GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Network Monitoring and Threat Detection In-Depth (Security 503)"
at http://www.giac.org/registration/gcia

GCIA Version 3.3.

Sean Heare

June 28, 2003

© SANS Institute 2004, Author retains full rights.

Table of Contents

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Assignment 1 – The State Of Intrusion Detection

Are IDS Doomed?

1.1 Introduction

Richard Stennion of the Gartner Group believes that Intrusion Detection Systems will be relegated to wireless LANs in the next two years. He says the following regarding IDS:

> The underlying problem with IDS is that enterprises are investing in technology to detect intrusions on a network. This implies they are doing something wrong and letting those attacks in. Enterprises investing money to alert them when the next SQL Slammer worm arrives is a waste of money. [1]

This paper examines whether IDS is truly doomed to obscurity. It examines the threats to its niche in the security world and some possibilities for the future.

1.2 Why Are IDS Doomed?

There are two different flavors of Intrusion Detection Systems: Network, and Host, each structured to accomplish the same goal in a different manner. Network Intrusion Detection Systems (NIDS) such as Snort, RealSecure Network Sensor, and Intrusion's SecureNet promiscuously collect packets that pass through their point on the network. Host Intrusion Detection Systems such as Tripwire, Symantec's Host IDS, and Intrusion's SecureHost, detect intrusion solely on the hosts where they reside through either file integrity and job process checks, the monitoring of packets addressed to the host, or a combination of both.

There are two models for determining when an intrusion is being detected: Anomalous and Event (also called Rule-Based or Misuse) Driven. Detection models based upon Event Driven detection alert an administrator only when traffic that conforms to a certain rule or signature designed to capture a specific type of hacking behavior is detected. The drawback to this is that the software can only defend the network against intrusions that conform to known signatures. This means that it is unlikely that an event-driven IDS will ever detect a zero-day exploit.

Detection models based upon detecting anomalous behavior of any kind are much better at proactively detecting zero-day exploits. However, they also are

[1] Mimoso, Michael. "Gartner Declares IDS Obsolete by 2005." http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci905961,00.html. June 12, 2003. (June 19, 2003)

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

expert systems and as such, must first learn what is normal behavior for a network. This provides an opportunity for a hacker who is aware of an Anomaly-based IDS that was recently installed. She could train the IDS to regard her malicious behavior as normal behavior.

Anomaly-driven IDS also have the drawback that they could block unusual, though legitimate traffic. Users working at odd times, i.e. late night project deadlines, or performing odd tasks could errantly be logged as a false positive.

A drawback common to all IDS is that they detect an intrusion attempt only after it is already underway. The need to engage hackers attempting these intrusions proactively and automatically has lead to the evolution of Intrusion Prevention Systems.

Network Intrusion Prevention Systems prevent access to protected systems by detecting attempts at intrusion as does IDS. However IPS is more proactive in that it resets malicious connections, thus preventing the intrusion from taking place.

Most IPS are inline, meaning that they sit directly on the wire examining traffic. This allows them to prevent access to the unauthorized system in real-time.

1.3 What Will Really Happen?

Because of the reasons above, many disciples of IPS, not just Stennion, have declared it to be the latest panacea. Closing off the network to hackers as they attack IPS will enable keeping the network secure alone, or at least without IDS.

It should be noted that IPS have drawbacks as well. IPS could potentially cut off legitimate traffic, much in the same way that anomaly driven IDS erroneously detect some legitimate traffic as false positives. The difference being, of course, that the IPS has cut off the traffic.

In order to prevent malicious traffic real-time, an IPS must straddle a network to analyze each packet passing into the network. The inline nature of an IPS means that it will introduce latency into the network and become a single point of network failure.

Beyond the drawbacks IPS possesses, Martin Roesch points out one defect that insures the longevity of IDS. Although an IPS is very proactive and very good at blocking out known malicious traffic, how will IPS know if a zero-day exploit gets through?[2] The only way to possess reasonable confidence that this has not happened, is to audit host IDS for changes and to audit Network IDS for odd network behavior.

[2] Roesch, Martin "IDS dead? ... Not So Fast!" http://napps.nwfusion.com/weblogs/security/002959.html#002959. June 16, 2003. (June 19, 2003).

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

IDS and IPS will each change their roles over the next few years. They will also adapt to cope with their present drawbacks. IDS Vendors will focus on Network Health/Management issues and Host Intrusion Detection. IDS Vendors have already begun working on IMS consoles that will manage distributed IDS from one point.

IPS Vendors are addressing the IPS single point of failure issue by introducing clustering and fault-tolerant systems. Clustering places multiple inline IDS on parallel lines to monitor the traffic passing through them. This way if one IDS fails then the other can take over duties across the full line.

Both IDS and IPS, as standalone units, do not defend an enterprise as well as these systems would do together, using a defense-in-depth strategy. However, the balkanized nature of handling multiple IDS/IPS overwhelms the most efficient administrators. Auditing intrusion logs is daunting, and then processing them and fine-tuning rules, or teaching an expert system about new networks, lead to the need for centralized analysis and management. IMS, with its holistic framework, forms that centralized defense-in-depth framework for incident detection, prevention and handling.

1.4 Why IPS And IDS Should Be "Doomed".

In his book, Inside The Tornado, Geoffrey Moore states that:

> [Paradigm] shifts begin with the appearance of a new of a new category of product that incorporates breakthrough technology enabling unprecedented benefits. It is immediately proposed as the natural replacement for a whole class of infrastructure, winning early converts and enthusiastic predictions of a new world order.[3]

According to Paul Proctor of NFR Security, Intrusion Management Systems "…are comprised of a family of discrete security-related technologies that protect IT assets against attack through a continuous process of exposure assessment, intrusion detection, prevention, analysis and response."[4]

In other words, IMS must defend an enterprise before, during, and after an attack by utilizing and coordinating a variety of security systems that include IDS and IPS. NFR, Sourcefire, Symantec, and Cisco have already introduced the first generation of such holistic management systems to the marketplace. Early adopters are using these systems, but the IMS available now still need to

[3] Moore, Geoffrey. Inside the Tornado : Marketing Strategies from Silicon Valley's Cutting Edge. 1999, Harper Business, New York. p. 4.
[4] Proctor, Paul "Intrusion Detection and Homeland Security - Ask the Expert – CIO", http://www2.cio.com/ask%5Cexpert/2002/questions/question1522.html. July 15, 2002 (June 19, 2003).

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

address some issues before truly being capable of disrupting standalone IDS/IPS technology.

The two major benefits of IMS over IPS or IDS are reduced cost of management and centralized correlation of intrusion data and rule management. In order to maximize these advantages IMS needs the following:

- ✓ An IMS must be capable of managing devices built by disparate manufacturers. Many already do, in fact, support other manufacturers closed standards.

  However, several methods of exchange of open Intrusion data have been proposed[5]. None of the Intrusion Management Systems on the market currently support an open standard of exchanging Intrusion Detection data. This weakness does not allow for maximization of central management of disparate security systems.

- ✓ All traffic between the management console and the sensors should be encrypted[6]. Most systems do this, but Cisco still offers Management systems that can maintain sensors and transfer files using FTP and Telnet[7]

- ✓ An IMS must be simple to install, configure and maintain. NFS and Sourcefire are already going this way with their new offerings, however, Symantec and Cisco Security Management are still daunting to install, configure and maintain.

1.5 Conclusion

It is apparent that some kind of centralized management would not only greatly aid those in charge of security for an enterprise, but would also reduce the cost of security to the enterprise. As standalone systems, both IPS and IDS should be made into parts of a Security Management suite. Instead of eliminating one or both, their union would be larger, yet easier to manage.

© SANS Institute 2004, Author retains full rights.

[5] IETF. "Intrusion Detection Exchange Format (idwg) Charter." http://www.ietf.org/html.charters/idwg-charter.html. April 10, 2003 (June 19, 2003).

[6] Bandy, P., Money, M. & Worstell, K. "Should communication between the sensor (or agent) and the monitor be encrypted?", http://www.sans.org/resources/idfaq/communication.php. 1999. (June 19, 2003)

[7] Cisco. "Cisco Secure Intrusion Detection System Sensor Configuration Note Version 3.0." http://www.cisco.com/univercd/cc/td/doc/product/iaabu/csids/csids6/12216_02.htm#xtocid16. Oct. 17, 2002. (June 19, 2003).

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

1.6 References

1. Mimoso, Michael. “Gartner Declares IDS Obsolete by 2005.” http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci905961,00.html. June 12, 2003. (June 19, 2003)

2. [1] Roesch, Martin “IDS dead? ... Not So Fast!” http://napps.nwfusion.com/weblogs/security/002959.html#002959. June 16, 2003. (June 19, 2003).

3. Moore, Geoffrey. Inside the Tornado : Marketing Strategies from Silicon Valley's Cutting Edge. 1999, Harper Business, New York. p. 4.

4. Proctor, Paul “Intrusion Detection and Homeland Security - Ask the Expert – CIO”, http://www2.cio.com/ask%5Cexpert/2002/questions/question1522.html. July 15, 2002 (June 19, 2003).

5. IETF. “Intrusion Detection Exchange Format (idwg) Charter.“ http://www.ietf.org/html.charters/idwg-charter.html. April 10, 2003 (June 19, 2003).

6. Bandy, P., Money, M. & Worstell, K. “Should communication between the sensor (or agent) and the monitor be encrypted?”, http://www.sans.org/resources/idfaq/communication.php. 1999. (June 19, 2003)

7. Cisco Systems. “Cisco Secure Intrusion Detection System Sensor Configuration Note Version 3.0.” http://www.cisco.com/univercd/cc/td/doc/product/iaabu/csids/csids6/12216_02.htm#xtocid16. Oct. 17, 2002. (June 19, 2003).

© SANS Institute 2004, Author retains full rights.

Assignment 2 – Network Detects

2.1 Attack One - RingZero

2.1.1 Source of Trace

The 2002.9.25 data file used in this analysis was downloaded from http://www.incidents.org/logs/Raw/2002.9.25. The destination IP addresses within the 32.245.0.0/16 network are problematic. The range of the packets detected ranged from 32.245.1.160 to 32.245.253.12. So, either the external router is improperly configured and is forwarding all 32.245.0.0/16 traffic to 32.245.166.0/24 or, more likely, 32.245.166.0/24 is the DMZ for the 32.245.0.0/16 domain. Since the Internal Network range is so large, this detect probably came from routers at an ISP or large corporation.

The sensor that captured these logs has been placed between an Internet-facing router and a router internal to the 32.245.0.0/16 (or 32.245.166.0/24) network. According to Ethereal, both routers possess MAC addresses conforming to ranges assigned to Cisco Systems.

A cursory search of Google for the path and filenames listed in the http headers collected at the beginning of the trace make it likely that the one of the customers in this trace can be identified. If this is the case then the filenames obtained from the website should be sanitized as well.

2.1.2 Detect was generated by

These alerts were generated using Snort Version 2.0.0 (Build 72) running on a Redhat Linux 8.0 (psyche) workstation. The rulesets used were downloaded on June 12th, 2003 from the snortrules-stable.tar.gz file found on the Snort website. The raw data was pulled through snort using the following command:

```
[sean@snorttest tmp]# snort -c /tmp/snort-2.0.0/etc/snort.conf -r
2002.9.25 -l log.2002.9.25
```

The -c switch tells Snort where to find the configuration file. The -r switch designates the file from which Snort should read data. The -l switch designates the directory to which alerts should be logged.

When this command was run the following alerts were two of many issued:

```
 [**] [1:618:4] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/24-22:45:28.206507 24.165.255.20:3271 -> 32.245.196.143:3128
TCP TTL:109 TOS:0x0 ID:59483 IpLen:20 DgmLen:48 DF
******S* Seq: 0xADAD66E3  Ack: 0x0  Win: 0xFAF0  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:620:3] SCAN Proxy (8080) attempt [**]
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
[Classification: Attempted Information Leak] [Priority: 2]
10/24-22:45:28.216507 24.165.255.20:3272 -> 32.245.196.143:8080
TCP TTL:109 TOS:0x0 ID:59484 IpLen:20 DgmLen:48 DF
******S* Seq: 0xADAE6105  Ack: 0x0  Win: 0xFAF0  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
...
```

The portion of the tcpdump that corresponds to this detect:

```
22:45:28.206507 20.255.165.24.cfl.rr.com.3271 >
32.245.196.143.squid: S 2913822435:2913822435(0) win 64240 <mss
1460,nop,nop,sackOK> (DF)
22:45:28.216507 20.255.165.24.cfl.rr.com.3272 >
32.245.196.143.webcache: S 2913886469:2913886469(0) win 64240
<mss 1460,nop,nop,sackOK> (DF)
22:45:31.176507 20.255.165.24.cfl.rr.com.3271 >
32.245.196.143.squid: S 2913822435:2913822435(0) win 64240 <mss
1460,nop,nop,sackOK> (DF)
22:45:31.176507 20.255.165.24.cfl.rr.com.3272 >
32.245.196.143.webcache: S 2913886469:2913886469(0) win 64240
<mss 1460,nop,nop,sackOK> (DF)
22:45:37.186507 20.255.165.24.cfl.rr.com.3272 >
32.245.196.143.webcache: S 2913886469:2913886469(0) win 64240
<mss 1460,nop,nop,sackOK> (DF)
22:45:37.186507 20.255.165.24.cfl.rr.com.3271 >
32.245.196.143.squid: S 2913822435:2913822435(0) win 64240 <mss
1460,nop,nop,sackOK> (DF)
...
```

Since the message portion of the rule that was set off has the keyword SCAN in it I looked through the scan.rules file and came across these two rules.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 3128 (msg:"SCAN Squid
Proxy attempt"; flags:S,12; classtype:attempted-recon; sid:618;
rev:4;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 8080 (msg:"SCAN Proxy
\(8080\) attempt"; flags:S,12; classtype:attempted-recon;
sid:620; rev:3;)
```

The sids are listed as 618 and 620. A search of the snort database for those sids yielded the following information regarding the rule documentation (http://www.snort.org/snort-db/sid.html?id=618)

This rule was triggered by an attempt from the Internet to connect (Syn Flag) via TCP to a host in the home network on port 3128. This is the default port for Squid, a Web Application Proxy Cache. Port 8080 is a default port for other kinds of Web Proxies.

© SANS Institute 2004, Author retains full rights.

2.1.3 Probability the source address was spoofed

Since this appears to be a reconnaissance scan spoofing is unlikely. In order to get information back from the scan a host has to be waiting for the SYN/ACK response to the SYN sent.

This almost certainly means that the host conducting the assault has been compromised from the outside. Since the attacking address is from the Roadrunner domain it is likely that this is a host attached to a cable modem.

The TTL for each of the packets in question is 109. This is within the expected range for a packet coming from a Windows9x/NT/ME/2000/XP host whose time to live decrements from 128. The attacker is most likely using a host running Microsoft Windows written after NT 3.51 (In Windows95 and later 128 became the default TTL).

2.1.4 Description of Attack

This is an attempt to find out if either Squid (TCP Port 3128) or another web proxy (TCP Port 8080) is running and open on the host in question. Web Proxies in particular are valuable for masking a perpetrator's identity when committing online fraud and attacks on other hosts. Also, searching for open services is one of the first steps in attempting to penetrate a network.

This stimulus most likely comes from a compromised Windows workstation connected to a broadband provider. The name resolved for the hacking IP address is within the cfl.rr.com (roadrunner) domain.

2.1.5 Attack Mechanism

The attack mechanism with the highest likelihood of matching this attack is a variant of the RingZero Trojan. RingZero can be configured to scan ports 8080 and 3128 alone and is running almost exclusively on compromised residential Windows workstations (All flavors except NT) attached to broadband.

Less likely, but still possible, are manual attempts at reconnaissance. The attack in question could be constructed running a variety of different applications that use TCP to connect to a daemon such as telnet, a web browser, or a custom script such as this one coded in Visual Basic (http://www.visualbasicforum.com/t50522.html).

2.1.6 Correlations

The Security Community identified RingZero in October of 1999.

An advisory from the National Infrastructure Protection Center describing the trojan and it's behavior:
http://www.nipc.gov/warnings/advisories/1999/99-024.htm

This SANS Advisory also describes a RingZero attack.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

http://www.sans.org/resources/idfaq/ring_zero.php

A profile regarding the Trojan itself found at f-secure.
http://www.f-secure.com/v-descs/ringzero.shtml

2.1.7 Evidence of Active Targeting

RingZero has no known active targeting. There is one other session meeting the same signatures in the capture from a different IP address going to a different IP address in the internal network. Neither of the Internal IP addresses are again targeted by any other attacks within the capture.

2.1.8 Severity

The severity of an attack is determined by rating the criticality and the lethality of an attack subtracted from countermeasures on the host and network, on a 1 to 5 scale, 1 being the lowest and 5 the highest.

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

The sole purpose of this attack is to reconnoiter hosts for open proxies. As such it is less critical than attacks that directly compromise the system but more critical than a simple port scan because there is a specific target (open proxies) being sought. Criticality is 2.

If this particular attack successfully received a response then it could lead to abuse of the open proxy. If Squid in particular responded then the hacker would be able to focus attacks on compromising Squid in particular. Lethality is 2.

Host and Network countermeasures are difficult to determine. Since there was no acknowledgement from the host in question it can be assumed that there is either no host running services on these ports at these IP addresses or that an ACL is in place in front of these hosts preventing successful reconnaissance by this attack.

> (2 + 2)(2 + 2) = 0

2.1.9 Defensive Recommendations

Defenses appear to be adequate from what is known of the network, as no information was released in response to the stimulus. However, if the internal network must host a proxy server that is accessible to the world, it should only be accessible outside the firewall via a Virtual Private Network. ACLs should be in place preventing direct connections to the proxy ports from the outside.

Preventing RingZero from infecting the internal network is not difficult since it is spread almost exclusively through e-mail. RingZero like other known malware can be circumvented by installing anti-virus software on hosts, scanning

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

incoming executables on the mail server for malware, and keeping anti-virus software updated.

If an administrator has spare time, (I know, I have never met one either.) then he could create a host that responded to RingZero stimulus (and other hostile stimulus) by sending e-mail to the Internet Service Provider's abuse mailbox with a log and description of the session. Alternately, the administrator could send the logs to DShield's Fightback. If the internal host's network card were assigned all the unassigned addresses for the domain, then it would pick up all the random port scans and process them. This would allow the administrator to focus on larger, more direct threats.

2.1.10 Multiple Choice Questions

```
22:45:28.206507 xx.yy.cfl.rr.com.3271 > aa.bb.196.143.squid: S
2913822435:2913822435(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)
22:45:28.216507 xx.yy.cfl.rr.com.3272 > aa.bb.196.143.webcache: S
2913886469:2913886469(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)
22:45:31.176507 xx.yy.cfl.rr.com.3271 > aa.bb.196.143.squid: S
2913822435:2913822435(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)
22:45:31.176507 xx.yy.cfl.rr.com.3272 > aa.bb.196.143.webcache: S
2913886469:2913886469(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)
22:45:37.186507 xx.yy.cfl.rr.com.3272 > aa.bb.196.143.webcache: S
2913886469:2913886469(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)
22:45:37.186507 xx.yy.cfl.rr.com.3271 > aa.bb.196.143.squid: S
2913822435:2913822435(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)
...
```

Which of the following is most likely shown in the trace above.

A. A legitimate user attempting to connect over VPN to proxy servers.
B. A trojaned host automatically scanning the Internet for open proxy servers.
C. A hacker running nmap to scan aa.bb.196.143
D. A legitimate user attempting to connect over the open Internet to proxy servers.

Answer: B

Explanation: A legitimate user would likely connect to one port or the other not both. The proxy packet would be encapsulated in an IPSec packet depending on where the capture was scanned. Hence from the information given the use of a VPN cannot be determined. Finally, a hacker running nmap would likely attempt to be stealthier and send out packets over a longer period of time.

2.1.11 Defense of analysis

The original posting of this analysis can be found at http://cert.uni-stuttgart.de/archive/intrusions/2003/06/msg00251.html.
Starplanet1000@yahoo.com.hk posted the following questions in regards to this analysis:

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

(1) 24.165.255.20 is IP address allocated to "Road Runner" which is a cable broadband company, A subsidiary of "TIME WARNER". Second, i do quick search from Google "Road Runner" + "scan". It happened that they scanned for open proxy/relay !!! . http://www.dshield.org/pipermail/list/2003-February/007139.php

Is this co-related to your finding ?

Or it is not a trojan that compromise your mentioned Win2K machine. If you said this is WIN2K machine, hv you verified with "p0f". The assumption of fingerprint OS just based on TTL might be too premature.

1st Answer: I concur. I should have been more specific than merely saying "The name resolved for the hacking IP address is within the cfl.rr.com (roadrunner) domain."

However, the second point has nothing to do with this detect. The scans examined in http://www.dshield.org/pipermail/list/2003-February/007139.php were in fact coming from Roadrunner's corporate offices.

Here is part of the original scan (http://www.dshield.org/pipermail/list/2003-February/007062.php) :

```
Feb 26, 2003 19:22:30.246 UTC  -  (TCP) 24.30.199.228 :
2049  >>>  151.202.16.167 : 8081...
```

A quick Dshield IP report of the attacking IP address comes up with:

IP Address: 24.30.199.228
HostName: securityscan.sec.rr.com

DShield Profile:		
	Country:	US
	Contact E-mail:	security@rr.com
	Total Records against IP:	0
	Number of targets:	0
	Date Range:	not reports

I agree that merely using the TTL for passive fingerprinting can lead to errors. In this case p0f comes up with the following:

```
24.165.255.20 [20 hops]: Windows XP Pro, Windows 2000 Pro
```

(2) 20.255.165.24.cfl.rr.com is not a IP Address allocated to "Road Runner". Instead, this address belongs to "Computer Science Corporation" (check with Dshield IP report).

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

2nd Answer: The hostname is actually the reverse-in-addr-arpa of the IP address. Hence 20.255.165.24.cfl.rr.com is in fact part of Road Runners residential service IP space. Here is the Dshield IP report on it.

IP Address: 24.165.255.20
HostName: 20.255.165.24.cfl.rr.com
DShield Profile:

Country:	US
Contact E-mail:	security@rr.com
Total Records against IP:	not processed
Number of targets:	select update below
Date Range:	to

(3) For defense recommendation, I don't [see] why we need to deploy VPN in this case. Normally, if you deploy "proxy" for internal user, there is no need to control it via VPN. If you are not allowed inbound proxy, why we don't impose a ingress filtering (ACL) on the border router/Firewall. If you allow inbound proxy(ie. reverse proxy), OS hardening and egress filtering is also important. In case of comprose, the attacker should NOT be allowed to use your proxy as step[pping]stone to attack[ing] another company. This is legally liable.

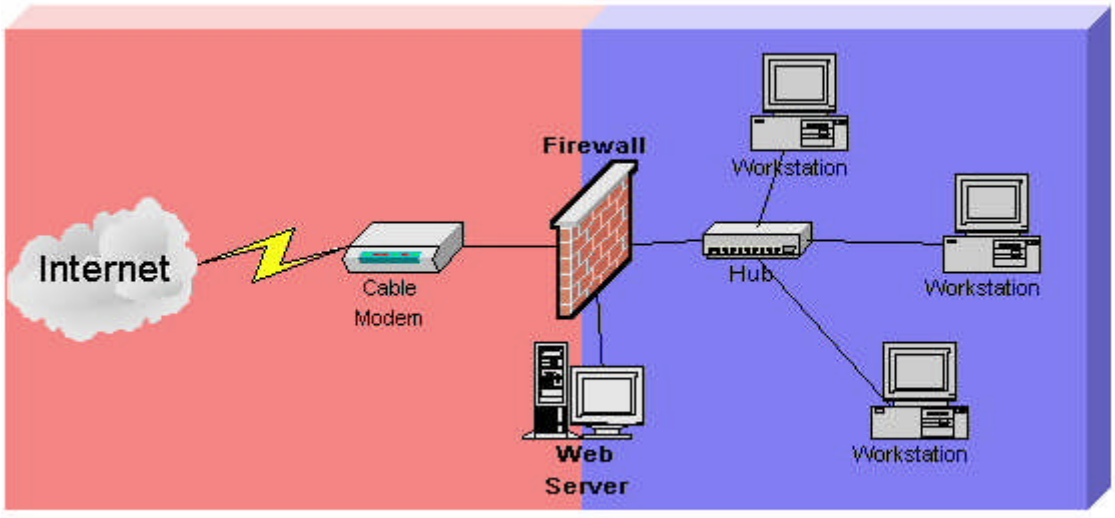
3rd Answer: I would only want to apply a VPN if in fact the organization in question was running a proxy service that telecommuters would need to access. There is no evidence of there being a proxy in this detect. However, I agree with you that if there were a proxy egress filtering and OS hardening should be included in the defense recommendations for the proxy.

2.2 Attack Two – Nimda attack on Residential System

2.2.1 Source of Trace

On June 14th, 2003 this trace was gathered from the web server on my home network:

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.



On my home network, a Cable Modem provides uplink to the service provider. Behind the cable modem is a Linux Firewall running IPTables. The Firewall has three NICs, one that connects to the outside world, one for the internal network and one to connect to the web server.

The internal network consists of a couple of Windows PCs, and a Linux workstation. The Web Server (Redhat Linux 8.0 running Apache 2.046) is isolated on it's own network, connected to the firewall's third NIC via a crossover cable.

2.2.2 Detect was generated by

These alerts were generated using Snort Version 2.0.0 (Build 72) running on the web server mentioned above. The rulesets used were downloaded on June 12th, 2003 from the snortrules-stable.tar.gz file found on the Snort website. The raw data was pulled through snort using the following command:

```
[sean@webserver tmp]# snort -c /tmp/snort-2.0.0/etc/snort.conf -l
log.2003.06.14
```

The -c switch tells Snort where to find the configuration file. The -l switch designates the directory to which alerts should be logged. Traffic was monitored simultaneously in the web server log. Between 18:34:54:780297 and 18:35:02.394756, 16 attempts to attack the web server were logged as coming from 68.33.141.19. These are the signatures matched followed by the coinciding entry in the web servers log file:

```
[**] [1:1256:7] WEB-IIS CodeRed v2 root.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/14-18:34:54.780297 68.33.141.19:2240 -> 192.168.1.4:80
TCP TTL:110 TOS:0x0 ID:16356 IpLen:20 DgmLen:112 DF
***AP*** Seq: 0x5B5BECB3  Ack: 0x36EF180B  Win: 0xFAF0  TcpLen: 20
[Xref => http://www.cert.org/advisories/CA-2001-19.html]
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
68.33.141.19 - - [14/Jun/2003:18:34:54 -0400] "GET
/scripts/root.exe?/c+dir HTTP/1.0" 404 12205

[**] [1:1256:7] WEB-IIS CodeRed v2 root.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/14-18:34:55.112554 68.33.141.19:2245 -> 192.168.1.4:80
TCP TTL:110 TOS:0x0 ID:16381 IpLen:20 DgmLen:110 DF
***AP*** Seq: 0x5B603F14  Ack: 0x3789C1C9  Win: 0xFAF0  TcpLen: 20
[Xref => http://www.cert.org/advisories/CA-2001-19.html]

68.33.141.19 - - [14/Jun/2003:18:34:55 -0400] "GET
/MSADC/root.exe?/c+dir HTTP/1.0" 404 12205

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/14-18:34:55.458802 68.33.141.19:2253 -> 192.168.1.4:80
TCP TTL:110 TOS:0x0 ID:16405 IpLen:20 DgmLen:120 DF
***AP*** Seq: 0x5B67D9AA  Ack: 0x36DB10B1  Win: 0xFAF0  TcpLen: 20

68.33.141.19 - - [14/Jun/2003:18:34:55 -0400] "GET
/c/winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/14-18:34:55.814369 68.33.141.19:2257 -> 192.168.1.4:80
TCP TTL:110 TOS:0x0 ID:16426 IpLen:20 DgmLen:120 DF
***AP*** Seq: 0x5B6C5F24  Ack: 0x374E654B  Win: 0xFAF0  TcpLen: 20

68.33.141.19 - - [14/Jun/2003:18:34:55 -0400] "GET
/d/winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205

[**] [1:1945:1] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]
06/14-18:34:56.158366 68.33.141.19:2270 -> 192.168.1.4:80
TCP TTL:110 TOS:0x0 ID:16451 IpLen:20 DgmLen:136 DF
***AP*** Seq: 0x5B76B67C  Ack: 0x37A6F998  Win: 0xFAF0  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884]

68.33.141.19 - - [14/Jun/2003:18:34:56 -0400] "GET
/scripts/..%255c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205

[**] [1:1288:5] WEB-FRONTPAGE /_vti_bin/ access [**]
[Classification: access to a potentially vulnerable web application]
[Priority: 2]
06/14-18:34:56.512403 68.33.141.19:2281 -> 192.168.1.4:80
TCP TTL:110 TOS:0x0 ID:16490 IpLen:20 DgmLen:157 DF
***AP*** Seq: 0x5B7FF5BE  Ack: 0x36E9AE7B  Win: 0xFAF0  TcpLen: 20

68.33.141.19 - - [14/Jun/2003:18:34:56 -0400] "GET
/_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
HTTP/1.0" 404 12205

[**] [1:1286:5] WEB-IIS _mem_bin access [**]
[Classification: access to a potentially vulnerable web application]
[Priority: 2]
06/14-18:34:56.857054 68.33.141.19:2298 -> 192.168.1.4:80
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
TCP TTL:110 TOS:0x0 ID:16536 IpLen:20 DgmLen:157 DF
***AP*** Seq: 0x5B8E155E  Ack: 0x374A6EC5  Win: 0xFAF0  TcpLen: 20

68.33.141.19 - - [14/Jun/2003:18:34:56 -0400] "GET
/_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
HTTP/1.0" 404 12205

[**] [1:982:6] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]
06/14-18:34:57.207869 68.33.141.19:2314 -> 192.168.1.4:80
TCP TTL:110 TOS:0x0 ID:16572 IpLen:20 DgmLen:185 DF
***AP*** Seq: 0x5B9B1F4E  Ack: 0x379DCA62  Win: 0xFAF0  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884]

68.33.141.19 - - [14/Jun/2003:18:34:57 -0400] "GET
/msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../win
nt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205

[**] [1:982:6] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]
06/14-18:34:57.545706 68.33.141.19:2333 -> 192.168.1.4:80
TCP TTL:110 TOS:0x0 ID:16609 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0x5BA9B491  Ack: 0x37BA39F7  Win: 0xFAF0  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884]

68.33.141.19 - - [14/Jun/2003:18:34:57 -0400] "GET
/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/14-18:34:57.879957 68.33.141.19:2340 -> 192.168.1.4:80
TCP TTL:110 TOS:0x0 ID:16649 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0x5BB0C611  Ack: 0x37BDF498  Win: 0xFAF0  TcpLen: 20

68.33.141.19 - - [14/Jun/2003:18:34:57 -0400] "GET
/scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205

[**] [1:981:6] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]
06/14-18:34:58.214125 68.33.141.19:2348 -> 192.168.1.4:80
TCP TTL:110 TOS:0x0 ID:16682 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0x5BB8BCC9  Ack: 0x37CDA509  Win: 0xFAF0  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884]

68.33.141.19 - - [14/Jun/2003:18:34:58 -0400] "GET
/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205

[**] [1:983:6] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]
06/14-18:34:58.561464 68.33.141.19:2356 -> 192.168.1.4:80
TCP TTL:110 TOS:0x0 ID:16713 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0x5BC0107D  Ack: 0x370D62FE  Win: 0xFAF0  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884]

68.33.141.19 - - [14/Jun/2003:18:34:58 -0400] "GET
/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
[**] [1:970:5] WEB-IIS multiple decode attempt [**]
[Classification: Web Application Attack] [Priority: 1]
06/14-18:34:58.913228 68.33.141.19:2363 -> 192.168.1.4:80
TCP TTL:110 TOS:0x0 ID:16735 IpLen:20 DgmLen:138 DF
***AP*** Seq: 0x5BC67AAF  Ack: 0x37B3B0A8  Win: 0xFAF0  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0333]

68.33.141.19 - - [14/Jun/2003:18:34:58 -0400] "GET
/scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 400 226

[**] [1:970:5] WEB-IIS multiple decode attempt [**]
[Classification: Web Application Attack] [Priority: 1]
06/14-18:35:01.990865 68.33.141.19:2368 -> 192.168.1.4:80
TCP TTL:110 TOS:0x0 ID:17054 IpLen:20 DgmLen:136 DF
***AP*** Seq: 0x5BCB2440  Ack: 0x37CE3E53  Win: 0xFAF0  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0333]

68.33.141.19 - - [14/Jun/2003:18:35:01 -0400] "GET
/scripts/..%%35c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 400 226

[**] [1:970:5] WEB-IIS multiple decode attempt [**]
[Classification: Web Application Attack] [Priority: 1]
06/14-18:35:02.062931 68.33.141.19:2483 -> 192.168.1.4:80
TCP TTL:110 TOS:0x0 ID:17070 IpLen:20 DgmLen:140 DF
***AP*** Seq: 0x5C2AAFF2  Ack: 0x3755BFDD  Win: 0xFAF0  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0333]

68.33.141.19 - - [14/Jun/2003:18:35:02 -0400] "GET
/scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404
12205

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
06/14-18:35:02.394756 68.33.141.19:2496 -> 192.168.1.4:80
TCP TTL:110 TOS:0x0 ID:17153 IpLen:20 DgmLen:136 DF
***AP*** Seq: 0x5C345F6A  Ack: 0x37B549D9  Win: 0xFAF0  TcpLen: 20

68.33.141.19 - - [14/Jun/2003:18:35:02 -0400] "GET
/scripts/..%252f../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205
```

These attacks occur several times a day suggesting that this is an automated attack scanning for random web hosts on the Internet by hosts compromised by a trojan.

2.2.3 Probability the source address was spoofed

The source address was not spoofed. This is almost definitely an automated attack from a compromised Windows host connected to broadband. The attacks occurred in rapid succession suggesting automation. The attack was almost certainly random since the host attacked is a web server on a residential network. Random attacks from a compromised host have no reason to conceal their true source address.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

2.2.4 Description of Attack

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0884 is the entry in the CVE database for the vulnerability that this attack is attempting to exploit.

```
18:34:54.747322 pcp03190206pcs.salsbr01.md.comcast.net.2240 >
192.168.1.4.http: S 1532751026:1532751026(0) win 64240 <mss
1460,nop,nop,sackOK> (DF)
18:34:54.747780 192.168.1.4.http >
pcp03190206pcs.salsbr01.md.comcast.net.2240: S 921638922:921638922(0)
ack 1532751027 win 5840 <mss 1460,nop,nop,sackOK> (DF)
18:34:54.775655 pcp03190206pcs.salsbr01.md.comcast.net.2240 >
192.168.1.4.http: . ack 1 win 64240 (DF)
```

In each of the attacks, the attacking host establishes a TCP connection between it and the web server. It then pushes an exploit over in the next packet. The attacking host starts each new connection on a significantly higher source port than the last connection, somewhere between 4 and 8. This indicates that the attacking host is opening multiple connections at once, and that it is probably attempting to attack several web servers simultaneously.

```
18:34:54.780297 pcp03190206pcs.salsbr01.md.comcast.net.2240 >
192.168.1.4.http: P 1:73(72) ack 1 win 64240 (DF)
18:34:54.913401 pcp03190206pcs.salsbr01.md.comcast.net.2240 >
192.168.1.4.http: R 1532751099:1532751099(0) win 0 (DF)
```

The first four such attacks:

```
68.33.141.19 - - [14/Jun/2003:18:34:54 -0400] "GET
/scripts/root.exe?/c+dir HTTP/1.0" 404 12205
68.33.141.19 - - [14/Jun/2003:18:34:55 -0400] "GET
/MSADC/root.exe?/c+dir HTTP/1.0" 404 12205
68.33.141.19 - - [14/Jun/2003:18:34:55 -0400] "GET
/c/winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205
68.33.141.19 - - [14/Jun/2003:18:34:55 -0400] "GET
/d/winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205
```

These first four attacks are attempting to exploit backdoors that CodeRed v2 leaves in different parts of the Windows directory tree.[Eeye] Since none of the backdoors exist in this case, the web server responds each time with a 404 and the attacking host closes each of the connections.

The next 12 attempts to access the server are variants of the web server folder traversal vulnerability found in Microsoft IIS 4.0 and 5.0 servers. Again, they all fail with a HTTP 404 message since this web server does not run IIS.

```
68.33.141.19 - - [14/Jun/2003:18:34:56 -0400] "GET
/scripts/..%255c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205
68.33.141.19 - - [14/Jun/2003:18:34:56 -0400] "GET
/_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
HTTP/1.0" 404 12205
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
68.33.141.19 - - [14/Jun/2003:18:34:56 -0400] "GET
/_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
HTTP/1.0" 404 12205
68.33.141.19 - - [14/Jun/2003:18:34:57 -0400] "GET
/msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../win
nt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205
68.33.141.19 - - [14/Jun/2003:18:34:57 -0400] "GET
/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205
68.33.141.19 - - [14/Jun/2003:18:34:57 -0400] "GET
/scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205
68.33.141.19 - - [14/Jun/2003:18:34:58 -0400] "GET
/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205
68.33.141.19 - - [14/Jun/2003:18:34:58 -0400] "GET
/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205
68.33.141.19 - - [14/Jun/2003:18:34:58 -0400] "GET
/scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 400 226
68.33.141.19 - - [14/Jun/2003:18:35:01 -0400] "GET
/scripts/..%%35c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 400 226
68.33.141.19 - - [14/Jun/2003:18:35:02 -0400] "GET
/scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404
12205
68.33.141.19 - - [14/Jun/2003:18:35:02 -0400] "GET
/scripts/..%252f../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 12205
```

2.2.5 Attack Mechanism

If either of the first two attacks had found root.exe then the Trojan would have attempted to use it to spread itself to the web server via TFTP. The next two attacks attempt to find open shares on the C: or D: drive (running the Windows NT OS) of the web server.

The web server folder traversal vulnerability found in Microsoft IIS 4.0 and 5.0 servers allows an attacker to use a malformed URL with Unicode characters in the place of regular characters. An un-patched IIS will decode the filename (and path) of a filename twice allowing a Unicode representation of the directory delimiter (slash) to be processed on the system as if a local user input it.

For example in the URL http://www.xyz.com/scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir, IIS interprets the %%35%63 as a (/). Hence this becomes reinterpreted as a command executable on the web server.

Each of the attacks by Nimda attempt to run the command line and list the directory in question. If any of the attacks were successful, Nimda would then be able to use the access to the command line on the system under attack to "…create open network shares on the infected computer, allowing access to the system. During this process the worm creates the guest account with Administrator privileges." [Symantec] Once this has been completed, Nimda propagates to the infected host.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

2.2.6 Correlations

Correlations are legion regarding Nimda spreading through attacking web servers in this manner:

The CERT advisory regarding Nimda:
http://www.cert.org/advisories/CA-2001-26.html

This link shows a tcpdump that demonstrates a Nimda attack against a host not running a web server.
http://personal.ie.cuhk.edu.hk/~shlam/pdemo/nimda.txt

The final SANS advisory concerning Nimda.
http://www.incidents.org/react/nimda.pdf

2.2.7 Evidence of Active Targeting

The non-sequential source port numbers mentioned earlier indicate that this web server is not the sole machine targeted. Since this server is on a residential network it is very unlikely that the targeting is anything but random.

2.2.8 Severity

Criticality: The web server is important. However, if it goes down in an attack no other service is affected. Criticality = 4.

Lethality: If the web server host were running the right OS and Web Server application then this attack would allow privilege escalation and the running of arbitrary code. It would also allow the malware to propagate itself from this web server. Lethality = 4.

System Countermeasures: The web server runs Redhat Linux 8.0 and Apache 2.046 in a chrooted jail. It has Tripwire installed and regularly audited. The system also has tcpwrappers restricting connections to the server.

The system has swatch for monitoring real-time threats. Swatch emails a log of attempts to attack the web server to the administrator as they happen. System Countermeasures = 4.

Network Countermeasures: The network does have a firewall in place in front of the web server. The firewall filters out all external initiated traffic except for that bound for the web server. The web server is isolated onto its own network segment and all external web traffic is redirected solely to it. Network Countermeasures = 3.

$$(4 + 4) – (4 + 3) = 1$$

Severity = 1

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

2.2.9 Defensive Recommendations

In this particular case the system was protected from attack because it was running a non-vulnerable web server and Operating System. The network is already physically segmented to isolate the web server and port 80 is blocked from accessing any other internal hosts by ACL.

2.2.10 Multiple Choice Question

Nimda attempts to exploit a "Web Server Folder Directory Traversal" Vulnerability to gain unauthorized access and privilege escalation to Web Servers. Which selection best describes the vulnerability?

A. IIS 4.0 and 5.0 servers misinterpret HTTP headers malformed with Unicode characters that in certain cases allow the execution of arbitrary commands.

B. Apache 1.3 and IIS 4.0 servers misinterpret HTTP headers malformed with Unicode characters that in certain cases allow the execution of arbitrary commands.

C. The mod_auth_any module allows attackers to execute arbitrary code through the placement of shell metacharacters in arguments.

D. A flaw in an ISAPI extension in IIS 4.0 and 5.0 allows for a possible buffer overflow to execute arbitrary code.

Answer: A

Explanation: Only IIS 4.0 and 5.0 are affected by this exploit that uses Unicode characters to stealthily allow access to the logical drive that the web server resides upon. mod_auth_any is an Apache module and although there is an IIS vulnerability involving ISAPI extensions, it does not involve the "Web Server Folder Directory Traversal" Vulnerability.

The original posting of this analysis can be found at http://cert.uni-stuttgart.de/archive/intrusions/2003/06/msg00250.html.

2.3 Attack Three – nmap ACK Scan or Errant Load Balancer?

2.3.1 Source of Trace

The 2002.9.26 data file used in this analysis was downloaded from http://www.incidents.org/logs/Raw/2002.9.26. The internal network range is likely to be 32.245.166.0/24 because the only source traffic in the trace from the 32.245.0.0/16 network is from within that subnet.

The destination IP addresses within the 32.245.0.0/16 network are problematic. The range of the packets detected ranged from 32.245.28.52 to 32.245.248.245.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

So either the external router is improperly configured and is forwarding all 32.245.0.0/16 traffic to 32.245.166.0/24 or (more likely) 32.245.166.0/24 is the DMZ for the 32.245.0.0/16 domain. Since the Internal Network is so large, this detect probably came from routers at an ISP or large corporation.

The sensor that captured these logs has been placed between an Internet facing router and a router internal to the 32.245.0.0/16 (or 32.245.166.0/24) network. According to Ethereal, both routers possess MAC addresses conforming to ranges assigned to Cisco Systems.

2.3.2 Detect was generated by

These alerts were generated using Snort Version 2.0.0 (Build 72) running on a Redhat Linux 8.0 (psyche) workstation. The rulesets used were downloaded on June 12th, 2003 from the snortrules-stable.tar.gz file found on the Snort website. The raw data was pulled through snort using the following command:

```
[sean@snorttest tmp]# snort -c /tmp/snort-2.0.0/etc/snort.conf -r
2002.9.26 -l log.2002.9.26
```

The -c switch tells Snort where to find the configuration file. The -r switch designates the file from which Snort should read data. The -l switch designates the directory to which alerts should be logged.

When this command was run the following alert came up several times regarding a host at 202.29.28.1:

```
[**] [1:628:2] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/26-03:05:09.446507 202.29.28.1:80 -> 32.245.90.118:80
TCP TTL:45 TOS:0x0 ID:39244 IpLen:20 DgmLen:40
***A**** Seq: 0x119  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]
```

The portion of the tcpdump that corresponds to this detect:

```
03:05:09.446507 202.29.28.1.http > 32.245.90.118.http: . ack 0 win 1400
03:05:14.446507 202.29.28.1.http > 32.245.90.118.http: . ack 1 win 1400
03:05:19.456507 202.29.28.1.http > 32.245.90.118.http: . ack 1 win 1400
18:47:17.726507 202.29.28.1.http > 32.245.28.52.http: . ack 0 win 1400
18:47:22.726507 202.29.28.1.http > 32.245.28.52.http: . ack 1 win 1400
18:47:27.776507 202.29.28.1.http > 32.245.28.52.http: . ack 1 win 1400
18:47:32.766507 202.29.28.1.http > 32.245.28.52.http: . ack 1 win 1400
```

Since the message portion of the rule that was set off has the keyword SCAN in it I looked through the scan.rules file and came across the following rule.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN nmap TCP";
flags:A,12; ack:0; reference:arachnids,28; classtype:attempted-recon;
sid:628; rev:2;)
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

This rule was activated because of a crafted TCP ACK packet sent from the Internet. ACK packets are normally sent to complete a three-way handshake to open a TCP connection. In this case the packet was sent without a SYN being sent first and a SYN/ACK being sent back.

2.3.3 Probability the source address was spoofed

These packets were obviously crafted since they are unsolicited ACKs. However despite that fact the attacker is probably engaged in reconnaissance and would then need a response. Spoofing of the address on these packets is unlikely. [http://www.whitehats.com/info/IDS28]

2.3.4 Description of Attack

202.29.28.1 sends unsolicited TCP ACK packets to two addresses within the 32.245.0.0 network. The ACKs are sent approximately 5 seconds apart. The first set of three is sent to 32.245.90.118 from 3:05:09 to 3:05:19. The remaining set is sent to 32.245.28.52 from 18:47:17 to 18:47:32.

The packets are sent from port 80 on the attacking host to port 80 within the home network. Windows and Linux IP stacks increment ports upward one for each new connection. A traceroute and a whois indicate that the attacking host resides in Thailand.

```
inetnum:        202.29.28.0 - 202.29.28.255
netname:        CHANDRA-TH
country:        TH
descr:          Rajabhat Institute Chandrakasem
admin-c:        AS133-AP
tech-c:         CS376-AP
status:         ASSIGNED NON-PORTABLE
changed:        noc@uni.net.th 20020704
changed:        phachara@uni.net.th 20021030
mnt-by:         MAINT-TH-UNINET
source:         APNIC
person:         Amnat Sawatnatee
address:        Rajabhat Institute Chandrakasem
address:        Chandrakasem, Ratchadaphisek Rd,
address:        Lardyoa,Jatujak Bangkok 10900
country:        TH
phone:          +66-2-5135690
fax-no:         +66-2-5121494
e-mail:         achandra@mozart.inet.co.th
nic-hdl:        AS133-AP
mnt-by:         MAINT-TH-UNINET
changed:        noc@uni.net.th 20020704
source:         APNIC
person:         CHALERM SRISAWAN
nic-hdl:        CS376-AP
e-mail:         chalerm@chandra.ac.th
address:        Chandrakasem,Ratchadaphisek Rd,Lardyoa,Jatujak
Bangkok10900
phone:          +66-2-5135690
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
fax-no:         +66-2-5121494
country:        TH
changed:        noc@uni.net.th 20020704
changed:        phachara@uni.net.th 20021030
mnt-by:         MAINT-TH-UNINET
source:         APNIC
```

The TTL for the packets is 45. The TTL falls in line with this box running some form of Unix. The packet is small (40 bytes) and the acknowledgement number inside the packet is 0.

All these facts point to the fact that the packet was intentionally manufactured but is not using nmap. Nmap increments its source port as it sends new ACKs on every system on which I have run it. Also, it provides a non-zero acknowledgement number.

Using a web browser and visiting the attacking host renders a challenge for a username and password to access what appears to be a web management console for an appliance built by Radware.

2.3.5 Attack Mechanism

The attack works by sending an unsolicited ACK packet to a host expecting to receive back a RST packet since there is obviously something wrong with the session. Returning a RST packet informs the attacking host that a host exists at the IP address attacked. This attack is usually used as a work-around for the fact that most sites on the Internet now block ICMP traffic from entering their networks. In this particular case, the attack is part of “Radware's proven proximity detection algorithms to choose the best ISP for outbound traffic." (http://www.westlakecom.com/linkproofloadbalancing.htm)

2.3.6 Correlations

Apparently this host in particular has been noticed on the incidents.org list. (http://cert.uni-stuttgart.de/archive/intrusions/2003/01/msg00027.html) shows the same rule being set off by this host at another site. Later on in the thread (http://cert.uni-stuttgart.de/archive/intrusions/2003/01/msg00039.html), Chris Brenton states that he believes the scans originate with a Radware load balancer.

Finally this posting (http://www.incidents.org/archives/intrusions/msg08129.html) also by Chris Brenton has a Radware LinkProof performing a whole series of tests beginning with the port 80 to port 80 ACK port scan. Apparently if the LinkProof finds a host on the other end of that connection, it then performs several tests. If it does not find a host, then all that would be seen are several ACKs from port 80 to port 80 at appproximately five seconds apart.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

2.3.7 Evidence of Active Targeting

Oddly enough, evidence of active targeting is difficult to determine. It is either an improperly configured user-defined destination that is tested by the Radware appliance to determine the health of each route available to it, or it is a random scan by the proximity detection algorithm. I tend to favor the latter, but can come to no conclusion without actually testing a Radware LinkProof appliance.

2.3.8 Severity

The severity of this attack is calculated with the following formula:

> **severity = (criticality + lethality) – (system countermeasures + network countermeasures)**

Each value is ranked on a scale from 1 (lowest) to 5 (highest).

Criticality: Although it would appear the attacker is targeting a web server, the case is that the attacker is looking for any host that will respond with a RST packet. Criticality = 1

Lethality: The worst that could happen if this attack succeeded would be that the attacker would run more tests and that the attacker would know that a host is active at this IP address. Lethality = 2

System countermeasures: It is impossible from the information given in the trace to tell if a host even exists at these IP addresses. System Countermeasures = 5

Network countermeasures: If indeed the hosts exists, then their RST packets were captured by an egress filter. Network Countermeasures = 3

> (1 + 2) – (5 + 3) = -5

2.3.9 Defensive Recommendations

Defenses can be improved against this type of intrusion by the installation of a stateful firewall and/or application proxy. A stateful firewall would drop an unsolicited ACK. An application proxy would, if it didn't drop the unsolicited ACK, drop the traffic for not being HTTP traffic.

2.3.10 Multiple Choice Questions

```
18:47:17.726507 x.y.z.1.http > a.b.c.52.http: . ack 0 win 1400
18:47:22.726507 x.y.z.1.http > a.b.c.52.http: . ack 1 win 1400
18:47:27.776507 x.y.z.1.http > a.b.c.52.http: . ack 1 win 1400
18:47:32.766507 x.y.z.1.http > a.b.c.52.http: . ack 1 win 1400
```

Which of the following is MOST LIKELY shown in the trace above?

A. nmap performing a scan of a web server.
B. Normal web traffic.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

C. A load balancing multi-homed router determining which route is optimal.
D. A disrupted three-way handshake between a web browser and a server.

Answer: C

Explanation: Normal web traffic and a disrupted three-way handshake would have the source using an ephemeral port. nmap increments upward it's source port by one with each new port scan.

The original posting of this analysis can be found at http://cert.uni-stuttgart.de/archive/intrusions/2003/06/msg00249.html.

© SANS Institute 2004, Author retains full rights.

Assignment 3 - Analyze This

3.1 Executive Summary

This analysis compiles five days of intrusion detection logs from sensors placed around a university into a list of false positives and attacks. At the end of the compilation there are defensive recommendations.

For the most part, despite the egregious amount of portscanning and attempts at hacking internal hosts, the largest issue on the campus according to these logs is file sharing in the dormitories and on some of the workstations using pooled addresses. This issue is followed closely by potential attacks by spammers attempting to exploit open mail relays and/or end-user systems. There were also sporadic exploits of end user systems by worms and Trojans.

Finally, there were occasions over the five-day period where traffic between two of the ISPs that support the university passed across the university's network. This could be indicative of

3.2 A list of files selected for analysis

The following files will be examined in this analysis. There are three kinds of files to be analyzed Scans, Alerts and Out of Specification

Scans	Alerts	Out of Spec
scans.030528	alert.030528	OOS_Report_2003_05_28_10754.txt
scans.030529	alert.030529	OOS_Report_2003_05_29_4562.txt
scans.030530	alert.030530	OOS_Report_2003_05_30_12242.txt
scans.030531	alert.030531	OOS_Report_2003_05_31_26395.txt
scans.030601	alert.030601	OOS_Report_2003_06_01_28596.txt

The scans files are generally the largest files generated by Snort. The only alerts listed in this file are portscans. This means that this version of Snort was using the portscan detection preprocessor.

Alert files contain packets that matched any of the rules listed in Snort. These rules are configured in the snort.conf and appear in this case to have been left at default.

Out of Specification files are just that, packets that are anomalous in that they do not follow the specifications for TCP, IP or some other protocol. Alerts in here would include TCP packets with contradictory flags (like ACK/FIN, or XMAS Tree), and truncated packets. These files could indicate network congestion, reconnaissance through OS Fingerprinting or bad programming on the part of a developer of a network application.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

3.3 Analysis

First a note about the log files. The data in them was evidently collected from multiple sensors arranged around the UMBC campus. Within each log file I noted that there were packets received significantly out of order by time. Time synchronization is not being done on a regular basis. Time synchronization of hosts, especially of security devices is essential to both incident analysis and response.

Some of the custom rules found in the alerts file need to be made clearer. "Contact Brian B", or "MY.NET.30.3 Activity" does not make it clear to everyone what this rule indicates, unless there is external documentation or access to the rules it makes analysis more difficult.

It appears that if the network has some form of packet filtering that it is very lenient or is at least installed behind where the Snort sensors have been placed. Port scans for the most part are coming from within the network, but those that come from outside are not prevented.

Hosts that trip the Snort rules generally fit into four categories: Compromised Hosts, File Sharers, Legitimate Users and Active Hackers.

Compromised Hosts are any hosts that have been overrun by a worm or Trojan and in this case are actively attempting to spread into the UMBC network. They set off portscanning rules as well as attempts to exploit vulnerabilities on servers. In these scans Compromised Hosts are mostly (but not exclusively) Broadband or Dialup users external to the network.

File Sharers are those hosts using any of the variety of file sharing services such as Kazaa, Gnutella, IRC XDCC, or WinMX. They set off out-of-spec rules and some custom rules made to track them (especially in the case of IRC where it appears there were problems with zombies). In these scans File Sharers are comprised of external hosts (mostly broadband users), Resnet users, and some pooled address departmental hosts.

Legitimate Users are those hosts who trip snort rules through non-malicious activity. Some users might have a session set up a source port as 65535 that would set off a false positive for possible code red traffic.

Active Hackers are those who consciously make attempts to compromise other hosts. They are extremely elusive, not necessarily due to skill but rather due to the noise caused by all the other alerts

In order to learn more about the network under investigation, I ran a batch host command on the IP addresses in the log files. This revealed the following components to the internal network:

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

MY.NET.203.29 - MY.NET.253.234: Some of these addresses are reserved for use in the dormitories. The dormitory networks are referred to at UMBC as Resnet.

MY.NET.168.10 - 249: - These addresses are reserved for a wireless network at the school.

MY.NET.150.10 - 249: - These addresses are reserved for the library system.

MY.NET.97.10 -249: - These addresses are reserved for PPP dialup connections.

MY.NET.81.2 - MY.NET.92.254: - For the most part, these addresses are devoted to the various departments at the school. They are designated as pooled addresses that may imply a DHCP arrangement for these addresses.

MY.NET.70 appears to be devoted to University Computing Services.

Some individual hosts worth mentioning:

MY.NET.100.230: this host is referred to a mailserver-ng, which could mean next generation or that it functions double duty as an NNTP server, nothing in the logs reveal this.

MY.NET.100.165: this is the Computer Science department's Web/FTP Server
MY.NET.24.47: is an FTP Server, MY.NET.24.21-23 are inbound mail servers.
MY.NET.12.2: Mail Server, MY.NET.12.4: SMTP server

3.4 Detects and Explanations

Scans

Over the five days monitored there were 4,895,983 scans detected from over 2,700 individual hosts. Although some scan alerts will be false positives, many others when correlated with other data will point to compromised hosts.

Scan Type	#
SYN	2889807
UDP	1993831
FIN	8253
VECNA	896
INVALIDACK	998
NULL	979
NOACK	644
UNKNOWN	392
XMAS	77
NMAPID	37
FULLXMAS	23

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

SYNFIN	28
SPAU	18
Total	4895983

SYN Scans (half-open): Hackers and trojans will use this scan to test for open ports by partially opening a TCP three-way handshake. Open ports send SYN/ACK response closed ports respond with a RST[8]. The majority of these scans are SYN scans. In these logs over 100 external hosts have sent 10,000 or more scans over five days. 211 internal hosts sent out 95,190 SYN scans. 3 of these hosts are responsible for over half the internal SYN scans; each sent over 10,000 scans during the period (MY.NET.83.69, 26,630, MY.NET.219.18 = 17,230; MY.NET.100.230, 10303). Further analysis will be applied to these hosts later in the report.

UDP Scans: Attackers use this form of reconnaissance to test for open UDP ports by exploiting the fact that most UDP stacks will send back an ICMP port unreachable message in response to a closed UDP port (even though the RFC technically does not require this). Rate limiting on most UDP stacks (to deal with Denial of Service) makes this a slow scanning method.[9]

Of the 1,993,831 UDP scans detected, the great majority (1,951,382) came from 214 hosts in the internal network. Most notably MY.NET.1.3 is responsible for 352,737 of them. Worth noting at this point, most of the internal UDP scanners were also internal SYN scanners. MY.NET.83.69 emitted 26630 SYN scans and 54492 scans

318 hosts from outside the network sent UDP scans(134.207.10.70, 10739; 63.250.195.10, 8108; 205.188.228.0, 7433). The third contributor of external UDP scans was actually several hosts working together from the above subnet.

FIN Scans: This portscan emits a TCP packet with the FIN flag set. On most TCP stacks, an open port will ignore a FIN packet sent to it when no session has been engaged. Closed ports respond with a RST packet. Microsoft systems do are not affected by FIN scans since they reply with an RST packet in both open and closed cases.[10]

MY.NET.219.18 sent 5,913 of the 8,253 FIN scans. All other hosts that sent these packets were under 140 scans for the time period. The great majority were under 10.

[8] Fyodor, "Nmap network security scanner man page." 2003. URL:http://www.insecure.org/nmap/data/nmap_manpage.html. 25 Jun 2003.

[9] Ibid.
[10] Ibid.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Vecna Scans: These portscans are named after their inventor. They are variants on the XMAS scan listed below[11]. Systems react to the Vecna scan much as they do to FIN (NULL and XMAS) scans. The Vecna scans are:

URG
PUSH
URG+FIN
PUSH+FIN
URG+PUSH

No external hosts yielded any Vecna scans. 148.63.164.189 sent the most with 192. More interestingly, 64.160.144.186 sent out a little bit of everything but UDP, 10 FINs, 11 Full XMAS, 137 Invalid ACK, 8 NMAPID, 225 NOACK, 273 NULL, 5 SPAU, 6 SYN, 11 SYNFIN, 75 UNKNOWN, 79 Vecna and 31 XMAS. I am uncertain as to the significance of that but it bumped itself up in priority because it looks weird.

Invalid ACK Scans: - Scans such as these are designed to determine whether a firewall holds state. Scans return either as filtered (stateful) if an RST comes back or non-filtered (stateless) if either nothing or an ICMP Host Unreachable message returns. Again there are no internal invalid ACK scans and again 64.160.144.186 comes in as the lead host. Looking at the table of scans 211.126.198.22 also has the same shotgun pattern with using the different kinds of scans.

NULL Scans: - Null scans are TCP Packets with no flags turned on at all. It seeks the same responses as other invalid flag scans (like XMAS, Vecna, and FIN). The external pair from above (64.160.144.186, 211.126.198.22) has the majority of these. 69.3.251.170, 151.196.38.234, and 141.156.142.127 demonstrate almost the same shotgun pattern (No FULLXMAS) coming in third and fourth here with NULL scans.

Unknown Scans: - Unknown scans essentially fall into the category of being defined as not being one of the other scans on the list. The external pair from above (64.160.144.186, 211.126.198.22) originate the majority of these. However there is one internal host, MY.NET.12.4, emitting these that must be investigated.

A quick check of the remaining categories reveals the same external hosts performing those scans. No further harvesting of the scan categories is necessary.

[11] Vecna, "Usual Iploggers Miss Some Variable Stealth Scans." 17 Jan 2000. URL: http://www.securityfocus.com/archive/1/42136. 25 June 2003.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Alerts

The following is a list of all the alerts found in the alert files over the five-day period.

#	Intrusion Alert
636214	SMB Name Wildcard
55627	CS WEBSERVER - external web traffic
21610	[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan. –
20129	spp_http_decode: IIS Unicode attack detected
15777	EXPLOIT x86 NOOP
14688	[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC
9597	MY.NET.30.4 activity
7487	External RPC call
6765	spp_http_decode: CGI Null Byte attack detected
6475	Queso fingerprint
3299	High port 65535 udp - possible Red Worm - traffic
3099	TCP SRC and DST outside network
2784	MY.NET.30.3 activity
2690	CS WEBSERVER - external ftp traffic
1736	IDS552/web-iis_IIS ISAPI Overflow ida nosize
1698	Null scan!
1240	Possible trojan server activity
920	Incomplete Packet Fragments Discarded
713	High port 65535 tcp - possible Red Worm - traffic
646	SNMP public access
556	[UMBC NIDS IRC Alert] Possible Incoming XDCC Send Request Detected.
299	NMAP TCP ping!
218	[UMBC NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC
160	TFTP - Internal TCP connection to external tftp server
121	Tiny Fragments - Possible Hostile Activity
104	SUNRPC highport access!
97	Notify Brian B. 3.54 tcp
87	TFTP - Internal UDP connection to external tftp server
86	IRC evil - running XDCC
82	Notify Brian B. 3.56 tcp
60	FTP passwd attempt
56	EXPLOIT x86 stealth noop
55	EXPLOIT x86 setuid 0
51	SMB C access
38	EXPLOIT x86 setgid 0
24	Probable NMAP fingerprint attempt
20	IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
17	RFB - Possible WinVNC - 010708-1
15	SYN-FIN scan!
12	NETBIOS NT NULL session
11	NIMDA - Attempt to execute cmd from campus host

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

10	EXPLOIT NTPDX buffer overflow
8	TFTP - External UDP connection to internal tftp server
7	Attempted Sun RPC high port access
6	connect to 515 from outside
4	External FTP to HelpDesk MY.NET.70.50
3	DDOS mstream client to handler
3	External FTP to HelpDesk MY.NET.70.49
3	External FTP to HelpDesk MY.NET.83.197
3	TCP SMTP Source Port traffic
3	TFTP - External TCP connection to internal tftp server
2	DDOS shaft client to handler
2	External FTP to HelpDesk MY.NET.53.29
2	[UMBC NIDS IRC Alert] User joining Warez channel detected. Possible XDCC bot
1	DDOS TFN Probe
1	EXPLOIT FTP passwd retrieval retr path
1	Fragmentation Overflow Attack
1	ICMP SRC and DST outside network
1	Traffic from port 53 to port 123
1	[UMBC NIDS IRC Alert] K\:line'd user detected, possible trojan.
1	[UMBC NIDS IRC Alert] Possible trojaned box detected attempting to IRC
1	[UMBC NIDS IRC Alert] User joining XDCC channel detected. Possible XDCC bot

The first practical step with these alerts is to determine which alerts were set off by traffic originating in the MY.NET network. From a practical standpoint these are hosts which administrators have a better chance of being able to effect change upon. Furthermore, hosts that are setting off alerts on the internal network have a much higher chance of being compromised.

Exclusively Internal Alerts

These are alerts that originated solely on the internal network.

#	Internal Intrusion Detection Alerts
14688	[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC
9597	MY.NET.30.4 activity
2784	MY.NET.30.3 activity
218	[UMBC NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC
86	IRC evil - running XDCC
11	NIMDA - Attempt to execute cmd from campus host
4	External FTP to HelpDesk MY.NET.70.50
3	External FTP to HelpDesk MY.NET.83.197
3	External FTP to HelpDesk MY.NET.70.49
2	External FTP to HelpDesk MY.NET.53.29
1	[UMBC NIDS IRC Alert] Possible trojaned box detected attempting to IRC

XDCC client detected attempting to IRC

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

XDCC (eXtended Direct Client-to-Client) is a means of transferring files directly over IRC often circumventing firewall rules. Although the rule says "attempting" the fact is that the XDCC sessions in some cases seem to be successful.

The following internal hosts engaged in XDCC sessions: MY.NET.88.163, MY.NET.83.100, MY.NET.198.221, MY.NET.91.151, MY.NET.80.209, MY.NET.83.173, MY.NET.80.149, MY.NET.140.136

MY.NET.30.x Activity

These custom alerts are put in place to monitor activity into MY.NET.30.X network. It looks like the rule is set only to catch inbound connections. There are apparently files being transferred since the same IP and port numbers are open on a number of the packets.

Looking at the distribution of ports open. These hosts appear to be Windows 2000 file servers with web servers in operation. It appears that 30.4 also runs a secure Novell IFolder service.[12] The one to eleven hits on different ports appear to be caused by port scanners.

30.3 Hits	Ports	30.4 Hits	Service
0	52080	1	IFolder Server
0	51443	4264	IFolder Server
0	17300	2	
0	8081	1	
803	8009	215	Apache JServ Protocol
1	8008	3	
11	3019	0	
5	1433	3	
1782	524	634	NCP
58	445	71	Win2K SMB
2	139	4	
1	137	7	
7	135	2	
112	80	4386	HTTP
1	25	1	
1	21	3	
2784		9597	

Possible sdbot floodnet detected attempting to IRC

SDBot is a backdoor Trojan that allows remote control of a Windows host via IRC.[13] All of these packets are coming from the MY.NET.97.X. All of them also

[12] Novell. "Things You Should Know During the Installation". 2003. URL:http://www.novell.com/documentation/lg/ifolder20/ifolder/data/ah00j8c.html. 2003 Jun 25.
[13] Sevcenco, Serghei. "Symantec Security Response - Backdoor.Sdbot." 2003 Jun 16. URL: http://securityresponse.symantec.com/avcenter/venc/data/backdoor.sdbot.html. 2003 Jun 25.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

appear to be unsuccessful, as the ports appear to be increasing on the source side while the ports on the destination side appear to move around between the open IRC ports.

There is only one host that appears to be infected by this Trojan (MY.NET.97.188 = 212). The fact that the rest of the hosts make one or two attempts to connect to IRC in this manner indicates that there may be an attacker on campus attempting to manage IRC zombies.

IRC evil - running XDCC
These are XDCC servers running on campus. They appear to be successfully sending files. The following hosts are engaged in hosting XDCC: MY.NET.80.209, MY.NET.132.24, MY.NET.88.163, MY.NET.218.38, MY.NET.219.90, MY.NET.132.27, MY.NET.218.206, MY.NET.219.14, MY.NET.5.43. Many of the same hosts that are hosting XDCC are also downloading files from XDCC in the "XDCC client detected attempting to IRC" alert above.

NIMDA - Attempt to execute cmd from campus host
Nimda among other worms attempts to gain access to Windows Servers running IIS by exploiting cmd.exe on those hosts[14]. Nimda however makes several attempts at attacking the host. Each of these hosts only make 1 to 3 attempts.

External FTP to HelpDesk MY.NET.X.X
I suppose this custom rule is to detect rogue FTP servers running on Helpdesk workstations. This being the case it appears to be detecting port scans as false positives.

Possible trojaned box detected attempting to IRC
This is also a box on the MY.NET.97.XX subnet (MY.NET.97.215). Without seeing the raw packet it is difficult to know what distinguishes this rule from the sdbot rule above.

Alerts that both originate Internally or Externally

ESRC	ISRC	Total Alerts	Intrusion Detected
635766	448	636214	SMB Name Wildcard
1235	18894	20129	spp_http_decode: IIS Unicode attack detected
15769	8	15777	EXPLOIT x86 NOOP
7479	8	7487	External RPC call
38	6727	6765	spp_http_decode: CGI Null Byte attack detected
6473	2	6475	Queso fingerprint
2264	1035	3299	High port 65535 udp - possible Red Worm - traffic
1715	21	1736	IDS552/web-iis_IIS ISAPI Overflow ida nosize

[14] Chien, Eric, "Symantec Security Response - W32.Nimda.A@mm." 15 Jan 2003. URL:http://www.sarc.com/avcenter/venc/data/w32.nimda.a@mm.html. 25 Jun 2003

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

644	596	1240	Possible trojan server activity
909	11	920	Incomplete Packet Fragments Discarded
331	382	713	High port 65535 tcp - possible Red Worm - traffic
86	74	160	TFTP - Internal TCP connection to external tftp server
13	74	87	TFTP - Internal UDP connection to external tftp server
9	8	17	RFB - Possible WinVNC - 010708-1

SMB Name Wildcard

Over 27,000 different hosts set off this alert over the five-day period. All internal hosts that did so only set off one packet. Despite the high rate of false positives many of the top talkers in this group are indeed scanning.

IP Address	Alertsd
64.110.60.147	7402
218.28.149.18	1085
211.152.59.26	1058
151.196.168.149	945
209.103.204.110	902
220.163.13.34	573
141.157.123.28	571
209.103.223.123	535

IIS Unicode attack detected

IIS Unicode attack detected is thrown by Snort's http_decode preprocessor. It attempts to detect activity by a variety of worms to take advantage of the Web Server folder traversal vulnerability found in Microsoft IIS 4.0 and 5.0 servers.

Although false positives are possible for a variety of reasons (urls using double-bytes, some CGI and Java)[15], hosts with a large number of alerts for this should undergo further scrutiny.

Internal sources of these alerts concern me more than internal destinations. Internal sources could be indicative of a compromised host. Over 390 internal hosts set off this alert. Here is the top ten:

IP Address	Alerts
MY.NET.153.152	3591
MY.NET.97.177	1500
MY.NET.97.126	591
MY.NET.88.171	581
MY.NET.152.165	507

[15] Leweling, Martin. "Re: [suse-security] Sy[s]tem Attack or ?" 27 Mar 2001. URL: http://lists.suse.com/archive/suse-security/2001-Mar/0371.html. 25 Jun 2003.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

MY.NET.153.176	492
MY.NET.84.216	374
MY.NET.97.58	317
MY.NET.97.221	301
MY.NET.91.2	240

EXPLOIT x86 NOOP
These alerts could be binary files download from our web servers. This alert is set off when executable files are transferred from or to a web server. You can usually tell the false positives from the fact that the source port will be 80 if it is a web server serving a binary to a requesting user.[16]

However, in this case, several of the attacking hosts are attempting to hack hosts apparently at random, trying to send code to port 80 on several hosts on the dialup network. The top talker for this alert, 138.88.171.119 (a DSL user from Verizon) does this.

External RPC Call
External RPC calls could be attempts to access a Sun portmapper service on a host. The portmapper service directs remote users to higher ports above 32000 that run the actual service.

In regards to these alerts, all of them appear to be portscans. Most of the attacking hosts are probably probing single internal hosts scanning a range of services and we happened to catch them with portmapper. However, the one host that generated the great majority of these alerts is looking specifically for a host running portmapper. Over seven thousand alerts were generated by 200.179.85.42 scanning what seems to be the entire MY.NET network. The packets were obviously crafted. The source and destination ports are both 111.

CGI Null Byte attack detected
This preprocessor rule is thrown if a %00 is seen in an HTTP request. As such it could potentially raise false positives on site using Cookies and site with Double Byte characters[17]. Because of this internal web servers should be the focus of this rule. Searching for internal web servers I noticed the following alerts.

```
05/29-08:15:51.868437  [**] spp_http_decode: CGI Null Byte attack
detected [**] 151.196.98.43:19702 -> MY.NET.24.34:80
05/29-08:15:52.344871  [**] spp_http_decode: CGI Null Byte attack
detected [**] 151.196.98.43:19741 -> MY.NET.24.34:80
05/29-08:20:45.440293  [**] spp_http_decode: CGI Null Byte attack
detected [**] 151.196.98.43:25995 -> MY.NET.104.177:80
```

[16] Nathan, Jeff, "Re: [Snort-users] SHELLCODE x86 NOOP." 7 Mar 2002. URL:http://archives.neohapsis.com/archives/snort/2002-03/0159.html. 25 Jun 2003.
[17] Roesch, Martin, Caswell, Brian. et al., "[Snort-users] Snort FAQ." 29 Jun 2002. URL:http://www.geocrawler.com/archives/3/4890/2002/6/0/9059445/. 25 Jun 2002.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
05/29-08:14:35.782122  [**] spp_http_decode: CGI Null Byte attack
detected [**] 151.196.98.43:14386 -> MY.NET.24.34:80
05/29-08:14:36.198726  [**] spp_http_decode: CGI Null Byte attack
detected [**] 151.196.98.43:14416 -> MY.NET.24.34:80
```

These packets deserve further investigation.

Queso Fingerprint
Queso is an OS Fingerprinting application. It sends malformed TCP packets to hosts to determine their operating systems[18]. Because of this once again the destination hosts are of greater interest because this may be reconnaissance and hence prelude to an attack.

The scans are launched against over 100 named hosts in the internal network. Not surprisingly many of the scanning hosts show up in the OOS top ten talkers.

High port 65535 udp/tcp - possible Red Worm – traffic
This rule could emit a false positive. Many file sharing and game services send files through high ports. Also, every host will eventually use 65535 to transmit information as part of normal operation of these protocols. Looking at the top ten (eleven cause I wanted to show http) ports on the other end of this connection most of them look like false positives.

Port	Service
5121	game
6257	winmx
25	smtp
6346	gnutella
21	ftp
1237	jabber?
2315	?
2472	?
1214	kazaa
7701	Java Client File sharing
80	http

Checking out the web traffic and FTP traffic all of it seems to be coming from legitimate servers.

IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
This appears to be a custom rule to deal with the ISAPI overflow vulnerability to which unpatched IIS systems are vulnerable. This alert is likely to be a real attack. However since this vulnerability was exposed in 2001, university systems

[18] T3, "The Science of OS Fingerprinting." 2001. URL:http://toorcon.org/2001/lineup/osfinger.ppt. 25 Jun 2003.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

should be patched. However Resnet and dialup systems may not be as well protected.

Searching through the source addresses of these alerts only three come up as being from inside the university network. All three are on dialup, all three also have prodigious amounts of network scans, CGI null byte character attacks, IIS exploit attempts and NIMDA cmd.exe exploit alarms.

IP Address	Time range of attacks
MY.NET.97.46	05/31/2003: 0117-0118 06/01/2003: 1723-1732
MY.NET.97.53	05/28/2003: 1040-1045 05/29/2003: 0704-0712
MY.NET.97.68	05/28/2003: 1739-741 05/29/2003: 0854 05/30/2003: 0051-0055 05/31/2003: 0235-0335

It is likely that Nimda downloaded via e-mail has compromised these systems. In order to find the actual compromised systems the IP addresses and timestamps will need to be correlated with user logs authenticating dialup sessions. It is in fact possible that there is more than one system involved with each IP address.

Possible Trojan server activity

This rule is thrown by sessions where one side of the session is using port 27374. It is highly probable if many alerts come from a host on this port that it has succumb to the SubSeven backdoor Trojan[19] (which runs as a service on this port by default). For about a minute on 5/29 MY.NET.24.47 engaged in a session that could be SubSeven traffic.

```
05/29-23:23:49.805765  [**] Possible trojan server activity [**]
MY.NET.24.47:20 -> 192.207.69.1:27374
05/29-23:23:50.083816  [**] Possible trojan server activity [**]
192.207.69.1:27374 -> MY.NET.24.47:20
05/29-23:23:50.084598  [**] Possible trojan server activity [**]
192.207.69.1:27374 -> MY.NET.24.47:20
05/29-23:23:50.084950  [**] Possible trojan server activity [**]
MY.NET.24.47:20 -> 192.207.69.1:27374
05/29-23:23:50.097639  [**] Possible trojan server activity [**]
MY.NET.24.47:20 -> 192.207.69.1:27374
05/29-23:23:50.160078  [**] Possible trojan server activity [**]
192.207.69.1:27374 -> MY.NET.24.47:20
05/29-23:23:50.160131  [**] Possible trojan server activity [**]
192.207.69.1:27374 -> MY.NET.24.47:20
```

Since this is an FTP server it is more than like an Active FTP session. Someone on MY.NET.24.44 engaged in similar behavior using 80 as a source port. Of

[19] Chezz, "SubSeven Frequently Asked Questions." 10 Oct 2002. URL: http://520038635832-0001.bei.t-online.de/s7/sub/help.shtml. 25 Jun 2003.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

course 24.44 happens to be a web server. These are both very likely to be false positives.

However, 67.2.167.3 scanned the MY.NET.230.X network for open SubSeven ports. Since attackers generally want information back from their scans this is probably a legitimate IP address (although it is likely to be a compromised machine itself).

TFTP – Internal TCP connection to external tftp server
This could be a Trojan disseminating out of the network[20]. These hosts should be monitored to see if they end up in any of the other alerts. MY.NET.83.69, MY.NET.91.252, MY.NET.97.22.

Events Exclusively External in Origin

#	Intrusion Detection
55627	CS WEBSERVER - external web traffic
21610	[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan. -
3099	TCP SRC and DST outside network
2690	CS WEBSERVER - external ftp traffic
1698	Null scan!
646	SNMP public access
556	[UMBC NIDS IRC Alert] Possible Incoming XDCC Send Request Detected.
299	NMAP TCP ping!
121	Tiny Fragments - Possible Hostile Activity
104	SUNRPC highport access!
97	Notify Brian B. 3.54 tcp
82	Notify Brian B. 3.56 tcp
60	FTP passwd attempt
56	EXPLOIT x86 stealth noop
55	EXPLOIT x86 setuid 0
51	SMB C access
38	EXPLOIT x86 setgid 0
24	Probable NMAP fingerprint attempt
20	IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
15	SYN-FIN scan!
12	NETBIOS NT NULL session
10	EXPLOIT NTPDX buffer overflow
8	TFTP - External UDP connection to internal tftp server
7	Attempted Sun RPC high port access
6	connect to 515 from outside
3	TFTP - External TCP connection to internal tftp server
3	TCP SMTP Source Port traffic
3	DDOS mstream client to handler

[20] Chien, Eric, "Symantec Security Response - W32.Nimda.A@mm." 15 Jan 2003. URL:http://www.sarc.com/avcenter/venc/data/w32.nimda.a@mm.html. 25 Jun 2003

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

2	DDOS shaft client to handler
2	[UMBC NIDS IRC Alert] User joining Warez channel detected. Possible XDCC bot
1	Traffic from port 53 to port 123
1	ICMP SRC and DST outside network
1	Fragmentation Overflow Attack
1	EXPLOIT FTP passwd retrieval retr path
1	DDOS TFN Probe
1	[UMBC NIDS IRC Alert] User joining XDCC channel detected. Possible XDCC bot
1	[UMBC NIDS IRC Alert] K\:line'd user detected, possible trojan.

CS Webserver – external web/ftp traffic
This custom rule is designed to monitor traffic from outside the UMBC network pointing at the Computer Science department web server. Examining the source ports and the destination ports I note no suspicious traffic in the logs generated by this rule. Without the raw packets not much more can be told.

IRC user /kill detected, possible Trojan
A /kill command on IRC is issued by an IRC Operator (not a Channel Operator). It boots a host from IRC for some egregious violation. So in most cases it is likely that this is a false positive.

There were over 50 hosts that received kills. All of the kills came from irc-1.aniverse.net (an IRC server apparently devoted to anime). The top three produced over 100 kills each: MY.NET.190.95, 17631; MY.NET.88.163, 3573; MY.NET.97.188, 214. They should be monitored for correlations in other alerts.

TCP SRC and DST outside network / ICMP SRC and DST outside network
Returning back to the first set of data this is the first issue that should be addressed. Apparently there are some hosts that are using UMBC as a transit domain. Generally universities are not in the business transit providers, so either there is a poorly configured routing protocol and/or some proxy abuse afoot.

Looking at the alerts 408 out of the 445 different source hosts in the alerts are sending packets to 67.80.77.94. Checking Dshield this host happens to be a Cablevision customer in New York.

All of the packets being sent are being sent to destination port 6112 from a variety of host ports. Several games happen to use this port, but why are they crossing through UMBC to get to the game server?

My first urge is to say that this is the result of spoofed IP addresses within the UMBC network, I am reluctant to just pin it to that. So time to drill down into the source IP addresses, here are the top ten hosts:

Source IP	Hits	Hostname
64.214.178.105	128	64-214-178-105.nrp5.brv.mn.frontiernet.net

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

24.192.204.120	104	CPE004005b8a148-CM00803786ab4f.cpe.net.cable.rogers.com
66.131.87.254	76	modemcable254.87-131-66.nowhere.mc.videotron.ca
68.37.33.153	73	pcp01722319pcs.union01.nj.comcast.net
24.42.178.103	64	rogers cable canada
68.115.14.127	63	c68.115.14.127.stp.wi.charter.com
68.45.128.11	60	pcp096318pcs.audubn01.nj.comcast.n
65.41.153.118	59	user118.net732.fl.sprint-hsd.net
68.9.94.198	58	ip68-9-94-198.ri.ri.cox.net.
68.168.217.85	52	fl-pbg-2b-b-85.atlsfl.adelphia.net
216.46.70.30	52	ip-216-46-70-30.dsl.nyc.megapath.net

Looking at these hosts 9 out of 11 of them are geographically to the north of the university. If the addresses were being spoofed I would expect a more random distribution. The next things to do are a few traceroutes.

One from Wisconsin…

```
traceroute to MY.NET.1.3 (MY.NET.1.3), 30 hops max, 40 byte packets
 1  e3-13.foundry1.cs.wisc.edu (198.133.224.116)  1.933 ms  1.572 ms  1.384 ms
 2  g0-11.cisco-border.cs.wisc.edu (128.105.1.89)  0.914 ms  0.946 ms  0.776 ms
 3  144.92.128.194 (144.92.128.194)  1.186 ms  1.305 ms  1.020 ms
 4  r-peer-vlan-1500.net.wisc.edu (146.151.164.49)  1.098 ms  1.106 ms  1.137 ms
 5  UWMadisonISP-atm0-0-252.core.wiscnet.net (216.56.1.17)  1.107 ms  1.374 ms  1.132 ms
 6  r-uwmilwaukee-isp-atm1-0-1.wiscnet.net (140.189.8.2)  3.768 ms  3.409 ms  3.389 ms
 7  205.213.118.2 (205.213.118.2)  12.915 ms  12.966 ms  12.928 ms
 8  nycmng-chinng.abilene.ucaid.edu (198.32.8.83)  33.276 ms  33.297 ms  33.620 ms
 9  washng-nycmng.abilene.ucaid.edu (198.32.8.85)  37.288 ms  38.882 ms  37.345 ms
10  dcne-abilene-oc48.maxgigapop.net (206.196.177.1)  37.487 ms  37.252 ms  37.336 ms
11  clpk-so3-1-0.maxgigapop.net (206.196.178.46)  37.667 ms  37.546 ms  37.547 ms
12  MY.NET.16.253 (MY.NET.16.253)  38.761 ms  38.927 ms  38.321 ms
```

and one from the University of Maryland…

```
traceroute to MY.NET.1.3 (MY.NET.1.3), 30 hops max
 1  Vlan5.css-nts-r1.net.umd.edu (128.8.5.252)  0.526 ms
 2  Gi5-1.css-core-r1.net.umd.edu (128.8.0.9)  0.399 ms
 3  Gi3-2.ptx-fw-r1.net.umd.edu (128.8.0.86)  0.380 ms
 4  129.2.0.242 (129.2.0.242)  0.489 ms
 5  ge-1-2-0.umcp-gw.net.ums.edu (131.118.255.229)  0.555 ms
 6  so-0-3-2.umab-gw.net.ums.edu (131.118.255.41)  1.475 ms
 7  pos0-0-0.umbc-gw.net.ums.edu (131.118.255.18)  2.026 ms
```

and one from the Faroe Islands. (I wanted someplace exotic)

```
1  feth1-0-0.bone2.olivant.fo (212.55.32.1)  0.750 ms  0.564 ms  0.610 ms
 2  212.55.32.98 (212.55.32.98)  2.238 ms  2.011 ms  1.761 ms
 3  Faereyjar-Lina-2-R-Thorsh-3004-gw.simnet.is (212.30.213.133)  15.361 ms Faereyjar-
Lina-1-R-Thorsh-3003-gw.simnet.is (212.30.213.129)  14.463 ms  14.577 ms
 4  ls-bb.isholf.is (157.157.173.202)  19.588 ms  16.239 ms  14.972 ms
 5  ls-usa-gw.isholf.is (157.157.173.200)  17.042 ms  14.689 ms  15.721 ms
 6  500.POS2-0.IG2.NYC4.ALTER.NET (157.130.0.201)  81.891 ms  78.647 ms  79.538 ms
 7  189.at-6-1-0.XR2.NYC9.ALTER.NET (152.63.0.29)  88.918 ms  105.046 ms  94.360 ms
 8  0.so-2-1-0.XL2.NYC9.ALTER.NET (152.63.23.141)  110.983 ms  96.590 ms  91.536 ms
 9  POS7-0.BR1.NYC9.ALTER.NET (152.63.18.221)  102.052 ms  111.439 ms  87.720 ms
10  204.255.174.130 (204.255.174.130)  106.896 ms  124.577 ms  111.759 ms
11  ewr-core-02.inet.qwest.net (205.171.17.129)  114.390 ms  132.226 ms  111.120 ms
12  ewr-core-03.inet.qwest.net (205.171.17.34)  90.617 ms  95.380 ms  104.585 ms
13  dca-core-02.inet.qwest.net (205.171.8.181)  113.609 ms  143.169 ms  142.120 ms
14  dca-core-03.inet.qwest.net (205.171.9.50)  117.296 ms  132.049 ms  115.609 ms
15  wdc-core-03.inet.qwest.net (205.171.8.214)  145.223 ms  113.564 ms  137.547 ms
16  wdc-edge-07.inet.qwest.net (205.171.24.130)  111.713 ms  126.185 ms  165.467 ms
17  63.146.1.34 (63.146.1.34)  133.276 ms  117.469 ms  138.458 ms
18  pos0-0-0.umbc-gw.net.ums.edu (131.118.255.18)  130.909 ms  117.073 ms  112.573 ms
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

I've performed several traces from different parts of the world and apparently in order to get to MY.NET.1.3, one either ends up going through MY.NET.16.253 or pos0-0-0.umbc-gw.net.ums.edu. This is good in that it allows for network robustness one of the border routers can go down and the network will still work. However, it is not difficult to imagine a situation with the poor configuration of an exterior gateway protocol where UMBC could find itself acting as a transit conduit between qwest and maxgigapop.

Null Scan
The majority of these scans were from a computer in Japan to what appears to be a library terminal at UMBC. The scans source port was 0 and the destination port was also 0. 1496 of the 1698 packets had identical port numbers (0 and 0). I think it may be the same type of traffic described in this posting (http://archives.neohapsis.com/archives/incidents/2000-09/0011.html).

SNMP Public Access
This rule catches external hosts accessing the Simple Network Management Protocol inside the UMBC campus. It is currently catching a series of scans that appear to be directed at the campus printers. These could be false positives sent out by Unix workstations printing to these printers and checking first using SNMP to see if they are operational.[21]

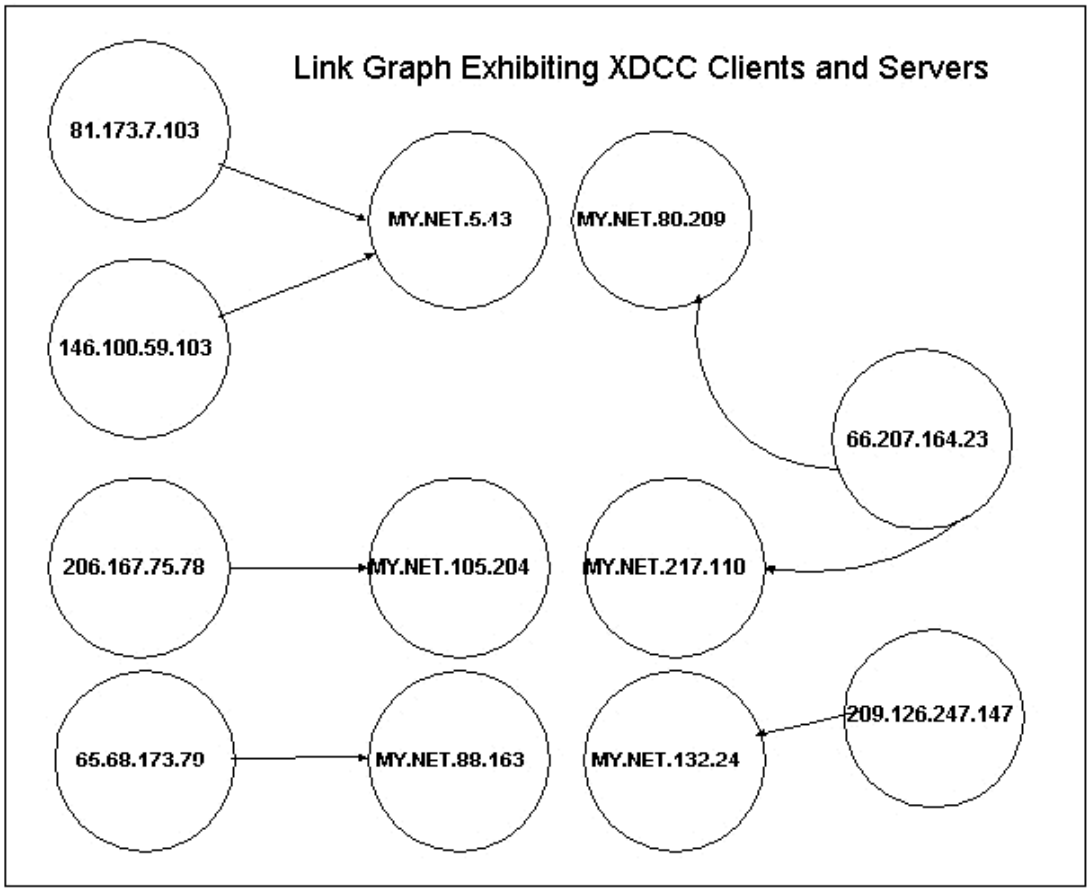
My question is why do people from the Navy need to print remotely to printers at UMBC? I would block the outside world from accessing this port in general.

[UMBC NIDS IRC Alert] Possible Incoming XDCC Send Request Detected.
XDCC servers being hosted by internal hosts probably set off this alert. This is not a very secure means of transferring files, furthermore files transferred in this way are often in violation of copyright laws and could open the university up to liability especially if the server is hosted upon a university managed computer (as opposed to a student's computer).

Fortunately there are few internal hosts found in these alerts.

[21] Lieberman, Jeff, "Re: Network printer trouble" 24 Jul 2000. URL:http://www.aplawrence.com/Bofcusm/434.html. 25 Jul 2003.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.



It appears that MY.NET.5.43 is serving more than one host. This could mean that it is acting as a XDCC file server. Since it has an IP address that is probably a university-managed computer, it should be investigated.

NMAP TCP ping!
See invalid ACK scans.

Tiny Fragments - Possible Hostile Activity
This alert means that attacking hosts are sending fragments of traffic which when reassembled could form either a teardrop attack or an attempt at reconnaissance by forcing the victim host into emitting an ICMP error response.

SUNRPC highport access!
From looking at the packets it appears this alert is tripped if someone accesses port 32771 on an internal server. All the web traffic comes from legitimate sites.

This was interesting; someone from the university using SSH to access their home computer which is connected to the Internet with Verizon DSL.

```
05/28-01:08:57 209.249.182.79:22 -> MY.NET.162.71:32771
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

These all appear to be false positives.

Notify Brian B. 3.54 tcp/Notify Brian B. 3.56 tcp
Apparently Brian B is to be alerted if anyone attempts to access MY.NET.3.54 or MY.NET.3.56, several external hosts attempted to contact those hosts over ports generally used by Windows systems (445, 135) and file servers (80, 21).

FTP passwd attempt
This alert is tripped if an attacker attempts to retrieve the passwd file from the FTP server (MY.NET.24.47). 217.171.225.84, 212.154.148.212, 218.4.166.53, and 67.113.225.69 all attempted this more than once.

3.5 Top Talkers

Scans
The top ten scanners by sheer number of scans are all on the internal network.

#	Source IP	# of scans	Explanation
1	MY.NET.1.3	353863	Network Health Monitor
2	MY.NET.217.146	169558	Resnet
3	MY.NET.218.230	104916	Resnet
4	MY.NET.83.69	81122	Deptarmental Pool
5	MY.NET.97.238	80391	Dialup
6	MY.NET.137.7	64920	Unknown (DNS?)
7	MY.NET.86.110	53621	Deptarmental Pool
8	MY.NET.84.178	47280	Deptarmental Pool
9	MY.NET.217.158	45992	Resnet
10	MY.NET.97.21	45039	Dialup

The top scanner is a Network Health Monitor. This can be marked off as a false positive. #6 appears to be a DNS server without a hostname of it's own, the top twenty hosts it is scanning are DNS servers to which it is connecting over port 53 UDP. I'm not sure whether it is a rogue DNS server or not but it has no hostname of it's own.

#2, #3, and #9 are hosts in Resnet. #2 and #9 are running Kazaa[22], 1214 is the port that is most scanned other ports above 1024 through 3999 are scanned in almost equal measure. #3 appears to be running WinMX, another file sharing service. The scans from #3 are almost exclusively UDP on port 6257[23].

[22] Scarborough, M. "Re: TCP Dest Port 1214." 1 Jun 2001. URL: http://www.incidents.org/archives/intrusions/msg03822.html. 25 Jun 2003.
[23] Buchanan, J. "Working Around ISP Port Blocks." 2003 URL:" http://homepage.ntlworld.com/j.buchanan/index.html?blocked.html," 25 Jun 2003

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

#4, #7 and #8 from their hostnames appear to be print spoolers. #4 appears to running eDonkey2000[24] another file sharing service. #7 and #8 both appear to running WinMX. If these hosts are print spoolers they need to be quarantined and purged of these file-sharing services.

#5 and #10 are on dialup. They may be the same machine dialing in at different times. Apparently these attackers are scanning for open SMB shares on a wide range of hosts.

Alerts
Here are the top ten non-scan related talkers by IP address:

IP Address	#
66.207.164.23	26836
MY.NET.88.163	11003
64.110.60.147	10648
MY.NET.83.100	8276
200.179.85.42	7827
216.39.48.2	6263
138.88.171.119	4852
66.77.73.236	4264
MY.NET.153.152	3742
66.117.30.14	3564

#1 66.207.164.23 irc-1.aniverse.net
This is the IRC server from which all the /kill commands were issued. The MOTD (Message of the Day) on the aniverse network states that they will deny access to anyone attempting to share copyrighted files on their networks.

#2 MY.NET.88.163 pooled address host
This is a pooled address host at the school running XDCC on the aniverse server listed above. It was also the server that received the highest amount of kill commands from aniverse.

#3 64.110.60.147 host-64-110-60-147.interpacket.net
This host did an SMB wildcard scan of over 7,000 hosts in the MY.NET network. It scanned each host once then set off 6 exploit X86 NOOP alerts at MY.NET.190.93. Oddly, it didn't scan 190.93. Five hours later 194.216.113.6 tripped off over 500 X86NOOP alerts against MY.NET.190.93. They could have been downloading files from it over an open Windows or Samba share. The only way to be certain is to check the packet capture.

---

[24] MetaMachine, "What Ports does eDonkey use?" 2002. URL:http://www.edonkey2000.com/cgi-bin/smartfaq/smartfaq.cgi?answer=1025114514&id=1025114052. 25 June 2003.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

#4 MY.NET.83.100 pooled host address
This is another pooled address host at the school running XDCC.

#5 200.179.85.42 Embratel
This was the host that was crafting packets in an attempt to scan for portscanner services. Scans are listed as being made by this address on port 111 on Dshield for this host on June 1st.

#6 216.39.48.2 trek21.sv.av.com
Throughout the five day period this host was interacting with the CS webserver. It is difficult to analyze this further without knowing more about the alert or the packets captured. All the connections were made to port 80 of the CS webserver, with occasional connections to port 80 of 30.4.

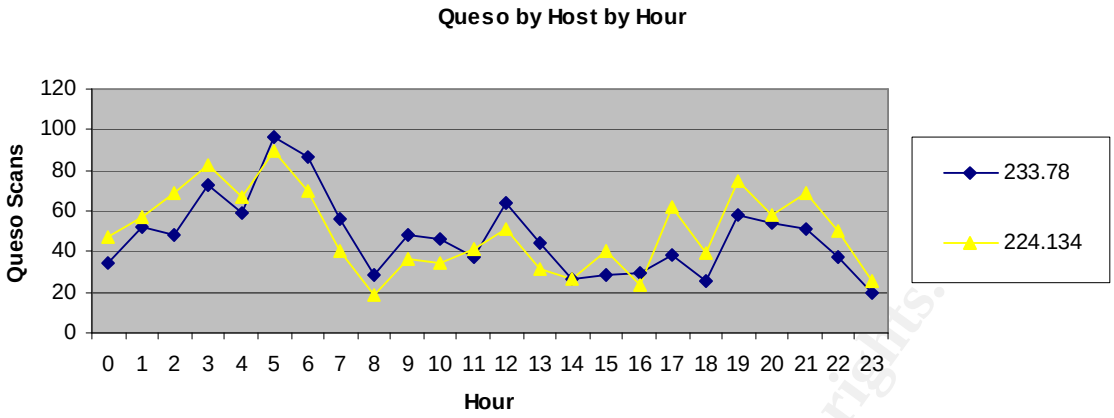
#7 138.88.171.119 pool-138-88-171-119.res.east.verizon.net
This talker first visited MY.NET.3.54 at 2115 on the 29th, then the talker visited MY.NET.30.3 and 30.4 five minutes later after that the talker performed portscans against a wide swath of the network. Finally an hour later, from 22:30 – 23:13, this talker set off over 3000 EXPLOIT x86 NOOP alerts against 50 hosts, most attempts were against hosts in the .198 subnet. It visited each site on port 80. Considering the pattern of this chatter it is almost definitely an attack.

#8 66.77.73.23 rulfwd002.sac2.fastsearch.net
This host is engaged in the same activity that the host in #6 is engaged. Again this is difficult to analyze without the packets or the rule.

#9 MY.NET.153.152 library terminal?
All the sites visited in this log are in Korea. This is a library user surfing Korean web pages for four hours. The Unicode alerts are false positives.

#10 66.117.30.14 New Horizon Collocations
This is the host responsible for most of the Queso alerts over the five-day period. This host is also showing up as the number one talker in OOS. In both cases it is attempting to communicate to two hosts in Resnet over destination port 1182 for the entire five-day period. These are the only alerts for these three hosts for the entire period. As you can see in this chart the streams sent from each Resnet host on port 1182 stay pretty much in synch the whole time.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.



OOS

At first, I was going to define top ten OOS talkers by attacking IP address. As I performed a cursory inspection of the attacking IP address file, I noticed several of the OOS packets were coming from the same subnets. So I wrote a script that took subnet of the attacking IP addresses and correlated them with destinations and destination Ports. I then found that the two scripts came up with very different answers as to what the top ten threats were:

Source -> Destination: DPort	#	Source Sub -> Destination: DPort	#
66.117.30.14 -> MY.NET.233.78:1182	4009	66.117.30 -> MY.NET.233.78:1182	4009
66.117.30.14 -> MY.NET.224.134:1182	3964	66.117.30 -> MY.NET.224.134:1182	3964
148.64.150.139 -> MY.NET.206.102:1995	1024	216.95.201 -> MY.NET.6.47:25	1103
210.253.206.180 ->MY.NET.211.26:6011	922	216.95.201 -> MY.NET.24.23:25	1078
148.63.164.189 -> MY.NET.240.202:2570	811	216.95.201 -> MY.NET.24.22:25	1076
80.34.74.51 -> MY.NET.87.70:4662	701	216.95.201 -> MY.NET.6.40:25	1026
148.64.48.213 -> MY.NET.235.202:3516	696	148.64.150 -> MY.NET.206.102:1995	1024
169.232.112.183 -> MY.NET.218.122:3859	534	216.95.201 -> MY.NET.24.21:25	976
200.196.36.61 -> MY.NET.218.2:4662	306	210.253.206 -> MY.NET.211.26:6011	922
148.64.19.65 -> MY.NET.150.220:1214	276	148.63.164 -> MY.NET.240.202:2570	811

Apparently there were attacks coming from many hosts in the 216.95.201.0 subnet attacking many hosts in MY.NET that would not have appeared in the top ten otherwise. Seeing that they were all on the same port it seems that they should be bundled into one attack. I next went back to a list of straight source IP addresses versus number of OOS packets sent, and used grep to form a new top ten attacking subnets:

Top Ten Talkers – OOS			
1.	66.117.30.14 = 7973 66.117.30.14 -> MY.NET.233.78:1182 = 4009 66.117.30.14 -> MY.NET.224.134:1182 = 3964	6.	193.230.240.106 = 1024 193.230.240.106 -> MY.NET.60.16:80 = 218 193.230.240.106 -> MY.NET.60.11:80 = 181 193.230.240.106 -> MY.NET.60.38:80 = 178 193.230.240.106 -> MY.NET.60.39:80 = 175 193.230.240.106 -> MY.NET.6.7:80 = 114 …
2.	216.95.201.0 = 5399 216.95.201.37 -> MY.NET.24.22:25 = 114 216.95.201.34 -> MY.NET.6.47:25 = 104 216.95.201.33 -> MY.NET.24.23:25 = 102		

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

	…
3.	209.47.197.0 = 1408 209.47.197.12 -> MY.NET.6.47:25 = 65 209.47.197.14 -> MY.NET.24.22:25 = 54 209.47.197.12 -> MY.NET.24.21:25 = 50 …
4.	148.64.150.139 -> MY.NET.206.102:1995 = 1024
5.	210.253.206.180 = 923 210.253.206.180 -> MY.NET.211.26:6011 = 922 210.253.206.180 -> MY.NET.185.48:6346 = 1
7.	148.63.164.189 = 923 148.63.164.189 -> MY.NET.240.202:2570 = 811 148.63.164.189 -> MY.NET.240.202:80 = 13
8.	213.186.35.9 = 881 213.186.35.9 -> MY.NET.97.23:81 = 24 213.186.35.9 -> MY.NET.97.23:8080 = 23 213.186.35.9 -> MY.NET.97.95:81 = 19 …
9.	148.64.48.213 = 703 148.64.48.213 -> MY.NET.235.202:3516 = 696 148.64.48.213 -> MY.NET.235.202:80 = 7
10.	80.34.74.51 -> MY.NET.87.70:4662 = 701

#1. 66.117.30.14

This host sends several thousand SYN packets to two resnet hosts over a time period of two minutes. All the packets sent are to destination port 1182. They do not appear to be crafted but are unusual in that they have ECN Echo and Congestion Window Reduce. No further interaction appears to occur but it is possible that the reactions the internal hosts had were not picked up because they did not activate any Snort rules. This could have been a SYN scan to see if a Wingate proxy was open on either host since it uses TCP 1182. This could also be a Gnutella user looking for files with the hosts failing to respond.

#2. 216.95.201.0/32, #3. 209.47.197.0/32

(These two talkers exhibit the same behavior and should be dealt with in the same manner at the same targets. For brevity I only discuss one of them here.)

These scans originate from several mail servers within dbhits.com, jsuati.com and apthits.com. A quick examination of whois, to further investigate these three domains, demonstrates that these domains share the same DNS servers (216.95.201.5 and 6) and that two of these companies sharing the same address and administrative staff (probably faked). The Whois information for these domains was found in the Tucows registration database:

```
Registrant:
1505820 Ontario Inc
610 Ford Drive Unit #1
Oakville, ON L6J 7V2
CA

Domain name: DBHITS.COM

Administrative Contact:
Administrator, Network admin@rapid-e.biz
610 Ford Drive Unit #1
Oakville, ON L6J 7V2
CA
(905) 337 8616 Fax: (905) 337 2882

Technical Contact:
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Administrator, Network admin@rapid-e.biz
610 Ford Drive Unit #1
Oakville, ON L6J 7V2
CA
(905) 337 8616 Fax: (905) 337 2882

Registration Service Provider:
easyDNS Technologies Inc., easydns@myprivacy.ca
+1.416.535.8672
http://www.easydns.com
This company may be contacted for domain login/passwords,
DNS/Nameserver changes, and general domain support questions.

Registrar of Record: TUCOWS, INC.
Record last updated on 09-May-2003.
Record expires on 09-May-2004.
Record Created on 09-May-2003.

Domain servers in listed order:
NS1.DBHITS.COM 216.95.201.5
NS2.DBHITS.COM 216.95.201.6

Registrant:
1505820 Ontario Inc
610 Ford Drive Unit #1
Oakville, ON L6J 7V2
CA

Domain name: JSUATI.COM

Administrative Contact:
Administrator, Network admin@jsuati.com
610 Ford Drive Unit #1
Oakville, ON L6J 7V2
CA
(905) 337 8616 Fax: (905) 337 2882

Technical Contact:
Administrator, Network admin@jsuati.com
610 Ford Drive Unit #1
Oakville, ON L6J 7V2
CA
(905) 337 8616 Fax: (905) 337 2882

Registrar of Record: TUCOWS, INC.
Record last updated on 29-Jan-2003.
Record expires on 27-Jan-2004.
Record Created on 27-Jan-2003.

Domain servers in listed order:
NS1.JSUATI.COM 216.95.201.5
NS2.JSUATI.COM 216.95.201.6

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

A third domain was registered here in the states using godaddy.com's registration services:

```
Registrant:
Aptimus, Inc.
95 S. Jackson St.
Suite 300
Seattle, Washington 98104
United States

Registered through: Go Daddy Software (http://www.godaddy.com)

Domain Name: APTHITS.COM
Created on: 09-Jun-03
Expires on: 09-Jun-04
Last Updated on: 09-Jun-03

Administrative Contact:
Admin, DNS  dnsadmin@aptimus.com
Aptimus, Inc.
95 S. Jackson St.
Suite 300
Seattle, Washington 98104
United States
4158962123
Fax – 4158962561
Technical Contact:
Admin, DNS  dnsadmin@aptimus.com
Aptimus, Inc.
95 S. Jackson St.
Suite 300
Seattle, Washington 98104
United States
4158962123
Fax – 4158962561
Domain servers in listed order:
NS1.APTHITS.COM
NS2.APTHITS.COM
```

Tracerouting NS1 and ns2.apthits.com reveals that they possess the same IP address as the nameservers above. These three different domains the IP addresses come from a contiguous block and appear to operate in unison. The different domains are probably in place to provide continuous operation in case one of the domains gets revoked.

Attacks from them appear to have been distributed across the hosts in an attempt to evade IDS analysis by keeping the portscan alerts from any one host very low. The scans are all seeking open SMTP ports. A typical scan looks like this: The scans went on continuously throughout the five-day period.

```
05/28-00:55:10.708757  [**] spp_portscan: PORTSCAN DETECTED from
216.95.201.37
(STEALTH) [**]
```

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
05/28-00:23:01.377856  [**] Queso fingerprint [**]
216.95.201.37:39878 -> MY.NET.24.21:25
05/28-00:23:01.413294  [**] Queso fingerprint [**]
216.95.201.37:44973 -> MY.NET.6.40:25
05/28-00:23:01.446850  [**] Queso fingerprint [**]
216.95.201.37:47767 -> MY.NET.24.23:25
05/28-00:55:15.531514  [**] spp_portscan: portscan status from
216.95.201.37: 3 connections across 3 hosts: TCP(3), UDP(0)
STEALTH [**]
05/28-00:55:21.957618  [**] spp_portscan: End of portscan from
216.95.201.37: TOTAL time(0s) hosts(3) TCP(3) UDP(0) STEALTH [**]
```

It appears that the attacker is looking within the MY.NET domain to find out what OS the hosts are running in addition to checking for open SMTP servers. At this point it seems very apparent that we are dealing with spammers.

The next question is if they compromised any machines. A check for non-portscan alerts leaving the campus these two packets appear:

```
05/30-18:18:36.892366  [**] High port 65535 tcp - possible Red Worm -
traffic [**] MY.NET.25.12:65535 -> 216.95.201.131:25
05/30-18:19:03.892033  [**] High port 65535 tcp - possible Red Worm -
traffic [**] MY.NET.25.12:65535 -> 216.95.201.131:25
```

MY.NET.25.12 is very like to be a compromised mail server. 216.95.201.131 is ui1.jsuati.com. It is quite probable that it is sending a message back to the spammers at this point saying it is ready to accept traffic from a new host. I say this because it is accepting null scans on port 0 from one host before the high port packets are sent to 216.95.201.131 and to another address for the rest of the scan.

```
05/29-13:53:26.842594  [**] Null scan! [**] 141.156.142.127:0 ->
MY.NET.25.12:0
05/29-15:31:04.081877  [**] Null scan! [**] 141.156.142.127:0 ->
MY.NET.25.12:0
05/29-15:31:05.012053  [**] Null scan! [**] 141.156.142.127:0 ->
MY.NET.25.12:0
05/29-15:31:05.105920  [**] Null scan! [**] 141.156.142.127:0 ->
MY.NET.25.12:0
05/29-15:35:13.476993  [**] Null scan! [**] 141.156.142.127:0 ->
MY.NET.25.12:1723
05/29-15:35:14.585737  [**] Null scan! [**] 141.156.142.127:0 ->
MY.NET.25.12:0
05/29-15:26:03.622699  [**] Null scan! [**] 141.156.142.127:9 ->
MY.NET.25.12:59678
05/29-15:26:03.713023  [**] Null scan! [**] 141.156.142.127:9 ->
MY.NET.25.12:59678
05/30-18:18:36.892366  [**] High port 65535 tcp - possible Red Worm -
traffic [**] MY.NET.25.12:65535 -> 216.95.201.131:25
05/30-18:19:03.892033  [**] High port 65535 tcp - possible Red Worm -
traffic [**] MY.NET.25.12:65535 -> 216.95.201.131:25
05/31-20:51:28.164716  [**] Null scan! [**] 151.196.38.234:0 ->
MY.NET.25.12:0...
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Neither Adore (If the MY.NET mail server is Linux) nor RC1 (If the MY.NET Mail server runs Windows) sends packets to port 25 from port 65535. This is probably an Adore variant, but it is difficult to tell more without the full packet information.

#4. 148.64.150.139, #7. 148.63.164.189, #9. 148.64.48.213
These appear to be Kazaa file sharing sessions between external users (both on Starband) and internal hosts in Resnet. It is Out-of-spec because there are no acknowledgement numbers when the PSH packets are sent.

#5. 210.253.206.180
This host was either sending out a SYN portscan much like the one sent from 66.117.30.14. However the destination port is 6001 TCP that would test whether MY.NET.211.26 is running X-windows Service. All in all it is likely this is another attempt at setting up a Gnutella session. There was one packet sent to MY.NET.185.45 that was of the same type except that it was sent to Gnutella's default port.

#6. 193.230.240.106
This host appears to be sending out a SYN scan with ECN flags set. This looks much like the first out-of-spec scan (66.117.30.14) and the scan before this one. The difference being that this scan is seeking Gnutella peers on port 80.

#8. 213.186.35.9
The SYN packets from this attacker are obviously crafted. The sequence number remains the same even as the port rises. This attacker is probing ports 23, 80, 81,1080, 3128, 6588, 8000, 8001, 8080, 8081 and 8888. Several internal hosts were tested by this scan all of them in the .97 or dialup network. It is quite possible that a worm is in operation on the attacking host.

#10. 80.34.74.51
This appears to be an eDonkey 2000 session between an internal host in the chemistry department and a host in Spain.

3.6 External Source Addresses Selected for Further Investigation

Besides the three already investigated above. The following two addresses come to mind as requiring further investigation:
67.2.167.3 – This IP address scanned a range of Internal Hosts for SubSeven. It turns out that it is a dialup address associated with Qwest.

```
0-1pool167-3.nas26.portland1.or.us.da.qwest.net
OrgName:    Qwest Communications
OrgID:      QWD3
Address:    950 17th Street
Address:    Suite 1900
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
City:       Denver
StateProv:  CO
PostalCode: 80202
Country:    US

NetRange:   67.0.0.0 - 67.7.255.255
CIDR:       67.0.0.0/13
NetName:    QWEST-BLK-5
NetHandle:  NET-67-0-0-0-1
Parent:     NET-67-0-0-0-0
NetType:    Direct Allocation
NameServer: DCA-ANS-01.INET.QWEST.NET
NameServer: SVL-ANS-01.INET.QWEST.NET
Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate:    2001-07-26
Updated:    2002-11-26

TechHandle: QN-ARIN
TechName:   NOC, NOC
TechPhone:  +1-703-363-3001
TechEmail:  support@qwestip.net

OrgAbuseHandle: QIA2-ARIN
OrgAbuseName:   Qwest IP Abuse
OrgAbusePhone:  +1-703-363-3001
OrgAbuseEmail:  abuse@qwest.net

OrgNOCHandle: QIN-ARIN
OrgNOCName:   Qwest IP NOC
OrgNOCPhone:  +1-703-363-3001
OrgNOCEmail:  support@qwestip.net

OrgTechHandle: QIA-ARIN
OrgTechName:   Qwest IP Admin
OrgTechPhone:  +1-888-795-0420
OrgTechEmail:  ipadmin@qwest.com
```

213.186.35.9 – This address appears to have been infected with the RingZero worm. Apparently it is a server at a server hosting facility in France.

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

inetnum:      213.186.35.0 - 213.186.35.255
netname:      OVH
descr:        Dedicated Hosting
descr:        http://www.ovh.com
country:      FR
admin-c:      OK217-RIPE
tech-c:       OK217-RIPE
status:       ASSIGNED PA
mnt-by:       OVH-MNT
notify:       noc@ovh.net
```

© SANS Institute 2004, Author retains full rights.

```
changed:      noc@ovh.net 20000126
source:       RIPE

route:        213.186.32.0/19
descr:        OVH ISP
descr:        Paris, France
origin:       AS16276
notify:       noc@ovh.net
mnt-by:       OVH-MNT
changed:      noc@ovh.net 20010217
source:       RIPE

person:       Octave Klaba
address:      SARL OVH
address:      140, quai du sartel
address:      59100 Roubaix
address:      France
phone:        +33 3 20 20 09 57
fax-no:       +33 3 20 20 09 58
nic-hdl:      OK217-RIPE
remarks:      ========================================
remarks:      support : support@ovh.com
remarks:      0 899 701 761 (france only)
remarks:      ========================================
remarks:      troubles:
remarks:      + network : abuse@ovh.net
remarks:      + spam    : http://www.spam-rbl.com
remarks:      ========================================
remarks:      peering : noc@ovh.net
remarks:      prefix 213.186.32.0/19
remarks:      - FreeIX (1Gbs) 213.228.3.244
remarks:      ========================================
e-mail:       noc@ovh.net
mnt-by:       OVH-MNT
changed:      noc@ovh.net 20021204
source:       RIPE
```

3.7 Internal Host Compromise or Anomalies

IP Address	Priority	Rationale	Recommended Procedure
MY.NET.88.163	High	Running XDCC*	Quarantine System, Test for Trojan, Educate User
MY.NET.80.209	High	Running XDCC*	Quarantine System, Test for Trojan, Educate User
MY.NET.25.12	High	Probable compromise	Quarantine Machine, Review Outbound Mail Traffic Flow, Install latest patches to sendmail.
MY.NET.97.68	High	Nimda	Quarantine System, Remove Trojan, Educate User
MY.NET.97.53	High	Nimda	Quarantine System, Remove Trojan, Educate User
MY.NET.97.46	High	Nimda	Quarantine System, Remove Trojan, Educate User
MY.NET.12.4	High	Unknown Scans from mail server	Review Packet Capture
MY.NET.5.43	Medium	Running XDCC	Quarantine System, Test for Trojan, Educate User
MY.NET.219.90	Medium	Running XDCC	Quarantine System, Test for Trojan, Educate User

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

MY.NET.219.14	Medium	Running XDCC	Quarantine System, Test for Trojan, Educate User
MY.NET.218.38	Medium	Running XDCC	Quarantine System, Test for Trojan, Educate User
MY.NET.218.206	Medium	Running XDCC	Quarantine System, Test for Trojan, Educate User
MY.NET.132.27	Medium	Running XDCC	Quarantine System, Test for Trojan, Educate User
MY.NET.132.24	Medium	Running XDCC	Quarantine System, Test for Trojan, Educate User
MY.NET.97.188	Low	Probable IRC Trojan	Place in Watch List
MY.NET.91.151	Low	XDCC Attempts	Place in Watch List
MY.NET.83.173	Low	XDCC Attempts	Place in Watch List
MY.NET.83.100	Low	XDCC Attempts	Place in Watch List
MY.NET.80.149	Low	XDCC Attempts	Place in Watch List
MY.NET.233.78	Low	Unexplained scans to 1182	Review Packet Capture, Block 1182 in Resnet
MY.NET.224.134	Low	Unexplained scans to 1182	Review Packet Capture, Block 1182 in Resnet
MY.NET.198.221	Low	XDCC Attempts	Place in Watch List
MY.NET.190.95	Low	Probable IRC Trojan	Place in Watch List
MY.NET.140.136	Low	XDCC Attempts	Place in Watch List

Probable Compromise
MY.NET.25.12 should be withdrawn from service and examined closely.

Running XDCC
The following hosts ran XDCC MY.NET.88.163, MY.NET.83.100, MY.NET.198.221, MY.NET.91.151, MY.NET.80.209, MY.NET.83.173, MY.NET.80.149, and MY.NET.140.136.

If a no copyrighted File Sharing policy is to be enacted these systems should be quarantined from the network and tested for Trojans.

IRC kill
A number of IRC kills this large almost has to involve an IRC zombie. The traffic of these three boxes should be analyzed to determine if it they indeed are Trojans. Telltale signs would include attempting denial of service attacks upon external hosts,

MY.NET.190.95, 17631; MY.NET.88.163, 3573; MY.NET.97.188, 214

File Sharing
File Sharing can open the University up to liability concerns regarding the pirating of software, music and videos through the Internet. Furthermore such downloads are bandwidth intensive and as such detrimental to network performance.

The following Resnet hosts have been involved with file sharing in various forms. The owners of these hosts should be reminded that file sharing and hosting file-sharing services could be illegal depending on the jurisdiction
.
MY.NET.217.146, MY.NET.218.230, MY.NET.217.158, MY.NET.206.102

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

The following hosts are pooled in various departments around the school. Whoever installed file sharing on these systems violated policy by installing an unauthorized service on a university managed computer and utilizing the university networks to transfer files. The hosts should be purged of the file sharing software.

MY.NET.83.69, MY.NET.86.110, MY.NET.84.178

MY.NET.83.69 is also running a TFTP service and should be investigated for a potential IRC trojan.

3.8 Defensive Recommendations

University Computer and Network Use Policy review

University Computer and Network User Policy should be reviewed and amended. Recent legal cases have made it apparent that academic institutions must take a more proactive role in stemming online piracy. Failure to do so could involve legal action from groups representing intellectual property holders, most notably the RIAA.[25]

This being the case language should be added to the policy stating that the illegal dissemination or collection of copyrighted material over the university network will not be tolerated. Specific applications should be listed in the policy as not being tolerated on the network such as Kazaa, Gnutella, and WinMX.

This policy should be implemented along with the firewall policy listed to below in order to make it more difficult for university hosts to share files over such networks. Despite the cost in managing this policy it should be enforced. It is far cheaper to have an administrator spend several hours a month sending warnings and cutting off access to violators than it would be to deal with potential legal issues with the RIAA.

User Education

User education is critical in any effort to maintain security. Users should be advised to stay up to date on the latest patches, close open shares on their computers, run anti-virus software and not download files from unknown sources. Perhaps a primer on Computer Security could be appended to a mandatory orientation course?

Firewall

My first recommendation is to block as many external out of spec scans of the network as possible by installing stateful firewalls on the perimeter of the network and placing ACLs that block scanning traffic. This will prevent hackers from gaining intelligence about the network, and more importantly lower the volume of

[25] Maki, Justine. “University cracks down on file sharing.” 21 Apr 2003. URL:http://www.collegian.psu.edu/archive/2003/04/04-21-03tdc/04-21-03dnews-04.asp. (28 Jun 2003).

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

non-critical alerts pouring into the network. I would also suggest running LaBrea over the as yet unallocated addresses in the UMBC domain and allowing free access to it.

Beyond that the firewall can also be used to block spoofed non-IANA assigned addresses, DShield's top ten list of scanners, and outbound sessions on IRC and inbound and outbound sessions on file sharing services.

Spam
It is quite probable that the campus has a problem with outgoing spam. The more internal hosts that are restricted from using port 25 outside of the university network the easier it will be to isolate compromised systems. The downside is that university hosts will be unable to send mail directly from external sites (like a student sending mail from a home account with an ISP).

After this restricted host traffic should be checked for attempts to send spam. A survey should be put together to discover what mechanism is being used to send the spam.

As this problem is addressed hosts should be monitored for unusual spikes in e-mail activity. Correlating data with an independent third party like senderbase.com could provide final verification that a problem has been solved.

As always university mail servers should be updated to the latest version of sendmail and the latest operating systems with all patches in place. Patches should be applied vigorously and preferably automatically. Other university systems should be checked to make sure they do not inadvertently have mail servers in operation.

Router Audit
One of both of the perimeter routers has an external routing protocol set in such a way that at times the university becomes the favored route between two foreign hosts. Unless the university is receiving compensation for this load this should not happen. An audit of the external routing protocols of the routers should be performed to correct this issue.

3.9 Methodology

In order to analyze the data I first concatenated each five day group of files together into a single file for each category: Alerts, Scans and OOS. Time as a dimension can be removed from these problems until the latter part of the analysis and less files makes for knowing where to seek each category of file.

Once that was complete I used perl scripts and grep techniques found in earlier practicals (http://www.giac.org/practical/Chris_Baker_GCIA.zip) and (http://www.giac.org/practical/chris_kuethe_gcia.html#3.0) to assist me with analyzing the data. Not getting all I wanted from these scripts, I wrote a couple

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

of my own. One that grabbed the hostname attached to addresses in batch (I could send it a list of addresses and it would generate hostnames to the addresses). I also wrote one to break down a set of detects by port.

I chose to first analyze the Out-of-specs. Besides being the smallest set of detects, the OOS was more likely to possess valuable traffic since many tools used by file-sharers and hackers tend to deliver out-of-spec packets.

Once the OOS analysis was complete, including the top ten talkers for OOS, I began work on the Scans. With the Scans, I actually broke down and used Access to perform most of the queries, though port analysis was still performed in perl.

The alerts table proved to be the most difficult to import into a database. The different types of alerts dumped into one file made direct import infeasible. So, I ended up breaking it down again using a perl script and the homogenized records into Access to assess a top ten talkers for Alerts.

Once that information was put in place along with the general list of alerts and scans I could begin correlating events together to search for unusual, anomalous, or detrimental activity. Without raw logs to verify the correlations the analysis pretty much ends at that point.

References

1. Mimoso, Michael. “Gartner Declares IDS Obsolete by 2005.” http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci905961,00.html. June 12, 2003. (June 19, 2003)

2. Roesch, Martin “IDS dead? ... Not So Fast!” http://napps.nwfusion.com/weblogs/security/002959.html#002959. June 16, 2003. (June 19, 2003).

3. Moore, Geoffrey. Inside the Tornado : Marketing Strategies from Silicon Valley's Cutting Edge. 1999, Harper Business, New York. p. 4.

4. Proctor, Paul “Intrusion Detection and Homeland Security - Ask the Expert – CIO”, http://www2.cio.com/ask%5Cexpert/2002/questions/question1522.html. July 15, 2002 (June 19, 2003).

5. IETF. “Intrusion Detection Exchange Format (idwg) Charter.“ http://www.ietf.org/html.charters/idwg-charter.html. April 10, 2003 (June 19, 2003).

6. Bandy, P., Money, M. & Worstell, K. “Should communication between the sensor (or agent) and the monitor be encrypted?”, http://www.sans.org/resources/idfaq/communication.php. 1999. (June 19, 2003)

7. Cisco. “Cisco Secure Intrusion Detection System Sensor Configuration Note Version 3.0.” http://www.cisco.com/univercd/cc/td/doc/product/iaabu/csids/csids6/12216_02.htm#xtocid16. Oct. 17, 2002. (June 19, 2003).

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

8. Fyodor, “Nmap network security scanner man page.” 2003. URL:http://www.insecure.org/nmap/data/nmap_manpage.html. 25 Jun 2003.

9. Ibid.

10. Ibid.

11. Vecna, “Usual Iploggers Miss Some Variable Stealth Scans.” 17 Jan 2000. URL: http://www.securityfocus.com/archive/1/42136. 25 June 2003.

12. Novell. “Things You Should Know During the Installation”. 2003. URL:http://www.novell.com/documentation/lg/ifolder20/ifolder/data/ah00j8c.html. 2003 Jun 25.

13. Sevcenco, Serghei. “Symantec Security Response - Backdoor.Sdbot.” 2003 Jun 16. URL: http://securityresponse.symantec.com/avcenter/venc/data/backdoor.sdbot.html. 2003 Jun 25.

14. Chien, Eric, “Symantec Security Response - W32.Nimda.A@mm.” 15 Jan 2003. URL:http://www.sarc.com/avcenter/venc/data/w32.nimda.a@mm.html. 25 Jun 2003

15. Leweling, Martin. “Re: [suse-security] Sy[s]tem Attack or ?” 27 Mar 2001. URL: http://lists.suse.com/archive/suse-security/2001-Mar/0371.html. 25 Jun 2003.

16. Nathan, Jeff, “Re: [Snort-users] SHELLCODE x86 NOOP.” 7 Mar 2002. URL:http://archives.neohapsis.com/archives/snort/2002-03/0159.html. 25 Jun 2003.

17. Roesch, Martin, Caswell, Brian. et al., “[Snort-users] Snort FAQ.” 29 Jun 2002. URL:http://www.geocrawler.com/archives/3/4890/2002/6/0/9059445/. 25 Jun 2002.

18. T3, “The Science of OS Fingerprinting.” 2001. URL:http://toorcon.org/2001/lineup/osfinger.ppt. 25 Jun 2003.

19. Chezz, “SubSeven Frequently Asked Questions.” 10 Oct 2002. URL: http://520038635832-0001.bei.t-online.de/s7/sub/help.shtml. 25 Jun 2003.

20. Chien, Eric, “Symantec Security Response - W32.Nimda.A@mm.” 15 Jan 2003. URL:http://www.sarc.com/avcenter/venc/data/w32.nimda.a@mm.html. 25 Jun 2003

21. Lieberman, Jeff, “Re: Network printer trouble” 24 Jul 2000. URL:http://www.aplawrence.com/Bofcusm/434.html. 25 Jul 2003.

22. Scarborough, M. “Re: TCP Dest Port 1214.” 1 Jun 2001. URL: http://www.incidents.org/archives/intrusions/msg03822.html. 25 Jun 2003.

23. Buchanan, J. “Working Around ISP Port Blocks.“ 2003 URL:” http://homepage.ntlworld.com/j.buchanan/index.html?blocked.html,” 25 Jun 2003

24. MetaMachine, “What Ports does eDonkey use?” 2002. URL:http://www.edonkey2000.com/cgi-bin/smartfaq/smartfaq.cgi?answer=1025114514&id=1025114052. 25 June 2003.

25. Maki, Justine. “University cracks down on file sharing.” 21 Apr 2003. URL:http://www.collegian.psu.edu/archive/2003/04/04-21-03tdc/04-21-03dnews-04.asp. (28 Jun 2003).

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.