Global Information Assurance Certification Paper

University X Security Audit: Q Trojan, P2P, and UDP Scanning

GCIA Practical
Version 4.0

Jesse Emerson
Rocky Mountain SANS
June 5-10, 2004

Submitted
November 16, 2004

Table of Contents

Abstract

This paper provides an executive summary outlining prudent security measures for protecting a university network. It also provides network topology as extrapolated from raw binary file captures. Three detects, BACKDOOR Q Trojan, Gnutella P2P Connect, and DNS named version attempt are reviewed in depth. There are also profile exercises of three suspicious external source IP addresses that generate port 0 UDP traffic, SHELLCODE x86 NOOP, and MISC Source Port 20 to <1024 alerts.

Although a minimal portion of the, data visualization is one of the focus items of this paper, primarily using Visual Insight's Advizor Workbench. The intent is to display meaningful information in graphic format in order to augment the detailed analysis provided in the body of the paper.

Document Conventions

When you read this practical assignment, you will see that certain words are represented in different fonts and typefaces. The types of words that are represented this way include the following:

command	Operating system commands are represented in this font style. This style indicates a command that is entered at a command prompt or shell.
`filename`	Filenames, paths, and directory names are represented in this style.
`computer output`	The results of a command and other computer output are in this style
URL	Web URL's are shown in this style.

Introduction / Executive Summary

CERT statistics and security surveys show that Cyber Security concerns are still on the rise. Network administrators state that "Security/Hackers" are the number one item that keeps them up at night[1]. Universities have an enormous challenge in keeping their networks and vital systems up and running in a secure state.

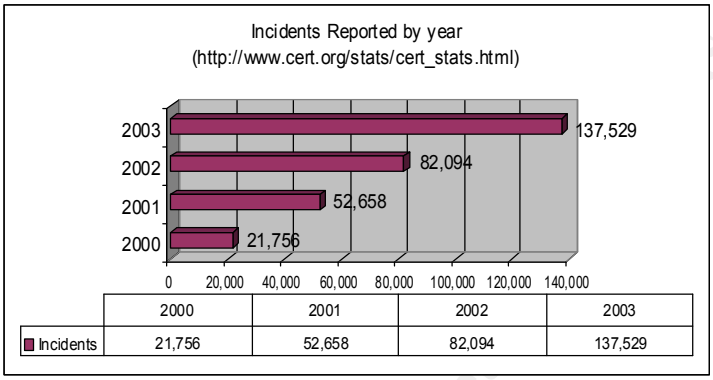

Figure 1[2]

There are many contributing factors to the difficulties in securing university networks. In order to foster research, learning, and discovery, universities often err to the side of lax security rather than inhibit the educational process. Universities also have a large user base with high turnover. Networks are distributed widely throughout campuses, with high bandwidth connections (often T1 or better) available in not only computer labs and classrooms, but also dorm rooms and on-campus housing. University networks typically have remote access capability offered for students as well. The open environment, high bandwidth availability, and the broad user base result in a significant security challenge for network administrators and security professionals. "Hackers" and attackers are also aware of this environment and the security challenges, making universities an ideal target.

This document is the result of an audit of nine days of IDS data from University X. Analysis of this traffic has resulted in an understanding of the network topology as well as a sense of the security posture of the university's network and systems. There is evidence of active scanning and probing in the examined IDS logs, as well as evidence of possible misuse, present threats, and potential compromises.

[1] Gaspar, Suzanne. "Security concerns dominate NW500 survey." Network World. 05 May, 2001. URL: http://www.nwfusion.com/research/2001/0507feat2.html (15 November 2004).

[2] "CERT/CC Statistics 1988-2004." CERT Coordination Center. 19 October 2004. URL: http://www.cert.org/stats/cert_stats.html (15 November 2004).

To maintain a healthy security posture, it is recommended that a Defense in Depth approach to network and system security is implemented. Networks should be segmented into logical groupings and ACL router controls or firewalls should be in place between segments. A suggested segmentation plan may include isolating the general student network (dormitory and on-campus housing), remote access networks, wireless networks, classroom networks, computer lab networks, faculty networks, externally accessed networks (in a DMZ), and infrastructure networks. This network segmentation will help to control worm outbreaks as well as control traffic flowing to and from vital systems.

An Acceptable Usage Policy should be in place and all users of the network should agree to this policy before being granted access. University managed systems should run local antivirus and firewall installations and configuration management/patch management strategies should be in place. Any vital business process systems of the university should reside on a limited access network and utilize change management. Host based IDS should be considered for servers and Intrusion Prevention or deep packet inspection firewall devices should be considered at gateways to sensitive network segments. The university is already employing Network IDS in the form of Snort systems, which is a good thing. The placement and configuration of these systems should be reviewed periodically for currency and proper tuning. There is evidence of false positive detects in the IDS data, which should be minimized to keep event volumes down and event values up. The university may also benefit from a combination of passive network discovery and active vulnerability scanning to gather information on network and system devices. Any services found running on the network should be checked for validity, and any vulnerabilities should be patched. Gateway and border firewall and router ACLs should be configured to not forward packets with source IP addresses that are non-routable.

A variety of IDS alerts were triggered by traffic on this network during the audit period. The following three detects identify possible device configuration errors, acceptable use violations, and active reconnaissance from external sources and will be addressed in detail.

1) Backdoor Q alerts, fired from broadcast source IP address 255.255.255.255 TCP port 31337 (ELEET in hacker-speak) to TCP port 515 on multiple hosts.

2) P2P Gnutella Client Connect alerts, responsible for a large amount of bandwidth consumption.

3) DNS named version attempt scanning from high ports to multiple destinations over the span of the audit period.

Detailed Analysis

Scenario

University X has contracted for a security audit using the binary network capture files from their Snort intrusion detection system.

Source of files used

The files used in this audit are from http://isc.sans.org/logs/Raw:

```
2002.4.18
2002.4.19
2002.4.20
2002.4.21
2002.4.22
2002.4.23
2002.4.24
2002.4.25
2002.3.26
2002.4.27
```

Network Topology

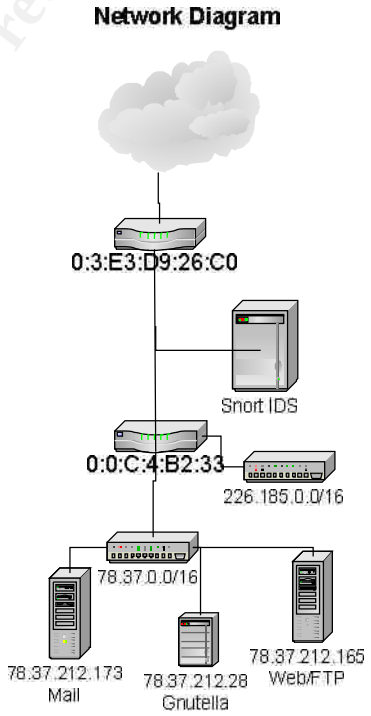

Figure 2

Due to the presence of only two Ethernet MAC addresses in the IDS data -- 0:0:C:4:B2:33 and 0:3:E3:D9:26:C0 -- both Cisco MACs, it is assumed that the IDS sensor is placed between two routing devices (possibly plugged into a spanned switch port, a hub, or in tap mode). Two class B address spaces appear to be in use on the University's internal network: 78.37.0.0/24 and 226.185.0.0/24. The 78.0.0.0/8 network is set aside as IANA Reserved. The 226.0.0.0/8 class A is in the IANA Multicast block and also marked as "reserved". Both address spaces are reporting TCP and UDP traffic through the internal router MAC address, suggesting that the university has opted to utilize these address ranges, most likely with Network Address Translation, on the internal networks. The simplest representation of this topology is drawn in Figure 2. The placement of the Snort IDS, the major servers on the network (web, ftp, and email), and the workstation running Gnutella are all identified in the network diagram.

Overview of identified detects

Below is a chart listing the various detects, with count, identified using a modified Snort ruleset to analyze the data in the nine days of packet captures.

Sig_ID	Message	Count
4	(http_inspect) BARE BYTE UNICODE ENCODING	2,089
0	P2P GNUTella client connect	1,487
1616	DNS named version attempt	331
184	BACKDOOR Q access	321
13	(http_inspect) NON-RFC HTTP DELIMITER	117
648	SHELLCODE x86 NOOP	64
653	SHELLCODE x86 0x90 unicode NOOP	44
2	(http_inspect) DOUBLE DECODING ATTACK	44
12	(http_inspect) APACHE WHITESPACE (TAB)	27
15	(http_inspect) OVERSIZE REQUEST-URI DIRECTORY	24
1390	SHELLCODE x86 inc ebx NOOP	13
1394	SHELLCODE x86 NOOP	11
18	(http_inspect) WEBROOT DIRECTORY TRAVERSAL	10
503	MISC Source Port 20 to <1024	8
7	(http_inspect) IIS UNICODE CODEPOINT ENCODING	8
46	(snort_decoder) WARNING: TCP Data Offset is less than 5!	6
16	(http_inspect) OVERSIZE CHUNK ENCODING	3
523	BAD-TRAFFIC ip reserved bit set	3
566	POLICY PCAnywhere server response	1
522	MISC Tiny Fragments	1
525	BAD-TRAFFIC udp port 0 traffic	1
621	SCAN FIN	1

Table 1

Figure 3 is a graphical representation of alerts, IP addresses, and Ports extracted from the IDS data.

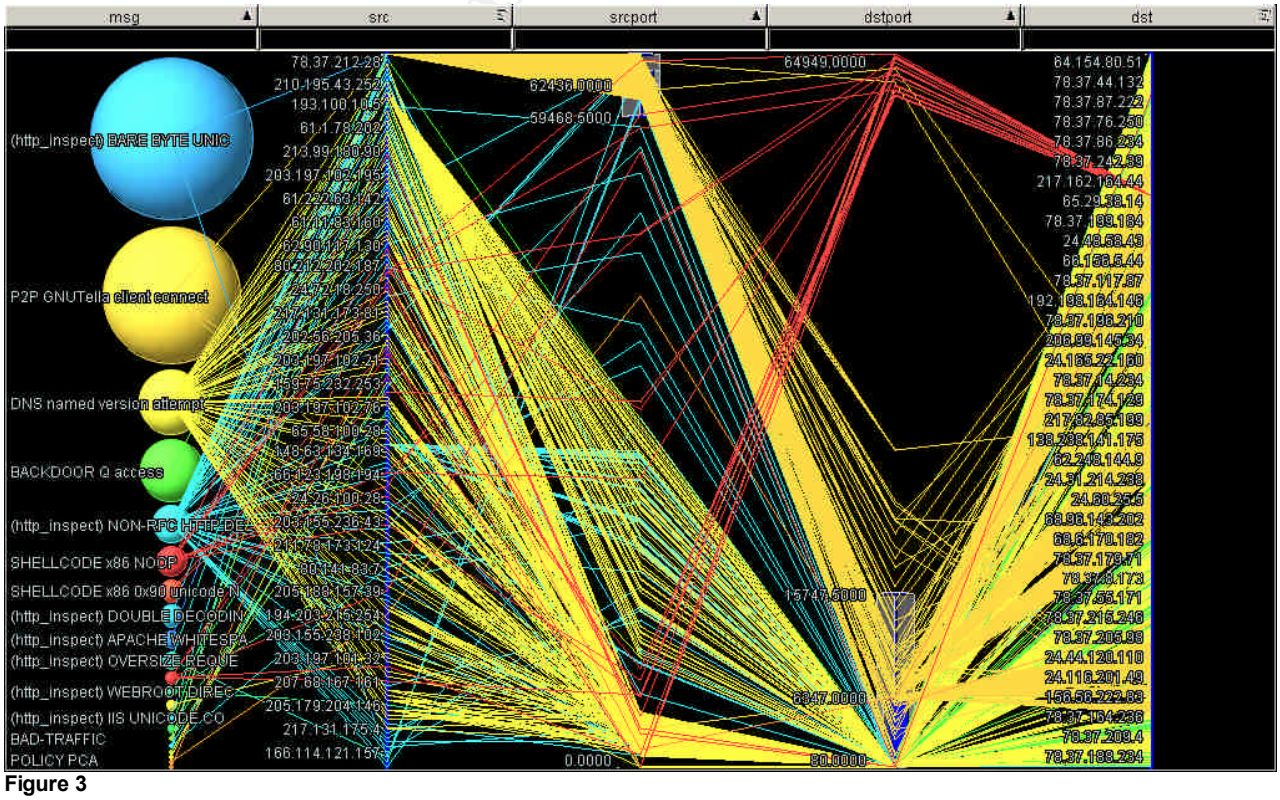

Figure 3

Detect 1: BACKDOOR Q access

The most suspicious traffic in the analyzed data is that which triggered the Backdoor Q alerts. This traffic was sourced from the broadcast address 255.255.255.255, port 31337 to multiple destinations on TCP port 515.

Description of detect

The Q Trojan is a stealth backdoor application that can allow a remote user with a "q stealth messenger" (qs) to execute commands or bounce traffic on a computer (primarily Unix flavors) running a "q Daemon" (qd); in the common version of Q, the qs and qd are compiled as a pair and a single qs cannot send commands to multiple qd devices. This Trojan was developed by "Mixter" in 1999 as a proof of concept tool. Q is difficult to detect on the network due to its use of encryption.[3] Q is a Trojan, not a Worm and therefore must be installed rather than self propagating. This means that if a box is running the Q Trojan, it has already been compromised.

Les Gordon has performed in depth research on the various versions of the Q Trojan. His excellent FAQ can be found here: http://www.sans.org/resources/idfaq/qtrojan.php .

Reason this detect was selected

This traffic was selected for a variety of reasons. If it is indeed the Q Trojan in action, this suggests that devices on the university's network have been compromised and an attacker is attempting to contact the q daemon no the compromised host.

The source port of 31337 (ELEET in hackerspeak) is one of the most "evil" ports in use on the internet, and often picked for use in malicious applications and by "script kiddies". This port is also very common in worms and Trojans including Back Orifice and ADM Worm. A lookup of port 31337 at http://www.treachery.net/tools/ports/lookup.cgi will return 22 different Trojans and worms that utilize this port.[4]

The IP address of 255.255.255.255 is reserved for broadcast and should not be routable from the internet if it is a datagram's *destination.* RFC 919 states, "The address 255.255.255.255 denotes a broadcast on a local hardware network,

[3] Gordon, Les. "What is the Q Trojan?" SANS FAQ. URL: http://www.sans.org/resources/idfaq/qtrojan.php (15 November 2004).
[4] "Treachery Unlimited." URL: http://www.treachery.net/tools/ports/lookup.cgi (15 November 2004).

which must not be forwarded."[5] However, no RFCs were found that specify that the *source* IP address 255.255.255.255 must not be forwarded. If this traffic did originate from the internet, configuring the router to drop "bogon"[6] source IP addresses, as well as 255.255.255.255 would be a prudent change. It does appear that the packets were forwarded from the internet due to the packets containing the external gateway Ethernet MAC address 0:3:E3:D9:26:C0.

The high number of destination IP addresses affected by this activity over a span of multiple days increases the severity with which the traffic should be addressed.

Detect was generated by

This full alarm below was generated in Snort 2.0.1 running in IDS mode. Although the binary files are dated as being from April 2002, the timestamps in the files are from May 2002.

```
snort -c /etc/snort/snort.conf -r 2002.4.21 -nnvX -A full

[**] [1:184:6] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
05/21-16:54:10.044488 255.255.255.255:31337 -> 78.37.14.234:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]
```

An example packet, as displayed by Snort in non-IDS mode is below:

```
C:\Documents and Settings\Administrator\Desktop\GCIA4.0>c:\snort\bin\snort -r
2002.4.20 > outtahere.txt

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/20-11:47:54.404488 255.255.255.255:31337 -> 78.37.92.243:515
TCP TTL:13 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
63 6B 6F                                          cko

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

This log states that at the time "05/20-11:47:54.404488", the source IP address "255.255.255.255"sent packet with the *acknowledge/reset* "***A*R**" flags set from port "31337" to the destination IP "78.37.92.243" on port "515". The only payload in these packets is the string "cko".

The time to live (ttl) on this packet is 13. An item of note is that all of the packets before 05/21 09:51 have a ttl of 13. After this time, all of the packets, with the exception of one (05/21-23:00, ttl 13), have a ttl of 14. This would indicate that

[5] Mogul, Jeffrey. "RFC 919 - Broadcasting Internet Datagrams." October 1984. URL: http://cert.uni-stuttgart.de/archive/intrusions/2003/01/msg00495.html (16 November 2004).
[6] Thomas, Rob. "Bogon Dotted Decimal List v2.5 02 AUG 2004." URL: http://www.cymru.com/Documents/bogon-dd.html (16 November 2004).

the packets originated from one hop further away, network topology speaking, before 05/21 09:51. Either the source of these packets physically moved on the network, indicating that this may be a mobile device, there was a change in network routing that caused the packets to take different path, or two different sources actually sent the packets, both with spoofed source addresses. The ttl value is reduced by one at each routing device along the way between source and destination, and can sometimes be used to fingerprint source system OS and location. The low ttl values on these packets do not logically map to a known OS fingerprint and further suggest that the packets may be crafted.

The rule that triggered on this packet is below:

```
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;)
```

The packets in question tripped this rule on the source IP address of 255.255.255.255 which falls into the range of source IP addresses specified in the rule. This rule is written to catch instances of the Q Trojan contro/activation packets that sometimes (rarely) will have a source IP address of 255.255.255.255.[7] The packets also matched flag requirements with the rst/ack flags, as specified in the "flags:A+" (Ack + something else) portion of the rule.[8]

Probability the source address was spoofed

Whether this is the Q Trojan or not, the source address of 255.255.255.255 is most definitely spoofed. The use of ack/rst flags would not result in any reply packets being sent from the destination IP addresses. In this case, if it is the Q Trojan, the "cko" payload would be control commands issued from the qs to the qd.

Attack Mechanism

If this is the Q Trojan at work, the attack mechanism could have succeeded with the passing of a command string to the server running the matching pair of the source q stealth messenger to q daemon. No subsequent activity violated a Snort rule from the destination IP addresses that were the targets of these attacks. This could mean that no machines were compromised; it may also mean that the compromised machine simply did not do anything to violate a Snort rule.

There are also a variety of vulnerabilities in the line printer daemon (lpd) that runs on port 515. This may be an attempt to attack vulnerable lpd systems. The use

[7] Gordon, Les. "What is the Q Trojan?" SANS FAQ. URL: http://www.sans.org/resources/idfaq/qtrojan.php (15 November 2004).
[8] The Snort project. Snort User's Manual. 11 August 2004. 70-71.

of the ack/rst flags will not result in a response from vulnerable systems, so this would not be effective as a discovery technique.

Correlations

This detect has been seen before, mostly by other IDS analysts examining similar files from http://isc.sans.org/logs/Raw/ with analysis recorded on the www.dshield.org

There is a comprehensive FAQ on the Q Trojan available at:
http://www.faqs.org/rfcs/rfc951.html

Similar attacks have been document in the Security Focus Incidents forums:
http://www.securityfocus.com/archive/75/194288

CERT Advisories regarding vulnerabilities in the Line Printer Daemon:
http://www.cert.org/advisories/CA-2001-30.html
http://www.cert.org/advisories/CA-2001-32.html

Another incident of source IP 255.255.255.255 traffic can be found here:
http://cert.uni-stuttgart.de/archive/intrusions/2003/01/msg00495.html

Evidence of active targeting

This traffic is directed at numerous hosts on the university network. This may be a diffusion technique used by an attacker to conceal the real destination that she is attempting to access. As there will be no response from these packets, this is not likely a reconnaissance attempt. If this is an attack on a vulnerability in LPR, the target does not appear to be known.

Severity

Severity = (criticality + lethality) - (system countermeasures + network countermeasures)

Severity = (3+4) - (1+3) = 3

Criticality = 3 Numerous systems were targeted, but not the entire network nor were any of the identified vital servers targeted.

Lethality = 4 If the attack is control traffic to a q daemon, the result could be significantly damaging.

System Countermeasures = 1 System countermeasures are unknown, and with a university environment, Trojan software could easily be placed on many machines without detection.

Network Countermeasures = 2 The packets were not blocked at a border router or firewall as far as can be distinguished from the packets. However IDS is in place, which improves this rating.

The traffic is graphically represented in Figure 4.

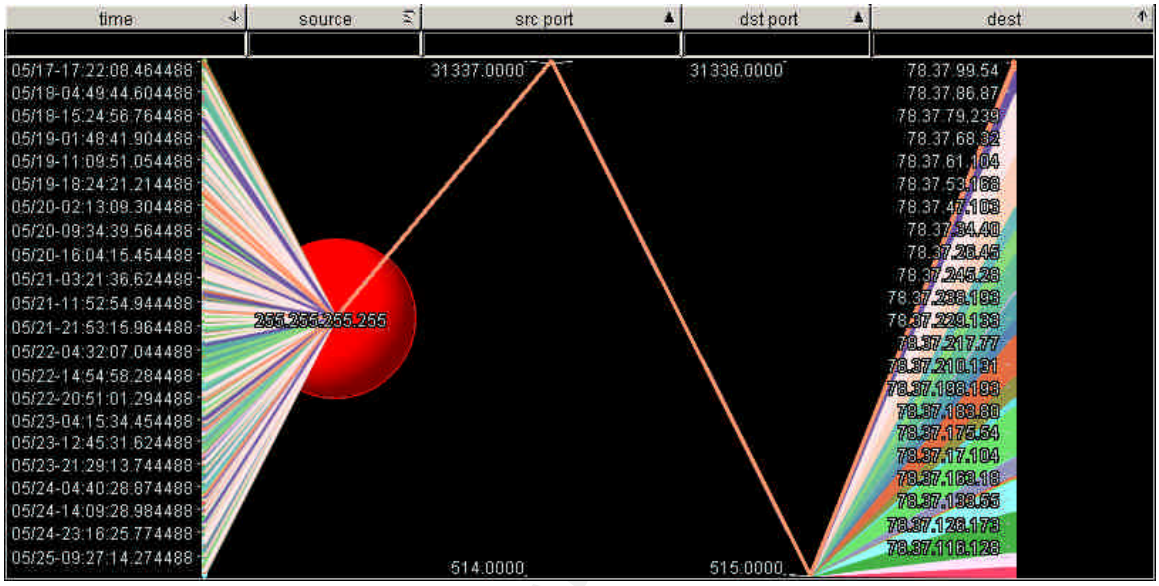

Figure 4

Detect 2: P2P GNUTella client connect

Description of detect

This alarm detects the use of Gnutella peer to peer file sharing software. This, and related activity, was the main source of Snort alarms and TCP traffic on the university network. There are many concerns with the use of Gnutella applications.

Reason this detect was selected

This detect was selected primarily due to the large volume of traffic caused by the Gnutella P2P activity. A single IP address running Gnutella applications within the university was the source of 3,694 Snort alarms. P2P software use may or may not be a violation of university acceptable use policies, but there are significant legal concerns (copyright, inappropriate images, etc) and security concerns related to use of these P2P applications. There is also the risk of

compromise through other malicious P2P users and the fact that spyware accompanies many of the Gnutella client installations. [9]

Detect was generated by

The full alarm below was generated in Snort 2.0.1 running in IDS mode. Although the binary files are dated as being from April 2002, the timestamps in the files are from May 2002.

```
snort -c /etc/snort/snort.conf -r 2002.4.21 -nnvX -A full

[**] [1:0:0] P2P GNUTella client connect [**]
[Priority: 0]
05/21-03:43:46.424488 78.37.212.28:62802 -> 67.80.240.25:6347
TCP TTL:124 TOS:0x0 ID:2283 IpLen:20 DgmLen:94 DF
***AP*** Seq: 0xEE96B337  Ack: 0xC10AD89C  Win: 0x4470  TcpLen: 20
```

An example packet, as displayed by Snort in non-IDS mode is below:

```
C:\Documents and Settings\Administrator\Desktop\GCIA4.0>c:\snort\bin\snort -r
2002.4.20 > outtahere.txt

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/20-11:44:35.254488 78.37.212.28:64677 -> 24.149.6.253:6382
TCP TTL:124 TOS:0x0 ID:58911 IpLen:20 DgmLen:94 DF
***AP*** Seq: 0x5E4BCDF7  Ack: 0x1D97B85B  Win: 0x4470  TcpLen: 20
47 4E 55 54 45 4C 4C 41 20 43 4F 4E 4E 45 43 54  GNUTELLA CONNECT
2F 30 2E 36 0D 0A 55 73 65 72 2D 41 67 65 6E 74  /0.6..User-Agent
3A 20 47 6E 75 63 6C 65 75 73 20 31 2E 36 2E 30  : Gnucleus 1.6.0
2E 30 0D 0A 0D 0A

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

This alert shows that the source IP 78.37.212.28 port 64677 is issuing the “GNUTELLA CONNECT” command to the external host 24.149.6.253 port 6382.

The rule that triggered on this packet is below:

```
alert tcp any any <> any any (msg:"P2P GNUTella client connect";
content:"GNUTELLA CONNECT";)
```

It was apparent from analyzing the traffic that Gnutella traffic was abundant, yet none of the Gnutella rules from the default rule set were violated with the replay configuration. The rule above was created to capture this traffic and simply looks for the string “GNUTELLA CONNECT” on any TCP port in traffic flowing either inbound or outbound. This may not be an efficient rule if analyzing traffic in real time, however for the purpose of post-event analysis, it gets the job done.

[9] “ FILE-SHARING PROGRAMS AND PEER-TO-PEER NETWORKS PRIVACY AND SECURITY RISKS.” UNITED STATES HOUSE OF REPRESENTATIVES COMMITTEE ON GOVERNMENT REFORM. May 2003. URL: http://reform.house.gov/UploadedFiles/P2P%20Security%20Report.pdf (15 November 2004).

Probability the source address was spoofed

There is very low probability that this source is spoofed. Gnutella requires two way communications, which are not possible with spoofed addresses.

Attack Mechanism

To quote the Declan Murphy, et.al., paper on P2P Security, "P2P networking allows your network to be open to various forms of attack, break-in, espionage, and malicious mischief. P2P doesn't bring any novel threats to the network, just familiar threats such as worms and virus attacks." [10]

Worms are commonly spread through P2P networks; examples are NIMDA, Mandragore and Spybot. Gnutella is also associated with significant spyware risk; many Gnutella programs such as Limewire, Grokster, and Bearshare install spyware and adware by default, regardless of the options picked during installation. Common spyware includes "ClickTillUWin" and "DIder.exe". These Trojans will run in the background of systems and record and report on activity performed on the system. [11]

According to www.unwantedlinks.com, "Gnutella hosts, which are called servents, establish a TCP connection with each of the other servents on the Gnutella network. After the connection is made, the other Gnutella servents send their list of searches throughout the Gnutella network. This traffic can run between 4,500 and 5,300 bytes per second. " [12] This can create significant bandwidth consumption.

Correlations

More information on the behavior and risks of P2P networks can be found at the following links:
http://www.unwantedlinks.com/Guntella-alert.htm
http://ntrg.cs.tcd.ie/undergrad/4ba2.02-03/p10.html
http://www.secretmaker.com/update/filesharing/default.html
http://reform.house.gov/UploadedFiles/P2P%20Security%20Report.pdf

[10] Murphy, Declan; Kelly, Jarlath; Curley, Keith; Vickery, John; O'keefe, Dan. "P2P Security." Networks and Telecommunications Research Group. March 2002. URL: http://ntrg.cs.tcd.ie/undergrad/4ba2.02-03/p10.html (14 November 2004).
[11] "Gnutella Peer to Peer File Sharing Nets Users More Than MP3 Files!" Unwanted Links. URL: http://www.unwantedlinks.com/Guntella-alert.htm (13 November 2004).
[12] "Gnutella Peer to Peer File Sharing Nets Users More Than MP3 Files!" Unwanted Links. URL: http://www.unwantedlinks.com/Guntella-alert.htm (13 November 2004).

According to http://isc.sans.org/trends.php , the common Bearshare port, 6346, registers 14,808 average distinct sources per day, and 98,549 average distinct destinations per day in their collected dataset as of this writing.[13]

Evidence of active targeting

A variety of other signatures were fired from the same internal source IP address, suggesting that the system may have been compromised via one of the attack mechanisms reviewed above. Examination of these other alarms reveals that they are mostly related to http_inspect violations and many are benign triggers. This system should, however, be examined for signs of compromise.

Figure 5 is a graphical representation of the activity associated with the source IP address 78.37.212.28.

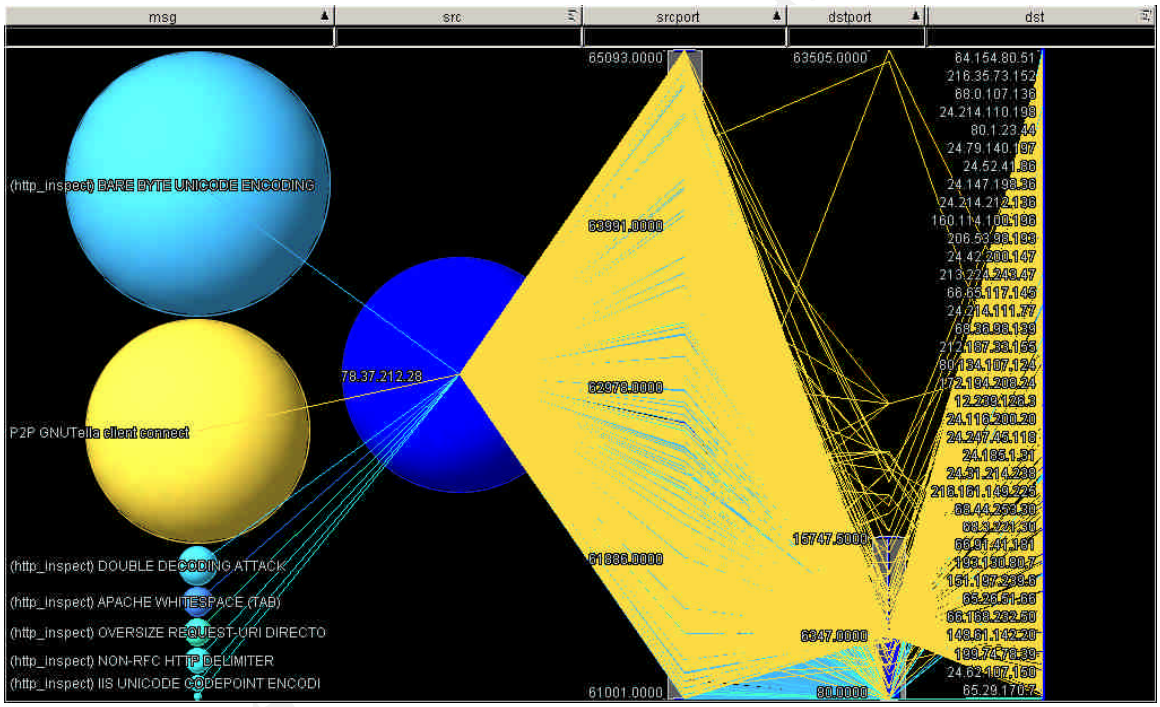

Figure 5

Severity

Severity = (criticality + lethality) - (system countermeasures + network countermeasures)

Severity = (1+4) - (1+2) = 2

Criticality = 1 There is nothing to suggest that the system generating the alarms is a critical system

[13] "Trends." Internet Storm Center. URL: http://isc.sans.org/trends.php (16 November 2004).

Lethality = 4 Many of the worms associated with P2P networks are very malicious and can cause system damage as well as infiltrate the internal network via the P2P network where they may wreak havoc. Spyware also poses a significant risk.

System Countermeasures = 1 System countermeasures are unknown, but do not appear to be in place.

Network Countermeasures = 2 P2P traffic is easy to detect with IDS, but can be difficult to filter at firewalls due to path discovery techniques built into the P2P applications and protocols. This traffic is obviously not contained.

Detect 3: DNS named version attempt

Description of detect

In the evaluated data set, 331 "DNS named version attempt" alerts fired from UDP packets with multiple source IP addresses, over multiple days, to a variety of destinations on both segments of the university's network. This is evidence of active scanning for BIND versions on multiple systems in the network

Reason this detect was selected

The volume of these events is one reason for further investigation. The fact that multiple source IP addresses, and many in the same class C address space, fired the alerts also raises the alert's interest level. There are numerous vulnerabilities in various versions of BIND, and results of a successful exploit of one of these vulnerabilities could result in Denial of Service, remote command execution, or information disclosure.[14]

Detect was generated by

This full alarm below was generated in Snort 2.2.0 running in IDS mode with csv output. Although the binary files are dated as being from April 2002, the timestamps in the files are from May 2002. For ease of analysis, the alarm has been put into a table with header rows.

```
snort -c /etc/snort/snort.conf -r 2002.4.21 -nnvX -A full
```

sig_id	id	sig_rev	Msg	timestamp

[14] "BIND Vulnerabilities." Internet Systems Consortium. 04 February 2004. URL: http://www.isc.org/index.pl?/sw/bind/bind-security.php (12 November 2004).

1616	312	6	DNS named version attempt				05/21-03:51:18.214488		
src		*srcport*	*Dst*		*dstport*	*proto*			
202.56.205.36		3244	78.37.60.98		53	UDP			
ethsrc	*ethdst*		*ethlen*	*ttl*	*tos*	*dgmlen*	*iplen*		
0:3:E3:D9:26:C0	0:0:C:4:B2:33		0x48	45	0	58	20		

An example packet, as displayed by Snort in non-IDS mode is below:

```
C:\Documents and Settings\Administrator\Desktop\GCIA4.0>c:\snort\bin\snort -vr
2002.4.21 > outtahere.txt

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/21-03:51:18.214488 202.56.205.36:3244 -> 78.37.60.98:53
UDP TTL:45 TOS:0x0 ID:312 IpLen:20 DgmLen:58
Len: 30
12 34 00 80 00 01 00 00 00 00 00 00 07 76 65 72  .4.............ver
73 69 6F 6E 04 62 69 6E 64 00 00 10 00 03        sion.bind.....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

The rule that triggered on this packet is below:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS named version attempt";
content:"|07|version"; offset:12; nocase; content:"|04|bind"; offset:12;
nocase; reference:arachnids,278; reference:nessus,10028; classtype:attempted-
recon; sid:1616; rev:6;)
```

This rule examines inbound network UDP traffic headed to port 53 and looks for the string “04” or “bind” starting in the 12th byte offset of the UDP header.

Probability the source address was spoofed

There is low probability that the source address is spoofed. This attack is reconnaissance and requires a response in order to be useful to the attacker.

Attack Mechanism

The Snort Signature Database explains this attack as follows: “An attacker can query a DNS server for the version of BIND running. Some versions of BIND, by default, respond to these queries while BIND version 9; by default, does not. A response to this query can assist an attacker in discovering servers that are potentially vulnerable to exploits associated with specific versions of BIND.”[15]

Running the Unix command 'dig @ns.com version.bind txt chaos' will execute this reconnaissance attack. [16]

[15] “DNS named version attempt.” Snort Signature Database. URL: http://www.snort.org/snort-db/sid.html?sid=1616 (10 November 2004).
[16] “DNS named version attempt.” Snort Signature Database. URL: http://www.snort.org/snort-db/sid.html?sid=1616 (10 November 2004).

The packets that fired these alerts can be classified as reconnaissance. The real concern would be subsequent attacks that attempt to exploit one of the many vulnerabilities present in various versions of BIND. These vulnerabilities range from information leakage to Denial of Service to remote execution of arbitrary code. [17]

Correlations

A list of various BIND Vulnerabilities, with associated vulnerable versions, can be found at the Internet Systems Consortium website
http://www.isc.org/index.pl?/sw/bind/bind-security.php

CERT® Advisory CA-1999-14 Multiple Vulnerabilities in BIND
http://www.cert.org/advisories/CA-1999-14.html

Secunia Advisories for BIND serious remote vulnerabilities:
http://secunia.com/advisories/7494/

According to http://isc.sans.org/trends.php , port 53 is a common target of scanning and an average of 25457 distinct sources scan this port per day in their collected data as of this writing.[18]

Evidence of active targeting

There are multiple external sources using this reconnaissance technique on the university's network. Some of these sources appear to have specific targets that they are querying and only fire a very low number of alerts, while other addresses are the source of multiple scans on multiple days. Figure 6 displays this information graphically.

Severity

Severity = (criticality + lethality) - (system countermeasures + network countermeasures)

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

Criticality = 3 This traffic impacted multiple hosts on the university's network over an extended period of time.

Lethality = 1 This is a pure reconnaissance detect at this time. The danger would be in subsequent attacks on systems found vulnerable in the scan.

[17] "BIND Vulnerabilities." Internet Systems Consortium. 04 February 2004. URL: http://www.isc.org/index.pl?/sw/bind/bind-security.php (12 November 2004).
[18] "Trends." Internet Storm Center. URL: http://isc.sans.org/trends.php (16 November 2004).

System Countermeasures = 1 System countermeasures are unknown.

Network Countermeasures = 2 Snort IDS is in place, however the port 53 UDP traffic appears to be allowed on all monitored segments of the network.

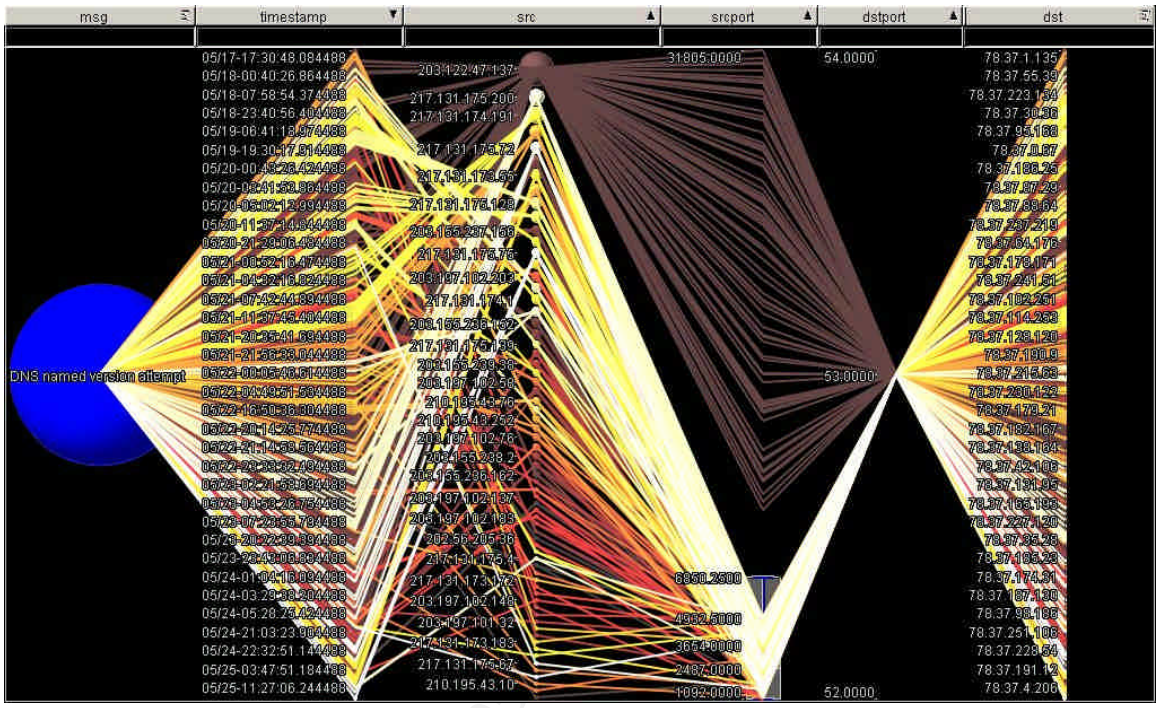

Figure 6

Network Statistics

Ports in use on the network

Table 2 is a result of analyzing all packets in the examined files and creating pivot charts with the data. The count is the number of packets with destination IP addresses in the university's internal ranges 78.37.0.0/24 and 226.185.0.0/24 with common or interesting destination ports. Port and service information is retrieved from treachery.net [19]

[19] "Treachery Unlimited." URL: http://www.treachery.net/tools/ports/lookup.cgi (15 November 2004).

Protocol	Port	Count	Common Services	Possible Malicious Services
TCP	21	575	File Transfer [Control]	Various
TCP	25	93	SMTP	Various
TCP	53	27	DNS	Various
TCP	80	613	HTTP	Various
TCP	137	4	Netbios-ns, trojans	[TROJAN] Chode, [TROJAN] Qaz
TCP	515	642	printer spooler	[TROJAN] ramen trojan, [TROJAN] lpdw0rm trojan
TCP	1080	124	socks	[TROJAN] SubSeven 2.2, [TROJAN] WinHole
TCP	3128	38	squid-http	[TROJAN] Reverse WWW Tunnel Backdoor, [TROJAN] RingZero
TCP	6346	10	gnutella (bearshare	
TCP	8080	489	HTTP Alternate (see port 80)	Various
TCP	20432	1		[TROJAN] Shaft
UDP	53	662	DNS	
UDP	5632	2	Pcanywherestat	

Table 2

Top Talkers

Inbound traffic statistics are a result of the number of packets destined to IP addresses in the university's network ranges, displayed in Table 3.
Outbound traffic statistics are a result of the number of packets sourced from IP addresses in the university's network range, displayed in Table 4.

Inbound Traffic (alarm target)	
Destination IP	Count
78.37.212.28	1,753
78.37.212.165	996
226.185.106.59	110
78.37.212.173	94
78.37.186.120	40
78.37.212.29	24
78.37.212.190	20

Table 4

Outbound Traffic (alarm source)	
Source IP	Count
78.37.212.28	15,069
78.37.212.165	65
226.185.106.176	23
226.185.106.59	2

Table 3

Suspicious External IP Profiles

Suspicious External IP Profile 1: 159.75.232.253

This source IP address is worthy of further investigation for the reasons listed below.

- "BAD-TRAFFIC udp port 0 traffic" Snort rule fired
- source port of 10000 is mildly suspicious
- destination port of 0 is much more suspicious

$ whois 159.75.232.253

```
OrgName:    LTX Corporation/Design Automation
OrgID:      LCA-1
Address:    LTX Park at University Avenue
City:       Westwood
StateProv:  MA
PostalCode: 02090
Country:    US

NetRange:   159.75.0.0 - 159.75.255.255
CIDR:       159.75.0.0/16
NetName:    LTX-COM
NetHandle:  NET-159-75-0-0-1
Parent:     NET-159-0-0-0-0
NetType:    Direct Assignment
NameServer: NS-240A.LTX.COM
NameServer: COMMX.LTX.COM
Comment:
RegDate:    1992-03-17
Updated:    2002-08-21

TechHandle: ZL12-ARIN
TechName:   LTX Corporation/Design Automation
TechPhone:  +1-781-467-5000
TechEmail:  is_admin@ltx.com

# ARIN WHOIS database, last updated 2004-11-15 19:10
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

Packet Sample:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/21-12:04:47.094488 159.75.232.253:10000 -> 78.37.212.28:0
UDP TTL:48 TOS:0x0 ID:49454 IpLen:20 DgmLen:464
Len: 436
4F 86 42 6F 00 00 00 01 01 01 01 01 01 01 01 01  O.Bo............
A9 55 BE 5C 2E 53 4D FF 34 B3 9B 11 3F D9 1A 0B  .U.\.SM.4...?...
86 68 A8 F0 9A 8E 9B 25 C8 89 DA AF 34 BA C5 C4  .h.....%....4...
BD F4 B6 D7 5B C6 2F 6D A9 A5 B9 D8 EB 73 FE 2A  ....[./m.....s.*
31 D2 34 34 0C 62 E5 40 17 BA 00 BB 4E F3 B8 10  1.44.b.@....N...
E1 0E 9E 6B 7F C9 12 A1 3E 2A 9B D2 DB F7 1A 05  ...k....>*......
3E 89 DB F9 62 57 9B 7D 9C 60 64 15 70 BF 33 8C  >...bW.}.`d.p.3.
F9 7D 37 65 4F 5D 48 29 3D 52 4B AB 6E C9 BB 4C  .}7eO]H)=RK.n..L
55 62 9D D8 FA A8 FF 3C 8B C2 BF 9F 51 9A CA 29  Ub.....<....Q..)
24 4F 12 30 16 32 28 69 B4 26 B7 BA AE E1 AC 24  $O.0.2(i.&.....$
1E 4A E1 1B 98 B9 6C D6 BC 28 D8 20 2B A6 44 74  .J....l..(. +.Dt
77 9E BB CC A1 DB 1A B7 40 5F 7F EA 65 55 9C 43  w.......@_..eU.C
5D 7D 61 51 B4 37 AE 10 5F 89 A9 E3 F2 5F 35 DA  ]}aQ.7.._...._5.
1A 29 9D 70 1B 6D A9 C4 E1 42 36 1A 20 1D 1C E0  .).p.m...B6. ...
F0 10 4F 46 93 82 E3 B5 F6 C7 5F 4A 59 8D 0B 27  ..OF......_JY..'
8D 60 85 36 F2 DA 81 A8 CC 0A 02 82 D6 48 13 51  .`.6.........H.Q
68 A1 99 0A 9A 1E 17 96 0D 6D 1A 96 18 D8 37 AC  h........m....7.
7A 47 13 52 FA 50 90 89 AA B1 5E E8 28 1E 09 17  zG.R.P....^.(...
59 0B B4 FB BC 3D DA 0E E3 03 E3 DB 83 CE ED 1D  Y....=..........
C0 59 A5 D9 A0 42 13 BF C8 47 2A C9 C9 5A CE D7  .Y...B...G*..Z..
5E 99 0E A8 76 15 BE 80 38 E5 F1 36 36 DD DC 98  ^...v...8..66...
B4 11 9B 38 1A B7 4C 4A 56 90 7C 01 09 6D 08 CD  ...8..LJV.|..m..
35 F4 CF 64 10 E5 27 53 BB EF 62 A1 FD 59 72 B8  5..d..'S..b..Yr.
D9 63 BF 42 88 4A D0 9E 43 DC 2B A2 CB D7 DB 24  .c.B.J..C.+....$
16 A3 3E 2E 58 F0 B5 F4 B2 A2 02 6B 4E 74 81 6F  ..>.X......kNt.o
```

```
70 A8 5F 81 BA EE CE 17 0D 8C 62 79 B8 C8 12 EF  p._.......by....
DB E6 BA 83 DD 59 23 52 8E 0C 8C D8 32 18 E5 8A  .....Y#R....2...
F1 28 EE 2A                                      .(.*

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Suspicious External IP Profile 2: 204.146.167.81

This source IP address is worthy of further investigation for the reasons listed below.

- TCP traffic from high source port (48741) to high destination port (63673) is not synonymous with "normal" traffic.
- Payload of packet is carrying evidence of a win32 kernel dump
- This traffic fired NOOP Shellcode x86 alerts

Operating System guess: according to the honeynet.org reference, the window size of 65535 (0xFFFF) corresponds with Cisco IOS 11.2[20]

```
$ whois 204.146.167.81

OrgName:    AT&T Global Network Services
OrgID:      ATGS
Address:    3200 Lake Emma Road
City:       Lake Mary
StateProv:  FL
PostalCode: 32746
Country:    US

NetRange:   204.146.0.0 - 204.146.255.255
CIDR:       204.146.0.0/16
NetName:    ATT-204-146-0-0-C
NetHandle:  NET-204-146-0-0-1
Parent:     NET-204-0-0-0-0
NetType:    Direct Allocation
NameServer: NS1.US.PRSERV.NET
NameServer: NS01.CA.US.IBM.NET
Comment:
RegDate:
Updated:    2004-07-20

OrgAbuseHandle: ATTAB-ARIN
OrgAbuseName:   ATT Abuse
OrgAbusePhone:  +1-919-319-8130
OrgAbuseEmail:  abuse@att.net

OrgTechHandle: ICC-ARIN
OrgTechName:   IP Customer Care
OrgTechPhone:  +1-888-613-6330
OrgTechEmail:  qhoang@att.com
```

[20] Spitzner, Lance. "Lists of fingerprints for passive fingerprint monitoring." 23 May, 2000. URL: http://project.honeynet.org/papers/finger/traces.txt (15 Nov. 2004).

```
# ARIN WHOIS database, last updated 2004-11-15 19:10
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

Packet Samples:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/20-11:28:36.264488 204.146.167.81:48741 -> 78.37.212.28:63673
TCP TTL:43 TOS:0x10 ID:12605 IpLen:20 DgmLen:1488
***A**** Seq: 0xFA94A4DB  Ack: 0x568ED7B9  Win: 0xFFFF  TcpLen: 20
55 00 83 C4 04 85 DB C6 44 3A 04 00 74 11 53 E8  U.......D:..t.S.
EC EA FF FF 83 C4 04 83 C8 FF 5F 5E 5D 5B C3 33  .........._^][.3
C0 5F 5E 5D 5B C3 5F 5E 5D C7 05 20 74 41 00 09  ._^][._^].. tA..
00 00 00 C7 05 24 74 41 00 00 00 00 00 83 C8 FF  .....$tA........
5B C3 90 90 90 90 90 90 90 90 90 90 90 90 90 90  [...............
56 8B 74 24 08 8B 46 0C A8 83 74 25 A8 08 74 21  V.t$..F...t%..t!
8B 46 08 50 E8 37 9F FF FF 8B 46 0C 83 C4 04 25  .F.P.7....F....%
.......
00 00 00 00 28 6E 75 6C 6C 29 00 00 00 00 00 00  ....(null)......
00 00 00 00 00 00 F0 3F 49 73 50 72 6F 63 65 73  .......?IsProces
73 6F 72 46 65 61 74 75 72 65 50 72 65 73 65 6E  sorFeaturePresen
74 00 00 00 4B 45 52 4E 45 4C 33 32 00 00 00 00  t...KERNEL32....
00 00 00 00 00 00 00 00 65 2B 30 30 30 00 00 00  ........e+000...
72 75 6E 74 69 6D 65 20 65 72 72 6F 72 20 00 00  runtime error ..
0D 0A 00 00 54 4C 4F 53 53 20 65 72 72 6F 72 0D  ....TLOSS error.
0A 00 00 00 53 49 4E 47 20 65 72 72 6F 72 0D 0A  ....SING error..
00 00 00 00 44 4F 4D 41 49 4E 20 65 72 72 6F 72  ....DOMAIN error
0D 0A 00 00 52 36 30 32 38 0D 0A 2D 20 75 6E 61  ....R6028..- una
62 6C 65 20 74 6F 20 69 6E 69 74 69 61 6C 69 7A  ble to initializ
65 20 68 65 61 70 0D 0A 00 00 00 00 52 36 30 32  e heap......R602
37 0D 0A 2D 20 6E 6F 74 20 65 6E 6F 75 67 68 20  7..- not enough
73 70 61 63 65 20 66 6F 72 20 6C 6F 77 69 6F 20  space for lowio
69 6E 69 74 69 61 6C 69 7A 61 74 69 6F 6E 0D 0A  initialization..
00 00 00 00 52 36 30 32 36 0D 0A 2D 20 6E 6F 74  ....R6026..- not
20 65 6E 6F 75 67 68 20 73 70 61 63 65 20 66 6F   enough space fo
72 20 73 74 64 69 6F 20 69 6E 69 74 69 61 6C 69  r stdio initiali
7A 61 74 69 6F 6E 0D 0A 00 00 00 00 52 36 30 32  zation......R602
35 0D 0A 2D 20 70 75 72 65 20 76 69 72 74 75 61  5..- pure virtua
6C 20 66 75 6E 63 74 69 6F 6E 20 63 61 6C 6C 0D  l function call.
0A 00 00 00 52 36 30 32 34 0D 0A 2D 20 6E 6F 74  ....R6024..- not
20 65 6E 6F 75 67 68 20                           enough

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Suspicious External IP Profile 3: 128.248.77.252

This source IP address is worthy of further investigation for the reasons listed below.

- Fired Snort alarms for “MISC Source Port 20 to <1024”
- 8 Syn packets to different hosts all w/ same timestamp
- Source port of 20 to destination port of 21

OS guess: information is inconsistent with fingerprint information found at honeynet.org,[21] however judging from the ttl of 243 and the DF setting, a guess is that this is Cisco IOS 12.0.

```
$ whois 128.248.77.252

OrgName:    University of Illinois at Chicago
OrgID:      UIAC
Address:    Computer Center
Address:    1940 West Taylor Avenue
City:       Chicago
StateProv:  IL
PostalCode: 60612
Country:    US

NetRange:   128.248.0.0 - 128.248.255.255
CIDR:       128.248.0.0/16
NetName:    UIC-NET
NetHandle:  NET-128-248-0-0-1
Parent:     NET-128-0-0-0-0
NetType:    Direct Assignment
NameServer: UIC-DNS1.UIC.EDU
NameServer: FRED.EECS.UIC.EDU
NameServer: GARCON.EECS.UIC.EDU
Comment:
RegDate:
Updated:    1993-06-24

TechHandle: EZ3-ARIN
TechName:   Zawacki, Edward
TechPhone:  +1-312-996-0658
TechEmail:  edz@uic.edu

# ARIN WHOIS database, last updated 2004-11-15 19:10
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

Packet Samples:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/24-05:20:17.504488 128.248.77.252:20 -> 78.37.215.210:21
TCP TTL:243 TOS:0x0 ID:54507 IpLen:20 DgmLen:40 DF
******S* Seq: 0xA3E10249  Ack: 0x0  Win: 0x3FFF  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 28 D4 EB 40 00 F3 06 04 3E 80 F8 4D FC 4E 25  .(..@....>..M.N%
0x0020: D7 D2 00 14 00 15 A3 E1 02 49 00 00 00 00 50 02  .........I....P.
0x0030: 3F FF 1A EA 00 00 00 00 00 00 00 00              ?...........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Analysis Process

Quoting the README on the raw file repository, “The log files are the result of a Snort instance running in binary logging mode. This means that only the packets

[21] Spitzner, Lance. “Lists of fingerprints for passive fingerprint monitoring.” 23 May, 2000. URL: http://project.honeynet.org/papers/finger/traces.txt (15 Nov. 2004).

that violate the ruleset will appear in the log."[22] No additional scrubbing or modification of the data in the files was performed as it is also specified in the README that "All of the IP addresses of the protected network space have been 'munged'". [23]

There is no way to easily emulate the rule set that was originally in use when the packets of interest were recorded. Rather than reverse engineer the ruleset, the rules below were added directly to snort.conf:

```
alert tcp any any <> any any (msg:"tcp traffic";)
alert udp any any <> any any (msg:"udp traffic";)
```

These additional rules enable the analyst to reconstruct a more complete picture of the network being audited. In this analysis, Snort 2.2.0 was configured to use the csv output plugin, which results in comma separated data that is easily portable for the application of analysis tools.

The Snort default ruleset plus the generic tcp and udp rules mentioned above resulted in over 22,000 rows of data. This data, in csv format, was then analyzed with a variety of tools.

My primary workstation for performing much of the analysis is a Windows XP platform; therefore many of the tools used in analysis are win32 based and GUI oriented. For Windows, I used the most recent version of Snort available, 2.2.0. I had difficulty generating alarms from the replay data files with this version of Snort initially and after many frustrating hours of Google-ing for assistance, a colleague discovered that adding the line, "config checksum_mode: none" to the snort.conf file would resolve this issue. I also had challenges with the version of WinPcap required for different versions of the tools I was using, such as Ethereal, Windump, and Snort. I attacked this rather inefficiently by keeping copies of the installs for different versions of Winpcap handy on my system. I also used a SUSE Linux 9.0 installation running Snort 2.0.1, tcpdump version 3.7.2 / libpcap version 0.7.2, and other native *nix tools to assist in some of the analysis. I found that the 2.0.1 version of Snort processed the capture files differently and would often result in different alerts than those generated on my Windows systems – I did not spend the time to dig into every discrepancy, but found this largely due to a different default ruleset and different features in Snort, such as the preprocessor configuration.

As for applications used, I spent a lot of time with "grep" and "awk" and print and a variety of bpf filters to dig through the data. I found myself falling back on more GUI and Win32 related tools to produce the slightly more polished results used in this paper. Microsoft Excel 2002 was used for simple arrangement and sorting of the data as well as for creating many of the tables, charts, and graphs used in

[22] "README." 05 April 2004. http://isc.sans.org/logs/Raw/README (15 November 2004).
[23] "README." 05 April 2004. http://isc.sans.org/logs/Raw/README (15 November 2004).

analysis. BrioQuery Explorer 6.5.2 was also used for analysis by easily filtering and arranging data, and creating pivot tables and charts. Visual Insight's Advizor Workbench 3.5 (http://www.advizorsolutions.com/) was used to create the Parabox link graphics -- this is an excellent data visualization tool and I used this heavily when arranging data looking for obscure patterns and anomalous traffic. I was able to use this valuable tool, which is very costly, by the good graces of my employer. Ethereal was used extensively to view packet details and retrieve information, both on Windows and Linux platforms. I found that the Ethereal version 0.10.7 had a merge function that could combine multiple capture files easier than tcpdump.

In conclusion, there are significant challenges with analyzing binary capture file without the benefit of knowing the ruleset used originally to capture the data. These challenges include the fact that only partial conversations can be reassembled in many cases and that the rules originally violated are not available to identify what was suspicious about the traffic to begin with. Data visualization tools can be beneficial in this case, and are a great way to begin the analysis process as the graphical representation of the data enables the application of anomaly based analysis techniques. The data analyzed in this practical is minor in comparison to volumes processed in many enterprise or MSSP environments, yet the same analysis techniques will scale to support much larger data set.

References

Gaspar, Suzanne. "Security concerns dominate NW500 survey." Network World. 05 May, 2001. URL: http://www.nwfusion.com/research/2001/0507feat2.html (15 November 2004).

"CERT/CC Statistics 1988-2004." CERT Coordination Center. 19 October 2004. URL: http://www.cert.org/stats/cert_stats.html (15 November 2004).

Gordon, Les. "What is the Q Trojan?" SANS FAQ. URL: http://www.sans.org/resources/idfaq/qtrojan.php (15 November 2004).

"Treachery Unlimited." URL: http://www.treachery.net/tools/ports/lookup.cgi (15 November 2004).

FILE-SHARING PROGRAMS AND PEER-TO-PEER NETWORKS PRIVACY AND SECURITY RISKS." UNITED STATES HOUSE OF REPRESENTATIVES COMMITTEE ON GOVERNMENT REFORM. May 2003. URL: http://reform.house.gov/UploadedFiles/P2P%20Security%20Report.pdf (15 November 2004).

Murphy, Declan; Kelly, Jarlath; Curley, Keith; Vickery, John; O'keefe, Dan. "P2P Security." Networks and Telecommunications Research Group. March 2002. URL: http://ntrg.cs.tcd.ie/undergrad/4ba2.02-03/p10.html (14 November 2004).

"Gnutella Peer to Peer File Sharing Nets Users More Than MP3 Files!" Unwanted Links. URL: http://www.unwantedlinks.com/Guntella-alert.htm (13 November 2004).

Spitzner, Lance. "Lists of fingerprints for passive fingerprint monitoring." 23 May, 2000. URL: http://project.honeynet.org/papers/finger/traces.txt (15 Nov. 2004).

"README." 05 April 2004. http://isc.sans.org/logs/Raw/README (15 November 2004).

"Trends." Internet Storm Center. URL: http://isc.sans.org/trends.php (16 November 2004).

Gilmore, John. "RFC 951 - Bootstrap Protocol." Internet RFC/STD/FYI/BCP Archives. September 1985. URL: http://www.faqs.org/rfcs/rfc951.html (13 November 2004).

Mogul, Jeffrey. "RFC 919 - Broadcasting Internet Datagrams." October 1984. URL: http://cert.uni-stuttgart.de/archive/intrusions/2003/01/msg00495.html (16 November 2004).

Thomas, Rob. "Bogon Dotted Decimal List v2.5 02 AUG 2004." URL: http://www.cymru.com/Documents/bogon-dd.html (16 November 2004).

Advizor Solutions, Inc. URL: http://www.advizorsolutions.com/ (16 November 2004).