In-Depth Intrusion Detection
GCIA Practical Assignment v3.1

Alejandro J. Froyo

SANS 2002 Orlando, Florida – USA

PART 1

The State of Intrusion Detection
Utilization of SNMP Scans as a Reconnaissance Technique

Introduction

Modern intrusion detection is in a constant state of flux. New techniques for information gathering, intrusion or detection evasion are growing at a steady pace. The adversarial relationship between 'Us' (the good guys) and 'Them' (the bad guys) is thriving mostly in part to the pervasive availability of malicious code found throughout the Internet. The vast majority of today's exploits and intelligence gathering exercises are rehashes of 'old veterans' being executed by individuals who lack appreciation or understating of the code. Therefore; they throw everything at a target hoping something hits the mark, which in turn makes a lot of noise in our intrusion detections systems and eventually makes an analyst wonder why would anyone send an IIS exploit if I am running Apache. In a field where any knowledge of your adversary is highly desirable, a talkative SNMP enabled target can provide accurate, irrefutable and privileged information to the attacker. This paper will examine the use of SNMP scans as a powerful intelligence-gathering tool that can convert simple exploit into a precision guided munitions.

Intelligence Gathering

Intelligence gathering is the cornerstone to any successful intrusion attempt. It can be further said, that any successful intelligence gathering exercise needs to be targeted, accurate and most important stay undetected by intrusion detection systems. Modern day information gathering methods are based on stimulus and response. The attacker sends a stimulus to the target host and the target responds in a predictable manner to stimulus. Correct interpretation of the response or sometimes the lack off one is critical to the success of the scan. How a target device responds can assist the attacker in mapping the network, determine countermeasures or help in the fingerprinting type of operating system. Intelligence gathering activity is always associated as a prelude to an attack, it is for this reason that scan must stay undetected for as long as possible. A fast and furious host scan will fire off alerts on any Network Intrusion Detection System (NIDS), but a scan that resembles a Texas Barbeque (low intensity and distributed through several days) has a very good chance of going undetected. NIDS use a variety of rules to identity intelligence gathering activities directed at the hosts they are protecting. NDIS detection rules are based on the tale-tale signs of a scan or an exploit. These fingerprints are compared against all packets that are directed at the protected network. If there is a match, an alert is generated and the event is logged. But, since the Internet has become is a very hostile place NIDS manufactures have begun desensitizing their scanners to reduce the number of 'background noise' alerts. A scan that is well planned and executed with patience can pass undetected and considered 'background noise'.

Modern day intelligence gathering tools push the envelope and bend the rules of IP networks. Tools like Nmap fabricate packets (stimulus) that would not exist in 'nature' in order to achieve a response from the target host. Most Nmap crafted packets have been classified and fingerprinted by most NIDS manufactures, and will set off an alert as possible Nmap activity. At the same time, simple stimulus packet like an SNMP probe has about the same chance of gathering information, but a reduced chance of detection by a NDIS, because it might be perceived as 'background noise'.

Background on SNMP

The Simple Network Management Protocol (SNMP) was intended to assist system administrators manage network-attached devices through the use of collection agents reporting to a central management console. SNMPv1 was first defined by the IETF (Internet Engineering Task Force) in RFC 1157. The goal of the IETF was to create an architecture, which could easily grow with future technology advances, place a minimal set of restriction on the management applications and use a small but effective command set. Moreover, the IETF established a sufficiently open framework that would allow for interoperability of diverse systems and their management with non-proprietary application suites. SNMP has been embraced as a remote management solution not only by the computer industry, but many other industries requiring remote management. Today, industrial equipment management solutions harness the simplicity of SNMP to manage everything from automated fabrication equipment in automobile plants to central heating and ventilation units in large buildings and even medical equipment at hospitals. I can be said that SNMPv1 is one of the few protocols that has gained acceptance, by all manufactures without putting their own spin on it's implementation.

The Nuts and Bolts of SNMP

As a protocol SNMP fully complies with the Open Systems Interconnect (OSI) and the TCP/IP reference model. SNMP can ride along a variety of transport protocols; in the case of IP based networks SNMP takes advantage of the User Datagram Protocol (UDP) transport protocol. The connectionless nature of UDP makes the SNMP packet extremely lightweight. ICMP packets from destination host to the source handle any transmission error reporting. This is expected behavior for error reporting for packets traveling through UDP. "SNMPv1 supports five different types of messages: **GetRequest**, **SetRequest**, **GetNextRequest**, **GetResponse**, and **Trap**. A single SNMP message is referred to as a Protocol Data Unit (PDU). These messages are described using

Abstract Syntax Notation One (ASN.1) and translated into binary format using Basic Encoding Rules (BER). SNMP request messages are sent from managers to agents."[1] Trap messages are sent from the SNMP enabled agents to designated managers or 'trap-catchers'.

SNMP messages are directed at the host's Managed Information Base or MIB. The MIB acts as a virtual repository for managed objects. "Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) [8] defined in the SMI. In particular, each object has a name, syntax, and an encoding. The name is an object identifier, an administratively assigned name, which specifies an object type. The object type together with an object instance serves to uniquely identify a specific instantiation of the object."[2] Each manufacturer of managed objects makes available their MIBs to the public for ease of integration to third-party management suites like: Unicenter or HP OpenView. The syntax of a MIB is defined in a RFC published by the ITEF. The standardization of MIBs and the openness by manufacturers to provide MIB information has greatly enhanced acceptance of system management through SNMP. With the appropriate MIB information it is very easy to build a query to retrieve very specific information from a SNMP managed host. Moreover, with the proper security profile it is very easy to affect the behavior of SNMP manage host.

SMNP security is based on a shared secret, known as the community name. Within the UPD transmission packet the community name is sent as part of the payload. The destination entity receives the message and parses data to reach the community string. The community name is sent in clear text form to destination host. The host then compares the community name against it's known profiles for authentication. If the community name exists the packet is forwarded for further processing if it's not present the whole packet is discarded. The profiles for know community names have three possible levels of access:

<u>Read</u>

Profiles with read access are permitted to query the host for any information available within the MIB. There is no way to restrict portions of information stored with the virtual repository (MIB), a community name with Read permission will be able to query **<u>all</u>** stored information.

<u>Read-Write</u>

Read-Write permission allows the source systems not only to perform read operations against the entire MIB, but also it allows changing existing values stored in the MIB. This level of permission can effectively make changes to

[1] Finlay, Ian A. "Vulnerability Note VU#854306: Multiple vulnerabilities in SNMPv1 request handling." <u>CERT Coordination Center</u>. Online. Internet. 2 February 2002

[2] McCloghrie, K. and Rose, M.T. 1991. "Management Information Base for Network Management of TCP/IP-based Internets: MIB-II," RFC 1213 (Mar.)

hardware configuration of the managed host, and if there is malicious intent, render the host inoperable.

Read-Create
Read-Create profiles can perform all actions available to read-write permission set, but can further alter the content of the MIB by inserting a new data string.

The inherent weakness of SMNPv1 security is in the transmission of the shared secret without any means of encryption or exchange of certificates. Therefore it would be very easy to 'guess' the community names used in a network by sniffing a network segment. Once the community name has been compromised, a strange birthmark in the form of a bulls-eye appears on the target hosts. At this point it is safe to say the network is at risk, especially if SNMP traffic is permitted inbound from the Internet.

Dissection of an SNMP Query at Packet Level

As mentioned before in IP based network SNMP rides along on a UDP packet. Therefore, it possesses all the attributes associated with UDP: No flow control, connectionless, stateless, very low overhead and supports broadcasts / unicast / multicast addresses. "A message consists of a version identifier, an SNMP community name, and a protocol data unit (PDU). A protocol entity receives messages at UDP port 161 on the host with which it is associated for all messages except for those which report traps (i.e., all messages except those which contain the Trap-PDU). Messages, which report traps, should be received on UDP port 162 for further processing. An implementation of this protocol need not accept messages whose length exceeds 484 octets."[3] Below is a typical SNMP GetRequest query using the public community name. The SNMP query was made using SNScan v1.04, courtesy of Foundstone Inc.

```
10.80.50.130.1289 > 10.10.5.115.161:  GetRequest(31) .1.2.840.10036.3.1.2.1.2.1
```

```
<4500  004a  016d  0000  8011  ece7  0a50  3282
 0a0a  0573>[0509  00a1  0036  dee2]{302c  0201
 0004  0670  7562  6c69  63a0  1f02  0426  805d
 1e02  0100  0201  0030  1130  0f06  0b2a  8648
 ce34  0301  0201  0201  0500}
```

The first portion of the packet in <Red> is the IP header information. Encapsulated within the IP Datagram in [Blue] is the UDP portion of the packet, followed in [Green] by the actual payload. By examining the hex dump we can determine that this is a pretty standard IPv4 packet with a IP header length of 20 bytes. Furthermore; we can confirm the transport mechanism is UDP by

[3] Case, Jeffrey D., Fedor, Mark S., Schoffstall, Martin L., and Davin, C. 1990. "Simple Network Management (SNMP)," RFC 1157 (May.)

observing the value of 9th byte that is set to 0x11. The packet is from source IP 10.80.50.130 and is using ephemeral port 1289 and is directed at IP 10.10.5.115 on well-known port 161 (SNMP). The community name 'public' is part of the payload and is represented in clear text form as Hex string 0x7075626c8963. The GetRequest is attempting to retrieve information form the MIB located at 1.2.840.100036.3.1.2.1.2.1.

```
10.10.5.115.161 > 10.80.50.130.1289:  GetResponse(31) noSuchName@ 1 .1.2.840.10036.3.1.2.1.2.1=

      <4500  004a  914d  0000  7d11  6007  0a0a  0573
       0a50  3282>[00a1  0509  0036  ddde]{302c  0201
       0004  0670  7562  6c69  63a2  1f02  0426  805d
       1e02  0102  0201  0130  1130  0f06  0b2a  8648
       ce34  0301  0201  0201  0500}
```

The destination host performs validation of the community name and looks for the information stored at the requested location in the MIB. In trace above the target host responds that no information was found at location 1.2.840.10036.3.1.2.1.2.1 in the host MIB. Notice the target host reply is sent using source port UPD 161 and that the public community name is included as part of packet payload. Foundstone's SNMP scanning tool has some built-in logic that if it receives GetResponse with noSuchName at the specified MIB location, it will retransmit SNMP query to a more generic MIB location. The scanning host performs a GetNextRequest to a more generic location (1.3) in the target's MIB.

```
10.80.50.130.1289 > 10.10.5.115.161:  GetNextRequest(18) .1.3

      <4500  003d  016e  0000  8011  ecf3  0a50  3282
       0a0a  0573>[0509  00a1  0029  8889]{301f  0201
       0004  0670  7562  6c69  63a1  1202  0101  0201
       0002  0100  3007  3005  0601  2b05  00}
```

This time the scan hits the mark and the target host responds to the SNMP query. As it can be seen in the trace below, we were able to gather the target's hardware platform and the version of the operating system it is running, in this case Windows 2000. Notice the community is name is present in the response and the rest of payload is all clear text.

```
10.10.5.115.161 > 10.80.50.130.1289:  GetResponse(156) .1.3.6.1.2.1.1.1.0="Hardware: x86 Family 6 Model
8 Stepping 6 AT/AT COMPATIBLE - Software: Windows 2000 Version 5.0 (Build 2195 Multiprocessor Free)"

      <4500  00c9  914e  0000  7d11  5f87  0a0a  0573
       0a50  3282>[00a1  0509  00b5  020c]{3081  aa02
       0100  0406  7075  626c  6963  a281  9c02  0101
       0201  0002  0100  3081  9030  818d  0608  2b06
       0102  0101  0100  0481  8048  6172  6477  6172
       653a  2078  3836  2046  616d  696c  7920  3620
       4d6f  6465  6c20  3820  5374  6570  7069  6e67
       2036  2041  542f  4154  2043  4f4d  5041  5449
424c  4520  2d20  536f  6674  7761  7265  3a20
5769  6e64  6f77  7320  3230  3030  2056  6572
7369  6f6e  2035  2e30  2028  4275  696c  6420
3231  3935  204d  756c  7469  7072  6f63  6573
736f  7220  4672  6565  29}
```

SNMP Intelligence Gathering in Action

Now that we have seen the basic mechanics of an SNMP query, we can begin to look at the possible uses of SNMP sweeps as an information gathering technique. For the purpose of facilitating a stable and controlled environment all SNMP stimulus packets have been generated using SNScan v1.04 from Foundstone, Inc. The SNScan tool allowed us to query a host with a user controllable community name, that can be directed at a variety of UDP ports. The tool permitted us to scan multiple IP address with the ability to randomize the scan pattern. SNScan is extremely quick at scanning hosts, and the folks a Foundstone, Inc. make it available to the public at a very attractive price – Free. The target hosts used in our exercise are running Windows 2000 and AIX 4.3. They are configured with various SNMP community names. Listed below are several scenarios illustrating how a single stimulus can have multiple responses, depending on the host or network configuration. With each response a variety on information is gathered, all painting a clearer picture on the target.

Directed SNMP Query – Host Not listening for SNMP Traffic

Our first test scenario the scanning host, actively targeting a host located at IP 192.168.0.3. The scanner is configured to look for community name 'public' and for host listening on port UDP 161. Our target host dose not have an SNMP service running.

Stimulus

```
192.168.0.25.3108 > 192.168.2.3.161:  GetRequest(31) .1.2.840.10036.3.1.2.1.2.1
```

Response

```
192.168.2.3 > 192.168.0.25: icmp: 192.168.0.3 udp port 161 unreachable (ttl 127, id 71)
```

The stimulus is a typical GetRequest directed at the host. Since there no service is running and listening on port udp 161, the target host send an ICMP error message announcing that udp port 161 is unreachable. Therefore, based of the host response we can determine to following information:

The target host is alive, which assists us in mapping of the target network. The host is not running SNMP. We can see the time-to-live (ttl) of the return ICMP pack is 127. By comparing operating systems base ttls we can say there is a strong probability that we are dealing with a Window 2000 (Windows 2000 hosts default ttl is 128).

Directed SNMP Query – Host listening for SNMP traffic with matching Community Name

In our test, the scanning host is using public as a community name and directing the scan at host located at IP 10.10.5.115. The target host has the SNMP service running and public as a valid SNMP community name with read access. This scenario might seen highly unlikely, after all what are the chances any host on the Internet would be listening for SNMP traffic with a community name like **public**? Well, the truth be known the odds are quiet high to encounter a host matching the configuration outlined above. The sad reality is that a variety of hardware vendors ship their products with SNMP enabled and set to a default community name like public.

Stimulus

```
10.80.50.130.1289 > 10.10.5.115.161:  GetRequest(31) .1.3
```

Response

```
10.10.5.115.161 > 10.80.50.130.1289:  GetResponse(156) .1.3.6.1.2.1.1.1.0= "IBM PowerPC
CHRP Computer.Machine Type: 0x0800004c Processor id: 006015254C00.Base Operating System
Runtime AIX version: 04.03.0003.0000.TCP/IP Client Support  version: 04.03.0003.0000"
```

As expected the scan proves to be successful and returns valuable information about the target host. Not only we confirm the host is alive, but we know the hardware platform, the type and version of OS. Since IBM makes their MIBs available to the public, now you can build your own customized SNMP query and gather as much information about the target. If attacker is crafty enough, there are a wide variety of MIB walkers, which will interrogate the virtual repository, and data mine every piece of information for your analysis. The most disturbing aspect associate with a community name match on an exposed system is when the community name has a security profile with Write or Create privileges. Once this level of access has been achieved, an intruder can actually manipulate hardware settings and configuration. If this seems a little out of someone's reach think again, there are toolkits available to brute force SNMP community names readily available. It's only then a matter of time until a matching name is found.

Directed SNMP Query – Router ACL Blocking SNMP Inbound Traffic

In this scenario we have placed and access control list (ACL) at the router blocking all UDP traffic directed at UDP port 161.

Stimulus

```
192.168.0.25.2301 > 10.10.1.200.161:  GetRequest(31) .1.2.840.10036.3.1.2.1.2.1
```

Response

```
192.168.0.254 > 192.168.0.25: icmp: host 10.10.1.200 unreachable - admin prohibited filter
```

In this stimulus the scanning host is still making a standard GetRequest directed at host 10.10.10.1.220. This time the response is not as favorable, our router returns an ICMP unreachable – admin prohibited filter. From the surface, this might seem like a dead end scan but it's quite the contrary. From the response received, we can infer that an access control list is present somewhere in our path to the target host. Most likely the ACL is enforced on a border router at the edge of the target network or at a firewall. Even if the router was silenced and not issue ICMP unreachable messages, we would still be intrigued by the black-hole phenomenon. Based on this result, a savvy hacker would shift priorities and attempt to map out the ACL rules on the router, before attempting any further host scans.

Directed SNMP Query – Host is Not Configured with same SNMP Community Name

Our next scenario the scanning host is targeting a host located at IP 10.10.1.200. The scanner is configured to look for community name 'private' and for a host listening on port UDP 161. Our target host has the SNMP service running but no matching community name.

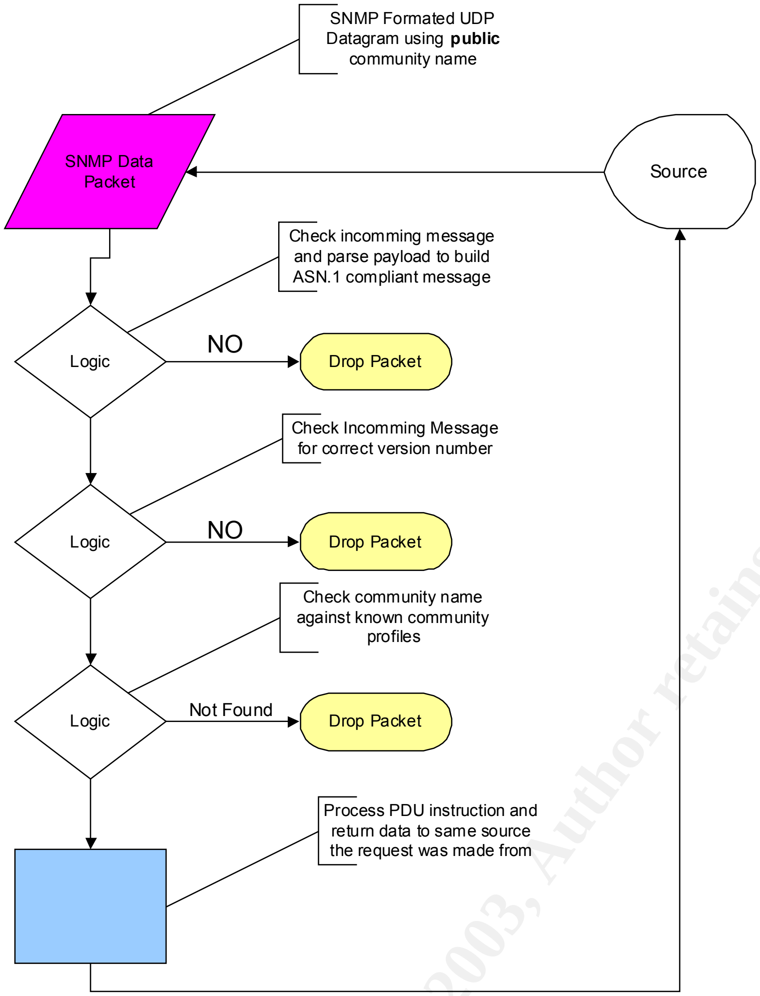
Stimulus

```
192.168.0.25.2301 > 10.10.1.200.161:  GetRequest(31) .1.2.840.10036.3.1.2.1.2.1
```

Response

```
No response received
```

The lack of any response from the host or any associated network device is similar to behavior experienced with a SNMP query directed at a silenced router using ACL to block inbound traffic. But, this black hole behavior can have an alternate explanation. The diagram below illustrates the logical flow, as outlined in RFC 1157, the SNMP packet is subjected to before a reply is sent back. Once the packet arrives at the destination it begins to move up the stack. The payload is subjected to a variety of parsings and logic checks, like: version checks and proper formatting. If at any step the data fails to deliver the expected result, the packet is dropped. Because the decision to drop the packet was taken at a higher layer of IP stack, no ICMP error message is sent back to the sender. According to layer 4 or below of OSI model, the packet was received and forwarded for additional processing. The operating system elected not to further process the request, and thus explaining the black hole behavior.


SNMP Formated UDP Datagram using **public** community name
SNMP Data Packet
Source
Check incomming message and parse payload to build ASN.1 compliant message
Logic
NO
Drop Packet
Check Incomming Message for correct version number
Logic
NO
Drop Packet
Check community name against known community profiles
Logic
Not Found
Drop Packet
Process PDU instruction and return data to same source the request was made from

Conclusion

As it has been seen with proper interpretation of responses from SNMP stimuli, valuable information about the target host and it's associated network infrastructure can be discovered. Moreover, if the SNMP stimuli posses a community string with READ-WRITE permissions, the target host behavior can be altered. At this point SNMP stops being a gathering tool and becomes an exploit tool. From the perspective of intelligence gathering technique SNMP can be used at the very least to efficiently map a target network. Given the wide use of community names like 'public' there is a very high probability of success of a more in-depth SNMP query scan reveling important host information. This assumption is especially true when the targets are home user devices (personal computers/broadband access points). In addition, the newly discovered vulnerability in the implementation of SNMPv1 has raised the awareness of SNMP not only as an intelligence-gathering tool, but also as the source for an exploit capable of substantial damage. In either case it underscores the need for a comprehensive security policy that takes in consideration the possibility of host information leaking through SNMP and taking corrective countermeasures. At the very least system administrators should choose SNMP community name strings that are difficult to guess and insure that all border routers filter inbound traffic to UDP port 161.

References

Case, Jeffrey D., Fedor, Mark S., Schoffstall, Martin L., and Davin, C. 1990. "Simple Network Management (SNMP)," RFC 1157 (May.)

Finlay, Ian A. "Vulnerability Note VU#854306: Multiple vulnerabilities in SNMPv1 request handling." CERT Coordination Center. Online. Internet. 2 February 2002

McCloghrie, K. and Rose, M.T. 1991. "Management Information Base for Network Management of TCP/IP-based Internets: MIB-II," RFC 1213 (Mar.)

Northcut, Stephen, Judy Novak, and Donald McLachlan. Network Intrusion Detection: An Analyst's Handbook. 2nd Ed. Indianapolis: New Riders, 2000.

Northcut, Stephen, et al. Intrusion Signatures and Analysis. Indianapolis: New Riders, 2001

Stevens, W. Richard. TCP/IP Illustrated, Volume 1: The Protocols. Boston: Addison-Wesley, 1994.

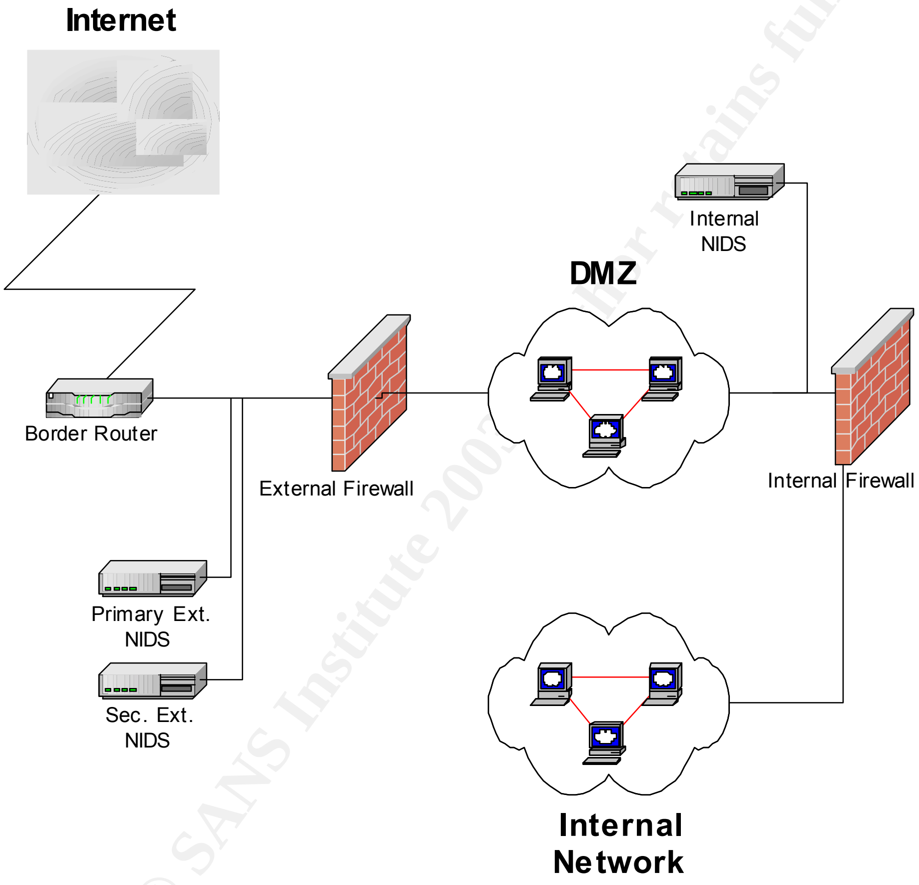
Other Resources

http://www.cert.org/tech_tips/snmp_faq.html
http://www.ee.oulu.fi/research/ouspg/protos/
http://www.foundstone.com/

PART 2

The Network Detects

The following detects are taken from a network segment at my place of work. I feel it's important to at least get a general overview of the network layout and some of the countermeasures in place. This particular network segment houses the company's business–to-business web based applications. Typical traffic is high with about 25,000 to 30,000 business partners accessing the web-based tools on a daily basis. The diagram below illustrates the basic network topology.



Sitting outside the external firewall are two network intrusion detection systems manufactured by two separate vendors. Our external firewall is setup to accept inbound traffic directed to port 80 and 443 only. There are some rules applied to outbound traffic from DMZ devices, nothing out of the ordinary. The exposed IP addresses are NATed into private addresses on the DMZ. Within our DMZ sit our load-balanced web server farm. The Web servers are locked down as per best practices and get patched on regular basis. Our network-engineering group maintains the firewalls, and they work on a daily basis at maintaining and improving the rule set. Devices from the internal network that provide back office services have their IP addresses NATed into DMZ private IP addresses. There are very well defined rules for in-bound traffic from the DMZ devices to the internal network.

First Detect – Nimda Web Scan

Source of the Trace:
The source of this capture is from my network at work.

Detect Generated By:
This detect was generated using the Windows port of Snort v1.84 using the standard rule set provide with the build package. Below, are the alerts generated by the Snort sensor. The log as been colorized to assist those not familiar with the Snort log format. The first portion of the log, in red, describes the rule set container. The second part in blue, describes the actual rule that cause the alert to trigger. The last part in green, informs us of the source IP and port information for both the source and destination. The WEB-IIS and WEB-FRONTPAGE Snort rules look for possible string matches within the packet payload. As Snort was inspecting each packet, it was being compared to the exploit strings, which the application has loaded into memory.

1. WEB-IIS CodeRed v2 root.exe access [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 12.239.52.5:3398 -> xxx.yyy.zzz.112:80
2. WEB-IIS cmd.exe access [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 12.239.52.5:3470 -> xxx.yyy.zzz.112:80
3. WEB-IIS cmd.exe access [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 12.239.52.5:3482 -> xxx.yyy.zzz.112:80
4. WEB-FRONTPAGE /_vti_bin/ access [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 12.239.52.5:3605 -> xxx.yyy.zzz.112:80
5. WEB-IIS .... access [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 12.239.52.5:3668 -> xxx.yyy.zzz.112:80
6. WEB-IIS .... access [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 12.239.52.5:3697 -> xxx.yyy.zzz.112:80
7. WEB-IIS cmd.exe access [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 12.239.52.5:3831 -> xxx.yyy.zzz.112:80
8. WEB-IIS cmd.exe access [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 12.239.52.5:3881 -> xxx.yyy.zzz.112:80
9. WEB-IIS cmd.exe access [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 12.239.52.5:3904 -> xxx.yyy.zzz.112:80
10. WEB-IIS .... access [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 12.239.52.5:3925 -> xxx.yyy.zzz.112:80

11. WEB-IIS .... access [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 12.239.52.5:4084 -> xxx.yyy.zzz.112:80
12. WEB-IIS cmd.exe access [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 12.239.52.5:4123 -> xxx.yyy.zzz.112:80

Probability the Source address was spoofed:

Low. The attacker must establish a TCP three-way handshake connection in order to compromise the system by delivering the virus payload.

Description of the Attack:

This is a scan for vulnerable Web servers. The scanning host is probably infected by a variant of the Nimda Worm.

Attack Mechanism:

The Nimda Worm attempts to spread by exploiting a weakness in directory transversal in un-patched IIS servers or looking for server which was previously compromised by the Code-Red exploit. A host, which has been infected by Nimda, begins actively scanning for Web servers. The scanning methodology used by Nimda is not truly random, as some people have been lead to believe. The scanning algorithm is described in CERT® Advisory CA-2001-26. But, in essence there is a 75% chance the infected host will scan for web servers, which are in the same first octet its IP address. In this particular detect our host is in the same class A network as the infected host. Once it finds a possible victim the infected server sends a number of GET requests to the target Web server. These requests are designed to exploit known vulnerabilities in directory transversal associated with Microsoft IIS 4.0-5.0. In addition, the infected host will try to enter the system, by utilizing a backdoor left by an other piece of malicious code, known as Code-Red. The Nimda web servers scan has a classic footprint:

1. GET /scripts/root.exe?/c+dir HTTP/1.0
2. GET /c/winnt/system32/cmd.exe?/c+dir HTTP/1.0
3. GET /d/winnt/system32/cmd.exe?/c+dir HTTP/1.0
4. GET /_vti_bin/..\../..\../..\../winnt/system32/cmd.exe?/c+dir c+dir c+dir HTTP/1.0
5. GET /_mem_bin/..\../..\../..\../winnt/system32/cmd.exe?/c+dir c+dir c+dir HTTP/1.0
6. GET /msadc/..\../..\../..\/..55../..c1../../.../winnt/system32/cmd.exe?/c+dir c+dir 32/cmd.exe?/c+dir HTTP/1.0
7. GET /scripts/..%c../winnt/system32/cmd.exe?/c+dir dir HTTP/1.0
8. GET /scripts/..%c../winnt/system32/cmd.exe?/c+dir dir HTTP/1.0
9. GET /scripts/..%c../winnt/system32/cmd.exe?/c+dir dir HTTP/1.0
10. GET /scripts/..\../winnt/system32/cmd.exe?/c+dir r dir HTTP/1.0
11. GET /scripts/..\../winnt/system32/cmd.exe?/c+dir r c+dir HTTP/1.0
12. GET /scripts/../../winnt/system32/cmd.exe?/c+dir r r HTTP/1.0

The first three entries are Nimda's attempt to use the backdoors left by Code-Red exploit. If this host were previously compromised, the virus would have used this vulnerability to infect the target host. The other GET requests are attempt by the virus to exploit vulnerabilities associated with directory transversal. In an un-patched IIS v4.0-5.0 web server is it possible to execute code in the context of the local system account. Through the use of this vulnerability any file could be accessed, Nimda is trying to run CMD.exe in order to run the malicious code to infect the target. The important part of the exploit is that it's running as system account, which as unchecked access to the entire system. With this level of access the malicious code can infect the target and turn into and other agent of propagation. This vulnerability is described in Microsoft security bulleting MS00-078

Correlations:
This attack footprint has been reported in CERT® Advisory CA-2001-26 as one of the methods of infection used by a host, which has been infected by the Nimda Virus.

Evidence of Active Targeting:
The traffic was directed at our host by the algorithm the virus uses to propagate itself. Not all hosts in our subnet were targeted only our hosts running web services.

Severity:
The attack was directed at the virtual IP of web server farm used by the business partners of the company. I would rate the **criticality** of the system as **4** given the fact if one server was compromised by the attack there are redundant systems available that could take it's place. If the scan would have been successful in finding a vulnerable system the Nimda virus could have infected it. I would rate the **lethality** of the attack against the targets as a **4**. In this particular case the system countermeasures are the heroes. The systems are patch on a regular basis and totally immune to this type of attack. I would rate the **system countermeasures** on this attack as a **5**. Against this type of attack the network countermeasures are poor. The traffic was directed at the target systems using TCP port 80 which the firewall has clear rules to permit. There is no packet inspection at the border router or at the firewall, which could have dropped the packets before reaching the target. I would have to rate the **network countermeasures** in this attack as **1**, very poor. Using the formula below the attack would have a severity of **2**.

Criticality	+	Lethality	-	System Countermeasures	Network Countermeasures	=	Severity
4		**4**	**-**	**5**	**1**	**=**	**2**

Defensive Recommendations:
There are no recommendations to further enhance our security posture. Our web servers are patch against this type of attack.

Test Question:
What exploit is the following footprint most resembled?

1. GET /scripts/root.exe?/c+dir HTTP/1.0
2. GET /c/winnt/system32/cmd.exe?/c+dir HTTP/1.0
3. GET /d/winnt/system32/cmd.exe?/c+dir HTTP/1.0
4. GET /_vti_bin/..\../..\../..\../winnt/system32/cmd.exe?/c+dir c+dir c+dir HTTP/1.0
5. GET /_mem_bin/..\../..\../..\../winnt/system32/cmd.exe?/c+dir c+dir c+dir HTTP/1.0
6. GET /msadc/..\../..\../..\/..55../..c1../../.../winnt/system32/cmd.exe?/c+dir c+dir 32/cmd.exe?/c+dir HTTP/1.0
7. GET /scripts/..%c../winnt/system32/cmd.exe?/c+dir dir HTTP/1.0
8. GET /scripts/..%c../winnt/system32/cmd.exe?/c+dir dir HTTP/1.0
9. GET /scripts/..%c../winnt/system32/cmd.exe?/c+dir dir HTTP/1.0
10. GET /scripts/..\../winnt/system32/cmd.exe?/c+dir r dir HTTP/1.0
11. GET /scripts/..\../winnt/system32/cmd.exe?/c+dir r c+dir HTTP/1.0
12. GET /scripts/../../winnt/system32/cmd.exe?/c+dir r r HTTP/1.0

A. Nimda
B. Code RED
C. Directory transversal
D. B and C
E. None of the above

The Answer is A

Second Detect – WEB-IIS cmd.exe Access

Source of the Trace:
The source of this capture is from my network at work.

Detect Generated By:
This detect was generated using the Windows port of Snort v1.84 using the standard rule set provide with the build package. Below, are the alerts generated by the Snort sensor. Once again the snort log has been colorized to help in

discerning the data fields. This time the time code for each alert has been left for purpose of correlation and analysis.

Snort Alert Log:

```
1.  04/23-17:39:18.477267  [**] [1:1002:2] WEB-IIS cmd.exe access [**] [Classification: Web
    Application Attack] [Priority: 1] {TCP} 217.136.124.125:2830 -> xxx.yyy.zzz.67:80
2.  04/23-17:39:19.547092  [**] [1:1002:2] WEB-IIS cmd.exe access [**] [Classification: Web
    Application Attack] [Priority: 1] {TCP} 217.136.124.125:2840 -> xxx.yyy.zzz.67:80
3.  04/23-17:39:19.612782  [**] [1:1002:2] WEB-IIS cmd.exe access [**] [Classification: Web
    Application Attack] [Priority: 1] {TCP} 217.136.124.125:2842 -> xxx.yyy.zzz.70:80
4.  04/23-17:39:20.380999  [**] [1:1002:2] WEB-IIS cmd.exe access [**] [Classification: Web
    Application Attack] [Priority: 1] {TCP} 217.136.124.125:2940 -> xxx.yyy.zzz.70:80
5.  04/23-17:39:55.429318  [**] [1:1002:2] WEB-IIS cmd.exe access [**] [Classification: Web
    Application Attack] [Priority: 1] {TCP} 217.136.124.125:4224 -> xxx.yyy.zzz.248:80
6.  04/23-17:39:55.478769  [**] [1:1002:2] WEB-IIS cmd.exe access [**] [Classification: Web
    Application Attack] [Priority: 1] {TCP} 217.136.124.125:4225 -> xxx.yyy.zzz.249:80
7.  04/23-17:39:55.902779  [**] [1:1002:2] WEB-IIS cmd.exe access [**] [Classification: Web
    Application Attack] [Priority: 1] {TCP} 217.136.124.125:4234 -> xxx.yyy.zzz.248:80
8.  04/23-17:39:55.969130  [**] [1:1002:2] WEB-IIS cmd.exe access [**] [Classification: Web
    Application Attack] [Priority: 1] {TCP} 217.136.124.125:4238 -> xxx.yyy.zzz.249:80
```

Probability the Source address was spoofed:

Low. The attacker must establish a state-full connection in order to compromise the system and be able to execute his exploit.

Description of the Attack:

This is a directed exploit to all web servers running IIS v4.0~5.0. The attacker is attempting to exploit a know vulnerability in Microsoft IIS which would allow for the arbitrary execution of code with privileged user rights. By looking at the Snort logs a pattern begins to appear, each web server is subjected to two WEB-IIS cmd.exe access attacks. The servers are attacked in ascending order and systems not running IIS are not subjected to the attack. This leads me to believe that the attacker has previously performed an intelligence gathering exercise on our network segment. By examining the sequence numbers from the attacker we can determine there are fairly close to one each other. This further strengthens the case for an automated script with previous knowledge of the network segment.

```
1.  17:39:18.477267 217.136.124.125.2830 > xxx.yyy.zzz.67.80: P
    3281944162:3281944284(122) ack 950063036 win 16560 (DF)
2.  17:39:19.547092 217.136.124.125.2840 > xxx.yyy.zzz.67.80: P
    3282623177:3282623302(125) ack 1114890526 win 16560 (DF)
3.  17:39:19.612782 217.136.124.125.2842 > xxx.yyy.zzz.70.80: P
    3282758089:3282758211(122) ack 2919033494 win 16560 (DF)
4.  17:39:20.380999 217.136.124.125.2940 > xxx.yyy.zzz.70.80: P
    3287945645:3287945770(125) ack 1616002298 win 16560 (DF)
5.  17:39:55.429318 217.136.124.125.4224 > xxx.yyy.zzz.248.80: P
    3353066464:3353066587(123) ack 3758430479 win 16680 (DF)
6.  17:39:55.478769 217.136.124.125.4225 > xxx.yyy.zzz.249.80: P
    3353127648:3353127771(123) ack 3745310113 win 16680 (DF)
```

```
7. 17:39:55.902779 217.136.124.125.4234 > xxx.yyy.zzz.248.80: P
   3353664775:3353664901(126) ack 3758874427 win 16680 (DF)
8. 17:39:55.969130 217.136.124.125.4238 > xxx.yyy.zzz.249.80: P
   3353898211:3353898337(126) ack 3745799070 win 16680 (DF)
```

The attack code is listed below:

```
1. GET /scripts/..%c../winnt/system32/cmd.exe?/c+dir+c:\ c:\ HTTP/1.1
2. GET /scripts/../../winnt/system32/cmd.exe?/c+dir+c:\ c:\ c:\ HTTP/1.1
```

Upon examination of the GET requests we see the attacker is attempting to invoke the command executive (cmd.exe). The syntax used in the attack is indicative that the attacker is attempting to exploit a know vulnerability in IIS servers, that would allow for directory transversal. This vulnerability is described in Microsoft security bulleting MS00-078 and in CERT® Vulnerability Note VU#111677. In addition has been entered in the Common Vulnerabilities and Exposures (CVE) database under CVE-2000-0884.

Correlations:

This scripted attack can be attributed to a variety of vulnerability scan utilities available throughout the Internet. One such s toolkit can be found at Packetstorm Security. According to the SANS institute, the foundation for this attack is listed as one of the top vulnerability associated to Windows. Based on the source IP information we where able to gather the following information from D-shield.org. As it can be seen the attacker is a broadband customer from one of Belgium's largest ISP.

Hostname:	adsl-64637.turboline.skynet.be
inetnum:	217.136.120.0 - 217.136.127.255
netname:	BE-SKYNET-20010125
descr:	Belgacom Skynet SA/NV
descr:	ADSL BAS Gent TL GO/PLUS
country:	BE
admin-c:	SN2068-RIPE
tech-c:	SN2068-RIPE

Further correlation of the attack can be seen from the TCPdump of the packets associated with the attack to host xxx.yyy.zzz.67:

```
17:39:18.477267 217.136.124.125.2830 > xxx.yyy.zzz.67.80: P
3281944162:3281944284(122) ack 950063036 win 16560 (DF)
0x0000   4500 00a2 c901 4000 6d06 61e1 d988 7c7d        E.....@.m.a...|}
0x0010   0000 0043 0b0e 0050 c39e 7e62 38a0 cfbc        .*.C...P..~b8...
0x0020   5018 40b0 3e2a 0000 4745 5420 2f73 6372        P.@.>*..GET./scr
0x0030   6970 7473 2f2e 2e25 632e 2e2f 7769 6e6e        ipts/..%c../winn
0x0040   742f 7379 7374 656d 3332 2f63 6d64 2e65        t/system32/cmd.e
0x0050   7865 3f2f 632b 6469 722b 633a 5c20 633a        xe?/c+dir+c:\.c:
0x0060   5c20 4854 5450 2f31 2e31 0d0a                    \.HTTP/1.1.
```

```
17:39:19.547092 217.136.124.125.2840 > xxx.yyy.zzz.67.80: P
3282623177:3282623302(125) ack 1114890526 win 16560 (DF)
0x0000   4500 00a5 c980 4000 6d06 615f d988 7c7d        E.....@.m.a_..|}
0x0010   0000 0043 0b18 0050 c3a8 dac9 4273 e11e        .*.C...P....Bs..
0x0020   5018 40b0 2a9d 0000 4745 5420 2f73 6372        P.@.*...GET./scr
0x0030   6970 7473 2f2e 2e2f 2e2e 2f77 696e 6e74        ipts/../../winnt
0x0040   2f73 7973 7465 6d33 322f 636d 642e 6578        /system32/cmd.ex
0x0050   653f 2f63 2b64 6972 2b63 3a5c 2063 3a5c        e?/c+dir+c:\.c:\
0x0060   2063 3a5c 2048 5454 502f 312e 310d 0a             .c:\.HTTP/1.1.
```

Note: Packets directed at the other 3 hosts are identical in construction.

Evidence of Active Targeting:
The attack was directed at only the Web servers running Microsoft IIS v4.0~5.0 and no other devices, it doesn't get more precise. Clearly there was previous scanning and mapping of the hosts to determine the target list.

Severity:
The attack was directed at all web servers found in the network segment. I would rate the **criticality** of the system as **4**. The lethality of the attack can only be gauged by its potential destructiveness. If the attack would have been successful, a very destructive piece of code could have been directed at the targets. For this reason I will rate the **lethality** of the attack as a **5**. The system countermeasures once again come to the rescue. The web servers are totally immune to this type of attack. I would rate the **system countermeasures** on this attack as a **5**. Against this type of attack the network countermeasures are poor. The traffic was directed at the target systems using TCP port 80 which the firewall has clear rules to permit. There is no packet inspection at the border router or at the firewall, which could have dropped the packets before reaching the target. I would have to rate the **network countermeasures** in this attack as **1**, very poor. Using the formula below the attack would have a severity of **3**.

Criticality	+	Lethality	-	System Countermeasures	Network Countermeasures	=	Severity
4		**5**	**-**	**5**	**1**	**=**	**3**

Defensive Recommendations:
There are no recommendations to further enhance our security posture. Our web servers are patch against this type of attack.

Test Question:
You encounter the following alert log entries in your Snort intrusion detection system. Please choose the best answer:

1. 04/23-17:39:18.477267 [**] [1:1002:2] WEB-IIS cmd.exe access [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 217.136.124.125:2830 -> xxx.yyy.zzz.67:80

```
2.  04/23-17:39:19.547092  [**] [1:1002:2] WEB-IIS cmd.exe access [**] [Classification: Web
    Application Attack] [Priority: 1] {TCP} 217.136.124.125:2840 -> xxx.yyy.zzz.67:80
3.  04/23-17:39:19.612782  [**] [1:1002:2] WEB-IIS cmd.exe access [**] [Classification: Web
    Application Attack] [Priority: 1] {TCP} 217.136.124.125:2842 -> xxx.yyy.zzz.70:80
4.  04/23-17:39:20.380999  [**] [1:1002:2] WEB-IIS cmd.exe access [**] [Classification: Web
    Application Attack] [Priority: 1] {TCP} 217.136.124.125:2940 -> xxx.yyy.zzz.70:80
5.  04/23-17:39:55.429318  [**] [1:1002:2] WEB-IIS cmd.exe access [**] [Classification: Web
    Application Attack] [Priority: 1] {TCP} 217.136.124.125:4224 -> xxx.yyy.zzz.248:80
6.  04/23-17:39:55.478769  [**] [1:1002:2] WEB-IIS cmd.exe access [**] [Classification: Web
    Application Attack] [Priority: 1] {TCP} 217.136.124.125:4225 -> xxx.yyy.zzz.249:80
7.  04/23-17:39:55.902779  [**] [1:1002:2] WEB-IIS cmd.exe access [**] [Classification: Web
    Application Attack] [Priority: 1] {TCP} 217.136.124.125:4234 -> xxx.yyy.zzz.248:80
8.  04/23-17:39:55.969130  [**] [1:1002:2] WEB-IIS cmd.exe access [**] [Classification: Web
    Application Attack] [Priority: 1] {TCP} 217.136.124.125:4238 -> xxx.yyy.zzz.249:80
```

What is the most probable explanation for these alerts:

A. Scan for servers running IIS
B. False positive cause by content distribution software using cmd.exe
C. Scripted exploit using directory transversal to run cmd.exe
D. None of the Above

The Answer is C

Third Detect – Large ICMP Packet Flood

Source of the Trace:

The source of this capture is from my network at work.

Detect Generated By:

This detect was generated using the Windows port of Snort v1.84 using the standard rule set provide with the build package. Below, are the alerts generated by the Snort sensor. The snort log has been colorized to help in discerning the data fields. This time the time code for each alert has been left for purpose of correlation and analysis.

Snort Alert Log:

```
1.  19:42:49.991387  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification: Potentially
    Bad Traffic] [Priority: 2] {ICMP} 68.64.60.159 -> xxx.yyy.zzz.70
2.  19:42:50.085020  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification: Potentially
    Bad Traffic] [Priority: 2] {ICMP} 68.64.60.159 -> xxx.yyy.zzz.70
3.  19:42:50.185615  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification: Potentially
    Bad Traffic] [Priority: 2] {ICMP} 68.64.60.159 -> xxx.yyy.zzz.70
4.  19:42:50.290688  [**] [1:499:1] MISC Large ICMP Packet [**] [Classification: Potentially
    Bad Traffic] [Priority: 2] {ICMP} 68.64.60.159 -> xxx.yyy.zzz.70
```

5. 19:42:50.485253 [**] [1:499:1] MISC Large ICMP Packet [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP} 68.64.60.159 -> xxx.yyy.zzz.70
6. 19:42:50.586498 [**] [1:499:1] MISC Large ICMP Packet [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP} 68.64.60.159 -> xxx.yyy.zzz.70
7. 19:42:50.682149 [**] [1:499:1] MISC Large ICMP Packet [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP} 68.64.60.159 -> xxx.yyy.zzz.70
8. 19:42:50.781323 [**] [1:499:1] MISC Large ICMP Packet [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP} 68.64.60.159 -> xxx.yyy.zzz.70
9. 19:42:51.186109 [**] [1:499:1] MISC Large ICMP Packet [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP} 68.64.60.159 -> xxx.yyy.zzz.70
10. 19:42:51.282210 [**] [1:499:1] MISC Large ICMP Packet [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP} 68.64.60.159 -> xxx.yyy.zzz.70
11. 19:42:51.561159 [**] [1:499:1] MISC Large ICMP Packet [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP} 68.64.60.159 -> xxx.yyy.zzz.70
12. 19:42:51.752060 [**] [1:499:1] MISC Large ICMP Packet [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP} 68.64.60.159 -> xxx.yyy.zzz.70
13. 19:42:52.013662 [**] [1:499:1] MISC Large ICMP Packet [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP} 68.64.60.159 -> xxx.yyy.zzz.70

Probability the Source address was spoofed:

There is a very probability that the source address was spoofed. The attacker has no intention to receive the ECHO replies

Description of the Attack:

This is an attempt by the attacker to flood our host with ICMP Echo Request. This flood attack is the most basic and crude form of denial of service. In this particular attack the attacker sent 14 ICMP Echo Request packets within a very short period of time. Upon detailed examination of the ICMP packet, we notice the size of the packet is set to 1500 bytes and the do not fragment flag is set, and of course it's an icmp echo request.

Windump Capture:

```
19:42:50.085020 68.64.60.159 > xxx.yyy.zzz.70: icmp: echo request (DF) (ttl 232, id 17409)

        4500  05dc  4401  4000  e801  3bd0  4440  3c9f
        0000  0046  0800  7e52  9abc  def0  0000  0000
        0000  0000  0000  0000  0000  0000  0000 0000
        0000  0000  0000  0000  0000  0000  0000 0000
```

Correlations:

This type of attack can be found under entry CVE-1999-0128 in the Common Vulnerabilities and Exposure database. The use of ICMP echo requests as a form of denial-of-service have been recorded since 1996 when CERT® issue advisory CA-1996-26 covering the denial-of-service via the ping command.

Evidence of Active Targeting:
The attack was directed at only the Web servers running Microsoft IIS v4.0~5.0 and no other devices, it doesn't get more precise. Clearly there was previous scanning and mapping of the hosts to determine the target list.

Severity:
The attack was directed at only one of web servers in the network segment. I would rate the **criticality** of the system as **4**. The lethality of the attack is relatively low as it was of short duration and the packet flow rate was less that 5 packets/per second. For this reason I will rate the **lethality** of the attack as a **2**. The web servers are immune to this type of attack; they are shielded by our firewall. I would rate the **system countermeasures** on this attack as a **5**. Against this type of attack the network countermeasures are excellent. Highly skilled network engineers maintain the firewall and there is an explicit rule to drop all ICMP echo request packets sent from any external address into the DMZ. Also, our bandwidth capacity from our ISP very large and did not encroach sufficiently to disrupt service levels. It is for this reasons that I would have to rate the **network countermeasures** to this attack as **5**. Using the formula below the attack would have a severity of **-4**.

Criticality	+	Lethality	-	System Countermeasures	Network Countermeasures	=	Severity
4		**2**	**-**	**5**	**5**	**=**	**-4**

Defensive Recommendations:
There are no recommendations to further enhance our security posture. Our Firewall is configured to drop all ICMP echo requests.

Test Question:
You are trying to find all ICMP echo request packets that have an MTU of 1500 and do not fragment flag set. Using the TCPDump provided below for reference, what BFE filter can be applied to accomplish this task?

```
19:42:50.085020 68.64.60.159 > xxx.yyy.zzz.70: icmp: echo request (DF) (ttl 232, id 17409)

        4500  05dc  4401  4000  e801  3bd0  4440  3c9f
        0000  0046  0800  7e52  9abc  def0  0000  0000
        0000  0000  0000  0000  0000  0000  0000 0000
        0000  0000  0000  0000  0000  0000  0000 0000
```

A. ip[2]=0xdc and icmp[0]=0
B. ip[2:2]=0x5dc and ip[6]=0x40 and icmp[0]=8
C. ip[6]=0x40 and icmp[0]=8

D. none of the above

The Answer is B

PART 3

Analyze This!!

The last part of the assignment we have been asked to perform a security audit to an educational institution. The audit is based on data provided by the SANS Institute. The data used in the analysis is comprised of port-scan and alert logs generated by the University's Snort intrusion detection system. The logs span across 5 days from 04-01-2002 through 04-05-2002. The files used in the analysis are summarized in the following matrix:

	Types of Data Collections		
Date:	Alert Logs:	Scan Logs:	Out-off-Spec Detections
04/01/2002	alert.020401.gz	scans.020401.gz	oos_Apr.01.2002.gz
04/02/2002	alert.020402.gz	scans.020402.gz	oos_Apr.02.2002.gz
04/03/2002	alert.020403.gz	scans.020403.gz	oss_Apr.03.2002.gz
04/04/2002	alert.020404.gz	scans.020404.gz	oss_Apr.04.2002.gz
04/05/2002	alert.020405.gz	scans.020405.gz	oss_Apr.05.2002.gz

Executive Summary

The report attached is a summary of a security audit performed on the University's network. The basis for the audit and the subsequent analysis are the Snort alert logs sent for review. The audit covered activity from April 1st, 2002 through April 5, 2002. We base our prioritization of alerts on the number of alerts received for said signature. At the same time we have ranked internal and external hosts base on the number of alerts attributed to them. During our analysis we have noted 1,049,957 alerts being recorded on the University's network intrusion detection system. These alerts matched 82 distinct signatures of the Snort detection engine. The distribution of the alerts is quite skewed, with the majority of the alerts being generated by hosts with the University's network space. Of these internal alerts, the majority can fall under the category of informational. Review of alerts generated by external host have reveled very selective targeting of few internal hosts. We have noted directed activity at hosts running web services and in particular Microsoft Internet Information Services.

Review of the 3,523,821 alerts triggered by scans, only 123,926 can be attributed to external hosts. We noticed a substantial amount of scanning activity performed by internal hosts targeting external hosts. Based on our previous experience increased scanning activity against any site is always a prelude to an attack. In this particular case the attacks might originate from host within the University's network. We understand many issues arise from the placing restrictions on information access in a University setting. We feel that several internal hosts have been compromised by external and internal sources. An in-depth review of the University's policy on access and usage should be undertaken and inform all internal parties of what constitutes accepted behavior.

The analysis set forward is based on empirical data in the form of logs from the University NIDS. Some assumptions were made as to the existence of certain hosts and the presence firewall or packet screening devices.

Analysis of Alert Signatures Recorded

During the collection period from 04-01-2002 through 04-05-2002 there were 1,049,957 events, which triggered an alert on the network intrusion detection system (NIDS). All alert events in the collection period matched 82 distinct signatures on the NIDS. The top 10 signature matches account for 97.63% of all alerts recorded. The table below breaks down the top 10 alerts received, based on volume and compares against the total number of alerts received.

Signatures	# Alerts Recorded	% of Total Alerts
connect to 515 from inside	636,038	60.58%
SNMP public access	92,595	8.82%
spp_http_decode: IIS Unicode attack detected	86,587	8.25%
SMB Name Wildcard	66,946	6.38%
spp_http_decode: CGI Null Byte attack detected	44,305	4.22%
ICMP Echo Request L3retriever Ping	33,491	3.19%
INFO MSN IM Chat data	22,006	2.10%
MISC Large UDP Packet	16,799	1.60%
High port 65535 udp - possible Red Worm - traffic	14,653	1.40%
INFO Inbound GNUTella Connect request	11,680	1.11%

Analysis of signature distribution data has yielded the following conclusions:

Connect to 515 from Inside

The majority of the alerts (60.58%) can be attributed to a signature looking for internal connections directed to TCP port 515. Port 515 is used by the LPR service for Unix. This kind of traffic is not uncommon if your network provides printing services through its Unix systems. The alerts appear to be focused on traffic directed at 5 hosts on the University's network. The distribution of the traffic can be seen on the table provided below.

Destination Hosts	Alerts	% of Total Alerts
MY.NET.150.198	331784	52.1642%
MY.NET.151.77	299713	47.1219%
MY.NET.150.83	4515	0.7099%
MY.NET.1.63	25	0.0039%
MY.NET.5.35	1	0.0002%

This traffic seems to be normal network traffic, but there is known vulnerability associated to LPR service running on some versions of Linux and variant of BSD. The vulnerability is listed on the CERT® website at URL: http://www.kb.cert.org/vuls/id/382365. We recommend the System Administrators review the version of the LPR service running on all Unix servers and in particular the 5 hosts listed on the table above.

Correlation:
The SANS Institute put out an advisory on increased scanning activity directed at TCP port 515. The advisory can be located at SANS website at the following URL: http://www.sans.org/newlook/alerts/port515.htm. In addition CERT® issued an advisory associated to un-patched version of LPRng. CERT® warned that a code defects could allow for the arbitrary execution of code, through the insertion of strings. The full technical merits of the advisory can be found at http://www.cert.org/advisories/CA-2000-22.html.

SNMP Public Access

There were 92,595 (8.82%) alerts triggered by traffic requesting access to an SNMP managed device using the public community name. All the alerts were generated by host internal to the University network directed to hosts within the University's network. The traffic looks like it could have been generated by management platforms (TNG, OpenView, Tivoli or Compaq Insight Manager) gathering SNMP information from their managed hosts. At the same time hacker trying to gather host target information through the use of SNMP queries could have generated this traffic. The alert pattern seems to show that 25 hosts attempted connections using the SNMP public string and 154 hosts were targeted.

	Sources	Destinations
SNMP public access	25	154

Based on our analysis and computer security best practices, we recommend the use of '**public**' as a community name be discontinued from all SNMP managed hosts. In addition, given the recently discovered vulnerability associated to the implementation of SNMPv1. We are further recommending that all devices managed through SNMP be patched to the latest version of code.

Correlation:

The Oulu University in Finland published its findings exposing a vulnerability associated to the implementation of SNMPv1 on February 12, 2002. The document outlined the testing and use of the PROTOS Test-Suite: c06-snmpv1 can be found at http://www.ee.oulu.fi/research/ouspg/protos/testing/c06/snmpv1/. CERT® issued advisory CA-2002-03 covering the discovery of multiple vulnerabilities associated to SNMPv1 and made similar recommendations to possible remedies for the exploit.

spp_http_decode: IIS Unicode attack detected

86,587 (8.25%) alerts triggered by traffic identified as possible IIS Unicode attacks. Unicode attacks are designed to inject characters into an http string in order to attempt arbitrary execution of code. Typically Unicode attacks are used in Website defacements or other type of system compromises. Based on the logs analyzed there where 182 source hosts identified as initiators of spp_http_decode: IIS Unicode attacks. As the same time, we identified 1017 target destinations to the attacks mentioned. The top 10 hosts involved in initiating attacks are listed in the table below:

Source Hosts	**Alerts**	**% of Total Alerts**
MY.NET.153.146	4850	5.60%
MY.NET.153.120	3434	3.97%
MY.NET.153.124	3336	3.85%
MY.NET.153.110	3136	3.62%
MY.NET.153.171	3097	3.58%
MY.NET.153.199	2731	3.15%
MY.NET.153.189	2569	2.97%
MY.NET.153.180	2444	2.82%
MY.NET.153.165	2244	2.59%
MY.NET.153.112	2138	2.47%
MY.NET.153.106	2052	2.37%
MY.NET.153.203	2000	2.31%

The top 10 hosts account for 39.30% of all recorded attacks. More interesting is the fact that they all are located on the same subnet MY.NET.153.0. This would lead us to believe that either more than one host has been compromised in MY.NET153.0 or this network segment is used in a student lab environment with little system administrator supervision. The majority of the spp_http_decode: IIS Unicode attacks are directed at hosts outside the University's network. The table below lists the top 10 hosts targeted by this type of attack.

Destination Hosts	Alerts	% of Total Alerts
211.115.213.202	8607	9.94%
211.115.213.207	2874	3.32%
211.233.29.218	2289	2.64%
211.32.117.26	1760	2.03%
61.78.53.102	1582	1.83%
211.32.117.31	1499	1.73%
211.233.28.18	1355	1.56%
211.110.11.145	1256	1.45%
211.32.117.189	1199	1.38%
211.233.28.53	1138	1.31%
211.233.28.55	1066	1.23%
211.233.28.44	1046	1.21%
211.239.154.101	966	1.12%

The top 10 destinations for the spp_http_decode: IIS Unicode attack account for 30.76% of all alerts received for this signature. Of interest is the particular focus of attacks directed to select class B networks.

Class-B Network	# of Alerts	Percentage of Total Alerts Received
211.233.0.0	29268	33.80%
211.115.0.0	16067	18.56%
211.32.0.0	14407	16.64%
	59742	69.00%

As it can be seen of the 86,587 alerts received for this attack, 59,742 (69.00%) of them were directed to one of the three class-b networks listed above. Some of these addresses are registered to a co-location/ISPs service in Korea, which may prove to have correlation with other attacks. This directed activity coming from the University's network and in particular from the MY.NET.153.0 should be further investigated. At the very least all workstations and servers operating in the MY.NET.153.0 segment should be inspected for signs of compromise in the form of Trojans or authorized activity by regular users.

Correlation:

The spp_http_decode: IIS Unicode attack has been documented from an incident handlers perspective in Potheri Mohan's GCIH practical, posted on the GIAC site at URL: http://www.giac.org/practical/Potheri_Mohan_GCIH.doc. A vulnerability report was posted on CERT® http://www.kb.cert.org/vuls/id/111677. The attack pattern is listed in the Common Vulnerabilities and Exposures database under CAN-2000-0884.

SMB Name Wildcard

66,944 (6.38%) alerts triggered by traffic identified as using SMB Name Wildcards. SMB wildcard traffic is usually associated with the anonymous enumeration of Windows shares. The table below lists the top 10 sources of the alert generated.

Source of Attack	# of Alerts	% of the Total Alerts Recorded
MY.NET.11.6	15052	22.48%
MY.NET.11.7	11500	17.18%
MY.NET.11.5	5655	8.45%
MY.NET.152.167	881	1.32%
MY.NET.152.168	781	1.17%
MY.NET.152.161	730	1.09%
MY.NET.152.177	722	1.08%
MY.NET.152.166	687	1.03%
MY.NET.152.172	678	1.01%
MY.NET.152.21	647	0.97%
MY.NET.152.183	645	0.96%

Of the 66,944 alerts generated 37978 (56.73%) can be attributed to one of the hosts in the top 10 list for SMB Name Wildcards. Further analysis of the data reveals that SMB traffic seems to be concentrated in the MY.NET.11.0 and MY.NET.152.0 network segments. This could be indicative of Windows or SAMBA enabled Unix hosts present on the segments. The table below lists the top 10 destination for the SMB Name Wildcards.

Destination of Attack	**# of Alerts**	**% of the Total Alerts Recorded**
MY.NET.11.6	14957	22.34%
MY.NET.11.7	11481	17.15%
MY.NET.11.5	5640	8.42%
MY.NET.152.167	883	1.32%
MY.NET.152.168	790	1.18%
MY.NET.5.4	750	1.12%

MY.NET.152.161	733	1.09%
MY.NET.152.177	730	1.09%
MY.NET.152.166	689	1.03%
MY.NET.152.172	676	1.01%
MY.NET.152.21	644	0.96%

As it can be seen there is no surprise to see that the top destinations for SMB Name Wildcards are located in the MY.NET.11.0 and MY.NET.152.0 network segments. This further enforces the theory of Windows or SAMBA hosts being present. Based on this information we strongly encourage that several protective measures be takes to safeguard any host using SMB.

Disable the anonymous enumeration of shares through the use of a Registry setting or by applying all inclusive security templates on Windows 2000 hosts. This kind of activity is also indicative of scans for port 137 looking for machines offering shares through the use of SMB.

Disable the creation of administrative or hidden shares on all Windows NT and 2000 hosts. This can be accomplished through registry entries and then the deletion of the $ shares. This not only will prevent the access to vital system files to unauthorized user, but it will help in protecting the host from Worms like Nimda. It should be noted that no SMB Name Wildcards alerts were recorded originating from external address.

Correlation:
The SANS institute issued an FAQ on port 137 scan, which directly relates to the SMB Name Wildcard alerts received. The FAQ can be found at URL: http://www.sans.org/newlook/resources/IDFAQ/port_137.htm.

spp_http_decode*: CGI Null Byte attack*

44,305 (4.22%) alerts were triggered by traffic identified as using spp_http_decode: CGI Null Byte attack signature. The attacks originated from 34 sources, most of them came from inside the University's network. There were 41 destinations to the attack, with the vast majority directed at host outside the University's network. The table below lists the top 10 sources of the alert generated for the spp_http_decode: CGI Null Byte attack:

Source Hosts	Alerts	% of Total Alerts
MY.NET.153.197	15829	35.73%
MY.NET.153.193	8730	19.70%
MY.NET.153.149	4386	9.90%
MY.NET.153.208	4279	9.66%

MY.NET.153.171	4139	9.34%
MY.NET.153.153	2222	5.02%
MY.NET.153.184	1365	3.08%
MY.NET.152.11	1169	2.64%
MY.NET.153.194	946	2.14%
MY.NET.153.210	627	1.42%

As we can see the overwhelming majority of the sources for the attack is focused on MY.NET.153.0 network segment. Further strengthens our analysis that machines in this network segment are compromised. The destinations of these attacks seem to be focused on a few hosts. The table below lists the top 10 destinations for the attacks.

Destination Hosts	**Alerts**	**% of Total Alerts**
209.10.239.135	26730	60.33%
152.163.210.75	6300	14.22%
207.189.79.124	3792	8.56%
207.189.75.40	2658	6.00%
205.188.132.67	2232	5.04%
216.241.219.22	1169	2.64%
206.61.145.3	402	0.91%
63.162.230.3	384	0.87%
MY.NET.5.96	172	0.39%
216.33.88.141	106	0.24%

It appears that host located at 209.10.239.135 is getting a lot of attention, accounting for 60.33% of all of the spp_http_decode: CGI Null Byte attacks. We decided to do a WHOIS (courtesy of ARIN) look-up for this address and we found it registered to the following ISP:

```
Globix Corporation (NETBLK-GLOBIXBLK3)
   295 Lafayette St- 3rd Fl
   NY, NY 10012
   US

   Netname: GLOBIXBLK3
   Netblock: 209.10.0.0 - 209.11.223.255
   Maintainer: PFMC

   Coordinator:
      Hostmaster, Globix Corporation  (GCH2-ARIN)  arin-
admin@GLOBIX.NET
      +1-212-334-8500 (FAX) 212.334.8615

   Domain System inverse mapping provided by:

   Z1.NS.NYC1.GLOBIX.NET 209.10.66.55
   Z1.NS.SJC1.GLOBIX.NET 209.10.34.55
   Z1.NS.LHR1.GLOBIX.NET 212.111.32.38
```

Based on our analysis we believe that hosts based on the University's network and in particular the MY.NET.153.0 network segment are being used to actively target commercial web sites with spp_http_decode: CGI Null Byte attacks. We feel it is necessary to review the policy of usage for hosts based on University property and address any deficiencies. Furthermore; we recommend that a close eye be kept on the user community operating host on the MY.NET.153.0 network segment and remind them of the University's policy pertaining to hacking.

Correlation:
This exploit is popular with web server bashers looking for a quick way to deface a site or gain some other type of un-authorized access. Incidents.org has recorded similar attacks using this exploit. The handler's reports can be view at http://www.incidents.org/archives/y2k/040301-1300.htm.

ICMP Echo Request L3retriever Ping

33,491 (3.19%) alerts were triggered by traffic identified as using ICMP Echo Request L3retriever ping. This type of alert has been attributed to a scanning tool from L3 Network's (now part of Symantec) called Retriever. Of the alerts recorded there were 164 sources and 15 distinct destination hosts. The origin for all alerts came from hosts inside and being directed to internal hosts. The traffic seems to be either intelligence gathering by someone inside the University network or the system administrator/network engineers are using the Retriever for legitimate uses. The table below lists the top 10 destination for ICMP Echo Request L3retriever Ping.

Destination Hosts	Alerts	% of Total Alerts
MY.NET.11.6	15095	45.07%
MY.NET.11.7	11557	34.51%
MY.NET.11.5	5651	16.87%
MY.NET.5.4	500	1.49%
MY.NET.10.49	414	1.24%
MY.NET.5.92	141	0.42%
MY.NET.5.96	71	0.21%
MY.NET.5.35	41	0.12%
MY.NET.5.119	6	0.02%
MY.NET.53.89	4	0.01%

The destination traffic seems to the focused on hosts around hosts on the MY.NET.11.0 network segment. It would be very advantageous to contact the system administrators responsible for these hosts and inquire about the possibility of any of them running scan/discovery/vulnerability detection tools against these hosts. The top 10 sources for these alerts are listed on the table below.

Source Hosts	Alerts	% of Total Alerts
MY.NET.152.167	880	2.63%
MY.NET.152.168	780	2.33%
MY.NET.152.161	729	2.18%
MY.NET.152.177	724	2.16%
MY.NET.152.166	692	2.07%
MY.NET.152.172	690	2.06%
MY.NET.152.21	656	1.96%
MY.NET.152.171	647	1.93%
MY.NET.152.163	645	1.93%
MY.NET.152.183	641	1.91%

The sources of L3retriever Pings are generated from hosts in the MY.NET.152.0 network segment. Based on our analysis for this type of alerts, we feel that further clarification is need from the system administrators to determine if the ICMP Echo Request need to categorized as hostile in nature.

Correlation:
Similar detects have been seen in the GCIA practical of Edward Peck. His practical can be located at the following URL: http://www.giac.org/practical/Edward_Peck_GCIA.doc.

INFO MSN IM Chat data

This detect is pretty common for a network used by educational institution. The alert signature was triggered by the detection of chat data sent from a MSN instant messenger client. Even though it represents 2.10% of all alerts generated, we feel that this type of traffic is normal. It should be noted of a growing trend to use instant messenger client software as a conduit for delivering socially engineered code to spread malicious code. If the University feels there is a substantial threat, it may be a good idea to circulate a best practices bulleting on the proper and safe use of IM tools.

Correlation:
Again Edward Peck' GCIA practical is a good source of correlation of this type of traffic: http://www.giac.org/practical/Edward_Peck_GCIA.doc.

MISC Large UDP Packet

This detect is largely related to a host directing an oversized crafted UDP packets at his target. This poor-man's denial of service proves to be quite effective if the attacker has plenty of bandwidth available and the target host dose not. The top 10 destination for this type of alert are listed on the table below:

Destination Hosts	Alerts	% of Total Alerts
MY.NET.153.171	5349	31.84%
MY.NET.153.174	3689	21.96%
MY.NET.153.153	2129	12.67%
MY.NET.153.164	1584	9.43%
MY.NET.153.110	1504	8.95%
MY.NET.153.121	780	4.64%
MY.NET.152.183	623	3.71%
MY.NET.153.157	621	3.70%
MY.NET.153.165	260	1.55%
MY.NET.150.215	212	1.26%

The majority of the alerts come attacks directed at hosts located in IP address MY.NET.153.171, MY.NET.174 and MY.NET.153.153. These 3 hosts were the destination for 66.47% of all Large UDP Packets. The top 10 source hosts involved in this type of attack are listed in the table below:

Source Hosts	Alerts	% of Total Alerts for this Signature
63.240.15.205	2129	12.67%
61.78.35.42	2106	12.54%
61.78.35.44	2027	12.07%
210.94.0.146	1584	9.43%
163.239.2.31	1504	8.95%
216.106.173.144	1474	8.77%
216.106.173.150	1295	7.71%
63.240.15.207	1216	7.24%
216.106.173.146	920	5.48%
211.115.206.105	780	4.64%

Even though this type of attack is usually performed using a spoofed IP address. It should be noted that 24.60% of all attacks originated from IP located in the 63.78.35.0 Class-C network. After doing a WHOIS (courtesy of D-Shield), we discovered that the address space is registered to Korea Telecom.

```
IP Address          : 61.78.32.0-61.78.35.255
Connect ISP Name    : KORNET
Connect Date        : 20010703
```

```
Registration Date  : 20010718
Network Name       : KORNET-IDC-JUNGANG-KTIDC

[ Organization Information ]
Orgnization ID     : ORG203787
Name               : CENTRAL DATA COMMUNICATION OFFICE
State              : SEOUL
Address            : 128-9 YEUNKEONDONG JONGROKU
Zip Code           : 110-460

[ Admin Contact Information]
Name               : GilSoon Park
Org Name           : KOREA TELECOM
State              : SEOUL
Address            : 128-9 Youngundong Chongroku
Zip Code           : 110-460
Phone              : +82-2-747-9213
Fax                : +82-2-766-5901
E-Mail             : gspark@kornet.net
```

Correlation:
This activity as been noted on several GIAC student practicals.

High port 65535 udp - possible Red Worm – traffic

14,653 (1.40%) alerts were triggered by traffic identified as possible Red Worm – traffic. This detect is common is an environment that is easily subjected to the introduction of Trojans or other type pf malicious code. The table below lists the top 10 hosts, which were the source for the most alerts.

Source	# of Alerts	% of Total Alerts for this Signature
MY.NET.6.48	3563	24.32%
MY.NET.6.49	3085	21.05%
MY.NET.6.52	2595	17.71%
MY.NET.6.50	2279	15.55%
MY.NET.6.51	891	6.08%
MY.NET.6.53	444	3.03%
MY.NET.6.60	317	2.16%
MY.NET.6.45	167	1.14%
64.124.157.16	144	0.98%
MY.NET.60.43	103	0.70%

As we can see a lot of alerts are being generated from hosts located in the MY.NET.6.48 network segment. We feel it would be a good idea to examine host in that subnet for possible compromise or virus infestation. The top 10 destination for the alerts generated is listed below:

Destination	# of Alerts	% of Total Alerts for this Signature

MY.NET.152.246	400	2.73%
MY.NET.152.251	386	2.63%
MY.NET.152.184	366	2.50%
MY.NET.152.183	356	2.43%
MY.NET.152.165	279	1.90%
MY.NET.152.177	276	1.88%
MY.NET.152.180	240	1.64%
MY.NET.153.202	235	1.60%
MY.NET.152.176	231	1.58%
MY.NET.152.163	219	1.49%

Based of the traffic patterns seen during or period of analysis we have seen elevated Worm/Trojan activity concentrated on the MY.NET.152.0 and MY.NET.153.0 subnets. We strongly encourage the system administrators responsible of those hosts systems to review their configuration, search for signed of compromise or breach, and make sure the anti-virus software is properly installed and up to date.

Correlation:

Virus and Trojan activity is very common in a network used by educational institution. Similar activity has been seen in following student practical:
http://www.giac.org/practical/REUBEN_RUBIO_GCIA.doc
http://www.giac.org/practical/Edward_Peck_GCIA.doc

INFO Inbound GNUTella Connect request

GNUTella is a popular peer-peer file sharing application. This particular alert is triggered when an inbound connection is detected attempting to initiate a peer-peer file transfer session. This kind of file sharing activity is very popular amount students; usually it involved the exchange of music or movies in digital format. They're where 11,680 (1.11%) alerts recorded matching this signature. The top 10 destination hosts are listed on the table below:

Destinations	**# of Alerts**	**% of Total Alerts for this Signature**
MY.NET.153.143	4244	36.34%
MY.NET.153.175	1972	16.88%
MY.NET.153.160	1683	14.41%
MY.NET.153.211	1588	13.60%
MY.NET.153.170	974	8.34%
MY.NET.153.194	631	5.40%
MY.NET.152.164	233	1.99%
MY.NET.153.164	158	1.35%
MY.NET.153.153	89	0.76%

MY.NET.150.209	71	0.61%

We recommend the University review its policy of usage in regards to the use of peer-to-peer distribution software. The use of this software is usually associated with copyright violations. In addition there is an increased risk for the introduction of malicious code (viruses/Trojans) via peer-to-peer exchanges.

Correlation:
Similar detects have been seen in the following student practicals:
http://www.giac.org/practical/Edward_Peck_GCIA.doc
http://www.giac.org/practical/Mike_Poor_GCIA.doc

Other Alerts of Interest

Below is a list of other alerts recorded during the period of analysis. They did not occur in large enough numbers to fall within the top 10 alerts recorded. We feel it's necessary to mention their occurrence with a brief description.

Alert Name	# Of Alerts
ICMP Echo Request Nmap or HPING2	5,664
Description:	
An ICMP Echo request was sent from a host using Nmap or HPING2. Both tools are used for scanning and OS fingerprinting. HPING2 can be used to craft packets.	

Alert Name	# Of Alerts
Watchlist 000220 IL-ISDNNET-990517	4,840
Description:	
This appears to be a custom rule designed to trigger an alert of the NIDS when traffic is detected from an ISP in Israel. Without further background knowledge it is difficult to assess any lethality.	

Alert Name	# Of Alerts
FTP DoS ftpd globbing	4,048
Description:	
This rule is triggered when the NIDS detects a wildcard character request send to an FTP with expecting to cause a denial of service. http://www.whitehats.com/info/IDS487	

Alert Name	# Of Alerts
ICMP Fragment Reassembly Time Exceeded	2,228
Description:	
ICMP error message sent by the destination host to the source, reporting re-assembly time for a fragmented packet has exceeded.	

Alert Name	# Of Alerts
ICMP Router Selection	1,490

Description:

This alert is triggered by the detection of multicast packets from routers making route distribution and announcements.

Alert Name	# Of Alerts
WEB-IIS view source via translate header	1,317

Description:

A source host attempting to view the html source code on the web server triggered the alert. This can lead to a system compromise by revealing backend connectivity embedded into the html code.

Alert Name	# Of Alerts
NMAP TCP ping!	841

Description:

The alert was triggered when a packet matching the signature an NMAP TCP ping was encountered by the NIDS. NMAP is used for scanning networks and performing OS fingerprinting.

Alert Name	# Of Alerts
WEB-MISC Attempt to execute cmd	723

Description:

The NIDS detected a HTTP traffic attempting to execute the command executive of the NT4.0 or Windows 2000 machine. The exploit is being attempted through the use of a well-known vulnerability, known as directory transversal.

Alert Name	# Of Alerts
INFO Outbound GNUTella Connect request	546

Description:

An outbound connection request has detected by an internal host running the peer-to-peer file sharing software GNUTella.

Alert Name	# of Alerts
WEB-IIS _vti_inf access	322

Description:

An alert was triggered when the NIDS detect traffic attempting to exploit a well-known vulnerability found in FrontPage extensions.

Alert Name	# of Alerts
Watchlist 000222 NET-NCFC	320

Description:

An alert was triggered when the NIDS detect traffic whose source address comes from 159.226.0.0 class B network. This network space is registered to The Computer Network Center Chinese Academy of Sciences

Alert Name	# of Alerts
ICMP Echo Request Windows	301

Description:
An alert was triggered when the NIDS detected ICMP Echo Request from a host running the Windows family of operating systems; this is mostly an informational alert.

Alert Name	**# of Alerts**
WEB-FRONTPAGE _vti_rpc access	299
Description:	
An alert was triggered when the NIDS detected an attempt run an exploit against a well-documented vulnerability against a host configured to use FrontPage extensions. This exploit is usually accompanied by other well-documented script attacks directed primarily at hosts running Microsoft IIS.	

Alert Name	**# Of Alerts**
Null scan!	271
Description:	
An alert was triggered when the NIDS detects a TCP scan with no flags set. This is a clearly fabricated packet looking for a predictable response from a host, perhaps this Null scan! was used in conjunction with other intelligence gathering activities / OS fingerprinting.	

Alert Name	**# Of Alerts**
WEB-CGI scriptalias access	158
Description:	
An alert was triggered when the NIDS detected a GET request attempting to exploit a well documented vulnerability of ScriptAlias which would allow the attacker to view the source of CGI scripts (CVE-1999-0236)	

Alert Name	**# of Alerts**
Possible Trojan server activity	138
Description:	
An alert was triggered when the NIDS detected activity from a Trojan server possibly operating in the internal network. 46 instances of this activity came from host MY.NET.5.83 using destination TCP port 27374 (SubSeven).	

Top 10 Hosts – Alerts

During the period of analysis there were 1,049,957 alerts recorded from 10,125 distinct hosts, of which 9,563 hosts were located outside the University's network space. The distribution of the alerts very askew, 996,258 (94.89%) of all alerts triggered came from an internal host. The table below lists the top 10 hosts, which were the source for the most alerts recorded.

Source	**#of Alerts**	**% of Total Alerts Recorded**
MY.NET.150.83	299723	28.55%

MY.NET.153.164	76134	7.25%
MY.NET.153.118	57453	5.47%
MY.NET.153.126	28181	2.68%
MY.NET.153.119	18217	1.74%
MY.NET.153.197	16880	1.61%
MY.NET.11.6	15052	1.43%
MY.NET.70.177	12354	1.18%
MY.NET.153.113	11893	1.13%
MY.NET.11.7	11501	1.10%

As it can be seen there are no external hosts in the top 10 list. We felt it necessary to build a separate top 10 list for external hosts. This way we could discern the external threats and better assess the University's exposure from the outside. The table below lists the top 10 external source hosts.

Source	#of Alerts	% of Total External Alerts
63.240.15.205	2129	3.96%
61.78.35.42	2106	3.92%
61.78.35.44	2027	3.77%
212.179.35.118	1890	3.52%
210.94.0.146	1584	2.95%
163.239.2.31	1504	2.80%
216.106.173.144	1474	2.74%
216.106.173.150	1297	2.42%
212.179.40.132	1285	2.39%
63.240.15.207	1216	2.26%

Of the top 10 external hosts responsible for alerts, eight were involved is sending exclusively MISC Large UDP Packets. We feel further analysis would lead to a dead-end, given the ease of IP spoofing when used in conjunction with UDP traffic. There were two hosts which figured in the top 10, which we feel should be looked more carefully. Hosts using IP address 212.179.35.118 and 212.179.40.132 triggered alerts for Watchlist 000220 IL-ISDNNET-990517. We performed WHOIS using the RIPE and gathered the following host information:

http://www.ripe.net/ripencc/pub-services/db/copyright.html

Host 212.179.35.118 - bzq-179-35-118.dcenter.bezeqint.net

```
inetnum:       212.179.35.96 - 212.179.35.127
netname:       EPLICATION-LTD
mnt-by:        INET-MGR
descr:         EPLICATION-LTD-HOSTING
country:       IL
```

```
admin-c:       ZV140-RIPE
tech-c:        MZ4647-RIPE
status:        ASSIGNED PA
notify:        hostmaster@isdn.net.il
changed:       hostmaster@isdn.net.il 20020312
source:        RIPE
route:         212.179.0.0/17
descr:         ISDN Net Ltd.
origin:        AS8551
notify:        hostmaster@isdn.net.il
mnt-by:        AS8551-MNT
changed:       hostmaster@isdn.net.il 19990610
source:        RIPE
person:        Zehavit Vigder
address:       bezeq-international
address:       40 hashacham
address:       petach tikva 49170 Israel
phone:         +972 52 770145
fax-no:        +972 9 8940763
e-mail:        hostmaster@bezeqint.net
nic-hdl:       ZV140-RIPE
changed:       zehavitv@bezeqint.net 20000528
source:        RIPE
```

Host 212.179.40.132 - station-131.gadot.org.il

```
inetnum:       212.179.40.128 - 212.179.40.255
netname:       KIBBUTZ-GADOT
descr:         KIBBUTZ-GADOT-LAN
country:       IL
admin-c:       ZV140-RIPE
tech-c:        NP469-RIPE
status:        ASSIGNED PA
notify:        hostmaster@isdn.net.il
mnt-by:        RIPE-NCC-NONE-MNT
changed:       hostmaster@isdn.net.il 20001015
source:        RIPE
route:         212.179.0.0/17
descr:         ISDN Net Ltd.
origin:        AS8551
notify:        hostmaster@isdn.net.il
mnt-by:        AS8551-MNT
changed:       hostmaster@isdn.net.il 19990610
source:        RIPE
person:        Zehavit Vigder
address:       bezeq-international
address:       40 hashacham
address:       petach tikva 49170 Israel
phone:         +972 52 770145
fax-no:        +972 9 8940763
e-mail:        hostmaster@bezeqint.net
nic-hdl:       ZV140-RIPE
changed:       zehavitv@bezeqint.net 20000528
source:        RIPE
```

We feel the University's network administrators should verify if these hosts are participating in sanctioned and legitimate network access. If they are not part of the approved list of external partners, the University should contact the ISP at the following email address: mailto:abuse@bezeqint.net and report the abuse.

Other External Hosts of Interest

We feel that some external hosts should be highlighted not for the amount of alerts generated by them, but for the lethality and selectiveness of the exploits they have used against University based hosts. We have included the ISP contact information, and we feel the Network Administrators should make them aware of the abuse.

HOST 68.50.252.86

Host IP	DNS Name
68.50.252.86	pcp01719950pcs.nrockv01.md.comcast.net

Attack Signature	# of Alerts
WEB-FRONTPAGE _vti_rpc access	23
WEB-IIS _vti_inf access	24

Primary Target
MY.NET.5.96
Description:

This attacker has singled-out MY.NET.5.96 for exploit. We feel the selectiveness by the attacker and particular exploit used against the target merit notification to the ISP.

WHOIS Information: Host 68.50.252.86

(Courtesy of ARIN & D-Shield)

```
Comcast Cable Communications, Inc. (NETBLK-JUMPSTART-1)
   3 Executive Campus, 5th Floor
   Cherry Hill, NJ 08002
   US

   Netname: JUMPSTART-1
   Netblock: 68.32.0.0 - 68.63.255.255
   Maintainer: CMCS

   Coordinator:
      Zeibari, Greg  (GZ64-ARIN)  gzeibari@comcastpc.com
      856-661-7929

   Domain System inverse mapping provided by:

   NS01.JDC01.PA.COMCAST.NET   66.45.25.71
   NS02.JDC01.PA.COMCAST.NET   66.45.25.72

   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

   Record last updated on 15-Jan-2002.
   Database last updated on  3-May-2002 20:01:14 EDT.

----------

Comcast Cable Communications, Inc. (NETBLK-JUMPSTART-DC-1)
   1107 Ritchie Rd.
   Capitol Heights, MD 20743
   US

   Netname: JUMPSTART-DC-1
   Netblock: 68.48.0.0 - 68.50.255.255
   Coordinator:
      Zeibari, Greg  (GZ64-ARIN)  gzeibari@comcastpc.com
      856-661-7929

   Domain System inverse mapping provided by:

   NS01.JDC01.PA.COMCAST.NET   66.45.25.71
   NS02.JDC01.PA.COMCAST.NET   66.45.25.72

   Record last updated on 22-Feb-2002.
   Database last updated on  3-May-2002 20:01:14 EDT
```

HOST 68.50.77.68

Host IP	DNS Name
68.50.77.68	pcp702357pcs.bowie01.md.comcast.net
Attack Signature	**# of Alerts**
WEB-FRONTPAGE _vti_rpc access	17
WEB-IIS _vti_inf access	13
Primary Target	
MY.NET.5.96	
Description:	

This attacker has singled-out MY.NET.5.96 for exploit. The attack is coming from the same ISP, a similar dialer pool and close Geographical proximity (Maryland). We feel that these to attacks might be related. We feel the University should contact COMCAST and make them aware of the abuse.

WHOIS Information: Host 68.50.77.68

(Courtesy of ARIN & D-Shield)

```
Comcast Cable Communications, Inc. (NETBLK-JUMPSTART-1)
   3 Executive Campus, 5th Floor
   Cherry Hill, NJ 08002
   US

   Netname: JUMPSTART-1
   Netblock: 68.32.0.0 - 68.63.255.255
   Maintainer: CMCS

   Coordinator:
      Zeibari, Greg  (GZ64-ARIN)  gzeibari@comcastpc.com
      856-661-7929

   Domain System inverse mapping provided by:

   NS01.JDC01.PA.COMCAST.NET   66.45.25.71
   NS02.JDC01.PA.COMCAST.NET   66.45.25.72

   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

   Record last updated on 15-Jan-2002.
   Database last updated on  3-May-2002 20:01:14 EDT.

----------

Comcast Cable Communications, Inc. (NETBLK-JUMPSTART-DC-1)
   1107 Ritchie Rd.
   Capitol Heights, MD 20743
   US
   Netname: JUMPSTART-DC-1
   Netblock: 68.48.0.0 - 68.50.255.255

   Coordinator:
      Zeibari, Greg  (GZ64-ARIN)  gzeibari@comcastpc.com
      856-661-7929

   Domain System inverse mapping provided by:

   NS01.JDC01.PA.COMCAST.NET   66.45.25.71
   NS02.JDC01.PA.COMCAST.NET   66.45.25.72

   Record last updated on 22-Feb-2002.
   Database last updated on  3-May-2002 20:01:14 EDT
```

HOST 207.172.11.147

Host IP	DNS Name
207.172.11.147	cache-1.lnh.md.webcache.rcn.net
Attack Signature	**# of Alerts**
WEB-CGI ksh access	74
WEB-FRONTPAGE _vti_rpc access	4
WEB-IIS _vti_inf access	4
WEB-IIS view source via translate header	2
IDS475/web-iis_web-webdav-propfind	1
Primary Target	
MY.NET.5.96	
Description:	

This attacker has singled-out MY.NET.5.96 for exploit with a variety of exploits. We feel that this was deliberate attempt to compromise a Web server on the University network. We recommend the network administrators contact the ISP and report the abuse.

WHOIS Information: Host 207.172.11.147

(Courtesy of ARIN & D-Shield)

```
RCN Corporation (NET-RCN-BLK-2)
   105 Carnegie Center
   Princeton, NJ 08540
   US

   Netname: RCN-BLK-2
   Netblock: 207.172.0.0 - 207.172.255.255
   Maintainer: RCN

   Coordinator:
      RCN Corporation  (ZR40-ARIN)  noc@rcn.com
      888-972-6622

   Domain System inverse mapping provided by:

   AUTH1.DNS.RCN.NET           207.172.3.20
   AUTH2.DNS.RCN.NET           206.138.112.20
   AUTH3.DNS.RCN.NET           207.172.3.21
   AUTH4.DNS.RCN.NET           207.172.3.22

   Record last updated on 04-Apr-2001.
   Database last updated on  3-May-2002 20:01:14 EDT
```

Top 10 Hosts – Scans

During the period of analysis, 3,523,821 portscans recorded from 861 distinct hosts; of which 346 hosts were located outside the University's network space. The distribution of the alerts is as follows, 3,399,895 (96.48%) were generated from source hosts within the University's network, only 123,926 (3.52%) originated from an external host. The table below lists the top 10 hosts, which were the source for the most alerts recorded.

Source IP Address	# of Alerts	% of Total Port Scans
MY.NET.60.43	462,096	13.11%
MY.NET.150.143	283,592	8.05%
MY.NET.6.45	196,947	5.59%
MY.NET.6.48	181,565	5.15%
MY.NET.6.49	179,920	5.11%
MY.NET.6.52	168,898	4.79%
MY.NET.6.50	136,484	3.87%
MY.NET.11.8	88,843	2.52%
MY.NET.6.53	83,955	2.38%
MY.NET.6.60	72,094	2.05%

As it can be seen there is not a single external host listed in the top 10 scanner list. We feel that it is necessary to create a separate top 10 table for scanning hosts outside the University's network. This will provide a better picture of the amount of reconnaissance and intelligence gathering being performed against the University.

Source IP Address	# of Alerts	% of Total External Port Scans
64.124.157.16	14867	12.00%
64.124.157.10	4860	3.92%
205.188.228.33	3560	2.87%
66.28.225.156	3314	2.67%
64.124.157.64	3272	2.64%
64.232.138.142	3251	2.62%
66.28.8.69	3033	2.45%
205.188.228.129	3001	2.42%
63.250.219.154	2812	2.27%
66.28.14.37	2798	2.26%

Based of our analysis we have been drawn to scans originating from hosts from the 64.124.157.0 network. Not only three of them are in the top 10 list, but they have been engaged in UDP scans of the University's network. We believe that these hosts might be trolling for Trojans and in particular looking for Red-Worm infected hosts. Below is an excerpt from the scan logs of April 5th.

```
1. 04/05-16:17:13.909429  [**] High port 65535 udp - possible Red
   Worm - traffic [**] 64.124.157.10:65535 -> MY.NET.153.45:65280

2. 04/05-16:31:59.363842  [**] spp_portscan: portscan status from
   64.124.157.10: 3 connections across 1 hosts: TCP(0), UDP(3) [**]
```

We feel that these hosts pose a threat to the stability of the university's computer network. We performed a WHOIS using the ARIN database and came up to with the following host information. We believe the University's network administrators should contact the ISP below and inform them of the activity originating from the 64.124.157.0 network.

```
Abovenet Communications, Inc. (NETBLK-ABOVENET)
   50 W. San Fernando Street, Suite 1010
   San Jose, CA 95113
   US

   Netname: ABOVENET
   Netblock: 64.124.0.0 - 64.125.255.255
   Maintainer: ABVE

   Coordinator:
        Metromedia Fiber Networks/AboveNet  (NOC41-ORG-ARIN)
        noc@ABOVE.NET
        408-367-6666
        Fax- 408-367-6688

   Domain System inverse mapping provided by:

   NS.ABOVE.NET                 207.126.96.162
   NS3.ABOVE.NET                207.126.105.146

   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

   Record last updated on 27-Apr-2001.

   Database last updated on  2-May-2002 19:58:54 EDT.
```

Correlation:
Red-Worm activity has been reported in several GIAC practicals:

James Conz practical http://www.giac.org/practical/James_Conz_GCIA.doc

Out-Off Spec. Packets

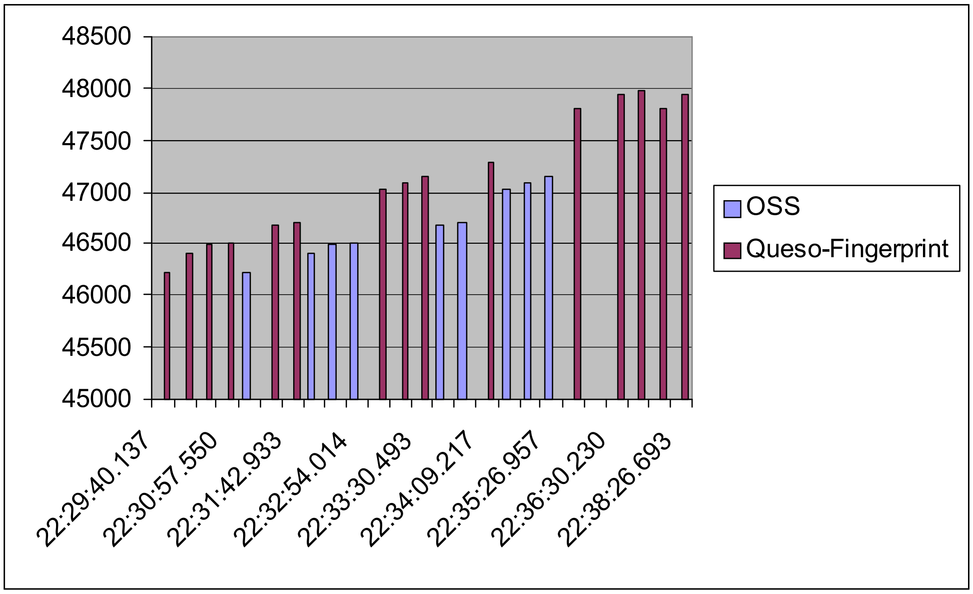
During the period of analysis we encountered 57 packets that were classified as out-of-specification. These are packets, that for one reason or and other, did not comply with the specification as outlined in RFCs from the IETF for IP traffic. We complied a list of the top 10 hosts that were the source of the OOS packets.

Source IP	# of OOS Packets	% of Total OOS Packets
217.80.78.17	13	22.81%
202.153.244.62	10	17.54%
142.51.44.123	7	12.28%
192.115.135.8	5	8.77%
211.37.21.179	3	5.26%
24.141.97.182	3	5.26%
24.83.3.75	3	5.26%
193.2.132.70	1	1.75%
205.251.182.200	1	1.75%
209.176.66.227	1	1.75%

After further analysis and some correlation of the alerts received. We determined that at time host 202.153.244.62 was performing a Queso-Fingerprint scan, the 10 out-of-spec packets attributed to host 202.153.244.62 occurred. The target of the attack/scan was host MY.NET.150.83. The table below shows the relation between the OSS packets and the Queso fingerprint scan to host MY.NET.150.83.

Time	Alert Type	Source Port #
22:29:40.137	Queso-fingerprint	46211
22:30:16.164	Queso-fingerprint	46411
22:30:53.970	Queso-fingerprint	46488
22:30:57.550	Queso-fingerprint	46509
22:31:36.601	OSS	46211
22:31:40.693	Queso-fingerprint	46686
22:31:42.933	Queso-fingerprint	46699
22:32:12.628	OSS	46411
22:32:50.433	OSS	46488
22:32:54.014	OSS	46509
22:33:05.686	Queso-fingerprint	47038
22:33:16.126	Queso-fingerprint	47087
22:33:30.493	Queso-fingerprint	47137
22:33:37.157	OSS	46686
22:33:39.396	OSS	46699
22:34:09.217	Queso-fingerprint	47288
22:35:02.149	OSS	47038
22:35:12.590	OSS	47087
22:35:26.957	OSS	47137

22:35:51.180	Queso-fingerprint	47798
22:36:05.681	OSS	47288
22:36:30.230	Queso-fingerprint	47947
22:36:51.984	Queso-fingerprint	47985
22:37:47.644	OSS	47798
22:38:26.693	OSS	47947
22:38:48.447	OSS	47985



The graph above illustrates the relation between the Queso scan and the OSS packets detected. As it can be seen port numbers are increasing as the scan proceeds, at the same time the OSS packets fit very continently in the gaps of the alerts generated by Queso-fingerprint. We feel the OSS packets are part of the Queso scan and the only reason they were not detected by NIDS is that the detect rules for Queso are not complete in Snort v1.84 by not incorporating the OSS packets.

We feel there is a strong possibility that this OS fingerprinting and out-of-spec packets received from host 202.153.244.62 are part of reconnaissance of the host, and could be the prelude to a directed attack on host MY.NET.150.83.

Conclusions

Based our security audit of the NIDS alert logs we would like to put forward our recommendations aimed at enhancing the security posture of the University's network. It is our opinion that the practice of defense in depth should be a primary axiom in determining defensive countermeasures used to protect the network assets. We further believe that in a University setting the strongest line of defense lay within the host itself. With this in mind we would like to put forward the following recommendation:

- Make sure that server class machines are not running unnecessary services. Streamline system configurations to reduce the number of listening ports.
- Stay on top of the latest patches and hot fixes for server class machines, especially those running the Windows family of operating systems.
- Any web server should be lock-down according to best practices set forth by CERT® or the SANS institute.
- Make sure that all host have an anti-virus program installed and that the virus definitions are updated on a regular basis.
- For systems that are exposed to the Internet, we recommend that some sort of host-based firewall be employed.
- Any host using SNMP should be reconfigured not to use public as a community string. In addition all systems running SNMP services should be upgraded as soon as possible.
- Enforce a strong password policy on all systems including accounts use to access databases.

If the University uses a Firewall we believe that is should be configured to drop all ICMP echo requests and silenced. As a rule of thumb an access control list should be applied to the external interface to drop all NetBIOS type traffic from entering or leaving the network. The same practice should be considered with SNMP traffic. We feel that some of the IDS trigger rules are superfluous, in particular the "**connect to 515 from inside**" IDS signature. This particular signature was responsible for 636,036 (60.58%) of all alerts recorded, and added a lot of unwanted noise to a very busy IDS system. We recommend the University review the rule-set employed by the IDS devices and trim any informational alerts similar to one mentioned above. We appreciate the opportunity you have given us to review and audit your network.

A Word about the Analysis

The amount of data involved in the analysis was quite daunting. In total five days worth of data amounted to 350GB of alert, scan and OOS logs. We used SnortSnarf v020316.1 to provide the brunt of the statistical data. We also used quite extensively Microsoft's Excel to: format tables, sort and subtotal the information extracted with SnortSnarf. We used a little Shareware tool called Search&Retrieve32, to do quick and dirty searches on the raw data. Not to mention creative uses of Notepad and Grep for data cleansing.

It was very surprising seeing a very powerful server with lots of RAM chewing on data for a couple of hours. The system took about 16 hours of processor time, to come up with the total for Alerts and Port scans. A word to the wise, if you are using SnortSnarf the typical alert file with need about 470MB of RAM to process (multiply by 5 = 2.35GB of RAM!!) and your port scan files will need 1.45GB of RAM to process (multiply by 5 = 7.25GB of RAM!!).

Resources:

http://www.cert.org/
http://www.cve.mitre.org/
http://www.giac.org/
http://www.incidents.org/
http://packetstorm.dnsi.info/
http://www.securityfocus.com/
http://www.whitehats.com/