Jackie Gough
GIAC Certified Intrusion Analyst Practical
Version 4.0
December 2004

Contents

Conventions Used in This Document

Heading 1 – 16pt Bold Arial, used to define main section titles
Heading 2 – 14pt Bold Arial, used to define main paragraph titles

Heading 3 – 12pt Bold Arial, used to define minor paragraph headings
Normal – 12pt Arial, used for normal text of the document
`Code Listing` – 10pt Courier New, is used for extract of code i.e. snort rules, tcpdump filters or packet captures
`Code Description` – 12pt Courier New, is used when breaking down a rule or packet capture into plain English, also used when giving examples of commands used.
References [#] – Where # is a number, this corresponds to the list of references held near the end of this paper on page 23
Coloured text is used to aid in the deciphering of rules and packets.

Abstract Overview

The detects analysed in this document are taken from one days worth of logs (2002.10.10)[1] from the SANS Internet Storm Center[2]. The addresses of the internal network have already been sanitised by SANS before the logs were posted.
For the purpose of this paper the internal network will be treated as if it were a Universities network.
Contained within this paper are an overview of all the detects from the log file noted above, focusing on 3 of the more serious or anomalous events. Further to that, insights into possible areas for compromise and defensive recommendations are made. Concluding with a list of references, which were used and influenced the writing this paper.

Executive Summary

A broad description of the bulk of the packets analysed would be that is was mostly reconnaissance traffic. This came in the form of probes and perimeter tests.
The network as it stands at the minute appears very open, there is little or no protection and it is only a matter of time before there is a serious and potentially embarrassing compromise of the networks critical infrastructure.

This may be a harsh description however, from my analysis it is truthful. During the 24 hour period that was analysed there were 14 different events, 5 of these were scans, 2 were intrusion attempts and 7 were classed as miscellaneous activity. There were no serious compromises made during this time, this is not to say that there are no compromised machines on the network, on the contrary there may be many nevertheless there is no concrete evidence to confirm this theory. Further, long term and full time monitoring is required to ensure that the network is kept safe and free from intrusion and compromise.

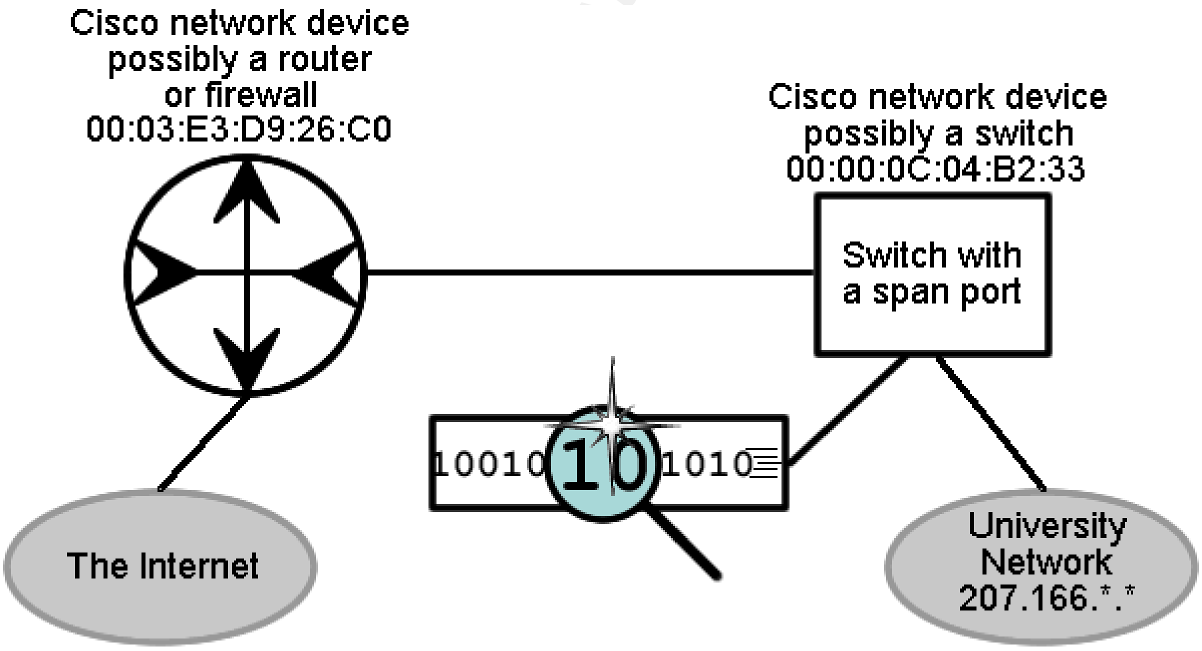
The scans were mainly from the United States with 1 limited scan from Taiwan and consisted of a search for proxies which is classified as noisy reconnaissance traffic, possibly designed or crafted to distract the intrusion detection analyst from more serious events happening.

The intrusion attempts were only thwarted by the attackers' lack of knowledge, should they have undertaken good reconnaissance and research we could be looking at a serious event, possible mass compromise of sensitive systems and embarrassment to the university.

Defensive recommendations are on page 20 near the end of this document and mainly consist of relatively basic and easy to implement actions to tune defences. Loosely, this means separating your outside facing systems i.e. you web server and external mail server from your sensitive/working inside network. Consequently should the need arise you could protect yourself by severing the outer network connection.

Network Topology

There is no tangible proof that any network countermeasures exist. We could assume that the traffic was captured from inside a Demilitarised Zone and that the traffic is NAT'd (Network Address Translated) but we cannot and should not, make any assumptions about the security of the network, although the existence of NAT seems likely and is partially backed up by the outward bound packets only coming from 2 IP addresses (207.166.87.157 & 207.166.40). However, for the events analysed the NATing would have to be static; this is improbable, due to the massive overhead of manually mapping all of the internal IP addresses. Accordingly when calculating severity using the severity = (criticality + lethality) – (system countermeasures + network countermeasures) the network countermeasures will be set to 1, as we are unsure if any network countermeasures exist.



Using the end point mapping facility in the protocol analyser Ethereal[3] we can ascertain that the flow from these routers is bi-directional Brett Hutley[4] posted to the SANS intrusions list[5] with the same findings.

An Overview of the Detects

All these events were analysed by Snort 2.2

> *Snort Version 2.2.0-ODBC-MySQL-FlexRESP-WIN32 (Build 30) By Martin Roesch, 1.7-WIN32 Port By Michael Davis, 1.8 - 2.x WIN32 Port By Chris Reid*

Using the latest-stable ruleset from 26th August 2004, on Windows 2000 Pro platform with service pack 4.

A span port or hub, the location the traffic was captured from.

🟨	This colour denotes an Event that is one of the three detects in this document.

	Event	Hits
1	SCAN Squid Proxy attempt	5,992
2	SCAN Proxy Port 8080 attempt	5,988
3	(spp_stream4) Possible RETRANSMISSION detection	65
4	SCAN nmap TCP	47
5	BACKDOOR Q access	46
6	(http_inspect) NON-RFC HTTP DELIMITER	37
7	(spp_stream4) STEALTH ACTIVITY (Vecna scan) detection	27
8	(http_inspect) BARE BYTE UNICODE ENCODING	24
9	SCAN SOCKS Proxy attempt	16
10	BAD-TRAFFIC tcp port 0 traffic	16
11	(snort_decoder) WARNING: TCP Data Offset is less than 5!	5
12	BAD-TRAFFIC ip reserved bit set	2
13	(http_inspect) WEBROOT DIRECTORY TRAVERSAL	1
14	(http_inspect) OVERSIZE REQUEST-URI DIRECTORY	1
	Total	**12,267**

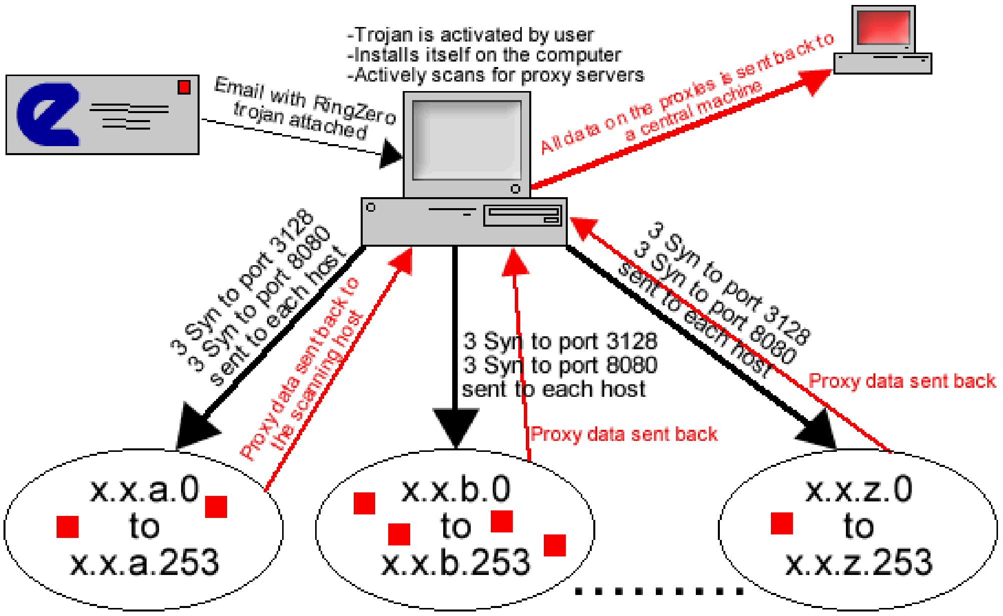
Event #	Brief Description	Notes
1	Both these events are triggered by one reason – RingZero, this is detailed in Detect 1.	This event is as a result of an infected / compromised box on the external network.
2		
3	The source of 64 of these events is 207.166.87.157 (the internal network) and is possibly caused by the obfuscation of data.	This event could be proved or disproved by analysing the data before it has been sanitised.
4	This event is a result of enabling some of the deleted rules to see what they would trigger. The rule was deleted due to the fact the NMAP is not as stupid as it once was.	Considered a non-event

5	Whilst the packets that triggered this event meet all the criteria for being Backdoor Q; it is not – Detect 2 investigates.	The source of these detects (255.255.255.255) is very obviously spoofed.
6	Two IPs 192.77.15.39 35 hits & 61.169.232.66 1 hit. Single target of 207.166.87.40 which is most likely a web server. Signature was stimulated by the ../ combination, which if you use certain commercial web site design software is common place.	A signature that needs the analyst to intervene before it can be written off.
7	The first packet was never acknowledged and so was continually re-transmitted triggering the vecna scan signature. A vecna scan is one where all the packets have illegal flags set, such as push on its' own.[6]	The initial packet had only the push flag set and was thus crafted.
8	Bare byte Unicode encoding is on a normal day a bit noisy and can be caused by normal web browsing.	One of the signatures that needs analyst intervention on each event in order to prove whether it is a false positive or not.
9	A brief scan for a socks proxy,	Exactly what it says on the tin.
10	Crafted – port 0 is a port that can be used against you. Port 0 is listed by IANA as reserved and is not really an official port, but it can however, be used in crafted packets as a source or destination port.	Crafted packets used for reconnaissance.
11	Possibly caused by collision/crafting or corruption	
12	An interesting event as the RFC for the IP security "evil" bit was not released until April 2003. Detect 3 investigates.	
13	This was an attempt to access a normally restricted area on a windows machine.	There was no response from the target – either wrong OS or blocked by firewall or ACLs.
14	An extra long URL – just like you would find in a complex search URL.	False Positive.

Detect 1 – Multiple Proxy Scans

Description of the Detect

This detect is the amalgamation of SCAN Squid Proxy Attempt & SCAN Proxy Port 8080 attempt. Both proxy scans emanate from a single host, 66.123.116.234. The two scans to 1988 IPs on the Internal Network were made in 10 minutes, the programmer of RingZero was either not afraid of the compromised boxes being discovered by intrusion analysts or has overlooked this element of network defence. The RingZero host did make itself the top talker in this capture file.
The RingZero trojan affects win32 based operating systems and comes as an executable email attachment, the user has to activate this, it is not self propagating. After installation the trojan actively and noisily searches for proxy servers (Squid on port 3128 and HTTP proxy on port 8080). Data is gathered from any responding targets and is stored and sent back to a central host as designated in the trojan source code.



The 🟥 signifies a proxy with either port 3128 or 8080 open.

Reason This Detect Was Selected

This detect was selected due to the fact that the signatures in question require the interpretation by an analyst and cannot be taken at face value; they must be investigated in order to find the true purpose behind them. If this scan has succeeded then the data it gathered, could be used to hide further intrusion attempts not only at this site but at other sites/networks as well.

Detect Was Generated By

The detects were generated by Snort 2.2

> *Snort Version 2.2.0-ODBC-MySQL-FlexRESP-WIN32 (Build 30) By Martin Roesch, 1.7-WIN32 Port By Michael Davis, 1.8 - 2.x WIN32 Port By Chris Reid*

Using the latest-stable ruleset from 26th August 2004, on Windows 2000 Pro platform with service pack 4. The deleted rules enabled; this is so that a historical representation of the data could be more accurate.

The rule `alert tcp $EXTERNAL_NET any -> $HOME_NET 3128 (msg:"SCAN Squid Proxy attempt"; flags:S,12; classtype:attempted-recon; sid:618; rev:4;)` triggered on an external network address with any source port going to any of the home network addresses on port 3128 with the Syn flag set and gave this full alert

```
[**] SCAN Squid Proxy attempt [**]
11/10-20:07:37.516507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x3E
```

```
66.123.116.234:3130 -> 207.166.43.247:3128 TCP TTL:108 TOS:0x0 ID:43563
IpLen:20 DgmLen:48 DF
******S* Seq: 0x1E69D7B9  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+

[**] SCAN Squid Proxy attempt [**]
11/10-20:07:40.656507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x3E
66.123.116.234:3130 -> 207.166.43.247:3128 TCP TTL:108 TOS:0x0 ID:43639
IpLen:20 DgmLen:48 DF
******S* Seq: 0x1E69D7B9  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+

[**] SCAN Squid Proxy attempt [**]
11/10-20:07:47.216507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x3E
66.123.116.234:3130 -> 207.166.43.247:3128 TCP TTL:108 TOS:0x0 ID:43857
IpLen:20 DgmLen:48 DF
******S* Seq: 0x1E69D7B9  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+
```

The rule `alert tcp $EXTERNAL_NET any -> $HOME_NET 8080 (msg:"SCAN Proxy Port 8080 attempt"; flags:S,12; flow:stateless; classtype:attempted-recon; sid:620; rev:10;)` triggered on an external network address and any source port with a destination address of the home network on port 8080 with the Syn flag set and below is a sample of a scan on one host.

```
[**] SCAN Proxy Port 8080 attempt [**]
11/10-20:05:31.456507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x3E
66.123.116.234:1777 -> 207.166.42.59:8080 TCP TTL:109 TOS:0x0 ID:39406
IpLen:20 DgmLen:48 DF
******S* Seq: 0x18C9B3E3  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+

[**] SCAN Proxy Port 8080 attempt [**]
11/10-20:05:34.656507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x3E
66.123.116.234:1777 -> 207.166.42.59:8080 TCP TTL:109 TOS:0x0 ID:39491
IpLen:20 DgmLen:48 DF
******S* Seq: 0x18C9B3E3  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+

[**] SCAN Proxy Port 8080 attempt [**]
11/10-20:05:41.206507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x3E
66.123.116.234:1777 -> 207.166.42.59:8080 TCP TTL:109 TOS:0x0 ID:39707
IpLen:20 DgmLen:48 DF
******S* Seq: 0x18C9B3E3  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

=+

From the two alert extracts above we can see that there are two different TTL (Time To Live). They only vary by one in all of the alerts, this is due to the varying routes taken by the packets.

Probability that the source address was spoofed

The probability of the address being spoofed is nil; the purpose of this part of the trojan is to gather information about proxies it is a reconnaissance scan, or at least attempted reconnaissance. The data has to be gathered and then passed onto a central host for storage and later for harvesting by the author/propagator of the trojan. Spoofing would nullify any reason for activating this scan.

The Attack Mechanism

In this detect the attacker has succeeded in infecting the source host 66.123.116.234 which is outside of our network. It is unknown if the scan yielded any useful information, as only anomalous packets are contained in the capture that was analysed. The trojan is scanning for proxy servers (Squid 3128 and HTTP 8080) and most likely did not stop at scanning the internal network. It probably started with a low network block and will continue well above the universities internal network range targeting other networks.

Correlations

RingZero was first discovered in October 1999 and is noted in Symantec's Security Response website [7]. They released a virus definitions update on the 26th Oct 1999 to protect against this trojan.
McAfee have documented on their website [8] that the RingZero.gen trojan was discovered on the 24th August 1999 but the description was not added until the 14th September 1999. The trojan characteristics concur with my research and findings apart from the scanning element which was documented by F-Secure [9] this was an invaluable source on information and reassurance that I was on the right track. The RingZero scanning was also noted by Susan Kovacevich in a posting to DShield on the 17 Oct 2002 [10] and also in her posted GCIA practical [11]

Evidence of active targeting

There is no evidence of active targeting in this detect; the trojan mearly scans network blocks indiscriminately and has no regard for hiding it's presence or covering it's tracks.

Severity

The scale for measuring criticality, lethality, system and network countermeasures is a lot like the F(factor) scale for tornados running from 1-really tame wind to 5-your flying along with your house, car and everything you own. In the case of criticality 5 would be your most critical systems, for example, the domain controllers &

authentication systems, network management systems and any databases or servers that are specifically crucial to your network functioning. A 5 in lethality would be just like an F5 tornado, that is to say it is the most destructive thing you will ever see in the wild. On the flip side a 5 in system countermeasures would be the most security available for that system, up to date Anti-Virus a properly configured personal firewall and an operating system that is fully patched. A 5 in network countermeasures would be routers that are properly configured for that network and its associated traffic, gateway servers with up to date Anti-Virus and a properly configured firewall.

Severity = (criticality + lethality) – (system countermeasures + network countermeasures)
The severity, is the calculation of the criticality of the system being targeted, plus the lethality of the attack, if it succeeded, minus any system countermeasures in place plus any network countermeasures in place.

Criticality = 4 – As little is known about the specific servers on the network, we have erred on the side of caution and guessed a 4 for possible domain controllers, databases and servers containing private/sensitive personnel data.
Lethality = 1 – the attack is not damaging to systems, the systems are just abused by the attacker.

System countermeasures = 1 – there are no system countermeasures that I can detect from this traffic.
Network countermeasures = 2 – there is no network countermeasures, other than the NAT setup that I can detect from this traffic.

(4+1) - (1+2) = Severity of 2, a reconnaissance scan with possible embarrassing consequences; this is mainly due to the lack of information on system and network countermeasures.

Detect 2 – Backdoor Q access… or is it?

Description of Detection

Backdoor Q is a trojan that allows the attacker to signal the target machine to open a port, that the attacker can use for many purposes including further compromise of the target or as a relay for attacking others. There was 46 events in the 24 hours analysed.

Reason This Detect Was Selected

High numbers may not in themselves reveal an attack but the odd payload in the packets warrant a much closer look. I do not believe that this is a straight cut Q detection.

Detect Was Generated By

The detects were generated by Snort 2.2

> *Snort Version 2.2.0-ODBC-MySQL-FlexRESP-WIN32 (Build 30) By Martin Roesch, 1.7-WIN32 Port By Michael Davis, 1.8 - 2.x WIN32 Port By Chris Reid*

Using the latest-stable ruleset from 26th August 2004, on Windows 2000 Pro platform with service pack 4.

The rule that detected this event was `alert tcp 255.255.255.0/24 any -> $HOME_NET any (msg:"BACKDOOR Q access"; dsize:>1; flags:A+; flow:stateless; reference:arachnids,203; classtype:misc-activity; sid:184; rev:6;)`
This translates to TCP traffic from IP address 255.255.255.0/24 any port going to the Home Network on any port with a payload of greater than 1 byte with the Ack plus any other flags set.

Probability That The Source Address Was Spoofed

The source address is undoubtedly spoofed; the address should not be routable, if the router is configured correctly/securely (but is not in our case).

The Attack Mechanism

I do not believe that this is a detection of Backdoor Q access even though it fills all the correct criteria; I believe that this is the detection of something more sinister. This is probably not a response to something, but it could be. It is more likely the stimulus for a dormant malicious code.

Below is a snipped section of some of the alerts.

```
[**] BACKDOOR Q access [**]
11/10-02:59:13.066507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x3C
255.255.255.255:31337 -> 207.166.98.5:515 TCP TTL:15 TOS:0x0 ID:0
IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+

[**] BACKDOOR Q access [**]
11/10-03:23:45.266507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x3C
255.255.255.255:31337 -> 207.166.112.119:515 TCP TTL:15 TOS:0x0 ID:0
IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+
```

<snip>

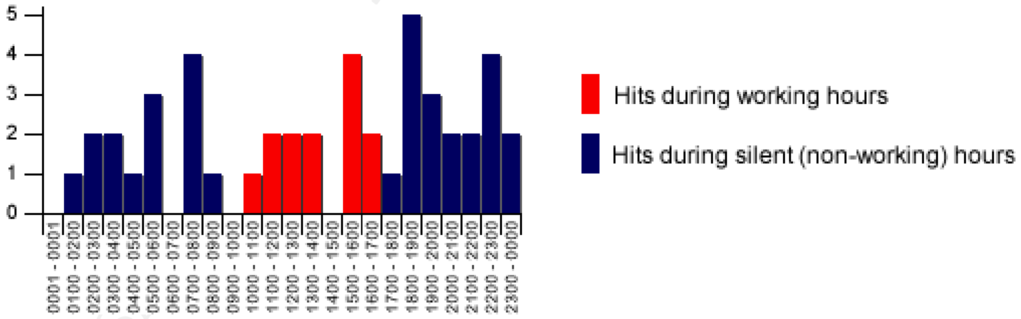
```
[**] BACKDOOR Q access [**]
11/10-00:48:33.406507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x3C
```

```
255.255.255.255:31337 -> 207.166.101.174:515 TCP TTL:15 TOS:0x0 ID:0
IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+

00 00 0c 04 b2 33 00 03  e3 d9 26 c0 08 00 45 00
00 2b 00 00 00 00 0f 06  51 75 ff ff ff ff cf a6
d4 fc 7a 69 02 03 00 00  00 00 00 00 00 00 50 14
00 00 06 9d 00 00 63 6b  6f 00 00 00
```

As you can see from this colourised alert and packet above of the alleged backdoor Q. The source port of 31337 is odd, it is a known sub-seven listening port however, it stands out from the crowd and is not stealthy in the slightest. This choice of source port along with the source address of 255.255.255.255, is crafted, although it seems a bizarre combination to be chosen because it does not blend in with other normal traffic. The destination port of 515 is typically or commonly linked with unix printing [38] but anything could be set up on this port. Sequence and acknowledgement numbers are set to 0. Both the Ack (Acknowledgement) and Rst (Reset) flags were set. Window size is 0, if this was designed to illicit a response, then it would fail; although "Q" does not always require a response. Not only due to the ack rst, no response from a windows system, the window size of 0 would prohibit the target from responding. The data content of cko is present in all the packets from 255.255.255.255, it is unknown at this time why the three letter combination is present in the data.

This event is one to many and may appear as a scan at first, but we should be careful not to disregard this event as benign reconnaissance or probing.
The overall timing for the 46 resets, was evenly spread over 24 hours; it is like the perpetrator of these events wanted to blend in with the normal network traffic.
Below is a graph of the timings of this event over the 24hours examined.



Correlations

Al Maslowski-Yerges[12] concurs with my thoughts that it may be a tool and that it should not be written off a harmless traffic, also mentioned is the possibility of this traffic being linked with an IRC server[13] and a subsequent worm.
I do not side with the idea that this traffic is harmless [14] I believe this detect to be much more sinister.
Trenton Riddle[15] used similar traffic in his GCIA paper and agrees with the distinguishing fact, that these packets, were very obviously crafted, but disagrees on

my statement that the attacker may be attempting to be stealthy which was also noted in a mailing list discussion from the Security Focus discussion forum [14] that stated that the traffic could be trying to evade detection [16]
Peter Storms' GCIA practical [17] analysis sits along side with my own and refers to Kerry Longs' GCIA practical [18] and mailing list discussion from Bob Fritton [19] that introduces the suggestion that it may be designed to wake sleeping trojans. Mr Storms' GCIA paper and his correlations feed a possible conspiracy theory on the source and purpose of this traffic.

Evidence of Active Targeting

As the sequence of events is not sequential and each target is separated by a non-exact amount of time, I believe that this is not random and may be specifically targeted using recon data from an earlier (perhaps much earlier) mission.

Severity

As there is a suspicion that this may have been targeted (it was most likely not random) and as I am unsure off exact network topology and the assets it holds, I will have to put Criticality at 3.
Lethality must be a 2, this attack targets TCP port 515, which is commonly associated with unix printers [38], which is not a high profile critical port.
As with detect 1 there is no evidence of system countermeasures this will be set to 1 and as the network countermeasures have already let in an illegal address, I have to set this at 1.

So (3+2)-(1+1)= 3, this is of relatively medium to low severity, although it should not be disregarded.

Detect 3 BAD-TRAFFIC IP Reserved Bit Set

Description of Detect

The reserved IP bit is not in itself an attack, although it could be a stimulus for something that needs activation, perhaps a trojan or a zombie box, on the other side it could be reconnaissance or it could be some sort of experiment just to see what happens.

Reason This Detect Was Selected

The date of this detect was 10th November 2002 which is 6 months before the funny RFC 3514 The Security Flag in the IPv4 Header [20]. I believe at this stage that these two packets were crafted and only two were sent in order to keep the source from appearing in the top ten of anything.

Detect Was Generated By

The detects were generated by Snort 2.2

> *Snort Version 2.2.0-ODBC-MySQL-FlexRESP-WIN32 (Build 30) By Martin Roesch, 1.7-WIN32 Port By Michael Davis, 1.8 - 2.x WIN32 Port By Chris Reid*

Using the latest-stable ruleset from 26th August 2004, on Windows 2000 Pro platform with service pack 4.

The following rule was triggered.

```
alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"BAD-TRAFFIC
ip reserved bit set"; fragbits:R; classtype:misc-activity;
sid:523; rev:5;)
```

Lets break down the rule and get to know it a little, `alert` = when this rule matches then alert it, `ip` = we are looking for IP, `$EXTERNAL_NET` = from the external network (a variable set up in snort.conf, see Annex A for the conf file), `any` = a connection from any source port, `->` = the direction of traffic, `$HOME_NET` = any home network IP (just like the external_net, it is a variable set in snort.conf, see Annex A for the conf file), `any` = a connection to any destination port, `(msg:"BAD-TRAFFIC ip reserved bit set";` = the message that is displayed when the alert is triggered and is displayed in our log as `[**] BAD-TRAFFIC ip reserved bit set [**]`, `fragbits:R;` = the bit we are looking at; this is the evil bit, `classtype:misc-activity;` = this is the classification of the alert (miscellaneous activity), `sid:523;` = is the snort ID and can be used to search snort.org[22] for additional data, `rev:5;)` = that this is revision 5 of this rule.

The real meat of this rule it the `fragbits:R;` this is the part that is number one in our pursuit of these potentially crafted packets.

The rule triggered the following two alerts.

```
[**] BAD-TRAFFIC ip reserved bit set [**]
11/10-19:26:12.136507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x3C
200.200.200.1 -> 207.166.178.227 TCP TTL:242 TOS:0x0 ID:0 IpLen:20
DgmLen:40 RB
Frag Offset: 0x0864   Frag Size: 0x0014
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+

[**] BAD-TRAFFIC ip reserved bit set [**]
11/10-19:46:13.636507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x3C
200.200.200.1 -> 207.166.103.105 TCP TTL:242 TOS:0x0 ID:0 IpLen:20
DgmLen:40 RB
Frag Offset: 0x0864   Frag Size: 0x0014
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+
```

We can also break these down and analyse them further rather than taking them on face value. The Internet Protocol Request for Comments, RFC 791[22] was invaluable in the description of this alert as well as interpretation of packets.

[**] BAD-TRAFFIC ip reserved bit set [**] is the message of the alert and can be set in the rule, see above for a breakdown of the rule.
11/10-19:46:13.636507 is the date 11/10 and time in hours, minutes, seconds and milliseconds 19:46:13.636507
0:3:E3:D9:26:C0 is the MAC address of the last hop that the packet encountered, in this case most likely a Cisco router, defined by the MAC (Media Access Control) address 00:03:E3.[53]
-> is the direction of the traffic, just like in the snort rule.
0:0:C:4:B2:33 is the MAC address of the next hop or the destination host, but in our case it is the next hop and it is also most likely a Cisco network device, defined by the MAC address 00:00:0C. [53]
type:0x800 means that we are looking at IP rather than the type being 0x806 which would be an ARP packet.
len:0x3C refers to the entire length of the captured packet which in these two instances is 60 bytes, this include the hardware header, the ip header and the embedded data.
200.200.200.1 is the source IP address
-> is the direction of traffic
207.166.103.105 is the destination IP address
TCP is the embedded protocol, defined in the IP header
TTL:242 is the Time To Live which in strict definition is the seconds that a packet has left to live, a more modern definition is the amount of hops the packet has left before it is discarded. The starting TTL for this packet was most likely 255 but it could have been crafted along with the rest of the packet.
TOS:0x0 is the Type Of Service which is set to zero, this translates to a routine packet with normal delay, normal throughput and normal reliability.
ID:0 this is the identification of the packet although an IP datagram can have an ID of zero it is highly unlikely, this could be another clue to packet crafting.
IpLen:20 is the length of the IP header which is a standard 20 bytes, this means that it is a normal IP header with no options (options would increase the IP header length and are padded with zeros (NoOps) to end on an 32bit boundary.
DgmLen:40 the length of the rest of the Datagram (Dgm) which is 40 bytes.
Frag Offset: 0x0864 the point at which this fragment is to be joined onto the rest of the fragments (if there were any).
Frag Size: 0x0014 this is the size of the fragment that will join onto the rest at the point of fragment offset.
That was a brief breakdown of a Snort Alert.

These are the two packets that caused the rule to trigger the alerts.

```
00 00 0c 04 b2 33 00 03  e3 d9 26 c0 08 00 45 00
00 28 00 00 88 64 f2 06  77 61 c8 c8 c8 01 cf a6
b2 e3 0e bf 00 50 4e d1  d5 8c 4e d1 d5 8c 00 04
00 00 df 0b 00 00 00 00  00 00 00 00
```

```
00 00 0c 04 b2 33 00 03  e3 d9 26 c0 08 00 45 00
00 28 00 00 88 64 f2 06  c2 db c8 c8 c8 01 cf a6
67 69 10 87 00 50 4e e4  2b 10 4e e4 2b 10 00 04
00 00 7d 91 00 00 00 00  00 00 00 00
```

A brief analysis of one of the packets –
The Ethernet Header – this contains the source MAC (`00 00 0c 04 b2 33`) and destination MAC (`e3 d9 26 c0 08 00`) as defined in the alert analysis.
The IP Header – `4` = IP version 4. `5` = the size of the IP header (note this number has be multiplied by 4) in this case 20 bytes. `00` = Type of service as defined in the alert analysis. `0028` which is 40 bytes and is the length of the IP datagram minus the IP header. `0000` is the IP identification. `8864` has to be broken down into binary to understand what flags are set, 1000 1000 0110 0100, the first 3 bits, 1 0 0, of this are the reserved bit, do not fragment and more fragments; the reserved bit is set and the others are not; the remaining binary 0 1000 0110 0100 is 864 and this is the fragment offset, which should be multiplied by 8 to find the true fragment offset, the lack of the more fragments bit being set would normally define this packet as being the last one. `f2` is the TTL of 242. `06` is the embedded protocol TCP meaning it is a TCP header that will follow this one. `c2db` is the checksum and it is incorrect , it has been obfuscated to protect any IPs (IPs were also obfuscated) from being reverse engineered. `c8 c8 c8 01` is the source IP of 200.200.200.1, `cf a6 67 69` is the destination IP address of 207.166.103.105.
The TCP Header – `1087` is the source port of 4231. `0050` is the destination port of 80 (commonly associated with HTTP). `4ee4 2b10` is the sequence number; this can be used for finding out if a packet was received and acknowledged. `4ee4 2b10` is the acknowledgement number and can be used to see what the last packet received by the host was (in this case the sequence and acknowledgement numbers are the same, more evidence towards packet crafting). `0004` contains the TCP header length and any flags set, in this case the header length is zero (which at this point it can't be because we are already 14 bytes into it) and that the reset flag is set, this is highly suspicious seeing as it was not resetting anything, or at least any packets that were captured in the data analysed. `0000` defines the window size of zero – which means that even if some sort of response was required it would not be able to send it. `7d91` is the TCP header checksum. `0000` is the urgent pointer which is zero. The above deciphering of the TCP header is going on the assumption that it would normally be a 20 byte header and not the 0 that it was crafted to say.

Probability That The Source Address Was Spoofed

I doubt that the source 200.200.200.1 was spoofed in this case however, it may have been; any response that these two packets stimulate (which may occur at a later date, thus no response is shown here) may be transmitted back to a different machine, perhaps one under the attackers control. So it would be wrong to say that this is categorically one way or another spoofed or not. There is not enough data to swing it either way.

Attack Mechanism

The long term consequences of this attack are unknown at this time, just because there was no immediate response, that triggered an alert, from the destination address does not mean that nothing has happened. The attack itself may have been thwarted by a firewall, although this is unconfirmed.

This attack is most likely recon, but done in a very crafty and stealthy way trying to fly below the radar. As it is recon the source address is most likely not spoofed or if it is then it is also a machine under the attackers control or at least one that the attacker has access to in order to check logs or report for replies to the stealth probes.

The attacker would carefully and most likely take time over crafting a special packet so that it would not bring too much attention. A likely crafting tool would be HPing2[23] or NetDude[24]. Attention to detail and care may have been taken over the first packet, but the second packet is virtually identical and that may show that the attacker is patient but not that patient and may have been eager to prove their 'leetness' to their peers.

I also have to note that even though the TTL is 242 and that may signify a Solaris 2.x machine [25] I believe that this has also been crafted in order to throw of an analyst in the event that these packets are captured and analysed, this is maybe a bit on the paranoid side but may be a possibility. Another clue to the likelihood of the packets being crafted is the setting of the fragment offset, this coupled with the lack of the more fragments flag being set may be a sign that this is the last fragment. However, we have not seen any packets that go with these or at least they have not been captured in the file that was analysed.

Correlations

Ron Shuck posted LOGS: GIAC GCIA Version 3.3 Practical Detect on Tue Feb 11 02:14:53 UTC 2003[26], this was describing his Detect 2: BAD TRAFFIC IP reserved bit set. Mr Shucks' data was gathered in a similar manner and even though he was using a different raw data download the output is of the same pattern and thus he has drawn similar conclusions as myself. This was also confirmed in his submitted paper[27].

Evidence of Active Targeting

There have been only two instances of the IP reserved bit being set in this capture. I believe that these two targets were picked for a reason, the attacker knows something, or indeed, thinks that they know. The two hits were also performed 20 minutes and 1 second apart starting with 18:26:12 then at 18:46:13 this was possibly to try and avoid being caught by anything stateful, 20 minutes is a long time in cyberspace.

Severity

Criticality – I would set at 3 as the target machine is unknown, it could be a web

server the probe to port 80 may back this up but there is no further information and I believe setting criticality any lower may make this detect too trivial.
Lethality – As I believe that this may be a targeted attack and that the attacker wants something that it unknown to us however, this particular detect we did not see any other related activity so I set lethality at 2.
System Countermeasures – There is no concrete evidence of any system countermeasures so level set to 1.
Network Countermeasures – There is a slight possibility of a firewall protecting the target network, but there is no proof of this, so the level must be set to 1.

(Criticality + Lethality) – (System Countermeasures + Network Countermeasures) = Severity
(3+2) – (1+1) = 3 (this detect would warrant further and deeper investigation)

Network Statistics

Passive operating system fingerprinting using P0f[28] gave no definitive intelligence on the type of operating systems, other than to say that most of the systems were probably windows 98 or 2000/XP, this was due to the lack of full TCP streams in the capture, p0f works best with TCP streams.

The top 10s in the tables below were produced by Sawmill [29] and tweaked for clarity and presentation.

Notes	**Data**
1 is the source of the RingZero scans 2 web browsing 3 the source of the Backdoor Q detects 4 web browsing 5 retransmissions of one packet, due to the source not receiving an Ack 6 socks scan 7 4 x 4 packets to port 0 8 4 x 4 packets from and to port 80 9 more traffic from and to port 80 10 small squid scan (port 3128)	**Top 10 Source Hosts**

	Source host	Hits	Country[30]
1	66.123.116.234	11,974	US
2	207.166.87.157	89	Internal
3	255.255.255.255	46	Broadcast
4	192.77.15.39	36	US
5	148.64.32.21	27	US
6	204.188.161.87	16	US
7	211.47.255.20	16	KR*
8	12.108.43.5	16	US
9	140.128.251.21	10	TW
10	65.26.84.47	6	US

*This IP is currently held by the KRNIC a National Internet Register. The IP is being held for allocation to its member ISPs in the future.[31]

1 web traffic
2 web traffic
3 retransmissions of one packet, due to the source not receiving an Ack
4 web traffic
5 target of port 0 traffic
6 target of RingZero scan
7 target of RingZero scan
8 target of RingZero scan
9 target of RingZero scan
10 target of RingZero scan
6-10 do look like a pattern is developing, however, these are just some of the IPs that received a few more proxy/squid scan packets.

* Hitbox.com is an ad server company and hitbox can be used to track web page usage, domain and IP information.[32]

Top 10 Destination Hosts

	Destination host	Hits	Location
1	207.166.87.40	37	Internal Net
2	64.154.80.51	30	Hitbox.com*
3	207.166.48.62	27	Internal Net
4	64.154.80.45	19	Hitbox.com*
5	207.166.184.92	16	Internal Net
6	207.166.39.37	12	Internal Net
7	207.166.39.38	12	Internal Net
8	207.166.41.38	12	Internal Net
9	207.166.42.37	12	Internal Net
10	207.166.44.38	12	Internal Net

References [33] to [38] were used to gain the information contained in the port description column. These are what the port are commonly used for and by this I mean that anything can be assigned to use any port or indeed crafted to use any port. 1-2 are directly linked to detect 1 4 is directly linked to detect 2	**Top 10 Destination Ports**

	Destination port	Port Description	Hits
1	3128	Squid Proxy	5,992
2	8080	HTTP Proxy	5,988
3	80	HTTP	167
4	515	LPD – Line Printer Daemon or occasionally standing in for port 514 on cisco	46
5	1690	NG-UMDS	27
6	0	(Reserved by IANA) An illegal port, but it can be used.	21
7	1080	Socks Proxy	16
8	1214	KAZAA	6
9	53	DNS	2
10	(empty)	No port was detected by Snort due to the tcp header length being 0 – these two events are both had the IP reserved bit set as noted in Detect 3 on page 11. Both when deciphered by hand had a destination port of 80	2

Three Most Suspicious External Sources

There is a common theme in the three most suspicious external sources and that is that they are unassigned IPs. This aspect on its own makes them suspicious and warrants further investigation. However, we also have further anomalous/weird traffic to make these three all the more suspicious.

The three IPs in question are:-

255.255.255.255 – A broadcast address, this is the most suspicious external source and warrants much further investigation, perhaps by sniffing (using TCPDump[39] or Snort with a custom ruleset) on the distant side of the external router. This is of course permission and law allowing. When using Samspade.org[30] to look up more information on this IP we get an error informing us that it is reserved by IANA.org[40] this was also analysed in Detect 2 [41].

172.20.10.199 – An address reserved for private use, and that should not be seen on the internet, see IANA reference [42]

211.47.25.20 – Is not assigned, however it is reserved by KRNIC[43] (a bit like APNIC or ARIN for Korea) for assignment to a Korean ISP, in a way this is more suspicious than the other two due to the fact is looks so ordinary it just does not leap out as “evil” straight away.

The trouble with all of the above is that they are most likely the stimulus for something i.e. a sleeping trojan, virus or malware of some variety. They do not need a response sent back, it is “fire and forget”.

OS fingerprinting would be a mostly futile effort, due to there being no real destination for the packets to return to and as the packets have had the IP spoofed they may have been more cautious and crafted the rest of the packet in order to throw anyone examining the packets off the scent completely.

SamSpade [44] was used to discover the geographic location and other information about the IPs.

Correlations

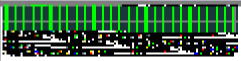
These correlations are in addition to those found within the 3 Network Detects Section.

The table below is the Block list from DShield [45] dated 30 Nov 04 it has been trimmed and slightly modified to fit the page. It features in this section because we could correlate the IPs listed contained in the table with the data that was analysed. This block list was also referenced in the Defensive Recommendation section in conjunction with security at routers and firewalls.

As you can see there are similar IPs to our data from the Network Statistics section [46], but the IPs analysed are not present in the current block list; this may be due to the analysed data being from November 2002 and being 2 years old at the time of the analysis.

Start	Net	Name	Country
69.3.132.0	/24	Covad Communications	US
69.142.126.0	/24		
211.239.150.0	/24	Korean Network Information	KR
66.144.164.0	/24	State of Ohio Network (NETBLK-NET-STATE-OHIO)	US
208.177.130.0	/24	Concentric Network Corporation (NETBLK-CONCENTRIC-BLK4)	US
68.94.8.0	/24	SBCInternetServices-Southwest	US
172.164.28.0	/24	America Online	US
80.171.64.0	/24	HanseNet Telekommunikation GmbH	DE
222.34.5.0	/24	CHINA RAILWAY TELECOMMUNICATIONS CENTER	CN
61.84.79.0	/24	Korean Information Network	KR
64.80.40.0	/24		
83.237.28.0	/24	ZAO MTU-Intel	RU
218.58.58.0	/24		
213.228.39.0	/24	Paris- France	FR
211.214.247.0	/24	Hanaro Telecom Inc	KR
219.179.220.0	/24	BB Technologies Corp.	JP
68.251.121.0	/24	AmeritechElectronicCommerce	US
61.134.45.0	/24	pingli county govment office	CN
217.233.234.0	/24	Deutsche Telekom AG	DE
221.170.249.0	/24	NEC Corporation	JP

This is the current top 10 targeted ports list from DShield[48]
As with the information above, the DShield top 10 targeted ports are not in common with the top 10 from the log file 2002.10.10. Attacks, expoits and abilites have evolved in the past two years and this could account for the lack of exact correlation.

Service Name	Port Number	Activity Past Month	Explanation
microsoft-ds	445		Win2k+ Server Message Block
epmap	135		DCE endpoint resolution
---	16990		
---	1026		
netbios-ssn	139		NETBIOS Session Service

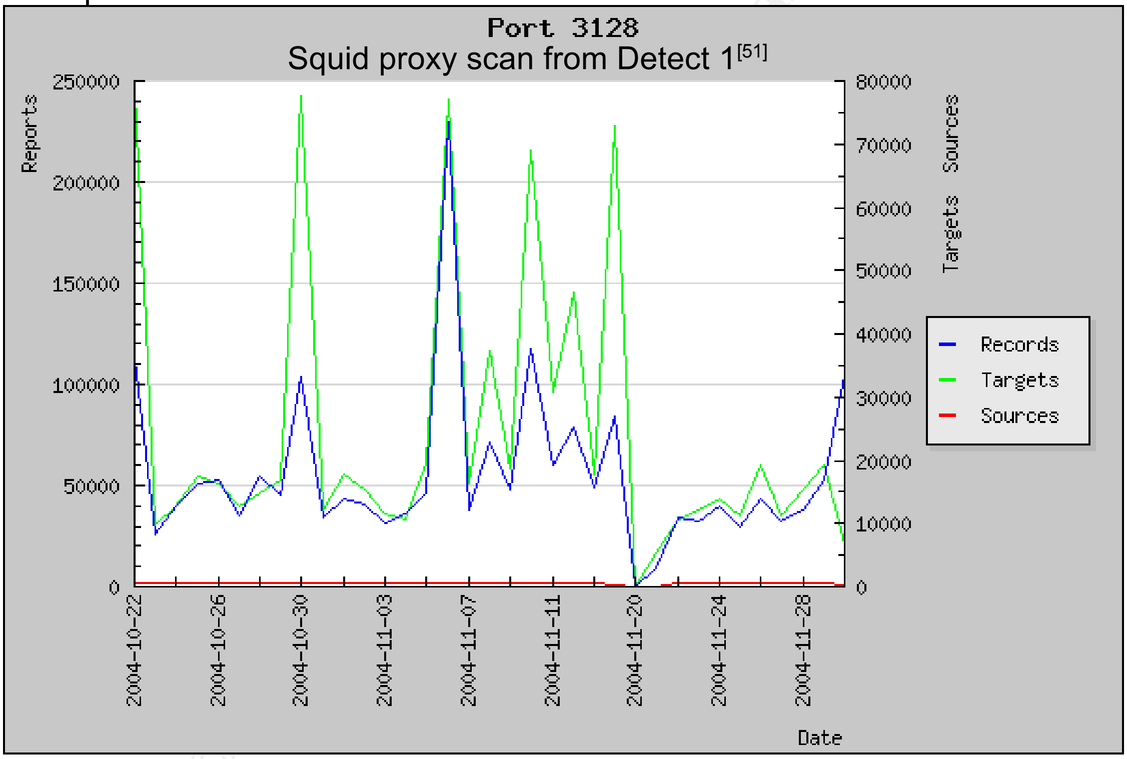
wnn6_Tw	22321		Wnn6 (Taiwanse input)
icq	1027		icq instant messanger
---	7674		
---	1025		
netbios-ns	137		NETBIOS Name Service

All the DShield links have been left intact as they also serve as a reference points.

The DShield Port of The Day[48] did however yield common results.

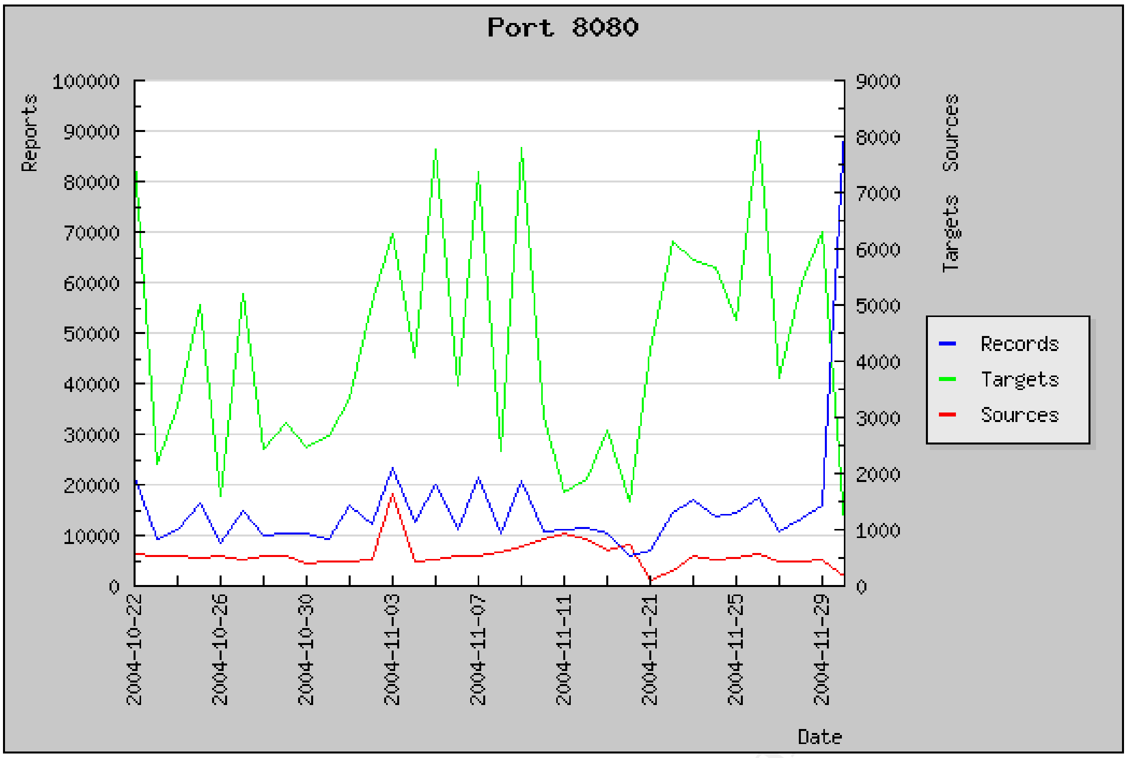
On 30th Nov 2004 they listed port 53 – DNS, Port 1080 - Proxy Servers, Port 137 – NETBIOS, Port 111 (rpc.statd), Port 80 – HTTP. This compares to the analysis port 53 is #9, port 1080 is #7 and port 80 is #3 on our top ten targeted port.

The DShield port reports on ports 3128[49] and 8080[50] show peaks and troughs in the targeting of these ports, as with most known ports. On the next page are two charts plotting the recent activity on ports 3128 and 8080, they show that both these ports are still being targeted in reasonable numbers although they do not appear in the top 10 list.



Squid proxy scan from Detect 1[51]

HTTP proxy scan from Detect 1[51]



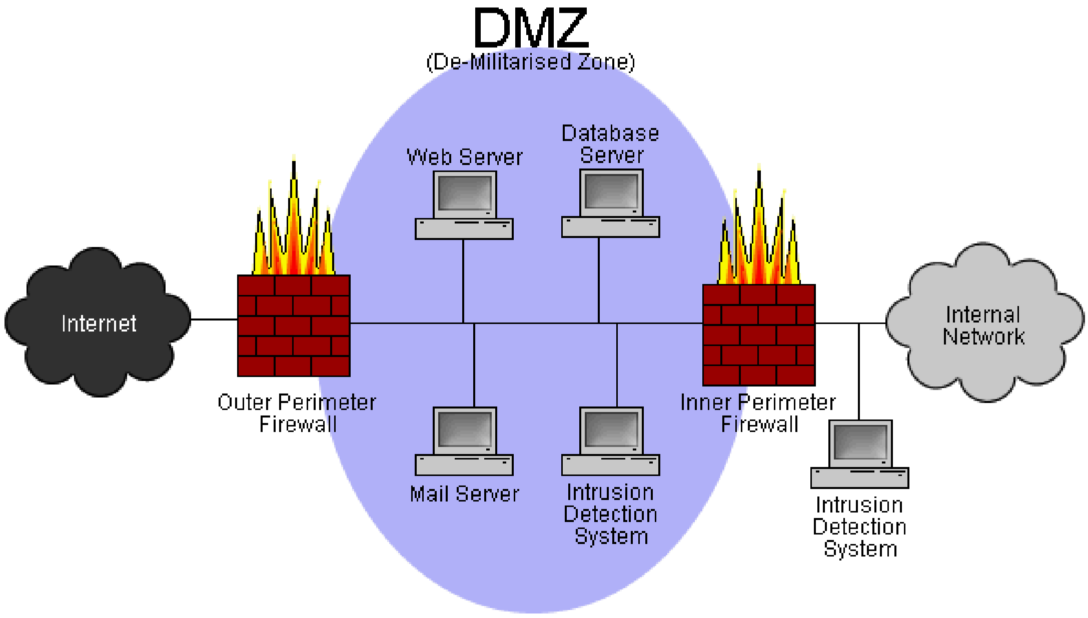
As you can see the activity does not seem to be directly linked to port 3128, port 8080 is much less targeted here then port 3128.

Insights

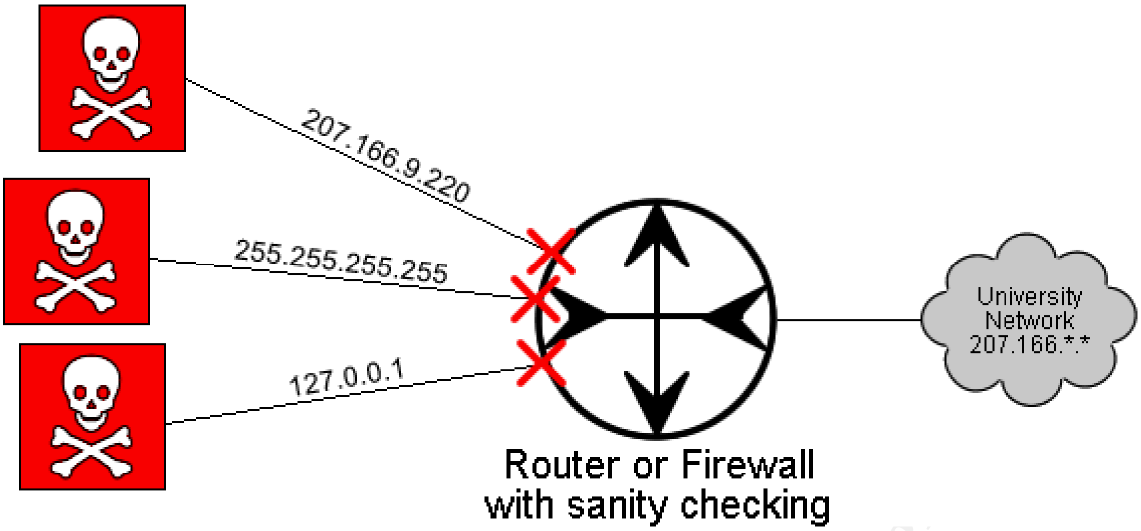
Due to the location of where the traffic was captured it is undetermined what the compromises/infections were, this is due to the inability to correlate the data from and IDS or packet capture from the other side of the router/firewall of the university. If the defensive recommendations are followed the data provided from the two sensors (one in the DMZ and one on the inside of the inner perimeter firewall) would be much more beneficial and a more complete analysis could be performed.

Defensive Recommendations

Based on my analysis I make these defensive recommendations.
As I know little about the structure of the internal network and any system or network security measures, I will assume that there are no security measures in place and give security recommendations as if there were none in place.
In the network topology [52] section there was a limited diagram of what we knew.
The traffic analysed was gathered from between two routers or network devices; it is most likely that the network devices were Cisco, judging from the MAC addresses [53] used.
A De-Militarised Zone (DMZ) is a safer way to host any of your web servers or external facilities.



The DMZ contains the outward or public facing part of the network, it is essential that all the machines in this zone are fully patched, protected with anti-virus and that it is always kept up to date and also protected by personal firewalls. All of the servers in this zone and the internal network should have synchronised clocks preferably set to UTC, this could be achieved by using a recognised NTP (Network Time Protocol) server on the internet or setting up a local one. All other workstations could be set to local time to aid the users, this also must be documented and the time accurately maintained in case of an intrusion into the system and for the correlation of logs. The internal network should also use a address allocated for private use, these are not routable across the Internet and are documented in RFC 1918 [54]. In addition to all of that an intrusion detected system with full packet capture should reside in this zone and be continually monitored 24/7, this is assuming that the systems are critical enough to warrant that sort of time and expense. Behind the Nat'd inner perimeter firewall should lie another intrusion detection system with full packet capture this is to correlate with the data from the IDS in the DMZ to ensure that the inner perimeter firewall policy and rules is adequate and that the university network is protected. On the inside of your protected internal network of workstation and servers; including your internal mail server and internal web proxy, both of these should have specialist anti-virus on them, mail-sweeper and spam filter on the mail server and a web-washer to filter/clean/deny html and other web related media on the web proxies. Any external facing database servers or external mail servers should be placed in the DMZ also. And all machines on the internal/external network, personal firewalls and anti-virus should be employed, monitored and updated by specialist personnel (i.e. sys or security administrators). Sanity checking on the nat'd firewall protecting the internal and external networks should also be enforced, this should include forbidding the internal network, loopback, broadcast and multicast addresses from entering from the external interface of the firewall. RFCs 1918 [54] and 2827 refer [55]


207.166.9.220
255.255.255.255
127.0.0.1
University Network 207.166.*.*
Router or Firewall with sanity checking

The IPs above are not an extensive list of what should be denied access to your network and are merely a starting point. Other addresses including the private ranges e.g. 10.*.*.* and 192.168.*.* for example should also be blocked as well as any multicast addresses i.e. 224.0.1.172 Nokia Cluster. In addition to all this it would be prudent to use or at least consult the DShield block list[45] when configuring the router, as this list is updated the router/firewall should also be continually updated as the list changes.
In addition to the broadcast address 255.255.255.255, 192.168.*.*, 10.*.*.*, 172.16-31.*.* and auto configuration IP 169.254.*.* ranges should also be blocked at the router/firewall noted in the defensive recommendations. For more extensive information on sanity checking and defeating DOS (Denial Of Service) attacks that use IP spoofing, refer to RFCs 1918 [54] and 2827 [55]

Analysis Process

The data selected for analysis was the 2002.10.10 log from the SANS raw logs file[1].
All data was processed and analysed on a Dell Latitude C810 Notebook with 512Mb RAM running Windows 2000 Pro with service pack 4. Notes and reports were also written on this hardware.

Additional Software:
Snort 2.2.0 Win32 build 30 available from snort.org[56]
EagleX version 2.1 from Engage Security available from Engage Security[57] with Snort 2.2.0 embeded into it – achieved by re-installing Snort into the EagleX snort path and then modifying the Step 3 Configuring the Output Plugins of the snort.conf file to input the data into the EagleX database with a similar line to this :- `output database: alert, mysql, host=localhost port=7788 dbname=snort user=snort password=EagleXsnort encoding=hex detail=full`
The above example uses the EagleX default usernames and passwords (just for an

example) and additional output plugin of `output alert_full: 2004-11-13.ids` was used to get a file that Sawmill could process.
This was all initiated using the following command
`C:\eaglex\snort\bin\snort -r c:\logs\2002.10.10 -c c:\eaglex\snort\etc\snort.conf -k none -e -l c:\log` this really means use snort, read from this file, use this configuration, ignore bad checksums, dump the Ethernet header and log to this file.
Sawmill 6.5.11[29] is commercial log analysis software.
WinDump[58] with WinPcap 3.0[60].
Ethereal[61] network protocol analyser.
I started out primarily using EagleX, but when examining the packets EagleX tries to resolve hostnames automatically and take forever when it tries to resolve an obfuscated IP. This whole slow process gets very irritating when you are desperate to see the packet hex, thus I ditched EagleX and moved to Sawmill. Sawmill can take in the alert output files and can be used to drill down and retrieve some very interesting facts about the data. The ability to use filters either entered manually or by virtue of drilling down through the information.
To back up Sawmill you have to examine the packet contents, that is the only way you can prove that the signature that Snort or indeed any IDS has alerted is a true positive or a false positive. Sawmill only takes in the data, it cannot process it on its' own. It relies on you (the analyst) to interpret the data and come to the correct conclusions. For the packet examination process WinDump and Ethereal were used. For nice hex printouts with no extra rubbish in them WinDump was used, this way I could examine specific packets without being clouded with the distractions of other packets. Ethereal was also used, with filters and with colourising rules. Colourising rules in Ethereal are extremely helpful and can be used to great effect when tracking patterns or anomalies.
All the tools I have used run reasonably well on Windows using the limited hardware I had (a bit of patience helps). No additional scripts or custom written programs were used in my log analysis process.

References

Including further reading references that influenced the writing of this paper.

References are listed here in numerical order with chapter header, informing what section that the reference first appears in.

Abstract Overview

[1] – Log file 2002.10.10 from the raw logs directory of the SANS Internet Storm Center http://isc.sans.org/logs/raw/2002.10.10

[2] – SANS Internet Storm Center – http://isc.sans.org © 2002-2004 The SANS Institute

Network Topology

[3] – Ethereal Network Protocol Analyser version 0.10.7 © 1998-2004 Gerald Combs Gerald@Ethereal.com

[4] – Hutley Brett, GIAC GCIA Version 3.4 Practical Detect dated Fri Sep 3 00:40:37 UTC 2004, http://lists.sans.org/pipermail/intrusions/2004-September/008410.html

[5] – SANS Intrusion Mailing list, http://lists.sans.org/pipermail/intrusions/

An Overview of the Detects

[6] - Glenn Forbes Fleming Larratt (*glratt rice.edu*) post to the neohapsis discussion lists made on Fri Feb 08 2002 - 16:06:01 CST, http://archives.neohapsis.com/archives/snort/2002-02/0152.html

Detect 1 – Proxy Scans

[7] - Symantec Security Response
http://securityresponse.symantec.com/avcenter/venc/data/pf/ringzero.trojan.html

[8] - Network Associates, McAfee Anti Virus
http://vil.nai.com/vil/content/v_10356.htm

[9] - F-Secure virus description
http://www.f-secure.com/v-descs/ringzero.html

[10] – Kovacevich, Susan GIAC GCIA Version 3.3. Practical Detect #1
http://www.dshield.org/pipermail/intrusions/2002-october/005644.php

[11] – Kovacevich, Susan GIAC GCIA Practical version 3.3
http://www.giac.org/practical/GCIA/Susan_Kovacevich_GCIA.pdf

Northcutt, Steven Intrusion Detection FAQ, What is the Ring Zero scan?
http://www.sans.org/resources/idfaq/ring_zero.php

DShield.org port report
http://www.dshield.org/port_report.php?port=8080
http://www.dshield.org/port_report.php?port=3128

Calculating Severity
http://www.dshield.org/pipermail/intrusions/2003-january/006745.php

Public Proxy Servers
http://www.publicproxyservers.com/index.html

Detect 2 – Backdoor Q Access

[12] – Al Maslowski-Yerges GCIA paper

http://www.giac.org/practical/GCIA/Al_Maslowski-Yerges_GCIA.pdf

[13] – Security Focus thread http://www.securityfocus.com/archive/75/182244/2002-11-04/2002-11-10/1

[14] – Security Focus thread http://online.securityfocus.com/archive/75/194288

[15] – Trenton Riddle GCIA paper http://www.giac.org/practical/Trenton_Riddell_GCIA.doc

[16] – Security Focus message http://www.securityfocus.com/archive/75/182133/2002-11-04/2002-11-10/2

[17] – Peter Storm GCIA paper http://cert.uni-stuttgart.de/archive/intrusions/2002/09/msg00192.html

SANS article on the Backdoor Q trojan written by Les Gordan http://www.sans.org/resources/idfaq/qtrojan.php

Message to Tod Beardsley (GCIA, MCSE) from Kevin Timm, http://cert.uni-stuttgart.de/archive/intrusions/2002/09/msg00192.html

Network Associates, McAfee Anti-Virus http://vil.nai.com/vil/content/v_100468.htm

Just for reference Backdoor Q can be found a mixters page http://mixter.warrior2k.com/

Detect 3 – BAD-TRAFFIC IP Reserved Bit Set

[18] – Bellovin S. AT&T Labs Research, Informational RFC, 3514, dated 1 April 2003 - http://www.faqs.org/rfcs/rfc3514.html

[19] – Kerry Long GCIA practical, http://www.giac.org/practical/GCIA/Kerry_Long_GCIA.pdf

[20] – Bob Frittons' mailing list post http://cert.uni-stuttgart.de/archive/intrusions/2002/09/msg00079.html

[21] – http://www.snort.org the home of the pig and associated info on it.

[22] – Information Sciences Institute University of Southern California, Internet Protocol, DARPA Internet Program Protocol Specification, September 1981, RFC 791 - http://www.faqs.org/rfcs/rfc791.html

[23] – Hping2 documentation and download available from http://www.hping.org/

[24] – Netdude documentation and download available from http://netdude.sourceforge.net/

[25] – Information on default TTL values for various operating systems

http://secfr.nerim.net/docs/fingerprint/en/ttl_default.html#overview

[26] – Shuck, Ron LOGS: GIAC GCIA Version 3.3 Practical Detect on Tue Feb 11 02:14:53 UTC 2003 http://www.dshield.org/pipermail/intrusions/2003-february/006840.php

[27] – Shuck, Ron GCIA Practical v3.3 http://www.giac.org/practical/GCIA/ron_shuck_GCIA.pdf

Network Statistics

[28] – P0f, Passive Operating Fingerprinting by Michal Zalewski available from http://lcamtuf.coredump.cx/p0f.shtml

[29] – Sawmill, a commercial log processor, Copyright © 2004 by Flowerfire, available from http://www.sawmill.net

[30] – Country of origin was gained from a whois at SamSpade. http://www.samspade.org

[31] – KRNIC (English site), http://www.nic.or.kr/www/english/

[32] – Hitbox.com information from Adbolish, http://www.adbolish.com/privacy.asp

[33] – IANA ports list http://www.iana.org/assignments/port-numbers

[34] – Ports Database http://www.portsdb.org/

[35] – Treachery.net online port lookup http://www.treachery.net/tools/ports/lookup.cgi

[36] – Dshield port lookup http://www.dshield.org/port_report.php

[37] – Internet Storm Centre port lookup http://isc.sans.org/port_report.php

[38] – ISS port reports on port 515 http://www.iss.net/security_center/advice/Exploits/Ports/515/default.htm

[39] – WinDump URL - http://windump.polito.it/, TCPDump URL - http://www.tcpdump.org/

Three Most Suspicious External IPs

[40] – URL used for the Samspade.org 255.255.255.255 lookup http://www.samspade.org/t/lookat?a=255.255.255.255, IANAs website is at http://www.iana.org.

[41] – Detect 2, page 8 of this paper

[42] – IANA link for IP 172.20.10.199 as part of a private address range RFC 3330,

http://www.rfc-editor.org/rfc/rfc3330.txt

[43] – KRNIC http://www.nic.or.kr/www/english/, link to 211.47.25.20
http://www.samspade.org/t/whois?a=211.47.25.20

[44] – SamSpade IP lookup facility, http://www.samspade.org

Correlations

[45] – DShield Block list, http://feeds.dshield.org/block.txt

[46] –Network Statistics, page 15 of this paper.

[47] – DShield Top 10 target ports http://www.dshield.org/topports.php

[48] – DShield Port of The Day, http://www.dshield.org/port_of_the_day.php

[49] – Current DShield report on port 3128,
http://www.dshield.org/port_report.php?port=3128

[50] – Current DShield report on port 8080,
http://www.dshield.org/port_report.php?port=8080

[51] – Detect 1, page 5 of this paper

Defensive Recommendations

[52] – Network Topology, page 3 of this paper

[53] – Coffer.com MAC Address finder, http://www.coffer.com/mac_find/

[54] - RFC 1918, http://www.faqs.org/rfcs/rfc1918.html

[55] – RFC 2827, http://www.faqs.org/rfcs/rfc2827.html

Analysis Process

[56] – Snort, available from the downloads section of
http://www.snort.org/downloads

[57] – EagleX, available from http://www.engagesecurity.com/downloads/#eaglex

[58] – Windump, the Win32 version of TCPDump[59], available from the downloads section of http://windump.polito.it

[59] – TCPDump, a packet sniffer with the ability to use BPF (Berkeley Packet Filters), available from http://www.tcpdump.org

[60] – WinPCap, *"the Free Packet Capture Library for Windows"*, available from the

downloads section of http://winpcap.polito.it/

[61] – Ethereal, *“The world's most popular network protocol analyzer”*, available from download section of http://www.ethereal.com/

[62] - Registro de domínios para a Internet no Brasil, http://registro.br/index.html

Annex A – Whois on IPs used in the Detects

Detect 1 – Multiple Proxy Scans

Single Source 66.123.116.234

Output and data from Samspade.org [30]

Country of origin – United States of America

Server Used: [whois.arin.net]

```
66.123.116.234 = [ adsl-66-123-116-234.apllab.com ]

  OrgName:    Pac Bell Internet Services
  OrgID:      PACB
  Address:    208 Bush St. 5000
  City:       San Ramon
  StateProv:  CA
  PostalCode: 94104
  Country:    US
  NetRange:   66.120.0.0 - 66.127.255.255
  CIDR:       66.120.0.0/13
  NetName:     PBI-NET-9
  NetHandle:  NET-66-120-0-0-1
  Parent:      NET-66-0-0-0-0
  NetType:    Direct Allocation
  NameServer: NS1.PBI.NET
  NameServer: NS2.PBI.NET
  Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
  Comment:    please send all abuse issue e-mails to abuse@pbi.net

  RegDate:    2001-05-01
  Updated:    2001-09-26
  TechHandle: PIA2-ORG-ARIN
  TechName:   IPAdmin-PBI
  TechPhone:  1-800-648-1626
  TechEmail:  IPAdmin-PBI@sbis.sbc.com

  OrgAbuseHandle: APB2-ARIN
  OrgAbuseName:   Abuse - Pacific Bell
  OrgAbusePhone:  1-800-648-1626
  OrgAbuseEmail:  abuse@pacbell.net

  OrgNOCHandle: SPBI-ARIN
  OrgNOCName:   Support - Pacific Bell Internet
  OrgNOCPhone:  1-800-648-1626
  OrgNOCEmail:  support@pacbell.net

  OrgTechHandle: PIA2-ORG-ARIN
  OrgTechName:   IPAdmin-PBI
  OrgTechPhone:  1-800-648-1626
  OrgTechEmail:  IPAdmin-PBI@sbis.sbc.com

  CustName:   PhysiciansLabSolution
  Address:    268 Bush Street
  City:       San Francisco
  StateProv:  CA
  PostalCode: 94104
  Country:    US
  RegDate:    2001-07-03
```

```
Updated:     2001-07-03
NetRange:    66.123.116.232 - 66.123.116.239
CIDR:        66.123.116.232/29
NetName:     SBCIS-10173-1763
NetHandle:    NET-66-123-116-232-1
Parent:      NET-66-120-0-0-1
NetType:     Reassigned
Comment:
RegDate:     2001-07-03
Updated:     2001-07-03
TechHandle:  PIA2-ORG-ARIN
TechName:    IPAdmin-PBI
TechPhone:   1-800-648-1626
TechEmail:   IPAdmin-PBI@sbis.sbc.com

OrgAbuseHandle: APB2-ARIN
OrgAbuseName:   Abuse - Pacific Bell
OrgAbusePhone:  1-800-648-1626
OrgAbuseEmail:  abuse@pacbell.net

OrgNOCHandle: SPBI-ARIN
OrgNOCName:   Support - Pacific Bell Internet
OrgNOCPhone:  1-800-648-1626
OrgNOCEmail:  support@pacbell.net

OrgTechHandle: PIA2-ORG-ARIN
OrgTechName:   IPAdmin-PBI
OrgTechPhone:  1-800-648-1626
OrgTechEmail:  IPAdmin-PBI@sbis.sbc.com

 ARIN WHOIS database  last updated 2004-12-09 19: 10
 Enter ? for additional hints on searching ARIN's WHOIS database.
```

Detect 2 – Backdoor Q access… or is it?

Single Source 255.255.255.255

Output and data from Samspade.org [30]

Country of origin – Unknown

Server Used: [none]

```
ERROR:   IP Range Reserved by IANA.org
```

Detect 3 – BAD-TRAFFIC IP Reserved Bit Set

Single Source 200.200.200.1

Output and data from Samspade.org [30]

Country of origin – Brazil

Server Used: [whois.registro.br]

```
200.200.200.1 = [  ]

 ERROR:  Unable to connect to whois.registro.br for 200.200.200.1 ...
Aborting
```

Samspade.org was unable to resolve the IP, however, it yielded the whois it was trying to connect to.

Output from the Brazilian Whois Registro [62]

```
% Copyright registro.br
%  The data below is provided for information purposes
%  and to assist persons in obtaining information about or
%  related to domain name and IP number registrations
%  By submitting a whois query, you agree to use this data
%  only for lawful purposes.
%  2004-12-10 10:53:13 (BRST -02:00)

inetnum:      200.200/16
asn:          AS4230
ID abusos:    GSE6
entidade:     EMBRATEL-EMPRESA BRASILEIRA DE TELECOMUNICAÇÕES SA
documento:    033.530.486/0001-29
responsável:  Gerência Internet EMBRATEL
endereço:     R. Alexandre Mackenzie, 75, 6 andar
endereço:     20221-410 - Rio de Janeiro - RJ
telefone:     (21) 21212507 []
ID entidade:  CAP12
ID técnico:   FSA82
inetrev:      200.200/16
servidor DNS: NS.EMBRATEL.NET.BR
status DNS:   09/12/2004 AA
último AA:    09/12/2004
servidor DNS: NS2.EMBRATEL.NET.BR
status DNS:   09/12/2004 AA
último AA:    09/12/2004
criado:       17/11/1999
alterado:     24/05/2002

ID:           CAP12
nome:         Gerencia Técnica de Operações Internet
e-mail:       domain-admin@EMBRATEL.NET.BR
endereço:     Rua Senador Pompeu, 119, 6 and
endereço:     20221-291 - Rio de Janeiro - RJ
telefone:     (21) 21212828 []
criado:       02/02/1998
alterado:     17/11/2004

ID:           FSA82
nome:         Gerência Técnica de Servidores Internet
e-mail:       hostmaster@EMBRATEL.NET.BR
endereço:     Rua Senador Pompeu, 119, 608
endereço:     20221-291 - Rio de Janeiro - RJ
telefone:     (021) 25192827 []
criado:       24/05/2002
alterado:     27/05/2002

ID:           GSE6
nome:         Grupo de Segurança Internet da Embratel
e-mail:       abuse@EMBRATEL.NET.BR
endereço:     R. Senador Pompeu, 119, 6. andar
```

```
endereço:      20080-001 - Rio de Janeiro - RJ
telefone:      (078) 21278 []
criado:        05/10/2000
alterado:      05/10/2000

remarks:      Security issues should also be addressed to
remarks:      nbso@nic.br, http://www.nbso.nic.br/
remarks:      Mail abuse issues should also be addressed to
remarks:      mail-abuse@nic.br

% whois.registro.br accepts only direct match queries.
% Types of queries are: domains (.BR), BR POCs, CIDR blocks,
% IP and AS numbers.
```