Global Information Assurance Certification Paper

GCIA CERTIFICATION PRACTICAL ASSIGNMENT

Version 3.0

TRENTON RIDDELL

March 6, 2002

Assignment #1 – Describe the State of Intrusion Detection

Making the Case for Intrusion Detection

Trenton Riddell
January 4, 2002

Introduction

As the Internet has grown so has the number and severity of attacks against its networks. While Internet worms like Code Red, Code Red II and Nimda make headline news, the latest email virus is snaking its way through corporate email systems. New vulnerabilities and attacks seem to be popping up more and more frequently. With all this activity going on, you'd think that security in-depth would be an easy sell to IT managers and C-level executives. Unfortunately, that's often not the case.

Network managers are saddled with the ever-increasing responsibility for securing the enterprise networks while trying to justify the rising costs to the executive. Couple that with the fact that intrusion detection is one of those things where you really don't know if, or to what extent, you are under attack until you start investigating.

Any astute corporate CFO will understand ROI based on proper risk analysis and asset valuation. But, how do you justify a system of prevention when the risk can be difficult to pin down, and what you're trying to protect is often somewhat intangible? How bad would it appear if your systems were being used to launch a Denial of Service attack against a business partner or customer? How much is the corporate reputation worth? Many IT managers will readily proclaim that they have never been compromised, they don't have Trojans on their systems, and they aren't being attacked. But, how do they know? Those same managers are often horrified to find out the sheer volume and severity of the attacks that occur on a daily basis. Risk Management is a central concept to information systems management. So why should intrusion detection be any different? Approaching intrusion detection as a method of risk management armed with statistics, rather than a security measure is a far better way to ensure the integrity of your environment and get the needed endorsement from the executive.

Setting the stage

How do you justify an Intrusion Detection system? It is easy to simply point the executive to sites like www.dshield.org to help illustrate the point of just how real Internet threats are, however these sites will hold little meaning to a non-technical manager without some kind of frame of reference. What is port 80? Do we have an FTP server? Are we vulnerable to this attack?

So how do you demonstrate the business value of an IDS to the Executive? The way I've done it, and I know many others have as well, is to install SNORT on a spare workstation and place it outside the firewall. It's free, it's stable, and most importantly SNORT doesn't require too much time or customization to get it working and providing usable data in a short timeframe. This works great to represent the volume of nasty traffic that's going on out there, but the drawback of

this approach is that the IDS is going to pick up all kinds of stuff that is simply not applicable. That is, you will see lots of attacks but if the ports aren't open or the hosts being attacked aren't accessible, the risk to the enterprise is low and that's not going to help you make your case.

The advantage to this method is that you get a good understanding of the kind of traffic your systems may be seeing and you don't need to invest a lot of time or money to get it in place. You can then take that information and turn it into statistics that your executive will understand. The key is being able to show how real the threat is, and how relevant that threat may be to your organization.

Why would someone attack us?

No doubt when you first set out to explain why an IDS would be a good thing for your organization, you are going to be asked this question, "Why would someone want to attack your systems?" Being able to answer it is important. Obviously, different types of companies are going to be attacked more often and hit harder than others. Often the answer is as simple as determining how big a target you are, what industry you are in, and how successful you are.

In January of 2002, information security consulting firm Riptech, [1] released an Internet Security Threat Report detailing attack trends for the last two quarters of 2001. In their report, Riptech provided attack trend analysis based on the size and type of company. According to that report, high tech and financial companies are attacked most often, while power and energy companies suffer the most severe attacks. Larger companies tend to have larger, more complex networks that offer more viable targets for attackers. Public companies tend to have greater visibility in the media due to advertising and are therefore attacked more than nonprofit or private firms. All of these issues need to be considered when evaluating the risk posture of your environment.

To determine your level of risk, it is helpful to create a corporate risk profile. When defining the types of criteria that go into your corporate profile and level of risk, you need to look at it objectively from the hacker's perspective with emphasis on risk management. Are you easy prey or a challenge? If the hacker was successful in compromising your systems, what could he exploit? What would the impact(s) of those exploits be to your organization?

Identify Vulnerable Systems

Before you can begin to identify and understand potential threats, and be able to quantify and qualify the level of risk, you need to know your own strengths and weaknesses. It is important to develop a baseline before you can begin to evaluate your risk posture.

In IT terms, you need to know what OS your hosts are running, what patches have been applied, what services are running on those hosts and what (if any) countermeasures are in place before you can determine if you are at risk from a specific attack. For example, you may see a lot of attacks directed at BIND services. However, if you aren't running that service, or if you are but it isn't a version that is vulnerable to the attack, the risk is low.

If you don't have a clear understanding of your environment you will be unable to assess the severity of *any* attack. If that is the case, every attack is severe simply because of all the unknowns.

Providing Metrics

Once you have an understanding of what type of attacks your systems are vulnerable to, and how severe those vulnerabilities may be, you can begin to create some metrics around your risk posture. It's important to remember who your audience may be at this point. Any IDS is going to generate false positives. Upper level managers or directors may not have a technical background, so while you want to be detailed in your presentation, the inclusion of those false positives in your metrics may give the wrong message. Distinguish the false positives from the actual attacks and, in the cases where there are genuine attacks, determine the intensity and severity of those attacks. There are many tools available like Snort2HTML, SnortSnarf or the ARIS analyzer from SecurityFocus that you can use to help correlate the data and generate meaningful reports.

In the reports that you are going to create, you'll want to provide enough details to make them meaningful but not too technical. Too much technical talk and the person you are trying persuade, the person that controls the purse strings, will simply tune out. You need to strike a balance that addresses the issues with relevant hard data. You want to get the point across that the threat is real, when there is a threat. Put emphasis on the several hundred scans and probes for SQL server or SNMP from China, rather than an obvious scan for BattleNet servers.

Some suggestions for the content include:

- The type of attacks logged.
- How many of each type?
- Who are the attackers (Country of origin)?
- What is the classification of the attack?
- Informational - Recon scans for vulnerable systems. Traffic dropped by the firewall.
- Warning. – Suspicious traffic. Possible compromise.
- Critical – Successful attack. Connection allowed but no compromise.
- Emergency – Successful attack. System compromised.
- What is the intent of the attack?

Trending

Graphs are an excellent way to show statistical data in a visual format. And it's a well-known fact that executive types find it easier to relate to graphs of all types; bar graphs, pie charts, line graphs. The great thing about graphs is that they make it very easy to show trends in data, and that is exactly what you want to do. You can't just hand someone a report out of the blue and say "Well, we had 500 scans last week." Is that a lot? You need to put the numbers in context.

One of the key things that will put you closer to getting the IDS that you know you need is being able to show trends in attacks over time. Trending allows you to illustrate the frequency of

specific attacks as well as changes in attack vectors over a broader timeline than day-to-day snapshots. One tool currently available that is great for this purpose is the ARIS analyzer from SecurityFocus.com [2]. After you upload your IDS logs to ARIS there are a number of "canned" reports that you can run against your data. They produce slick looking graphs that can help you show information on the number of attacks over a user defined period, the services that are being attacked, the country of origin and most importantly, attack trending.

Conclusions

Coming up with a sound argument around why your environment may need an Intrusion Detection System can be a difficult task. With IS budgets getting tighter every year, an IDS can seem a very expensive proposition. Enterprise class intrusion detection with both host and network based sensors can easily cost thousands of dollars. This is not including the added expense of staff, preferably trained security experts, to monitor and maintain them. At what point do the benefits justify the cost?

The most successful approach I've found is to introduce IDS as a function of risk management and mitigation armed with clear metrics. Define the threats based on your corporate risk profile and identify your vulnerabilities with how they relate to confidentiality, availability, and integrity. If you can, install a machine running SNORT on your perimeter and start to gather attack data. Separate the false positives from the actual attacks and begin to compile the data into meaningful reports. These reports should demonstrate things like the most common attacks, most active attacker, attack classification and any possible trends.

I can't guarantee that by following the steps and suggestions outlined that you will be successful in convincing your executive that some kind of IDS is necessary. However, at the very least you may be able to bring about an awareness of the very real threats and attacks that enterprises connected to the Internet are subject to every day. Good luck.

References

[1] Belcher, Tim etal. Riptech Internet Security Threat Report. Riptech, Inc. January, 2002 http://www.riptech.com/pdfs/Security%20Threat%20Report.pdf

[2] Allen, Julia H. The CERT Guide to System and Network Security Practices. ISBN 0-201-73723, Addison-Wesley, May 2001

[3] Pipkin, Donald L. Information Security, Protecting the Global Enterprise. Prentice Hall Canada Inc., 2000. ISBN 0-13-017323-1

[4] Blacharski, Dan. Network Security in a Mixed Environment. ISBN: 0-764-531522, IDG Books, 1998.

[5] McHugh, John, etal. Defending Yourself: The Role of Intrusion Detection Systems. September/October 2001.

Assignment #2 - Network Detects

Detect #1 - Broadcast to Printer Port

SNORT Alert

```
[**] BACKDOOR Q access [**]
01/07-11:09:40.390759 255.255.255.255:31337 -> my.net.53.145:515
TCP TTL:13 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 2B 00 00 00 00 0D 06 E5 B9 FF FF FF FF XX XX  .+..............
0x0020: 35 91 7A 69 02 03 00 00 00 00 00 00 00 00 50 14  5.zi..........P.
0x0030: 00 00 98 E1 00 00 63 6B 6F 00 00 00              ......cko..
```

When I started seeing this alert, I set up Windump to capture additional traffic. I've also been seeing these with real IP's and SYNs instead of the RST/ACK but, these just seem more interesting.

Supporting Windump log

```
15:54:02.080085 255.255.255.255.31337 > my.net.27.181.515: R 0:3(3) ack 0 win 0
0x0000  4500 002b 0000 0000 0e06 fe95 ffff ffff      E..+............
0x0010  XXXX 1bb5 7a69 0203 0000 0000 0000 0000      ....zi..........
0x0020  5014 0000 b2bd 0000 636b 6f00 0000           P.......cko...

16:49:59.288884 255.255.255.255.31337 > my.net.133.118.515: R 0:3(3) ack 0 win 0
0x0000  4500 002b 0000 0000 0d06 95d4 ffff ffff      E..+............
0x0010  XXXX 8576 7a69 0203 0000 0000 0000 0000      ...vzi..........
0x0020  5014 0000 48fc 0000 636b 6f00 0000           P...H...cko...

<snip>

18:41:02.714418 255.255.255.255.31337 > my.net.102.210.515: R 0:3(3) ack 0 win 0
0x0000  4500 002b 0000 0000 0d06 b478 ffff ffff      E..+.......x....
0x0010  XXXX 66d2 7a69 0203 0000 0000 0000 0000      ..f.zi..........
0x0020  5014 0000 67a0 0000 636b 6f00 0000           P...g...cko...

19:57:05.973015 255.255.255.255.31337 > my.net.177.34.515: R 0:3(3) ack 0 win 0
0x0000  4500 002b 0000 0000 0d06 6a28 ffff ffff      E..+......j(....
0x0010  XXXX b122 7a69 0203 0000 0000 0000 0000      ..."zi..........
0x0020  5014 0000 1d50 0000 636b 6f00 0000           P....P..cko...

(….. many more follow)
```

Source of Trace

The source of this trace and initial detect was a PC running SNORT v1.8.3-WIN32 (build 88) located outside the perimeter router on my network.

Detect Generated By

The detect was generated by SNORT rule:

```
alert tcp 255.255.255.0/24 any -> $HOME_NET any (msg:"BACKDOOR Q access"; flags:A+;
dsize: >1;  reference:arachnids,203; sid:184;  classtype:misc-activity; rev:3;)
```

The Windump log was captured with the command line filter:
Windump -n -X -s 1514 "src port 31337 and dst port 515" > broadcast.txt

Probability the Source Address Was Spoofed

Extremely high. The source address is a network broadcast address.

Attack Description

This would seem to be reconnaissance scans looking for hosts that might be vulnerable to LPR/LPRng holes. The packet obviously crafted. The sequence number is always 0, it uses the script kiddie favorite "elite" port and the source address is the network broadcast address. Not only is there no attempt to be stealthy, broadcast traffic shouldn't even make it pass most perimeter routers.

Putting aside the obvious crafting for a minute, the attacker is sending a packet with both the ACK and RST bits set to destination port 515. The bit combination would suggest that a previous SYN had been sent from my network to a closed port on the attacking host. All of the destination IPs in the attack are valid IPs in my address space, however all outbound traffic though the firewall NATed. This could not be part of previous communication since the addresses are never allowed out. A quick grep of the firewall log confirms that there was no previous outbound communication from any of the destination addresses.

Attack Mechanism

The intent would be seem to be to elicit some response from the destination host indicating that port 515 on the destination host is active and listening. This would then allow the attacker to begin more active targeting against that host to see if it was vulnerable to holes like buffer overflows in the LPR.

The attacker sends packets with both an ACK and RST flag. However, in order for the attack to generate any kind of response the victim must not only respond to a RST but also respond to a packet with the broadcast address as the source. I don't know of any implementation of TCP that would do that. The RST flag terminates the connection. When a host receives a packet with a RST flag it should just silently discard it.

My best guess at this point is that this is an exploit that is looking for host that may have unpatched LPR holes and this traffic is coming from poorly configured attack tool or maybe even an attempt at a worm.

Correlation

I have found similar attacks documented in both the Incidents.org site and 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

Evidence of Active Targeting

Low. The scan was directed at the entire address space although it was for a specific service.

Severity

Target Criticality: 2
Attack Lethality: 1
System Countermeasures: 3 (Our systems are not vulnerable to this attack)
Network Countermeasures: 5 (broadcast traffic is dropped at the firewall)

Severity: (2 + 1) - (3 + 5) = -5

Defense Recommendations

No changes are necessary to defend against this attack from an external source. The perimeter router and firewall reject inbound/outbound broadcast traffic and access to TCP/515. However, this does not preclude the possibility of a better-orchestrated internal attack. An audit of all internal UNIX hosts to determine exposure to lpd vulnerabilities is recommended.

Question 1

An unsolicited RST-ACK should generate what response?

A. ACK
B. SYN
C. None
D. SYN/ACK

Answer: C. The packet should be silently dropped.

Detect #2 SYN-FIN SCAN

Source 1 - SNORT

```
01/04-21:37:43.005000  [**] [100:1:1] spp_portscan: PORTSCAN DETECTED to port 21 from
161.132.145.130 (STEALTH) [**]

01/04-21:37:43.003880  [**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan)
detection
[**] {TCP} 161.132.145.130:21 -> MY.NET.0.2:21

01/04-21:37:43.014453  [**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan)
detection
[**] {TCP} 161.132.145.130:21 -> MY.NET.0.3:21

01/04-21:37:43.023846  [**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan)
detection
[**] {TCP} 161.132.145.130:21 -> MY.NET.0.4:21

01/04-21:37:43.033689  [**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan)
detection
[**] {TCP} 161.132.145.130:21 -> MY.NET.0.5:21
```

<SNIP>

```
01/04-21:48:35.802544  [**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan)
detection
[**] {TCP} 161.132.145.130:21 -> MY.NET.255.253:21

01/04-21:48:35.812296  [**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan)
detection
[**] {TCP} 161.132.145.130:21 -> MY.NET.255.254:21

01/04-21:48:35.821581  [**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan)
detection
[**] {TCP} 161.132.145.130:21 -> MY.NET.255.255:21

01/04-21:48:40.751000  [**] [100:2:1] spp_portscan: portscan status from
161.132.145.130: 178 connections across 178 hosts: TCP(178), UDP(0) STEALTH [**]

01/04-21:48:44.026000  [**] [100:3:1] spp_portscan: End of portscan from
161.132.145.130: TOTAL
time(653s) hosts(63685) TCP(63848) UDP(0) STEALTH [**]
```

SNORT Portscan log

```
Jan  4 21:37:42 161.132.145.130:21 -> MY.NET.0.1:21 SYNFIN ******SF
Jan  4 21:37:43 161.132.145.130:21 -> MY.NET.0.2:21 SYNFIN ******SF
Jan  4 21:37:43 161.132.145.130:21 -> MY.NET.0.3:21 SYNFIN ******SF

<snip>.....

Jan  4 21:48:35 161.132.145.130:21 -> MY.NET.255.253:21 SYNFIN ******SF
Jan  4 21:48:35 161.132.145.130:21 -> MY.NET.255.254:21 SYNFIN ******SF
Jan  4 21:48:35 161.132.145.130:21 -> MY.NET.255.255:21 SYNFIN ******SF
```

Source 2
A quick grep of the firewall logs to make sure they aren't getting through....

```
Jan  4 21:37:43 my.firewall Jan 05 2002 04:37:09: %PIX-6-106015: Deny TCP (no
connection)  from 161.132.145.130/21 to MY.NET.80.91/21 flags FIN SYN  on interface
outside

Jan  4 21:37:43 my.firewall Jan 05 2002 04:37:09: %PIX-3-106010: Deny inbound tcp src
outside:161.132.145.130/21 dst inside:MY.NET.80.92/21
```

<snip>

Source of Trace

The source of this trace and initial detect was a PC running SNORT v1.8.3-WIN32 (build 88) located outside the perimeter router on my network running with the command parameters of:

Snort -A fast -X -c snort.conf -l c:\snort183\snort\pigdox

The second source is from my PIX firewall.

Detect Generated By

The initial detects was generated by SNORT with **spp_stream4 preprocessor module** active with follow-up traces from PIX firewall logs.

Although, the spp_stream4 preprocessor picked this up, I would have thought this type of scanning would have triggered an alert from one of the following rules:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SYN FIN";flags:SF;
reference:arachnids,198; classtype:attempted-recon; sid:624; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN synscan portscan"; id: 39426;
flags: SF;reference:arachnids,441; classtype:attempted-recon; sid:630; rev:1;)
```

I'm not sure if this may be a flaw in SNORT or if it is by design that when the preprocessor is active, these rules are not fired. My understanding at this point is that if there were a single SYN-FIN scan the explicit rule would have triggered the alert, but since there multiple packets from the same source the Stream4 preprocessor was triggered.

Probability the Source Address Was Spoofed

Unlikely. The attacker needs to determine if a host is active and listening. Therefore, he must receive a response to the packets to determine which host(s) has the target port open.

Attack Description

The traffic seems to have the symptoms of a SynScan probe (SF flags, and src port == dst port), but without the ID and widow size I can't be sure. Also SNORT didn't trigger the specific rule for SynScan. That may indicate that NMAP or even Hping2 could have been used.

Regardless of the tool that was used to generate the traffic, this is a reconnaissance probe looking for hosts listening on port 21.

Attack Mechanism

The purposes of the SYN-FIN flag are to both get a response from a host and avoid detection. Because some systems allow FINs to pass though, the attacker is hoping that his packets will make it to the target. Also, because the FIN flag signals the end of the connection, some systems do not log these packets. The attacker is looking for a response that would indicate a potential target for more directed attack

Correlation

Doing a search on the web, I can across others that had similar scans as well as the same issues with SNORT and the preprocessor.

http://www.incidents.org/archives/intrusions/msg02801.html

http://www.incidents.org/archives/intrusions/msg02799.html

http://vanguard.hawkeye.ac/snort/193/190/58/src193.190.58.6.html

Looking up the IP address for attack correlation in online security sites failed to find any other reported victims. However, a general search for the IP address brought up a hit in the attrition.org defacement archives. It seems that the host is a web server in Peru and had been hacked in the past. We could probably safely assume that a hacker still has control, or has once again compromised the system and is being used to remotely scan my network.

http://archives.neohapsis.com/archives/defaced/2001-04/0156.html

Evidence of Active Targeting

The attack was directed that the address range not a specific target. He was looking for FTP servers but there is no evidence that this was a directed attack.

Severity

Target Criticality: 4 (This was a complete scan of all hosts listening on port 21.)
Attack Lethality: 4 (We have several FTP servers)
System Countermeasures: 3 (Known FTP servers are up-to-date. They may be rogue or unknown servers that could be vulnerable)
Network Countermeasures: 5 (The firewall is dropping the SF packets)

Severity: (4 + 4) - (3 + 5) = 0

Defense Recommendations

The defenses for this attack were fine; the firewall dropped the SF packets. But this does bring into question how many hosts may be listening on port 21 and how many of them have up-to-date patches. I would recommend an internal audit of the environment for FTP servers and a listing of relevant applied patches.

Question 2

What is usually the purpose of packets with a SYN-FIN flag combination?

A. Terminate an existing connection
B. Avoid IDS or firewall detection
C. Transmit data
D. None of the above

Answer: C. Since the FIN flag signals the end of a connection many systems do not log them. The SYN-FIN flag combination is often used to attempt to both get a response from a host and avoid detection.

Detect #3 – Amplification

SNORT ALERT

```
01/17-18:45:48.155561  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.0.0
01/17-18:45:48.159659  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.0.255
01/17-18:45:48.159890  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.1.0
01/17-18:45:48.165276  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.1.255
01/17-18:45:48.171554  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.2.0
01/17-18:45:48.178344  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.3.0
01/17-18:45:48.178875  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.2.255
01/17-18:45:48.182408  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.3.255
01/17-18:45:48.191030  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.4.255
01/17-18:45:48.192487  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.4.0
01/17-18:45:48.202841  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.5.0
01/17-18:45:48.204541  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.6.0
01/17-18:45:48.204783  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.5.255
01/17-18:45:48.213078  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.6.255
01/17-18:45:48.217610  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.7.0
01/17-18:45:48.217882  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.7.255
01/17-18:45:48.225653  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.8.255
01/17-18:45:48.226996  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.8.0
01/17-18:45:48.230723  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.9.0
01/17-18:45:48.239002  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.10.0
01/17-18:45:48.240069  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.9.255
01/17-18:45:48.243149  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.10.255
01/17-18:45:48.248661  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.11.0
01/17-18:45:48.253135  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.11.255
01/17-18:45:48.261641  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.12.0
01/17-18:45:48.262074  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.12.255
01/17-18:45:48.264978  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.13.0
01/17-18:45:48.269912  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.13.255
```

```
01/17-18:45:48.273848  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.14.0
01/17-18:45:48.283066  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.14.255
01/17-18:45:48.287136  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.15.0
01/17-18:45:48.291752  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 62.211.151.242 -> MY.NET.15.255

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.155561 62.211.151.242 -> MY.NET.0.0
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:0  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 E7 98 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 00 00 08 00 E7 59 10 A6 00 00 00 00 00 00 00 00  .....Y..........
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.159659 62.211.151.242 -> MY.NET.0.255
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:5  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 E6 99 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 00 FF 08 00 E2 59 10 A6 05 00 00 00 00 00 00 00  .....Y..........
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.191030 62.211.151.242 -> MY.NET.4.255
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:45  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 E2 99 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 04 FF 08 00 BA 59 10 A6 2D 00 00 00 00 00 00 00  .....Y..-.......
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............


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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.213078 62.211.151.242 -> MY.NET.6.255
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:65  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 E0 99 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 06 FF 08 00 A6 59 10 A6 41 00 00 00 00 00 00 00  .....Y..A.......
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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


[**] MISC Large ICMP Packet [**]
01/17-18:45:48.217610 62.211.151.242 -> MY.NET.7.0
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:70  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 E0 98 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 07 00 08 00 A1 59 10 A6 46 00 00 00 00 00 00 00  .....Y..F.......
```

```
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.217882 62.211.151.242 -> MY.NET.7.255
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:75  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 DF 99 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 07 FF 08 00 9C 59 10 A6 4B 00 00 00 00 00 00 00  .....Y..K.......
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.225653 62.211.151.242 -> MY.NET.8.255
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:85  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 DE 99 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 08 FF 08 00 92 59 10 A6 55 00 00 00 00 00 00 00  .....Y..U.......
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.226996 62.211.151.242 -> MY.NET.8.0
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:80  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 DF 98 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 08 00 08 00 97 59 10 A6 50 00 00 00 00 00 00 00  .....Y..P.......
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.230723 62.211.151.242 -> MY.NET.9.0
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:90  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 DE 98 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 09 00 08 00 8D 59 10 A6 5A 00 00 00 00 00 00 00  .....Y..Z.......
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.239002 62.211.151.242 -> MY.NET.10.0
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:100  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 DD 98 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 0A 00 08 00 83 59 10 A6 64 00 00 00 00 00 00 00  .....Y..d.......
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.240069 62.211.151.242 -> MY.NET.9.255
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:95  ECHO
```

```
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 DD 99 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 09 FF 08 00 88 59 10 A6 5F 00 00 00 00 00 00 00  .....Y.._.......
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.243149 62.211.151.242 -> MY.NET.10.255
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:105  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 DC 99 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 0A FF 08 00 7E 59 10 A6 69 00 00 00 00 00 00 00  ....~Y..i.......
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.248661 62.211.151.242 -> MY.NET.11.0
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:110  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 DC 98 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 0B 00 08 00 79 59 10 A6 6E 00 00 00 00 00 00 00  ....yY..n.......
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.253135 62.211.151.242 -> MY.NET.11.255
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:115  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 DB 99 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 0B FF 08 00 74 59 10 A6 73 00 00 00 00 00 00 00  ....tY..s.......
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.261641 62.211.151.242 -> MY.NET.12.0
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:120  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 DB 98 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 0C 00 08 00 6F 59 10 A6 78 00 00 00 00 00 00 00  ....oY..x.......
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.262074 62.211.151.242 -> MY.NET.12.255
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:125  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 DA 99 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 0C FF 08 00 6A 59 10 A6 7D 00 00 00 00 00 00 00  ....jY..}.......
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
```

```
01/17-18:45:48.264978 62.211.151.242 -> MY.NET.13.0
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:130  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 DA 98 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 0D 00 08 00 65 59 10 A6 82 00 00 00 00 00 00 00  ....eY..........
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.269912 62.211.151.242 -> MY.NET.13.255
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:135  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 D9 99 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 0D FF 08 00 60 59 10 A6 87 00 00 00 00 00 00 00  ....`Y..........
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.273848 62.211.151.242 -> MY.NET.14.0
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:140  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 D9 98 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 0E 00 08 00 5B 59 10 A6 8C 00 00 00 00 00 00 00  ....[Y..........
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.283066 62.211.151.242 -> MY.NET.14.255
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:145  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 D8 99 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 0E FF 08 00 56 59 10 A6 91 00 00 00 00 00 00 00  ....VY..........
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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

[**] MISC Large ICMP Packet [**]
01/17-18:45:48.287136 62.211.151.242 -> MY.NET.15.0
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:150  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 D8 98 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 0F 00 08 00 51 59 10 A6 96 00 00 00 00 00 00 00  ....QY..........
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............

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


[**] MISC Large ICMP Packet [**]
01/17-18:45:48.291752 62.211.151.242 -> MY.NET.15.255
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:155  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 D7 99 3E D3 97 F2 XX XX  ....@.*...>.....
0x0020: 0F FF 08 00 4C 59 10 A6 9B 00 00 00 00 00 00 00  ....LY..........
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............
```

=+

Source of Trace

The source of this trace and initial detect was a PC running SNORT v1.8.3-WIN32 (build 88) located outside the perimeter router on my network running with the command parameters of:

Snort -A fast -X -c snort.conf -l c:\snort183\snort\pigdox

Detect Generated By

The alert was tiggered by the following rule in the Misc ruleset for SNORT.

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"MISC Large ICMP Packet"; dsize:
>800; reference:arachnids,246; classtype:bad-unknown; sid:499; rev:1;)
```

Probability the Source Address Was Spoofed

High. There is a very good probability that the source address was spoofed. ICMP does not require a three-way handshake. If the source address was not spoofed, and my network responded to ICMP echo requests, the result would be a DoS against the source host.

Attack Description

It would appear as though someone is trying to use my network as a broadcast amplifier in a SMURF-like DoS attack against host 62.211.151.242. The ICMP echo requests are most likely forged and are directed at the IP broadcast addresses.

Attack Mechanism

Older IP stacks used the first address (0) in a subnet as the broadcast address; most new IP stacks use the last address (255). A SMURF attack depends on the broadcast address acting as an amplifier. A single echo request could result in many echo replies. If the attacker sends multiple echo requests to multiple amplifier networks the resulting echo replies overwhelm the victim and prevent it from processing any other network traffic effectively creating a denial of service.

There are three parities to a SMURF attack; the attacker, the intermediary and the victim. The attacker directs ICMP echo request packets to the IP broadcast addresses on an Intermediary Network from the spoofed source address of the victim. When the Intermediary receives the echo request to the IP broadcast address of its network, the result is that potentially every host on the broadcast network will reply with an ICMP echo reply packet to the source address.

Correlation

Smurf attacks are explained in detail in CERT Advisory CA-1998-01
http://www.cert.org/advisories/CA-1998-01.html

Checked http://www.netscan.org/ to verify that my site is acting as ICMP amplifier.

Evidence of Active Targeting

There isn't really any active targeting of my network, however the victim is being actively targeted.

Severity

Target Criticality: 1
Attack Lethality: 3 (If successful, the result could been a DoS on the remote host.)
System Countermeasures: 1 (Internal hosts do respond to ICMP echo requests.)
Network Countermeasures: 5 (The perimeter router and firewall do not allow respond to inbound ICMP)

Severity: (4 + 3) - (1 + 5) = 2

Defense Recommendations

Our defenses are reasonable. We are blocking inbound and outbound ICMP on the firewall and IP-directed broadcasts are disabled at the perimeter routers.

Question #3

In the following Smurf attack, who is the intended victim?

```
01/17-18:45:48.159659 172.16.151.242 -> 192.168.0.255
ICMP TTL:42 TOS:0x0 ID:0 IpLen:20 DgmLen:28 DF
Type:8  Code:0  ID:42512   Seq:5  ECHO
0x0000: 08 00 02 06 F0 AD 00 60 47 40 FA E9 08 00 45 00  .......`G@....E.
0x0010: 00 1C 00 00 40 00 2A 01 E6 99 AC 10 97 F2 C0 A8  ....@.*...>.....
0x0020: 00 FF 08 00 E2 59 10 A6 05 00 00 00 00 00 00 00  .....Y..........
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00              ............
```

A. Destination Host
B. Destination Network
C. Source Host
D. None of the above

Answer: C. The spoofed source address is the actual target of Smurf attack. The broadcast responses are directed at the spoofed source creating a potential denial of service.

Detect 4 - RPC Scan

SNORT ALERT

```
01/14-15:57:57.673000  [**] [100:1:1] spp_portscan: PORTSCAN DETECTED from
63.195.129.179 (THRESHOLD 4 connections exceeded in 0 seconds) [**]
01/14-15:58:01.108000  [**] [100:2:1] spp_portscan: portscan status from
63.195.129.179: 374 connections across 374 hosts: TCP(374), UDP(0) [**]
01/14-15:58:05.034000  [**] [100:2:1] spp_portscan: portscan status from
63.195.129.179: 583 connections across 583 hosts: TCP(583), UDP(0) [**]
01/14-15:58:09.030000  [**] [100:2:1] spp_portscan: portscan status from
63.195.129.179: 513 connections across 513 hosts: TCP(513), UDP(0) [**]
01/14-15:58:14.087000  [**] [100:2:1] spp_portscan: portscan status from
63.195.129.179: 196 connections across 196 hosts: TCP(196), UDP(0) [**]
01/14-15:58:25.283000  [**] [100:3:1] spp_portscan: End of portscan from
63.195.129.179: TOTAL time(14s) hosts(1663) TCP(1666) UDP(0) [**]
```

PORTSCAN LOG.

```
Jan 14 15:57:57 63.195.129.179:4088 -> MY.NET.0.0:111 SYN ******S*
Jan 14 15:57:57 63.195.129.179:4089 -> MY.NET.0.1:111 SYN ******S*
Jan 14 15:57:57 63.195.129.179:4090 -> MY.NET.0.2:111 SYN ******S*
Jan 14 15:57:57 63.195.129.179:4091 -> MY.NET.0.3:111 SYN ******S*
Jan 14 15:57:57 63.195.129.179:4092 -> MY.NET.0.4:111 SYN ******S*
Jan 14 15:57:57 63.195.129.179:4093 -> MY.NET.0.5:111 SYN ******S*
Jan 14 15:57:57 63.195.129.179:4094 -> MY.NET.0.6:111 SYN ******S*
Jan 14 15:57:57 63.195.129.179:4095 -> MY.NET.0.7:111 SYN ******S*
Jan 14 15:57:57 63.195.129.179:4139 -> MY.NET.0.51:111 SYN ******S*
Jan 14 15:57:57 63.195.129.179:4207 -> MY.NET.0.119:111 SYN ******S*
Jan 14 15:57:57 63.195.129.179:4277 -> MY.NET.0.189:111 SYN ******S*
Jan 14 15:57:57 63.195.129.179:4342 -> MY.NET.0.254:111 SYN ******S*
```

<snip>

```
Jan 14 15:58:11 63.195.129.179:3835 -> MY.NET.14.151:111 SYN ******S*
Jan 14 15:58:11 63.195.129.179:3836 -> MY.NET.14.152:111 SYN ******S*
Jan 14 15:58:11 63.195.129.179:3838 -> MY.NET.14.154:111 SYN ******S*
Jan 14 15:58:11 63.195.129.179:3839 -> MY.NET.14.155:111 SYN ******S*
Jan 14 15:58:11 63.195.129.179:3841 -> MY.NET.14.157:111 SYN ******S*
Jan 14 15:58:11 63.195.129.179:3842 -> MY.NET.14.158:111 SYN ******S*
Jan 14 15:58:11 63.195.129.179:3844 -> MY.NET.14.160:111 SYN ******S*
```

<snip>

The attacker continued to scan the whole Class B address range.

Windump

I had been seeing quite a lot of RPC traffic, from several different sources. I set up Windump to capture the traffic.

```
15:57:57.582427 63.195.129.179.4088 > MY.NET.0.0.111: S 2737409014:2737409014(0) win
32120 <mss 1460,sackOK,timestamp 45262313 0,nop,wscale 0> (DF)
0x0000  4500 003c 14df 4000 3106 e0e3 3fc3 81b3      E..<..@.1...?...
0x0010  XXXX 0000 0ff8 006f a329 8bf6 0000 0000      .......o.)......
0x0020  a002 7d78 8e72 0000 0204 05b4 0402 080a      ..}x.r..........
0x0030  02b2 a5e9 0000 0000 0103 0300                ............
```

```
15:57:57.589426 63.195.129.179.4089 > MY.NET.0.1.111: S 2732319457:2732319457(0) win
32120 <mss 1460,sackOK,timestamp 45262313 0,nop,wscale 0> (DF)
0x0000  4500 003c 14e0 4000 3106 e0e1 3fc3 81b3      E..<..@.1...?...
0x0010  XXXX 0001 0ff9 006f a2db e2e1 0000 0000      .......o........
0x0020  a002 7d78 37d3 0000 0204 05b4 0402 080a      ..}x7...........
0x0030  02b2 a5e9 0000 0000 0103 0300                ............

15:57:57.595300 63.195.129.179.4090 > MY.NET.0.2.111: S 2744534115:2744534115(0) win
32120 <mss 1460,sackOK,timestamp 45262313 0,nop,wscale 0> (DF)
0x0000  4500 003c 14e1 4000 3106 e0df 3fc3 81b3      E..<..@.1...?...
0x0010  XXXX 0002 0ffa 006f a396 4463 0000 0000      .......o..Dc....
0x0020  a002 7d78 d594 0000 0204 05b4 0402 080a      ..}x............
0x0030  02b2 a5e9 0000 0000 0103 0300                ............

15:57:57.600418 63.195.129.179.4091 > MY.NET.0.3.111: S 2731407469:2731407469(0) win
32120 <mss 1460,sackOK,timestamp 45262313 0,nop,wscale 0> (DF)
0x0000  4500 003c 14e2 4000 3106 e0dd 3fc3 81b3      E..<..@.1...?...
0x0010  XXXX 0003 0ffb 006f a2cd f86d 0000 0000      .......o...m....
0x0020  a002 7d78 2251 0000 0204 05b4 0402 080a      ..}x"Q..........
0x0030  02b2 a5e9 0000 0000 0103 0300                ............
```

<snip>

```
16:01:14.792770 63.195.129.179.1329 > MY.NET.254.30.111: S 2955857684:2955857684(0)
win 32120 <mss 1460,sackOK,timestamp 45282100 0,nop,wscale 0> (DF)
0x0000  4500 003c 3d01 4000 3106 baa2 3fc3 81b3      E..<=.@.1...?...
0x0010  XXXX fe1e 0531 006f b02e cf14 0000 0000      .....1.o........
0x0020  a002 7d78 fdab 0000 0204 05b4 0402 080a      ..}x............
0x0030  02b2 f334 0000 0000 0103 0300                ...4........

16:01:14.795233 63.195.129.179.1330 > MY.NET.254.31.111: S 2949265877:2949265877(0)
win 32120 <mss 1460,sackOK,timestamp 45282100 0,nop,wscale 0> (DF)
0x0000  4500 003c 3d02 4000 3106 baa0 3fc3 81b3      E..<=.@.1...?...
0x0010  XXXX fe1f 0532 006f afca 39d5 0000 0000      .....2.o..9.....
0x0020  a002 7d78 934d 0000 0204 05b4 0402 080a      ..}x.M..........
0x0030  02b2 f334 0000 0000 0103 0300                ...4........

16:01:14.804321 63.195.129.179.1331 > MY.NET.254.32.111: S 2941861741:2941861741(0)
win 32120 <mss 1460,sackOK,timestamp 45282100 0,nop,wscale 0> (DF)
0x0000  4500 003c 3d03 4000 3106 ba9e 3fc3 81b3      E..<=.@.1...?...
0x0010  XXXX fe20 0533 006f af59 3f6d 0000 0000      .....3.o.Y?m....
0x0020  a002 7d78 8e24 0000 0204 05b4 0402 080a      ..}x.$..........
0x0030  02b2 f334 0000 0000 0103 0300                ...4........

16:01:14.807272 63.195.129.179.1332 > MY.NET.254.33.111: S 2945066776:2945066776(0)
win 32120 <mss 1460,sackOK,timestamp 45282100 0,nop,wscale 0> (DF)
0x0000  4500 003c 3d04 4000 3106 ba9c 3fc3 81b3      E..<=.@.1...?...
0x0010  XXXX fe21 0534 006f af8a 2718 0000 0000      ...!.4.o..'.....
0x0020  a002 7d78 a646 0000 0204 05b4 0402 080a      ..}x.F..........
0x0030  02b2 f334 0000 0000 0103 0300                ...4........
```

<snip> … Many more follow.

Source of Trace

The source of this trace and initial detect was a PC running SNORT v1.8.3-WIN32 (build 88) located outside the perimeter router on my network running with the command parameters of:

Snort -A fast -X -c snort.conf -l c:\snort183\snort\pigdox

Additional trace is provided by Windump

Detect Generated By

This detect was generated by Snort's portscan preprocessor which detects UDP packets or TCP SYN packets directed at four different port in less than three seconds. The preprocessor can be enabled in Snort config file with the following line:

Preprocessor portscan: $HOME_NET 4 3 portscan.log

Probability the Source Address Was Spoofed

Low. The traffic is all TCP SYNs; attacker is expecting a response.

Attack Description

RPC portmapper maps the ports at which services are running. RPC (Remote Procedure Call) allows programs on one host to execute programs on other hosts. They are most often used to access network services. There are multiple vulnerabilities documented for RPC vulnerabilities and over the past year, TCP 111 was one of the most probed for ports.

Attack Mechanism

This is was a fairly simple, fast and noisy scan for RPC portmapper services. The attacker was looking for a specific port and therefore we might conclude that is was reconnaissance or general network mapping in preparation for a follow-up attack. Once the attacker had a list of hosts that are listening running the service, he would target those hosts for additional probing to determine the OS and what specific vulnerabilities that each may vulnerable to. In many cases RPC services are vulnerable to buffer overflows, which could grant the attacker root access.

Correlation

The following CERT advisories detail the attacks and vulnerable systems.

Vulnerability in ToolTalk RPC Service:
http://www.cert.org/advisories/CA-1998-11.html

Format String Vulnerability in CDE ToolTalk
http://www.cert.org/advisories/CA-2001-27.html

Buffer Overflow Vulnerability in rpc.cmsd
http://www.cert.org/advisories/CA-1999-08.html

Input Validation problem in rpc.statd
http://www.cert.org/advisories/CA-2000-17.html

Evidence of Active Targeting

There was no evidence of active targeting. This was a simple network mapping, reconnaissance attempt for systems running RPC portmapper..

Severity

Target Criticality: 4
Attack Lethality: 4 (If successful, the attacker could execute a root compromise)
System Countermeasures: 3 (Modern OS, although the patch level is uncertain)
Network Countermeasures: 5 (The firewall blocked the attempt)

Severity: (4+4) - (3 +5) = 0

Defense Recommendations

For the most part our defenses are adequate. The firewall is blocking access to any systems that might be vulnerable. However, that does not rule out an attack from an inside source. Because of that, I recommend that all Unix systems be assessed to their patch level and all applicable patches be applied ASAP.

Question #4

RPC Portmapper typically runs at what port?

A. 137
B. 110
C. 111
D. 445

Answer: If portmapper is running on a host it usually resides at port 111.

Detect 5 - Stealthy SNMP SCAN

```
01/19-20:47:52.345000  [**] [100:1:1] spp_portscan: PORTSCAN DETECTED from
140.171.179.168 (THRESHOLD 4 connections exceeded in 0 seconds) [**]
01/19-20:47:57.333000  [**] [100:2:1] spp_portscan: portscan status from
140.171.179.168: 16 connections across 16 hosts: TCP(0), UDP(16) [**]
01/19-20:48:02.390000  [**] [100:3:1] spp_portscan: End of portscan from
140.171.179.168: TOTAL time(0s) hosts(16) TCP(0) UDP(16) [**]

Jan 19 20:47:52 140.171.179.168:4189 -> MY.NET.0.1:161 UDP
Jan 19 20:47:52 140.171.179.168:4189 -> MY.NET.1.1:161 UDP
Jan 19 20:47:52 140.171.179.168:4189 -> MY.NET.2.1:161 UDP
Jan 19 20:47:52 140.171.179.168:4189 -> MY.NET.4.1:161 UDP
Jan 19 20:47:52 140.171.179.168:4189 -> MY.NET.3.1:161 UDP
Jan 19 20:47:52 140.171.179.168:4189 -> MY.NET.5.1:161 UDP
Jan 19 20:47:52 140.171.179.168:4189 -> MY.NET.6.1:161 UDP
Jan 19 20:47:52 140.171.179.168:4189 -> MY.NET.8.1:161 UDP
Jan 19 20:47:52 140.171.179.168:4189 -> MY.NET.7.1:161 UDP
Jan 19 20:47:52 140.171.179.168:4189 -> MY.NET.9.1:161 UDP
Jan 19 20:47:52 140.171.179.168:4189 -> MY.NET.10.1:161 UDP
Jan 19 20:47:52 140.171.179.168:4189 -> MY.NET.12.1:161 UDP
Jan 19 20:47:52 140.171.179.168:4189 -> MY.NET.13.1:161 UDP
Jan 19 20:47:52 140.171.179.168:4189 -> MY.NET.11.1:161 UDP
Jan 19 20:47:52 140.171.179.168:4189 -> MY.NET.14.1:161 UDP
Jan 19 20:47:52 140.171.179.168:4189 -> MY.NET.15.1:161 UDP
```

A day later…

```
01/20-06:16:30.345000  [**] [100:1:1] spp_portscan: PORTSCAN DETECTED from
140.171.179.168 (THRESHOLD 4 connections exceeded in 0 seconds) [**]
01/20-06:16:57.614000  [**] [100:2:1] spp_portscan: portscan status from
140.171.179.168: 16 connections across 16 hosts: TCP(0), UDP(16) [**]
01/20-06:17:47.716000  [**] [100:3:1] spp_portscan: End of portscan from
140.171.179.168: TOTAL time(0s) hosts(16) TCP(0) UDP(16) [**]

Jan 20 06:16:30 140.171.179.168:4189 -> MY.NET.2.2:161 UDP
Jan 20 06:16:30 140.171.179.168:4189 -> MY.NET.0.2:161 UDP
Jan 20 06:16:30 140.171.179.168:4189 -> MY.NET.1.2:161 UDP
Jan 20 06:16:30 140.171.179.168:4189 -> MY.NET.5.2:161 UDP
Jan 20 06:16:30 140.171.179.168:4189 -> MY.NET.3.2:161 UDP
Jan 20 06:16:30 140.171.179.168:4189 -> MY.NET.6.2:161 UDP
Jan 20 06:16:30 140.171.179.168:4189 -> MY.NET.4.2:161 UDP
Jan 20 06:16:30 140.171.179.168:4189 -> MY.NET.7.2:161 UDP
Jan 20 06:16:30 140.171.179.168:4189 -> MY.NET.8.2:161 UDP
Jan 20 06:16:30 140.171.179.168:4189 -> MY.NET.10.2:161 UDP
Jan 20 06:16:30 140.171.179.168:4189 -> MY.NET.11.2:161 UDP
Jan 20 06:16:30 140.171.179.168:4189 -> MY.NET.12.2:161 UDP
Jan 20 06:16:30 140.171.179.168:4189 -> MY.NET.9.2:161 UDP
Jan 20 06:16:30 140.171.179.168:4189 -> MY.NET.13.2:161 UDP
Jan 20 06:16:30 140.171.179.168:4189 -> MY.NET.14.2:161 UDP
Jan 20 06:16:30 140.171.179.168:4189 -> MY.NET.15.2:161 UDP
```

Then again the next day!

```
01/20-15:45:15.294000  [**] [100:1:1] spp_portscan: PORTSCAN DETECTED from
140.171.179.168 (THRESHOLD 4 connections exceeded in 0 seconds) [**]
01/20-15:45:26.069000  [**] [100:2:1] spp_portscan: portscan status from
140.171.179.168: 16 connections across 16 hosts: TCP(0), UDP(16) [**]
```

```
01/20-15:45:48.612000  [**] [100:3:1] spp_portscan: End of portscan from
140.171.179.168: TOTAL time(0s) hosts(16) TCP(0) UDP(16) [**]

Jan 20 15:45:15 140.171.179.168:4189 -> MY.NET.0.3:161 UDP
Jan 20 15:45:15 140.171.179.168:4189 -> MY.NET.2.3:161 UDP
Jan 20 15:45:15 140.171.179.168:4189 -> MY.NET.1.3:161 UDP
Jan 20 15:45:15 140.171.179.168:4189 -> MY.NET.3.3:161 UDP
Jan 20 15:45:15 140.171.179.168:4189 -> MY.NET.4.3:161 UDP
Jan 20 15:45:15 140.171.179.168:4189 -> MY.NET.5.3:161 UDP
Jan 20 15:45:15 140.171.179.168:4189 -> MY.NET.6.3:161 UDP
Jan 20 15:45:15 140.171.179.168:4189 -> MY.NET.7.3:161 UDP
Jan 20 15:45:15 140.171.179.168:4189 -> MY.NET.8.3:161 UDP
Jan 20 15:45:15 140.171.179.168:4189 -> MY.NET.9.3:161 UDP
Jan 20 15:45:15 140.171.179.168:4189 -> MY.NET.10.3:161 UDP
Jan 20 15:45:15 140.171.179.168:4189 -> MY.NET.11.3:161 UDP
Jan 20 15:45:15 140.171.179.168:4189 -> MY.NET.12.3:161 UDP
Jan 20 15:45:15 140.171.179.168:4189 -> MY.NET.13.3:161 UDP
Jan 20 15:45:15 140.171.179.168:4189 -> MY.NET.14.3:161 UDP
Jan 20 15:45:15 140.171.179.168:4189 -> MY.NET.15.3:161 UDP
```

A check of the WINDUMP log reveals:

```
20:47:52.321916 140.171.179.168.4189 > MY.NET.0.1.161:  GetNextRequest(37)
.1.3.6.1.2.1.1.1 .1.3.6.1.2.1.1.3[|snmp]
0x0000  4500 0109 b6e9 0000 6c11 c422 8cab b3a8      E.......l..".....
0x0010  XXXX 0001 105d 00a1 00f5 b195 3081 ea02      .....]......0...
0x0020  0100 0406 7075 626c 6963 a181 dc02 0100      ....public......
0x0030  0201 0002 0100 3081 d030 0b06 072b 0601      ......0..0...+..
0x0040  0201 0101 0500 300b 0607 2b06 0102 0101      ......0...+.....
0x0050  0305                                         ..

20:47:52.322216 140.171.179.168.4189 > MY.NET.1.1.161:  GetNextRequest(37)
.1.3.6.1.2.1.1.1 .1.3.6.1.2.1.1.3[|snmp]
0x0000  4500 0109 b6ea 0000 6c11 c321 8cab b3a8      E.......l..!....
0x0010  XXXX 0101 105d 00a1 00f5 b095 3081 ea02      .....]......0...
0x0020  0100 0406 7075 626c 6963 a181 dc02 0100      ....public......
0x0030  0201 0002 0100 3081 d030 0b06 072b 0601      ......0..0...+..
0x0040  0201 0101 0500 300b 0607 2b06 0102 0101      ......0...+.....
0x0050  0305                                         ..
```

<snip>

```
20:47:56.341308 140.171.179.168.4189 > MY.NET.254.1.161:  GetNextRequest(37)
.1.3.6.1.2.1.1.1 .1.3.6.1.2.1.1.3[|snmp]
0x0000  4500 0109 b7eb 0000 6c11 c51f 8cab b3a8      E.......l.......
0x0010  XXXX fe01 105d 00a1 00f5 b394 3081 ea02      .....]......0...
0x0020  0100 0406 7075 626c 6963 a181 dc02 0100      ....public......
0x0030  0201 0002 0100 3081 d030 0b06 072b 0601      ......0..0...+..
0x0040  0201 0101 0500 300b 0607 2b06 0102 0101      ......0...+.....
0x0050  0305                                         ..

20:47:56.342803 140.171.179.168.4189 > MY.NET.255.1.161:  GetNextRequest(37)
.1.3.6.1.2.1.1.1 .1.3.6.1.2.1.1.3[|snmp]
0x0000  4500 0109 b7ec 0000 6c11 c41e 8cab b3a8      E.......l.......
0x0010  XXXX ff01 105d 00a1 00f5 b294 3081 ea02      .....]......0...
0x0020  0100 0406 7075 626c 6963 a181 dc02 0100      ....public......
0x0030  0201 0002 0100 3081 d030 0b06 072b 0601      ......0..0...+..
0x0040  0201 0101 0500 300b 0607 2b06 0102 0101      ......0...+.....
0x0050  0305
```

Source of Trace

The source of this trace and initial detect was a PC running SNORT v1.8.3-WIN32 (build 88) located outside the perimeter router on my network running with the command parameters of:

Snort -A fast -X -c snort.conf -l c:\snort183\snort\pigdox

Detect Generated By

This detect was generated by Snort's portscan preprocessor which detects UDP packets or TCP SYN packets directed at four different port in less than three seconds. The preprocessor can be enabled in Snort config file with the following line:

Preprocessor portscan: $HOME_NET 4 3 portscan.log

Probability the Source Address Was Spoofed

Low. The IP is a valid and live. The attacker needs a response in order to continue the attack.

Attack Description

This attack is an attempt to query SNMP information using the default community string of "public". Many implementations of SNMP use a default community string of "public" which often remains unchanged.

There are numerous documented vulnerabilities in SNMP. The successful exploitation of these may allow an unauthorized person to gain privileged access to core network servers and devices, create a denial of service or cause the network to become unstable.

Attack Mechanism

This is an attempt to access the "public" SNMP community string on the network. The attack appears to be scripted; the attempts are very close together, the source port is always the same. The attacker also appears to be attempting to be stealthy in his attacks by only targeting 16 hosts at a time.

Correlation

There are many SNMP vulnerability advisories. The most recent and widespread is CA-2002-03:

http://www.cert.org/advisories/CA-2002-03.html

A lookup of the IP using www.dshield.org/ipinfo.php verifies that this IP has been used to attack other networks

Evidence of Active Targeting

Yes. The attacker is actively scanning the hosts 1,2,3 in the first fifteen subnets of the Class B. Often, network planners will allocate the first few IP addresses of a subnet to core network devices (switches, routers, firewalls). Many of these devices run SNMP for management. It would seem that this attacker is browsing for network devices with the default community string.

Severity

Target Criticality: 5 (The targets seem to be core network devices)
Attack Lethality: 4 (The attack is a reconnaissance scan)
System Countermeasures: 1 (The community string on many core network devices and hosts has not been changed from the default of public.)
Network Countermeasures: 3 (All traffic on SNMP ports is dropped by the perimeter firewall, but SNMP is used internally)

Severity: (5+4) - (1+3) = 5

Defense Recommendations

There are several very easy things you can do to protect you network from SNMP attacks:

- In order to prevent SNMP from being exploited from outside your network, UDP 161 and 162 should be blocked at the perimeter both inbound and outbound.
- If SNMP is not being used or is not needed, disable the service.
- All SNMP community strings should be changed from the default to something that is not obvious.
- Community strings should be password protected with strong passwords for both Read-only and Read-write access.
- Run an SNMP scanner like SNMPwalk against your network to find SNMP agents.
- Create a custom IDS rule to pick up SNMP attacks
- Apply all patches ASAP.

Question #5

SNMP uses which protocol for communication?

A. TCP
B. UDP
C. ICMP
D. All of the above

Answer: B. SNMP uses UDP for communication

3.0 - Analyze This

3.1 - Executive Summary

This section will be an analysis of 5 days of Snort logs, which will result in a summary of alerts, detects by severity and volume, top 10 talkers, and information on selected external hosts. The analyst will attempt to identify possible system compromise, attack trending, general areas of concern and defensive recommendations where applicable.
Files Analyzed

Alerts, Scans and out-of-spec (OOS) files from January 6, 2002 to January 10, 2002 were downloaded and analyzed. These were not chosen at random, rather they were chosen based on the type of organization and a general understanding of the user base to try to get a good sampling of traffic.

It was my understanding that the source files were from a college or university environment. It was my belief that because of this, if data were taken from several weeks prior to the time students would have been away for the holidays, and the traffic patterns would not have been an accurate sampling of day-to-day activity. The dates were chosen based on the assumption that at the time of the sampling the students should have been back at school although classes may not have started yet, creating an opportunity to capture relevant data over a short timeframe. I felt that this would be a more legitimate sampling and would be more relevant than if data were taken from a prior period.

3.2 - File Selection

A complete listing of source data is as follows:

Alerts	*Scans*	*Out-of-Spec*
Alert020106.ids	Scans020106	Oos_Jan6-2002
Alert020107.ids	Scans020107	Oos_Jan7-2002
Alert020108.ids	Scans020108	Oos_Jan8-2002
Alert020109.ids	Scans020108	Oos_Jan9-2002
Alert020110.ids	Scans020110	Oos_Jan10-2002

3.3 - Analysis Process

Taking a cue from the GCIA practical assignments of others, I started the analysis using PERL based Snort log-processing tool called SNORTSNARF. However it was quickly found that this tool produced HTML reports that were approximately 15 MB in size. Opening these on my analysis workstation was found to be very process intensive given the hardware that was available, and with limited analysis capabilities since "drilling down" often took almost as long to as it took to open the file.

It became apparent that I needed to off load the processing of the logs. As an alternative, I decided to use the ARIS Analyzer available free online from SecurityFocus (http://aris.securityfocus.com). However, I was concerned that the submitted logs may be used the to generate aggregated incident and attack statistics and that this data may skew those numbers. SecurityFocus was contacted and I was assured that the sample data would not affect their reporting.

First attempts to upload the raw data resulted in only the external addresses being extracted and the internal MY.NET.x.x addresses were ignored. It was determined that this was due to the analyzer only recognizing numeric fields in the dotted decimal format. MY.NET would have to be replaced with bogus addresses.

The analysis workstation from which I was working is on the Windows platform, which lacks any built in command line tools to find and replace text within files. Since each of the downloaded alert and scan files was approximately 15-20MB in size I needed something to automate the conversion. A search on the web for some kind of freeware or shareware tool to assist with the conversion did not turn up an adequate tool to assist with the task. I decided to do it the hard way and opened the logs in a text editor and began the slow process of running a find/replace for MY.NET with 10.1.

Once the logs had been altered they were uploaded to ARIS where the data was sorted and canned reports generated using the built in reporting engine.

3.4 - Top Talkers

Using the compiled data, the following reports were generated using the ARIS reporting engines. These graphs are intended to show the top ten most active IP addresses by volume of incidents including probes. It show again be noted that the IP addresses of the supplied data were changed from sanitized MY.NET to 10.1. All addresses in the 10.1.x.x range are internal.

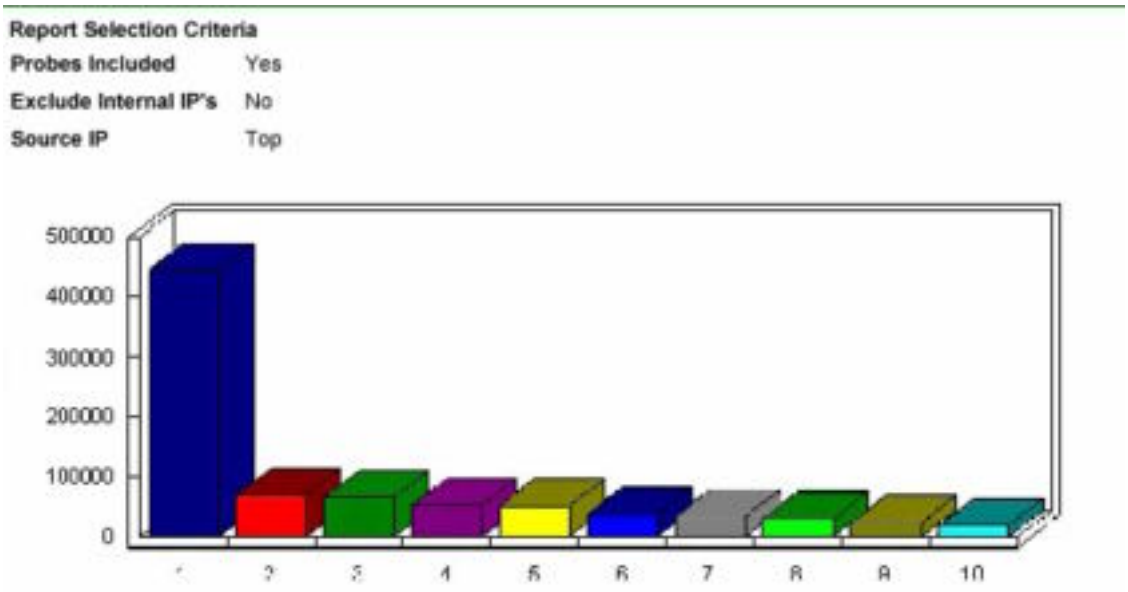


Figure 1.1

Figure 1.1: The above graph the Y-axis represents the volume of events (including probes), while the X-axis represents the responsible IP (See Figure 1.2)

Figure 1.2

Performing some analysis on some on top talkers we can see that there is some probing for known Trojans. This may indicate that these hosts have already been compromised and are attempting to propagate.

10...60.43	Internal Address	0	-	High port 65535 udp - possible Red Worm - traffic
10...60.43	Internal Address	0	27512	Generic UDP Fortacon Probe

Looking at the number one top talker we see a single probe to a high port as well as quite a bit of probing on UDP.

Red Worm, aka Adore, infects Linux systems using vulnerabilities in BIND, wu-ftpd, rpc.statd and lpd services. These are the same vulnerabilities that were used by the Ramen and Lion worms.

Red Worm scans for vulnerable hosts on Class B subnets. When a vulnerable host is found, it attempts to download the main worm part from a web server located in China, in a similar way that Lion worm does.

[illegible]	Internal Address	[illegible]	1	Back Orifice BackDoor Probe
[illegible]	Internal Address	[illegible]	363	High port 60333 udp possible Red Worm traffic
[illegible]	Internal Address	[illegible]	7272	Generic UDP Portscan Probe

Here we see the same type of activity from the #5 top talker with the addition of a Back Orifice probe.

When we run the same report excluding probes we see a significant shift in both the volume of events and the attacking hosts. Clearly, this indicates there is a large volume of probing activity occurring on your network. The vast majority of which is originating from hosts within your network.

Report Selection Criteria

Probes Included	No
Exclude Internal IP's	No
Source IP	Top

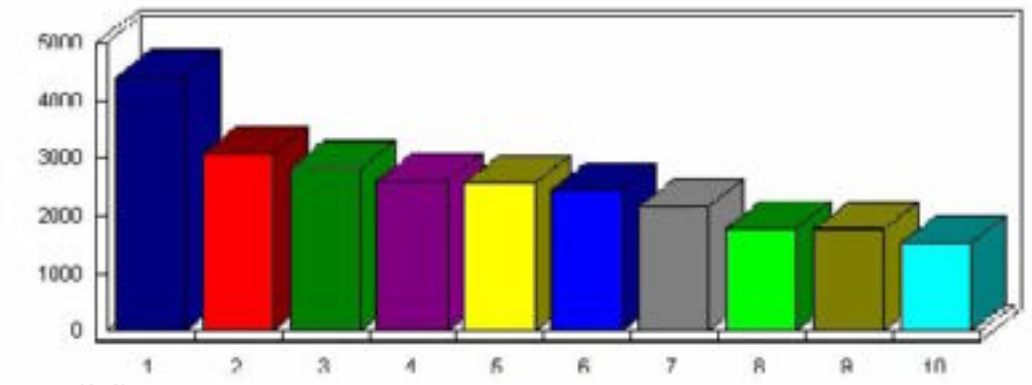


Figure 1.3

Figure 1.3: In the above graph, the Y-axis represents the volume of events (not including probes), while the X-axis represents the responsible IP (See Figure 1.4)

	Attacking IP Address	# Attacks
1	10.1.1.53.122	4302
2	10.1.1.51.108	3083
3	10.1.1.53.197	2814
4	216.106.166.211 1216-106-166-211.ibeam.com	2605
5	10.1.1.53.111	2573
6	10.1.1.53.174	2455
7	10.1.1.53.171	2173
8	10.1.1.53.123	1782
9	211.233.45.41	1770
10	10.1.1.53.116	1641

Figure 1.4

3.5 – Detects

The following list of detects is prioritized on number of occurrences. This list does not take probes into account and is meant to show only those events that may be attacks, intrusion attempts, or signs of already compromised hosts.

	Attacks	# Attacks
1	Microsoft IIS 4.0 / 5.0 Extended UNICODE Directory Traversal Attack	10582
2	Large UDP Packet Traffic Attack	15291
3	Generic FTP Wildcard Attack	660
4	Generic HTTP "cmd.exe" Request Attack	161
5	Generic X86 Buffer Overflow (TCP NOPS) Attack	146
6	Microsoft FrontPage Server Extensions DoS Attack	38
7	Nipd Remote Buffer Overflow Attack	32
8	Generic X86 Buffer Overflow (setreuid(0)) Attack	30
9	Allaire ColdFusion Administrator Password DoS Attack	29
10	Generic X86 Buffer Overflow (setegid(0)) Attack	15

Figure 1.5

3.5.1 - Microsoft IIS 4.0/5.0 Extended UNICODE Directory Transversal Attack

This detect indicates attempts to browse the file system of a host running either IIS 4.0 or 5.0 through a double dot "../" Directory transversal exploit. The attack works by using the Unicode representation of "/" or "\" in a requested URL. If the system has not filtered these out, the attack may be able to display arbitrary web-readable files.

3.5.2 - Large UDP Packet Traffic Attack

This detect is triggered by UDP packets that are larger than 8000 bytes. Large UDP packets are common on some networks but packets of this size are unusual can are a possible area of concern since UDP can be used as a covert communications channel with Trojans, as control traffic for DoS attacks.

3.5.3 – Generic FTP WildCard Attack

Many ftp servers are vulnerable to resource saturation attacks. In these attacks the user requests a long directory name that includes wildcard characters. When the server attempts to process the request the wildcard characters produce extremely long strings. The server can't allocate sufficient memory for the process and becomes inoperable.

3.5.4 – Generic HTTP "cmd.exe" Request Attack

This detect is triggered due to a HTTP request which included an attempt to execute the Windows NT/2000 command interpreter, cmd.exe. Attackers and Internet worms to trying to execute commands on the web server often use to this type of attack. If

successful the attacker would be able to execute arbitrary commands or gain access to the file system.

A great many of these were accompanied by UNICODE attacks which may indicate a Trojan or a scripted attack.

3.5.5 – Generic X86 Buffer Overflow (TCP NOPS) Attack

This detect indicates that a string of the character 0x90 (NOP) was detected. Many remote buffer overflow exploits pad their packets with NOP (no-operation) bytes. However, this does not necessarily mean that the traffic is malicious. The sequence can often occur in the transmission of binary files.

3.5.6 – Microsoft FrontPage Server Extensions DoS Attack

This signature is related to a failure of Microsoft IIS FrontPage Server Extensions to handle exception conditions. Microsoft FrontPage Server Extensions allow administrators to remotely manage web content. By supplying malformed data to FPSE, IIS could be subject to a DoS. FPSE do not need to be used in order for the DoS to be successful, it just has to be installed.

3.5.7 – NTPD Remote Buffer Overflow Attack

This signature identifies possible attack against a NTP (Network Time Protocol) daemon. NTP uses UDP to synchronize time between Internet timeservers and internal servers and network devices. Many versions of ntpd are vulnerable to buffer overflow exploits. If an attacker is successful in the exploitation of these vulnerabilities s/he may be able to crash the daemon and run arbitrary code on the host.

3.5.4 – Generic X86 Buffer Overflow (Setuid (0)) Attack

This signature is prone to false alarms since it is very short and does not contain any context information. As a result, it can be set off in cases where binary data is transferred from outside the network. That would seem to be the case here.

I analyzed the traffic and found that in every case the source port was TCP/6699, which is used by Napster, or the destination port was TCP/1214, which is associated with Kazaa. Both are popular peer-to-peer file sharing programs.

3.5.4 – Allarie Cold Fusion Administrator Password DoS Attack

This signature relates to a possible DoS attack against Allaire Colfusion 4.5.1. It seems that the software has a faulty password parsing mechanism for authentication requests. Because of this, it is possible to create a denial of service attack against ColdFusion 4.5.1 by inputting a string of over 40,000 characters into the password field on the

Administrator login page. While the server tries to parse the password, the CPU becomes unresponsive, requiring the application to be restarted.

3.5.4 – Generic X86 Buffer Overflow (Setgid (0)) Attack

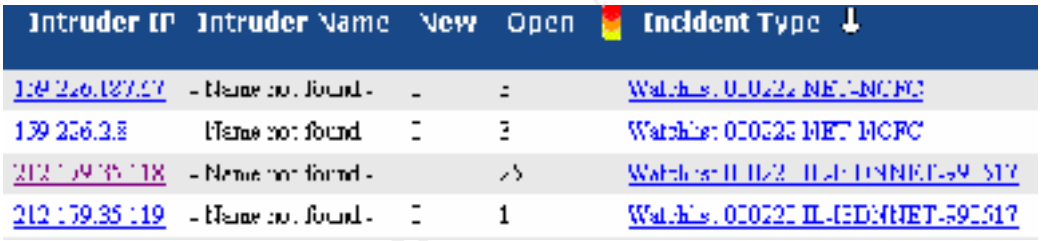
This alert indicates that they may have been an exploit attempt using the x86 platform setgid(0) system call. However, like the setuid(0) signature, this is prone to false alarms. Upon investigation in to the traffic I found that this was also most likely Napster, and Kazaa traffic.

3.6 - Selected External Hosts

The following IP addresses were selected based on the volume of alerts and criticality of the alert.

A. Watchlist 000220 IL-ISDNNET-990517

This host was selected because it triggered the Watchlist alert. These Watchlist alerts are triggered by traffic from specific host in Israel and China.



IP Address: 212.179.35.118
HostName: 212.179.35.118
DShield Profile: Country:
IL

Contact E-mail:
abuse@bezeqint.net

Total Records against IP:

Whois:% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html

inetnum: 212.179.0.0 - 212.179.255.255
netname: IL-ISDNNET-990517

```
descr:      PROVIDER
country:    IL
admin-c:    NP469-RIPE
tech-c:     TP1233-RIPE
tech-c:     ZV140-RIPE
tech-c:     ES4966-RIPE
status:     ALLOCATED PA
mnt-by:     RIPE-NCC-HM-MNT
changed:    hostmaster@ripe.net 19990517
changed:    hostmaster@ripe.net 20000406
changed:    hostmaster@ripe.net 20010402
source:     RIPE

route:      212.179.0.0/17
descr:      ISDN Net Ltd.
origin:     AS8551
notify:     hostmaster@isdn.net.il
mnt-by:     AS8551-MNT
changed:    hostmaster@isdn.net.il 19990610
source:     RIPE

person:     Nati Pinko
address:    Bezeq International
address:    40 Hashacham St.
address:    Petach Tikvah  Israel
phone:      +972 3 9257761
e-mail:     hostmaster@isdn.net.il
nic-hdl:    NP469-RIPE
changed:    registrar@ns.il 19990902
source:     RIPE

person:     Tomer Peer
address:    Bezeq International
address:    40 Hashakham St.
address:    Petakh Tiqwah  Israel
phone:      +972 3 9257761
e-mail:     hostmaster@isdn.net.il
nic-hdl:    TP1233-RIPE
changed:    registrar@ns.il 19991113
source:     RIPE

person:     Zehavit Vigder
address:    bezeq-international
address:    40 hashacham
address:    petach tikva 49170 Israel
phone:      +972 52 770145
```

fax-no: +972 9 8940763
e-mail: hostmaster@bezeqint.net
nic-hdl: ZV140-RIPE
changed: zehavitv@bezeqint.net 20000528
source: RIPE

person: Eran Shchori
address: BEZEQ INTERNATIONAL
address: 40 Hashacham Street
address: Petach-Tikva 49170 Israel
phone: +972 3 9257710
fax-no: +972 3 9257726
e-mail: hostmaster@bezeqint.net
nic-hdl: ES4966-RIPE
changed: registrar@ns.il 20000309
source: RIPE

B. External Top Talkers

This host was selected because they were the only external hosts in the top talkers with probes list. As you can see from the Dshield.org IP info, you are not the only site being probed by this host.

IP Address: 205.188.228.33
HostName: mslb3.spinner.com
DShield Profile: Country:
US

Contact E-mail:
tosgeneral@aol.com

Total Records against IP:
14474

Number of targets:
132

Date Range:
2002-01-29 to 2002-02-27

Ports Attacked (up to 10):
Port 6970
Attacks 1

Whois: America Online, Inc (NETBLK-AOL-DTC)
22080 Pacific Blvd
Sterling, VA 20166
US

Netname: AOL-DTC
Netblock: 205.188.0.0 - 205.188.255.255

Coordinator:
America Online, Inc. (AOL-NOC-ARIN) domains@AOL.NET
703-265-4670

Domain System inverse mapping provided by:

DNS-01.NS.AOL.COM	152.163.159.232
DNS-02.NS.AOL.COM	205.188.157.232

Record last updated on 27-Apr-1998.
Database last updated on 2-Mar-2002 19:57:03 EDT.

The ARIN Registration Services Host contains ONLY Internet Network Information: Networks, ASN's, and related POC's. Please use the whois server at rs.internic.net for DOMAIN related Information and whois.nic.mil for NIPRNET Information.

C. FTP WildCard Attacks.

This host was selected because it was the external address with greatest number of FTP attacks.

IP Address: 205.132.40.185
HostName: alpha185.emu.EDU
DShield Profile: Country:
US

Contact E-mail:
marpled@EMU.EDU

Whois:
```
Eastern Mennonite University (NETBLK-EMU-EDU)
   1200 Park Road
   Harrisonburg, VA  22801
   US

   Netname: EMU-EDU
   Netblock: 205.132.40.0 - 205.132.44.255

   Coordinator:
      Marple, Daniel E  (DEM3-ARIN)  marpled@EMU.EDU
      (540)432-4479 (FAX) (540)432-4444

   Domain System inverse mapping provided by:

   INSERV1.EMU.EDU             199.111.20.10
   SIRSISUN.EMU.EDU            205.132.42.69

   Record last updated on 05-Dec-1997.
   Database last updated on  2-Mar-2002 19:57:03 EDT.

The ARIN Registration Services Host contains ONLY Internet
Network Information: Networks, ASN's, and related POC's.
Please use the whois server at rs.internic.net for DOMAIN
related
Information and whois.nic.mil for NIPRNET Information
```

D. Microsoft IIS 4.0/5.0 Extended UNICODE Directory Transversal

This host is the external address with the highest number of this type of attack.

Intruder IP	Intruder Name	New	Open	Incident Type
80.56.144.47	f144047.upc-f.chello.nl	0	84	Microsoft IIS 4.0 / 5.0 Extended UNICODE Directory Traversal Attack

IP Address: 80.56.144.47
HostName: f144047.upc-f.chello.nl
DShield Profile: Country:
NL

Contact E-mail:
abuse@chello.com (bounced) (bounced)

Total Records against IP:
159

Number of targets:
47

Date Range:

2002-02-08 to 2002-02-20

Whois:

```
% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-
services/db/copyright.html

inetnum:      80.56.0.0 - 80.57.255.255
netname:      NL-CHELLO-20000509
descr:        Provider Local Registry
country:      NL
admin-c:      GM3659-RIPE
tech-c:       HTK1-RIPE
status:       ALLOCATED PA
mnt-by:       RIPE-NCC-HM-MNT
mnt-lower:    CHELLO-MNT
mnt-routes:   CHELLO-MNT
changed:      hostmaster@ripe.net 20000509
changed:      hostmaster@ripe.net 20010516
changed:      lir-help@ripe.net 20011211
source:       RIPE

route:        80.56.0.0/16
descr:        UPC.nl Network Services
descr:        Chello Customers
descr:        The Netherlands
origin:       AS8209
notify:       registry@upc.nl
notify:       hostmaster@chello.at
mnt-by:       CHELLO-MNT
changed:      hostmaster@chello.at 20010622
source:       RIPE

role:         Hostmaster Telekabel Wien
address:      chello Broadband GmbH
address:      Internet Services
address:      Reumannplatz 7
address:      A-1100 Vienna
address:      Austria
phone:        +43 1 96062
fax-no:       +43 1 96062 5666
e-mail:       hostmaster@chello.at
trouble:      help@chello.at
admin-c:      AK991-RIPE
tech-c:       SB9000-RIPE
tech-c:       MH392-RIPE
tech-c:       AK991-RIPE
nic-hdl:      HTK1-RIPE
notify:       hostmaster@chello.at
notify:       hm-dbm-msgs@ripe.net
mnt-by:       CHELLO-MNT
changed:      hostmaster@chello.at 20010906
```

```
source:       RIPE

person:       Gary Mendel
address:      Koningin Wilhelminaplein 2-4
address:      1062 HK Amsterdam
address:      The Netherlands
phone:        +31 20 355-2777
fax-no:       +31 20 355-2755
e-mail:       hostmaster@chello.com
nic-hdl:      GM3659-RIPE
notify:       hostmaster@chello.at
mnt-by:       CHELLO-MNT
changed:      hostmaster@chello.at 20010212
source:       RIPE
```

E. Compaq Management Agent Web File Access

The Compaq Insight Management Agent provides access to system information via a web browser. There are vulnerabilities that could allow a user to read files on the local system. Unless the AMI group is providing remote administration of university hosts, there is no reason for this access.

IP Address: 216.218.248.3
HostName: 216.218.248.3
DShield Profile: Country: US

Contact E-mail: hostmaster@he.net

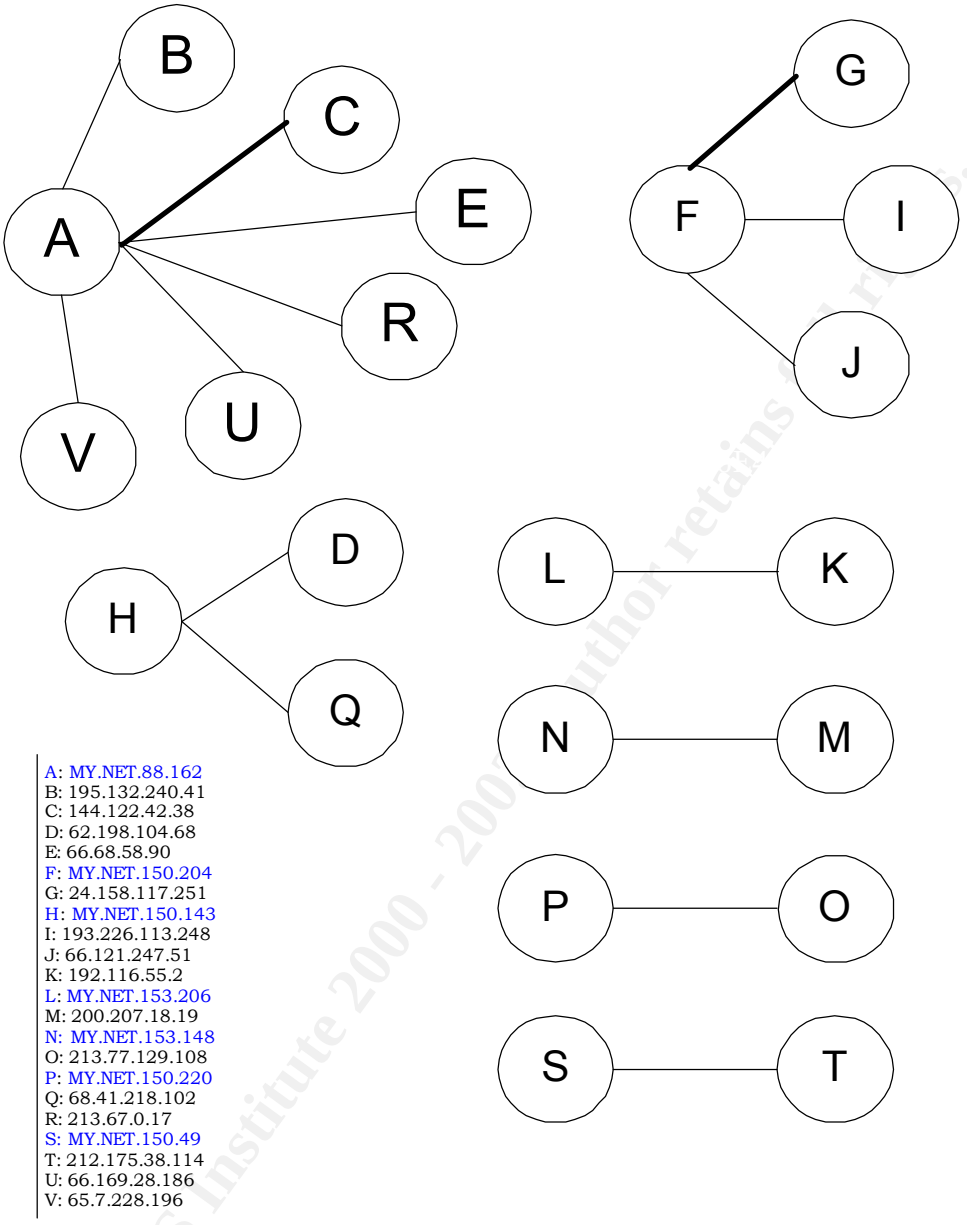
Whois:

```
Hurricane Electric (NETBLK-HURRICANE-1)
   760 Mission Court
   Fremont, CA 94539
   US

   Netname: HURRICANE-1
   Netblock: 216.218.128.0 - 216.218.255.255
   Maintainer: HURC

   Coordinator:
      Hurricane Electric  (ZH17-ARIN)  hostmaster@he.net
      510 580 4100

   Domain System inverse mapping provided by:

   NS1.HE.NET                  216.218.130.2
```

```
   NS2.HE.NET                  216.218.131.2
   NS3.HE.NET                  216.218.132.2

   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

   Record last updated on 23-Feb-2000.
   Database last updated on  2-Mar-2002 19:57:03 EDT.

----------

AMI Group (AMI), The (NETBLK-HURRICANE-CE0576-211)
   50 Vashell Way Ste. 200
   Orinda, CA 94563
   US

   Netname: HURRICANE-CE0576-211
   Netblock: 216.218.248.0 - 216.218.248.31

   Coordinator:
      Dyer, Chad  (CD769-ARIN)  dyer@aimnews.com
      925-254-4456 x110

   Record last updated on 22-Feb-2002.
   Database last updated on  2-Mar-2002 19:57:03 EDT.

The ARIN Registration Services Host contains ONLY Internet
Network Information: Networks, ASN's, and related POC's.
Please use the whois server at rs.internic.net for DOMAIN related
iInformation and whois.nic.mil for NIPRNET Information.
```

3.7 - Link Graph of OOS

To help with the analysis of the OOS packets, I have created link graphs to illustrate the host communications. These graphs show the links between nodes with weight given to the nodes that had the most number of connections. There were 22 hosts involved in all of the OOS files downloaded. After the analysis we can clearly we can see that there are distinct groupings of communicating nodes



In the first group, Node A is the obviously the focal point with a great deal of communication occurring between Node A and Node C. Although we do not see active traffic from Node A to Node C in the OOS files we can perform some basic traffic analysis and make some educated guesses.

When this traffic is analyzed with attention paid to the source and destination ports, the vast majority of the traffic occurs between ephemeral ports on Node C to TCP/1214 on Node A. The Kazaa file-sharing program utilizes this port. We also see traffic directly after this from 1214 on Node C to ephemeral ports on Node A. With this we can assume that there is a two-way communication occurring. The other Nodes communicating with Node A do not have the same two-way communication. Instead, Node A appears to acting as a Kazaa server for the rest of the Nodes.

In the second grouping of Nodes we see similar activity, however in this case Node F is acting only as a Kazaa server and is not initiating connections to external Nodes.

The third grouping is a little worrisome. The traffic from Node D to Node H originates from source port 10103 to a destination port 1883. I'm not sure what this traffic is. The destination port for the traffic between Node Q to Node H is TCP/9876. This is a known port for the Cyber Attacker Trojan. Because of this the traffic from Node D and Node H becomes more suspicious and suggests that Node H may have been compromised.

The other four link graphs depict one-to-one relationships. The traffic between Node K and Node L is all destined for TCP/6699. Napster is known to run on this port and is the most likely explanation for the traffic. Node P and Node S are both running Kazaa to which Node O and Node T are connecting.

I'm not sure what to make of the communication between Node M and Node N. The source port for Node M is TCP/916 and the destination is TCP/34450. Based on the ports alone, this could simply be a reply from M to N, however in the content of the packet there is the following string "GET /2297/Luda". This may turn out to be nothing but I believe that further investigation may be warranted.

3.8 - Summary & Defensive Recommendations

University networks have a history of being open and accessible. This is done in an effort to foster and encourage intellectual creativity. Unfortunately, it is that very openness that is creating many of the security issues that have been plaguing your environment. However, it is possible to maintain that accessibility while securing the network from compromise and vulnerability exploitation.

There is a lot of traffic that should be considered inappropriate occurring in your network. Probing, peer-to-peer file sharing, remote connection attempts to internal systems, possible Trojan activity, etc. Most of this is from an internal host. One of the first steps that you need to take toward securing the university network is to develop and circulate a Terms of Use or Acceptable Use Policy. You need to set some limits on student activities. Clearly define what is acceptable for students to do on the network, what is not, and the consequences for violating those terms. This should go a long way in preventing some of the more intentional intrusion attempts as well as help you identify unusual traffic that needs to be investigated further.

Second, there are good indications that many internal systems may have been compromised. These systems need to be isolated and analyzed. Where systems show signs of installed Trojans or compromise, they need to be cleaned. The operating systems and applications need to be patched and any unneeded or unused services should be disabled.

Third, your firewall and/or perimeter router seems to be very lenient around what it allows into the network. Firewall and perimeter routers should only allow access to necessary services from

the Internet. The means adopting a methodology of deny all except that which is explicitly allowed. Traffic like ICMP and SNMP should be automatically dropped. The logs from the IDS, Firewall, router and servers need to be checked at least every day for signs of unusual activity.

In order to achieve defense in-depth, the router and firewall should employ different mechanisms for traffic management. For example, if the router is performing basic packet filtering, the firewall could be an application firewall performing stateful inspection. This will give you the most flexibility for configuring rulesets and prevent attackers from slipping past similar perimeter devices.

There is quite a lot of work ahead to bring the traffic under control. I believe that all of these measures will go a long way towards securing your network without destroying the open and accessible environment that you currently enjoy.

References

Nothcutt, Stephen et al. Intrusion Detection Signatures and Analysis. Indianapolis, IN New Riders, January 2001

Scambray, Joel, et al Hacking Exposed: Network Security Secrets and Solutions 2nd Edition. Berkely, CA: Osborne/Mcgraw-Hill, 2001

http://www.securityfocus.com
http://aris.securityfocus.com/
http://www.dshield.org/ipinfo.php
http://www.sans.org
http://www.incidents.org
http://www.cert.org
http://www.practicallynetworked.com/sharing/app_port_list.htm
http://www.iana.org/assignments/port-numbers
http://www.iss.net/security_center/alerts/
www.simovits.com/sve/nyhetsarkiv/1999/nyheter9902.html