Global Information Assurance Certification Paper

GIAC Certified Intrusion Analyst

Christopher R. Hetner

Practical assignment Version 3.1

Table of Contents

Assignment 1 – Describe the State of Intrusion Detection

How Intrusion Detection Technologies Enhance Security

Abstract

Intrusion Detection Systems (IDS) are in an emerging state and considered an integral component of the layered information security model. They offer a more in depth approach for system administrators to perform analysis, detection, monitoring, and response as it pertains to compromising TCP/IP packets. The underlining packet capturing technologies, audit processes and attack signature databases are what create the IDS as a functional information security entity. However, the lack of proliferation, poor implementation and monitoring methodologies, and improper placement will create a false sense of security. This paper will address current attack patterns, IDS technologies, and guidance as to effectively implement an IDS.

Information Security Attacks and Trends

Information system attacks occur throughout all types of organizations, which include educational institutions, healthcare, financial, insurance, and governments. Attack methodologies will differ based on the systems that are targeted. However, the sophistication and lethality of attacks are becoming more challenging to thwart and detect. Many organizations create a false sense of security by simply installing a firewall to permit and block certain Internet traffic or Virtual Private Networks (VPNs) to encrypt and protect sensitive date. Firewalls and VPNs do provide layers of necessary security but they fall short in distinguishing between malicious and authorized traffic, thus with a lack of comprehensive analysis, a compromise of information is inevitable. The following statistics help illustrate the need for the appropriate placement of an IDS in an organization:

April 7, 2002: *Based on responses from 503 computer security practitioners in U.S. corporations, government agencies, financial institutions, medical institutions and universities, the findings of the "2002 Computer Crime and Security Survey" confirm that the threat from computer crime and other information security breaches continues unabated and that the financial toll is mounting.*
Highlights of the "2002 Computer Crime and Security Survey" include:

- *Ninety percent of respondents (primarily large corporations and government agencies) detected computer security breaches within the last twelve months.*
- *Eighty percent acknowledged financial losses due to computer breaches.*
- *Forty-four percent (223 respondents) were willing and/or able to quantify their financial losses. These 223 respondents reported $455,848,000 in financial losses. As in previous years, the most serious financial losses occurred through theft of proprietary information (26 respondents reported $170,827,000) and financial fraud (25 respondents reported $115,753,000).*
- *For the fifth year in a row, more respondents (74%) cited their Internet connection as a frequent point of attack than cited their internal systems as a frequent point of attack (33%).*

- *Thirty-four percent reported the intrusions to law enforcement. (In 1996, only 16% acknowledged reporting intrusions to law enforcement.)* [1]

October 7, 1999: *Hackers apparently working from Russia have systematically broken into Defense Department computers for more than a year and took vast amounts of unclassified but nonetheless sensitive information, U.S. officials said Wednesday. Besides penetrating the Pentagon's defenses, the hackers have raided unclassified computer networks at Energy Department nuclear weapons and research labs, at the National Aeronautics and Space Administration and at many university research facilities and defense contractors, officials said.* " [2]

June 1, 1999: *After NATO jets hit the Chinese Embassy in Belgrade in May, hackers from China attacked a handful of U.S. government sites, including one maintained by the Energy Department. In an unrelated incident, the official White House site was shut down briefly because of an attempt to tamper with it by unidentified hackers, officials said.* Reuters. *White House Threatens to Punish Hackers* [online]. [3]

April 6, 1999: *The nation's three nuclear weapons labs have shut down their classified computer systems for at least a week to beef up network security. Three preeminent Energy Department facilities halted operations Friday on all computers that handle secret information, in response to an unfavorable information security rating in a DOE audit of last year, according to Los Alamos National Laboratory spokesman Jim Danneskiold. The other two labs affected by the shutdown are Lawrence Livermore National Laboratory and Sandia National Laboratories...All three facilities will undertake several initiatives to improve security, including conducting computer security and threat awareness training; devising stricter access policies and tougher enforcement; implementing more rigorous procedures for transferring information from classified to unclassified computers; and establishing new intrusion detection measures.* [4]

March 5, 1999: *The Pentagon today confirmed that attacks against U.S. military computers over the past few months are under special investigation by law enforcement and intelligence authorities. Deputy Defense secretary John Hamre briefed the House Armed Services Committee on the matter in a classified meeting February 23, according to the House Armed Services Committee. He warned legislators that the attackers were not merely individual hackers, and said part of the problem may stem from the cooperation of insiders within the U.S. military staff.... Hamre told the committee that the Pentagon detects between 80 and 100 hacker "events" every day. The Pentagon must investigate approximately one in ten of these.... One security expert said that while attacks from Russian and other foreign nations was nothing new, the new breed of hacks posed grave threats in their sophistication. "There is a steadily increasing number of these attacks," said Alan Paller, director of research for The SANS Institute. "And*

---

[1] Available WWW: http://www.gocsi.com/press/20020407.html
[2] Cyber-theft of Sensitive U.S. Files Traced to Russia." *Chicago Sun-Times.* October 7, 1999.

[3] Available WWW: <URL: http://news.cnet.com/news/0-1005-200343118.html? tag=st.cn.1> (1999).
[4] Shankland, Stephen. (CNET News.com). *U.S. Weapons Labs Shut Down Classified Networks* [online]. Available WWW: <URL: http://news.cnet.com/news/0-1003-200-340847.html? tag=st.ne.1002> (1999).

there are more of these that have three characteristics that set them apart." The first of these is that attacks are coming simultaneously from multiple, coordinated sites. The second is that the attacks are coming with more stealth, escaping the detection of intrusion monitoring systems by limiting the number of "pings," or connections. "These are coming in just under the detection threshold, at one every hour, or every three days," said Paller. "They're coming from patient people, who are usually more professional than children." [5]

The aforementioned security breaches could seriously compromise the United States national security and cost companies billions of dollars. But more importantly, this information could lead to future catastrophic events if in the wrong hands. Military and government facilities apply an in depth defense strategy as it pertains to their physical boundaries. An armed guard at the front gate, cameras, electronic fencing, biometrics, locks, motion and heat sensors, alert systems, 24x7 monitoring stations and smart cards all create layers to enforce this defense approach. The same approach with multiple layers of protection should be applied to the information security objective that includes edge router security, firewalls, access controls, encryption, IDS, intrusion monitoring and alerting, and host based security. There should've been a system in place to measure, detect, and alert once the malicious network activity occurred.

Attacks on information systems are real and increasing. The number of attacks has increased since the early 1990's based on the proliferation of expanded Internet network infrastructure, number of connected hosts, distributed systems, higher and less expensive bandwidth availability, and the use of insecure protocols. Also factoring in the threat of special interest groups, foreign governments, hacking groups, and terrorists, combined with anonymity creates a landscape that will exponentially expand the severity and types of network attacks. Attacks are becoming increasingly complex and sophisticated with the ability to bypass most firewalls and traffic filtering devices. Networks of all types should be in a position to measure, detect, and respond to malicious traffic types. For example, an IDS provides a means to capture raw data packets off the wire and filter the traffic in order to discern any malicious intent. This ability to detect, if strategically positioned, is crucial for providing a real time alert and audit trail system that tracks the malicious activity for analysis.

Cert Coordination Center has been observing malicious Internet activity for over 10 years. The following trends are the most updated trend analysis issued by Cert that justifies why attacks are increasing.

Trends

1. **Automation; speed of attack tools: websites throughout the Internet provide hundreds of downloadable executables and/or precompiled hacking/cracking programs that can be easily installed onto a mid level computer user. The tools in conjunction with higher available bandwidth create a landscape for speed and automation.**

[5] Festa, Paul. (CNET News.com). *Defense Department Fights off Hackers* [online]. Available WWW: <URL: http://news.cnet.com/news/0-1005-200-339584.html?tag=st.ne.1005-200-343118> (1999).

2. **Increasing sophistication of attack tools: the level of sophistication is largely due to the ability to obtain information and open source development initiatives.**
3. **Faster discovery of vulnerabilities: the ongoing demands for enhanced technological features create a competitive environment that is driven by the concept "time to market". Unfortunately hardware and software companies are expediting the development of their product in such a fashion that overlooks the security within the development cycles.**
4. **Increasing permeability of firewalls: firewalls are designed to restrict and permit selective services in and out of the network. Unfortunately the services themselves are being used as a vehicle to compromise systems. (Mobile codes, Active X)**
5. **Increasingly asymmetric threat: because of the interdependence of the Internet, it is non trivial to launch an attack on a victim while using a large number of distributed systems. Therefore making the asymmetric nature a continued threat.**
6. **Increasing threat from infrastructure attacks: these are attacks that broadly affect key components of the Internet. For example; if ISP core routers, IDC network infrastructures, and DNS hosts were compromise it would affect a large number of organizations and users on the Internet.** [6]

IDS Technologies

Without a means to capture and analyze malicious traffic patterns, it is difficult to understand how a system was compromised, which reduces the ability to prevent future attacks. Therefore, capturing intrusion information is an essential component to fill in the gaps within a layered secured information systems. In this section I will discuss the fundamental technologies that make up an IDS and how all the pieces work together. I'll start by explaining the core elements of IDS and compare the differences between network based IDS and host based IDS.

IDS technology is an immature yet rapidly growing technology. It would be considered an emerging market from a business perspective. New companies create a unique edge to their technology and then become absorbed by a larger company. Rapidly evolving attacks make the existing IDS technologies obsolete, which forces companies to constantly patch, redesign, and update their IDS technologies. Therefore, we are facing an outlook that will continue to change because of emerging attacks and competition.

IDS technologies can be broken down into network based IDS and host based IDS. The two disciplines apply the same principles of detecting malicious activity but are focused at different layers of protection. Network IDS sniff the line to capture raw data packets so that it can identify maliciously crafted packets (i.e. either a SYN flood or crafted ICMP activity). Host based IDS rely on operating system and application level activity to discern malicious activities. For instance a host based IDS would be able to create an alert if an intruder was attempting to make changes in the Windows 2000 registry or if IIS was compromised with a Nimda worm that caused unusual activity. Both technologies complement each other when properly positioned to

[6] Cert http://www.cert.org/archive/pdf/attack_trends.pdf

implement an in depth defense strategy. The core principle of in depth defense is to create a layered security approach starting from the physical through the application layer. Each layer deserves its own protective measures.

A network based IDS consists of fundamental core elements that make the system work. The fundamental elements that consist of IDS are:

- Sensor with a promiscuous network interface that sniffs the line and obtains network traffic information
- Filtering subsystem that is targeting specific networks or traffic types
- Secure communications channel from sensor to packet analyzing system
- Attack signature database that can be customized
- Correlation or knowledgebase system to actually discern attacks from packet characteristics
- Monitoring system to present patterns
- Alert system when an intrusion is detected
- Audit log
- Incident response when an intrusion occurs

Network IDS

Network IDS technologies rely on capturing raw data packets to perform its analysis against any malicious activity. The two primary components are sensors and management stations. The sensor is configured with a promiscuous and management interface. It then requires a target area so that it can begin sniffing packets. Sensors are connected to switches that consist of either single or multiple networks (VLANS). It is critical to capture the targeted data stream by mirroring the VLAN to the promiscuous port interface of the connected switch. This will ensure that all VLANS are captured through that port. Once data is being captured it is the responsibility of the sensor to forward all packet information to the monitoring system. The packet data will then be correlated against a predefined attack signature database that will perform a level of intelligence to identify patterns. An alarm will triggered if any pattern matches are made against the attack signature database.

Host IDS

Host IDS are operating system dependant so that the system understands the pattern of compromise relative to its environment. The primary components are host based agents and a management stations. The agent resides on the host which conforms itself to the operating system. Agents are usually installed between the application interfaces and the operating system kernel. The primary purpose for this installation is to identify when system calls made from the application level to the operating system kernel so that it can identify abnormal activities. The agents consist of various severity levels depending on the rule base created. The agent to the management station, which then creates an alarm if abnormal activity is detected, forwards all host-based activities.

Host IDS should be though of as an in the box detection measure. Alerts will occur if someone attempts to edit the Windows 2000 register or if the Nimda worm has infected the host. Unix alerts will be triggered if there was any attempt to modify /etc/passwd file on the local system or manipulating the /etc/hosts.rhosts file which permits users to define other trusted users and hosts. All mentioned activity would create an alert that indicates someone attempting to compromise the host.

Network IDS Advantages

Lower cost of ownership- network IDS are positioned to detect malicious activity at the network level that provides more coverage against multiple hosts. They do not require any software to be installed at the host layer and reduces the amount of resources that would monitor each host. Therefore, network IDS is a lower cost solution for organizations with a wide array of hosts.

Detection of network layer attacks- network IDS analyze raw data packets, which host-based IDS do not. In the event of a Denial of Service (DOS) attack, the network IDS would apprehend the packet stream prior the reaching the host (i.e. TCP Syn flood attempts). This type of preemptive detection yields obvious advantages in creating a secure infrastructure that ultimately is designed to provide system protection.

Evidence retention- network IDS captures real time data packets, which is then forwarded to an out of band monitoring system. This creates a scenario where an attacker could not remove any of the intrusion logs. Host based IDS are susceptible to system compromise, which in most cases the attacker will remove any evidence by deleting intrusion logs.

Detection of unsuccessful attacks network based IDS that are deployed in non-filtered areas of the network will detect all malicious attempts. This data is valuable based on the fact that attempts can be viewed and intelligence applied against such attempts. It is useful to know what malicious network activity is being filtered and reinforces that the firewall is working properly.

Real-time detection and response network IDS detect malicious attacks in real-time. This gives the response team the ability to quickly deploy a preventative solution that perhaps can reduce the amount of damage performed on the protected systems. An attacker may be in the process of performing a DOS attack against a targeted network. Real-time detection will in some cases, based on the functionality of the network IDS, have the ability to shun or block the source IP address of the attacker. Thereby mitigating the destructiveness of the DOS attack.

Host Based IDS Advantages

Verifies success or failure of attacks host-based IDS use system logs that contain events to measure whether an attack was successful or not with greater accuracy and fewer false positives that a network IDS. Network IDS inherently create more false positives than host based IDS. However, host based IDS complement network IDS because the host IDS logs will verify whether the system has been compromised.

Monitors specific system activities host based IDS monitors user and file access activity, including file accesses, change to file permissions, attempts to install new executables and attempts to access privileged services. Host based IDS can also monitor activities that are normally executed only by an administrator. The host based IDS can detect an improper change as soon as it is executed by referencing the operating system logs.

Detects attacks that network basked systems miss host based systems can detect attacks that cannot be viewed by network based IDS. For example, system console attacks cannot be viewed from the network.

Well suited for switched and encrypted environments switches allow for large networks to be segmented into smaller networks (VLANS). As a result, it can be difficult to identify the best locations for deploying a network based IDS to achieve sufficient network coverage. Host based IDS provides greater visibility in a switched environment by residing on as many critical hosts as needed.

Certain types of encryption also present a challenge to network based IDS. Host based IDS do not have this limitation.

No additional hardware required host based IDS are installed on existing hardware systems. No additional hardware is required therefore reducing the cost for deployment.

Lower cost of entry host based IDS are less expensive to implement and deploy that the network IDS sensors.

Active vs. passive approach host based IDS are typically implemented as active tools, whereby implementation of certain security policies in addition to alerting on an intrusion will also stop malicious activity. On the other hand, network based IDS is typically a passive monitoring tool.

Challenges Associated With Intrusion Detection Systems

Changes in IDS technology are becoming more prominent in terms of enhancing its capabilities. It is critical for an organization to clearly define the expectations, strategies and requirements for deployment of IDS technologies. Based upon these critical factors, most organizations do not have the skill sets and resources associated with defining deployment requirements. Therefore, creating challenges in effectively deploying IDS technologies.

False Positive Rates are challenging because it requires the analyst to discern between legitimate and malicious traffic. False positives can overwhelm the analyst by tracking down these alerts while overlooking the true malicious activity.

Network IDS Placement is critical because one cannot just simply place an IDS sensor on a switch. The sensors sniffing interface needs to be installed onto a switch port that is spanned to view all traffic that traverses the switch. Furthermore decisions have to be made where to place the IDS sensor (i.e. public, dmz, or private networks). Many organizations fail to understand the strategies associated with placing the IDS sensors.

IDS Signature Creation methodologies involve a complete understanding of exactly the type of services that are being used on the network. Whether it is SMTP, HTTP, FTP, and NETBIOS traffic. It is critical to survey the environment that one is analyzing and fine-tune the attack signatures to reflect this environment. This methodology is overlooked in many deployments, which creates alerts that are not a true threat to the environment that is being protected.

Attack Signature Updates are critical because of the ongoing complexity and proliferation of attack methodologies. It is not uncommon for an IDS attack signature database to be updated on a monthly basis. Unfortunately, many organizations do not update their attack signature databases and fall behind with regards to identifying new threats.

Monitoring and alerting are critical components to the intrusion detection process because it creates an event that requires action to be taken. Monitoring techniques and processes are becoming more advanced and comprehensive based on the ability to capture logs from various sources. For example, routers, firewalls, hosts, NIDS, HIDS, and applications can all yield critical information as it pertains to intrusion activity. The challenge presents itself when all log sources ultimately have to be normalized and presented in such a view that yields useful information. Correlation analyses against all log sources for a given event provides the analyst with more data to support any derived conclusions. Therefore strengthening the ability to identify malicious activity.

More detail can be found at: Network vs. Host-based Intrusion Detection; A guide to Intrusion Detection Technology http://secinf.net/info/ids/nvh_ids/

IDS Sensor Deployment

IDS sensors are dependant upon capturing real time data packets as they traverse the network infrastructure. Therefore it is essential that all IDS sensors be positioned to capture the targeted networks. It will not work optimally if the sensor is not placed correctly. The challenge is to identify which data networks are most interesting from an intrusion detection perspective. Many deployments are not positioned properly because the sensitivity of the data has not been classified; sensors are not placed correctly, and lack of updated attack signatures.

Sensitivity of data needs to be classified in order to qualify exactly how to position an IDS sensor. Risk assessments should be performed against the protected infrastructure in order to quantify data sensitivity. The results will provide a clear understanding as to how the sensors should be deployed.

Outside the Firewall

IDS sensors positioned on the public facing side of the firewall can yield interesting information. Placing IDS outside the firewall allows the sensor to see all attacks coming in from the Internet or public interface. If the attack is TCP based and the firewall blocks the attack, the IDS system

may not be able to detect the attack. Many attacks can be detected only by matching a string signature. The string is not sent unless the TCP three-way handshake is completed. [7]

Although a sensor outside the firewall cannot detect some attacks, this is the best sensor location to detect attacks. The benefit to the site is that analysts can see the kinds of attacks to which their site and firewall are exposed. It also gives a measurement as to what the firewall is blocking. An attack sequence may occur for some time while the firewall is denying all inbound attempts towards the protected hosts. IDS sensors placed outside the firewall would yield information pertaining to the attack sequence. This information will help better understand attack methodologies, targeted attempts, and the effectiveness of applied protective measures.

A common method of hacking is called footprinting, which is the means to perform reconnaissance against the targeted network. The hacker must harvest a wealth of information to execute a focused attack. As a result, attackers will gather as much information as possible about all aspects of an organization's security posture. This network reconnaissance action is the essential component to a successful attack. A countermeasure that is usually employed to thwart and identify reconnaissance probes is the positioning of IDS sensors outside the firewall. [8]

Reconnaissance Techniques and Purposes

The following are reconnaissance techniques and purposes. These attacks should be detected by an outside the firewall IDS sensor in order to gain intelligence to better position a defense strategy. These methods are achieved by manipulating the TCP/IP packets through tools such as NMAP and Fscan:

Techniques

TCP Connect Scan: this is the most basic form of TCP scanning. It is used to open a connection to every listening port on the machine and does not require any special privileges.

TCP SYN Scan: this technique is often referred to as "half-open" scanning, because you don't open a full TCP connection. A SYN packet is sent, as if you are going to open a real connection and you wait for a response. A SYN| ACK indicates the port is listening. A RST/ACK indicates that the port is not listening.

Stealth FIN Scans: this scan will sometimes bypass firewall and packet filters because they primarily look for the SYN packet for applying filtering rules. The FIN scan uses a bare FIN packet as a probe, which will usually surprise the host and any filtering device.

---

[7] Stephen Northcutt, Judy Novak. Network Intrusion Detection An Analysts Handbook, Second Edition. Indianapolis: New Riders, 2000. 157-159

[8] Stephen Northcutt, Judy Novak. Network Intrusion Detection An Analysts Handbook, Second Edition. Indianapolis: New Riders, 2000. 157-159

ICMP Scanning: sending icmp echo request packets to every IP address on the network will indicate which hosts are alive.

UDP Scans: this method is used to determine which UDP ports are open on a host. The technique is to send 0 byte UDP packets to each port on the target machine. If an ICMP port unreachable message is received, then the port is closed, otherwise it is open.

IP protocol Scan: this method is used to determine which IP protocols are supported on a host. The technique is to send raw IP packets without any further protocol header to each specified protocol on the target machine. If an ICMP port unreachable message is received, then the port is closed, otherwise it is open.

ACK Scan: this advanced method is usually used to map out firewall rule sets. In particular, it can help determine whether a firewall is stateful or just a simple packet filter that blocks incoming SYN packets. This scan type sends an ACK packet (with random looking acknowledgement/ sequence numbers) to the ports specified. If a RST comes back, the port is classified as unfiltered. If nothing comes back then the port is filtered.

Purposes

TCP/IP fingerprinting: this method is designed to reveal the identity of the remote host. Subtleties are in the underlying operating system network stack are compared to a known database to determine OS.

Fragmentation: fragmentation occurs when an IP datagram traveling on a network has to traverse a network with a MTU that is smaller than the size of the datagram. Although fragmentation is normal, it is possible to craft fragments for the purposes of avoiding detection by packet filtering devices and intrusion detection systems. The idea is to split the tcp header over several packets.

IP Spoofing: spoofing is a method by which the scanner conceals its source identity with a falsified IP address. Signs of a spoofed source ip are the presence of the RST flag in the packet. A 3-way handshake needs to occur between the source and destination ip address. 1- Syn…. 2- Syn-Ack… 3- Ack… If #3 has a RST flag it is indicative that the source ip address didn't expect the Syn-Ack, didn't initiate the handshake and naturally responded with a RST.

Network mapping: this is the ability to identify with all hosts on a network that are alive and will resolve host names.

Banner Grabbing: this is the ability to discern information about a host by performing specific scans against the application.

Crypto Scanning: IPSec scanning would target port UDP 500 for ISAKMP activity to possibly sniff key exchange. Also scan vpn devices for keys, encryption methods, and rules.

Enumeration is the process of extracting valid account, netbios and exported resource names through active connection systems. (Network resources, users, groups, applications and banners). This can be classified as another form of reconnaissance with a more targeted effort in obtaining system information. IDS sensors placed outside the firewall can capture these attempts. The following are examples of enumeration activity:

Null Sessions: create an unauthenticated session against a target server to obtain network information, shares, users, groups, and registry keys.

NetBIOS Enumeration: once access to the target host, the next step is to interrogate the network through net view to find other targets on the same wire. This is a great method to exploit shares against the target host.

SNMP Enumeration: SNMP agents are typically installed on machines to provide system information and the SNMP agents are accessible through community strings. Since SNMP v1, with very limited security functions is still the prevalent version of SNMP implemented in the networks today, identifying a device with SNMPv1 agent provides a vast source of information for hacks.

Windows 2000 Active Directory Query: all user accounts can be queried.

CGI Abuses: All interactive web sites (e-commerce and dynamic) use some form of scripting to communicate and produce the output. (i.e. /cgi-bin/homepage.pl?user=ray runs the program homepage.pl in order to generate content specific to the user "ray". Attackers can exploit these types of scripts in order to augment malicious activity.

The previous reconnaissance methods and purposes would yield interesting information as a result of IDS strategic deployments and proper logging. For instance, NMAP scanning activity should be detected by a properly positioned sensor, which will result in the ability to take corrective action against the source of such scanning. One common method is to have the sensor write a dynamic access list to the edge router in order to shun or block the source scanning IP address. The understanding of these methods and purposes supported with the appropriate actions make for an environment that is taking a proactive step in securing the infrastructure.

Inside the Firewall

Another strategic placement of an IDS sensor is inside the firewall. There are many reasons for this placement strategy. One being if the attacker can find the sensor outside the firewall, s/he may attack it so that there is less chance of his/her activities being audited. An IDS inside the firewall presents less vulnerability through added protection than systems outside the firewall. Inside the firewall IDS sensors can also yield information pertaining to packets that have been accepted through the firewall as well as packets decrypted by the firewall in case of VPN connections, which can create a baseline as to the effectiveness of filtering. A firewall may be

configured to accept inbound TCP HTTP access against a web server. However, the HTTP request could contain a malicious string, which will ultimately compromise the protected server. A URL request may appear to be legitimate though the true motive is an IIS compromise (i.e. **http://x.x.x.x:/msadc/.. %255c../..%255c../..%255c/..%c1%1c../..%c1%1c.. /..%c1%1c../winnt/system32/cmd.exe?/c+dir**). This URL string is a clear attempt to bypass Microsoft Windows system32 root directory and launch an executable. It appears to be the Nimda Worm signature. Firewalls typically will not view this as a malicious attempt, however an inside the firewall IDS sensor would capture, log, and trigger an alarm based on this event.

Compromised servers within the protected network can cause havoc by spreading its malicious code throughout the network. IDS sensors within the protected network will capture this activity and reduce the amount of damage through a layered alerting system. I've implemented IDS sensors on numerous network segments behind a series of firewalls to maximize traffic auditing. Next generation IDS sensors can be implemented in a chassis based switch. This makes it possible to provide packet-capturing capabilities on the switch media. By mirroring certain ports or virtual local area networks (VLANS) to a single port, the IDS sensor will capture and analyze packets just as if they were on a shared segment.

VPN Gateways

When considering the deployment of IPSec, which is a VPN technology, you are extending the security perimeter of your network to include areas that are not of the autonomous domain and out of the security controls (hotels, remote networks). Intrusion Detection is a technology that is designed to monitor network connections that are exposed to non-trusted networks. The expansion of IPSec will introduce more IDS that will be used to analyze traffic deriving from, or destined to, the IPSec device. IPSec tunnels from remote sites or remote users are less likely to be spoofed traffic because the origins of the traffic are known through symmetric keys, message digests and encryption.

Any attack can be met with a strong response from the IDS that may include a shun or TCP reset. The importance of IDS deployment for IPSec tunnels relies on the fact that all IP traffic is permitted through the tunnel. This setup and reliance on IDS will thwart most of the attacks from remote sites. (i.e. IIS worms and Mail viruses can easily traverse the IPSec tunnel with the assistance of an Active Directory Share.) IDS can also be used after encryption to validate that only encrypted traffic is sent and received by VPN devices. I've also seen instances where IDS sensors are positioned outside the VPN gateway. This will ensure that only IKE and ESP traffic types are destined to the VPN gateway. [9]

[9] http://www.cisco.com/warp/public/cc/so/cuso/epso/sqfr/safev_wp.htm

Conclusion

IDS technologies have proven to be a critical component within the layered security model. They provide a means to detect, measure, analyze and respond to targeted malicious attempts. Most organizations do not have the means and knowledge to effectively deploy IDS technologies. Therefore the lack of proliferation, poor implementation methodologies, and improper placement has created a false sense of security. I've mentioned security compromises through citing incidents that occurred against the Department of Defense and have cost industries billions of dollars. However, recent advances in IDS technologies and methodologies have demonstrated an approach for enhanced security posture that includes log aggregation from sources other that IDS sensors. For example, analysts have found it useful to obtain and analyze logs from routers, firewalls, hosts, and IDS sensors, which can all be correlated in a centralized database to strengthen the intrusion detection process. It is now the responsibility of all IT security professionals to ensure that their organization is positioned to embrace Intrusion Detection Systems.

References

1. Available WWW: http://www.gocsi.com/press/20020407.html

2. Cyber-theft of Sensitive U.S. Files Traced to Russia." *Chicago Sun-Times*. October 7, 1999.

3. Available WWW: <URL: http://news.cnet.com/news/0-1005-200343118.html? tag=st.cn.1> (1999).

4. Shankland, Stephen. (CNET News.com). *U.S. Weapons Labs Shut Down Classified Networks* [online]. Available WWW: <URL: http://news.cnet.com/news/0-1003-200-340847.html? tag=st.ne.1002> (1999).

5. Festa, Paul. (CNET News.com). *Defense Department Fights off Hackers* [online]. Available WWW: <URL: http://news.cnet.com/news/0-1005-200-339584.html?tag=st.ne.1005-200-343118> (1999).

6. Cert http://www.cert.org/archive/pdf/attack_trends.pdf
7. Network vs. Host-based Intrusion Detection; A guide to Intrusion Detection Technology http://secinf.net/info/ids/nvh_ids/
8. Stephen Northcutt, Judy Novak. Network Intrusion Detection An Analyst's Handbook, Second Edition. Indianapolis: New Riders, 2000. 157-159

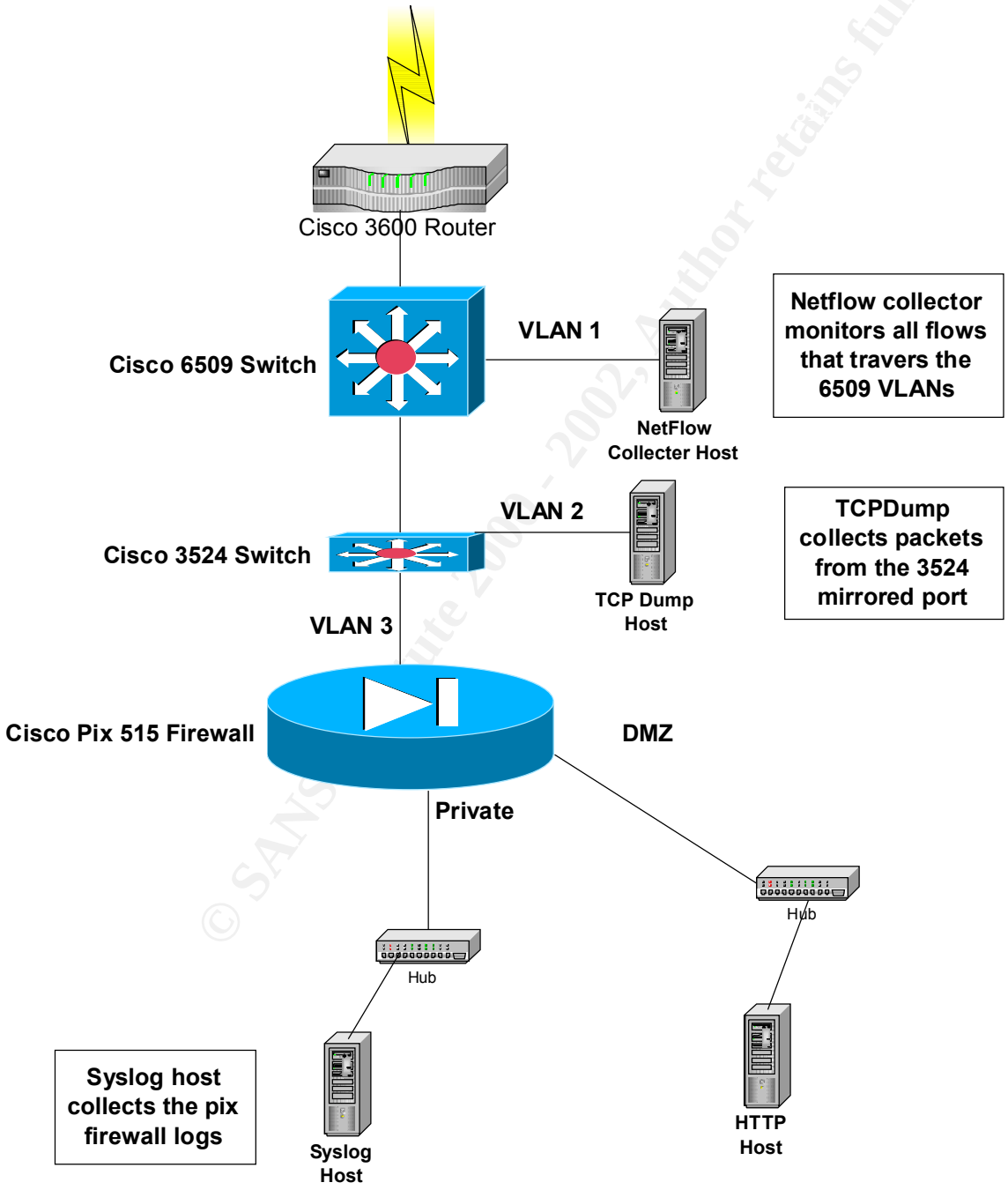
9. http://www.cisco.com/warp/public/cc/so/cuso/epso/sqfr/safev_wp.htm

Assignment 2 – Network Detects

INTRODUCTION

Network Topology:

This is the depiction of my network topology used for assignment 1.



Log Output Description and Format

Cisco Pix Firewall Syslog Format:

The Cisco pix firewall protects both the DMZ and private network resources. All inbound access to hosts is denied as of this time by the firewall rule set. The firewall uses a stateful inspection engine to examine all packets that attempt to traverse it.

More detailed Cisco pix log and code descriptions can be found at www.cisco.com

Log Descriptions:

Dec 31 10:53:31 216.a.b.c %PIX-6-302001: Built inbound TCP connection 1237 for faddr 216.112.x.x/1200 gaddr 216.w.w.w/80 laddr 172.16.2.3/80
This log format is described as date and time, public firewall ip address, pix code, built inbound TCP connection against, source ip address, source port, destination translated ip address, destination port, destination private ip address, and destination port.

Dec 31 10:53:32 216.a.b.c %PIX-6-106015: Deny TCP (no connection) from 216.112.x.x/1200 to 216.w.w.w/80 flags RST on interface outside
This log format is described as date and time, public firewall ip address, pix code, Deny or Permit TCP connection against, source ip address, source port, destination translated ip address, destination port, and the flag sent to the firewall interface which is a Reset.

Dec 31 10:53:33 216.a.b.c %PIX-5-304001: 216.112.x.x Accessed URL 216.w.w.w:/_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
This log format is described as date and time, public firewall ip address, pix code, description of the attempted session, and URL detail.

Dec 31 10:53:33 216.a.b.c %PIX-6-302002: Teardown TCP connection 1237 faddr 216.112.x.x/1200 gaddr 216.w.w.w/80 laddr 172.16.2.3/80 duration 0:00:01 bytes 2877 (TCP Reset-O)
This log format is described as date and time, public firewall ip address, pix code, teardown the TCP connection, source ip address, source port, destination translated ip address, destination port, and the flag sent to the firewall interface, which is a Reset.

TCP Dump Format
TCP Dump is being captured from a Linux host connected to a mirrored switch port.

13:37:15.469464 x.x.x.210.1353 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)
This log format is described as time, source ip address, source port, destination ip address, destination port, flag set, sequence number, bytes in packet, window size, and don't fragment.

Note: All IP addresses in the following traces have been sanitized.

Detect #1 TCP SYN Flood Attack

TCP Dump log output

13:37:15.469464 x.x.x.210.1353 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)
13:37:15.471395 x.x.x.159.1737 > a.b.c.200.109: S 674711609:674711609(0) win 65535 (DF)
13:37:15.474313 x.x.x.19.1712 > a.b.c.200.123: S 674711609:674711609(0) win 65535 (DF)
13:37:15.478826 x.x.x.235.1379 > a.b.c.200.11: S 674711609:674711609(0) win 65535 (DF)
13:37:15.478954 x.x.x.52.1970 > a.b.c.200.137: S 674711609:674711609(0) win 65535 (DF)
13:37:15.481713 x.x.x.13.1374 > a.b.c.200.25: S 674711609:674711609(0) win 65535 (DF)
13:37:15.483662 x.x.x.240.1723 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:15.486561 x.x.x.70.1583 > a.b.c.200.53: S 674711609:674711609(0) win 65535 (DF)
13:37:15.489081 x.x.x.67.1114 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)
13:37:15.491035 x.x.x.191.1956 > a.b.c.200.81: S 674711609:674711609(0) win 65535 (DF)
13:37:15.493474 x.x.x.107.1510 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)
13:37:15.495921 x.x.x.84.1487 > a.b.c.200.109: S 674711609:674711609(0) win 65535 (DF)
13:37:15.501322 x.x.x.214.1027 > a.b.c.200.123: S 674711609:674711609(0) win 65535 (DF)
13:37:15.503355 x.x.x.77.1026 > a.b.c.200.137: S 674711609:674711609(0) win 65535 (DF)
13:37:15.505685 x.x.x.27.1763 > a.b.c.200.25: S 674711609:674711609(0) win 65535 (DF)
13:37:15.506233 x.x.x.54.1110 > a.b.c.200.11: S 674711609:674711609(0) win 65535 (DF)
13:37:15.510580 x.x.x.16.1577 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:15.511456 x.x.x.139.1008 > a.b.c.200.53: S 674711609:674711609(0) win 65535 (DF)
13:37:15.515433 x.x.x.120.1717 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)
13:37:15.517816 x.x.x.136.1775 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)
13:37:15.518413 x.x.x.111.1854 > a.b.c.200.81: S 674711609:674711609(0) win 65535 (DF)
13:37:15.520484 x.x.x.44.1756 > a.b.c.200.109: S 674711609:674711609(0) win 65535 (DF)
13:37:15.523462 x.x.x.142.1948 > a.b.c.200.123: S 674711609:674711609(0) win 65535 (DF)
13:37:15.527580 x.x.x.60.1608 > a.b.c.200.11: S 674711609:674711609(0) win 65535 (DF)
13:37:15.527770 x.x.x.56.1506 > a.b.c.200.137: S 674711609:674711609(0) win 65535 (DF)
13:37:15.530157 x.x.x.222.1198 > a.b.c.200.25: S 674711609:674711609(0) win 65535 (DF)
13:37:15.532980 x.x.x.216.1139 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:15.537895 x.x.x.124.1533 > a.b.c.200.53: S 674711609:674711609(0) win 65535 (DF)
13:37:15.539742 x.x.x.40.1374 > a.b.c.200.81: S 674711609:674711609(0) win 65535 (DF)
13:37:15.539872 x.x.x.121.1044 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)
13:37:15.545249 x.x.x.48.1960 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)
13:37:15.547200 x.x.x.60.1257 > a.b.c.200.109: S 674711609:674711609(0) win 65535 (DF)
13:37:15.550056 x.x.x.210.1541 > a.b.c.200.123: S 674711609:674711609(0) win 65535 (DF)
13:37:15.672153 x.x.x.32.1875 > a.b.c.200.137: S 674711609:674711609(0) win 65535 (DF)
13:37:15.672487 x.x.x.125.1749 > a.b.c.200.11: S 674711609:674711609(0) win 65535 (DF)
13:37:15.674895 x.x.x.214.1919 > a.b.c.200.25: S 674711609:674711609(0) win 65535 (DF)
13:37:15.679357 x.x.x.98.1133 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:15.682413 x.x.x.3.2006 > a.b.c.200.53: S 674711609:674711609(0) win 65535 (DF)
13:37:15.684439 x.x.x.245.1569 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)
13:37:15.687269 x.x.x.243.1695 > a.b.c.200.81: S 674711609:674711609(0) win 65535 (DF)
13:37:15.689193 x.x.x.157.1785 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)

13:37:15.689693 x.x.x.188.1385 > a.b.c.200.109: S 674711609:674711609(0) win 65535 (DF)
13:37:15.694718 x.x.x.124.1450 > a.b.c.200.123: S 674711609:674711609(0) win 65535 (DF)
13:37:15.697159 x.x.x.249.1198 > a.b.c.200.137: S 674711609:674711609(0) win 65535 (DF)
13:37:15.699587 x.x.x.50.1099 > a.b.c.200.11: S 674711609:674711609(0) win 65535 (DF)
13:37:15.701649 x.x.x.76.1722 > a.b.c.200.25: S 674711609:674711609(0) win 65535 (DF)
13:37:15.704089 x.x.x.118.1599 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:15.706505 x.x.x.147.1046 > a.b.c.200.53: S 674711609:674711609(0) win 65535 (DF)
13:37:15.708827 x.x.x.239.1047 > a.b.c.200.81: S 674711609:674711609(0) win 65535 (DF)
13:37:15.708980 x.x.x.131.1723 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)
13:37:15.713731 x.x.x.205.1699 > a.b.c.200.109: S 674711609:674711609(0) win 65535 (DF)
13:37:15.713752 x.x.x.81.1881 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)
13:37:16.001554 x.x.x.106.1376 > a.b.c.200.53: S 674711609:674711609(0) win 65535 (DF)
13:37:16.002162 x.x.x.72.1295 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:16.007049 x.x.x.55.1701 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)
13:37:16.008981 x.x.x.80.1004 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)
13:37:16.009482 x.x.x.132.1470 > a.b.c.200.81: S 674711609:674711609(0) win 65535 (DF)
13:37:16.011392 x.x.x.182.1958 > a.b.c.200.109: S 674711609:674711609(0) win 65535 (DF)
13:37:16.014281 x.x.x.220.1576 > a.b.c.200.123: S 674711609:674711609(0) win 65535 (DF)
13:37:16.016266 x.x.x.120.1486 > a.b.c.200.137: S 674711609:674711609(0) win 65535 (DF)
13:37:16.019502 x.x.x.49.1853 > a.b.c.200.11: S 674711609:674711609(0) win 65535 (DF)
13:37:16.023642 x.x.x.57.1490 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:16.024194 x.x.x.50.1577 > a.b.c.200.25: S 674711609:674711609(0) win 65535 (DF)
13:37:16.026492 x.x.x.14.1717 > a.b.c.200.53: S 674711609:674711609(0) win 65535 (DF)
13:37:16.031001 x.x.x.118.1060 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)
13:37:16.034063 x.x.x.67.1155 > a.b.c.200.81: S 674711609:674711609(0) win 65535 (DF)
13:37:16.035887 x.x.x.77.1266 > a.b.c.200.109: S 674711609:674711609(0) win 65535 (DF)
13:37:16.036002 x.x.x.168.1876 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)
13:37:16.041349 x.x.x.113.1055 > a.b.c.200.123: S 674711609:674711609(0) win 65535 (DF)
13:37:16.043445 x.x.x.146.1342 > a.b.c.200.137: S 674711609:674711609(0) win 65535 (DF)
13:37:16.043780 x.x.x.86.1572 > a.b.c.200.11: S 674711609:674711609(0) win 65535 (DF)
13:37:16.046206 x.x.x.219.1737 > a.b.c.200.25: S 674711609:674711609(0) win 65535 (DF)
13:37:16.050732 x.x.x.250.1148 > a.b.c.200.53: S 674711609:674711609(0) win 65535 (DF)
13:37:16.051229 x.x.x.133.1506 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:16.056246 x.x.x.67.1212 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)
13:37:16.058192 x.x.x.207.1448 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)
13:37:16.058277 x.x.x.57.1617 > a.b.c.200.81: S 674711609:674711609(0) win 65535 (DF)
13:37:16.061113 x.x.x.11.1506 > a.b.c.200.109: S 674711609:674711609(0) win 65535 (DF)
13:37:16.063543 x.x.x.79.1416 > a.b.c.200.123: S 674711609:674711609(0) win 65535 (DF)
13:37:16.068121 x.x.x.245.1530 > a.b.c.200.137: S 674711609:674711609(0) win 65535 (DF)
13:37:16.071039 x.x.x.14.1978 > a.b.c.200.11: S 674711609:674711609(0) win 65535 (DF)
13:37:16.073062 x.x.x.183.1837 > a.b.c.200.25: S 674711609:674711609(0) win 65535 (DF)
13:37:16.073132 x.x.x.1.1765 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:16.077960 x.x.x.233.1793 > a.b.c.200.53: S 674711609:674711609(0) win 65535 (DF)
13:37:16.080902 x.x.x.58.1912 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)
13:37:16.081341 x.x.x.207.1788 > a.b.c.200.81: S 674711609:674711609(0) win 65535 (DF)
13:37:16.082786 x.x.x.107.1440 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)

13:37:16.087775 x.x.x.57.1849 > a.b.c.200.109: S 674711609:674711609(0) win 65535 (DF)
13:37:16.088058 x.x.x.24.1415 > a.b.c.200.123: S 674711609:674711609(0) win 65535 (DF)
13:37:16.092715 x.x.x.252.1554 > a.b.c.200.137: S 674711609:674711609(0) win 65535 (DF)
13:37:16.095416 x.x.x.218.1074 > a.b.c.200.25: S 674711609:674711609(0) win 65535 (DF)
13:37:16.095534 x.x.x.132.1447 > a.b.c.200.11: S 674711609:674711609(0) win 65535 (DF)
13:37:16.097425 x.x.x.84.1462 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:16.099915 x.x.x.32.1682 > a.b.c.200.53: S 674711609:674711609(0) win 65535 (DF)
13:37:16.102336 x.x.x.186.1801 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)
13:37:16.107768 x.x.x.104.1920 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)
13:37:16.108164 x.x.x.158.1494 > a.b.c.200.81: S 674711609:674711609(0) win 65535 (DF)
13:37:16.109802 x.x.x.208.1649 > a.b.c.200.109: S 674711609:674711609(0) win 65535 (DF)
13:37:16.112192 x.x.x.74.1991 > a.b.c.200.123: S 674711609:674711609(0) win 65535 (DF)
13:37:16.116959 x.x.x.246.1580 > a.b.c.200.11: S 674711609:674711609(0) win 65535 (DF)
13:37:16.118035 x.x.x.103.1255 > a.b.c.200.137: S 674711609:674711609(0) win 65535 (DF)
13:37:16.122245 x.x.x.123.1340 > a.b.c.200.25: S 674711609:674711609(0) win 65535 (DF)
13:37:16.122450 x.x.x.179.1939 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:16.124841 x.x.x.242.1130 > a.b.c.200.53: S 674711609:674711609(0) win 65535 (DF)
13:37:16.127240 x.x.x.18.1736 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)
13:37:16.131698 x.x.x.253.1036 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)
13:37:16.132320 x.x.x.248.1028 > a.b.c.200.81: S 674711609:674711609(0) win 65535 (DF)
13:37:16.136995 x.x.x.138.1792 > a.b.c.200.123: S 674711609:674711609(0) win 65535 (DF)
13:37:16.137153 x.x.x.142.1897 > a.b.c.200.109: S 674711609:674711609(0) win 65535 (DF)
13:37:16.141817 x.x.x.193.1432 > a.b.c.200.137: S 674711609:674711609(0) win 65535 (DF)
13:37:16.144762 x.x.x.132.1209 > a.b.c.200.11: S 674711609:674711609(0) \win 65535 (DF)
13:37:16.147000 x.x.x.173.1337 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:16.147166 x.x.x.69.1890 > a.b.c.200.25: S 674711609:674711609(0) win 65535 (DF)
13:37:16.329674 x.x.x.33.1772 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)
13:37:16.334405 x.x.x.64.2008 > a.b.c.200.109: S 674711609:674711609(0) win 65535 (DF)
13:37:16.334486 x.x.x.254.1233 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)
13:37:16.336749 x.x.x.158.1635 > a.b.c.200.123: S 674711609:674711609(0) win 65535 (DF)
13:37:16.341698 x.x.x.152.1586 > a.b.c.200.11: S 674711609:674711609(0) win 65535 (DF)
13:37:16.342205 x.x.x.224.1801 > a.b.c.200.137: S 674711609:674711609(0) win 65535 (DF)
13:37:16.347178 x.x.x.189.1811 > a.b.c.200.25: S 674711609:674711609(0) win 65535 (DF)
13:37:16.349256 x.x.x.245.1983 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:16.352203 x.x.x.93.1136 > a.b.c.200.53: S 674711609:674711609(0) win 65535 (DF)
13:37:16.354099 x.x.x.1.1695 > a.b.c.200.81: S 674711609:674711609(0) win 65535 (DF)
13:37:16.354579 x.x.x.25.1562 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)
13:37:16.356897 x.x.x.237.1991 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)
13:37:16.358984 x.x.x.61.1260 > a.b.c.200.109: S 674711609:674711609(0) win 65535 (DF)
13:37:16.361489 x.x.x.116.1911 > a.b.c.200.123: S 674711609:674711609(0) win 65535 (DF)
13:37:16.364349 x.x.x.181.1549 > a.b.c.200.137: S 674711609:674711609(0) win 65535 (DF)
13:37:16.366321 x.x.x.250.1495 > a.b.c.200.11: S 674711609:674711609(0) win 65535 (DF)
13:37:16.371534 x.x.x.33.1135 > a.b.c.200.25: S 674711609:674711609(0) win 65535 (DF)
13:37:16.371755 x.x.x.102.1468 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:16.374225 x.x.x.87.1540 > a.b.c.200.53: S 674711609:674711609(0) win 65535 (DF)
13:37:16.379136 x.x.x.211.1468 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)

13:37:16.381142 x.x.x.23.1320 > a.b.c.200.81: S 674711609:674711609(0) win 65535 (DF)
13:37:16.383509 x.x.x.252.1869 > a.b.c.200.109: S 674711609:674711609(0) win 65535 (DF)
13:37:16.383851 x.x.x.233.1982 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)
13:37:16.386473 x.x.x.179.1760 > a.b.c.200.123: S 674711609:674711609(0) win 65535 (DF)
13:37:16.391446 x.x.x.71.1377 > a.b.c.200.137: S 674711609:674711609(0) win 65535 (DF)
13:37:16.393434 x.x.x.222.1924 > a.b.c.200.25: S 674711609:674711609(0) win 65535 (DF)
13:37:16.394001 x.x.x.86.1513 > a.b.c.200.11: S 674711609:674711609(0) win 65535 (DF)
13:37:16.396250 x.x.x.110.1705 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:16.401217 x.x.x.15.1015 > a.b.c.200.53: S 674711609:674711609(0) win 65535 (DF)
13:37:16.403754 x.x.x.229.1334 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)
13:37:16.405689 x.x.x.168.1452 > a.b.c.200.81: S 674711609:674711609(0) win 65535 (DF)
13:37:16.408538 x.x.x.181.1838 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)
13:37:16.410956 x.x.x.188.1339 > a.b.c.200.123: S 674711609:674711609(0) win 65535 (DF)
13:37:16.411003 x.x.x.172.1015 > a.b.c.200.109: S 674711609:674711609(0) win 65535 (DF)
13:37:16.415781 x.x.x.30.1107 > a.b.c.200.11: S 674711609:674711609(0) win 65535 (DF)
13:37:16.415935 x.x.x.140.1657 > a.b.c.200.137: S 674711609:674711609(0) win 65535 (DF)
13:37:16.418144 x.x.x.44.1340 > a.b.c.200.25: S 674711609:674711609(0) win 65535 (DF)
13:37:16.422851 x.x.x.222.1123 > a.b.c.200.53: S 674711609:674711609(0) win 65535 (DF)
13:37:16.422907 x.x.x.161.1431 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:16.425592 x.x.x.3.1506 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)
13:37:16.427978 x.x.x.35.1243 > a.b.c.200.81: S 674711609:674711609(0) win 65535 (DF)

Source of Trace
The source of this trace derives from one of the /24 subnets that I manage and monitor.

Detect was generated by
This detect was captured by tcp dump logging.

Probability the source address was spoofed
The source address in this attack was most definitely spoofed based on a number of facts. The first giveaway is the same TCP Initial Sequence Number (ISN) is used for every connection attempt sourcing from various source ip addresses.

13:37:15.469464 x.x.x.210.1353 > a.b.c.200.95: S **674711609:674711609**(0) win 65535 (DF)
13:37:15.471395 x.x.x.159.1737 > a.b.c.200.109: S **674711609:674711609**(0) win 65535 (DF)

Also take notice to the repetitive use of the SYN flag set, which indicates that the source ip address is not interested in a response that could be classified as a denial of service (dos) attack. DOS attacks are predominately spoofed source addresses, which also reinforces the probability that the source ip address is spoofed.

Description of the attack
This is an attempt by a spoofed source ip address to overwhelm the target host through sending TCP SYN packets. TCP SYN packets are the initiating portions of the TCP 3-way handshake:
Sender → SYN
Receiver → SYN/ACK

Sender → ACK

The tcp dump trace shows no sign of a TCP SYN/ACK, which indicates that the attacker is not interested in receiving a response. Every TCP SYN request requires the target host to process the packet by responding with an SYN/ACK. However, the attack is designed so that the target host becomes overwhelmed causing a denial of service condition by forcing it to process a high volume of SYN requests. A CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks have been issued in 1996.

The CVE entry that depicts this attack is CVE-1999_0116 described as "Denial of service when an attacker sends many SYN packets to create multiple connections without ever sending an ACK to complete the connection, aka SYN flood". http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0116.

Attack mechanism

SYN Flooding is the easiest and most common denial of service attack used against internet-based hosts. The attack leverages a very common "flaw" in the way hosts' handles incomplete connections to cause the server to overwhelm internal resources by processing packets, which results in causing it to either crash or become unresponsive to legitimate connections. The following is an analysis of the tcp dump packet detail that caused the denial of service condition.

13:37:15.469464 x.x.x.210.1353 > a.b.c.200.95: **S** 674711609:674711609(0) win 65535 (DF)

The **S** flag or SYN in the tcp dump trace indicates that the attacker is interested in making a connection.

13:37:15.471395 x.x.x.159.1737 > a.b.c.200.109: S 674711609:674711609**(0)** win 65535 (DF)

The **(0)** indicates that no data is being sent in the packet. However the first 40 bytes of the TCP/IP packet are present, which is the TCP/IP header information only.

13:37:15.474313 **x.x.x.19**.1712 > a.b.c.200.123: S **674711609:674711609**(0) win 65535 (DF)
13:37:15.478826 **x.x.x.235**.1379 > a.b.c.200.11: S **674711609:674711609**(0) win 65535 (DF)
13:37:15.478954 **x.x.x.52**.1970 > a.b.c.200.137: S **674711609:674711609**(0) win 65535 (DF)
13:37:15.481713 **x.x.x.13**.1374 > a.b.c.200.25: S **674711609:674711609**(0) win 65535 (DF)
13:37:15.483662 **x.x.x.240**.1723 > a.b.c.200.39: S **674711609:674711609**(0) win 65535 (DF)
13:37:15.486561 **x.x.x.70**.1583 > a.b.c.200.53: S **674711609:674711609**(0) win 65535 (DF)

Notice the source ip address and the associated ISN numbers. This is a clear indication of a spoofed source ip address. Each new connection from a different ip address would create a new and unique ISN number. However this is not the case throughout this trace. It is a clear sign of a crafted packet through high predictability against the ISN numbering convention.

Establishing a TCP connection requires the exchange of three packets:

Sender → SYN
Receiver → SYN/ACK
Sender → ACK

The first with a SYN (for Synchronize) bit from the client, then SYN/ACK in return from the server, and finally ACK (for ACKnowledge) back from the client. The connection is then established; but if there is a delay in completing the handshake, the server re-tries (sending SYN/ACK) several times, and waits with the necessary resources to accept the next packet. Re-try and timeout periods can add up to over three minutes per bogus connection. I will now analyze the timing sequence of the crafted packets:

13:37:16.394001 x.x.x.86.1513 > a.b.c.200.11: S 674711609:674711609(0) win 65535 (DF)
13:37:16.396250 x.x.x.110.1705 > a.b.c.200.39: S 674711609:674711609(0) win 65535 (DF)
13:37:16.401217 x.x.x.15.1015 > a.b.c.200.53: S 674711609:674711609(0) win 65535 (DF)
13:37:16.403754 x.x.x.229.1334 > a.b.c.200.67: S 674711609:674711609(0) win 65535 (DF)
13:37:16.405689 x.x.x.168.1452 > a.b.c.200.81: S 674711609:674711609(0) win 65535 (DF)
13:37:16.408538 x.x.x.181.1838 > a.b.c.200.95: S 674711609:674711609(0) win 65535 (DF)

The timing of these packets demonstrates the attackers ability to send a high volume of packets at the same time, which indicates that high bandwidth, is a contributing factor in making this attack effective. SYN floods are one of the most efficient packet attacks, consuming the greatest amount service with the least effort. It falsifies the initial handshake of the TCP connection with spoofed source IP addresses that the target machine cannot receive a response.

Furthermore the targeted ports are only 53, 67, 95, 81, 109,123, 11, 137, 25, and 39, which reinforces the fact that these are not randomized source IP addresses. In fact these have been crafted with this set of destination ports programmed into this attack.

Correlations

I've seen similar traffic characteristics on previous traces. CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks reporting this activity back in 1996 "half-open connections data structure on the victim server system will eventually fill; then the system will be unable to accept any new incoming connections until the table is emptied out". Another CERT Advisory CA-2000-21 Denial of Service Vulnerabilities in TCP/IP Stacks mentions, "any system that allows critical resources to be consumed without bound is subject to denial of service attacks".

According to RFC 793, "a three-way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's." When new connections are created, an ISN generator is employed in the host that selects a new 32-bit ISN. The generator is bound to a 32 bit clock whose low order bit is incremented roughly every 4 microseconds. "Thus, the ISN cycles approximately every 4.55 hours. Since we assume that segments will stay in the network no more than the Maximum Segment Lifetime (MSL) and that the MSL is less than 4.55 hours we can reasonably assume that ISN's will be

unique. The network trace shows the same ISN number applied by all initiating hosts within a class C subnet which clearly defies the RFC specifications for ISN's.
http://rfc.sunsite.dk/rfc/rfc793.html

Evidence of active targeting

Based upon my observations the attacker was using crafted packets that are designed to cause a syn flood denial of service condition against my target host. There are no other logs supporting any attempt to target other hosts on my subnet. Therefore, this is clearly an active targeted host that was eventually compromised as a result of SYN Flooding. The persistence and relentlessness of the source hosts TCP SYN packets further support this contention.

Severity

Item	Rating	Comment
Criticality	4	Network Flow and Bandwidth Reporting Production Server.
Lethality	2	The attack is a SYN flood that causes a denial of service condition against the target host.
System Countermeasures	3	The target host has the latest operating system patches and all traffic is allowed to the target.
Network Countermeasures	1	The target host is not protected by a firewall.
Severity	**2**	Severity = (Criticality + Lethality) – (System + Net Countermeasures)

Defensive recommendation

Stateful firewalls with syn flood protection so that all inbound tcp syn packets are intercepted by the firewall on behalf of the server. The syn flood protection mechanism will drop the packets if no response was received from the attacking host and could identify crafted packets, which will also cause the firewall to drop packets. An IDS sensor with the ability to detect and write dynamic ACL's to the router so to shun the source attacker is another technique. Rate limiting can also be applied to the edge routers. Therefore protecting the target host from this type of overwhelming traffic.

Multiple Choice Test Question

Consider the ISN:
13:37:15.474313 **x.x.x.19**.1712 > a.b.c.200.123: S **674711609:674711609**(0) win 65535 (DF)
13:37:15.478826 **x.x.x.235**.1379 > a.b.c.200.11: S **674711609:674711609**(0) win 65535 (DF)
13:37:15.478954 **x.x.x.52**.1970 > a.b.c.200.137: S **674711609:674711609**(0) win 65535 (DF)
13:37:15.481713 **x.x.x.13**.1374 > a.b.c.200.25: S **674711609:674711609**(0) win 65535 (DF)

When new connections are created, an ISN generator is employed in the host that selects a new 32-bit ISN. The generator is bound to a 32 bit clock whose low order bit is incremented roughly every:

a) 40 seconds
b) 1 minute
c) 4 microseconds
d) 40 microseconds

The answer is c 4 microseconds. As per RFC 793 hosts increment new ISN's roughly every 4 microseconds, which is bound to a 32-bit clock.

Detect #2 "Nimda Worm" scanning Attack

Dec 31 10:53:29 216.a.b.c %PIX-5-304001: 216.112.x.x Accessed URL 216.w.w.w :/scripts/..%255c../winnt/system32/cmd.exe?/c+dir

Dec 31 10:53:29 216.a.b.c %PIX-6-302002: Teardown TCP connection 1235 faddr 216.112.x.x/1162 gaddr 216.w.w.w/80 laddr 172.16.2.3/80 duration 0:00:01 bytes 321 (TCP Reset-O)

Dec 31 10:53:30 216.a.b.c %PIX-6-302001: Built inbound TCP connection 1236 for faddr 216.112.x.x/1178 gaddr 216.w.w.w/80 laddr 172.16.2.3/80

Dec 31 10:53:30 216.a.b.c %PIX-5-304001: 216.112.x.x Accessed URL 216.w.w.w:/_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir

Dec 31 10:53:31 216.a.b.c %PIX-6-302002: Teardown TCP connection 1236 faddr 216.112.x.x/1178 gaddr 216.w.w.w/80 laddr 172.16.2.3/80 duration 0:00:01 bytes 2877 (TCP Reset-O)

Dec 31 10:53:31 216.a.b.c %PIX-6-302001: Built inbound TCP connection 1237 for faddr 216.112.x.x/1200 gaddr 216.w.w.w/80 laddr 172.16.2.3/80

Dec 31 10:53:32 216.a.b.c %PIX-6-106015: Deny TCP (no connection) from 216.112.x.x/1178 to 216.w.w.w/80 flags RST on interface outside

Dec 31 10:53:33 216.a.b.c %PIX-5-304001: 216.112.x.x Accessed URL 216.w.w.w:/_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir

Dec 31 10:53:33 216.a.b.c %PIX-6-302002: Teardown TCP connection 1237 faddr 216.112.x.x/1200 gaddr 216.w.w.w/80 laddr 172.16.2.3/80 duration 0:00:01 bytes 2877 (TCP Reset-O)

Dec 31 10:53:34 216.a.b.c %PIX-6-302001: Built inbound TCP connection 1238 for faddr 216.112.x.x/1220 gaddr 216.w.w.w/80 laddr 172.16.2.3/80

Dec 31 10:53:34 216.a.b.c %PIX-6-106015: Deny TCP (no connection) from 216.112.x.x/1200 to 216.w.w.w/80 flags RST on interface outside

Dec 31 10:53:35 216.a.b.c %PIX-5-304001: 216.112.x.x Accessed URL 216.w.w.w:/msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/system32/cmd.exe?/c+dir

Dec 31 10:53:36 216.a.b.c %PIX-6-302002: Teardown TCP connection 1238 faddr 216.112.x.x/1220 gaddr 216.w.w.w/80 laddr 172.16.2.3/80 duration 0:00:01 bytes 3584 (TCP Reset-O)

Dec 31 10:53:36 216.a.b.c %PIX-6-302001: Built inbound TCP connection 1239 for faddr 216.112.x.x/1261 gaddr 216.w.w.w/80 laddr 172.16.2.3/80

Dec 31 10:53:37 216.a.b.c %PIX-6-106015: Deny TCP (no connection) from 216.112.x.x/1220 to 216.w.w.w/80 flags RST on interface outside

Dec 31 10:53:37 216.a.b.c %PIX-5-304001: 216.112.x.x Accessed URL 216.w.w.w:/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir

Dec 31 10:53:38 216.a.b.c %PIX-6-302002: Teardown TCP connection 1239 faddr 216.112.x.x/1261 gaddr 216.w.w.w/80 laddr 172.16.2.3/80 duration 0:00:01 bytes 362 (TCP Reset-O)

Dec 31 10:53:38 216.a.b.c %PIX-6-302001: Built inbound TCP connection 1240 for faddr 216.112.x.x/1293 gaddr 216.w.w.w/80 laddr 172.16.2.3/80

Dec 31 10:53:39 216.a.b.c %PIX-5-304001: 216.112.x.x Accessed URL 216.w.w.w:/scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir

Dec 31 10:53:40 216.a.b.c %PIX-6-302002: Teardown TCP connection 1240 faddr 216.112.x.x/1293 gaddr 216.w.w.w/80 laddr 172.16.2.3/80 duration 0:00:01 bytes 2857 (TCP Reset-O)

Dec 31 10:53:40 216.a.b.c %PIX-6-302001: Built inbound TCP connection 1241 for faddr 216.112.x.x/1316 gaddr 216.w.w.w/80 laddr 172.16.2.3/80

Dec 31 10:53:41 216.a.b.c %PIX-6-106015: Deny TCP (no connection) from 216.112.x.x/1293 to 216.w.w.w/80 flags RST on interface outside

Dec 31 10:53:41 216.a.b.c %PIX-5-304001:216.112.x.x Accessed URL 216.w.w.w:/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir

Dec 31 10:53:42 216.a.b.c %PIX-6-302002: Teardown TCP connection 1241 faddr 216.112.x.x/1316 gaddr 216.w.w.w/80 laddr 172.16.2.3/80 duration 0:00:01 bytes 2857 (TCP Reset-O)

Dec 31 10:53:43 216.a.b.c %PIX-6-302001: Built inbound TCP connection 1242 for faddr 216.112.x.x/1341 gaddr 216.w.w.w/80 laddr 172.16.2.3/80

Dec 31 10:53:43 216.a.b.c %PIX-6-106015: Deny TCP (no connection) from 216.112.x.x/1316 to 216.w.w.w/80 flags RST on interface outside

Dec 31 10:53:44 216.a.b.c %PIX-5-304001: 216.112.x.x Accessed URL 216.w.w.w:/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir

Dec 31 10:53:44 216.a.b.c %PIX-6-302002: Teardown TCP connection 1242 faddr 216.112.x.x/1341 gaddr 216.w.w.w/80 laddr 172.16.2.3/80 duration 0:00:01 bytes 2857 (TCP Reset-O)

Dec 31 10:53:45 216.a.b.c %PIX-6-302001: Built inbound TCP connection 1243 for faddr 216.112.x.x/1371 gaddr 216.w.w.w/80 laddr 172.16.2.3/80

Dec 31 10:53:46 216.a.b.c %PIX-6-106015: Deny TCP (no connection) from 216.112.x.x/1341 to 216.w.w.w/80 flags RST on interface outside

Dec 31 10:53:46 216.a.b.c %PIX-5-304001: 216.112.x.x Accessed URL 216.w.w.w:/scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir

Dec 31 10:53:47 216.a.b.c %PIX-6-302002: Teardown TCP connection 1243 faddr 216.112.x.x/1371 gaddr 216.w.w.w/80 laddr 172.16.2.3/80 duration 0:00:01 bytes 323 (TCP Reset-O)

Dec 31 10:53:47 216.a.b.c %PIX-6-302001: Built inbound TCP connection 1244 for faddr 216.112.x.x/1388 gaddr 216.w.w.w/80 laddr 172.16.2.3/80

Dec 31 10:53:48 216.a.b.c %PIX-6-106015: Deny
TCP (no connection) from 216.112.x.x/1371 to 216.w.w.w/80 flags RST on interface outside

Dec 31 10:53:48 216.a.b.c %PIX-5-304001: 216.112.x.x Accessed URL 216.w.w.w:/scripts/..%%35c../winnt/system32/cmd.exe?/c+dir

Dec 31 10:53:49 216.a.b.c %PIX-6-302002: Teardown TCP connection 1244 faddr 216.112.x.x/1388 gaddr 216.w.w.w/80 laddr 172.16.2.3/80 duration 0:00:01 bytes 321 (TCP Reset-O)

Dec 31 10:53:50 216.a.b.c %PIX-6-302001: Built inbound TCP connection 1245 for faddr 216.112.x.x/1412 gaddr 216.w.w.w/80 laddr172.16.2.3/80

Dec 31 10:53:50 216.a.b.c %PIX-6-106015: Deny TCP (no connection) from 216.112.x.x/1388 to 216.w.w.w/80 flags RST on interface outside

Dec 31 10:53:51 216.a.b.c %PIX-5-304001: 216.112.x.x Accessed URL 216.w.w.w:/scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir

Dec 31 10:53:51 216.a.b.c %PIX-6-302002: Teardown TCP connection 1245 faddr 216.112.x.x/1412 gaddr 216.w.w.w/80 laddr 172.16.2.3/80 duration 0:00:01 bytes 325 (TCP Reset-O)

Source of Trace
The source of this trace derives from a network that I manage and monitor.

Detect was generated by

This detect was captured by a 3 Com syslog server that is extracting logs from a Cisco Pix firewall.

Probability the source address was spoofed

The source address in this attack was not spoofed, as the source address appears to be compromised with the Nimda Worm and is interested in receiving a response so that it can gain access to the web server directories. The logs source from a firewall that permits inbound HTTP traffic against port 80 from source any to destination server 172.16.2.3 which is translated to a public address. This is an HTTP request and the firewall completes the TCP 3 way handshake on behalf of the server to ensure that source address responds before establishing a session to the web server. The firewall builds an inbound TCP connection to the web server; it then indicates the URL being accessed; a connection related message appears indicating the TCP connection has been terminated with a byte count by the web server; and finally the firewall discards a TCP packet that had no associated connection within the firewalls connection table because of the TCP RESET flag set.

Description of the attack

This is an attempt by a compromised client to scan and execute commands on an IIS web server. CERT has described this activity as from client to web server scanning for and exploitation of various Microsoft IIS 4.0/5.0 directory traversal vulnerabilities (VU#111677 and CA-2001-12). CERT has also described this activity as from client to web server scanning for the back doors left behind by the "Code Red" (IN02001-09) and "sadmind/IIS (CA-2001-11) worms. This type of scanning dates back to previous CGI Directory Traversal attacks. However, this specific trace resembles the "Nimda Worm" scanning process and has the potential to affect IIS servers running Windows NT and 2000. This attack is best characterized by the Nimda Worm based upon the source IP address characteristics have the same first octet as the destination address and scanning for the IIS Unicode vulnerabilities, which are the URL directory traversal characteristics.

Pix firewall did an excellent job in obtaining the malformed URL attempts against the target host. All logs are forwarded from the pix firewall private interface to a protected syslog server. The logs are then filed for further analysis.

Attack mechanism

The Nimda worm works by compromising client machines through MIME email message consisting of two sections. The first section of MIME is the "text/html" type but it contains no text, so the email appears to have no content. The second section is defined as MIME type "audio/x-wav", but it contains a base64-encoded attachment named "readme.exe". Automatic Execution of Embedded MIME Types is applicable to any mail software running on a x86 platform that uses Microsoft Internet Explorer 5.5 SP1 or earlier which automatically runs the enclosed attachment and infects the machine with the worm. Likewise, the worm infected client machines begin scanning for vulnerable IIS servers.

Nimda looks for backdoors left open by previous IIS worms: Code Red II [IN-2001-09] and sadmind/IIS worm [CA-2001-11]. It also attempts to exploit various IIS Directory Traversal vulnerabilities (VU#111677 and CA-2001-12). The selection of potential target IP addresses follows these rough probabilities:

1. 50% of the time, an address with the same first two octets will be chosen
2. **25% of the time, an address with the same first octet will be chosen**
3. 25% of the time, a random address will be chosen

The trace fits the # 2 criteria in that the first octet of source and destination ip addresses match: Dec 31 10:53:44 **216.a.b.c** %PIX-5-304001: **216.112.x.x** Accessed URL **216.w.w.w**:/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir

The worm infected client machine attempts to transfer a copy of the Nimda code to any IIS server by scanning and discovering vulnerable systems. The infected machine is interested in augmenting a directory traversal so that it can mount the worm to traverse each directory in the system and writes a MIME-encoded copy of itself to disk using file names with .eml or .nws extensions.

216.112.x.x Accessed URL 216.w.w.w:/scripts/..%c1%9c../***winnt/system32/cmd.exe?/c+dir:*** the directory path would put the Nimda worm code in a position to reside within the Windows operating systems root directory which can result in having the code execute numerous violations with root access. In order to further expose the machine, the worm enables the sharing of the c: drive, creates a "Guest" account on the Windows NT and 2000 systems, and adds this account to the "Administrator" group. The Nimda worm infects existing binaries on the system by creating Trojan horse copies of legitimate applications. These Trojan horse versions of the applications will first execute the Nimda code, and then complete their intended function. More information can found at: http://www.cert.org/advisories/CA-2001-26.html

Correlations:

I have seen this type of activity in IIS Web Server logs and also referred to CERT advisories that depict this type of activity. Here a few notices listed within the CVE:
CVE-2000-0731 Directory traversal vulnerability in Worm HTTP server allows remote attackers to read arbitrary files via a ….. (dot dot) attack.

This is the Cert Advisory: http://www.cert.org/advisories/CA-2001-26.html
Securiteam has a series of papers describing the process of "Fingerprinting Port 80 Attacks: A Look into Web Server, and Web Application Attack Signatures"
http://www.securiteam.com/securityreviews/6H00C1535K.html

More detailed technical description found at:
http://www.europe.f-secure.com/v-descs/nimda.shtml

Evidence of active targeting:

The target host is a HTTP host running IIS Web server that was protected by a pix firewall which permitted http port 80 inbound. Abnormal URL's with malicious intent were destined for a Web server. Evidence is indicated by multiple attempts from various sources to attack my single Web server. Furthermore, other log sources throughout the network indicate the similar directory traversal attempts. Therefore it is reasonable to conclude that this was a randomized attempt to find compromised Web servers on my entire network, which is not an active targeted attempt.

Severity

Item	Rating	Comment
Criticality	3	Staging Web Server
Lethality	4	The scan can present problems by mounting the worm within the operating system directory
System Countermeasures	5	The Web server has the latest patches applied
Network Countermeasures	3	The host is protected by a firewall that permits inbound HTTP access.
Severity	**-1**	Severity = (Criticality + Lethality) – (System + Net Countermeasures)

Defensive recommendation:

Make sure that a stateful aware firewall is in place and also apply the latest operating system patches to the IIS Web server. It would be helpful to view the packet data by means of an IDS sensor positioned on the same network segment of the HTTP server and install supporting URL scanning protection measures on the Web server. Additional Nimda Worm filtering can be performed at the edge routers. Microsoft IIS Lockdown Tool:

http://www.microsoft.com/downloads/release.asp?ReleaseID=33961&area=search&ordinal=2

Multiple Choice Test Question

Dec 31 10:53:51 216.a.b.c %PIX-5-304001: 216.112.x.x Accessed URL 216.w.w.w:/scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir

Which of the following best describes this URL attempt?

a) Code Red
b) Nimda Worm
c) sadmind/IIS
d) Denial of Service Attack

The answer is b Nimda Worm. Nimda looks for backdoors left open by previous IIS worms and 25% of the time, an address with the same first octet will be chosen. The worm infected client machine attempts to transfer a copy of the Nimda code to any IIS server by scanning and discovering vulnerable systems.

Detect #3 "SubSeven" Trojan scanning Attack

Mar 01 16:09:55 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 66.66.116.60/4222 dst Intf2: a.b.c.221/27374 by access-group "pub"
Mar 01 16:09:55 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 66.66.116.60/4223 dst intf2: a.b.c.222/27374 by access-group "pub"
Mar 01 16:09:58 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 66.66.116.60/4222 dst intf2 :a.b.c.221/27374 by access-group "pub"
Mar 01 16:09:58 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 66.66.116.60/4223 dst intf2: a.b.c.222/27374 by access-group "pub"
Mar 01 16:10:04 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 66.66.116.60/4222 dst intf2 :a.b.c.221/27374 by access-group "pub"
Mar 01 16:10:04 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 66.66.116.60/4223 dst intf2 :a.b.c.222/27374 by access-group "pub"
Mar 01 17:58:03 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 24.222.119.82/2930 dst intf2: a.b.c.221/27374 by access-group "pub"
Mar 01 17:58:04 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 24.222.119.82/2931 dst intf2: a.b.c.222/27374 by access-group "pub"
Mar 01 17:58:06 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 24.222.119.82/2931 dst intf2 :a.b.c.222/27374 by access-group "pub"
Mar 01 17:58:07 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 24.222.119.82/2930 dst intf2: a.b.c.221/27374 by access-group "pub"
Mar 01 17:58:12 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 24.222.119.82/2931 dst intf2: a.b.c.222/27374 by access-group "pub"

Mar 01 17:58:13 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 24.222.119.82/2930 dst intf2 : a.b.c.221/27374 by access-group "pub"
Mar 01 18:48:16 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 216.124.228.60/2213 dst intf2: a.b.c.221/80 by access-group "pub"
Mar 01 18:48:18 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 216.124.228.60/2213 dst intf2: a.b.c.221/80 by access-group "pub"
Mar 01 18:54:50 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 63.252.14.70/1191 dst intf2: a.b.c.221/27374 by access-group "pub"
Mar 01 18:54:50 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 63.252.14.70/1192 dst intf2: a.b.c.222/27374 by access-group "pub"
Mar 01 18:54:53 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 63.252.14.70/1191 dst intf2: a.b.c.221/27374 by access-group "pub"
Mar 01 18:54:53 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 63.252.14.70/1192 dst intf2: a.b.c.222/27374 by access-group "pub"
Mar 01 18:54:59 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 63.252.14.70/1191 dst intf2: a.b.c.221/27374 by access-group "pub"
Mar 01 18:54:59 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 63.252.14.70/1192 dst intf2: a.b.c.222/27374 by access-group "pub"
Mar 01 19:06:23 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 216.234.127.106/1045 dst intf2:a.b.c.221/27374 by access-group "pub"
Mar 01 19:06:24 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 216.234.127.106/1046 dst intf2:a.b.c.222/27374 by access-group "pub"
Mar 01 19:06:26 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 216.234.127.106/1045 dst intf2: a.b.c.221/27374 by access-group "pub"
Mar 01 19:06:26 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 216.234.127.106/1046 dst intf2: a.b.c.222/27374 by access-group "pub"
Mar 01 19:06:32 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 216.234.127.106/1045 dst intf2: a.b.c.221/27374 by access-group "pub"
Mar 01 19:06:33 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 216.234.127.106/1046 dst intf2: a.b.c.222/27374 by access-group "pub"
Mar 01 19:45:47 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 63.209.234.102/4654 dst intf2: a.b.c.221/27374 by access-group "pub"
Mar 01 19:45:47 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 63.209.234.102/4655 dst intf2: a.b.c.222/27374 by access-group "pub"
Mar 01 19:45:49 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 63.209.234.102/4654 dst intf2 : a.b.c.221/27374 by access-group "pub"
Mar 01 19:45:50 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 63.209.234.102/4655 dst intf2: a.b.c.222/27374 by access-group "pub"
Mar 01 19:45:55 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 63.209.234.102/4654 dst intf2 : a.b.c.221/27374 by access-group "pub"
Mar 01 19:45:56 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 63.209.234.102/4655 dst intf2 : a.b.c.222/27374 by access-group "pub"
Mar 01 19:46:07 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 63.209.234.102/4654 dst intf2: a.b.c.221/27374 by access-group "pub"
Mar 01 19:46:08 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 63.209.234.102/4655 dst intf2: a.b.c.222/27374 by access-group "pub"

Mar 01 20:03:01 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 142.177.214.240/4735 dst intf2: a.b.c.221/27374 by access-group "pub"
Mar 01 20:03:01 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 142.177.214.240/4738 dst intf2: a.b.c.222/27374 by access-group "pub"
Mar 01 20:03:04 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 142.177.214.240/4738 dst intf2: a.b.c.222/27374 by access-group "pub"
Mar 01 20:03:04 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 142.177.214.240/4735 dst intf2: a.b.c.221/27374 by access-group "pub"
Mar 01 20:03:10 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 142.177.214.240/4738 dst intf2: a.b.c.222/27374 by access-group "pub"
Mar 01 20:03:10 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 142.177.214.240/4735 dst intf2 : a.b.c.221/27374 by access-group "pub"
Mar 01 20:58:07 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 66.66.116.60/3789 dst intf2: a.b.c.221/27374 by access-group "pub"
Mar 01 20:58:08 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 66.66.116.60/3790 dst intf2: a.b.c.222/27374 by access-group "pub"
Mar 01 20:58:10 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 66.66.116.60/3789 dst intf2: a.b.c.221/27374 by access-group "pub"
Mar 01 20:58:11 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 66.66.116.60/3790 dst intf2: a.b.c.222/27374 by access-group "pub"

Source of Trace
The source of this trace derives from a network that I manage and monitor.

Detect was generated by

This detect was captured by a 3 Com syslog server that is extracting logs from a Cisco Pix firewall.

Mar 01 20:58:11 1.1.1.1 %PIX-4-106023: Deny tcp src outside: 66.66.116.60/3790 dst intf2: a.b.c.222/27374 by access-group "pub"

Probability the source address was spoofed

The probability that the source addresses in this attack are spoofed is low, as the source addresses are probing for TCP port 27374 and interested in a response.

Description of the attack

These connection attempts to TCP port 27374, one of the default ports used by SubSeven to listen for network traffic, sourced from the Internet from a series of hosts over a 6 hour period. The source IP addresses derive from various sources but are specifically targeting my subnet and TCP port 27374. My firewall denies all ports destined to the protected subnet and logged all attempts to the protected syslog server. The SubSeven Trojan horse scan the Internet looking for

machines that might be compromised on behalf of the hacker. Therefore creating a situation of high anonymity. According to the National Infrastructure Protection Center (NIPC) "Previously released variants of SubSeven have allowed remote attackers to obtain all cached information including, for example, passwords, play audio files, access a web cam, and capture screenshots." http://www.nipc.gov/warnings/advisories/2000/00-056.htm

SubSeven has become the most popular remote access Trojan. It is classified as the easiest to use and most powerful Trojan. The reasons are:

- It is actively maintained and updates.
- The program includes a scanner and can communicate with a slave computer to scan on its behalf.
- There have been reported contests for cracked sites using SubSeven
- It supports port redirection so that any attack can be funneled through a victim's machines.
- Contains extensions that work with ICP, AOL IM, MSN Messenger, and Yahoo messenger, including password sniffing, posting messages, and other features.

Attack mechanism

The first stage of the Trojan Horse attack is to get the program installed on a machine. The next stage is to scan the Internet looking for machines that might be compromised. However, most of the techniques used don't tell the attacker where their victim machine is. Therefore the attacker must scan the Internet looking for the machines they might have compromised. The machines sourcing from the Internet are scanning for one of the default SubSeven TCP ports (27374). Other ports used by SubSeven are: TCP 1080, 1234, 1243, 2773, 2774, 5873, 6667, 6711, & 6776. Additional Trojans use the TCP port 27374, some of which include Bad Blood, EGO, Lion, Ramen, Seeker, The Saint, Ttfloader, and Webhead. One of the SubSeven characteristics is that it supports a scanning utility that is designed to communicate with a slave computer on behalf of the true source. If the attacker discovers that the port is open it will attempt to mount the Trojan, then attempt to get users' passwords, put or get arbitrary files, and so on.

Correlations

I found a number of sites that support the type of attempts that occurred against the pix firewall.

This site is maintained by ISS:
http://advice.networkice.com/advice/exploits/ports/27374/default.htm

This site is maintained through SANS and site sample SubSeven probing logs:

2002-02-22	00:55:33	66.108.130.228	xx.xx.xx.xx	Tcp	1982	27374
2002-02-22	00:55:36	66.108.130.228	xx.xx.xx.xx	Tcp	1982	27374
2002-02-22	00:55:42	66.108.130.228	xx.xx.xx.xx	Tcp	1982	27374

http://www.incidents.org/archives/intrusions/msg02988.html
http://www.incidents.org/archives/intrusions/msg03386.html

Evidence of active targeting:

The SubSeven scan sourced from various hosts but targeted the same default TCP port number 27374. SubSeven scans the Internet looking for machines that might be compromised. Therefore we can conclude that this was the result of randomized scanning and not an active target.

Severity

Item	Rating	Comment
Criticality	5	Web Server
Lethality	4	The scan can present problems by mounting the Trojan and causing a undesirable condition
System Countermeasures	5	The Web server has the latest patches applied
Network Countermeasures	5	The firewall denied all inbound connections to the SubSeven TCP port number 27374
Severity	**-1**	Severity = (Criticality + Lethality) – (System + Net Countermeasures)

Defensive recommendation:

Make sure that a stateful aware firewall is in place and also apply the latest operating system patches to the IIS Web server. It would be helpful to view the packet data by means of an IDS sensor positioned on the outside of the firewall. I would also recommend an anti virus program that constantly updated the signatures.

Multiple Choice Test Question

Which of the following ports is best characterized as used by SubSeven?

a) 200
b) 5000
c) 27374
d) 21

The answer is c 27374. One of the SubSeven characteristics is that it supports a scanning utility that is designed to communicate with a slave computer on behalf of the true source. One of the most used ports is 27374.

Assignment 3 "Analyze This" Scenario

Security Audit Analysis Results Overview for the SANS GIAC University

I have been asked to provide a network security audit for the SANS GIAC University by analyzing logs from their Snort intrusion detection system and produce an analysis report. I have extracted 5 days worth of logs from the SANS web site www.incidents.org/logs starting from 4/1/02 to 4/5/02. There are 3 types of logs provided; scans, alerts, and out of spec packets (OOS). All three-log types were pulled from the same days. The following are the log file sources and dates:

Alerts:

alert.020331.gz	01-Apr-2002 00:05	1.7M
alert.020401.gz	02-Apr-2002 00:06	3.8M
alert.020402.gz	03-Apr-2002 00:07	4.1M
alert.020403.gz	04-Apr-2002 00:06	4.3M
alert.020404.gz	05-Apr-2002 00:07	4.5M

OOS:

oos_Apr.1.2002.gz	01-Apr-2002 06:03	1k
oos_Apr.2.2002.gz	02-Apr-2002 06:05	1k
oos_Apr.3.2002.gz	03-Apr-2002 06:06	1k
oos_Apr.4.2002.gz	04-Apr-2002 06:03	1k
oos_Apr.5.2002.gz	05-Apr-2002 06:01	1k

Scans:

scans.020331.gz	01-Apr-2002 00:11	2.4M
scans.020401.gz	02-Apr-2002 00:12	4.8M
scans.020402.gz	03-Apr-2002 00:12	4.8M
scans.020403.gz	04-Apr-2002 00:12	5.6M
scans.020404.gz	05-Apr-2002 00:12	5.2M

An analysis of each log types and specific event information detected are included within this report. Insight and summaries are offered into internal compromised systems or possible malicious activity traversing the Universities network. Correlation of data is provided through the use of the following organizations: Mitre CVE, Snort, Whitehats, Securiteam, Cert, SANS GIAC, CERT, and SecurityFocus.

What follows is a summary of the network security audit process as a result of the Snort logs analysis provide to us by SANS University. Specific information is included about security vulnerabilities detected as a result of the Snort logs analysis. The analysis also offers insights into potential internal system compromises. Finally, based on the analysis, I have presented a recommendation that details the proper positioning of network security layers to mitigate any future malicious network activity.

Network Security Analysis Process

Note: Analysis was performed without knowledge of the network topology and a complete network security posture audit. The basis and conclusions resulting from the network security analysis are derived from all the Snort logs proved by the SANS GIAC University.

Data Collection

All data was retrieved from the SANS web site www.incidents.org/logs. Three different data sets were used as part of the analysis. These were made up of:

- Snort alerts recorded in "fast" mode. These are the **scan files** that make up the bulk of the data.
- Snort **alerts files** recorded in "full" mode.
- Snort alerts recorded with full decode output. These are the **OOS files**.

Analysis Technique

Research on analysis tools was performed by reviewing students' previous assignments and by visiting the Snort web site. Unfortunately my analysis systems have limited resource to run memory intensive applications, such as Snort Snarf. Therefore, I used a series of Grep utilities to sort through the Snort logs. I removed all port scanning activity in an effort to filter the alert files. Alerts were selected for further analysis based upon volume, severity, and quality of the log data provided.

Alert Logs: The alert analysis describes the Snort alert logs, most active IP addresses that triggered the alert, correlations, insights on potentially compromised machines, potential non malicious activity, and defense recommendations.

Top 10 Talkers: Most active IP addresses are isolated for further analysis within the context of that specific signature. The Scan log analysis was ported into a SQL database that sorted based on source/destination IP address/ports. The resulting scan log data was used to derive at the top "10 Talkers". An additional top 5 external ip address listing was identified as part of the analysis and isolated for gathering the IP address registration information. Furthermore, alert, scans, and out of spec logs, are correlated to further enhance the analysis. Insight was provided as to whether the alert was triggered by legitimate and/or malicious traffic.

OOS: Out of Spec packets were isolated and used for independent analysis. Correlations for OOS packets were performed against Snort alert, scan logs and external sources to further enhance the analysis process.

ALERTS

Snort Alerts Sorted By Frequency

Snort Alerts	Frequency
SNMP public access	**93239**
spp_http_decode: IIS Unicode attack detected	**84649**
SMB Name Wildcard	**66848**
spp_http_decode: CGI Null Byte attack detected	**40040**
MISC Large UDP Packet	**16760**
INFO Inbound GNUTella Connect request	**13378**
INFO Outbound GNUTella Connect request	**3184**
WEB-IIS view source via translate header	**1475**
ICMP Router Selection	**1331**
FTP DoS ftpd globbing	**1073**
WEB-CGI scriptalias access	**196**
FTP CWD / - possible warez site	**54**

SNMP Public Access

Brief description of the attack

Snort identified 93,239 SNMP packets on the University's network that were sourced and destined from a range of internal hosts. The most active destination host is **MY.NET.150.195 and appears to be the most interested in receiving SNMP requests.** The SNMP Public access alert indicates that an intruder is attempting to connect to a host that is running SNMP on TCP or UDP port 161. The mechanism for this attack is to retrieve information via SNMP using well-known default passwords. A high volume of SNMP attempts is demonstrated through this alert trace. This traffic can be summarized as two possibilities; 1- the SNMP connections are legitimate through the means of a system monitoring utility (i.e. Compaq Insight Manager or CA TNG) or 2- the source IP addresses are curious students that are performing brute force attempts using automated SNMP cracking tools. If option 2 is correct then once cracked the attacker can

obtain detailed information about the network and systems. This information can lead to further enhance the capabilities of future attacks and should be taken seriously.

The traffic summary below depicts the most active source and destination IP addresses during this alert activity. I would investigate these hosts in more detail to discern the truth behind this alert.

Most active sources triggering this attack signature

Source IP	Packet Count	Destination IP
MY.NET.88.145	**9579**	**MY.NET.113.202**
MY.NET.88.181	**9559**	**MY.NET.150.195**
MY.NET.88.159	**8463**	**MY.NET.5.248**
MY.NET.70.177	**6976**	**MY.NET.151.114**
MY.NET.70.177	**5732**	**MY.NET.5.97**
MY.NET.88.203	**2159**	**MY.NET.151.86**
MY.NET.70.177	**906**	**MY.NET.150.195**
MY.NET.88.207	**632**	
MY.NET.88.136	**447**	

Defensive recommendation

All SNMP requests should be blocked at the perimeter of the network. There is no reason to share management protocol information about the network and systems to the public. Firewall rules should be applied that deny all TCP and UDP ports 161 inbound connection requests. The second recommendation would be to ensure that the SNMP community strings are made private and complex to crack.

spp_http_decode: IIS Unicode attack detected

Brief description of the attack

Snort created 84,649 IIS Unicode attack alerts. This attack signature is triggered when http port 80 connection attempts occur with a particular URL string that depict a directory traversal attempt. The purpose of this attack is to list directory contents, view files, delete files and execute arbitrary commands against the Microsoft IIS Web server indexing service functionality. Furthermore, the high volume of connection attempts would indicate the use of some type of automation tool. The vulnerability lies in the code that allows the IIS Web server to interact with the indexing service functionality. The problem is due to the indexing service filters not performing bound checking against input buffers

The high volume of packets sourcing from the MY.NET would indicate that these source hosts are running an automated scanning program that searches for vulnerable hosts. It appears that

the source IP addresses are scanning for the exploitation of various Microsoft IIS 4.0 / 5.0 directory traversal vulnerabilities. However, snort.org indicates that this could be the result of false positives related to sites that use cookies with URL encoded binary data, or if you're scanning port 443 and picking up SSL encrypted traffic. These IP addresses should be further investigated for possible compromise.

What follows is a sample of the IIS Unicode crafted URL attempt:

http://address.of.iis5.system/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir+c:

This sample URL was found at: http://www.securityfocus.com/archive/1/140091

According to the Christof Voemel practical:
http://www.giac.org/practical/Christof_Voemel_GCIA.txt

"While the obvious directory transversal http://www.victim.com/../../winnt/system32/cmd.exe doesn't work because IIS strips off "../..", the unpatched IIS doesn't recognize the equivalent Unicode representation "..%C1%1C.." as directory traversal and allows so the attacker to execute commands."

Most active IP addresses triggering this attack signature

Source IP	**Packet Count**
MY.NET.153.203	2923
MY.NET.153.124	2914
MY.NET.153.189	2302
MY.NET.153.112	2138
MY.NET.88.148	2082
MY.NET.88.254	1925
MY.NET.88.171	1494
MY.NET.153.211	1489
MY.NET.88.243	1377
MY.NET.153.167	1253
MY.NET.153.146	767
MY.NET.153.206	502

Defensive recommendation

HTTP access should be blocked by a firewall if there is web server is not required to server content to the public Internet. Microsoft has released several patches that remedy the buffer overflow vulnerabilities inherent within IIS Web Servers. Cert provides a vulnerability note at http://www.kb.cert.org/vuls/id/111677.

I would also recommend a gateway filtering system that would reduce the threats of Internet worms and viruses against the internal private network.
Microsoft IIS Lockdown Tool:
http://www.microsoft.com/downloads/release.asp?ReleaseID=33961&area=search&ordinal=2

Correlations

I have seen similar attack sequences in several different environments. SANS mentions a Unicode Web Interrogator tool that has been used:
http://www.incidents.org/archives/intrusions/msg04082.html.

Cert describes the Nimda worm characteristics:
http://www.cert.org/advisories/CA-2001-26.html

http://www.snort.org/docs/faq.html#4.17

Christof Voemel provide insight into the IIS Unicode characteristics:
http://www.giac.org/practical/Christof_Voemel_GCIA.txt

IIS Unicode Trace: http://www.incidents.org/archives/intrusions/msg04066.html

SMB Name Wildcard

Brief description of the attack

The SMB Name Wildcard alert is triggered based on connection attempts made to UDP port 137, which is the ever so popular Netbios service. The source is making connections to obtain Netbios names that are associated with the IP address. Netbios is a component of the Microsoft file sharing protocol that issues domain, user, and host id information.

This information is useful to the attacker so that she can further enhance an attack by obtaining knowledge of the target host. Netbios scanning against UDP port 137 can be classified as reconnaissance activity. The traffic most often occurs when source and destination ports are both UDP 137. Although this signature alert is only interested in the destination port meeting the criteria.

Given the volume and flow of the traffic it appears that this is normal Microsoft Netbios network activity. Furthermore, the fact that all traffic is confined within MY.NET reduces the probability of this activity being purported by an external attacker.

Most active sources triggering this attack signature

Source IP	Packet Count
MY.NET.11.7	22508
MY.NET.152.18	9073
MY.NET.152.12	1204
MY.NET.152.249	1188
MY.NET.152.176	1188
MY.NET.152.169	1182
MY.NET.152.45	1133
MY.NET.152.20	1124
MY.NET.152.180	1097
MY.NET.152.181	1021
MY.NET.152.173	921

Defensive recommendation

Netbios access (ports 135 – 137) should be blocked by a firewall destined to all internal Microsoft hosts from the public Internet. There is usually no reason for external public Internet traffic to perform Netbios queries against internal private hosts.

Correlations

I've seen similar attacks in the wild within my network environment and also through http://www.incidents.org/archives/y2k/052300-0800.htm.

Jeff Holland has provided a good analysis: http://www.giac.org/practical/Jeff_Holland_GCIA.doc

spp_http_decode: CGI Null Byte attack detected

Brief description of the attack

The attacker is exploiting HTTP port 80 which is the standard port used for websites. Security holes exist within the web applications and servers that enable the attacker to either gain administrative access to the web site, or even the web server. The CGI component is a popular means to compromise a web server because it creates a gateway from HTTP to the backend server executables. Furthermore, the introduction of a null byte can make the attack more lethal by fooling the web application into thinking a different file type has been requested.

Common CGI null byte requests will trick the application into thinking the filename ends in one of its predefined acceptable file types. The attacker is leveraging the web applications inability to check for valid file requests. This routine can lead to a buffer overflow, which can result in privilege escalation of the web server.

It appears that the packets are sourcing from the MY.NET network and connecting to a multitude of hosts on port 80. The volume and speed associated with the source packets would indicate that the host is running an automated program. However, snort.org indicates that this could be the result of false positives related to sites that use cookies with URL encoded binary data, or if you're scanning port 443 and picking up SSL encrypted traffic. This attack signature can be correlated to the **spp_http_decode: IIS Unicode attack detected** Snort alert. These IP addresses should be further investigated for possible compromise.

Most active sources triggering this attack signature

Source IP	Packet Count
MY.NET.153.197	15829
MY.NET.153.149	4386
MY.NET.153.171	4139
MY.NET.153.153	2222
MY.NET.152.11	1169
MY.NET.153.194	946
MY.NET.153.210	627
MY.NET.88.189	126
MY.NET.150.206	64
MY.NET.152.21	45

Defensive recommendation

I would investigate these source hosts as to why they are sending such packets. Antivirus host based scanners should be able to discover the introduction of a worm within the operating system.

Correlations

I have seen similar attack sequences in several different environments. SANS mentions a CGI Null Byte attack scan at http://www.incidents.org/archives/intrusions/msg00967.html.

http://www.snort.org/docs/faq.html#4.17

MISC Large UDP Packet

Brief description of the attack

This alert was triggered due to abnormally large UDP packets traversing the network. Large UDP packets could be a denial of service attempt or a covert channel.

Notice the packet data size:

alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"MISC Large UDP Packet"; **dsize: >4000**; reference:arachnids,247; classtype:bad-unknown; sid:521; rev:1;)

According to http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids247&view=event the criteria for the Large UDP packets denial of service signature is a 4,000 > + bytes IP header and no packet data. A stateful UDP session normally uses small UDP packets, having a payload of no more than 10 bytes. Packets that are reasonably bigger are suspicious of containing control traffic, which can be classified as denial of service or covert channel activity.

Most active sources triggering this attack signature

Source IP	**Packet Count**
63.240.15.205	**2129**
61.78.35.42	**2106**
61.78.35.44	**2027**
163.239.2.31	**1504**
216.106.173.144	**1474**
216.106.173.150	**1295**
63.240.15.207	**1216**
216.106.173.146	**920**
211.115.206.105	**780**
140.142.8.72	**618**
63.250.205.43	**539**

Alert Trace

120131: 04/04-10:11:16.056638 [**] MISC Large UDP Packet [**]
211.115.206.105:4855 -> MY.NET.153.121:3281
120133: 04/04-10:11:16.156393 [**] MISC Large UDP Packet [**]
211.115.206.105:4855 -> MY.NET.153.121:3281
120134: 04/04-10:11:16.249613 [**] MISC Large UDP Packet [**]
211.115.206.105:4855 -> MY.NET.153.121:3281
120135: 04/04-10:11:16.350858 [**] MISC Large UDP Packet [**]
211.115.206.105:4855 -> MY.NET.153.121:3281
120138: 04/04-10:11:16.463254 [**] MISC Large UDP Packet [**]
211.115.206.105:4855 -> MY.NET.153.121:3281
120139: 04/04-10:11:16.558720 [**] MISC Large UDP Packet [**]
211.115.206.105:4855 -> MY.NET.153.121:3281
120140: 04/04-10:11:16.664398 [**] MISC Large UDP Packet [**]
211.115.206.105:4855 -> MY.NET.153.121:3281
120141: 04/04-10:11:16.746224 [**] MISC Large UDP Packet [**]
211.115.206.105:4855 -> MY.NET.153.121:3281

The packets are not sent fast enough to constitute a DOS attack. An average of 8 packets per second are traversing the network, which is not going to cause damage. Therefore the source has

nothing to gain with respect to spoofing. Furthermore, the 216.106.173.0 network is sourcing from iBEAM Broadcasting Corporation, which would indicate that the traffic type from this source is probably multimedia. However, I would not rule out the possibility of malicious traffic based on other source IP addresses within this trace. (Refer to the 5 external addresses and registration information section)

Defensive recommendation

Limit the amount of UDP traffic permitted into your network. A stateful firewall can accomplish this task.

Correlations

I have seen abnormal size UDP packets throughout my network resulting streaming media or gaming traffic.
http://www.whitehats.com/info/IDS247

INFO Inbound GNUTella Connect request (INFO Outbound GNUTella Connect request)

Brief description of the attack

Gnutella is a Napster like application designed to facilitate file sharing among Internet users. It is designed as a distributed file-sharing tool for users to build shared databases. Snort alerted based on inbound scans attempting to discover such Gnutella clients. Denial of service conditions can result when multiple Gnutella users make their pc or network available to the public Internet for managing file transfers. It appears that multiple clients are running Gnutella for file sharing purposes. This activity can lead to malicious file propagation and utilizes bandwidth.

The following IP addresses received the highest volumes of scans.

MY.NET.88.194	**MY.NET.**88.223	**MY.NET.**153.170	**MY.NET.**153.171	**MY.NET.**153.174
MY.NET.153.143	**MY.NET.**150.209	**MY.NET.**153.160	**MY.NET.**152.164	**MY.NET.**153.211
MY.NET.153.194	**MY.NET.**153.175	**MY.NET.**153.170		

The destination port 6346 remained unchanged throughout the alert log.

Snort created a series of outbound GNUTella Connect request alerts that should indicate which IP addresses responded to port 6346.

The following IP addresses sent a response to TCP port 6346.

MY.NET.88.194	**MY.NET.**88.223	**MY.NET.**153.170	**MY.NET.**153.171	**MY.NET.**153.174
MY.NET.153.143	**MY.NET.**150.209	**MY.NET.**153.160	**MY.NET.**153.164	**MY.NET.**153.211
MY.NET.153.194	**MY.NET.**153.175	**MY.NET.**152.185		

Defensive Recommendation

These traffic conditions could lead to future compromise based upon the sheer nature of file system sharing with the public Internet. A piece of malicious code could easily propagate from the GNUTella client and then throughout the private network. My suggestion would be to remove the GNUTella client application from the IP addresses above and deny the inbound port connections with a state full firewall.

Correlations

Cert addresses the importance of introducing unknown code into your computing environment: http://www.cert.org/research/isw/isw2000/papers/18.pdf

Goerge Bakos has a solid analysis: http://www.giac.org/practical/George_Bakos.html

WEB-IIS view source via translate header

Snort created 1475 WEB-IIS view source via translate header alerts. This alert is indicative of an attempt to view web server source scripts, which is a flaw in the IIS web server application. MY.NET.5.96 is the only destination IP address that this alert was created against. Although this signature is specific to IIS, it may have been caused by other exploits. It is possible to force the Microsoft IIS Web server to send back the source of known scriptable files to the attacker if the HTTP header contains a command character with "Translate: f at the end and a slash / is appended to the end of the URL. The Snort signature specifically looks for the Translate: f at the end and a slash / in the packet data. The source IP addresses in this alert are not spoofed because it is a TCP based connection and the attacker is interested in receiving a response.

Most active sources triggering this attack signature

Source IP	**Source IP**
68.55.112.252	**172.154.183**
68.55.180.51	**68.33.17.209**

68.55.201.228	**68.33.11.168**
68.55.198.171	**68.33.26.89**
68.55.178.213	**172.168.152.39**
68.55.0.142	**64.192.55.25**
68.55.56.152	**151.200.59.85**
68.55.240.96	**151.200.46.77**
68.55.228.85	**206.215.11.254**
68.55.116.105	**68.49.34.74**
68.55.200.227	

Defensive recommendation

It is recommended to apply the latest version and patch for Microsoft IIS Web server to remedy the potential for server compromise. I would also investigate the MY.NET.5.96 host to ensure that it is not compromised.

Correlations

http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids305&view=event
http://online.securityfocus.com/bid/1578

ICMP Router Selection

This suspicious activity created 1331 snort alerts. ICMP Router selection alert will be triggered by snort if the ICMP message type is set to 10, which is the router discovery message. Signature definitions indicate that this alert could be a sign of denial of service activities. However, I do not believe this is the case.

The following alert is a sample of the log trace that was issued:

03/31-02:07:29.302582 [**] ICMP Router Selection [**] MY.NET.150.165 -> 224.0.0.2
03/31-02:07:32.533746 [**] ICMP Router Selection [**] MY.NET.150.165 -> **224.0.0.2**
03/31-02:07:35.526758 [**] ICMP Router Selection [**] MY.NET.150.165 -> **224.0.0.2**
03/31-02:45:42.612522 [**] ICMP Router Selection [**] MY.NET.150.165 -> **224.0.0.2**

It appears that the MY.NET.150.165 network is sending the All Routers Multicast ICMP packets. The packets appear to be sent in groups of 2-4 requests. Before a host can send an IP packet beyond its directly connected subnet, it must discover an IP address of at least one operational router on that subnet. This is usually accomplished by sending multicast packets during the startup phase. The source address is attempting to discover a router gateway based upon the destination address of 224.0.0.2, which is an All Routers Multicast attempt. The response to the initiating host will be a discovery of all routers that are connected to its subnet.

The Microsoft Knowledge Base contains an article that gives info on how to disable IRDP. It can be found at:

http://support.microsoft.com/support/kb/articles/q216/1/41.asp

Correlations:

More details can be found at:

http://online.securityfocus.com/bid/578/info/

http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids174&view=event

http://rfc.sunsite.dk/rfc/rfc1256.html

FTP DoS ftpd globbing

Snort created 1073 FTP DoS ftpd globbing alerts. This alert is classified as a denial of service condition caused by an FTP client sending a malicious string to the FTP service. According to Whitehats a legitimate client making a wildcard request can cause this signature, although this is uncommon. Furthermore, the volume of alerts also supports the conjecture that this is not a legitimate FTP request. Many FTP server implementations are vulnerable to permit attackers to gain root privileges by performing a buffer overflow against the FTP daemon.

alert TCP $EXTERNAL any -> $INTERNAL 21 (msg: "IDS487/ftp_dos-ftpd-globbing"; flags: A+; **content: "|2f2a|";** classtype: denialofservice; reference: arachnids,487;)

The snort alert signature is very specific to the content data detail. Contents, 2f2a, and a destination port of 21 (FTP) would cause this alert to trigger. This is abnormal content and cannot be viewed as normal or legitimate traffic.

Most active sources triggering this attack signature

Source IP	**Packet Count**
128.12.57.36	**193**
129.237.88.160	**132**
134.121.154.120	**88**
199.17.198.61	**82**
134.82.142.60	**76**
164.76.179.54	**73**
208.134.66.26	**61**
134.82.143.42	**55**
128.187.250.23	**50**
164.111.21.93	**42**
168.27.250.185	**37**

The following IP addresses were targeted in the attack sequence. I recommend evaluating these destination IP addresses in more detail to discern any malicious files.

Destination IP
MY.NET.150.46
MY.NET.152.174
MY.NET.152.183
MY.NET.152.178
MY.NET.153.194
MY.NET.88.233
MY.NET.153.171
MY.NET.152.180
MY.NET.152.164
MY.NET.153.197
MY.NET.152.185

Defensive recommendation

I recommend denying all inbound FTP by implementing a state full firewall and implement a solution that would create an encrypted connection. A VPN solution would be appropriate to ensure authentication, encryption, non-repudiation, and message integrity while disabling FTP to the public.

Correlations
http://www.securiteam.com/unixfocus/Globbing_Vulnerabilities_in_Multiple_FTP_Daemons.html

http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids487&view=signatures

WEB-CGI scriptalias access

Snort created 196 alerts for WEB-CGI scriptalias access targeted against the MY.NET.5.96 host. The attack mechanism compromises the web server by sending a particular URL string which enables the attacker to view the source CGI scripts that are normally executables. This is accomplished by adding multiple forward slashes in the URL.

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-CGI scriptalias access"; flow:to_server; flow: A+; uricontent: "///"; reference:cve,CVE-1999-0236; reference:bugtraq,2300; reference:arachnids,227; classtype:attempted-recon; sid:873; rev:3;)

These are the most active source IP addresses.

Source IP
68.50.79.192
68.55.176.169

24.162.83.132

Snort also created the WEB-IIS view source via translate header alert against the MY.NET.5.96 host. This is should be taken seriously due to the fact that the MY.NET.5.96 host has had previous HTTP attack attempts. The host could be compromised and should be further investigated.

Defensive recommendation

It is recommended to apply the latest version and patch for Microsoft IIS Web server to remedy the potential for server compromise. I would also investigate the MY.NET.5.96 host to ensure that it is not compromised.

Correlations

http://www.snort.org/snort-db/sid.html?id=873

http://www.whitehats.com/info/IDS227

http://online.securityfocus.com/bid/2300

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0236

FTP CWD / - possible warez site

Snort generated 54 Possible Warez Site alerts. This alert is triggered when a remote user logs into an FTP server with the username of Warez. Warez is a common login credential that is used against a compromised server. The compromised server or "Warez distribution bot" is used to control the distribution of pirated software and movies through bot networks. These underground networks consist of compromised high bandwidth servers at web hosting companies, ISP's, and universities that are designed to file sharing and distribution. Majority of the compromised servers infected with the file sharing software are unknowingly participating in a massive underground operation. There are many automation tools that search for compromised Warez ftp servers.

Source IP Address: 194.38.83.245
Most Active Destination IP Addresses:

Destination IP
MY.NET.150.59
MY.NET.153.220

MY.NET.150.83
MY.NET.150.101
MY.NET.150.147
MY.NET.153.219
MY.NET.150.231
MY.NET.150.84
MY.NET.150.197
MY.NET.150.195

Snort Log:
57811: 03/31-14:12:30.662351 [**] FTP CWD / - possible warez site [**] 194.38.83.245:3625 -> MY.NET.150.59:21
57812: 03/31-14:12:30.856218 [**] FTP CWD / - possible warez site [**] 194.38.83.245:3625 -> MY.NET.150.59:21
57819: 03/31-14:12:31.235637 [**] FTP CWD / - possible warez site [**] 194.38.83.245:3625 -> MY.NET.150.59:21

These supporting scan logs demonstrate the TCP SYN port scanning traffic characteristics associated with the source IP address 194.38.83.245:

Mar 31 06:37:58 194.38.83.245:4151 ->MY.NET.150.101:21 SYN ******S*
Mar 31 06:38:00 194.38.83.245:4153 ->MY.NET.150.103:21 SYN ******S*
Mar 31 06:38:00 194.38.83.245:4156 ->MY.NET.150.106:21 SYN ******S*
Mar 31 06:37:58 194.38.83.245:4157 ->MY.NET.150.107:21 SYN ******S*
Mar 31 06:38:00 194.38.83.245:4163 ->MY.NET.150.113:21 SYN ******S*
Mar 31 06:38:00 194.38.83.245:4164 ->MY.NET.150.114:21 SYN ******S*
Mar 31 06:38:00 194.38.83.245:4172 ->MY.NET.150.122:21 SYN ******S*
Mar 31 06:38:00 194.38.83.245:4175 ->MY.NET.150.125:21 SYN ******S*
Mar 31 06:38:00 194.38.83.245:4183 ->MY.NET.150.133:21 SYN ******S*
Mar 31 06:38:00 194.38.83.245:4186 ->MY.NET.150.136:21 SYN ******S*
Mar 31 06:37:58 194.38.83.245:4189 ->MY.NET.150.139:21 SYN ******S*
Mar 31 06:37:58 194.38.83.245:4192 ->MY.NET.150.142:21 SYN ******S*

Mar 31 06:37:58 194.38.83.245:4193 ->MY.NET.150.143:21 SYN ******S*
Mar 31 06:37:59 194.38.83.245:4197 ->MY.NET.150.147:21 SYN ******S*
Mar 31 06:38:00 194.38.83.245:4215 ->MY.NET.150.165:21 SYN ******S*
Mar 31 06:38:00 194.38.83.245:4220 ->MY.NET.150.170:21 SYN ******S*
Mar 31 06:38:00 194.38.83.245:4222 ->MY.NET.150.172:21 SYN ******S*
Mar 31 06:37:59 194.38.83.245:4245 ->MY.NET.150.195:21 SYN ******S*
Mar 31 06:37:59 194.38.83.245:4247 ->MY.NET.150.197:21 SYN ******S*

194.38.83.245 has a determined interest in connecting to TCP port 21.

Defensive Recommendation

First investigate the servers in the most active destination ip address list. Evaluate the file structure, disk space and ensure that directories and files have not been added. Secondly block FTP inbound via a state full firewall and implement a solution that would create an encrypted connection. A VPN solution would be appropriate to ensure authentication, encryption, non-repudiation, and message integrity while disabling FTP to the public.

Correlations
http://www.snort.org/snort-db/sid.html?id=546
http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids327&view=research
http://www.securiteam.com/securitynews/5ZP021575W.html

SCANS

Sources

Top 10 Source IP Addresses

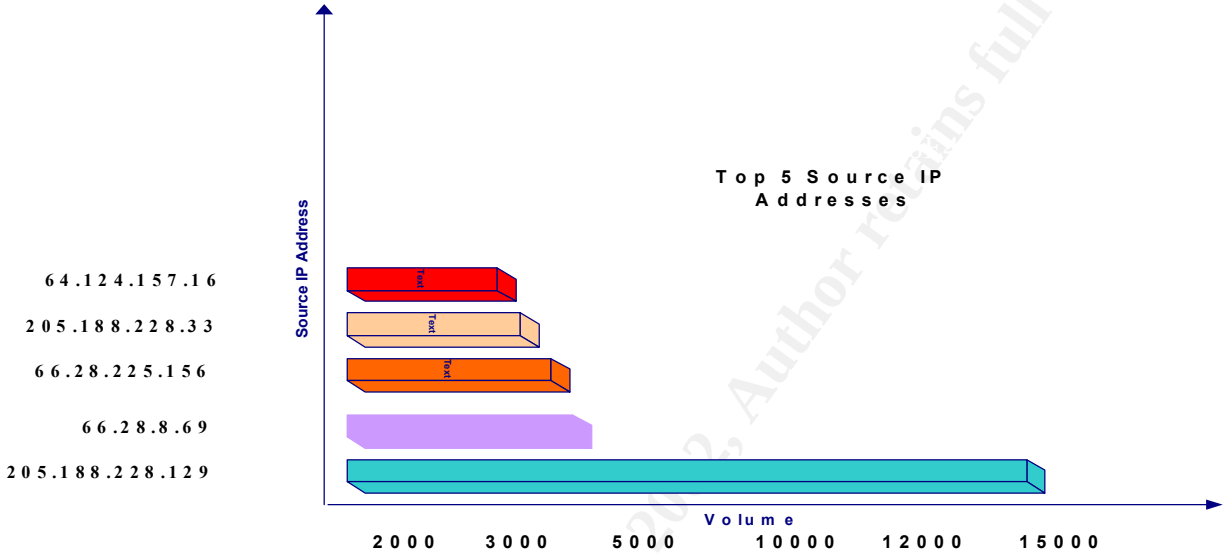
Source IP	Packet Count
MY.NET.60.43	454348
MY.NET.150.143	313477
MY.NET.6.45	180115
MY.NET.6.48	173712
MY.NET.6.49	162448
MY.NET.6.52	147949
MY.NET.6.50	130941
MY.NET.6.53	78079
MY.NET.150.113	69079
MY.NET.6.60	67485

Top 10 Source Ports

Source Port	Packet Count
7000	450807
123	360645
7001	294630
0	185613
28800	126825
1057	118349
137	90246
514	71533
1347	53681
2196	49984

Top 5 Sourced External IP Addresses

Source IP	Packet Count
64.124.157.16	14867
205.188.228.33	3560
66.28.225.156	3314
66.28.8.69	3033
205.188.228.129	3001



Link Graph Showing the top 5 sources IP addresses

Destinations

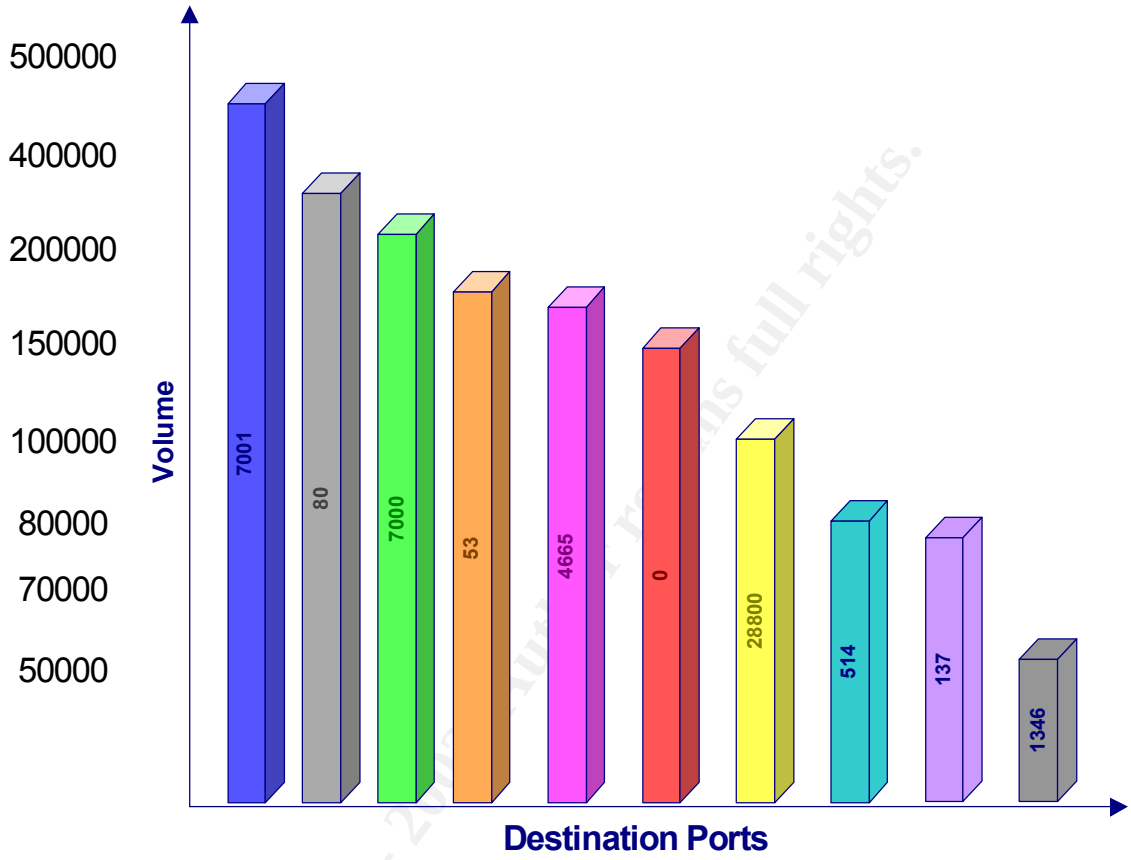
Top 10 Destinations IP Addresses

Destination IP	Packet Count
MY.NET.1.3	112887
MY.NET.1.7	82282
MY.NET.6.45	78535
MY.NET.1.4	75228
MY.NET.60.43	54928
MY.NET.11.6	43583
MY.NET.6.53	35079
MY.NET.11.7	32824
MY.NET.6.60	32428

MY.NET.153.209	29682

Top 10 Destination Ports with services descriptions.

Destination Port	**Packet Count**	**Services Description**
7001	450953	Freak 88 (trojan)
80	382833	HTTP, Ack Cmd (trojan)
7000	260729	Exploit Translation Server (trojan)
53	188769	Domain
4665	183660	EDONKEY 2000 Server
0	151311	PING
28800	102913	?
514	82382	RPC Backdoor (trojan)
137	78997	NETBIOS
1346	53652	Alta Analytics License Manager



Link Graph showing the top 10 destination ports

FIVE EXTERNAL IP ADDRESS REGISTRATION INFORMATION

The following external ip addresses have been selected from the alerts, scans, and OOS logs. The selection criteria are based upon the volume, frequency and severity of the logs.

External IP Addresses Selected from Alert Logs

194.38.83.245

inetnum: 194.38.74.0 - 194.38.90.255
netname: CH-URBA-NET
descr: WAN Network of Urbanet SA
descr: CP215, 1000 Lausanne 22
descr: Switzerland
country: CH
admin-c: CCU-RIPE

tech-c: CCU-RIPE
rev-srv: eiger.urbanet.ch
rev-srv: salantin.urbanet.ch
status: ASSIGNED PA
notify: hostmaster@urbanet.ch
mnt-by: CH-URBA-NET-MNT
changed: jerome.tissieres@cablecom.ch 20011023
source: RIPE

route: 194.38.80.0/21
descr: CH-URBA-NET
origin: AS8493
notify: hostmaster@urbanet.ch
mnt-by: CH-URBA-NET-MNT
changed: jerome.tissieres@cablecom.ch 20011023
source: RIPE

role: Cablecom gmbh NOC
address: Av. de Lausanne 57
address: CH-1110 Morges
phone: +41 21 802 81 11
fax-no: +41 21 802 81 51
e-mail: hostmaster@urbanet.ch
admin-c: YB204-RIPE
tech-c: FJ165-RIPE
tech-c: JT1657-RIPE
tech-c: RIPE17-RIPE
nic-hdl: CCU-RIPE
notify: jerome.tissieres@cablecom.ch
notify: frederic.jutzet@cablecom.ch
mnt-by: CH-URBA-NET-MNT
changed: jerome.tissieres@cablecom.ch 20010709
source: RIPE

163.239.2.31

Sogang University (NET-SOGANG-NET)
Seoul
KR
Netname: SOGANG-NET
Netblock: 163.239.0.0 - 163.239.255.255
Coordinator:
Villarreal, Felix M. (FMV3-ARIN) [No mailbox]
(82)(2) 705-8492
Domain System inverse mapping provided by:
CCS.SOGANG.AC.KR 163.239.1.1

NS.HANA.NM.KR 203.232.127.1
Record last updated on 08-Oct-1992.
Database last updated on 16-May-2002 19:59:02 EDT

External IP Addresses Selected from Scan Logs

64.124.157.16

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 22-May-2002 20:00:05 EDT.

205.188.228.33

America Online, Inc (NETBLK-AOL-DTC)
22080 Pacific Blvd
Sterling, VA 20166
US
Netname: AOL-DTC
Netblock: 205.188.0.0 - 205.188.255.255
Coordinator: America Online, Inc. (AOL-NOC-ARIN) domains@AOL.NET 703-265-4670
Domain System inverse mapping provided by:
DNS-01.NS.AOL.COM 152.163.159.232
DNS-02.NS.AOL.COM 205.188.157.232

Record last updated on 27-Apr-1998.
Database last updated on 22-May-2002 20:00:05 EDT.

External IP Addresses Selected from OOS Packet Logs

192.115.135.8

192.115.128.0 - 192.115.135.255
ACTCOM-BLOCK-5
Actcom - Active Communications Ltd.
IL
AP53
AP53
ASSIGNED PA
MAINT-AS4148
vects@actcom.co.il 20010828
RIPE
192.115.128.0/21
ACTCOM - Active Communications Ltd.
Haifa Tower, 63a Herzl St
Haifa, Israel
AS4148
MAINT-AS4148
vects@actcom.net.il 20020407
RIPE

Amir Plivatsky
ACTCOM - Active Communication Ltd.
P.O.Box 5402
Haifa 31054
Israel
+972 4 8676115
+972 4 8676088

e-mail: amir@actcom.co.il
AP53
hank@vm.tau.ac.il 19950129
registrar@ns.il 19960704
RIPE

142.51.44.123
Laurentian University (NET-LAURENTIANCS)
Dept. of Mathematics and Computer Science, 935 Ramsey Lake Rd.
Sudbury, ON P3E 2C6
CA
Netname: LAURENTIANCS
Netblock: 142.51.0.0 - 142.51.255.255
Maintainer: LARU
Coordinator:
Melanson, Gilles (GM382-ARIN) gilles@cs.laurentian.ca
(705) 675-1151 x2335 (FAX) (705) 673-6591

Domain System inverse mapping provided by:
NS.LAURENTIAN.CA 142.51.1.52
NS2.LAURENTIAN.CA 142.51.1.53

OUT OF SPEC PACKETS

Queso fingerprint

04/02-13:29:53.403159 192.115.135.8:45920 -> MY.NET.5.92:80
TCP TTL:46 TOS:0x0 ID:20276 DF
21S***** Seq: 0xC12F8504 Ack: 0x0 Win: 0x1638
TCP Options => MSS: 1422 NOP NOP SackOK NOP WS: 0

=+
04/02-13:29:55.503336 192.115.135.8:45921 -> MY.NET.5.92:80
TCP TTL:46 TOS:0x0 ID:33039 DF
21S***** Seq: 0xC0B04A0E Ack: 0x0 Win: 0x1638
TCP Options => MSS: 1422 NOP NOP SackOK NOP WS: 0

Snort Alert Trace

178735: 04/02-13:27:56.963760 [**] Queso fingerprint [**] 192.115.135.8:45920 -> MY.NET.5.92:80
178751: 04/02-13:27:59.063595 [**] Queso fingerprint [**] 192.115.135.8:45921 -> MY.NET.5.92:80

OOS Notes:

The reserved bits are set within the TCP flag sets. The first 2 bits of the TCP flags are reserved bits and are presented as "21". Reserved bits should not be set under any conditions. Although, according to RFC 2481, a proposal has been presented that explains how the reserved fields within the TCP header can be used for congestion control through the Explicit Congestion Notification (ECN) protocol. This proposal has not been completely adopted and the type of service within this OOS packet is not set, which leads to suspicion. A TCP SYN flag is also set within this packet. Queso does a good job in hiding the reserved bits by setting it in a TCP SYN

packet. What makes the reserved bits "stealthy" is that if an analyst were to use TCP dump to view this trace, the reserved bits would not have been seen unless the packet hex dump data was disclosed. Reserved bits represented in a TCP packet can also be a motivation to avoid firewalls and intrusion detection systems causing these filtering systems to misinterpret the entire frame. Another suspicious element to this packet is that it has no Type Of Service set within the header. A standard application would have triggered the packet as to the TOS setting.

This description of the OOS notes associated with the Queso fingerprinting presents a compelling argument. However, there is a possibility that this may be a false positive. According to the following RFC: http://www.ietf.org/rfc/rfc2884.txt?number=2884, the IETF has released a Linux kernel module for ECN. It is compatible with Linux kernel version 2.0.32, 2.2.5, and 2.3.43. Therefore it is reasonable to conclude that there is a possibility these OOS packet traces could be sourcing from a Linux host that contains the ECN protocol within the stack.

Correlation

http://www.whitehats.com/info/IDS29
http://www.iss.net/security_center/advice/Intrusions/2000321/default.htm
http://www.incidents.org/detect/ecn.php
http://www.ietf.org/rfc/rfc2481.txt?number=2481
http://www.ietf.org/rfc/rfc2884.txt?number=2884

NMAP Scanning "ALL TCP FLAGS SET!!"
NMAP compare packet output to NMAP functionality.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
04/04-01:52:50.349038 142.51.44.123:9 -> MY.NET.88.162:1900
TCP TTL:115 TOS:0x0 ID:59175  DF
2*SFR**U Seq: 0x4BE012C   Ack: 0xA55BC07F   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
04/04-01:54:27.669069 142.51.44.123:21 -> MY.NET.88.162:1900
TCP TTL:115 TOS:0x0 ID:10557  DF
*1SF*PAU Seq: 0x4BE012C   Ack: 0xA55BC0A8   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK NOP NOP TS: 0 0 EOL EOL EOL
EOL
```

=+
04/04-01:54:39.727093 142.51.44.123:1900 -> MY.NET.88.162:1214
TCP TTL:115 TOS:0x0 ID:10560 DF
*1SFRPAU Seq: 0x12CA55B Ack: 0x33C0AC Win: 0x5010
01 2C A5 5B 00 33 C0 AC 1E BF 50 10 22 38 65 D9 .,.[.3....P."8e.
00 00 00 00 00 00 ......

Supporting Scan Logs

08273: Apr 4 00:40:51 142.51.44.123:1900 -> MY.NET.88.162:1214 FULLXMAS 1*UAPRSF RESERVEDBITS
13524: Apr 4 01:10:25 142.51.44.123:1900 -> MY.NET.88.162:1214 FIN *******F
19779: Apr 4 01:47:57 142.51.44.123:1900 -> MY.NET.88.162:1214 SYNFIN 1*****SF RESERVEDBITS
20160: Apr 4 01:50:53 142.51.44.123:9 -> MY.NET.88.162:1900 NOACK *2U**RSF RESERVEDBITS
20389: Apr 4 01:52:31 142.51.44.123:21 -> MY.NET.88.162:1900 INVALIDACK 1*UAP*SF RESERVEDBITS
20413: Apr 4 01:52:43 142.51.44.123:1900 -> MY.NET.88.162:1214 FULLXMAS 1*UAPRSF RESERVEDBITS
22175: Apr 4 02:02:38 142.51.44.123:1900 -> MY.NET.88.162:1214 INVALIDACK **UA*RSF
23661: Apr 4 02:13:13 142.51.44.123:1900 -> MY.NET.88.162:1214 NOACK **U*P*S*

23712: Apr 4 02:13:37 142.51.44.123:1900 -> MY.NET.88.162:1214 NOACK 12**PRSF RESERVEDBITS
24095: Apr 4 02:16:30 142.51.44.123:2445 -> MY.NET.88.162:1214 XMAS **U*P**F
24424: Apr 4 02:18:30 142.51.44.123:2445 -> MY.NET.88.162:1214 VECNA *2**P**F RESERVEDBITS

OOS Notes

The reserved bits are set within the TCP flag sets. The first 2 bits of the TCP flags are reserved bits and are presented as "21". Reserved bits should not be set under any conditions. Although, according to RFC 2481, a proposal has been presented that explains how the reserved fields within the TCP header can be used for congestion control through the Explicit Congestion Notification (ECN) protocol. This proposal has not been completely adopted and the type of service within this OOS packet is not set, which leads to suspicion. Another suspicious element to this packet is that it has no Type Of Service set within the header. A standard application would have triggered the packet as to the TOS setting. However the most obvious characteristic of a crafted packet is the overwhelming presence of TCP flags set within the packet. NMAP is a scanning program that allows curious individuals and network administrators to determine the

type of services that are enabled on the target hosts. NMAP has a wealth of TCP Flag setting options:

- TCP SYN Scan: S
- ACK Scan: A
- SYN FIN Scan: SF
- XMAS Scan: SFUP
- Operating System Fingerprinting

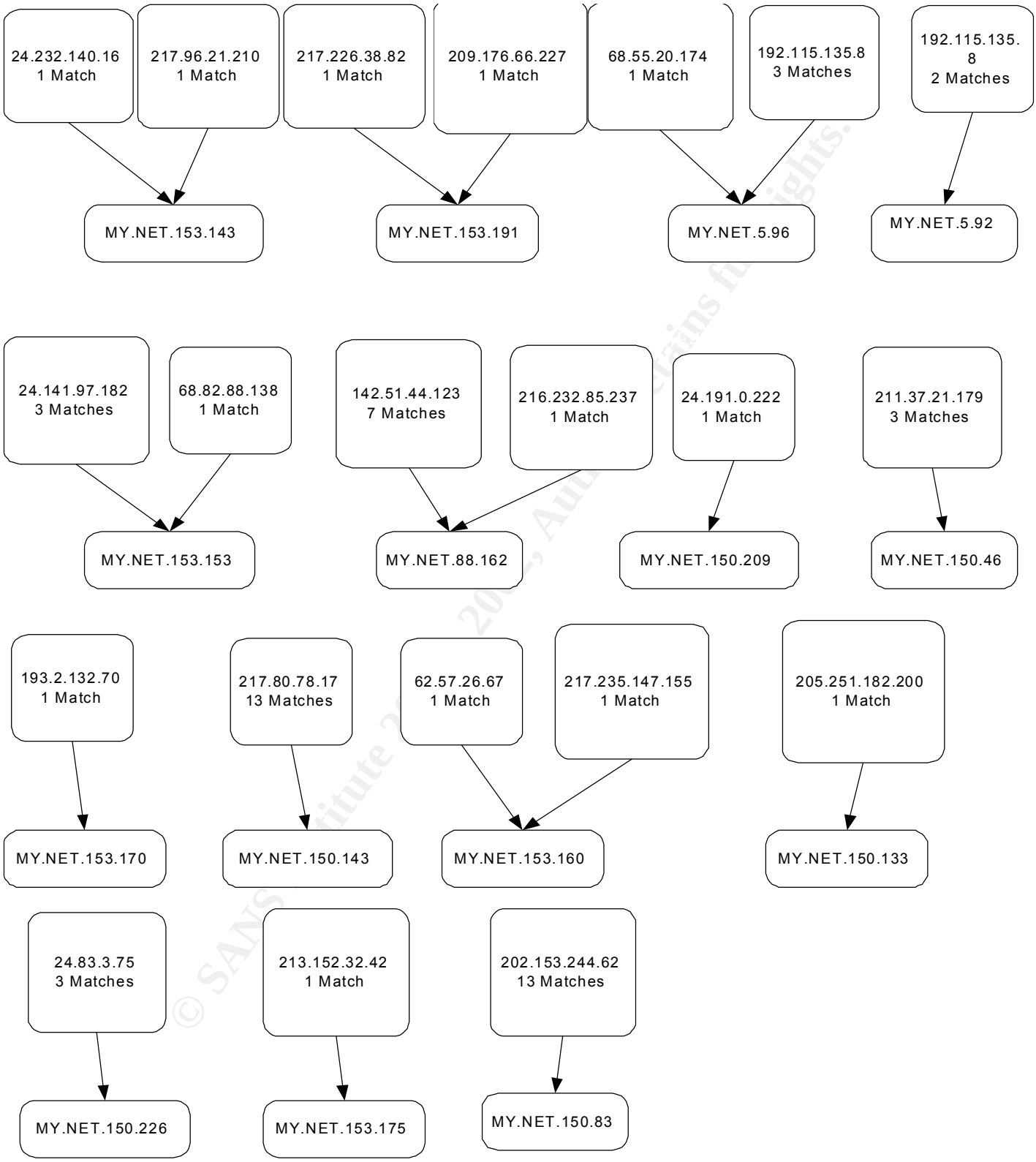
These packets are clearly crafted with the intent to bypass firewalls and packet filters so that the curious source can ascertain the type of services running on the target host.

Correlations:

http://www.ietf.org/rfc/rfc2481.txt?number=2481
http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids5&view=event
http://www.insecure.org/nmap/

OOS Link Graph

The following illustration is a link graph including OOS packet snort logs from the SANS Universities network. The graph will demonstrate the relationships between the frequency and types of source and destination IP addresses within the OOS packet snort logs.



Anomalous Activity and Compromised Machines on the Network

Identifying compromised machines from a network's logs that are unfamiliar does not guarantee 100% analysis accuracy. Given the amount of data used for analysis and having knowledge of network traffic normalcy, I was able to derive at a list of possible compromised machines. Here is a list that summarizes the suspected machines and actions to be taken

- The internal machine **MY.NET.150.195** should be inspected to ensure the proper access control measures are taken against secure SNMP traffic. (Which hosts should be connecting to SNMP and how strong are the community strings?)

- The following internal machines that need to be inspected for potential Nimda or Code Red Worm infections based upon the outbound http requests and URL characteristics. It is possible that the code could propagate to external destinations while deriving from the Universities **MY.NET** network. Anti virus and operating system integrity checks need to be performed against these hosts.

MY.NET.153.203	**MY.NET.153.197**
MY.NET.153.124	**MY.NET.153.149**
MY.NET.153.189	**MY.NET.153.171**
MY.NET.153.112	**MY.NET.153.153**
MY.NET.88.148	**MY.NET.152.11**
MY.NET.88.254	**MY.NET.153.194**
MY.NET.88.171	**MY.NET.153.210**
MY.NET.153.211	**MY.NET.88.189**
MY.NET.88.243	**MY.NET.150.206**
MY.NET.153.167	**MY.NET.152.21**
MY.NET.153.146	

- The **MY.NET.153.121** machine should be inspected for any infected files used for covert channel communications.

- The internal host **MY.NET.5.96** appears to be receiving inbound http requests with URL strings that represent Nimda or Code Red Worm characteristics. I would investigate this machine by running anti virus and operating system integrity checks.

- The **MY.NET.150.165** is perform ICMP All Routers Multicast requests and should be investigated in more detail.

- The following internal machines triggered alarms associated to inbound FTP traffic. FTP is inherently insecure based upon the ability to mount files from an anomalous source. Anti virus and operating system file integrity checks need to be performed.

MY.NET.150.46	**MY.NET.150.59**
MY.NET.152.174	**MY.NET.153.220**
MY.NET.152.183	**MY.NET.150.83**

MY.NET.152.178	**MY.NET.150.101**
MY.NET.153.194	**MY.NET.150.147**
MY.NET.88.233	**MY.NET.153.219**
MY.NET.153.171	**MY.NET.150.231**
MY.NET.152.180	**MY.NET.150.84**
MY.NET.152.164	**MY.NET.150.197**
MY.NET.153.197	**MY.NET.150.195**
MY.NET.152.185	

- The MY.NET.5.92 and MY.NET.88.162 internal machines have been a reconnaissance target through the means of crafted inbound packets and needs to be investigated.

Overall Site Network Security Posture Recommendations

My recommendation at this time is to strengthen the internal network borders through segmentation with the use of firewalls and network intrusion detection systems. This would give the University a greater degree of protection from activity inside the trusted network. Furthermore, an audit trail that integrates logs from routers, firewalls, hosts and network ids are essential to the monitoring and assessment process that provides effectiveness of the security policy's implementation. The additional logging can also be valuable in tracking your internal and external activities with the ability to perform correlation analysis against all interconnected networks.

The ingress and egress points of your network require the tuning of intrusion detection system signatures in order to reduce the frequency of false positives. This will give the analyst greater opportunity to deal with real events, rather than tracking down false events. Additionally, a DMZ network should be created on the firewall that will provide an isolated publicly accessible network for Web, FTP, Media and Mail services, while eliminating the ability for public Internet traffic to traverse the private network. This is a critical point of the network since the University is currently permitting these services to traverse the private network.