria1.il.home.com, @home network, 450 Broadway Street, Redwood City, CA 94063, US
024.189.216.251	ool-18bdd8fb.dyn.optonline.net, Optimum Online, 1111 Stewart Avenue, Woodbury, NY 11797, US
061.010.019.164	cm61-10-19-164.hkcable.com.hk, HK Cable TV Ltd, Cable Multi-Media, HK
061.013.106.035	c29.h061013106.is.net.tw, NATO International Corp., 9F-1, No.79, Sec. 1, Hsin Tai Wu Rd., Taipei, Taiwan, R.O.C
061.142.200.004	CHINANET Guangdong province network, Data Communication Division, China Telecom, CN
062.110.146.003	Servizio Informatica SRL, V. Dell, I- 40016 San Giorgio di Piano BO, Italy
063.023.174.061	1Cust61.tnt1.muskegon.mi.da.uu.net, UUNET Technologies, Inc., 3060 Williams Drive, Suite 601, Fairfax, va 22031, US
063.097.226.002	UUNET Technologies, Inc., 3060 Williams Drive, Suite 601, Fairfax, va 22031, US
064.158.160.082	unknown.Level3.net, Level 3 Communications, Inc.1025 Eldorado Boulevard, Broomfield, CO 80021, US
064.228.084.102	HSE-Toronto-ppp134825.sympatico.ca, Bell Nexxia, 350-181 Bay St., Toronto, ON M5J-2T3, CA
064.229.068.232	HSE-Toronto-ppp174833.sympatico.ca, Bell Nexxia, 350-181 Bay St., Toronto, ON M5J-2T3, CA
066.074.208.214	66-74-208-214.san.rr.com, ROADRUNNER-WEST, 13241 Woodland Park Road, Herndon, VA 20171, US
111.111.111.111	IANA Reserved Address
132.199.101.019	pc6734.physik.uni-regensburg.de, University of Regensburg, Universitaetsstrasse 31, Regensburg, 8400, DE

IP Address	Name/Location
132.229.131.040	pandora.debian.org, Rijks Universiteit Leiden, Niels Bohrweg 1, P.O. Box 9512, 2300 RA Leiden, NL
133.025.193.054	HOSEI-NET, Hosei University [JPNIC]
133.025.193.234	res-1b-193-234.k.hosei.ac.jp, HOSEI-NET, Hosei University [JPNIC]
158.075.057.004	hetman.loiv.torun.pl, Nicolaus Copernicus University, University Networking Technology Centre, PL
159.226.000.000/16	The Computer Network Center Chinese Academy of Sciences, P.O. Box 2704-10, Institute of Computing Technology Chinese Academy of Sciences, Beijing 100080, China, CN
167.206.254.176	Cablevision Systems Corp., One Media Crossways, Woodbury, NY 11797, US
168.070.145.077	pcd120077.netvigator.com, Hongkong Telecom, Hongkong Telecom Tower, Taikoo Place, 979 King's Road, Quarry Bay, HK
193.226.113.248	248.valahia.ro, InterComp, Bucharest, ROMANIA, RO
193.251.010.016	ANice-101-2-1-16.abo.wanadoo.fr, France Telecom IP2000 ADSL BAS, BAS for services FTI-1 and FTI-2, FR
195.082.167.059	Instytut Podstaw Inzynierii Srodowiska Polskiej Akademii Nauk, Zabrze, Poland, PL
195.179.000.028	is1.blocksberg.com, Baumm + Baumm Produktentwicklung, Olgastr. 9, D-80636 Muenchen, DE
195.222.189.075	East View Publication, MOSCOW INFORMATION AGENCY, 6/3 Azovskaia str, 113149 Moscow, Russia, RU
199.183.024.194	vger.kernel.org, ICG NetAhead, Inc. (NET-ICG-BLK-BLK4-C), 532 Race St., San Jose, CA 91526, US
200.206.165.019	200-206-165-19.dsl.telesp.net.br, Comite Gestor da Internet no Brasil, R. Pio XI, 1500, Sao Paulo, SP 05468-901, BR
202.063.219.126	nszx104.137.szptt.net.cn, CubeXS Private Limited, Internet Service Provider, Data Entry, Software House, 310-311 Kassam Court, B.C. 9, Block 5, Clifton, Karachi, Pakistan, PK
202.104.139.195	Topearch Printed Circuits (Chinanet - Guangdong province network)
202.224.218.044	InfoSphere (NTT PC Communications, Inc.)
203.075.048.252	CHTD, Chunghwa Telecom Co.,Ltd., Data-Bldg.6F, No.21, Sec.21, Hsin-Yi Rd., Taipei Taiwan 100, TW
204.167.220.253	glahb.theassociates.com, The Associates (NETBLK-ASSOCIATES-220-21), 300 E Carpenter Fwy 3rd flr, Irving, TX 75062, US
205.188.153.099	fes-d003.icq.aol.com, America Online, Inc, 22080 Pacific Blvd, Sterling, VA 20166, US
205.188.233.121	g2lb4.spinner.com, America Online, Inc, 22080 Pacific Blvd, Sterling, VA 20166, US
205.188.233.153	g2lb5.spinner.com, America Online, Inc, 22080 Pacific Blvd, Sterling, VA 20166, US
205.188.233.185	g2lb6.spinner.com, America Online, Inc, 22080 Pacific Blvd, Sterling, VA 20166, US

IP Address	Name/Location
205.188.244.121	g2lb2.spinner.com, America Online, Inc, 22080 Pacific Blvd, Sterling, VA 20166, US
205.188.246.121	g2lb3.spinner.com, America Online, Inc, 22080 Pacific Blvd, Sterling, VA 20166, US
209.014.208.091	91-pool1.ras10.floca.tii-dial.net, AGIS (NETBLK-AGIS-BLK10), 3601 Pelham Road, Dearborn, MI 48124, US
209.181.206.099	dialupA99.cdrr.uswest.net, U S WEST - Interact Internet Services, 600 Stinson Blvd NE, Minneapolis, MN 55413, US
210.077.146.001	A3Dial-Net (Beijing, China – An ISP offering dialup service)
210.077.146.033	A3Dial-Net (Beijing, China – An ISP offering dialup service)
210.103.058.065	Gido Elementary School, Kyonggi, 894 Madudong Koyangsiilsanku, 412-290, Korea
210.223.052.151	Kermonet, Kyonggi, 1471-4 Kermo-Dong Siheung, Korea
210.241.238.230	CHTD, Chunghwa Telecom Co.,Ltd., Data-Bldg.6F, No.21, Sec.21, Hsin-Yi Rd., Taipei Taiwan 100, TW
211.046.039.194	Soksong Elementary School, CHUNGNAM, 28-2 Junpyung-ri Jungan-myun Kongju-city, Korea
211.079.076.065	Savecom International Inc., 3F, No.8, 19 Lane, Xing-Zheng Street, Shindian City, Taipei Taiwan 231, TW
211.100.112.190	CHINATDT, Dial UP User IP Pool, No. 1 Beishatan Deshengmen Wai, Beijing, CN
211.114.044.002	Seoul Wonchon Elementary School, Seoul, 21 Banpo-Dong Secho-Ku, Korea
211.120.040.002	parco-online.com, PARCOCITY, Parco-City Co., Ltd., Japan
211.135.120.218	zaqd38778da.zaq.ne.jp, OCT-NET, Osaka CableTV Corp., Japan
211.180.236.194	Chung Woo Design, 494-85 Yongkang-dong Mapo-gu, 121-070, Seoul, Korea
211.217.077.163	Korea Telecom, 128-9 Youngundong Chongroku, 110-460, Seoul, Korea
211.240.028.066	ITBusiness, 202 Namkang B/D, 692-3 Daerim3 Dong, Youngdeungpo Gu, 150-073, Seoul, Korea
212.046.064.180	has.a.pet.lab-rat.co.uk, Mirage Networking Ltd (ISP based in London GB), c/o Grid9 Internet Solutions Ltd, Scottish Provident House, HA1 1BX Harrow, United Kingdom
212.179.000.000/16	ISDN Net Ltd., Bezeq International, 40 Hashakham St., Petakh Tiqwah, Israel
212.209.158.149	Semko AB, Torshamnsgatan 43, Box 1103, 164 22 Kista, Sweden
212.227.251.013	placetobee-portale.de, DE-SCHLUND-980910, Schlund+Partner GmbH & Co., PROVIDER, DE
213.023.045.252	dsl-213-023-045-252.arcor-ip.net, ARCOR-IP-NET2, Mannesmann Arcor AG & Co, Koelner Str. 5, D-65760 Eschborn, Germany, DE
213.039.073.250	SETE SA, Chemin Des Tuileries 3- 5, 1293 Bellevue, CH
213.046.030.084	d30084.upc-d.chello.nl, TK-EDE-CABLE, Chello Almere, cablemodems block 8, NL

IP Address	Name/Location
213.116.114.212	1Cust212.tnt10.rtm1.nl.uu.net, UUNET-DAN-NL, c/o Internet House, 332 Science Park, Cambridge, CB4 4BZ, UK
213.116.168.124	1Cust124.tnt37.rtm1.nl.uu.net, UUNET-DAN-NL, c/o Internet House, 332 Science Park, Cambridge, CB4 4BZ, UK
213.117.006.207	1Cust207.tnt44.rtm1.nl.uu.net, UUNET-DAN-NL, c/o Internet House, 332 Science Park, Cambridge, CB4 4BZ, UK
216.004.030.025	Business Internet Inc., 3625 Queen Palm Drive Tampa, FL 33619 US
216.150.152.145	wiredforlife5.spyral.net, CUBE Computer Corporation, 11 Skyline Dr., Hawthorne, NY 10532, US
216.235.163.151	ideaone-151.itgdata.net, Basin Electric Power Cooperative, 1717 East Interstate Avenue, Bismarck, ND 58503, US
216.235.163.163	ideaone-163.itgdata.net, Basin Electric Power Cooperative, 1717 East Interstate Avenue, Bismarck, ND 58503, US
217.057.019.030	CDC COMPUTER DATA CONTROL, via Zamenhoff, 430, I- 36100 Vicenza VI, Italy
225.130.160.002	Multicast Networks, Internet Assigned Numbers Authority, 4676 Admiralty Way, Suite 330, Marina del Rey, CA 90292-6695, US
225.130.160.003	Multicast Networks, Internet Assigned Numbers Authority, 4676 Admiralty Way, Suite 330, Marina del Rey, CA 90292-6695, US