Global Information Assurance Certification Paper

GCIA Version 3.4, Revised 1/12/04
Submitted January 12, 2004
Author Kevin D. Knox
Title: Hacking in Deep Rough

Table of Contents

Part 1 - Describe the state of intrusion detection - "Network IDS in a Switched Environment"

Abstract

VLANs are a necessary component of our network infrastructure in today's Gigabit/s and 100Mbit/s network environments. With today's 12/24/36/72 port switches it becomes prudent and economic to allocate ports efficiently and effectively such that switches are used to their maximum potential and costs are reduced. Allocating VLANs provides Defense in Depth, DiD, by isolating networks based on function and application. Access controls are then used at the border firewall or router to grant or deny privileges to the appropriate users, groups and services. This component of the GCIA practical will discuss the merits and drawbacks of using TAPS, hubs and port mirroring to implement Network IDS in a legacy pre-DiD environment.

Introduction

One of the interesting things that occurs during the security phase of an already existing installation, is the rush to install NIDS and sniffers in an environment where no capability existed. The switching devices of the future will undoubtedly have this aspect under control. But existing technology forces us to use that which is in place today. This includes layer 1, 2, 3 and 4 switches, hubs, TAPS and other devices. The concept of DiD came to the forefront over the last several years with those on the leading edge in technology providing the necessary leadership. According to RealSecure's Advanced RealSecure Student Guide Chapter 4 several solutions exist for monitoring networks and systems which were previously unmonitored. These include hubs, TAPs, TAP with dual NIDs and TAPs consolidated through a switch.

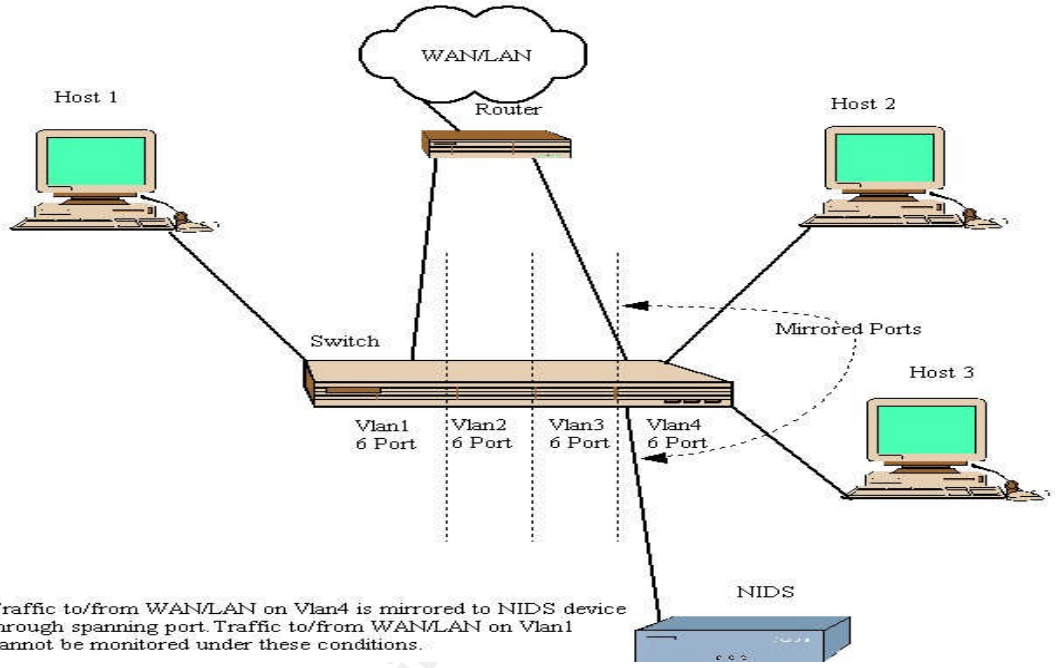
DiD and VLANs

Stephen Northcutt's Network Intrusion Detection An Analyst's Handbook, Pg 246 addresses the issue of Hardware-Based ID. He discusses the limitations of inspecting and capturing traffic at wire speed. He elaborates on the need to implement hardware ID at the switch and/or router as a necessary component of our DiD strategy. By being able to implement ACLs and traffic filters at the switch or router, the model for DiD becomes stronger in that most if not all network traffic will be inspected.

In dealing with network systems lacking provisions for DiD, i.e. NIDS and sniffers, one becomes aware of inherent problems. Where to attach the new NID or sniffer device? Typically the network administrator will partition a switch based on need. This means if the admin has a 4 node network, then the admin will allocate perhaps 4 to 6 ports for each VLAN; one port for the uplink to the router or firewall and the remaining ports for hosts or servers. If he does this for all 24 ports, then he will have room for 4 – 6 port VLANs as shown in Drawing #1. As time passes and security threats are realized and addressed by management, the need for NIDS and sniffer devices comes to the

forefront. Drawing #1 shows that Vlan4 consists of 2 hosts, a NID and an uplink to a router. The port to the router has been mirrored to the NID. Traffic between Host 2 and 3 is not seen by the NID. Traffic from either Host 2 or 3 to the WAN/LAN is seen by the router and the NID. This allows a measure of protection for Vlan4, in that the segment is now protected from external entity's traffic ingressing and egressing the segment by a NID.

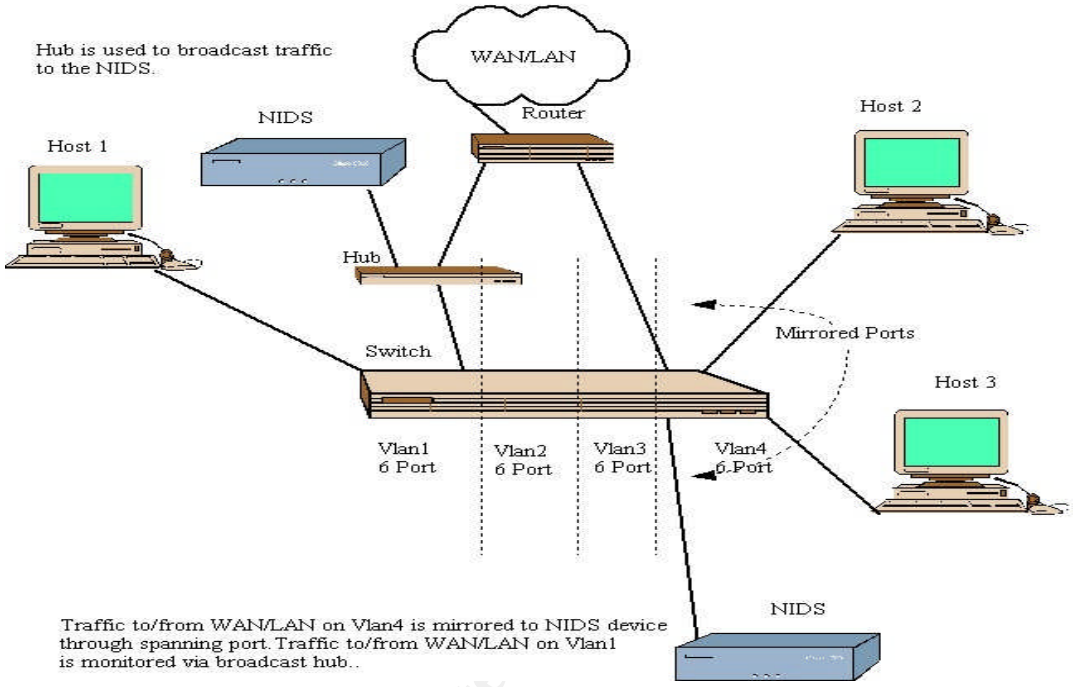
Drawing # 1



TAPs and Hubs

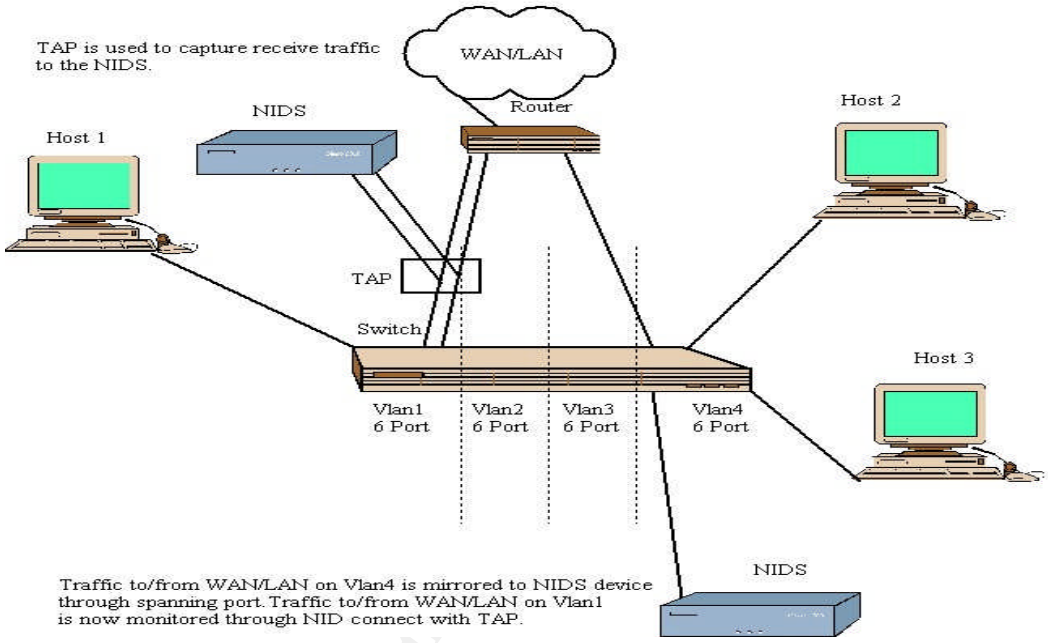
Typically, switches come with the capability to mirror 1 port at a time. With 4 separate VLANs on a switch, the need arises to attach NIDS to each network segment to provide DiD. We can certainly mirror one port to another, but what becomes of the other network segments when the desire is to have continuous 100% packet inspection? This is where hubs and TAPs come into play and the limitations of each are exposed. Drawing # 1 shows a switch with 4 defined VLANs and only Vlan4 monitored. The problem here is that the other 3 vlans are left unprotected due spanning port limitations. To alleviate this we can place hubs on the other network segments and attach NID devices to those hubs for traffic inspection. See Drawing # 2. Traffic between the WAN/LAN in Drawing #2 and Host 1 is inspected by the NID attached to the hub. Because a hub will broadcast traffic to all ports, collisions can occur. According to the "*Architecture Issues"* component of our Beginning Analysis, Intrusion Detection in Depth course material "A hub is an inexpensive low-end solution for half-duplex traffic". This solution is inappropriate for a network with other than low bandwidth and throughput.

Drawing # 2



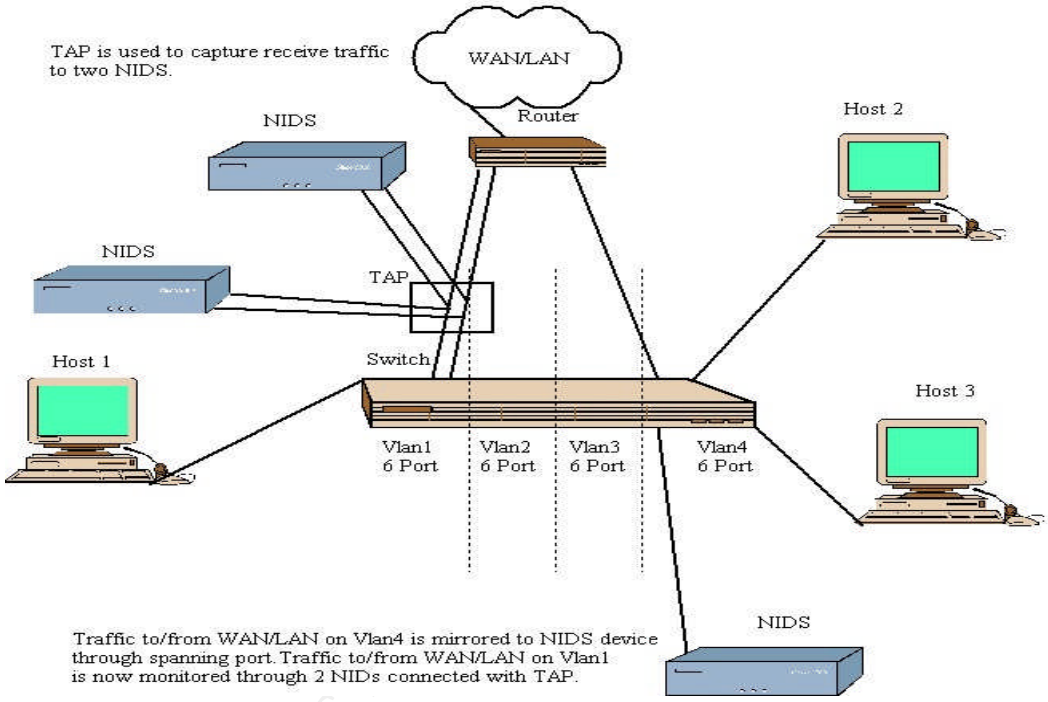
This is where the network TAP or Test Access Port becomes instrumental. As seen in Drawing # 3, the TAP is a hardware or layer 1 device introduced to the ethernet such that traffic can be inspected by a resident NID. The TAP solution allows security admins to obtain traffic without an impact on throughput. The TAP also does not introduce a single point of failure. The problem, as documented in *Network Intrusion Detection Systems Important IDS Network Security Vulnerabilities* located at URL: http//www.toplayer.com/pdf/WhitePapers/wp_network_intrusion_system.pdf, is that the transmit and receive cable pairs are split and the NID will only see half the traffic if directly connected. What this implies is that any stateful inspection for various signatures involving syn, syn/ack, ack will cause a myriad of false positives. Signatures such as Synflood will be triggered continuously. If the triggering parameters are tuned high enough to eliminate false positives, then these signatures will be completely ignored thus leaving the network exposed to such exploits.

Drawing # 3



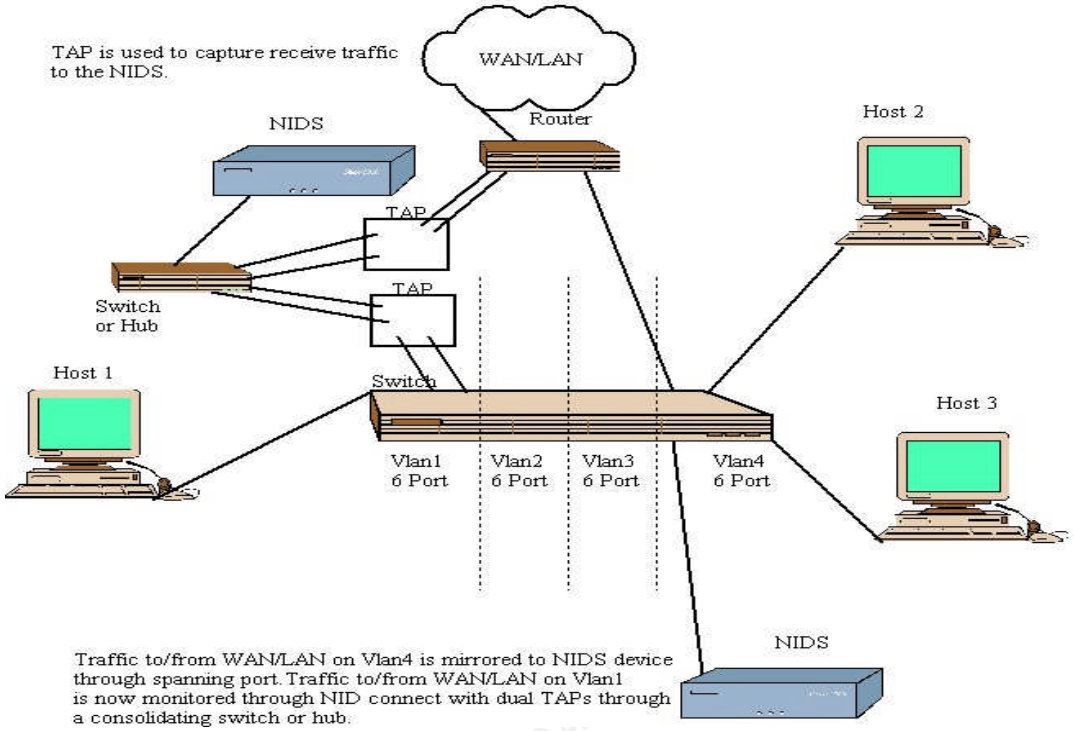
Drawing # 4 shows the case where a single TAP and 2 NID devices are used to inspect traffic. One NID will inspect traffic coming into the segment and the other NID will capture traffic leaving the segment. Drawbacks to this solution are that the hardware becomes expensive and that only receive data is inspected as was the previous configuration. According to ISS in their How To Guide Intrusion Detection Systems located at URL:http//www.snort.org/docs/iss-placement.pdf "Without extra modifications the solution cannot monitor traffic in both directions.

Drawing # 4



Drawing # 5 shows the case where TAPs are consolidated through a switch or hub. This solution will only require 1 NID device thus reducing cost. In the case of the RealSecure solution, the NID becomes fully functional.

Drawing # 5



Loading and Oversubscription

Although we can place a firewall at the border, internal threats and vulnerabilities still exist. DiD stipulates layers of defense for our networks and this implies network as well as host based IDS coupled with AV protection.

To accomplish the concept of DiD, each network segment must be protected by a NID. According to *Network Intrusion Detection Systems Important IDS Network Security Vulnerabilities* located at URL: http//www.toplayer.com/pdf/WhitePapers/wp_network_intrusion_system.pdf, the 100 Mbit/s NID can actually only detect 60-80 Mbit/s. With multiple VLANs on a single switch connected to different routers the switch becomes congested and so will the NID. For example, if the uplinks to the firewall or router is on the 100Mbit/s port, and all ports are 80% loaded, then you will see oversubscription by the switch and the likelihood that packets will be dropped. This document also elaborates on the fact that the 1 Gbit/s NID will only see 400-600 Mbit/s. In the case where multiple VLANs are allocated on a single switch and the uplink to the router or firewall is on the Gigabit port, the switch and NID will become oversubscribed when loading approaches 40 to 60 percent.

Layer 4 and ASICs

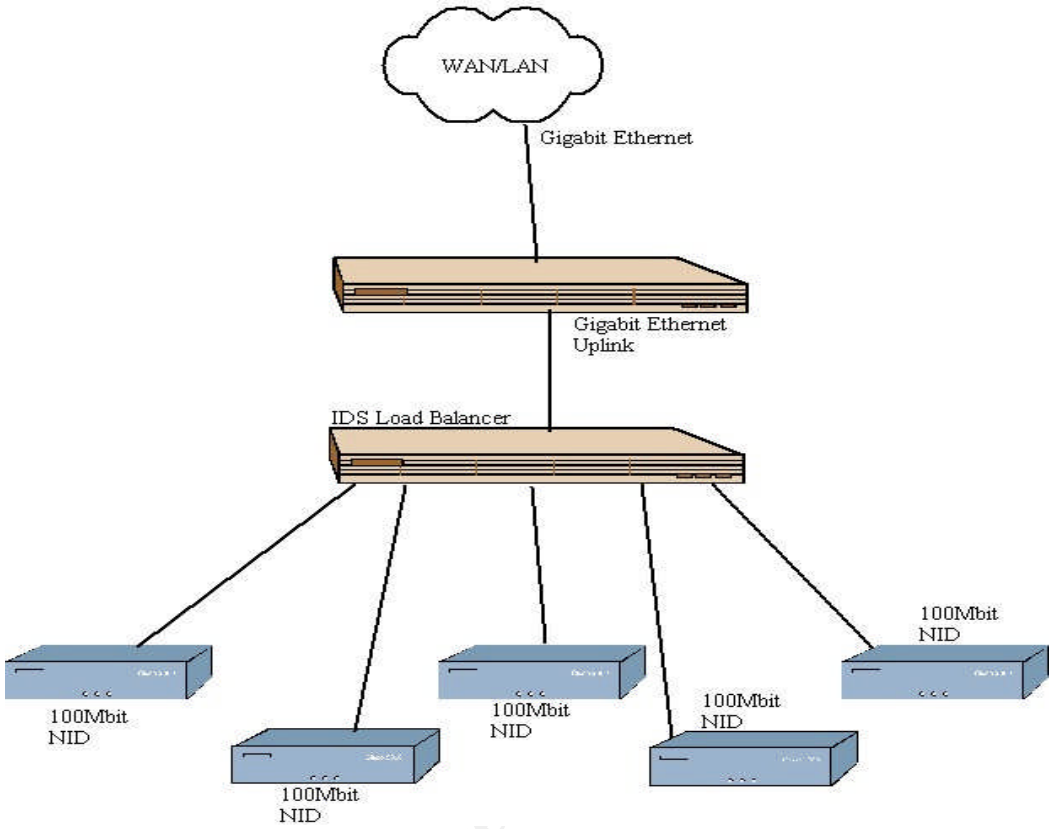
Where does all of this information take us? According to the *Layer 4 Switching White Paper* located at

URL:http//www.telco.com/products/IPswitching/MultiLayer/t5pro/?f=/wp/layer_4_switches_010402.pdf, application switching or service differentiation is the future. The paper elaborates on the past ability of software based routers to differentiate between applications based on software algorithims. The layer 4 solution now resides in the ability of the hardware to inspect transport layer header information to allow decision making at wire speed. The paper *Multi-layer switching in the enterprise* located at URL:http//www.networkmagazineindia.com/200301/cover13.shtml, states that "Legacy routers use software running on microprocessors to forward packets. Switching routers, on the other hand, use hardware, namely, Application Specific Integrated Circuits(ASICs)". This capability results in a 100-fold improvement in performance. The document also states that the layer 4 solution eliminates the performance loss associated with implementing security features. Access to information can be controlled by the user's application instead of blocking all users and allowing only those specified access.
The ASIC solution fits well in the DiD strategy in that multiple layers of protection are implemented. Not only is the router doing access control and application filtering, the NID is able to continue it's function in another layer of defense.

NID Load Balancing
Drawing #6 shows the case for an IDS load balancer. In this configuration, the traffic is treated as flow based rather than packet based. The TopLayer balancer allows network traffic to be balanced across a suite of NIDs. Each chunk of traffic is recognized based on flow rather than packet based. This is likened to an entire conversation versus a word in a sentence between individuals.

As load increases on the network, the IDS goal is to have 100% attack recognition. With 5 – 100Mbit network sensors distributed across the balancer, traffic to and from the 1 Gbit uplink will be inspected up to approximately 300 – 400 Mbit/s. If traffic increases above this threshold then more NIDs can be allocated to provide the 100% coverage necessary.

Drawing # 6



Conclusion

Just what is the bottom line when it comes to outfitting a legacy network infrastructure with NID and DiD models in mind? Many options exist. The future of implementing hardware based ID in conjunction with layer 4 service oriented smart devices and load balancing NIDs is at our doorstep. The security goal of 100% traffic inspection and detection is well on it's way to becoming a reality. The capability of these new techniques to achieve inspection and detection at wire speed is becoming reality. As our networks age and devices become obsolete, they will be replaced. When this occurs, DiD and ID should be addressed in the design and implementations phase of the network replacement.

References

Northcutt,Stephen. Network Intrusion Detection, An Analysts Handbook. Indianapolis: New Riders Publishing, Jun. 99. 246

Stevens, W. Richard. TCP/IP Illustrated,Volume 1. Reading: Addison-Wesley, 1994. 56,246

ISS. Advanced RealSecure. Student Guide. Atlanta: Internet Security Systems, 1998. 59-82.

"*Architecture Issues"* component of our Beginning Analysis Course of Intrusion Detection in Depth. 2003.

Edwards, Simon. "Network Intrusion Detection Systems: Important IDS Network Security Vulnerabilities". Top Layer Networks. September 2002.
URL:http://www.toplayer.com/pdf/WhitePapers/wp_network_intrusion_system.pdf

Laing, Brian. "How To Guide: Intrusion Detection Systems". Internet Security Systems. 2000. Sovereign House.
URL:http://www.snort.org/docs/iss-placement.pdf

"Layer 4 Switching White Paper". Revision 1. January 4, 2002. Telco Systems, A BATM Company.
URL:http://www.telco.com/products/IPswitching/MultiLayer/t5pro/?f=/wp/layer_4_switches_010402.pdf

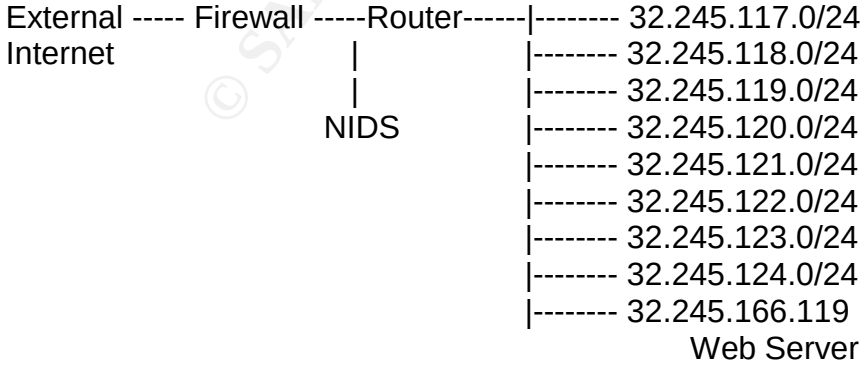
Birje, Uday. "Multi-layer switching in the enterprise". Network Magazine India.
URL:http://www.networkmagazineindia.com/200301/cover13.shtml,

Part 2---Network Detects

Network Detect #1

Source of Trace
The detect was sourced from URL:http//www.incidents.org/logs/Raw/2002.9.19.

Based on the results of a snort generated analysis of the raw tcpdump log file, it appears that the network IDS is located outside the 32.245.0.0/16 network segments. From my viewpoint and the volume of alerts generated, the network appears to look like this



Over 7400 alerts were directed at the 32.245.0.0/16 network segments.

Detect was generated by
The following events of interest were generated by Snort Version 2.0.3 executed against the 2002.9.19 log file.

[**] SCAN Squid Proxy attempt [**]
10/19-022900.976507 24.190.48.2351239 -> 32.245.117.1183128
TCP TTL120 TOS0x0 ID3032 IpLen20 DgmLen44 DF
******S* Seq 0x591D0730 Ack 0x0 Win 0x2000 TcpLen 24
TCP Options (1) => MSS 1460

[**] SCAN Proxy (8080) attempt [**]
10/19-022902.706507 24.190.48.2351271 -> 32.245.117.1238080
TCP TTL120 TOS0x0 ID63960 IpLen20 DgmLen44 DF
******S* Seq 0x591D0898 Ack 0x0 Win 0x2000 TcpLen 24
TCP Options (1) => MSS 1460

[**] SCAN Squid Proxy attempt [**]
10/19-023852.786507 24.190.48.2353837 -> 32.245.124.673128
TCP TTL120 TOS0x0 ID36486 IpLen20 DgmLen44 DF
******S* Seq 0x59A1AE30 Ack 0x0 Win 0x2000 TcpLen 24
TCP Options (1) => MSS 1460

[**] SCAN Proxy (8080) attempt [**]
10/19-023852.816507 24.190.48.2353836 -> 32.245.124.678080
TCP TTL120 TOS0x0 ID37510 IpLen20 DgmLen44 DF
******S* Seq 0x59A1AE27 Ack 0x0 Win 0x2000 TcpLen 24
TCP Options (1) => MSS 1460

The snort rules which generated these alerts were

alert tcp $EXTERNAL_NET any -> $HOME_NET 3128 (msg"SCAN Squid Proxy attempt"; flagsS,12; classtypeattempted-recon; sid618; rev4;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 8080 (msg"SCAN Proxy \(8080\) attempt"; flagsS,12; classtypeattempted-recon; sid620; rev3;)

These rules basically state that any external source host and port bound for the internal protected network on port 3128 or port 8080 constitutes a reconnaissance attempt by the source host. The snort id, sid, for 618 gives the following detailed information

“This event indicates that an attempt has been made to scan a host.
This may be the prelude to an attack. Scanners are used to ascertain which ports a host may be listening on, whether or not the ports are filtered by a firewall and if the host is vulnerable to a particular exploit.”

The sid for 620 gives the following description

“This event indicates that an attempt has been made to scan a host.
This may be the prelude to an attack. Scanners are used to ascertain which ports a host may be listening on, whether or not the ports are filtered by a firewall and if the host is vulnerable to a particular exploit.”

The above information was obtained at URL:http//www.snort.org/cgi-bin/sigs-search.cgi?sid=sid+.

The rule header consists of this snipet “alert tcp $EXTERNAL_NET any -> $HOME_NET 3128”.
As stated above, any external source or port directed at an internal host on ports 3128 and/or 8080 will trigger these rules and alerts will be generated.
The rule options are “(msg"SCAN Squid Proxy attempt"; flagsS,12; classtypeattempted-recon; sid618; rev4;)”.
The alert message keyword, msg, will be "SCAN Squid Proxy attempt". The semicolon represents a new rule option.
In this case, “flagsS,12”, represent the keyword “flags” with a value of S,12. This represents a tcp packet where the Syn flag is set and the reserved bits are masked and their value is of no concern.
The rule option classtype signifies a priority 2 classification of “attempted-recon” as specified in the classification.config file of snort.

The snort command used to generate the above alerts is as follows

snort -d -h 32.245.0.0/16 -l c\snort203\bin\log2 -c c\snort203\bin\snort.conf -r 2002.9.19 -k none

-d Dump the application layer
-h Set the Home reference network
-l Log to directory
-c Use the snort.conf file
-r Read from the binary input file
-k none No checksum mode

This scan activity continued for almost 10 minutes against the following network segments

32.245.117.0/24
32.245.118.0/24
32.245.119.0/24
32.245.120.0/24
32.245.121.0/24
32.245.122.0/24
32.245.123.0/24
32.245.124.0/24

The following windump command was used to verify the time frame

windump -r 2002.9.19 -n "src host 24.190.48.235 and dst net 32.245.0.0/16" and "dst port 8080 or dst port 3128"

-r	Read from binary input file
-n	Do not resolve addresses to names
src host	Host of origin
dst net	Network of targets
dst port	Ports of interest

The result of the windump command is as follows

Beginning Scan
022900.976507 IP 24.190.48.235.1239 > 32.245.117.118.3128 S
14950746081495074608(0) win 8192 <mss
1460> (DF)
022902.706507 IP 24.190.48.235.1271 > 32.245.117.123.8080 S
14950749681495074968(0) win 8192 <mss
1460> (DF)

(snipped)

Same pattern from 32.245.117.0 to 32.245.117.253.
Same pattern from 32.245.118.0 to 32.245.118.253.
Same pattern from 32.245.119.0 to 32.245.119.253.
Same pattern from 32.245.120.0 to 32.245.120.253.
Same pattern from 32.245.121.0 to 32.245.121.253.
Same pattern from 32.245.122.0 to 32.245.122.253.
Same pattern from 32.245.123.0 to 32.245.123.253.
Same pattern from 32.245.124.0 to 32.245.124.253.

Ending Scans
023852.786507 IP 24.190.48.235.3837 > 32.245.124.67.3128 S
150376811215037681l2(0) win 8192 <mss
1460> (DF)

023852.816507 IP 24.190.48.235.3836 > 32.245.124.67.8080 S
15037681031503768103(0) win 8192 <mss
1460> (DF)

According to advisory 99-024 of the NIPC located at
URL:http//www.nipc.gov/warnings/advisories/1999/99-024.htm
this scan pattern has the earmarks of a host infected with the RingZero Trojan. The host at 24.190.48.235 scans for open 3128 or 8080 ports in the 32.245.0.0/16 networks. If the attacking host finds these ports open it will collect the addresses and forward the

information to a data collection script on a controlling host. Each infected host will continue to do distributed reconnaissance until cleaned of the Trojan.

Probability the source address was spoofed

There is very little probability that the source address was spoofed for the following reasons

- Sequence numbers increment as would be expected.
  According to TCP/IP Illustrated Volume 1 by Stevens, page 232, the sequence numbers should increment by 64000 every .5 second or 128,000 every second. This implies 1280 for every 1/100 of a second. This can be seen in these 2 packets

  023346.186507 IP 24.190.48.235.2395 > 32.245.120.66.8080 S
  14987174191498717419(0) win 8192 <mss1460> (DF)
  023346.196507 IP 24.190.48.235.2478 > 32.245.120.82.8080 S
  14987184861498718486(0) win 8192 <mss1460> (DF)

The difference between 1498718486 and 1498717419 is 1067.
This is reasonably close to the expected 1280 increment. The difference can be attributed to the resolution of the dump. The values beyond 1/100 of a second are not reliable.

- IP id numbers increment as would be expected.
- Source ports increment as would be expected.
- If the attacker is indeed a RingZero infected machine then this activity is recon and the goal is to collect information concerning open hosts to report.
- This scanning activity stopped according to the next day logs located at URL:http//www.incidents.org/logs/Raw/2002.9.20. Evidently, the attacking host was either cleaned or shutdown.

The following windump command was run to verify this probability
windump -vv -r 2002.9.19 -n src host 24.190.48.235 and dst net 32.245.0.0/16
The –vv component implies verbose mode.

The following windump results indicate the above references

022903.966507 IP (id 30425) 24.190.48.235.1233 > 32.245.117.117.8080 S
14950745481495074548(0) win 8192 <mss 1460> (DF)
022903.966507 IP (id 30681) 24.190.48.235.1254 > 32.245.117.121.8080 S
14950747481495074748(0) win 8192 <mss 1460> (DF)
022904.716507 IP (id 41945) 24.190.48.235.1309 > 32.245.117.130.8080 S
14950751591495075159(0) win 8192 <mss 1460> (DF)
022904.716507 IP (id 42201) 24.190.48.235.1310 > 32.245.117.130.3128 S
14950751651495075165(0) win 8192 <mss 1460> (DF)
022904.736507 IP (id 43225) 24.190.48.235.1315 > 32.245.117.131.8080 S
14950751891495075189(0) win 8192 <mss 1460> (DF)

The same command was run against the next day log file, 2002.9.20, with no scanning activity coming from 24.190.48.235.

Description of attack

An open proxy server sometimes offer a tunnel through which hackers can hide their tracks and assume the address of the proxy.
An open proxy server allows others to direct traffic through it and gives the appearance that traffic came through the proxy thus masking the hackers identity.

Attack Mechanism

The RingZero attack searches for open hosts with ports 80, 3128 or 8080 open. Although this detect does not show evidence of scanning for port 80 the scan for 3128 and 8080 is highly suspicious. The potentially infected host, 24.190.48.235, will scan for hosts with these ports open and report back to a central data collection server. If the scanned host completes the 3 way handshake for a targeted port, the infected RingZero host will report this information back to the central data repository and the targeted host could then be used as a proxy to launch other more sophisticated attacks.
It has been speculated that the attack vector is through e-mail of an attached screen saver or game.

Correlations

The web site at URL:http//www.nipc.gov/warnings/advisories/1999/99-024.htm discusses the RingZero trojan and it's implications.
Tim White discusses this Trojan in an article for the securityfocus web site located at URL:http//www.securityfocus.com/archive/75/31239
A search of 24.190.48.235 on the DShield site returns no information concerning this IP address as a source or target of attacks.
Running the same windump as referenced previously against the file URL:http//www.incidents.org/logs/Raw/2002.9.18 resulted in the same activity as seen in the 2002.9.19 dump file. However, running windump against the file URL:http//www.incidents.org/logs/Raw/2002.9.20 resulted in no evidence of scanning from the 24.190.48.235 host.
The scans from 24.190.48.235 occurred over a 10 minute interval on both the 18th and 19th.
There is no evidence of scanning from this host in the September 20th dump file.

Evidence of Active Targeting

There is no evidence of active targeting at a single host but rather an attempt to actively target the 32.245.117.0/24 through 32.245.124.0/24 network segments. No response was found originating from the 32.245.0.0/16 to the scanning host 24.190.48.235.
The following windump command was run to search for any response from the 32.245.0.0/16 network segments to the scanning host

windump -r 2002.9.19 -n src net 32.245.0.0/16 and dst host 24.190.48.235

No hosts on the 32.245.0.0/16 segments responded to the original syn requests from 24.190.48.235.

Severity

Using the severity formula(severity = (criticality + lethality) – (system countermeasures + network countermeasures)), I have assigned the following values and reasons for these assignments

Criticality 2
Reason Scan directed at hosts on network segments and not individually targeted.
Lethality 1
Reason Older exploit with patched systems is unlikely to succeed.
Countermeasures 6
System 3
Reason No responses from scanned segments. Probably patched, wrapped and firewalled .
Network 3
Reason Scans were seen on the networks. Firewall and IDS should have blocked and/or killed the scan.
Severity = (2 +1) – 6 = -3

Defensive Recommendations

Although no evidence of response to the scan stimuli exists, the packet scans still penetrated the perimeter of the 32.245.0.0/16 segments. These ports can be blocked at the filtering gateway and/or statefull firewall.
In addition, NIDS can be deployed to eliminate or kill these scans based on event filtering, signature analysis and/or periodicity of occurrence.

Multiple Choice Test Question

If tcpdump or windump is used to capture raw binary data on your network and your dump file shows that several ports on multiple hosts of your network have been scanned, which response is best

A) Post logs to incident.org.
B) Contact the associated sysadmins and verify scanned ports are not open.
C) Examine dump file for evidence of response to scanner from scanned hosts.
D) Delete the logs and and ignore the alerts.

Answer C

Reason In any scan, the analyst is primarily concerned with evidence of response to the stimulus. Sysadmins may very well verify that a port is closed but an infected host may open or close ports at their leisure. Posting logs to incident.org does not alleviate the security analyst from practicing due diligence.

Questions from the community

Date 12/16/03
Kevin,

I am preparing for my assignment 2 of GCIA ver 3.4 also, I read your detect below and came up with one question. What made you think this pattern of scanning has anything to do with ringzero. I mean did you googled for port 3128 and "scan squid proxy attempt" scans ?
Just curious what led you to believe it has something to do with an automatic attack tool.
Ethin@myrealbox.com
Thanks
Ethin

Response to Ethin

Part of our course material covered the RingZero trojan and identified ports 80, 3128 and 8080 as scanned ports by the trojan. I may be way off base on this, being a novice in this area, but the exercise expects one to speculate on why we may see this traffic. I do not have any tangible evidence of RingZero, but I did speculate on the scan and what perhaps could have been the stimulus. I have discovered in the last day that part of the signature discussed in the course material concerning RingZero is that the ip id field remains the same. Could this be changed to impersonate legitimate ip ids? I believe so. The nature of the scan suggests to me that it is automated. It appears that each network segment was completely scanned, although I have not seen a specific pattern.

Thanks for you questions.

kdk

Evidence of Active Targeting
>There is no evidence of active targeting at a single host but rather an
>attempt to actively target the 32.245.117.0/24 through 32.245.124.0/24
>network segments. No response was found originating from the 32.245.0.0/16
>to the scanning host 24.190.48.235.

Why would this be? You have not made any mention, that I have noticed about your knowledge of the IDS ruleset for outgoing responses, therefore if there was a response would it have been recorded?
ov@mdsi.bc.ca

Response to Oliver

I see where I may have been confused. Evidently, the outgoing IDS ruleset potentially has a snort rule option set as follows

```
alert tcp 32.245.0.0/16 3128  -> any any  (flags SA; \
resp rst_rcv;)
```

In this case the resp option indicates to kill the receiving socket responding from the proxy port and not log the event.
I would think that all pertinent events would be logged though. I imagine if a site were being constantly scanned, then I can see where one may disable logging of the known

scans to preserve disk space and not experience a DoS on the disk by having the disk space exhausted with log entries.

Thanks for the comments.

You wrote
Who owns 24.190.x.x? Given that information, what conclusion can you draw on the reliability of any Dshield information (or lack thereof)?

Response
I believe I understand your implications. Evidently, this segment belongs in the cable tv spectrum and more than likely are not very concerned with security and intrusions or reporting. This company, being an ISP probably does not care what their paying customers may or may not be doing. I probably should have come to this realization on my own, but did not.
Thanks for the insight.

You also wrote
Since the log files which you analyzed only contain logs for data which generated alerts, are you sure there was no response given by the 32.245.0.0/16 network?

Response
You are absolutely right. I am not sure that a host on this segment did not respond. To be truthful, I initially believed the logs to be raw tcpdump files from some network interface and that the data was not initially filtered. When I ran snort and windump against these binaries, I guess I became confused by the magnitude of the information and the various events of interest. Since my posting, I have discovered exactly that of which you speak. During my analysis, I kept asking "Why am I not seeing any syn/ack or fins" only to get bogged down in the minutae of the snort alerts.

I believe my second detect will be more thorough than my first.
Jason.Thompson@xwave.com

Network Detect #2

Source of Trace
The detect was sourced from URL:http//www.incidents.org/logs/Raw/2002.5.1.

Based on the results of a snort generated analysis of the raw log file, it appears that the network IDS is located outside the 236.185.0.0/16 network segments. The volume of alerts generated suggests the network appears to look like this

External ----- Firewall -------- Router ------- | -------- 236.185.0.0/16
Internet |
                          NIDS

Detect was generated by

The following snort priority 1 events of interest were generated by Snort Version 2.0.3 executed against the 2002.5.1 log file. I decided to examine this event because of the indicated priority being “high”.

[**] [113903] SHELLCODE x86 inc ebx NOOP [**]
[Classification Executable code was detected] [Priority 1]
06/01-084918.704488 62.168.63.24580 -> 226.185.106.17664359
TCP TTL40 TOS0x0 ID7548 IpLen20 DgmLen1500 DF
AP Seq 0xEA7ACE08 Ack 0x26BE8B47 Win 0x7D78 TcpLen 20

[**] [113903] SHELLCODE x86 inc ebx NOOP [**]
[Classification Executable code was detected] [Priority 1]
06/01-084918.874488 62.168.63.24580 -> 226.185.106.17664359
TCP TTL40 TOS0x0 ID7579 IpLen20 DgmLen1500 DF
AP Seq 0xEA7AD648 Ack 0x26BE8C39 Win 0x7D78 TcpLen 20

[**] [113903] SHELLCODE x86 inc ebx NOOP [**]
[Classification Executable code was detected] [Priority 1]
06/01-084919.034488 62.168.63.24580 -> 226.185.106.17664359
TCP TTL40 TOS0x0 ID7607 IpLen20 DgmLen1500 DF
AP Seq 0xEA7AE56C Ack 0x26BE8D2B Win 0x7D78 TcpLen 20

[**] [113903] SHELLCODE x86 inc ebx NOOP [**]
[Classification Executable code was detected] [Priority 1]
06/01-084919.194488 62.168.63.24580 -> 226.185.106.17664359
TCP TTL40 TOS0x0 ID7659 IpLen20 DgmLen1500 DF
AP Seq 0xEA7AF565 Ack 0x26BE8E1D Win 0x7D78 TcpLen 20

[**] [113903] SHELLCODE x86 inc ebx NOOP [**]
[Classification Executable code was detected] [Priority 1]
06/01-084919.494488 62.168.63.24580 -> 226.185.106.17664359
TCP TTL40 TOS0x0 ID7726 IpLen20 DgmLen1500 DF
AP Seq 0xEA7B0134 Ack 0x26BE9002 Win 0x7D78 TcpLen 20

The snort rules which generated these alerts were

alert ip $EXTERNAL_NET any -> $HOME_NET $SHELLCODE_PORTS (msg"SHELLCODE x86 inc ebx NOOP"; content"|43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43|"; classtypeshellcode-detect; sid1390; rev3;)

These rules basically state that any external source host and port bound for the internal protected network on the shell code port of 80, as defined in the snort.conf file, constitutes an attempt by the source host to execute code on the destination host.

The sid for 1390 gives the following description

Summary This event is generated when an attempt is made to execute shellcode on a host in the protected network from a source external to that network.
Impact This set of instructions can be used as a NOOP to pad buffers on an x86 architecture machines.
Detailed Information This is the x86 opcode for 'inc ebx'. This can be used as a NOOP in an x86 architecture, however as with all shellcode rules, this can cause false positives. Check to see if you are ignoring shellcode rules on web ports, as this will reduce false positives.

The above information was obtained at URL:http//www.snort.org/cgi-bin/sigs-search.cgi?sid=sid+.

The rule header consists of this snipet “alert ip $EXTERNAL_NET any -> $HOME_NET $SHELLCODE_PORTS”. As stated above, any external source or port directed at an internal host on port 80and the appropriate content will trigger these rules and alerts will be generated.
The rule options are “(msg"SHELLCODE x86 inc ebx NOOP"; content"|43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43|"; classtypeshellcode-detect; sid1390; rev3;)”.
The alert message keyword, msg, will be " SHELLCODE x86 inc ebx NOOP". The semicolon represents a new rule option. In this case, content of ascii character 43 repeated in succession 24 times is a characteristic of this signature.

The rule option classtype signifies a priority 1 classification of “shellcode-detect” as specified in the classification.config file of snort. The classification.config file shows the following entry
“config classification shellcode-detect,Executable code was detected,1”.
The number 1 at the end represents the priority.

This exploit is directed at the ebx register on x86 machines.

The snort command used to generate the above alerts is as follows

snort -d -h 226.185.0.0/16 -l c\snort203\bin\log -c c\snort203\bin\snort.conf -r 2002.5.1 -k none

-d	Dump the application layer
-h	Set the Home reference network
-l	Log to directory
-c	Use the snort.conf file

-r	Read from the binary input file
-k none	No checksum mode

I decided to take a look at the payload of the above referenced packets and ran the following windump command

windump -vv -X -r 2002.5.1 -n src host 62.168.63.245 and dst host 226.185.106.176
The various options are described below

-vv	Be verbose
-X	Dump hex and ascii payload
-r	Read from binary input file
-n	Do not resolve addresses to names
src host	Host of origin
dst host	Host of destination

The resulting ascii output of the windump command is as follows

46 43 FF DB 00 43 01 0C 0C 0C 10 0E 10 20 12 12	FC...C..... ..
20 43 2D 26 2D 43 43 43 43 43 43 43 43 43 43	C-&-CCCCCCCCCCC
43 43 43 43 43 43 43 43 43 43 43 43 43 43 43	CCCCCCCCCCCCCCC
43 43 43 43 43 43 43 43 43 43 43 43 43 43 43	CCCCCCCCCCCCCCC
43 43 43 43 43 43 43 FF C0 00 11 08 00 55 00 6A	CCCCCCC.....U.j

As can be seen from this output the alpha character “C” is repeated over 24 times and will trigger the shell-code alert. These are NOOP characters on the x86 platform and could indicate that the source host is attempting to pad the buffer with NOOP characters prior to introducing an exploit in machine code.

To take a closer look at the ethernet frame header, the following windump command was issued

windump -e -r 2002.5.1 -n host 62.168.63.245 and host 226.185.106.176
Again the windump options are described below

-e	Dump the ethernet frame header
-r	Read from binary input file
-n	Do not resolve addresses to names
host	Capture traffic between the 2 hosts

084918.704488 03e3d926c0 00c4b233 0800 1514 IP 62.168.63.245.80 > 2
26.185.106.176.64359 P 393391668039339181 40(1460) ack 650021703 win 32120 (DF)
084918.874488 03e3d926c0 00c4b233 0800 1514 IP 62.168.63.245.80 > 226.185.106.176.64359 P 21123572(1460) ack 243 win 32120 (DF)
084919.034488 03e3d926c0 00c4b233 0800 1514 IP 62.168.63.245.80 >

226.185.106.176.64359 P 59887448(1460) ack 485 win 32120 (DF)
084919.194488 03e3d926c0 00c4b233 0800 1514 IP 62.168.63.245.80 >
226.185.106.176.64359 P 1007711537(1460) ack 727 win 32120 (DF)
084919.494488 03e3d926c0 00c4b233 0800 1514 IP 62.168.63.245.80 >
226.185.106.176.64359 P 1310014560(1460) ack 1212 win 32120 (DF)

From this output it can be inferred that the host on our protected network initiated a connection to the web server at 62.168.63.245 and that the web server is responding back to our protected server with data as represented by the Push and ack flags being set. We do not see the initial tcp 3 way handshake because the dump data is only initiated when the snort alert triggers.
We can also see that the 1514 bytes on the wire represent the 1460 bytes of data + 20 bytes for the ip header + 20 bytes for the tcp header + 14 bytes for the ethernet header. This is referenced in *TCP/IP Iiustrated Volume 1,* by Stevens on page 56. These 14 bytes of information are represented by 6 bytes for the MAC destination address, 6 bytes for the MAC source addresss and 2 bytes for the frame type. In this case the frame type is 0x0800 which indicates an IP Datagram as indicated on page 23 of the Stevens book.

Probability the source address was spoofed

There is very little probability that the source address was spoofed for the following reasons
As can be seen in the following output from windump the ip ids change as would be expected.
windump -vv -r 2002.5.1 -n src host 62.168.63.245
The –vv component implies verbose mode. Output has been snipped for clarity.

084918.704488 IP (id 7548) 62.168.63.245.80 > 226.185.106.176.64359 P
393391668039339181 40(1460) ack 650021703 win 32120 (DF)
084918.874488 IP (id 7579) 62.168.63.245.80 > 226.185.106.176.64359 P
21123572(1460) ack 243 win 32120 (DF)
084919.034488 IP (id 7607) 62.168.63.245.80 > 226.185.106.176.64359 P
59887448(1460) ack 485 win 32120 (DF)
084919.194488 IP (id 7659) 62.168.63.245.80 > 226.185.106.176.64359 P
1007711537(1460) ack 727 win 32120 (DF)
084919.494488 IP (id 7726) 62.168.63.245.80 > 226.185.106.176.64359 P
1310014560(1460) ack 1212 win 32120 (DF)

In addition, the ack numbers correspond to that which is expected. According to Steven's *TCP/IP Iiustrated Volume 1* on page 234 tcpdump/windump will display sequence numbers on the initial syn segment and all subsequent sequence numbers as relative offsets from the original sequence number. This also holds true for the ack numbers. As can be seen from the above output the ack numbers increment as could be expected.

Description of attack

The shellcode attack signature is that the attacker will attempt to gain access to a vulnerable host and push data onto the stack of the vulnerable host hoping to reach a point on the stack where it becomes full and then the attackers executable code can be introduced to the victim. There is something known as a shellcode sled where the attacker slides his NOOP instructions along the payload of the data in hopes to find the point on the stack where the attackers code is then executed. This particular alert is directed at x86 machines and the ebx register.

Attack Mechanism

The attack mechanism is to push a string of data in the payload of a connection until the attackers appended code becomes executed. The "43" ascii characters are NOOP instructions to the x86 cpu and simply use cpu cycles without executing any command instruction. Once the buffer length has been exhausted arbitrary code can then be executed from the packet payload.

Correlations

Dshield at URL:http//www1.dshield.org/ipinfo.php?ip=62.168.63.245&Submit=Submit returns the following information regarding 62.168.63.245

HostNamephoebe-i.czechia.com
There is no information on the DShield site as to attacks directed to or from this host.

The 226.185.106.176 address returns the following information from DShield

```
NetRange    224.0.0.0 - 239.255.255.255
CIDR        224.0.0.0/4
NetName     MCAST-NET
NetHandle   NET-224-0-0-0-1
Parent
NetType     IANA Special Use
NameServer  FLAG.EP.NET
NameServer  STRUL.STUPI.SE
NameServer  NS.ISI.EDU
NameServer  NIC.NEAR.NET
Comment     This block is reserved for special purposes.
Comment     Please see RFC 3171 for additional information.
```

RFC 3171 located at URL:http//www.faqs.org/rfcs/rfc3171.html returns the following information

225.0.0.0 - 231.255.255.255 RESERVED

Evidently the 226.185.106.176 address belongs in the multicast group for the IANA.

Evidence of Active Targeting

There is no evidence of active targeting. In fact, this alert could very well be nothing more than a false positive. As indicated by the following windump command, the payload of the data references some sort of .jpeg file which according to a discussion thread posted at the following URL:http//archives.neohapsis.com/archives/sf/ids/2002-q2/0019.html could very well trigger a false alarm of snort.

```
20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 43 6F 6E    Keep-Alive..Con
74 65 6E 74 2D 54 79 70 65 3A 20 69 6D 61 67 65    tent-Type image
```

2F 6A 70 65 67 0D 0A 0D 0A FF D8 FF E0 00 10 4A /jpeg.....

The command used to generate the above information is as follows
windump –Xev -r 2002.5.1 -n src host 62.168.63.245 and dst host 226.185.106.176

-X	Dump payload in ascii
-e	Dump the ethernet frame header
-v	verbose
-r	Read from binary input file
-n	Do not resolve addresses to names
host	Capture traffic between the 2 hosts

Severity
Using the severity formula(severity = (criticality + lethality) – (system countermeasures + network countermeasures)), I have assigned the following values and reasons for these assignments

Criticality 1
Reason The 226.185.106.176 host has ttl fingerprints resembling a Win2k host.
Lethality 1
Reason Shellcode directed at a Win2k host is unlikely to succeed.
Countermeasures 6
System 3
Reason Although the ttl fingerprint suggest Win2k, I am not sure as to patch level and/or added security features.
Network 2
Reason Too much uncertainty surrounding the network measures in place.
Severity = (1 +1) – 5 = -3

Defensive Recommendations
Make sure the host is patched to the latest levels. Install added security mechanisms such as anti-virus protection, tcpwrappers and ssh. Make sure no default or weak passwords are in place.

Multiple Choice Test Question
Although running windump with the –e flag may result in the following output
181904.914488 00c4b233 03e3d926c0 0800 1514 IP 226.185.106.176.6486
0 > 64.154.80.51.80 P 83395359783395505 7(1460) ack 3461013700 win 33580
Account for the difference in data pushed, 1460 bytes, and bytes on the wire 1514 .
Select the best answer.

A) 1460 data bytes and 54 bytes of padding.
B) 1460 data bytes and 54 bytes for the ip and tcp headers.
C) 1460 data bytes and 54 bytes for all frame headers.
D) 1460 data bytes and 20 bytes ip header and 20 bytes tcp header and 14 bytes ethernet header.

Answer C
Reason Unless you can see the hex dump and verify the ip and tcp header length, then the assumption that each is 20 bytes long could be wrong.

Network Detect #3

Source of Trace
The detect was sourced from URL:http//www.incidents.org/logs/Raw/2002.9.27.

Based on the results of a snort generated analysis of the raw log file, it appears that the network IDS is located outside the 32.245.0.0/16 network segments. The volume of alerts generated suggests the network appears to look like this

External ----- Firewall -------- Router ------- | -------- 32.245.0.0/16
Internet |
NIDS

Detect was generated by
The following snort priority 2 events of interest were generated by Snort Version 2.0.3 executed against the 2002.9.27 log file. I selected this alert to investigate information pertaining to the nmap scan.

[**] [16282] SCAN nmap TCP [**]
[Classification Attempted Information Leak] [Priority 2]
10/26-195414.366507 140.128.251.2180 -> 32.245.15.14280
TCP TTL50 TOS0x0 ID44378 IpLen20 DgmLen40
A* Seq 0xA3 Ack 0x0 Win 0x578 TcpLen 20
[Xref => http//www.whitehats.com/info/IDS28]

[**] [16282] SCAN nmap TCP [**]
[Classification Attempted Information Leak] [Priority 2]
10/26-195417.366507 140.128.251.2180 -> 32.245.15.14280
TCP TTL50 TOS0x0 ID44651 IpLen20 DgmLen40
A* Seq 0x107 Ack 0x0 Win 0x578 TcpLen 20
[Xref => http//www.whitehats.com/info/IDS28]

[**] [16282] SCAN nmap TCP [**]
[Classification Attempted Information Leak] [Priority 2]
10/27-010007.616507 62.0.23.6180 -> 32.245.166.12180
TCP TTL49 TOS0x0 ID2092 IpLen20 DgmLen40
A* Seq 0xEC Ack 0x0 Win 0x578 TcpLen 20
[Xref => http//www.whitehats.com/info/IDS28]

The snort rules which generated these alerts were

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg"SCAN nmap TCP"; flagsA,12; ack0; referencearachnids,28; classtypeattempted-recon; sid628; rev2;)

This rule states that any external source host and any port bound for the internal protected network on any port with the ack flag set and a value of zero indicates an attempt by the source host to scan the destination host with the nmap program.

The sid for 628 gives the following description

"Summary: This event is generated when the nmap port scanner and reconnaissance tool is used against a host."
"Detailed Information: Some versions of Nmap's TCP ping, if selected, sends a TCP ACK with an ACK number = 0.
Nmap can use TCP ping as a second alternative to ICMP Ping."

The above information was obtained at URL:http//www.snort.org/cgi-bin/sigs-search.cgi?sid=sid+.

The rule header consists of this snipet "alert tcp $EXTERNAL_NET any -> $HOME_NET any" As previously stated, any external source or port directed at an internal host on any port with the protocol being tcp and the ack flag set with a value of 0x0 will trigger this rule and alerts will be generated.
The rule options are "(msg"SCAN nmap TCP"; flagsA,12; ack0; referencearachnids,28; classtypeattempted-recon; sid628; rev2;)"

The alert message keyword, msg, will be "SCAN nmap TCP ". The semicolon represents a new rule option. In this case, the ack flag set with a value of 0x0. Also, the "flagsA,12" component, represents the keyword "flags" with a value of A,12. This represents a tcp packet where the Ack flag is set and the reserved bits are masked and their value is of no concern. The "referencearachnids, 28" directs the analyst to reference material on the nmap scan exploit.
The rule option classtype signifies a priority 2 classification of "attempted-recon" as specified in the classification.config file of snort.

The snort command used to generate the above alerts is as follows

snort -d -h 32.245.0.0/16 -l c\snort203\bin\log2 -c c\snort203\bin\snort.conf -r 2002.9.27 -k none

-d	Dump the application layer
-h	Set the Home reference network
-l	Log to directory
-c	Use the snort.conf file
-r	Read from the binary input file
-k none	No checksum mode

I decided to take a look and see if any other activity was directed at the protected hosts and ran the following windump command

windump -vv -r 2002.9.27 -n host 32.245.15.142 or host 32.245.166.121

The various options are described below

-vv	Be verbose
-r	Read from binary input file
-n	Do not resolve addresses to names
host	Look at any traffic with source or destination as 32.245.15.142 or 32.245.166.121

The resulting output of the windump command is as follows

195414.366507 IP (tos 0x0, ttl 50, id 44378, len 40) 140.128.251.21.80 > 32.245.15.142.80 . 163163(0) ack 0 win 1400
195417.366507 IP (tos 0x0, ttl 50, id 44651, len 40) 140.128.251.21.80 > 32.245.15.142.80 . 100100(0) ack 1 win 1400
010007.616507 IP (tos 0x0, ttl 49, id 2092, len 40) 62.0.23.61.80 > 32.245.166.121.80 . 236236(0) ack 0 win 1400
It can be seen from this output that the scanning host at 140.128.251.21 sent an ack with a value of 0x0 to 32.245.15.142 port 80 and that the windump output incremented the ack number by 1 as is it's default configuration specifies. The output also shows another scanner directing a nmap tcp scan at 32.245.166.121. I will concentrate this analysis on the first two scans of the 32.245.15.142 host.

To take a closer look at the ethernet frame header and hex output, the following windump command was issued

windump -ex -r 2002.9.27 -n host 32.245.15.142 or host 32.245.166.121

Again the windump options are described below

-e	Dump the ethernet frame header
-x	Dump the header and payload in hex
-r	Read from binary input file
-n	Do not resolve addresses to names
host	Capture traffic between either of the 2 hosts

195414.366507 03e3d926c0 00c4b233 0800 60 IP 140.128.251.21.80 > 32.245.15.142.80 . ack 0 win 1400

0000	4500 0028 ad5a 0000 3206 1044 8c80 fb15
0016	20f5 0f8e 0050 0050 0000 00a3 0000 0000
0032	5010 0578 dde7 0000 0000 0000 0000

195417.366507 03e3d926c0 00c4b233 0800 60 IP 140.128.251.21.80 > 32.245.15.142.80 . ack 1 win 1400

0000 4500 0028 ae6b 0000 3206 0f33 8c80 fb15
0016 20f5 0f8e 0050 0050 0000 0107 0000 0000
0032 5010 0578 dd83 0000 0000 0000 0000
010007.616507 03e3d926c0 00c4b233 0800 60 IP 62.0.23.61.80 > 32.245.166.121.80 . ack 0 win 1400

0000 4500 0028 082c 0000 3106 4fe1 3e00 173d
0016 20f5 a679 0050 0050 0000 00ec 0000 0000
0032 5010 0578 770d 0000 0000 0000 0000

From this output it can be seen that the source host sends an ack to the protected host at 32.245.15.142. The ack id is represented in the 8th byte offset of the tcp header which begins with 0050 0050. It can be seen in the second packet that the 8th byte offset of the tcp header is also 0000 0000 which indicates a value of 0x0. This is contrary to the previous windump output which shows an ack value of 1. This is caused by the default configuration of windump in that the syn and ack values are represented as relative offsets from the original syn or ack flag.

We can also see that the 60 bytes on the wire represent the 20 bytes for the ip header + 20 bytes for the tcp header + 14 bytes for the ethernet header. This is referenced in *TCP/IP Iiustrated Volume 1,* by Stevens on page 56. These 14 bytes of information are represented by 6 bytes for the MAC destination address, 6 bytes for the MAC source addresss and 2 bytes for the frame type. In this case the frame type is 0x0800 which indicates an IP Datagram as indicated on page 23 of the Stevens book. According to RFC 894 located at URL:http//www.ietf.org/rfc/rfc0894.txt?number=894 the minimum size for an Ethernet frame is 46 bytes. To handle this pad bytes of 0000 0000 0000 have been inserted. This accounts for the 60 bytes on the wire. 20 ip header + 20 tcp header + 14 ethernet header + 6 bytes of padding.

Probability the source address was spoofed

It is not likely that the source address was spoofed.
Concentrating on the 140.128.251.21 host, we infer that the scanner is gathering information as to whether port 80 of the 32.245.15.142 will respond to the scanner with a ack/rst or no response at all. With the nmap scanner capturing the response to the stimulus, the scanner can determine the OS of the internal host and concentrate their efforts towards exploiting known vulnerabilities which may exist on the particular OS. Unfortunately, we do not see any response dumps from the host because of the nature of the source dump. We only see output when a snort alert is triggered.

Another item of interest in the Ethernet header dump is that the MAC addresses do not make sense. All three source MAC addresses are the same even though the ip addresses are not. Also, all three destination MAC addresses are the same even though

the ip addresses are not. I believe that this is a result of obfuscation on behalf of the dump file collectors.

Description of attack

The nmap scan is an attempt to identify internal servers listening on a specific port. In a stateless environment the scan will bypass or elude the firewall and allow the packet through to the destination. If this packet attempted to pass through a firewall which maintained state, then this packet would be dropped because of the lack of a corresponding syn.

Attack Mechanism

The attack mechanism is to send a crafted packet using nmap to a destination host behind a border filtering device. Because the firewall probably does not allow inbound icmp echo-requests, the attacker will use nmap to identify open ports behind the firewall or filtering device. The attacker will send an ack to the internal host, bypassing the stateless firewall and elicit a response back from the attacked host. This response will more than likely be of the form of a ack/rst back to the scanner if the port is not open or no response if the port is open. The response characteristics and signature of the responder will allow nmap to identify the OS. This will aid the attacker in formulating a more specific attack.

Correlations

According to DShield this address has been assigned to APNIC or the Asian Pacific Network Information Centre. Searching the APNIC at URL:http//www.apnic.net/apnic-bin/whois.pl results in the following

inetnum 140.128.0.0 - 140.128.255.255
netname TANET
descr Taiwan Academic Network
descr Ministry of Education computer Center

The following information was garnered from URL:http//www.whitehats.com/cgi/arachNIDS/Show?_id=ids28&view=event

"Summary: This event indicates that a remote user has used the NMAP portscanning tool to probe the server. An NMAP TCP ping was sent to determine if a host is reachable."
"How Specific: This event is specific to a particular exploit, but the packet payload is not considered as part of the signature to detect the attack."

The CVE number is CAN-1999-0523
Although whitehats references this CVE number at their site, the context of the correlation is concerned with icmp echo and ping not nmap.

The following information comes from URL:http//www.iss.net/security_center/advice/Intrusions/2000310/default.htm

> Firewalls can block incoming connections by blocking only the first few frames of a connection. Hackers can therefore "pierce" firewalls by

crafting what appear to be responses. Since the firewall believes these to be legitimate responses, they forward them on through. This technique cannot be used to compromise the target system, but it can be used to scan the system. If the target of this hacker scan is able to process the indicated traffic, it will send a message back to the hacker. The intent is to inform the sender of a communications error. However, it really informs the hacker that there is something there that they could potentially hack. The hacker's next steps are to find ways to get around the firewall in order to reach this target.

Evidence of Active Targeting

There is definitely evidence of active targeting. The scanner directed the nmap scan at specific hosts behind the protected network. This indicates that the scanner knew that for which the scanner sought. By virtue of the scan being used to bypass a firewall implies potential knowledge of the network architecture.

Severity

Using the severity formula(severity = (criticality + lethality) – (system countermeasures + network countermeasures)), I have assigned the following values and reasons for these assignments

Criticality 3
Reason The 32.245.15.142 host has been scanned directly. The firewall was bypassed.
Lethality 1
Reason According to whitehats at URL:http//www.whitehats.com/cgi/arachNIDS/Show?_id=ids28&view=research this is probably an older version of nmap that sets the ack to zero and may not be quite effective.
Countermeasures 6
System 3
Reason I am not sure the OS is up to date and patched.
Network 3
Reason I cannot say that the scan did not succeed. The firewall may not be restrictive.
Severity = (3 +1) – 6 = -2

Defensive Recommendations

Make sure the host is patched to the latest levels. Install a restrictive firewall. Log all traffic in and out of the perimeter in a circular file and retain at a specific periodicity to your log server. Save and truncate as necessary to preserve disk space.

Multiple Choice Test Question

If a scanner such as nmap solicits a response from a host by setting the ack flag what will the response probably be?

A) None
B) Ack/rst

C) Fin
D) It depends
Answer D

Reason If the packet passes through a statefull firewall the packet should be dropped. If the packet passes through a stateless firewall, different OSes will behave differently. See URL:http//www.insecure.org/nmap/nmap-fingerprinting-article.html

Assignment 3 --- Analyze This

Executive Summary

The abc co. was contracted by The University to provide insight as to their state of cyber security. Our mission is to analyze the data taken from their networks and make effective recommendations for their plan of action to increase their security. We have collected many scan, alert and out of spec files to analyse. We will provide the necessary information to management in order for them to make informed decisions concerning The University's network assets and security.

The general methodology used in this analysis was to load up a MSAccess database with the below referenced log files and then parse through the various fields with various SQL queries identifying significant events of interest. Various other tools were used in the analysis of these mass of logs. One of these tools was wingrep, a windows text search engine. Another tool used was snort_sort.pl which sorts snort alert files by alert into an html file. Also used to some extent was TextTools which allows the analyst to pipe and sort files. The text files were initially viewed using the sort, type and the more function from a DOS window in order to understand the file contents. The smaller files were brought into MSWord, formatted and saved as straight text files. The data was then imported into the MSAccess database as text delimited files.
The larger scan files were imported directly into a MSAccess database, formatted, exported and then re-imported to split the various fields. Due to the huge volume of data and the limitations of MSAccess, the resulting database had to be split to make various queries more efficient. For example, one scan file contained over 3 million rows of data. The scan files were split between 2 different databases, similar tables and queries created and output was consolidated in the report. Tables were created within the database to hold alert, scan and oos data for each day. Snort_sort.pl was used to create html output of the various alert logs for each day. The alerts were inspected and queries were generated against the database to identify the Top scanners, alerters and oos entires. Once this information was gathered, the top scanners, alerters and oos entries were "beaten" against the alert logs to find correlations amongst the top alerters, scanners and oos log file entries.
These databases aided by the other referenced tools were the tools and mechanisms used in making this analysis.

I am sure The University executives want to know what we have uncovered regarding The University's overall cyber security and what they need to do to protect their computer assets.

I have identified several areas where the university will need to become proactive in taking charge of their networks. The areas of interest are

Trojan server activity.
Trojan activity.
Sun RPC activity.
IRC activity.
Print activity.

Files Used

File	Size	File	Size	File	Size
scans.031202.gz	93M	alert.031202.gz	18M	oos_report_031202.txt	3M
scans.031203.gz	211M	alert.031203.gz	33M	oos_report_031203.txt	3M
scans.031204.gz	194M	alert.031204.gz	32M	oos_report_031204.txt	3M
scans.031205.gz	192M	alert.031205.gz	36M	oos_report_031205.txt	3M
scans.031206.gz	213M	alert.031206.gz	34M	oos_report_031206.txt	3M

Assumptions

In selecting files to evaluate, I discovered that the out of spec files contained all the same content after 10/27/2003. This fact was brought to SANS attention and I was instructed to use those files as they exist. These out of spec files will be used in conjunction with the scan and alert files to provide insight into the network vulnerabilities and risk exposure.

Based on information in the log files, I have determined that the protected network is the MY.NET.0.0/16. The log files referenced above are relative to this network. The scan files above do not contain any information with regards to MY.NET.0.0/16 but rather contain volumes of information surrounding the 130.85.0.0/16 segments. My assumption is that the MY.NET.0.0/16 and the 130.85.0.0/16 are the same. The IDS which captured the log files is located at the border to the MY.NET.0.0/16 network segments. My understanding is that the version of snort used is an older version, perhaps 1.8, but I cannot be certain.
MY.NET and 130.85 correspond to the below listed information from DShield

OrgName	University of Maryland Baltimore County
OrgID	UMBC
Address	1000 Hilltop Circle
City	Baltimore
StateProv	MD
PostalCode	21250
Country	US

NetRange	130.85.0.0 - 130.85.255.255
CIDR	130.85.0.0/16
NetName	UMBCNET

NetHandle NET-130-85-0-0-1
Parent NET-130-0-0-0-0
NetType Direct Assignment
NameServer UMBC5.UMBC.EDU
NameServer UMBC4.UMBC.EDU
NameServer UMBC3.UMBC.EDU
Comment
RegDate 1988-07-05
Updated 2000-03-17

TechHandle JJS41-ARIN
TechName Suess, John J.
TechPhone +1-410-455-2582
TechEmail jack@umbc.edu

Alerts and Analysis

Over the five day period beginning 12/02/03 and ending 12/06/03, 37 different snort alerts were generated. The following snort alerts were associated with the top alerter for each day of the 5 day period and the highly suspect internal alerts.

Alerts Seen for Each Day From the Top Alerter		Highly Suspect Internal Alerts
SUNRPC highport access!	12/02/03	Connect to 515 from inside
SYN-FIN scan!	12/03/03	Possible Trojan server activity 27374
Trojan server activity port 51443	12/04/03	UMBC NIDS IRC Alert
MY.NET.30.3 activity—Ports 524,8009	12/05/03	2--High port 6535 **udp/tcp** – possible Red Worm
MY.NET.30.3&4 activity---sport 1033	12/06/03	UMBC NIDS IRC sdbot /kill activity

I will concentrate this analysis on the events associated with the most active hosts and/or ports for each day. From there I will look at the alerts which The University should be highly concerned.

The following is a summary of the Top 10 external hosts which triggered snort alarms for each day on incoming traffic in the alert logs

12/02 src alerts	12/02 src hits	12/03 src alerts	12/03 src hits	12/04 src alerts	12/04 src hits	12/05 src alerts	12/05 src hits	12/06 src alerts	12/06 src hits
68.3.197.224	1802	202.5.152.235	8347	68.33.138.193	1172	68.50.114.89	16104	68.57.90.146	1340
68.34.120.219	798	68.34.120.219	1085	68.32.127.158	768	67.21.63.15	5836	64.12.31.4	744
131.92.177.18	687	68.55.62.79	386	131.92.177.18	492	68.48.90.101	2501	68.55.62.79	402
68.55.62.79	202	68.55.195.133	211	68.55.62.79	340	68.32.122.89	1821	62.163.87.57	309
68.55.52.234	121	63.84.193.226	198	68.57.90.146	292	68.54.168.204	1368	68.50.114.89	155
68.55.53.222	108	195.217.253.40	164	128.153.198.225	177	68.57.90.146	857	68.55.144.24	140
68.55.27.157	96	66.196.72.50	124	66.196.72.23	144	68.55.85.180	438	68.55.27.157	120
66.196.72.17	77	68.55.53.222	113	68.55.27.157	143	68.55.62.79	360	165.247.85.240	111
64.242.195.86	72	66.196.72.53	112	63.251.52.75	143	68.55.27.157	346	165.247.95.11	104
66.196.72.15	58	66.196.72.14	96	66.196.72.29	135	68.55.113.194	295	151.196.165.180	102

Top Alerter for 12/02

If we examine the top alert on 12/02 we find 1802 entries in the alert log from 68.3.197.224 as shown below. Researching the html output for this IP address shows the following snort alert associated with this communication

SUNRPC highport access!

#Entries	Field2	src	srcport	Field5	dst	dstport
1802	12/02-201503.552636	68.3.197.224	6882	->	MY.NET.97.36	32771

This activity shows this src going after the RPC port on an internally protected host. This is certainly indicative of a concerted attempt to exploit the internal host. The source port indicates a programmatic or crafted package to access MY.NET.97.36.
In investigating this alert I decided to search the Sun website and discovered the following information at this URL
http//sunsolve.sun.com/pub-cgi/retrieve.pl?doctype=coll&doc=secbull/142&type=0&nav=sec.sba

> **Understanding the Vulnerability** The rpcbind program is a server that converts RPC program numbers into universal addresses. When an RPC service is started, it tells rpcbind the address at which it is listening, and the RPC program numbers it is prepared to serve. When a client wishes to make an RPC call to a given program number, it first contacts rpcbind on the server machine to determine the address where RPC requests should be sent. Under Solaris 2.x, rpcbind listens not only on TCP port 111 and UDP port 111, but also on a UDP port number greater than 32770. The exact number depends on the OS release and architecture. This results in a large number of packet filters which intend to block access to rpcbind/portmapper being ineffective. Instead of sending requests to TCP or UDP port 111, the attacker simply sends them to the other UDP port. This vulnerability allows an attacker to obtain remote RPC program information even if TCP or UDP port 111 is being filtered. It can also aid an attacker to gain unauthorized access to hosts running vulnerable versions of the software.

Defensive Recommendations

The recommended defense against this type of problem is to patch your systems and block port 32771 through 34000 at the border.

Further activity on the protected network associated with RPC is shown with the following output from the 12/02 alert log

Field2	src	srcport	Field5	dst	dstport
12/02-112741.687895	128.174.80.128	443	->	MY.NET.163.142	32771
12/02-112741.689538	128.174.80.128	443	->	MY.NET.163.142	32771
12/02-121730.584629	66.35.250.150	80	->	MY.NET.100.203	32771
12/02-121730.601001	66.35.250.150	80	->	MY.NET.100.203	32771
12/02-121730.601569	66.35.250.150	80	->	MY.NET.100.203	32771
12/02-121730.602335	66.35.250.150	80	->	MY.NET.100.203	32771
12/02-180117.954955	216.109.118.68	80	->	MY.NET.97.159	32771
12/02-180118.098615	216.109.118.68	80	->	MY.NET.97.159	32771

12/02-180118.099025	216.109.118.68	80	->	MY.NET.97.159	32771

These detects may be nothing more than false positives where the internal host made a http or https request to legitimate web servers from their ephemeral ports and the web server responded back via port 80 and 443. I would still recommend to the University to inspect all machines associated with this vulnerability.

Inspecting the scan logs for 12/02 and destination port 32771 reveals the following

Date	hou	min	sec	src	srcport	dir	dst	dstport	flags
2	20	15	27	130.85.110.72	12203	->	4.33.164.104	32771	UDP

There were 2,092 of these scans from this internal host to the 4.33.164.104 server. The time frame corresponds to the first 1802 alerts above. These scans appear to be crafted packets because the source port remains the same for all 2,092 scans.

Correlation

The question here is “Why is the protected host scanning an external server?”. In researching the 4.33.164.104 machine the following information was provided by DShield

IP Address 4.33.164.104

HostName crtntx1-ar12-4-33-164-104.crtntx1.dsl-verizon.net

DShield Profile

Country	US
Contact E-mail	abuse@genuity.com
AS Number	0
Total Records against IP	not processed
Number of targets	select update below
Date Range	to

TechHandle VOH1-ARIN
TechName Hostmaster, Verizon Online
TechPhone +1-800-927-3000
TechEmail hostmaster@bizmailsrvcs.net

Recommendations

The 130.85.110.72 needs to be inspected for compromise by The University’s security group.

The below listed detect was part of the output generated by searching for port 32771 in the scan log for 12/02. This is probably a false alarm.

Date	hou	min	sec	src	srcport	dir	dst	dstport	flags
2	20	16	7	130.85.97.62	6112	->	211.217.245.233	32771	UDP

Top Alerter for 12/03

The top alerter on 12/03 was 202.5.152.235. Querying the alert database for that day returns some interesting information for which the University need be concerned. The table below shows attempts on port 21 of the protected segments. The attempts began with MY.NET.1.1 and ended with MY.NET.136.2. Approximately 8,347 attempts were made on the control ports of any ftp servers behind the protected segments. What is shown here is the beginning and ending scans. The scans in between have been snipped.

Field1	Date/Time	src	srcport	Field	dst	dstport
9697	12/03-033307	202.5.152.235	21	->	MY.NET.1.1	21
18043	12/03-034436	202.5.152.235	21	->	MY.NET.136.29	21

The following alert is associated with the above information

SYN-FIN scan!

This represents an attempt by the source to map internal ftp servers by bypassing a stateless firewall or router. The question is whether any of the internal servers answered this scan. Searching the scan, alert and oos files for internal sources of MY.NET.0.0/16 to destination port 21 returns no responses back to 202.5.152.235.
The time frame of the syn-fin scan lasted for 11 minutes. All 8,347 scans occurred between 0333 and 0345 on 12/03. Searching the oos and scan files on 12/03 for activity during this time frame resulted in the following

dat	ho	mi	src	srcpo	dir	dst	dst	flags	flags2
3	33	7	202.5.152.235	21	->	130.85.1.1	21	SYNFIN	******SF
3	44	36	202.5.152.235	21	->	130.85.136.29	21	SYNFIN	******SF

8,448 scans are seen in the scan log for 12/03 and the time frame is as expected. The scanner ranged from MY.NET.1.1 to MY.NET.136.29. What is shown here is the beginning and ending scans. The scans in between have been snipped.

The OOS log files show no entries associated with the above source IP.

Recommendations

Someone is looking to see if port 21 is open. This may be recon for an attack.
This should probably be considered a hostile scan and action need be taken. The action could be that the source of the scan is blocked at the perimeter.

Correlation

As can be seen by the information below this source is known to DShield as an attacker.

IP Address 202.5.152.235

HostName 202.5.152.235

DShield Profile

Country	PK
Contact E-mail	ateeqk@hotmail.com
AS Number	0

Total Records against IP	512
Number of targets	512
Date Range	2003-12-17 to 2003-12-17

Top 10 Ports hit by this source

Port	Attacks	Start	End
21	22	2003-12-17	2003-12-17

Top Alerter for 12/04

The top alerter for 12/04 was 68.33.138.193. Examining the log files for 12/04 and source or destination as 68.33.138.193 reveals the following information

Field2	src	srcport	Fiel	dst	dstport
12/04-070547.787585	68.33.138.193	26783	->	MY.NET.30.4	80
12/04-070532.514029	68.33.138.193	26781	->	MY.NET.30.4	80

51 entries directed at port 80 of the protected segment.

Field2	src	srcport	Fiel	dst	dstport
12/04-195941.422837	68.33.138.193	27374	->	MY.NET.24.74	443
12/04-195941.513470	68.33.138.193	27374	->	MY.NET.24.74	443
12/04-195941.526879	68.33.138.193	27374	->	MY.NET.24.74	443
12/04-195941.489321	MY.NET.24.74	443	->	68.33.138.193	27374
12/04-195941.423333	MY.NET.24.74	443	->	68.33.138.193	27374
12/04-195941.489330	MY.NET.24.74	443	->	68.33.138.193	27374
12/04-195941.455876	MY.NET.24.74	443	->	68.33.138.193	27374
12/04-195941.422851	MY.NET.24.74	443	->	68.33.138.193	27374
12/04-195941.526908	MY.NET.24.74	443	->	68.33.138.193	27374

SubSeven activity.

Field2	src	srcport	Fiel	dst	dstport
12/04-070544.168372	68.33.138.193	26785	->	MY.NET.30.4	51443
12/04-070544.187853	68.33.138.193	26785	->	MY.NET.30.4	51443
12/04-070546.408024	68.33.138.193	26786	->	MY.NET.30.4	51443
12/04-070546.416250	68.33.138.193	26786	->	MY.NET.30.4	51443
12/04-070546.523522	68.33.138.193	26786	->	MY.NET.30.4	51443

<Snipped>

Field2	src	srcport	Fiel	dst	dstport
12/04-070801.508045	68.33.138.193	26898	->	MY.NET.30.4	51443
12/04-070801.501616	68.33.138.193	26898	->	MY.NET.30.4	51443
12/04-070801.417232	68.33.138.193	26898	->	MY.NET.30.4	51443
12/04-070801.363171	68.33.138.193	26898	->	MY.NET.30.4	51443
12/04-070801.423501	68.33.138.193	26898	->	MY.NET.30.4	51443

1117 entries with these source and destination hosts and destination port. The source port ranged from 26785 to 26898.

1178 log entries associated with port 80, 443, 27374 and 51443.
High volume of traffic from 0705 to 0708 directed at port 51443 of MY.NET.30.4.
Searching the snort output for 12/04 and 68.33.138.193 shows the following snort alert

Possible trojan server activity

Searching google for port 51443 directed me to the Novell web site at URL:http//www.novell.com/coolsolutions/netware/features/a_ifolder_21_protected_nw.html.
The information provided there references ifolder, a Novell product. Evidently ifolder provides secure services for users to connect via ssl. Based on the aforementioned logs the host at MY.NET.30.4 is probably a Novell server and the client, 68.33.138.193, is simply accessing those services.
A correlating document was also found through google and is part of a SANs practical URL:http//www.giac.org/practical/GCIA/Loic_Juillard_GCIA.pdf
In the content of this practical Juillard Loic speaks of port 51443 and states, "A user typically accesses to port 80 or 8009 and is redirected to port 51443 for web authentication and access to the files."

Indeed if we look at the initial output from the alert log files we do see the source connect to MY.NET.30.4 on port 80. From there we see the traffic directed at port 51443.

Field2	src	srcport	Fiel	dst	dstport
12/04-070547.787585	68.33.138.193	26783	->	MY.NET.30.4	80
12/04-070532.514029	68.33.138.193	26781	->	MY.NET.30.4	80

In trying to correlate the alert activity associated with this alert, I queried the scan and oos logs for 12/04 with the following output from the oos logs.

Field	Date/Time	src	srcport	dir	dst	dstport
1336	10/27-143052.509252	129.25.25.142	54401	->	MY.NET.30.4	80

It appears that the MY.NET.30.4 server is a web server of sorts. Just for grins, I decided to point my browser at URL:http//130.85.30.4 What I discovered was the **Welcome to NetWare 6 at UMBC** website.

Recommendations

From all indications it appears that this alert is a false positive concerning port 54431 and port 80. The initial query of the alert logs did show activity concerning port 27374, the SubSeven trojan port, directed at another protected host, MY.NET.24.74, shown below.

Field2	src	srcport	Fiel	dst	dstport
12/04-195941.422837	68.33.138.193	27374	->	MY.NET.24.74	443
12/04-195941.513470	68.33.138.193	27374	->	MY.NET.24.74	443
12/04-195941.526879	68.33.138.193	27374	->	MY.NET.24.74	443
12/04-195941.489321	MY.NET.24.74	443	->	68.33.138.193	27374
12/04-195941.423333	MY.NET.24.74	443	->	68.33.138.193	27374
12/04-195941.489330	MY.NET.24.74	443	->	68.33.138.193	27374
12/04-195941.455876	MY.NET.24.74	443	->	68.33.138.193	27374

12/04-195941.422851	MY.NET.24.74	443	->	68.33.138.193	27374
12/04-195941.526908	MY.NET.24.74	443	->	68.33.138.193	27374

The obvious recommendation here is that the ssl server at MY.NET.24.74 needs to be inspected and scanned for viruses, trojans and compromise.

Top Alerter for 12/05

The top alerter for 12/05 was 68.50.114.89. Examining the alert log files for 12/05 and source or destination as 68.50.114.89 reveals 16,104 entries with 68.50.114.89 as the source and destination ports of 80, 524 and 8009. This activity can be seen in the following output

Field2	src	srcport	Fiel	dst	dstport
12/05-	68.50.114.89	2793	->	MY.NET.30.3	80
12/05-	68.50.114.89	2793	->	MY.NET.30.3	80
12/05-	68.50.114.89	2793	->	MY.NET.30.3	80
12/05-	68.50.114.89	2794	->	MY.NET.30.3	80
12/05-	68.50.114.89	2794	->	MY.NET.30.3	80
12/05-	68.50.114.89	2794	->	MY.NET.30.3	80
12/05-	68.50.114.89	2794	->	MY.NET.30.3	80
12/05-	68.50.114.89	2793	->	MY.NET.30.3	80

Field2	src	srcport	Fiel	dst	dstport
12/05-180243.073007	68.50.114.89	2728	->	MY.NET.30.3	524
12/05-180243.133479	68.50.114.89	2728	->	MY.NET.30.3	524

<snipped>

Field2	src	srcport	Fiel	dst	dstport
12/05-231427.890021	68.50.114.89	2729	->	MY.NET.30.4	524
12/05-233404.972298	68.50.114.89	2729	->	MY.NET.30.4	524

2,164 of the above entries directed at MY.NET.30.3 and .4.

Field2	src	srcport	Fiel	dst	dstport
12/05-180738.452047	68.50.114.89	2795	->	MY.NET.30.3	8009
12/05-180738.507659	68.50.114.89	2795	->	MY.NET.30.3	8009

<snipped>

Field2	src	srcport	Fiel	dst	dstport
12/05-175458.842313	68.50.114.89	2604	->	MY.NET.30.4	8009
12/05-175500.128994	68.50.114.89	2604	->	MY.NET.30.4	8009

13,930 of the above entries directed at MY.NET.30.3 and .4.

Researching the snort_sort.pl html output of the alert logs for 12/05 reveals the following

MY.NET.30.3 activity

This is a custom snort alert setup by The University to monitor traffic on their internal servers.

In searching google for “Port 8009” information was returned which related port 8009 to the Apache Tomcat server. Evidently, Tomcat runs locally on port 8009 and as clients make connection to a web server on port 80, Tomcat redirects itself to port 8009. If that port is left open at the firewall then a compromise or denial of service is possible.

Top Alerter for 12/06

The top alerter for 12/06 was 68.57.90.146. Examining the alert log files for 12/06 and source or destination as 68.57.90.146 reveals 1340 entries with 68.57.90.146 as the source and destination ports of 524. This activity can be seen in the following output

Date/Time	Source	sport	Field5	Destination	dport
12/06-094836.69	68.57.90.146	1033	->	MY.NET.30.3	524
12/06-092853.92	68.57.90.146	1033	->	MY.NET.30.3	524

<snipped>

Date/Time	Source	sport	Field5	Destination	dport
12/06-104625.49	68.57.90.146	1033	->	MY.NET.30.3	524
12/06-104627.51	68.57.90.146	1033	->	MY.NET.30.3	524

795 alert log entries with 68.57.90.146 as the source and source port 1033.

Date/Time	Source	sport	Field5	Destination	dport
12/06-083528.265411	68.57.90.146	3242	->	MY.NET.30.4	524
12/06-083651.769531	68.57.90.146	3242	->	MY.NET.30.4	524
12/06-083528.400921	68.57.90.146	3242	->	MY.NET.30.4	524
12/06-083651.733066	68.57.90.146	3242	->	MY.NET.30.4	524
12/06-083528.422609	68.57.90.146	3242	->	MY.NET.30.4	524
12/06-083651.750990	68.57.90.146	3242	->	MY.NET.30.4	524
12/06-083528.577429	68.57.90.146	3242	->	MY.NET.30.4	524
12/06-083528.533289	68.57.90.146	3242	->	MY.NET.30.4	524
12/06-083528.444970	68.57.90.146	3242	->	MY.NET.30.4	524
12/06-083528.555297	68.57.90.146	3242	->	MY.NET.30.4	524

10 entries with 3242 as the source port.

Date/Time	Source	sport	Field5	Destination	dport
12/06-182611.835849	68.57.90.14	3645	->	MY.NET.30.3	524
12/06-182611.817533	68.57.90.14	3645	->	MY.NET.30.3	524

<snipped>

Date/Time	Source	sport	Field5	Destination	dport
12/06-203955.082332	68.57.90.14	3961	->	MY.NET.30.3	524
12/06-203944.556684	68.57.90.14	3961	->	MY.NET.30.3	524

535 entries as above directed at MY.NET.30.3 port 524.

Again, searching the alert html output for 12/06 reveals the following snort alert

MY.NET.30.3 activity

Evidently, the UMBC has set up a snort alert triggering when activity is seen on the MY.NET.30.3 and MY.NET.30.4 servers. This activity is then logged.

Checking the scan and oos logs for activity associated with this hosts reveals no information associated with 68.57.90.146.

Correlations

In researching port 524 the following information was found at URL:**http//razor.bindview.com/publish/advisories/adv_novellleak.html**

> Due to a combination of legacy support and default settings, Novell Netware servers using native IP will leak system information via TCP port 524 when properly queried. In mixed Novell/Microsoft environments, information regarding Microsoft devices is leaked via the Service Advertising Protocol (SAP) table. Third party products, such as those used to synchronize directory services between environments can further the problem. Essentially, a remote attacker can gather the equivalent information provided by the console command "display servers" and the DOS client command "cx /t /a /r" without authentication.
> All Novell Netware servers running IP (with port 524 open) can be queried and all objects with Public read access can be enumerated. Information such as account names, server services, and other various objects can be gathered. In mixed environments, such as environments with a mixture of IPX and IP, some IPX objects that are managed and communicate with IP-based servers can be leaked, and in an environment with Microsoft NT some NT objects can be leaked. The information gathered could be used to enumerate user account names and technologies deployed, which could be used for a future attack.
> The scope of the impact is limited to the internal network unless TCP port 524 access is allowed through a firewall or from dialup technologies deployed internally.

Recommendations

Investigate the MY.NET.30.3 and .4 servers for compromise. Make sure port 524 is blocked at the firewall or router.

Internal Source Alerts

So far we have identified the external top alerter for each day and looked at the associated traffic. I will now concentrate on those events related to the protected network which show signs of compromise or hostile activity.
Most of the scans and alerts coming from outside the network will be inspected, dropped and identified at the border. The internally generated alerts and scans are more indicative of problems that The University will need to address.

The following table from the alert files represents the Top 10 internal hosts generating snort alerts for each day of the 5 day audit period

12/02 src mynet	hits	12/03 src mynet	12/03 hits	12/04 src mynet	12/04 hits	12/05 src mynet	12/05 hits	12/06 src mynet	12/06 hits
MY.NET.162.41	1389	MY.NET.162.41	3026	MY.NET.162.41	3014	MY.NET.162.41	3259	MY.NET.162.41	3353
MY.NET.21.92	392	MY.NET.11.6	319	MY.NET.11.6	335	MY.NET.42.1	2017	MY.NET.21.92	1700
MY.NET.21.79	354	MY.NET.21.69	278	MY.NET.21.92	262	MY.NET.11.6	353	MY.NET.21.67	1577
MY.NET.21.67	350	MY.NET.21.67	262	MY.NET.21.79	261	MY.NET.12.4	219	MY.NET.21.68	1500
MY.NET.21.69	298	MY.NET.111.179	227	MY.NET.21.67	243	MY.NET.21.92	92	MY.NET.21.69	1268
MY.NET.21.68	280	MY.NET.21.68	218	MY.NET.21.68	203	MY.NET.21.69	79	MY.NET.21.79	1235
MY.NET.111.179	191	MY.NET.21.92	196	MY.NET.21.69	202	MY.NET.21.68	76	MY.NET.11.6	396

MY.NET.11.6	129	MY.NET.21.79	176	MY.NET.70.176	117	MY.NET.21.67	61	MY.NET.75.13	89
MY.NET.24.34	76	MY.NET.75.13	59	MY.NET.24.44	92	MY.NET.150.44	53	MY.NET.163.76	55
MY.NET.42.4	63	MY.NET.150.198	57	MY.NET.98.100	83	MY.NET.150.198	52	MY.NET.70.164	31

As can be seen from the above chart the MY.NET.162.41 was by far the leading alert generator for all 5 days. Let's take a look at this servers activity for all 5 days.

#Entries	Date/Time	src	srcport	Field5	dst	dstport
1039	12/02-000455.442047	MY.NET.162.41	721	->	128.183.110.242	515
2909	12/03-000037.605808	MY.NET.162.41	721	->	128.183.110.242	515
1915	12/04-000044.106883	MY.NET.162.41	721	->	128.183.110.242	515
3255	12/05-000008.742255	MY.NET.162.41	721	->	128.183.110.242	515
3353	12/06-000026.875695	MY.NET.162.41	721	->	128.183.110.242	515

We can see that the majority of this host's activity was directed at the 128.183.110.242 host on port 515. The "#Entries" column displays the entries in the alert log files for each day associated with port 515. The source and destination ports do not change. The alert associated with this host is as follows

connect to 515 from inside

Port 515 is typically associated with a print server and port 721 is identified in the IANA located at URL:http//www.iana.org/assignments/port-numbers as unassigned. The print server located at 128.183.110.242 corresponds to NASA according to the whois database at URL:http//swhois.net

Correlations

In the practical located at URL:http//www.giac.org/practical/GCIA/John_Hally_GCIA.pdf, Mr. Hally addresses the issue of the code red worm. In he states, "Port 515 is typically used for print services, and I believe that this is the case here. There are a few worms out there, Code Red in particular, that will scan for active print servers and try to exploit them in order to infect."
His conclusion is that the traffic is probably legitimate on the fact that the traffic shows a consistent pattern of following the school day's activity.

Inspecting the time frame of the above alerts gives the following information for 12/02

Date/Time	src	srcport	Field5	dst	dstport
12/02-104252.716419	MY.NET.162.41	721	->	128.183.110.242	515
12/02-104753.166426	MY.NET.162.41	721	->	128.183.110.242	515

<snipped>

Date/Time	src	srcport	Field5	dst	dstport
12/02-234720.414262	MY.NET.162.41	721	->	128.183.110.242	515
12/02-234732.432172	MY.NET.162.41	721	->	128.183.110.242	515

The pattern holds true for 12/02 but changes for the next 4 days as shown below

Date/Time	src	srcport	Field5	dst	dstport
12/06-000026.875695	MY.NET.162.41	721	->	128.183.110.242	515
12/06-000032.884667	MY.NET.162.41	721	->	128.183.110.242	515

<snipped>

Date/Time	src	srcport	Field5	dst	dstport

12/06-234251.278909	MY.NET.162.41	721	->	128.183.110.242	515
12/06-234736.705463	MY.NET.162.41	721	->	128.183.110.242	515

The pattern referenced by Mr. Hally is seen on 12/02 but changes for the next 4 days. The activity associated with port 515 is seen for all 24 hours and does not correspond to a normal school day's printing activity.

Recommendations
Inspect the MY.NET.162.41 for worms, viruses and Trojans.

One item identified in the alert logs was the Trojan server alerts within all 5 alert files. The following trojan servers were identified

Source
MY.NET.100.165
MY.NET.12.4
MY.NET.12.6
MY.NET.12.7
MY.NET.24.33
MY.NET.24.34
MY.NET.24.44
MY.NET.24.74
MY.NET.25.21
MY.NET.5.20
MY.NET.6.7
MY.NET.83.83

The snort alert generated was

Possible trojan server activity

438 alerts were generated in the 5 day period beginning December 2nd, 2003. The source port for these alerts originated from ports 25, 80, 110, 143, 443, 995 and 8379.

The destination hosts and port were garnered in a similar manner and the information is provided below

Destination port 27374 --- SubSeven Trojan port
Destination hosts

dest			
12.107.80.100	162.129.227.25	209.68.148.251	66.218.66.65
12.159.180.253	198.200.181.20	213.249.187.33	68.32.120.134
12.47.98.180	199.173.224.2	216.136.227.10	68.32.123.20
129.41.62.88	200.106.9.70	216.64.222.155	68.33.113.23
130.184.5.31	204.29.185.182	216.99.185.50	68.33.138.193
134.113.4.207	205.170.0.236	62.68.228.160	68.49.56.175
138.78.20.8	205.222.240.2	63.174.69.230	68.55.192.50
143.83.127.196	208.161.242.74	64.95.24.245	68.55.195.148
148.129.74.40	208.199.82.216	65.167.144.157	69.139.225.163
	209.165.168.2	66.186.131.185	80.15.127.74

The universities network and security admins need to be aware of the following trojans associated with port 27374 as referenced from Dshield at URL:http//www.dshield.org

Protocol Service Name
tcp SubSeven [trojan] SubSeven
tcp SubSeven [trojan] SubSeven
tcp BadBlood [trojan] Bad Blood
tcp EGO [trojan] EGO
tcp FakeSubSeven [trojan] Fake SubSeven
tcp Lion [trojan] Lion
tcp Ramen [trojan] Ramen
tcp Seeker [trojan] Seeker
tcp Subseven2.1.4DefCon8 [trojan] Subseven 2.1.4 DefCon 8
tcp SubSeven2.1Gold [trojan] SubSeven 2.1 Gold
tcp SubSeven2.2 [trojan] SubSeven 2.2
tcp SubSevenMuie [trojan] SubSeven Muie
tcp TheSaint [trojan] The Saint
tcp Ttfloader [trojan] Ttfloader
tcp Webhead [trojan] Webhead

The following output represents the MY.NET.12.6 mail server working with the 216.64.222.155 host on the SubSeven port.

Info	Info1	Info2	Source	sport	dir	Destination	dport
Possible	trojan	server	MY.NET.12.6	25	->	216.64.222.155	27374
Possible	trojan	server	216.64.222.155	27374	->	MY.NET.12.6	25
Possible	trojan	server	216.64.222.155	27374	->	MY.NET.12.6	25
Possible	trojan	server	216.64.222.155	27374	->	MY.NET.12.6	25
Possible	trojan	server	MY.NET.12.6	25	->	216.64.222.155	27374
Possible	trojan	server	216.64.222.155	27374	->	MY.NET.12.6	25
Possible	trojan	server	MY.NET.12.6	25	->	216.64.222.155	27374
Possible	trojan	server	MY.NET.12.6	25	->	216.64.222.155	27374
Possible	trojan	server	216.64.222.155	27374	->	MY.NET.12.6	25
Possible	trojan	server	216.64.222.155	27374	->	MY.NET.12.6	25
Possible	trojan	server	216.64.222.155	27374	->	MY.NET.12.6	25
Possible	trojan	server	216.64.222.155	27374	->	MY.NET.12.6	25
Possible	trojan	server	216.64.222.155	27374	->	MY.NET.12.6	25
Possible	trojan	server	MY.NET.12.6	25	->	216.64.222.155	27374
Possible	trojan	server	216.64.222.155	27374	->	MY.NET.12.6	25
Possible	trojan	server	216.64.222.155	27374	->	MY.NET.12.6	25
Possible	trojan	server	216.64.222.155	27374	->	MY.NET.12.6	25
Possible	trojan	server	216.64.222.155	27374	->	MY.NET.12.6	25
Possible	trojan	server	MY.NET.12.6	25	->	216.64.222.155	27374
Possible	trojan	server	216.64.222.155	27374	->	MY.NET.12.6	25
Possible	trojan	server	216.64.222.155	27374	->	MY.NET.12.6	25
Possible	trojan	server	216.64.222.155	27374	->	MY.NET.12.6	25
Possible	trojan	server	MY.NET.12.6	25	->	216.64.222.155	27374

Correlations
In researching the above traffic the following information was garnered from SANS at this URL:
http//www.sans.org/resources/idfaq/subseven.php

"SubSeven is a trojan for the windows platform. It comes at least in two parts a client and a server. The client is used by the hacker to connect to the victim' s machine. Once the server.exe is installed on the victim's machine the hacker has full access to the victim's machine.'

Known Information about SubSeven Known TCP ports for SubSeven
1243
6711
6712
6713
6776
Known TCP ports for SubSeven 2.1
27374

Recommendations
The University's network and security admins will need to examine each machine listed above for virus, worm and/or Trojan infection.

The next alert indicating trojan activity which the university need be aware is as follows

[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.

Source			
12.178.196.125	206.252.192.194	217.8.38.20	66.252.10.217
128.193.0.30	208.185.81.227	24.229.1.18	66.252.11.105
129.27.9.248	208.185.81.251	38.117.33.163	66.252.13.40
130.233.48.242	209.151.249.50	61.6.39.100	66.40.25.214
193.110.95.1	209.47.9.196	62.235.13.228	66.90.81.227
195.47.220.2	209.67.60.33	63.102.226.240	69.36.232.118
199.184.165.133	216.152.64.155	64.12.165.56	69.42.74.6
202.91.34.9	216.194.70.11	64.124.0.204	69.50.177.102
203.167.224.18	216.194.70.8	64.36.90.138	82.96.64.2
204.152.184.80	216.194.70.9	64.71.177.228	
204.91.240.100	216.32.207.207	66.132.147.58	
205.177.13.100	216.82.127.46	66.150.99.99	
	217.160.142.142	66.197.0.145	

Source ports associated with the above hosts are 6665, 6667, 6669 and 7000.
The destinations within the universities protected segments are as follows

dest			
MY.NET.100.75	MY.NET.153.90	MY.NET.42.10	MY.NET.42.7
MY.NET.15.71	MY.NET.21.69	MY.NET.42.12	MY.NET.42.8
MY.NET.152.251	MY.NET.24.10	MY.NET.42.3	MY.NET.53.139
	MY.NET.42.1	MY.NET.42.4	MY.NET.60.11

MY.NET.60.16	MY.NET.97.134	MY.NET.97.21	MY.NET.97.67
MY.NET.60.38	MY.NET.97.135	MY.NET.97.22	MY.NET.97.70
MY.NET.60.39	MY.NET.97.141	MY.NET.97.229	MY.NET.98.51
MY.NET.60.40	MY.NET.97.160	MY.NET.97.232	MY.NET.98.55
MY.NET.70.159	MY.NET.97.171	MY.NET.97.246	MY.NET.98.71
MY.NET.75.134	MY.NET.97.172	MY.NET.97.29	MY.NET.99.51
MY.NET.82.96	MY.NET.97.182	MY.NET.97.34	
MY.NET.97.10	MY.NET.97.197	MY.NET.97.42	
MY.NET.97.101	MY.NET.97.202	MY.NET.97.57	
MY.NET.97.12	MY.NET.97.208	MY.NET.97.63	

Correlations

In searching google for information regarding this particular alert information was provided concerning the above listed ports.

Servers currently linked to the Mysteria IRC Network

Pacific.CA.US.Mysteria.Net Fremont, California, US
Connection OC-3
Ports 6665,6666,6667,6668,6669,6670,7000
Tiamat.FL.US.Mysteria.Net Tampa, Florida, US
Connection T3
Ports 6665,6666,6667,6668,6669,6670,7000
Xanth.GA.US.Mysteria.Net Atlanta, Georgia, US
Connection DS-3
Ports 6665,6666,6667,6668,6669,6670,7000

Recommendations

What The University is seeing here is Internet Relay Chat ingressing and egressing it's networks. To control this activity, I would suggest placing an IRC server in your DMZ, blocking IRC internally and instructing your clients to access the DMZ IRC server for communications to other IRC communities.

Another finding in this analysis is that the university has potentially compromised systems as exampled by the following snort alert

High port 65535 udp - possible Red Worm – traffic
High port 65535 tcp - possible Red Worm – traffic

48 different source hosts are involved with this worm activity. 930 of the source ports were 65535.

source			
MY.NET.100.13	MY.NET.12.6	MY.NET.24.27	MY.NET.24.74
MY.NET.100.165	MY.NET.152.182	MY.NET.24.34	MY.NET.25.11
MY.NET.100.230	MY.NET.152.19	MY.NET.24.35	MY.NET.25.12
MY.NET.12.4	MY.NET.163.76	MY.NET.24.44	MY.NET.25.66
	MY.NET.24.20	MY.NET.24.48	MY.NET.25.67

MY.NET.25.68	MY.NET.5.20	MY.NET.83.83	MY.NET.97.50
MY.NET.25.69	MY.NET.53.50	MY.NET.87.225	MY.NET.97.67
MY.NET.25.70	MY.NET.6.7	MY.NET.97.13	MY.NET.97.90
MY.NET.25.71	MY.NET.60.11	MY.NET.97.14	MY.NET.98.100
MY.NET.25.72	MY.NET.60.17	MY.NET.97.144	MY.NET.98.12
MY.NET.25.73	MY.NET.70.164	MY.NET.97.181	
MY.NET.29.3	MY.NET.70.176	MY.NET.97.209	
MY.NET.42.6	MY.NET.75.27	MY.NET.97.21	

71 different destination hosts on the protected network were involved with this scan activity. 954 of the destination ports were port 65535.

dest			
MY.NET.1.3	MY.NET.163.76	MY.NET.42.19	MY.NET.70.185
MY.NET.100.16	MY.NET.190.16	MY.NET.42.4	MY.NET.71.248
MY.NET.110.21	MY.NET.24.10	MY.NET.42.5	MY.NET.75.27
MY.NET.111.13	MY.NET.24.20	MY.NET.42.6	MY.NET.80.148
MY.NET.112.17	MY.NET.24.27	MY.NET.5.20	MY.NET.81.112
MY.NET.12.4	MY.NET.24.34	MY.NET.53.147	MY.NET.83.109
MY.NET.12.6	MY.NET.24.35	MY.NET.53.50	MY.NET.83.83
MY.NET.130.15	MY.NET.24.44	MY.NET.53.51	MY.NET.83.98
MY.NET.150.13	MY.NET.24.48	MY.NET.53.56	MY.NET.84.234
MY.NET.152.17	MY.NET.24.74	MY.NET.53.57	MY.NET.97.13
MY.NET.152.18	MY.NET.25.10	MY.NET.53.59	MY.NET.97.14
MY.NET.152.18	MY.NET.25.11	MY.NET.6.49	MY.NET.97.181
MY.NET.152.24	MY.NET.25.66	MY.NET.6.62	MY.NET.97.50
MY.NET.152.24	MY.NET.25.68	MY.NET.6.7	MY.NET.98.100
MY.NET.152.25	MY.NET.25.69	MY.NET.60.11	MY.NET.98.12
MY.NET.153.15	MY.NET.25.70	MY.NET.60.17	
MY.NET.153.16	MY.NET.25.72	MY.NET.66.29	
MY.NET.153.99	MY.NET.25.73	MY.NET.70.164	
	MY.NET.29.3	MY.NET.70.176	

Correlations:

According to the practical written by Joe Bowling located at the following URL: http://www.giac.org/practical/GCIA/Joe_Bowling_GCIA.pdf

"Red Worm aka Adore worm works similar to the Lion and Ramen Worms. The Worm attacks Linux boxes that have LPRng, rpc-statd, wu-ftpd and BIND services running. Adore installs a Trojan that upon activation (an icmp packet of a certain length it would then open a root shell to allow a remote user to connect. Adore also scans random B class networks looking for other host to infect."

Recommendations:

Patches have been developed for this problem.
The URL:http://www.sans.org/y2k/adore.htm has more information on this issue.

The University's admin people will use the above list to identify their infected hosts and patch accordingly. A utility exists to clean this problem and can be found at the following URL:
http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm

Another indication that The University should be concerned is the following snort alert log entry:

[UMBC NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC

The following sdbot alerts were pulled from the alert logs:

Field7	Field8	Field10	Fi	Field	source	sport	Fiel	dst	dport
sdbot	floodnet	attempting	to	IRC	MY.NET.97.10	3168	->	216.152.64.155	6665
sdbot	floodnet	attempting	to	IRC	MY.NET.97.10	3452	->	216.152.64.155	6666
sdbot	floodnet	attempting	to	IRC	MY.NET.97.10	3890	->	216.152.64.155	6665
sdbot	floodnet	attempting	to	IRC	MY.NET.97.10	3962	->	216.152.64.155	6665
sdbot	floodnet	attempting	to	IRC	MY.NET.97.10	3983	->	216.152.64.155	6665
sdbot	floodnet	attempting	to	IRC	MY.NET.97.10	4192	->	216.152.64.155	6666
sdbot	floodnet	attempting	to	IRC	MY.NET.97.10	4208	->	216.152.64.155	6666
sdbot	floodnet	attempting	to	IRC	MY.NET.97.10	4262	->	216.152.64.155	6665
sdbot	floodnet	attempting	to	IRC	MY.NET.97.10	4264	->	216.152.64.155	6666
sdbot	floodnet	attempting	to	IRC	MY.NET.97.120	1419	->	213.186.42.35	6667
sdbot	floodnet	attempting	to	IRC	MY.NET.97.93	2685	->	213.186.42.35	6667
sdbot	floodnet	attempting	to	IRC	MY.NET.53.59	3910	->	213.186.35.9	6667
sdbot	floodnet	attempting	to	IRC	MY.NET.53.59	4060	->	213.186.35.9	6667

Correlations:
This issue is similar to the previous IRC problem. In a practical written by Daniel Clark at the following URL:http://www.giac.org/practical/GCIA/Daniel_Clark_GCIA.pdf,
Mr. Clark speaks of IRC users issuing kill commands to other users and possible illegal file sharing activities. Indeed we see this type of activity is our alert logs where the snort alarm is "[UMBC NIDS IRC Alert] IRC user /kill detected, possible Trojan". The MY.NET.97.10 protected host can be identified in both alerts and should be thoroughly inspected..
Additional information is available at the following URL:
http://www.digitalirc.net/index.htm

Recommendations:
As stated previously, The University's admins will use the above list to inspect their clients for worms, Trojans and/or compromise. If possible, block IRC at the border, use an internal IRC server in your DMZ and make effective use of Network Address Translation.

The Out of Spec files show the following hosts seeing unusual tcp/ip traffic occurring on the universities internal networks:

Hosts	MY.NET.100.13	MY.NET.100.165	MY.NET.100.230

MY.NET.101.89	MY.NET.162.235	MY.NET.25.72	MY.NET.69.217
MY.NET.109.9	MY.NET.162.67	MY.NET.25.73	MY.NET.69.229
MY.NET.110.150	MY.NET.162.87	MY.NET.250.178	MY.NET.69.249
MY.NET.111.140	MY.NET.163.71	MY.NET.29.3	MY.NET.7.97
MY.NET.111.21	MY.NET.168.181	MY.NET.29.66	MY.NET.70.129
MY.NET.111.52	MY.NET.24.19	MY.NET.30.4	MY.NET.70.162
MY.NET.112.152	MY.NET.24.20	MY.NET.42.1	MY.NET.70.225
MY.NET.112.159	MY.NET.24.33	MY.NET.42.3	MY.NET.70.231
MY.NET.112.164	MY.NET.24.34	MY.NET.5.20	MY.NET.73.112
MY.NET.112.172	MY.NET.24.35	MY.NET.53.120	MY.NET.75.3
MY.NET.12.4	MY.NET.24.44	MY.NET.53.37	MY.NET.80.105
MY.NET.12.6	MY.NET.24.47	MY.NET.53.45	MY.NET.84.143
MY.NET.12.7	MY.NET.24.74	MY.NET.53.54	MY.NET.84.186
MY.NET.149.231	MY.NET.25.10	MY.NET.55.97	MY.NET.84.198
MY.NET.150.133	MY.NET.25.11	MY.NET.6.14	MY.NET.84.232
MY.NET.150.83	MY.NET.25.12	MY.NET.6.7	MY.NET.97.122
MY.NET.153.141	MY.NET.25.66	MY.NET.60.14	MY.NET.97.14
MY.NET.153.182	MY.NET.25.67	MY.NET.60.16	MY.NET.97.196
MY.NET.153.186	MY.NET.25.68	MY.NET.60.17	MY.NET.97.52
MY.NET.153.32	MY.NET.25.69	MY.NET.60.38	MY.NET.97.79
MY.NET.153.92	MY.NET.25.70	MY.NET.60.39	
MY.NET.153.94	MY.NET.25.71	MY.NET.69.181	

The ports associated with the above listed hosts are:

Ports	16959	2670	3654	466	54672
	1757	2671	3660	4662	54673
0	1758	2679	3847	4790	54674
1088	1764	2690	39824	4791	59190
110	1844	2734	42328	4798	59846
11272	1847	2740	4315	4799	61131
113	1996	28979	43749	4800	61170
1158	20255	3091	443	4802	6346
1214	2089	3146	4466	4806	64168
1225	21	3148	4589	4809	64970
1297	22	3150	45915	49227	65000
1336	2273	3151	4592	49281	6881
1341	2310	3227	4593	54666	6895
1365	2311	3244	4597	54667	80
143	25	33401	4598	54668	9291
1473	26368	33696	4601	54669	
1481	2664	3456	4602	54670	
1561	2669	3531	4605	54671	

The following table shows 4 internal hosts generating unusual traffic in the OOS files:

Date/Time	src	srcport	dir	dst	dstport
10/29-13:11:33.609276	MY.NET.12.2	25	->	172.131.11.154	1847

10/28-22:48:26.682354	MY.NET.12.4	143	->	216.133.76.11	49281
10/27-06:05:44.079272	MY.NET.12.4	110	->	66.91.177.102	1158
10/28-01:46:07.572997	MY.NET.12.4	143	->	68.55.203.157	26368
10/28-21:21:00.854267	MY.NET.12.4	993	->	68.55.111.126	2311
10/28-02:31:08.482976	MY.NET.12.4	143	->	66.93.54.236	59846
10/28-21:33:14.903154	MY.NET.12.4	143	->	151.196.170.179	61131
10/28-12:08:41.694205	MY.NET.12.4	143	->	68.55.114.159	64970
10/27-22:59:09.005840	MY.NET.12.4	993	->	68.48.12.185	28979
10/27-12:40:39.603159	MY.NET.12.4	143	->	68.32.126.162	49227
10/27-08:15:51.978619	MY.NET.12.4	143	->	68.55.160.245	3227
10/29-03:04:29.121265	MY.NET.12.4	143	->	MY.NET.250.178	3660
10/29-10:09:16.615537	MY.NET.12.4	993	->	66.93.118.119	59190
10/29-11:54:59.048717	MY.NET.12.4	143	->	MY.NET.168.181	64168
10/28-21:21:00.866310	MY.NET.12.4	993	->	68.55.111.126	2310
10/27-23:31:55.132885	MY.NET.12.6	25	->	161.58.201.9	61170
10/28-01:52:45.238528	MY.NET.12.6	25	->	161.58.155.247	33401
10/28-01:55:21.602662	MY.NET.12.6	25	->	161.58.155.247	33696
10/28-04:00:51.768531	MY.NET.12.6	25	->	161.58.155.247	45915
10/28-04:56:17.779156	MY.NET.12.6	25	->	12.224.87.84	20255
10/28-16:09:47.683107	MY.NET.12.6	25	->	128.175.32.87	16959
10/28-23:42:54.760192	MY.NET.12.6	25	->	209.76.40.140	3244
10/29-00:34:43.738406	MY.NET.12.6	25	->	218.79.232.167	1481
10/29-00:41:53.700316	MY.NET.12.6	25	->	161.58.155.247	42328
10/29-00:53:58.726302	MY.NET.12.6	25	->	161.58.155.247	43749
10/28-15:55:35.483842	MY.NET.12.7	443	->	64.76.49.153	1297

This traffic is a result of corrupted packets and/or failing network interfaces or devices. The number of Out of Spec alerts seen over the 5 day period in question is 45,985. The number of different source addresses in the OOS files is 480. The number of different source ports associated with the OOS files is 7546. Total number of different destination ip addresses within the OOS files is 108. The total number of different destination ports is 104.

Note the null port in the Out of Spec ports list. What we are seeing here is tcp/ip traffic where there is no source or destination port. This is not accepted as legitimate traffic on the network and is discarded.

Top 10 Lists:

The following table from the alert files represents the Top 10 internal hosts generating snort alerts for each day of the 5 day audit period:

12/02 src mynet	hits	12/03 src mynet	12/03 hits	12/04 src mynet	12/04 hits	12/05 src mynet	12/05 hits	12/06 src mynet	12/06 hits
MY.NET.162.41	1389	MY.NET.162.41	3026	MY.NET.162.41	3014	MY.NET.162.41	3259	MY.NET.162.41	3353
MY.NET.21.92	392	MY.NET.11.6	319	MY.NET.11.6	335	MY.NET.42.1	2017	MY.NET.21.92	1700
MY.NET.21.79	354	MY.NET.21.69	278	MY.NET.21.92	262	MY.NET.11.6	353	MY.NET.21.67	1577
MY.NET.21.67	350	MY.NET.21.67	262	MY.NET.21.79	261	MY.NET.12.4	219	MY.NET.21.68	1500
MY.NET.21.69	298	MY.NET.111.179	227	MY.NET.21.67	243	MY.NET.21.92	92	MY.NET.21.69	1268
MY.NET.21.68	280	MY.NET.21.68	218	MY.NET.21.68	203	MY.NET.21.69	79	MY.NET.21.79	1235
MY.NET.111.179	191	MY.NET.21.92	196	MY.NET.21.69	202	MY.NET.21.68	76	MY.NET.11.6	396
MY.NET.11.6	129	MY.NET.21.79	176	MY.NET.70.176	117	MY.NET.21.67	61	MY.NET.75.13	89

MY.NET.24.34	76	MY.NET.75.13	59	MY.NET.24.44	92	MY.NET.150.44	53	MY.NET.163.76	55
MY.NET.42.4	63	MY.NET.150.198	57	MY.NET.98.100	83	MY.NET.150.198	52	MY.NET.70.164	31

The following is a summary of the Top 10 external hosts which triggered snort alarms for each day on incoming traffic in the alert logs:

12/02 src alerts	12/02 src hits	12/03 src alerts	12/03 src hits	12/04 src alerts	12/04 src hits	12/05 src alerts	12/05 src hits	12/06 src alerts	12/06 src hits
68.3.197.224	1802	202.5.152.235	8347	68.33.138.193	1172	68.50.114.89	16104	68.57.90.146	1340
68.34.120.219	798	68.34.120.219	1085	68.32.127.158	768	67.21.63.15	5836	64.12.31.4	744
131.92.177.18	687	68.55.62.79	386	131.92.177.18	492	68.48.90.101	2501	68.55.62.79	402
68.55.62.79	202	68.55.195.133	211	68.55.62.79	340	68.32.122.89	1821	62.163.87.57	309
68.55.52.234	121	63.84.193.226	198	68.57.90.146	292	68.54.168.204	1368	68.50.114.89	155
68.55.53.222	108	195.217.253.40	164	128.153.198.225	177	68.57.90.146	857	68.55.144.24	140
68.55.27.157	96	66.196.72.50	124	66.196.72.23	144	68.55.85.180	438	68.55.27.157	120
66.196.72.17	77	68.55.53.222	113	68.55.27.157	143	68.55.62.79	360	165.247.85.240	111
64.242.195.86	72	66.196.72.53	112	63.251.52.75	143	68.55.27.157	346	165.247.95.11	104
66.196.72.15	58	66.196.72.14	96	66.196.72.29	135	68.55.113.194	295	151.196.165.180	102

The following represents the top 10 scanners during the audit period:

12/02 Scanner	12/02 hit	12/03 Scanner	12/03 hit	12/04 Scanner	12/04 hit	12/05 Scanner	12/05 hit	12/06 Scanner	12/05 hit
220.82.114.160	12727	69.11.233.139	21031	129.241.80.233	25690	194.125.56.130	18944	66.246.86.158	399268
195.116.220.68	10787	207.173.16.33	18904	211.235.32.13	20962	64.123.3.73	18582	134.88.49.226	20635
81.80.26.210	8441	129.27.3.14	18904	143.248.37.61	15943	67.120.144.225	18097	63.98.122.116	14932
80.117.195.211	7698	205.188.149.20	18904	131.94.10.184	7786	217.225.208.219	13502	203.251.21.98	14183
195.182.177.82	7165	195.54.102.4	18757	211.223.69.119	7735	193.198.32.154	12281	24.57.89.36	12158
219.65.71.21	6379	129.27.9.248	18702	211.253.213.56	7000	218.149.79.252	10224	219.148.237.1	11150
200.206.97.62	5154	207.96.122.250	18694	66.166.152.46	4626	217.56.8.98	10097	203.90.91.10	9084
218.146.246.37	4903	195.121.6.196	18686	80.181.214.39	4400	12.101.201.74	9297	80.136.249.23	8273
211.52.230.243	2999	140.99.102.4	18649	195.199.36.202	2271	219.97.92.211	7737	12.159.151.44	6971
81.255.61.67	199	193.110.95.1	18634	216.151.126.98	1584	169.207.179.51	7731	210.22.202.77	6644

The following represents the top 10 scanners over the 5 day period.

Scanner	Hits
66.246.86.158	399268
129.241.80.233	25690
69.11.233.139	21031
211.235.32.13	20962
134.88.49.226	20635
194.125.56.130	18944
205.188.149.20	18904
129.27.3.14	18904
207.173.16.33	18904
195.54.102.4	18757

The Top 10 destination hosts and hits for each day as seen from the scan logs is as follows:

12/02 dst	12/02 hits	12/03 dst	12/03 hits	12/04 dst	12/04 hits	12/05 dst	12/05 dst hits	12/06 dst	12/06 hits
192.26.92.30	10162	69.6.22.10	18749	69.6.22.10	15231	192.26.92.30	14665	192.26.92.30	14078
192.5.6.30	8310	69.6.22.11	18530	69.6.22.11	15106	69.6.22.11	13019	69.6.22.10	12154
192.55.83.30	7336	192.26.92.30	16675	192.26.92.30	14275	69.6.22.10	12654	69.6.22.11	12145
69.6.22.10	7122	192.5.6.30	13718	192.5.6.30	11457	192.5.6.30	12096	192.5.6.30	11108
69.6.22.11	6979	65.248.79.40	12978	65.248.79.40	11301	203.20.52.5	10536	192.55.83.30	9447
4.33.164.104	6641	192.55.83.30	11606	203.20.52.5	10285	192.55.83.30	10049	131.118.254.33	8364
69.6.25.125	6012	207.115.86.54	10944	192.55.83.30	9492	65.248.79.40	9671	203.20.52.5	8119
69.6.25.84	5940	203.20.52.5	10937	131.118.254.33	8709	131.118.254.3	9350	61.31.195.161	7696
131.118.254.33	5912	216.109.116.17	9312	216.109.116.17	8241	216.109.116.1	8520	216.109.116.17	7568

The Top 10 destination ports and hits for each day identified in the scan files were:

12/02 dport	12/02 hits	12/03 dport	12/03 hits	12/04 dport	12/04 hits	12/05 dports	12/05 dport	12/06 dports	12/06 hits
135	643245	135	1455177	135	1513525	135	1612613	135	1710803
53	537924	53	928583	53	815245	53	888920	53	746966
21	22527	80	79527	22321	293015	80	63932	80	60618
4000	17872	4000	34867	7674	75565	20168	51809	1257	34835
3389	12771	1257	21041	3389	25699	4000	30391	25	28626
80	11337	21	17211	23	20971	25	27534	4899	16230
6346	8739	554	12522	80	18478	3410	19008	554	14942
22321	8408	3389	12166	25	16266	22321	18976	22321	12768
1314	4464	5900	10499	554	15943	7070	18591	445	12156
32770	4011	4899	9422	21	11433	21	15988	6257	11423

The following is the top 10 destination ports as seen by the alert logs:

12/02 dports	12/02 hits	12/03 dports	12/03 hits	12/04 dports	12/04 hits	12/05 dports	12/05 hits	12/06 dports	12/06 hits
32771	1819	21	8361	515	3804	8009	13930	515	3352
515	1580	515	3251	80	1895	524	11001	524	2582
524	1338	80	2350	524	1561	51443	6169	80	1027
80	1136	51443	1303	51443	1118	515	3258	32771	849
51443	775	137	910	137	820	137	2678	137	694
137	591	524	629	65535	293	80	1120	65535	138
65535	152	25	322	0	238	65535	272	135	86
0	79	53	146	27374	132	110	196	445	84
53	62	135	126	25	131	53	117	53	75
25	58	32771	122	53	111	135	113	51443	63

The following is the total destination port hits as seen by the alert logs:

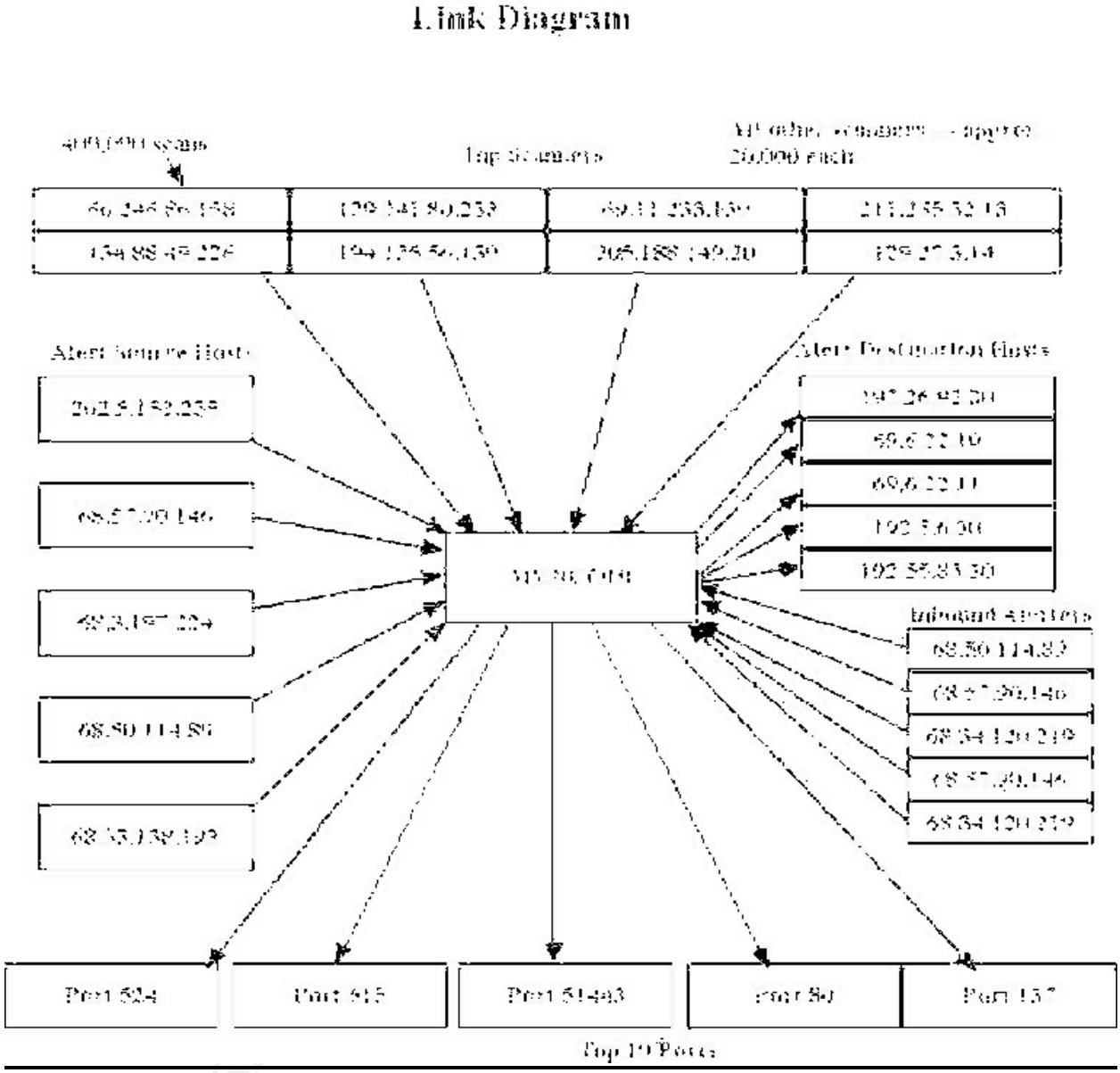
dport	5 day hits
524	17111
515	15245
51443	9428
80	7528
137	5693

32771	2790
65535	855
25	511
53	511
135	325
0	317

Top 10 Out of Spec entries:

oos src	oos hits	oos dst	oos dst hits
195.111.1.93	2446	MY.NET.12.6	2651
213.54.173.25	277	MY.NET.24.34	2528
158.196.149.6	264	MY.NET.24.44	1203
195.101.94.10	226	MY.NET.84.14	391
66.225.198.20	224	MY.NET.112.1	349
195.101.94.20	204	MY.NET.100.1	338
61.135.129.99	140	MY.NET.111.5	279
63.71.152.2	140	MY.NET.6.7	260
216.95.201.13	131	MY.NET.100.2	193
195.101.94.20	126	MY.NET.60.14	121

Link Diagram:



Registration Details for 5 External Hosts:

The below address was chosen because of it's activity in a hostile scan.

IP Address: 202.5.152.235

HostName: 202.5.152.235

DShield Profile:

Country:	PK
Contact E-mail:	ateeqk@hotmail.com
AS Number:	0
Total Records against IP:	512
Number of targets:	512
Date Range:	2003-12-17 to 2003-12-17

Top 10 Ports hit by this source:

Port	Attacks	Start	End
21	22	2003-12-17	2003-12-17

The below listed information was chosen because of its activity as an external target.
IP Address: 4.33.164.104
HostName: Crtntx1-ar12-4-33-164-104.crtntx1.dsl-verizon.net
DShield Profile:

Country:	US
Contact E-mail:	abuse@genuity.com
AS Number:	0
Total Records against IP:	not processed
Number of targets:	select update below
Date Range:	to

TechHandle: VOH1-ARIN
TechName: Hostmaster, Verizon Online
TechPhone: +1-800-927-3000
TechEmail: hostmaster@bizmailsrvcs.net

The below listed information was chosen because of its activity with port 515.

OrgName: National Aeronautics and Space Administration
OrgID: NASA
Address: AD33/Office of the Chief Information Officer
City: MSFC
StateProv: AL

PostalCode: 35812
Country: US

NetRange: 128.183.0.0 - 128.183.255.255
CIDR: 128.183.0.0/16
NetName: GSFC
NetHandle: NET-128-183-0-0-1
Parent: NET-128-0-0-0-0
NetType: Direct Allocation
NameServer: NS.GSFC.NASA.GOV
NameServer: NS2.GSFC.NASA.GOV
Comment:
RegDate: 1993-04-01
Updated: 2003-02-05

TechHandle: ZN7-ARIN
TechName: National Aeronautics and Space Administration
TechPhone: +1-256-544-5623
TechEmail: dns.support@nasa.gov

The below information was chosen because of its activity with SubSeven.

OrgName: Cable & Wireless
OrgID: EXCW
Address: 3300 Regency Pkwy
City: Cary
StateProv: NC
PostalCode: 27511
Country: US

ReferralServer: rwhois://rwhois.exodus.net:4321/

NetRange: 216.64.192.0 - 216.64.223.255
CIDR: 216.64.192.0/19
NetName: CH3-2
NetHandle: NET-216-64-192-0-1
Parent: NET-216-0-0-0-0
NetType: Direct Allocation
NameServer: DNS01.EXODUS.NET
NameServer: DNS02.EXODUS.NET
NameServer: DNS03.EXODUS.NET
NameServer: DNS04.EXODUS.NET
Comment: * Rwhois reassignment information for this block is available at:
Comment: * rwhois.exodus.net 4321
Comment: * For abuse please contact abuse@exodus.net
RegDate:

Updated: 2002-08-20

TechHandle: ZC221-ARIN
TechName: Cable & Wireless
TechPhone: +1-919-465-4023
TechEmail: ip@gnoc.cw.net

OrgAbuseHandle: ABUSE11-ARIN
OrgAbuseName: Abuse
OrgAbusePhone: +1-877-393-7878
OrgAbuseEmail: abuse@exodus.net

OrgNOCHandle: NOC99-ARIN
OrgNOCName: Network Operations Center
OrgNOCPhone: +1-800-977-4662
OrgNOCEmail: trouble@cw.net

OrgTechHandle: EIAA-ARIN
OrgTechName: Exodus IP Address Administration
OrgTechPhone: +1-888-239-6387
OrgTechEmail: ipaddressadmin@exodus.net

OrgTechHandle: GIAA-ARIN
OrgTechName: Global IP Address Administration
OrgTechPhone: +1-919-465-4096
OrgTechEmail: ip@gnoc.cw.net

The 66.246.86.158 address was chosen for registration because of it being identified as the top scanner.

Whois:

```
OrgName:    Net Access Corporation
OrgID:      NAC
Address:    1719 STE RT 10E
Address:    Suite 111
City:       Parsippany
StateProv:  NJ
PostalCode: 07054
Country:    US

NetRange:   66.246.0.0 - 66.246.191.255
CIDR:       66.246.0.0/17, 66.246.128.0/18
NetName:    NAC-NETBLK06
NetHandle:  NET-66-246-0-0-1
Parent:     NET-66-0-0-0-0
NetType:    Direct Allocation
NameServer: NS1.NAC.NET
NameServer: NS2.NAC.NET
Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
Comment:
Comment:    * Reassignment information for this network is available
```

```
Comment:    * available at whois.nac.net 43
RegDate:    2002-03-08
Updated:    2003-12-04

TechHandle: ZN77-ARIN
TechName:   Net Access Corporation
TechPhone:  +1-800-638-6336
TechEmail:  legal@nac.net

OrgAbuseHandle: ABUSE156-ARIN
OrgAbuseName:   Abuse Department
OrgAbusePhone:  +1-800-638-6336
OrgAbuseEmail:  abuse@nac.net

OrgNOCHandle: NOC270-ARIN
OrgNOCName:   Network Operations Center
OrgNOCPhone:  +1-973-590-5050
OrgNOCEmail:  network@nac.net

OrgTechHandle: ZN77-ARIN
OrgTechName:   Net Access Corporation
OrgTechPhone:  +1-800-638-6336
OrgTechEmail:  legal@nac.net

OrgTechHandle: AR97-ARIN
OrgTechName:   Rubenstein, Alex
OrgTechPhone:  +1-973-590-5101
OrgTechEmail:  alex@nac.net
```

Methodology:

The methodology used in this analysis was as follows:

- Gather log files for analysis using ftp.
- Determine content of files using DOS type, more and sort functions.
- Determine how to order content logically. Columns/Rows.
- Import text files into 15 separate tables.
- Separate ports from source and destination columns using “Replace”.
- Export tables to text files with “space” separating fields.
- Re-import text files into tables with space separating fields.
- Resulting database contained 15 separate tables. 5 alert, 5 scan, 5 oos.
- Query each day for top 10 internal and external alerters, scanners and oos entries.
- Create queries to find top 10 in each log.
- Build separate tables from the top 10 in each log.
- Create snort html output using snort_sort.pl for each day.
- Browse html output to determine alerts generated.
- Search html files for MY.NET using wingrep.
- Copy output to Wordpad and format to text file.
- Import as tables into MSAccess from space delimited text files.

- Generate queries to correlate events and alerts by IP.

Example Queries:

The following query was created with Access using the QBE or the query by example graphic and was modified to select all traffic not including the home network, MY.NET.0.0/16:

```
SELECT DISTINCTROW First(scans1202.dst) AS [dst Field], Count(scans1202.dst) AS
NumberOfDups
FROM scans1202
GROUP BY scans1202.dst
HAVING (((Count(scans1202.dst))>1)) and scans1202.dst not like "*130.85.*";
```

In a like manner the following is an example of querying the database such that only traffic associated with 130.85.0.0/16 is extracted.

```
SELECT DISTINCTROW First(scans1202.dst) AS [dst Field], Count(scans1202.dst) AS
NumberOfDups
FROM scans1202
GROUP BY scans1202.dst
HAVING (((Count(scans1202.dst))>1)) and scans1202.dst like "*130.85.*";
```

The above 2 example queries were iterated over the 15 database tables and data collected into other tables based on fairly simple criteria. It was in this manner that tables were built holding the pertinent information required of the practical.

References:

ISS. Advanced RealSecure. Student Guide. Atlanta: Internet Security Systems, 1998. 59-82.

"*Architecture Issues"* component of our Beginning Analysis Course of Intrusion Detection in Depth. 2003.

Edwards, Simon. "Network Intrusion Detection Systems: Important IDS Network Security Vulnerabilities". Top Layer Networks. September 2002.
URL:http://www.toplayer.com/pdf/WhitePapers/wp_network_intrusion_system.pdf

Laing, Brian. "How To Guide: Intrusion Detection Systems". Internet Security Systems. 2000. Sovereign House.
URL:http://www.snort.org/docs/iss-placement.pdf

"Layer 4 Switching White Paper". Revision 1. January 4, 2002. Telco Systems, A BATM Company.
URL:http://www.telco.com/products/IPswitching/MultiLayer/t5pro/?f=/wp/layer_4_switches_010402.pdf

Birje, Uday. “Multi-layer switching in the enterprise”. Network Magazine India. URL:http://www.networkmagazineindia.com/200301/cover13.shtml,

Northcutt, Stephen. Network Intrusion Detection, An Analysts Handbook. Indianapolis: New Riders Publishing, Jun. 99. 246

Stevens, W. Richard. TCP/IP Illustrated,Volume 1. Reading: Addison-Wesley, 1994. 56, 232, 246

Log Files for Network Detects:
URL:http://www.incidents.org/logs/Raw/2002.9.18
URL:http://www.incidents.org/logs/Raw/2002.9.19
URL:http://www.incidents.org/logs/Raw/2002.9.20
URL:http://www.incidents.org/logs/Raw/2002.9.27.
URL:http://www.incidents.org/logs/Raw/2002.5.1

Log Files for Network Analysis:
URL:http://www.incidents.org/logs/oos/oos_report_031202
URL:http://www.incidents.org/logs/oos/oos_report_031203
URL:http://www.incidents.org/logs/oos/oos_report_031204
URL:http://www.incidents.org/logs/oos/oos_report_031205
URL:http://www.incidents.org/logs/oos/oos_report_031206
URL:http://www.incidents.org/logs/scans/scans.031202.gz
URL:http://www.incidents.org/logs/scans/scans.031203.gz
URL:http://www.incidents.org/logs/scans/scans.031204.gz
URL:http://www.incidents.org/logs/scans/scans.031205.gz
URL:http://www.incidents.org/logs/scans/scans.031206.gz
URL:http://www.incidents.org/logs/alerts/alert.031202.gz
URL:http://www.incidents.org/logs/alerts/alert.031203.gz
URL:http://www.incidents.org/logs/alerts/alert.031204.gz
URL:http://www.incidents.org/logs/alerts/alert.031205.gz
URL:http://www.incidents.org/logs/alerts/alert.031206.gz

Other Reference URLs:
URL:http://www.snort.org/cgi-bin/sigs-search.cgi?sid=sid+.
URL:http://www.nipc.gov/warnings/advisories/1999/99-024.htm
URL:http://www.nipc.gov/warnings/advisories/1999/99-024.htm
URL:http://www.securityfocus.com/archive/75/31239
URL:http://www1.dshield.org/ipinfo.php?ip=62.168.63.245&Submit=Submit
URL:http://www.faqs.org/rfcs/rfc3171.html
URL:http://archives.neohapsis.com/archives/sf/ids/2002-q2/0019.html
URL:http://www.whitehats.com/info/IDS28]
URL:http://www.ietf.org/rfc/rfc0894.txt?number=894
URL:http://www.apnic.net/apnic-bin/whois.pl
URL:http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids28&view=event

URL:http://www.iss.net/security_center/advice/Intrusions/2000310/default.htm:
URL:http://www.insecure.org/nmap/nmap-fingerprinting-article.html
URL:http://www.novell.com/coolsolutions/netware/features/a_ifolder_21_protected_nw.html
URL:http://sunsolve.sun.com/pub-cgi/retrieve.pl?doctype=coll&doc=secbull/142&type=0&nav=sec.sba
URL:http://razor.bindview.com/publish/advisories/adv_novellleak.html
URL:http://www.giac.org/practical/GCIA/Daniel_Clark_GCIA.pdf
URL:http//www.giac.org/practical/GCIA/Loic_Juillard_GCIA.pdf
URL:http//www.giac.org/practical/GCIA/John_Hally_GCIA.pdf
URL:http//www.sans.org/resources/idfaq/subseven.php
URL:http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm
URL:http://www.digitalirc.net/index.htm
URL:http//www.dshield.org
URL:http//www.iana.org/assignments/port-numbers
URL:http//swhois.net