Global Information Assurance Certification Paper

GIAC GCIA Certification
Assignment 3.4 (With Q1 Option 1)
Blaine Hein
May 21, 2004

Integration of IDS Event Analysis into a Risk Management Process

Introduction

An Intrusion Detection System (IDS) is not a silver bullet that will solve all of your network security problems with the flip of a switch and the pressing of a few keys. The vast majority of work to run an IDS begins at the point where the IDS system writes a log entry and moves on. Event analysis of these log entries is a critical piece of the puzzle. This paper will expand upon the analysis process provided in the IDS certification track at SANS with a framework for the integration of IDS event analysis into a risk management process.

This paper uses the Canadian Risk Management Framework [CSE1] published by Communications Security Establishment in 1996 as a basis for the discussion. However, any validated risk management approach is suitable for this integration approach. For example, the US (NIST) and Australia provide different models for Risk Management which may be more appropriate for some readers.

Motivation

Integration of your IDS system into your RM process allows for the adoption of a single set of terminology to describe the security and residual risks of your network environment. When dealing with upper management, this is important when trying to justify changes in the network architecture due to newly discovered vulnerabilities. When money is required, the argument of accepting a new and quantified risk vs. spending money to mitigate the risk becomes an easier discussion for all parties to understand. The addition of data from the IDS assists in the process of quantifying the risk for management.

Definitions

Risk Management

Risk management means many things to many people. There have been several discussions on risk management within the SANS community to date. The Canadian Government formal definition of risk management is:

> … defining what is at risk, the relative magnitude of risk, the causal factors, and what to do about the risk. Options for managing risk include reduction, transfer, avoidance and acceptance. Risk can be reduced by implementing a managed system architecture, which includes operational, procedural, physical, personnel, and technical security components.[i]

The risk management model is depicted in Figure 1 below. In its simplest form, the goal of risk management is to reduce the exposure of a system to an

acceptable level. The major components of risk management and their applicability to IDS event analysis will be described within this paper.

Threat and Risk Assessment (TRA)

A significant portion of the technical analysis in risk management has been categorized as a Threat and Risk Assessment (TRA). This portion is identified with a square box at the center of the risk management model shown in Figure 1.

> Risk is a function of the consequences (or impact) of an undesirable event and the likelihood of that event occurring. Risk assessment is the process whereby risk relationships are analyzed, and an estimate of the risk of asset compromise is developed. Compromise includes unauthorized disclosure, destruction, removal, modification, or interruption. Options for managing risk include reduction, transfer, avoidance, and acceptance.[ii]

Figure 1 - Risk Management Model [CSE1]

The TRA process defines risk as a combination of Impact and Likeliness. Impact and likeliness are each influenced by the asset, threat and vulnerability. The common scales used for impact and likeliness are High, Medium, and Low. Figure 2 depicts the relationship between these components and the calculation of the resulting impact and likeliness. In this model, risk is calculated as the cross product of the Impact and the Likeliness of the event.

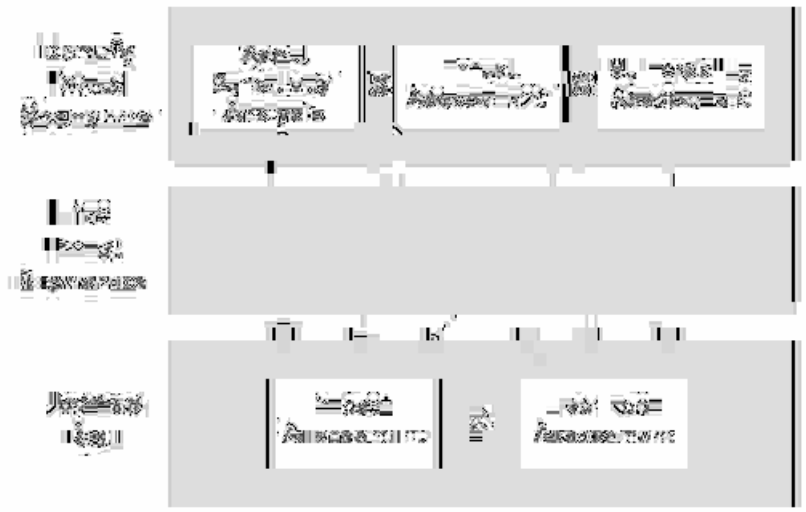


Figure 2 - Threat and Risk Assessment (TRA) Process [CSE1]

Intrusion Detection System (IDS) Event Analysis

Event analysis of the IDS data is all about determining what happened (if anything,) and how bad it was. Even with the detailed information provided by modern IDS software products, weeding out the false positives is not a trivial task. As there will likely be multiple events of interest every time the logs are reviewed, event analysis is also about prioritizing resources against the most severe events first. The SANS Intrusion Detection Immersion Curriculum [SANS1] provides a definition of Severity.

Severity = (criticality + Lethality) – (system + net countermeasures)

The four components, which make up severity, are each ranked on a five-point scale where 1 is the least significant and 5 is the most significant.
Criticality relates to the importance of the asset to the organization. A user workstation is less critical than the corporate mail server. Lethality of the attack relates to the impact of the attack. A denial of service (DOS) attack against a user workstation is less lethal than a DOS attack against the entire network. System and network countermeasures relate to the degree of protection that is in place to prevent these attacks from succeeding. Examples include the use of modern patched operating systems and the use of a validated restrictive firewall.

Strategic vs. Tactical

Strategic: of great importance within an integrated whole or to a planned effect. [Merriam-Webster's Dictionary] In general this would be a consideration of the long term or big picture.

Tactical: of or relating to tactics : as (1) **:** of or relating to small-scale actions serving a larger purpose (2) **:** made or carried out with only a limited or immediate end in view. [Merriam-Webster's Dictionary]

One of the difficulties in writing this paper was determining how to explain the differences in paradigm between IDS event analysis and Risk Management.

Risk management methodologies traditionally concentrate on long-term changes to the network environment. They are often used as a planning tool during the design of a network to determine the level of risk associated with the network architecture. In this manner risk management would be considered to be a strategic tool.

IDS systems concentrate on the momentary state of the network. They have the potential to provide near real time information on attacks against the network. They do not perform the analysis of the attack, nor do they guide the operator towards the best way to fix the problem. In this manner an IDS would be considered to be a tactical tool.

Analysis Methodology

An IDS system can be modeled as a miniature risk management methodology. To illustrate this point, the following paragraphs will describe the mapping of an IDS along with a few sound administrative practices onto the risk management model depicted in Figure 1. The term miniature is utilized since an IDS is only one of many security components required to achieve comprehensive network security.
The risk management model shown in Figure 1 is commonly used as a strategic tool. This means that it is commonly used in environments that change relatively slowly. This model is also commonly used as a planning tool to build new networks or to make significant changes to existing networks.

While the authors of this document did not specifically plan for the dynamic or tactical environment in which an IDS operates, nothing present in this model prevents the adaptation to the tactical environment.

While the diagram normally read in a clockwise manner, a few deviations in the order of presentation will make the mapping of an IDS system clearer.

Planning

The planning process includes several components listed below. Without expanding on each of them individually, this step is covered off by the sound network administration practices. If the analyst does not understand what needs to be protected, there is no possibility of success.

- Aim
- Scope
- Boundary

- Gathering Information
- System Description
- Targeted risk and required certainty

In the strategic environment, it is clear that there will be significant work to capture all of the alterations that have occurred since the previous time the process was run. While this is not the proper way to run a risk management methodology, it is often a fact of life.

In the dynamic environment the planning step may be considered as nearly unchanged. Here the question will be "What changed in the network today?" instead of "What does my network look like?"

This step is still performed informally in many organizations today. Whether this information is documented, or just stored in the gray matter of the network guru, it is critical that the information be current. The risk management process forces this information to be captured onto a media which eliminates the single point of failure should your network guru find a better paying career, get downsized, or play tag with a greyhound bus.

Operations and Maintenance (O&M)

This is where an IDS system would fit in. As previously mentioned, an IDS system resembles a miniature risk management methodology. O&M examples include configuration management, backup, hardware/software maintenance/upgrades, security monitoring, and security verification. Security monitoring and verification includes user education, antivirus software, host and network vulnerability scans, red (Tiger) teams and intrusion detection systems.

In adapting this risk management model to a tactical environment, there are a few differences in terminology. The "Change required" output maps to an "event of interest" from the IDS. This is the trigger event for running the model. In the case where multiple events occur simultaneously (e.g. multiple events are found while processing the IDS log), the events would be processed together.

Threat and Risk Assessment

After a quick review of the "State of the network" with your toolbox of goodies to look for new connections, networks, computers, etc., it is time to run the events of interest through the TRA process (or the SANS severity process). Including this step on a daily basis can save time in the analysis step.

This step most closely resembles IDS event analysis. While the TRA process outlined in [CSE1] follows a different paradigm than the severity equation in [SANS1] both have the same end goal. The differences between the two processes are discussed in more detail below. The output of this step is a prioritized list of events with an indication of severity (or Impact and Likeliness using the Canadian model.)

Managing Risk

The right hand column of Figure 1 includes several steps, which may be summarized as the process of managing risk. These steps outline a decision process that includes the acceptance, reduction, avoidance or transfer of the risk. While Figure 1 suggests a strategic view of the world, this process is equally suitable for a tactical or short-term approach as well.

This step is designed to formalize the decision process in order to avoid the application of Band-Aid fixes. The process forces the administrator to consider all aspects of security instead of relying solely on network or computer security patches.

Accreditation

Accreditation is normally more of a strategic concept rather than tactical. Keeping this step in the process is important as a reminder of the administrator's responsibility for configuration management of the network. Additionally, it is important to remember the conditions under which the system accreditation was originally granted. While an operating system upgrade to a new version may enhance computer security, the upgrade may also create new vulnerabilities.

Integration of an IDS into the Risk Management Process

The modeling of an IDS system as a miniature risk management process is only part of the solution. What is still needed is a single integrated process that covers all aspects of risk management. The difficulty is in combining the two without impeding the responsiveness of the IDS. To accomplish this, we need to understand how and where the Tactical and Strategic components of this system need to interoperate.

A full TRA process on a large network could take several weeks or even months to complete. The IDS event analysis needs to be completed in the order of minutes or hours. The process described here depends on the initial completion of a TRA for the network.

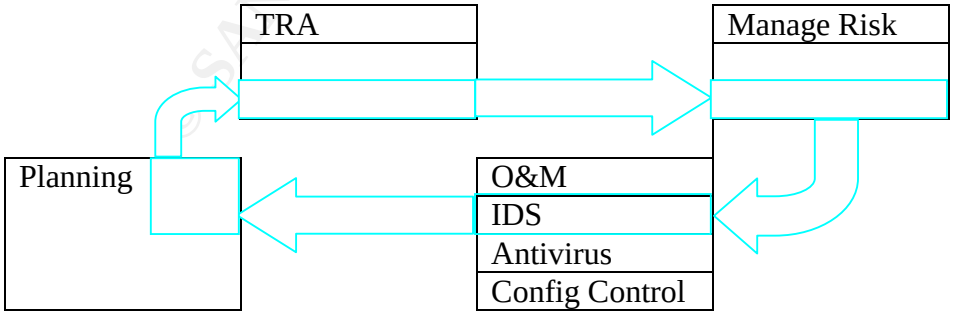


Figure 3: Integration Process

Prior to starting event analysis it is important to answer the question "What changed in the network today?" In this manner, the planning step shown above will be updated on a regular basis.

When an event of interest is flagged by the IDS system, it is compared with the original summary table of the TRA to determine whether the item was considered in the previous TRA. If the event type was not considered, make a note to add this information. If the event type was considered, compare the predicted Threat with any information that was gathered about the event (or try to gather more information while analyzing the event.) Compare the impact and likeliness with the actual impact and rate of occurrence for the event.

For events that require remedial action to resolve an existing vulnerability the Manage Risk steps are utilized. It is important to remember the targeted residual risk from the original TRA. When the event type was considered during the initial TRA, the safeguard selection process corresponding to that vulnerability needs to be reviewed to determine whether system and network security is still sufficient.

Information to be shared

The network documentation from the Planning step, the TRA summary table, and the "Residual Risk" of the initial RM/TRA process are the three major inputs to the IDS. In return, the IDS provides a valuable tool for the validation of the vulnerability analysis performed in the TRA. The IDS output can provide the volume and type of malicious traffic being seen by the IDS sensor. Additionally, indications of new threats/vulnerabilities, which may not have been considered in the original TRA, will be captured through the use of a single TRA/EA process. Finally, in the event of a detected compromise, details of the attack from the IDS will be used as input to the RM process for determining the appropriate actions to fix the compromise.

Choosing the analysis methodology that is right for you

There is no right or wrong answer to the method chosen for event analysis or for a TRA. The methodology that is right for the financial industry is not necessarily right for a government or the healthcare industry. Features of the industry sector you work in will likely determine the methodology you need to implement. In order to determine which process is right for you, consider the following points.

The Ranking System: Qualitative vs. Quantitative

This subject is the most difficult portion of both the TRA and the EA processes. What differentiates a 1 from a 5 or a high from a low in any of these systems. From personal experience performing TRAs the use of numbers implies a quantitative approach, which is interpreted as meaning that a 4 is twice as significant as a 2 (either good or bad).
Financial institutions will tend to assign dollar values and probabilities of occurrence to the outputs of the event analysis. This is probably the most

quantitative approach available today as the end result is a predicted dollar loss. Governments on the other hand tend to have information as their major asset. This information is normally ranked in importance from Unclassified to Top Secret. Protection of this information is normally based on Policies or standards of minimum protection. The use of dollar values in this case is not appropriate since it is very difficult to place a dollar value on a national secret. In this case a scale of high to low in an arbitrary number of steps would be more appropriate.

The Ranking System: Cumulative vs. Multiplicative

Financial institutions implementing a quantitative TRA may utilize a multiplicative approach in order to define an expected loss per year or per event. The formula for determining the seriousness of an event would be in the form of Cost per event multiplied by the predicted rate of occurrence of the event. With qualitative approaches, the choice of addition vs. multiplication becomes a much more vague concept.

Compliance Requirements

Prior to choosing a TRA methodology or product, ensure that you understand your end compliance goal. If your goal is to improve system security voluntarily, then your range of choices remains wide. If your organization or government requires compliance with a national or ISO standard, then you will need to choose a methodology in line with that standard. The most widely recognized, high level standard in the area of information security is currently the ISO 17799 "Code of Practice for Information Security Management" [ISO1]

Conclusions

Integration of your IDS into your risk management framework is an important step in achieving
Selection of a risk management process is as difficult as the selection of an IDS system. The good news is that they are all free for the adoption. There are several references available on the Internet along with other relevenat papers in the SANS reading room.

References and Related Reading:

[CSE1] Communications Security Establishment, "A Guide to Security Risk Management for Information Technology Systems," 1996. http://www.cse.dnd.ca/

[CSE2] Communications Security Establishment, "A Guide to Risk Assessment And Safeguard Selection for Information Technology Systems," 1996. http://www.cse.dnd.ca/

[ISO1] International Standard ISO/IEC 17799:2000 Code of Practice for Information Security Management, 2000.

[SANS1] Intrusion Detection Immersion Curriculum – SANS, (Track 3 Intrusion Detection In-Depth, 2001.

Prabhacker, Rajasingham, “Threat and Risk Assessments: Some Issues”, April 2001, http://www.sans.org/infosecFAQ/audit/risk.htm

End Notes

[1] CSE, “A Guide to Security Risk Management for Information Technology Systems,” 1996.
[1] CSE. “A Guide to Risk Assessment And Safeguard Selection for Information Technology Systems,” 1996

Question 2 Network detects

Source of Trace:
Location of file

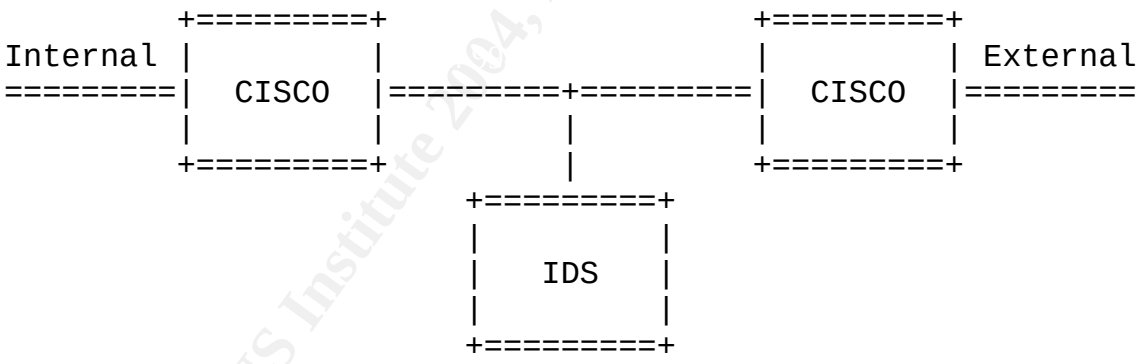
This detect was extracted from the file 2002.9.28 downloaded from http://www.incidents.org/logs/Raw/.

Further supporting information (in the form of 2 more very similar detects were obtained from the file 2002.9.29 from the same web site.

For the sake of clarity the network layout and description of the detect are based on the first file.

Network Layout

All of the captured packets contain two MAC addresses beginning with [00:00:0C] on the internal side and [00:03:E3] on the external side. When these two MAC address fragments are referenced to a vendor MAC address table (http://www.coffer.com/mac_find/) the resulting vendor is CISCO in both cases. This means that the IDS probe is physically located between 2 CISCO routers with no other communicating hosts between the two routers. The internal network appears to be the address range of 32.245.0.0/16



Detect was generated by:

Raw Detect Information

Snort instusion detection system version 2.1.2 for Win32 with the default rule set. Rule triggered is

```
[**] [116:97:1] (snort_decoder): Short UDP packet, length field > payload
length [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
10/28-04:10:12.336507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x5C
80.63.124.198:0 -> 32.245.98.91:0 UDP TTL:112 TOS:0x0 ID:314 IpLen:20 DgmLen:78
Len: 129
```

"Raw" packet extracted from SNORT without ruleset (based on a search for packets from 80.63.124.198 turned up the following single datagram:

```
10/28-04:10:12.336507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x5C
80.63.124.198:0 -> 32.245.98.91:0 UDP TTL:112 TOS:0x0 ID:314 IpLen:20 DgmLen:78
0x0000: 0E 25 00 00 00 89 00 3A 4D 16 01 00 00 10 00 01  .%.....:M.......
0x0010: 00 00 00 00 00 00 20 43 4B 41 41 41 41 41 41 41  ...... CKAAAAAAA
0x0020: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
0x0030: 41 41 41 41 41 41 41 00 00 21                    AAAAAAA..!
```

As an interesting bit of trivia, note the error in Snort indicating a source port of 0 instead of 3621. To eliminate all confusion, here is the real packet extracted with WinDump:

```
windump -xXn -r ..\in\2002.9.28 -s 0 net 80.63.0.0/16
04:10:12.336507 IP 80.63.124.198.3621 > 32.245.98.91.0: udp 129
0x0000   4500 004e 013a 0000 7011 e4f8 503f 7cc6        E..N.:..p...P?|.
0x0010   20f5 625b 0e25 0000 0089 003a 4d16 0100        ..b[.%.....:M...
0x0020   0010 0001 0000 0000 0000 2043 4b41 4141        ...........CKAAA
0x0030   4141 4141 4141 4141 4141 4141 4141 4141        AAAAAAAAAAAAAAAA
0x0040   4141 4141 4141 4141 4141 4100 0021            AAAAAAAAAAA..!
```

Explanation of Trigger Stimulus

Bytes 2 and 3 of the captured packet indicate that the total length of the IP packet and included data are 0x0043 (78 bytes) in length (highlighted in yellow). However, Bytes 24 and 25 (Decimal) indicate that the UDP datagram length is 0x0089 (137 Bytes) in length. As this is larger than the entire 78 byte length of the IP datagram, an alert is triggered.

While the malformed packet size information is of interest, the remainder of the packet information is of more interest and is actually irrelevant to the original detect. This will be discussed in much more detail in section 4.

Probability the source address was spoofed:

Type of Detect

UDP attacks directed at port 0 are unlikely to be anything other than an attempt at a DOS attack. Based on that assumption, the source address would highly likely be spoofed. The only reference to this I have found to Port 0 DOS attacks is CVE-1999-0675 (detailed in section 4 below) which refers to a Checkpoint Firewall -1 vulnerability. As there is no evidence of a VPN involved in this configuration, and as this vulnerability is now approaching 5 years old, this is not likely to be the actual attack implemented here.

However, based on "google" searches for information on the payload (CKAAAAAA…), I believe that this is an unsuccessful attempt at a crafted packet, and that the detection based on the current rule set was purely coincidental. As will be shown in the analysis in section 4, this attack was likely intended as an information gathering exercise prior to the launching of a real attack. Based on that assumption, the IP address(es) would not be spoofed. More detail to justify the validity of this assumption is provided in section 4.

Potential Source of Attack

inetnum: 80.63.124.0 - 80.63.124.255
netname: TELEDANMARK-ADSL-USERS
descr: TDC NetExpres users
country: DK
Source IP address from 2002.9.28 file

IP Address: 211.223.8.0-211.223.8.255
Network Name: KORNET-INFRA000001
Connect ISP Name: KORNET
Connect Date: 20031129
Registration Date: 20031209
Source IP address from 2002.9.29

The two source IP addresses in question resolve to the two geographically distinct areas, though caution must be used as the event date predates the Korean registration. There are not sufficient numbers of events to make a positive determination, but this still looks to me like the IP addresses are not spoofed due to the apparent information gathering mission.

Description of attack:

The analysis of this attack is somewhat speculative as there were two distinct oddities to this packet which seem to be unrelated. There is the incorrect size of the packet and the apparent insertion of 2 bytes within the UDP header. These two irregularities lead to this detect having similarities to the following detects:

- Port 0 DOS / Packet length mismatch
- System Scan / SMB Name Wildcard

DOS

Based on the packet sent, there is a small possibility that this was a DOS attack against the router. If the packet size had been correct, the following alert would have triggered based on the destination port being 0.

alert udp $EXTERNAL_NET any <> $HOME_NET 0 (msg:"BAD-TRAFFIC udp port 0 traffic"; reference:cve,CVE-1999-0675; reference:nessus,10074; classtype:misc-activity; sid:525; rev:5;)

In its current form, the UDP datagram length rule triggers before the Port 0 rule is parsed. As a result the port 0 rule is not triggered.

There is no evidence that the destination host even exists. Therefore, it is impossible to confirm or deny the DOS intent of this packet with this level of detail. However, this would be a stretch of the imagination. The one clear item from the log files; if this was a DOS attack against the router, it did not succeed.

System Scan

The fact that a single packet was sent in a one day period, and two packets of a very similar nature were sent on the second day, makes me believe that this is an attempt at a slow scan. It is quite possible that there is a script tool (with a bug?) which is implementing this scan. An interesting exercise would be to send this packet to an existing system and look for a response to determine if this is really a bug, or an attempt at subterfuge to throw the IDS analyst off the scent.

If the 2 bytes at offset 22 and 23 were deleted and the remaining datagram were shifted left accordingly (to correct the "bug"), the packet would have been destined for port 137. With this assumption, the remainder of the packet would have made more sense as the packet length would then be 0x003a (58 bytes) which would have matched the IP datagram length of 78 bytes (58 bytes UDP datagram + 20 bytes IP header.) The resulting packet (with the last two bytes intact would be the port 137 SMB Name Wildcard attack. The default rule set shipped with Snort 2.1.12 does not appear to have a rule to detect this attack

Evidence of Packet Crafting

The following two packets were captured from the 2002.9.29 data file (the next day.) The existence of two packets in close proximity allows us to perform a little more analysis.

```
[**] (snort_decoder): Short UDP packet, length field > payload length
[**]
10/29-21:16:27.826507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x5C
211.223.8.214:0 -> 32.245.161.79:0 UDP TTL:109 TOS:0x0 ID:25767
IpLen:20 DgmLen:78
Len: 129

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

[**] (snort_decoder): Short UDP packet, length field > payload length
[**]
10/29-21:16:33.676507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x5C
211.223.8.214:0 -> 32.245.161.117:0 UDP TTL:109 TOS:0x0 ID:62887
IpLen:20 DgmLen:78
Len: 129

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+
```

```
C:\Snort\bin>windump -xXn -r ..\in\2002.9.29 -s 0 net 211.223.0.0/16
21:16:27.826507 IP 211.223.8.214.1026 > 32.245.161.79.0: udp 129
0x0000   4500 004e 64a7 0000 6d11 33e8 d3df 08d6        E..Nd...m.3.....
0x0010   20f5 a14f 0402 0000 0089 003a 0696 0100        ...O.......:....
0x0020   0010 0001 0000 0000 0000 2043 4b41 4141        ..........CKAAA
0x0030   4141 4141 4141 4141 4141 4141 4141 4141        AAAAAAAAAAAAAAAA
0x0040   4141 4141 4141 4141 4141 4100 0021            AAAAAAAAAAA..!
21:16:33.676507 IP 211.223.8.214.1026 > 32.245.161.117.0: udp 129
0x0000   4500 004e f5a7 0000 6d11 a2c1 d3df 08d6        E..N....m.......
0x0010   20f5 a175 0402 0000 0089 003a 0670 0100        ...u.......:.p..
0x0020   0010 0001 0000 0000 0000 2043 4b41 4141        ..........CKAAA
0x0030   4141 4141 4141 4141 4141 4141 4141 4141        AAAAAAAAAAAAAAAA
0x0040   4141 4141 4141 4141 4141 4100 0021            AAAAAAAAAAA..!
```

The source port is the same for both packets (highlighted in Yellow), and the "error" in all three packets is identical. These are crafted packets.

Apparent bug in recent versions of Snort

All three detected packets have source ports of 0 reported by Snort. However, the packets do not have this source port as can be seen in the Windump packets.

Attack mechanism:

DOS

Packet size mismatches are common denominators in several attacks. Normally they attempt to cause a crash, or cause the execution of selected code. In this case, a poor UDP implementation would be left with a buffer under run possibly resulting in random data if the buffer was not properly initialized. This random data could theoretically cause a crash.

SMB Name Wildcard

This is the normal structure of the Netbios name query. When resolving a name with only the IP address available, windows machines will send these queries as part of normal operations. The notable string in the udp datagram "CKAAA…" is generated from a null Netbios name, "00 00 00…" as a wildcard with a translation function being performed to complete the mapping.

However, when these queries arrive from an external network, they represent a threat to the network based on the information provided in the response which could include:

(Source: http://www.digitaltrust.it/anachnids/IDS177/research.html)

1. The NetBIOS name of the server.

2. The Windows NT workgroup domain name.

3. Login names of users who are logged into the server.

4. The name of the administrator account if they are logged into the server.

The detected packets are not generated by the standard netbios name server. Instead they have been crafted to provide a more directed approach to scanning a network.

Correlations:

The SMB Name Wildcard, the Port 0 DOS, and the packet length mismatches are not new attacks.

SMB wildcard attacks have been reported quite extensively from 1999 onwards. The following links provide a quite good analysis.

http://www.sans.org/resources/idfaq/port_137.php

http://www.digitaltrust.it/arachnids/IDS177/research.html

http://www.giac.org/practical/GCIA/Sebastien_Pratte_GCIA.pdf

http://www.finchhaven.com/pages/incidents/030102_udp_137.html

CAN-1999-0621

The port 0 DOS against the Checkpoint Firewall-1 was documented in 1999.

CVE-1999-0675

The detected packet (including the apparent error has been previously detected by Andrew Evans on 9 July 2003

(http://cert.uni-stuttgart.de/archive/intrusions/2003/07/msg00071.html) and by

Reto Baumann on 7 Mar 2003.

(http://cert.uni-stuttgart.de/archive/intrusions/2003/03/msg00090.html)

Evidence of active targeting:

This attack does not represent active targeting of particular systems. Quite the opposite, this attack is the general fishing exercise (looking for systems, and looking for any response to the probing.)

Severity:

severity = (criticality + lethality) (system countermeasures + network countermeasures)

Criticality 1: This attack is designed to perform a slow scan of a network while attempting to mislead the analyst of the real intent. However, at best, only the existence of a system on the network would be determined. Had the packets been crafted correctly, the attack would have also yielded more information relating to the roles of the detected systems (e.g. workstations vs servers. This correction would result in the Criticality being raised to 2 in my opinion.

Lethality 1: This attack only provides mapping information for targeting a future directed attack against the network. There is no immediate damage to the system, only a reduction in "security by obscurity."

System countermeasures 2: There are no responses to these packets. However, as there are only 3 captured packets over a 2 day period that does not provide a warm fussy feeling that the internal hosts are well protected. Based on the network configuration discussed below, I am not inclined to be charitable in the area of system countermeasures.

Network countermeasures 2: As the network owner has created the RAW log files for our studies, it is a safe assumption that there actually is an IDS system located at the same location. However, in order to limit the vast number of low tech attacks and noise caused by the lack of good networking practices out on the Internet, I would expect to see a better configuration on the CISCO router connecting this network to the INTERNET. Based on sound practices for router configurations, I would not expect to see source broadcast addresses nor LAN protocols entering from the WAN. The current configuration allows ports 137 and 139 in through the external router, along with source broadcast addresses. This adds an unnecessary burden on the security analyst.

Overall (1 + 1) - (2+2) = - 2. The most formulated working variation on this detect is the SMB Name Wildcard which is effectively a pre attack probe. While the countermeasures did not score well, this is partially due to the unknown nature of the scanned network.

Defensive recommendation:

The network involved with this detect appears to be quite large. The location of the IDS sensor appears to be as close to the outside of the network as possible. As such, a second IDS inside the firewall is needed to determine what makes it into the network. In this manner the analyst has a better picture of where rules need to be improved while still providing insight as to who is knocking on the door.

In addition, there are some basic housekeeping rules which need to be implemented on the external router. Even if the router is owned by the ISP, these recommendations should still be implemented. First: LAN protocols and addresses do not belong on the WAN. This means that the standard network O/S protocols such as TCP ports 135, 139, 445 and UDP ports 137, 138, and 445 should not pass through the external router in either direction (as source or destination ports.) Private and Wildcard IP addresses should also be blocked in both directions through the external firewall.

These rules should, as much as possible, also be implemented on the internal router. In the case where several internal routers converge to a single external router, this may not always be possible to completely implement.

Multiple choice test question:

```
21:16:27.826507 IP 211.223.8.214.1026 > 32.245.161.79.0: udp 129
0x0000   4500 004e 64a7 0000 6d11 33e8 d3df 08d6        E..Nd...m.3.....
0x0010   20f5 a14f 0402 0000 0050 0696 0100 0010        ...O.....:......
0x0020   0001 0000 0000 0000 2043 4b41 4141 4141        .........CKAAAAA
0x0030   4141 4141 4141 4141 4141 4141 4141 4141        AAAAAAAAAAAAAAAA
0x0040   4141 4141 4141 4100 0021 0021                  AAAAAAAAA..!..
```

Q: What rule will trigger based on the above packet?

A) [**] [116:97:1] (snort_decoder): Short UDP packet, length field > payload length [**]
B) [13]alert udp $EXTERNAL_NET any <> $HOME_NET 0 (msg:"BAD-TRAFFIC udp port 0 traffic"; reference:cve,CVE-1999-0675; reference:nessus,10074; classtype:m isc-activity; sid:525; rev:5;)
C) Both
D) None

A: A

The Snort decoder runs before the other rule files. A questions has been raised regarding processing order of Snort version 1.9.1, but I have not been able to confirm this behavior)

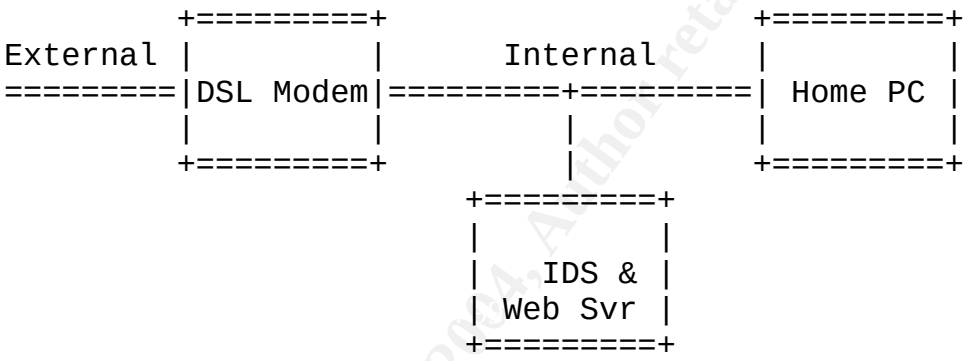
Source of Trace #2:

Location of trace

This detect was extracted from a tcpdump capture off of my home network on 22.04.2004 between 08:52:48 (UTC) and 19:53:29 (UTC).

Network Layout

My home network consists of a DSL modem, hub, and 2 computers. The first computer is a general purpose system (not logged into the DSL modem during the capture timeframe.) The second computer is configured as an IDS analysis system with Snort, Apache, MySQL, ACID, and EtherPeek. As the described configuration was not generating interesting packets needed to complete my assignment, the personal firewall was turned off for the duration of the capture. IP address assignment is dynamic, so there was no advance indication to the world that this system with several listening services would be live.



Detect was generated by:

Snort intrusion detection system version 2.1.2 for Win32.

There are three detects analyzed here as they are actually parts of the same attack. I originally selected the first detect shown here as I could not find a significant amount of analysis already performed. However, shortly into the analysis, I determined a link with an alert on an earlier packet. Later on when I started to gather more technical evidence, I found another detect was also included in the same data stream. As such the three alerts are included together (not in chronological order.)

First Selected rule triggered is

```
[**] [119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
04/22-08:14:09.456949 0:2:3B:2:67:D2 -> 0:10:A7:5:62:CA type:0x8864
len:0x5DE
211.45.217.3:54291 -> 217.136.26.209:80 TCP TTL:107 TOS:0x0 ID:5528
IpLen:20 DgmLen:1480 DF
***A**** Seq: 0x6EEA5655  Ack: 0xBF8BFB3F  Win: 0xFFFF  TcpLen: 20
```

The raw packet from WinDump is:

```
08:14:09.456949 PPPoE  [ses 0x9616] IP 1482: IP 211.45.217.3.54291 >
XXX.XXX.XXX.209.80: . 1441:2881(1440) ack 1 win 65535 (DF)
0x0000   1100 9616 05ca 0021 4500 05c8 1598 4000        .......!E.....@.
0x0010   6b06 540d d32d d903 0000 00d1 d413 0050        k.T..-.........P
0x0020   6eea 5655 bf8b fb3f 5010 ffff 7d98 0000        n.VU...?P...}...
0x0030   b102 b102 b102 b102 b102 b102 b102 b102        ................
.
.
.
0x02f0   b102 b102 b102 b102 b102 b102 b102 b190        ................
0x0300   9090 9090 9090 9090 9090 9090 9090 9090        ................
.
.
.
0x05c0   9090 9090 9090 9090 9090 9090 9090 9090        ................
```

In a more user friendly format for display (Etherpeek): this packet looks like:

```
Packet Info
  Flags:                0x00
  Status:               0x00
  Packet Length:        1506
  Timestamp:            09:14:09.456949000 04/22/2004
Ethernet Header
  Destination:          00:10:A7:05:62:CA  Unex Tech:05:62:CA
  Source:               00:02:3B:02:67:D2  Redback Net:02:67:D2
  Protocol Type:        0x8864  PPPoE Session
PPP over Ethernet
  Version:              1
  Type:                 1
  Code:                 0x00  Session
  Session Id:           38422
  Length:               1482
Point-to-Point Protocol
  PPP                   0x0021  IP, Internet Protocol
IP Header - Internet Protocol Datagram
  Version:              4
  Header Length:        5  (20 bytes)
  Differentiated Services:%00000000
                        0000 00.. Default
                        .... ..x. Reserved
                        .... ...x Reserved

  Total Length:         1480
  Identifier:           5528
  Fragmentation Flags:  %010
                        0.. Reserved
                        .1. Do Not Fragment
                        ..0 Last Fragment

  Fragment Offset:      0  (0 bytes)
  Time To Live:         107
  Protocol:             6  TCP - Transmission Control Protocol
  Header Checksum:      0x540D
  Source IP Address:    211.45.217.3
  Dest. IP Address:     XXX.XXX.XXX.209
TCP - Transport Control Protocol
  Source Port:          54291
  Destination Port:     80  http
```

```
  Sequence Number:        1860851285
  Ack Number:             3213622079
  TCP Offset:             5  (20 bytes)
  Reserved:               %000000

  TCP Flags:              %010000  .A....
                          0. .... (No Urgent pointer)
                          .1 .... Ack
                          .. 0... (No Push)
                          .. .0.. (No Reset)
                          .. ..0. (No SYN)
                          .. ...0 (No FIN)

  Window:                 65535
  TCP Checksum:           0x7D98
  Urgent Pointer:         0
  No TCP Options
HTTP - Hyper Text Transfer Protocol
  Continuation of existing HTTP
stream<STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><ST
X><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><S
TX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><
STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX>
<STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX
><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><ST
X><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><S
TX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><
STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX>
<STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX
><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><ST
X><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><S
TX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><
STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX>
<STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX
><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><ST
X><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><S
TX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><
STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX>
<STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX><STX
><STX><STX><STX><STX>
FCS - Frame Check Sequence
  FCS:                    0x00000000  Calculated
```

After a short examination of the related packets the following detect is also seen to be related to the same attack as it appears immediately prior to the first detect and within the same http session.

Detect #2 rule triggered is (Packet previous to detect #1):

```
[**] [119:15:1] (http_inspect) OVERSIZE REQUEST-URI DIRECTORY [**]
04/22-08:14:09.452902 0:2:3B:2:67:D2 -> 0:10:A7:5:62:CA type:0x8864
len:0x5DE
211.45.217.3:54291 -> 217.136.26.209:80 TCP TTL:107 TOS:0x0 ID:5527
IpLen:20 DgmLen:1480 DF
***A**** Seq: 0x6EEA50B5  Ack: 0xBF8BFB3F  Win: 0xFFFF  TcpLen: 20
```

The Etherpeek decode of this packet is as follows.

```
Packet Info
  Flags:                0x00
  Status:               0x00
  Packet Length:        1506
  Timestamp:            09:14:09.452902000 04/22/2004
Ethernet Header
  Destination:          00:10:A7:05:62:CA  Unex Tech:05:62:CA
  Source:               00:02:3B:02:67:D2  Redback Net:02:67:D2
  Protocol Type:        0x8864  PPPoE Session
PPP over Ethernet
  Version:              1
  Type:                 1
  Code:                 0x00  Session
  Session Id:           38422
  Length:               1482
Point-to-Point Protocol
  PPP                   0x0021  IP, Internet Protocol
IP Header - Internet Protocol Datagram
  Version:              4
  Header Length:        5  (20 bytes)
  Differentiated Services:%00000000
                        0000 00.. Default
                        .... ..x. Reserved
                        .... ...x Reserved

  Total Length:         1480
  Identifier:           5527
  Fragmentation Flags:  %010
                        0.. Reserved
                        .1. Do Not Fragment
                        ..0 Last Fragment

  Fragment Offset:      0  (0 bytes)
  Time To Live:         107
  Protocol:             6  TCP - Transmission Control Protocol
  Header Checksum:      0x540E
  Source IP Address:    211.45.217.3
  Dest. IP Address:     217.136.26.209
TCP - Transport Control Protocol
  Source Port:          54291
  Destination Port:     80  http
  Sequence Number:      1860849845
  Ack Number:           3213622079
  TCP Offset:           5  (20 bytes)
  Reserved:             %000000

  TCP Flags:            %010000  .A....
                        0. .... (No Urgent pointer)
                        .1 .... Ack
```

```
                        .. 0... (No Push)
                        .. .0.. (No Reset)
                        .. ..0. (No SYN)
                        .. ...0 (No FIN)

  Window:               65535
  TCP Checksum:         0xD044
  Urgent Pointer:       0
  No TCP Options
HTTP - Hyper Text Transfer Protocol
  Continuation of existing HTTP stream
  Binary Data:
  SEARCH /........  53 45 41 52 43 48 20 2F 90 02 B1 02 B1 02 B1 02
  ................  B1 02 B1 02 B1 02 B1 02 B1 02 B1 02 B1 02 B1 02
.
.
.
  ................  B1 02 B1 02 B1 02 B1 02 B1 02 B1 02 B1 02 B1 02
FCS - Frame Check Sequence
  FCS:                  0xE64D141C  Calculated
```

The third detect

```
[**] [1:648:6] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
04/22-08:14:09.762991 0:2:3B:2:67:D2 -> 0:10:A7:5:62:CA type:0x8864
len:0x5DE
211.45.217.3:54291 -> 217.136.26.209:80 TCP TTL:107 TOS:0x0 ID:5611
IpLen:20 DgmLen:1480 DF
***A**** Seq: 0x6EEA5BF5  Ack: 0xBF8BFB3F  Win: 0xFFFF  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS181]
```

In the interest of space I will not include the decode here. Suffice it to say that the http data section of the packet included 1440 “no operation” (NOOP) (0x90) codes. On average there were 15 of these packets per attack. The combination of these three detects were found two further times within the file.

Explanation of Trigger Stimulus

Detect #1

Normally, the Http command included within this packet should have begun with a valid HTTP method such as GET. Instead, the first data packet is simply a binary data stream. The HTTP 1.1 methods are defined in RFC2616. The omission of the HTTP method triggers the rule “Bare Byte Unicode Encoding”

Detect #2

The http data field within this packet starts with the string “SEARCH /” which conforms to the http method encoding standard. Therefore, the rule “Bare Byte Unicode Encoding” does not fire. While the HTTP method “search” in this packet is not in the HTTP 1.1 specification, it is included in the “Web-based Distributed Authoring and Versioning” (WebDAV) specification. However, the length of the http data field is larger than the configured maximum for a directory query which is programmed as a default of 300 in the preprocessor section of the Snort.conf

file. This triggers the “OVERSIZE REQUEST-URI DIRECTORY.” As this rule triggers before the bulk of the ruleset is applied to the packet, rule 1500 does not fire. (I am waiting for a Win32 binary for Snort 2.1.13 to fix this one)

Detect #3

Within the HTTP data field of the packet are a large number of (0x90) binary instructions which translate to the (NOOP) instruction in x86 assembly language. A NOOP instruction is simply an instruction which tells the CPU to wait one command interval and then load and execute the next command. Anywhere outside of a wait loop in a program, this is always a sign of bad things to come.

Probability the source address was spoofed:

For this attack to succeed, the attacker must complete a three way handshake and push data into a vulnerable web server. In the absence of predictable sequence numbers this is not possible. Once the three way handshake has been successfully completed, and the offending packet(s) pushed to the web server, the next step is to try to open a telnet session to the hacked machine on the designated port (e.g.31337) to determine if the attack resulted in a compromise instead of just a crash. The end conclusion is that the source address is unlikely to be spoofed.

Potential Sources of Attacks

A query to http://ww2.arin.net/whois/ provides the following information on the three sources for the detected packets.

Search results for: 211.45.217.3

```
inetnum:      211.45.192.0 - 211.45.255.255
netname:      BORANET-NET-211-45-192
descr:        DACOM Corp.
descr:        Facility-based Telecommunication Service Provider
descr:        providing Internet leased-ine, on-line service, BLL etc.
country:      KR
admin-c:      DB50-AP
tech-c:       DB50-AP
mnt-by:       APNIC-HM
mnt-lower:    MAINT-KR-DACOM
changed:      hm-changed@apnic.net 20021202
status:       ALLOCATED PORTABLE
source:       APNIC
```

Search results for: 24.213.249.141

```
OrgName:    Road Runner-Commercial
OrgID:      RCNY
Address:    13241 Woodland Park Road
City:       Herndon
StateProv:  VA
```

```
PostalCode: 20171
Country:    US

ReferralServer: rwhois://ipmt.rr.com:4321

NetRange:   24.213.128.0 - 24.213.255.255
CIDR:       24.213.128.0/17
NetName:    RR-COMMERCIAL-NYC-3
NetHandle:  NET-24-213-128-0-1
Parent:     NET-24-0-0-0-0
NetType:    Direct Allocation
NameServer: NS1.BIZ.RR.COM
NameServer: NS2.BIZ.RR.COM
NameServer: DNS4.RR.COM
Comment:
RegDate:    2003-03-06
Updated:    2003-08-29
```

Search results for: 217.81.77.151

```
inetnum:       217.80.0.0 - 217.89.31.255
netname:       DTAG-DIAL14
descr:         Deutsche Telekom AG
country:       DE
admin-c:       DTIP
tech-c:        DTST
status:        ASSIGNED PA
remarks:
*************************************************************
remarks:       * ABUSE CONTACT: abuse@t-ipnet.de IN CASE OF HACK
ATTACKS, *
remarks:       * ILLEGAL ACTIVITY, VIOLATION, SCANS, PROBES, SPAM, ETC.
*
remarks:
*************************************************************
mnt-by:        DTAG-NIC
changed:       ripe.dtip@telekom.de 20001026
changed:       ripe.dtip@telekom.de 20030211
source:        RIPE
route:        217.80.0.0/12
descr:         Deutsche Telekom AG, Internet service provider
origin:        AS3320
mnt-by:        DTAG-RR
changed:       rv@NIC.DTAG.DE 20001027
source:        RIPE
person:       DTAG Global IP-Addressing
address:       Deutsche Telekom AG
address:       D-90492 Nuernberg
address:       Germany
phone:         +49 180 5334332
fax-no:        +49 180 5334252
e-mail:        ripe.dtip@telekom.de
nic-hdl:       DTIP
mnt-by:        DTAG-NIC
changed:       ripe.dtip@telekom.de 20031013
```

```
source:       RIPE
person:       Security Team
address:      Deutsche Telekom AG
address:      Germany
phone:        +49 180 5334332
fax-no:       +49 180 5334252
e-mail:       abuse@t-ipnet.de
nic-hdl:      DTST
mnt-by:       DTAG-NIC
changed:      abuse@t-ipnet.de 20030210
source:       RIPE
```

The three source IP addresses resolve to three geographically distinct areas, specifically Korea, United States, and Germany.

Description of attack:

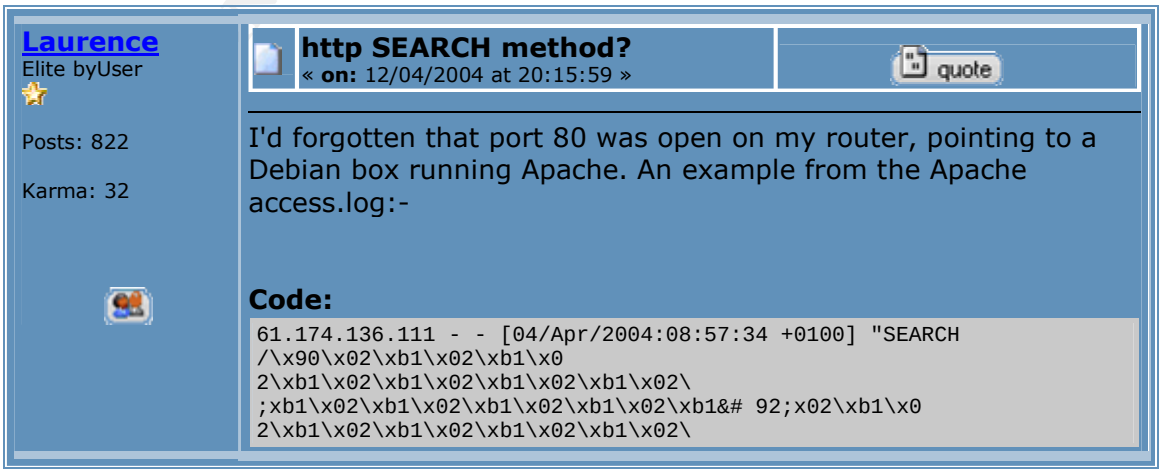
To clarify the attack, the detects will now be reordered into the originally received order instead of the order upon which the detects were selected for analysis.

THREE WAY HANDSHAKE (No detect)

One of the requirements for this attack is the establishment of an HTTP session which of course requires the three way handshake. These packets were easily found in the raw packet logs. No detects are linked to this activity as this was a connection to an existing HTTP server with Snort configured with the line "var HTTP_SERVERS $HOME_NET" in the snort.conf file.

OVERSIZE REQUEST-URI DIRECTORY

The next packet received by the web server began with the information "SEARCH /..... " where the information trailing the "/" was a stream of hex bytes starting with 0x90, 0x02, 0xb1, 0x02, 0xb1,,, After a bit of a tutorial read on http methods from various sources, I finally found reference to the search method being part of WebDAV by a google search on the keywords ("http method" search x02 xb1") The messages thread refers to WebDAV and Microsoft Security Bulletin MS03-007.



Laurence
Elite byUser

Posts: 822

Karma: 32

http SEARCH method?
« **on:** 12/04/2004 at 20:15:59 »

quote

I'd forgotten that port 80 was open on my router, pointing to a Debian box running Apache. An example from the Apache access.log:-

Code:

```
61.174.136.111 - - [04/Apr/2004:08:57:34 +0100] "SEARCH
/\x90\x02\xb1\x02\xb1\x0
2\xb1\x02\xb1\x02\xb1\x02\xb1\x02\
;xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1&# 92;x02\xb1\x0
2\xb1\x02\xb1\x02\xb1\x02\xb1\x02\
```

```
;xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1&# 92;x02\xb1\x0
2\xb1\x02\xb1\x02\xb1\x02\xb1\x02\
;xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1&# 92;x02\xb1\x0
2\xb1\x02\xb1\x02\xb1\x02\xb1\x02\
;xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1&# 92;x02\xb1\x0
2\xb1\x02\xb1\x02\xb1\x02\xb1\x02\
;xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1&# 92;x02\xb1\x0
2\xb1\x02\xb1\x02\xb1\x02\xb1\x02\
;xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1&# 92;x02\xb1\x0
-------------- ;------------snip- -------------- ------

About 6 screenfuls of the same

-------------- ;-------------- ;snip------------- ---
\x90\x90\x90\x90\x90\x90\x90\x90\
x90\x90\x90\x90\x90\x90\x90\x90\x90 2;x90\x90\x9
0\x90\x90\x90\x90\x90\x90\x90\x90\
;x90\x90\x90\x90\x90\x90\x90\x90\x90&q uot; 414 340 "
-" "-"
```

Now this looks like a buffer overrun attempt, and Apache (bless it) gives a 414 "buzz off".

But my question is what is "SEARCH"? I can't find any reference as an HTTP method. Is it an IIS thing?

elfin
byUsers Moderator
byUsers Moderation Team
Elite byUser

Gender:
Posts: 695

Karma: 21

Re: http SEARCH method?
« **Reply #1 on:** 12/04/2004 at 20:48:35 »

SEARCH
Proposed only. The index (etc) identified by the URL is to be searched for something matching in some sense the enclosed message. How does the client know what message fromats are acceptable to the server? (Suggestion of Fred Williams)
more info (though this might not be it)

http://radinfo.musc.edu/~eugenem/blog/archives/000365.php
I think this page mention's something similar, and there are a few links from it.

HTH

Jellyroll
byUsers Administrator

Posts: 7314

Re: http SEARCH method?
« **Reply #2 on:** 12/04/2004 at 21:20:48 »

It's an attempt to trigger a Buffer Overflow in some M$ WebDav thingy by the "Goabot" or "Agobot" worm. MS03-007 describes the vulnerability, but I'm too tired to dignify it with a link.

A search google search on the words WebDAV and Exploit turned up a good tutorial on a previous very similar shellcode exploit which is available at: http://home.comcast.net/~merana296463/files/fatelabs-ntdll-analysis.pdf

Since the binary data is not the same, this is not the exact exploit captured here, but it demonstrates the techniques involved and shows the resulting logs and results of a successful attack.

Bare Byte Unicode Encoding

This detect was the original one selected for this analysis. In reality it is just part of the stream of data being sent to the web server. As Snort is currently stateless, the HTTP analysis is currently only performed on a per packet basis. This is explained in the README.http_inspect file: as extracted below

> This initial version of HttpInspect only handles stateless processing. This means that HttpInspect looks for HTTP fields on a packet by packet basis, and will be fooled if packets are not reassembled. This works fine when there is another module handling the reassembly, but there are limitations in analyzing the protocol. That's why future versions will have a stateful processing mode which will hook into various reassembly modules.

On its own this is an HTTP packet that does not conform to the http standard in that the data field does not begin with a HTTP method, and instead begins with Unicode bytes. The reason that this packet does not flag as a SHELLCODE x86 NOOP is due to the fact that the NOOP hex values do not occur until late in the data field. The current Snort rule #648 only searches for a string of NOOPs until a depth of 128 bytes. The NOOPs don't start until well after.

SHELLCODE x86 NOOP

This is the meat of the attack which is effectively to pad a large number of NOOPs into the code space of the program and hope that an instruction has a jump instruction into this active code space. We do not see the active code uploading to the server because (un)fortunately the Apache web server intervenes prior to the arrival of the interesting code. Apache is not vulnerable to the IIS vulnerability and provides the following error message back to the offending host. ***HTTP/1.1 414 Request-URI Too Large<CR><LF>***

Note: Unless you turn on word wrapping in WordPad, you may not see the happy end to that message (the 414 reply) as WordPad seems to have difficulty with displaying these long lines properly.

Analysis Common to All Detects

All of the detects from a single IP address came from the same http session, but that does not prove whether they are part of the same event or whether they are independent events. My next step was to open the Apache log file and confirm the outcome of the attack if possible. I was quite interested to know whether I could prove to myself that the attacks were unsuccessful, or whether I would be rebuilding the system from scratch. The times in the log files appear different until you note that the Apache log is UTC +2 and the system log is UTC +1 (CET). The following log entry shows that the collection of packets have been reassembled into a single malicious HTTP Search request of over 6 Kbytes in length. Over 27 Kbytes of NOOPs and other related binary information were thrown at the server, but the server generated a response and ended the session well before they had all been delivered.

This entry was found in the Apache 2.0 access.log

```
211.45.217.3 - - [22/Apr/2004:08:14:10 +0200] "SEARCH
/\x90\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x
02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\
x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1
.
.
.
\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x9
0\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x
90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\
x90\x90\x90\x90\x90HTTP/1.0" 414 366
```

The end result of this all: Exploit is the WebDAV Exploit a.k.a. Windows NTdll.dll Buffer overflow as documented in CAN 2003-0109.

Evidence of Packet Crafting

The rate and types of packets delivered to the web server makes it probable that some form of script or worm was used to generate and deliver the packets. Of interest here is that while the source application properly initiated the three way handshake, it did not make use of the push flag, and tended to ignore both the FIN and multiple Reset flags that were sent back to the attacker by the web server. Ignoring the reset flag could either be accomplished by the code, or by filtering at the attacker's firewall (as a hacker defense against active response mechanisms in an IDS.)

Attack mechanism:

This attack is a classic, buffer overflow. The attack attempts to exploit a buffer overflow in the WebDAV Search method by pumping a large number of NOOPs at the web server followed by a block of code which will nominally create a root shell for the attacker to log into. As the attacker has no knowledge of exactly where in the memory space the code will be downloaded, he effectively needs to have a range of memory addresses affected which will lead to his code. A large

number of NOOPs achieves this goal as they will all execute in sequence leading to the attack code. The random chance is where the program will branch to, and whether this will just lead to a bad crash, or to the execution of the attack code. There are lots of books written on the subject, but the document "Analysis of the ntdll.dll WebDAV Exploit, Fate Research Labs Internet Warfare and Intelligence Team http://www.fatelabs.com, Date: Tue. March 25, 2003" provides a concise walk through of the events including logs from successful attacks. While this is not the exact attack shown here (the binary code is different in this attack,) the information is still quite relevant. Likely this is a variant on the original script.

Correlations:

The WebDAV exploit was first announced in CAN 2003-0109 in Feburary 2003 under the title of "Windows NTdll.dll Buffer overflow".
Other references of user are:
http://www.whitehats.com/info/IDS181
http://www.snort.org/snort-db/sid.html?sid=648
http://home.comcast.net/~merana296463/files/fatelabs-ntdll-analysis.pdf
http://www.securityfocus.com/bid/7116
http://www.microsoft.com/technet/security/bulletin/MS03-007.mspx
http://www.whitehats.com/info/ids181

README.http_inspect from the Snort 2.1.12 documents

The SecurityFocus web site appeared to be a little too helpful as it provided downloads for 10 different attack scripts to implement this exploit. While having the code from these exploits is useful in decoding new exploits, it is also helps the script kiddies to be hackers.

There is a short thread on the Snort-Sigs mailing list (2004-04-01 21:05) started by Tyler Hudak with another variant on this attack (where the initial packet is less than 300 bytes (or the relevant preprocessor is disabled). In his case, the "MISC WebDAV search access" rule triggers. As the second detect he describes is a x86 ShellCode NOOP, I suspect that the data again differs from my attack, but that the end goal is the same.

Finally an excellent in depth analysis is provided by Brandon Young GCIH at http://www.giac.org/practical/GCIH/Brandon_Young_GCIH.pdf

Evidence of active targeting:

In order to be successful previous reconnaissance should be done in order to find active IIS servers which might be vulnerable. In this case, a response on Port 80 was seen to be good enough to launch the script even though simply opening the default home page would have hinted to the attacker that he was barking up the wrong tree… Well when all you have is a hammer, all your problems look like Microsoft IIS.

Severity

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

The attacker guessed the right code platform and operating system, but guessed the wrong web server. As the web server information was

Criticality **3**: This attack was launched against a Windows XP workstation configured as a Web server. As this is one of the main systems I am currently relying on for Snort analysis to complete this assignment, this represents a very important asset. As it is not the central web server at the office, it does not rate a higher score

Lethality **1**: As the attack was doomed to failure the moment I chose Apache, this could not be a very lethal attack.

System countermeasures 2: In order to entice enough packets to perform analysis, I disabled the personal firewall on the system as a calculated risk. However, the system is fully patched for the operating system, applications and antivirus.
Network countermeasures 2: The DSL modem does not have a built in firewall. Each system (other than the IDS system with the web server) has a personal firewall configured to block all incoming connections. There is an active IDS system on the network which is currently being reviewed almost daily (looking for the perfect detect for analysis #3)

Severity = (3 + 1) - (2 + 2) = 0

Overall this attack would rates as a **0** mainly due to the fact that the attacker did not have the knowledge to correctly profile the intended victim.

Defensive recommendation

The temporary disabling of the personal firewall has already been corrected and is unlikely to be repeated. Current plans to improve security include the installation of a dedicated firewall with IDS built in. As this network currently has no requirement for published services to the Internet, the firewall will have all incoming connections blocked. The personal firewalls will remain in service, but file sharing between internal hosts may be allowed after the dedicated firewall is installed.

Multiple choice test question:

```
alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"SHELLCODE x86 NOOP";
content: "|90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; depth: 128;
reference:arachnids,181; classtype:shellcode-detect; sid:648; rev:4;)
```

Based on the above rule, would the following packet be logged as a SHELLCODE x86 NOOP exploit?

A) Yes
B) No, the IP data field must start with 90 90 90 90 90 90 …
C) Insufficient information is given to determine the answer
D) No, the detected content is too far from the beginning of the packet

```
08:14:09.456949 PPPoE  [ses 0x9616] IP 1482: IP 211.45.217.3.54291 >
XXX.XXX.XXX.209.80: . 1441:2881(1440) ack 1 win 65535 (DF)
0x0000   1100 9616 05ca 0021 4500 05c8 1598 4000        .......!E.....@.
0x0010   6b06 540d d32d d903 0000 00d1 d413 0050        k.T..-.........P
0x0020   6eea 5655 bf8b fb3f 5010 ffff 7d98 0000        n.VU...?P...}...
0x0030   b102 b102 b102 b102 b102 b102 b102 b102        ................
.
.        (deleted section all b102)
.
0x02f0   b102 b102 b102 b102 b102 b102 b102 b190        ................
0x0300   9090 9090 9090 9090 9090 9090 9090 9090        ................
.
.        (deleted section all 9090)
.
0x05c0   9090 9090 9090 9090 9090 9090 9090 9090        ................
```

Answer: **D**

The sequence that this rule triggers on does not start until offset 0x2ff which is well after the rule depth of 128 bytes

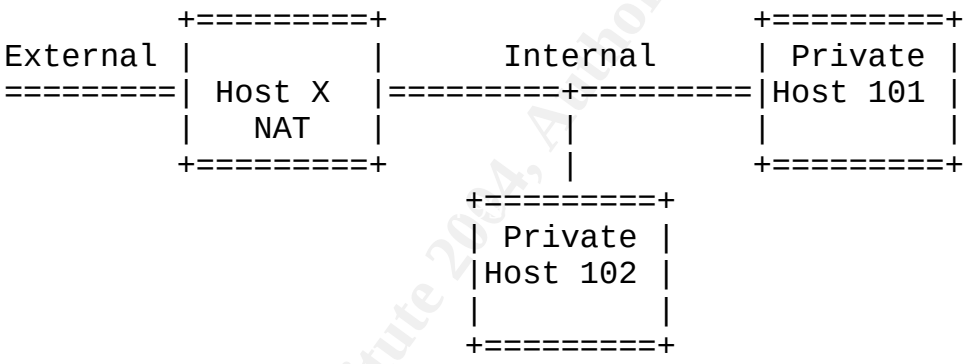
Source of Trace #3:

Location of trace

This detect was extracted from the tcpdump file named 2002.8.22 downloaded from downloaded from http://www.incidents.org/logs/Raw/. The timestamps on these packets indicate that they were actually captured on 2003.08.22 between 14:34 and 15:36

Network Layout

Based on the captured packets, the network configuration consists of 2 computers with private IP addresses (192.168.2.101 and 102) connected to the INTERNET via a third system (host X) which is performing NAT translations for host 102. Insufficient information is available to determine the IP address of NAT host X, but this host appears to be programmed as the default gateway for host 102. Only three MAC addresses appear on this network. Packet capture was made inside the network (remote possibility that it was on host 102 as Windump is installed on that system, but also possible as a receive only lan analyzer.)



Detect was generated by:

Snort intrusion detection system version 2.1.2 for Win32.

Rule triggered

```
[**] [1:2351:1] NETBIOS DCERPC ISystemActivator path overflow attempt
little endian [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
08/22-14:34:23.861089 0:D0:59:2B:7A:57 -> 0:50:DA:C5:9D:8B type:0x800
len:0x5EA
192.168.2.101:32777 -> 192.168.2.102:135 TCP TTL:64 TOS:0x0 ID:15427
IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xBBCCB264  Ack: 0x3704D4B  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 167847 11838
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0352]
```

Explanation of Trigger Stimulus

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 135 (msg:"NETBIOS DCERPC
ISystemActivator path overflow attempt little endian";
flow:to_server,established; content:"|05|"; distance:0; within:1;
byte_test:1,&,16,3,relative; content:"|5c 00 5c 00|";
byte_test:4,>,256,-8,little,relative; reference:cve,CAN-2003-0352;
classtype:attempted-admin; sid:2351; rev:1;)
```

This decodes to Write an alert when a TCP packet from an External network on any port sends a packet to a host on the internal network port 135 and the following additional conditions are met:

- *Flow:to_server,established*; (the stream is flowing to the server and a TCP session has been established
- *Content:"|05|;distance:0;within:1*;(At byte offset 0 in the TCP Data the hex data "05" appears.
- *byte_test:1, &,16,3,relative*; 3 bytes after the "05" compare the next byte against 16 (0x10)
- *content:"|5c 00 5c 00|"*; search for hex data "5C 00 5C 00" found at offset "0x56C" within this packet
- *byte_test:4,>,256,-8,little,relative;* 8 bytes before the end of the previous content match compare the next 4 bytes (i.e. the 4 bytes immediately preceeding the last content match to ensure that they are greater than 256 (when those 4 bytes interpreted as least significant byte first)

Probability the source address was spoofed:

Somewhat less than none. The attacking host is internal to the network and on the same segment as the packet logger. The attack is based on an established TCP session between attacker and victim with the intent of establishing a RootShell.

Source of Attack

Internal computer at IP address 192.168.2.101 (MAC Address 00:D0:59:2B:7A:57)

Description of attack:

The attacking computer utilized the kaht2.exe exploit code to send a buffer overflow to the DCOM RPC service on a Windows system. The linkage to the kaht2 code is based on a very strong match between the captured code from the log file, and the shell code extracted from the kaht2 program. The kaht2 program was downloaded from http://www.securityfocus.com/bid/8205/exploit/.
Only 2 bytes of the code differ between the two sets of binary code. This difference is possibly related to the listening port (53) for the root shell. Immediately after the delivery of this payload the attacking system opened a connection to the exploited system (root shell) on port 53 where he was able to browse directories, execute applications that were on the system (Windump), and view raw http packets (and replies) being transmitted between that workstation and the Internet.

Attack mechanism:

This is a classic buffer overflow overflow as implemented by the Blaster, MSBlast, LoveSan, scripts and the Nachi, and Welchia worms (reference to ???)

The destination is port 135 and the DCOM RPC service where the script or worm overflows a buffer to insert a shell code. In this case the shellcode immediately started listening for connections on port 53 providing the attacker with full access on the system.

Correlations:

This eploit was first published in July 2003
http://www.microsoft.com/technet/security/Bulletin/MS03-026.mspx
http://www.microsoft.com/technet/security/bulletin/MS03-039.mspx
http://www.securityfocus.com/bid/8205
Snort\rules\netbios.rules
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0352
http://www.cert.org/advisories/CA-2003-16.html
http://www.cert.org/advisories/CA-2003-19.html
The exploit is readily available in either script or worm varieties and has been categorized under several names and variants.

Evidence of active targeting:

This attack is interesting in that it is an insider attack. In this case the attacker is able to monitor the activities of the user of host 102 without significant risk of detection.

Severity

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

Criticality 2: Host 102 appears to be a windows 2000 Professional system (build 2195) with peer web services installed (based on the directory structure browsed by the attacker which included a inetpub directory in C:\ The owner of the system may be named Jamie (another directory in C:\) The system has at least some administrative tools (Windump) installed. However, this configuration does not seem to be a critical installation based on the network layout.

Lethality 5: The attack was completely successful with no apparent error messages being logged by the system (the Port 135 session was terminated gracefully without error after the shell code was delivered.) The attack makes no changes to the filesystem nor changes to the registry making its detection difficult unless the attacker executes further less subtle attacks.

System countermeasures 1: While the exploits had only received less than 2 months of publicity, the publicity available was quite high profile. As such, the system should have been patched by late August when the attack occurred.

Network countermeasures 2: The network connection is utilizing NAT which provides a good start towards security. Specifically, packets addressed to any ports which are not explicitly mapped through the NAT box to a particular host on the inside network have no place to go.

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

This is an attack which requires immediate action to ensure that the system is not used as an attack platform against other systems, and to protect the privacy of host 102's personal information. Defensive recommendation.

Defensive recommendation:

As this is an insider attack, special attention must be directed at properly handling the incident. Management and legal advise must be sought to ensure that the appropriate civil and / or criminal legal actions are taken. Initial steps to preserve and isolate both systems needs to be taken, as well as isolating the systems from the INTERNET (as we do not know yet what else is lurking in the two systems. Once the legal mire has been sorted, the system needs to be rebuilt (after preserving and quarantining data) including all of the relevant patches. A good review on the recovery of compromised systems can be found at

Multiple choice test question:

Based on the following rule from Snort 2.*

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 135 (msg:"NETBIOS DCERPC
ISystemActivator path overflow attempt little endian";
flow:to_server,established; content:"|05|"; distance:0; within:1;
byte_test:1,&,16,3,relative; content:"|5c 00 5c 00|";
byte_test:4,>,256,-8,little,relative; reference:cve,CAN-2003-0352;
classtype:attempted-admin; sid:2351; rev:1;)
```

Which of the following stimulus will trigger the above rule (assuming all other criteria are met)

A) The string "5c 00 5c 00" within the TCP data of the packet
B) The Hex data "0x5c005c00" within the UDP data of the packet
C) The string "attempted-admin" within the TCP data of the packet
D) The Hex data "0x05" in the first byte of the TCP data of the packet

Answer **D**:

Not A (looking for hex not string)
Not B (looking for TCP packets)
Not C (Attempted-Admin is the class type for the alert and does not appear in the packet)

Security Audit results for
University of Maryland, Baltimore County
Review period 07 April 2004 to 11 April 2004
Prepared by Blaine Hein
21 May 2004

Executive Summary:
This audit reviewed the security posture of the University of Maryland, Baltimore County computer networks for the period of 7 through 11 April 2004.

# of Alert Events Processed:	1,713,320
# of Scans Processed	15,666,221
# of Out of Spec Packets Processed	12,833

Compromised Systems: **139**

The main source of compromised systems relates to the continued presence of the Red Worm (Adore), Mstream, and NIMDA malware within the network and the continued vulnerability to the Shellcode x86 NOOP exploit.

The following changes are recommended:

Policy:
- Update the appropriate use policy http://www.umbc.edu/oit/security/policy/2-UMBC/IT-01-final.html
- Review the network information published on the INTERNET. Some of the information should not be published to INTERNET servers

Procedures and Implementation:
- Update the security information and tools web pages located at http://www.umbc.edu/oit/security/
  - add pointers to MS Update, or add a System Update Server (SUS) to the network.
- Upgrade the IDS sensors and move to storing alert / scan / OOS data in databases
- Review the border security policy with a goal of creating a list of ports that should NEVER pass in or out of the network.
- Create a list of servers which are dedicated only to internal activities, and block those ip addresses from passing through the firewall

Files analysed

Alerts		Scans		OOS		Comments
Name	**Size (Bytes)**	**Name**	**Size (Bytes)**	**Name**	**Size(Bytes)**	
alert.040407	16,954,984	scans.040407	250,592,571	oos_report_040403.txt	393,216	Data 04/07
alert.040408	41,540,552	scans.040408	77,902,222	oos_report_040404.txt	1,981,440	Data 04/08
alert.040409	45,943,614	scans.040409	173,015,040	oos_report_040405.txt	2,032,640	Data 04/09
alert.040410	56,463,241	scans.040410	321,861,992	oos_report_040406.txt	532,480	Data 04/10
alert.040411	55,627,652	scans.040411	217,055,232	oos_report_040407.txt	3,456,000	Data 04/11
				oos_report_040408.txt	1,341,440	Data 04/12
				oos_report_040409.txt	516,096	Data 04/13
				oos_report_040410.txt	1,638,400	Data 04/14
				oos_report_040411.txt	360,448	Data 04/15
TOTAL	216,530,043		1,040,427,057		12,252,160	**1,269,209,260**

Additional OOS files were included due to errors in file dates in the logs to ensure that the entire date range was covered. The actual dates of the data within the OOS reports are shown in the comments section.

The following is a list of detects from the alert log, prioritized by number (top to bottom) and severity (with color coding). The legend for the color coding is included at the bottom of the table. Following the table, the alerts with a high probability of compromise (RED) and some of the medium level attacks (yellow) are described including a list of probable and possible compromised hosts for each.

Alert Log	Quantity
spp_portscan: portscan status from	1300429
spp_portscan: End of portscan from	157977
spp_portscan: PORTSCAN DETECTED (STEALTH)	91423
spp_portscan: PORTSCAN DETECTED (seconds)	69954
EXPLOIT x86 NOOP	28822
130.85.30.3 activity	12994
SMB Name Wildcard	12170
High port 65535 tcp - possible Red Worm - traffic	10664
130.85.30.4 activity	10207
Tiny Fragments - Possible Hostile Activity	8005
DDOS mstream handler to client	3253
Null scan!	1125
NMAP TCP ping!	1098
Possible trojan server activity	1081
External RPC call	930
SUNRPC highport access!	637
Incomplete Packet Fragments Discarded	511
TCP SRC and DST outside network	309
High port 65535 udp - possible Red Worm - traffic	244
ICMP SRC and DST outside network	210
[UMBC NIDS] Internal MiMail alert	158
[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.	147
DDOS shaft client to handler	142
[UMBC NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC	108
FTP passwd attempt	100
TCP SMTP Source Port traffic	83
IRC evil - running XDCC	72
EXPLOIT x86 setuid 0	66
SMB C access	55
[UMBC NIDS] External MiMail alert	47
connect to 515 from outside	46
EXPLOIT x86 setgid 0	33
EXPLOIT x86 stealth noop	28
[UMBC NIDS IRC Alert] Possible drone command detected.	25
RFB - Possible WinVNC - 010708-1	24
FTP DoS ftpd globbing	22
[UMBC NIDS IRC Alert] Possible Incoming XDCC Send Request Detected.	17

NIMDA - Attempt to execute cmd from campus host	15
Attempted Sun RPC high port access	14
TFTP - Internal UDP connection to external tftp server	14
SYN-FIN scan!	13
EXPLOIT NTPDX buffer overflow	10
EXPLOIT x86 NOPS	8
DDOS mstream client to handler	6
Probable NMAP fingerprint attempt	6
TFTP - External TCP connection to internal tftp server	4
External FTP to HelpDesk	3
NETBIOS NT NULL session	3
PHF attempt	2
[UMBC NIDS IRC Alert] User joining XDCC channel detected. Possible XDCC bot	2
[UMBC NIDS IRC Alert] K\:line'd user detected, possible trojan.	2
Fragmentation Overflow Attack	1
[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC	1
Total Number of Events in Alert Log	**1713320**

Legend:
Probable compromise or DDOS related Attacks
Reconnaissance (includes internal compromised systems)
Policy Violations

DDOS mstream handler to client : DDOS mstream client to handler
These two alerts indicate that there is activity in creating a DDOS attack through the recruitment of handlers and agents. DDOS utilizes the client →handler →agent → victim structure to achieve the control and amplification factor for the attack. As there are currently no communications with mstream agents, we are not yet faced with the imminent launch of the attack. Based on the results of a few SQL queries against the alert log database we can see that there are external DDOS mstream clients communicating with internal mstream handlers with bidirectional communications occurring during an 8 hour window near the end of the audit period. The system highlighted in green is further detailed in a link graph later in this report. 04/10-22:45:00.635453 [**] DDOS mstream handler to client [**] 130.85.84.235:12754 -> 82.48.242.184:4662 04/10-23:33:19.441631 [**] DDOS mstream client to handler [**] 62.42.66.52:4662 -> 130.85.84.235:12754

04/10-23:33:19.447750 [**] DDOS mstream handler to client [**]
130.85.84.235:12754 -> 62.42.66.52:4662
04/10-23:33:21.370007 [**] DDOS mstream handler to client [**]
130.85.84.235:12754 -> 62.42.66.52:4662
04/10-23:33:23.376086 [**] DDOS mstream client to handler [**]
62.42.66.52:4662 -> 130.85.84.235:12754
04/10-23:33:25.365360 [**] DDOS mstream client to handler [**]
62.42.66.52:4662 -> 130.85.84.235:12754
04/10-23:33:25.369519 [**] DDOS mstream handler to client [**]
130.85.84.235:12754 -> 62.42.66.52:4662
04/11-02:26:09.897585 [**] DDOS mstream handler to client [**]
130.85.84.235:15104 -> 217.236.97.47:4662
04/11-05:40:38.174781 [**] DDOS mstream client to handler [**]
213.180.193.68:45101 -> 130.85.60.38:15104
04/11-06:00:02.340196 [**] DDOS mstream handler to client [**]
130.85.84.235:15104 -> 81.102.85.92:4662
04/11-06:00:02.436560 [**] DDOS mstream handler to client [**]
130.85.84.235:15104 -> 81.102.85.92:4662

Compromised systems:

Handlers:	External Clients:
130.85.84.23	62.42.66.52
130.85.60.17	213.231.96.32
130.85.97.28	212.195.102.30
130.85.1.4	213.180.193.68
130.85.110.7	

References:
Snort SID 247, 248, 249, 250
CAN-2000-0138

In mid 2003 Mario Ricci reported 71 alerts across 3 hosts from this DDOS attack. (http://www.giac.org/practical/GCIA/Mario_Ricci_GCIA.pdf) During this audit, 3259 total alerts were recorded, but only 5 internal hosts were seen to be compromised and the majority of alerts were generated by a handler attempting to communicate with a client which was no longer responding.

DDOS shaft client to handler
Similar to the Adore DDOS, the Shaft DDOS utilizes the client →handler →agent → victim structure to achieve the control and amplification factor for the attack. The Snort description for this detect indicates that "It is also possible that this event may be generated when any host attempts to discover or detect a Shaft handler." As there are no responses to the client from the handler, it is probable that this is a false positive. However it is likely that this system was previously compromised by the Shaft DDOS and that the clients are looking for a handler that no longer exists on that system.
Relevant systems: Targets Unix systems Compromised systems: None. However, the targeted host 130.85.84.235 is already on the "to be rebuilt" list due to other compromises.
Snort: SID 230 Arachnids: 254 http://security.royans.net/info/posts/bugtraq_ddos3.shtml http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0138

NIMDA - Attempt to execute cmd from campus host	
The NIMDA virus is capable of propagating in at least five unique methods (File, email, web worm, LAN, and netbios). The alert referenced here appears to be a customized rule as it does not match with the message text from current Snort signatures. The alerts logged are all packets sent from the 130.85 subnet (hence the reference to campus host) and they are all destined to port 80 of hosts on external subnets (therefore the web worm mode of propagation.) When using the web worm mode of operation, the standard snort rules will capture the relevant vulnerability being targeted instead of specifically the NIMDA virus. The vulnerabilities exploited by the nimda worm are listed in IN-2001-09, CA-2001-11, CA-2001-12, US-CERT VU#111677, MS00-078, and MS00-057.	
Relevant systems: Targets Windows 95, 98, ME, 2000 Probable compromised systems	
130.85.97.36 130.85.10.79 130.85.97.228 130.85.97.166 130.85.97.69	130.85.97.180 130.85.97.74 130.85.97.25 130.85.17.45
References: www.f-secure.com/v-descs/nimda.shtml http://www.cert.org/advisories/CA-2001-26.html http://www.cert.org/incident_notes/IN-2001-09.html http://www.cert.org/advisories/CA-2001-11.html http://www.kb.cert.org/vuls/id/111677 http://www.cert.org/advisories/CA-2001-12.html http://www.microsoft.com/technet/security/bulletin/MS00-078.asp http://www.microsoft.com/technet/security/bulletin/ms00-057.asp	

High port 65535 tcp (and udp) - possible Red Worm - traffic

Red worm is more commonly known as the Adore worm. The worm targets Linux systems. Infection is accomplished through exploiting one of several vulnerabilities including LPRng, rpc-statd, wu-ftpd and BIND on unpatched systems. Once infection is achieved, the worm downloads a tar ball to install the remaining executables. The worm replaces 'ps' with a trojaned version, and also replaces klogd (kernel message logger) with a trojan backdoor version. After attempting to email sensitive information to four addresses, the worm creates a cron job (scheduled execution of an application or script), cleans up and reboots the system, leaving the backdoor in place. Indications of infection include the presence of the directory /usr/bin/adore, and suspicious activity on port 65535 TCP or UDP. The site from which the tar ball is downloaded still exists (details below) but as this is an Asian language site it is difficult to determine whether the site still serves the files. Recommended recovery includes imaging the infected systems and then rebuilding them from scratch. Other worms which use this port include RC1 and SINS.

Nslookup go.163.com Non-authoritative answer:
Name: sms.163.com
Addresses: 202.108.34.26, 202.108.34.35, 202.108.36.247, 202.108.42.133
202.108.42.134, 202.108.42.138, 202.108.42.139, 202.108.248.23, 202.108.248.24
Aliases: go.163.com

Affected systems: Linux

130.85.60.16	130.85.25.68	130.85.24.33	130.85.12.7
130.85.84.235	130.85.25.73	130.85.97.41	130.85.97.11
130.85.97.51	130.85.153.35	130.85.98.87	130.85.25.11
130.85.97.92	130.85.24.34	130.85.60.17	130.85.82.8
130.85.153.8	130.85.6.7	130.85.150.253	130.85.97.70
130.85.97.217	130.85.24.20	130.85.60.14	130.85.97.94
130.85.97.104	130.85.70.225	130.85.1.3	130.85.70.197
130.85.97.106	130.85.12.6	130.85.71.248	130.85.5.100
130.85.97.196	130.85.34.5	130.85.34.11	130.85.82.43
130.85.97.182	130.85.24.44	130.85.6.62	130.85.153.94
130.85.25.69	130.85.25.66	130.85.152.21	130.85.24.74
130.85.34.14	130.85.111.34	130.85.97.59	130.85.97.88
130.85.25.67	130.85.97.87	130.85.97.175	
130.85.25.71	130.85.82.79	130.85.25.10	
130.85.25.70	130.85.97.124	130.85.12.4	

References:
http://securityresponse.symantec.com/avcenter/venc/data/linux.adore.worm.html
http://uk.mcafee.com/virusInfo/default.asp?id=description&virus_k=99064
LPRng: http://www.cert.org/advisories/CA-2000-22.html
wu-ftpd 2.6: http://www.cert.org/advisories/CA-2000-13.html
Bind: http://www.cert.org/advisories/CA-2001-02.html
rpc.statd: http://www.cert.org/advisories/CA-2000-17.html

[UMBC NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC			
Sdbot is an IRC Trojan which implements a backdoor to the system. The Trojan is launched by sending the program to a vulnerable irc client via a DCC send command. When the Trojan is launched it connects back to a predetermined chat room where the sender is then able to execute commands including floodnet (DOS). The rule appears to be based on the destination port 7000. The following systems should be checked for irc installations and the sdbot trojan			
130.85.112.152 130.85.112.163 130.85.150.199 130.85.151.75 130.85.153.174	130.85.153.195 130.85.42.2 130.85.43.10 130.85.66.56 130.85.70.96	130.85.80.224 130.85.80.28 130.85.80.5 130.85.84.235 130.85.97.44	130.85.97.66 130.85.97.95
http://www.norman.com/News/eNews/2003/11309/en-us			

[UMBC NIDS] Internal (and External) MiMail alert			
This worm, which is also known as the w32/MyDoom Worm exploits two windows based vulnerabilities including the cookie based script exploit as described in MS02-015, and exploit against Mime Encapsulated Aggregate Html. Both exploits involve the execution of scripts cached on the local hard drive in the context of the local computer zone instead of the Internet zone where they arrived from. The result is a combination of Mass Mailing from the victim host and peer to peer file sharing. Of interesting note is that both the a and b variants of MyDoom should have become inactive as of February 12th and March 1st 2004 respectively. Obviously they have been modified… There were 3 internal systems communicating outbound and one external system communicating inbound			
130.85.110.82	130.85.97.94	130.85.97.135	130.85.12.6 (from External)
http://securityresponse.symantec.com/avcenter/venc/data/w32.mimail.a@mm.html http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS02-015.asp http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS03-014.asp http://www.axial.co.uk/niksun/W32MyDoom%20Worm%20Detection.pdf			

EXPLOIT x86 NOOP
This is a classic shellcode buffer overflow attack which can be quickly adapted to any available software vulnerability with only minor difficulty. Over the last year this exploit has been used against the Microsoft IIS WebDAV (ntdll.dll) vulnerability. The result of a successful exploit is normally a root shell access to the system. This is normally just the start of trouble as the first step performed is to load other malware onto the system to achieve greater goals such as DDoS or portscanning to find other vulnerable systems.
http://www.whitehats.com/info/IDS181 http://www.snort.org/snort-db/sid.html?sid=648 http://home.comcast.net/~merana296463/files/fatelabs-ntdll-analysis.pdf http://www.securityfocus.com/bid/7116 http://www.microsoft.com/technet/security/bulletin/MS03-007.mspx

http://www.whitehats.com/info/ids181 http://www.fatelabs.com

Possible trojan server activity			
This is another customized rule which triggers on activity with a source or destination port value of 27374. This port is currently associated with a long list of Trojans including (Bad Blood, Ttfloader, The Saint, SubSeven Muie, SubSeven, Seeker, Ramen, Lion, Fake SubSeven, EGO, Webhead, Muerte) (http://grc.com/port_27374.htm). Much of the traffic captured on this rule appears to be scanning activity looking for previously compromised systems and most of the internal systems appear to drop this traffic. There are, however some notable exceptions where the hosts have responded to the traffic and continued with a TCP session. These systems will be cross referenced with the remaining Alerts, Scans, and OOS files to determine whether they are compromised. Systems which responded to the scan and are also logged as sources to other alerts will be categorized as probable compromise. Systems which answered the scan, but are not linked as the source for other alerts will be categorized as possible compromise.			
Relevant systems: This is likely to affect all platform types to some degree.			
Probable compromise: (Two way communications)			
130.85.12.4	130.85.190.203	130.85.24.44	130.85.60.17
130.85.12.6	130.85.190.93	130.85.24.74	130.85.84.235 See Link Graph
130.85.190.1	130.85.190.95	130.85.34.11	
130.85.190.102	130.85.190.97	130.85.6.15	130.85.97.87
130.85.190.202	130.85.24.34	130.85.6.7	
Possible compromise: (No response to the external system logged)			
130.85.16.90	130.85.55.27	130.85.69.22	130.85.97.92
References: http://grc.com/port_27374.htm			

SUNRPC highport access!
The rpcbind application on vulnerable solaris systems listens on a UDP port > 32770 in addition to the standard port 111 tcp/udp. This creates the potential to bypass firewall rules which may block access to port 111 from external systems. If the connection is successful, access to the rpcbind service will provided the attacker reconnaissance information regarding other rpc applications that are running on the system and how to access them. In this case, while there are packets delivered to 24 different systems, there are no return packets logged by the IDS. As we do not have a copy of this particular rule, we can not confirm whether the packets were dropped, or whether the replies were not captured by the IDS ruleset.
Possible affected systems: Older versions of SUN Solaris (pre 2.6) The following systems may have been subject to detailed reconnaissance of the RPC applications running on them:

130.85.82.106 130.85.70.37 130.85.70.154 130.85.97.13 130.85.25.66 130.85.97.44	130.85.97.235 130.85.97.61 130.85.60.11 130.85.97.168 130.85.97.22 130.85.97.223	130.85.24.70 130.85.97.20 130.85.100.203 130.85.97.55 130.85.97.213 130.85.97.15	130.85.34.14 130.85.97.144 130.85.34.5 130.85.97.172 130.85.97.46 130.85.60.38
IDS429/RPC_PORTMAP-LISTING-32771 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0632 http://advice.networkice.com/Advice/Intrusions/2001729/default.htm http://www.securityfocus.com/bid/205/discussion/			

[UMBC NIDS IRC Alert] K\:line'd user detected, possible trojan.	
This internal rule appears to watch for a specific host name, domain name, or geographic region which matches an entry in an irc server configuration file. Most likely this is accomplished by matching on the error message sent back to the k-lined user from the irc server. Where the k-line list is being used to block systems with known Trojans or worms this can be an indication of a Trojan, or a worm attempting to access an irc server. This may also be an indication that users (or the domain to which they belong) have been blocked due to inappropriate behavior. **High possibility for false positives**	
Relevant systems: all platforms	
130.85.84.203	130.85.97.158
http://encyclopedia.thefreedictionary.com/K-line http://www.ircbeginner.com/ircinfo/h-klines.html	

[UMBC NIDS IRC Alert] User joining XDCC channel detected. Possible XDCC bot
The use of IRC is effectively a nightmare to system administrators due to its ability to subvert the network security policy. It is the same features which make it so popular with the user community. New tools such as XDCC extend this problem by making it easier and more automated for users to find information to download. These tools automate much of the process and do not require user participation (hence the name bot). The differentiation between a bot and a trojan in this case is very fine
Relevant systems: all
http://www.cs.rochester.edu/~bukys/host/tonikgin/EduHacking.html

External FTP to HelpDesk
Internal rule to track dangerous activity towards a high value system. The helpdesk contains significant information on the status of the network. In addition, the helpdesk may also contain software applications of high value (or network tools that you may not wish to have used against you from outside your own network…) In this case, there is no traffic logged in the reverse direction. As we do not have the UMBC rulesets these systems should be checked. If FTP is enabled on these systems, the server logs should be checked to see what was uploaded or downloaded.
Possible affected systems:

130.85.70.50 130.85.70.49 130.85.53.29

130.85.30.3 activity and 130.85.30.4 activity
All traffic to these two systems is currently being monitored looking for signs that these two systems are This is common practice after rebuilding a compromised machine. The first system may be a Novell Ser large number of packets being sent to port 524. (Novell Connecting Point). The second system has a lar packets sent to ports 51443 and 80 (90% of the total) leading me to believe that this is a web server (Port as an example port for Novell HTTP servers in their documentation.)
http://www.novell.com/documentation/ifolder21/index.html?page=/documentation/ifolder21/readme/data
These systems have been monitored for a significant time. Marshall Heilman discusses these two systems (http://www.giac.org/practical/GCIA/Marshall_Heilman_GCIA.doc.pdf) At that point in time, the most indicated in his paper were 524 135 445 80 4000. Now, 524 accounts for 95% of all 13000 alerts, and pc for only 3.5%. The remaining ports each account for less than 1% of alerts. Mr Heilman also discusses t watched system. Once again, the number of actively probed ports has decreased.

Flags from OOS Log

FLAGS	NUMBER	FLAGS	NUMBER	FLAGS	NUMBER
12****S*	12399	12U***S*	2	12*AP***	1
********	223	12UAPRSF	2	12*AP**F	1
****P***	94	****P*SF	1	12*APRS*	1
12***R**	33	****PRSF	1	12U***SF	1
12*A**S*	16	***AP*SF	1	12U**R*F	1
*****RSF	4	**U***SF	1	12U**RS*	1
U***	4	**U**RSF	1	12U*P***	1
*2*A*RSF	3	*2****SF	1	12U*PR**	1
*2U***SF	3	*2***RSF	1	12U*PR*F	1
******SF	2	*2*A**SF	1	12U*PRS*	1
**U*P*SF	2	*2UA*RSF	1	12U*PRSF	1
1***P*SF	2	1****RSF	1	12UA*R**	1
12*****F	2	1**AP*SF	1	12UA*RSF	1
12****SF	2	1*U*PRSF	1	12UAP***	1
12**PRS*	2	1*UA*RSF	1	12UAP*S*	1
12*A*R**	2	12***R*F	1	12UAPR**	1
12*AP*SF	2	12**P*S*	1	12UAPR*F	1
12*APRSF	2	12*A**SF	1	12UAPRS*	1

Flags from SCANS log							
Flags	Qty	Flags	Qty	Flags	Qty	Flags	Qty

******S*	9208956	*2UAPRSF	7	1**AP*SF	4	*2*AP***	2
*******F	58785	******SF	6	1*U*P***	4	*2U*P*SF	2
12****S*	7619	****P**F	6	1*UA***F	4	1******F	2
********	490	****PRS*	6	1*UA*RS*	4	1***P***	2
***A*R*F	477	****PRSF	6	12******	4	1***P**F	2
*2***R**	60	**UAPR*F	6	12***RSF	4	1***PRSF	2
****P***	58	*2****S*	6	12**P***	4	1**A*RSF	2
URSF	51	*2**P*SF	6	12**P*S*	4	1**APR**	2
**U*P*S*	40	*2**PR**	6	12*A**S*	4	1*U*****	2
URS*	36	*2UA***F	6	12*A*R**	4	1*U***S*	2
U***	34	*2UA*RS*	6	12U*P***	4	1*U**RSF	2
1****R**	30	1*******	6	12U*PRS*	4	1*U*P**F	2
12***R**	29	1***PR**	6	12UA**S*	4	1*U*P*S*	2
*2*A**S*	26	1**A****	6	***A*RSF	3	1*U*PR*F	2
1**A*R**	23	1**A**S*	6	**U****F	3	1*U*PRS*	2
****P*S*	22	1*U****F	6	**U***SF	3	1*UAPRS*	2
*2U*P**F	19	1*UA*R**	6	**U**R*F	3	1*UAPRSF	2
12*A**SF	18	1*UA*R*F	6	**UA*RS*	3	12****SF	2
***APRSF	17	*****RS*	5	**UAP*S*	3	12**P*SF	2
*2UA*R**	17	***A**SF	5	*2***R*F	3	12**PR*F	2
*2U**R*F	16	**UA*RSF	5	*2**PR*F	3	12**PRSF	2
*2*A****	14	*2***RSF	5	*2**PRS*	3	12*A*R*F	2
***A*RS*	13	*2**P*S*	5	*2*A*RS*	3	12*AP**F	2
*2U*PR**	13	*2*AP*SF	5	*2*A*RSF	3	12*APR**	2
*2U*PRSF	13	*2*APR**	5	*2*AP**F	3	12*APR*F	2
12UAPRSF	13	*2*APR*F	5	*2U*****	3	12U***S*	2
*2U****F	12	*2U**R**	5	*2U***S*	3	12U**R**	2
U*S*	11	*2U*PR*F	5	*2UAP*S*	3	12U*P*S*	2
*2U***SF	11	*2UA*R*F	5	*2UAPR**	3	12U*P*SF	2
*2UA****	11	1*UA*RSF	5	1*****S*	3	12U*PR*F	2
*2UAP**F	11	1*UAP***	5	1*****SF	3	12U*PRSF	2
1*U***SF	11	1*UAP**F	5	1***P*S*	3	12UA****	2
***APR*F	10	12*****F	5	1**A*RS*	3	12UA***F	2
UASF	10	12***R*F	5	1**AP*S*	3	12UA*R**	2
UA*R	10	12**PRS*	5	1*U**R**	3	12UAP*SF	2
*2*A*R**	10	12*A****	5	1*U*PR**	3	12UAPR*F	2
*2U**RS*	10	12*APRSF	5	1*U*PRSF	3	12UAPRS*	2
*2U*P***	10	12UA*RS*	5	1*UA****	3	*****RSF	1
*2UA**SF	10	12UAP**F	5	1*UA**S*	3	**UAP*SF	1
*2UAP***	10	12UAP*S*	5	1*UA**SF	3	*2U*P*S*	1
****P*SF	9	****PR*F	4	1*UAPR*F	3	*2UAP*SF	1
****PR**	9	**U*P**F	4	12***RS*	3	1****RSF	1
U*P*	9	**U*P*SF	4	12*A*RSF	3	1**A*R*F	1
**U*PRSF	9	**U*PR**	4	12*AP***	3	1**AP***	1
UAPR	9	**U*PRS*	4	12*AP*S*	3	1**APRS*	1
*2*A***F	9	*2******	4	12*AP*SF	3	1**APRSF	1

*2*A**SF	9	*2****SF	4	12U****F	3	1*U**R*F	1
*2*A*R*F	9	*2**P**F	4	12U**R*F	3	1*UAP*S*	1
*2*AP*S*	9	*2*APRS*	4	12U*PR**	3	12**P**F	1
*2U**RSF	9	*2*APRSF	4	12UA**SF	3	12*A***F	1
*2U*PRS*	9	*2UAPR*F	4	12UA*RSF	3	12*A*RS*	1
*2UA**S*	9	*2UAPRS*	4	12UAPR**	3	12*APRS*	1
12U***SF	9	1****R*F	4	***AP*SF	2	12U**RS*	1
*2***** F	8	1****RS*	4	***APRS*	2	12U**RSF	1
*2UA*RSF	8	1***P*SF	4	**U**R**	2	12U*P**F	1
*****R*F	7	1***PR*F	4	**U*PR*F	2	12UA*R*F	1
UAS*	7	1***PRS*	4	**UAPRS*	2	12UAP***	1
**UA*R*F	7	1**A***F	4	*2**P***	2		
*2***RS*	7	1**A**SF	4	*2**PRSF	2		

The flags from the OOS log indicate that the majority of the scans are based on SYN flag with Reservedbits, or the null (no flags set) scan. The information in the scans log shows a very large number of syn packets being sent with FIN scans and Reserved bits with SYN following at a distant second and third places. These three flag combinations account for over 99% of the scans.

Top Talkers based on number of logged events originated

Scans.log		Alerts.log		OOS.log	
Host	Quantity	Host	Quantity	Host	Quantity
130.85.1.3	2886669	130.85.1.3	185463	68.54.84.49	3568
130.85.111.51	1621815	130.85.1.4	151888	202.144.28.167	699
130.85.153.35	1523458	130.85.111.51	145160	202.54.60.162	343
130.85.81.39	1187999	130.85.81.39	110335	66.225.198.20	333
130.85.70.96	1130813	130.85.153.35	101951	130.85.199.20	325
130.85.112.152	1082055	130.85.110.72	48286	80.54.249.132	284
130.85.1.4	795875	130.85.84.235	47465	141.224.64.4	276
130.85.66.56	334882	130.85.25.70	43042	193.170.194.27	216
130.85.84.235	294411	130.85.25.71	42388	80.38.206.68	214
130.85.42.2	253160	130.85.70.96	36925	204.92.130.11	209

The highlighting between the Scans and the Alerts logs indicates that the same hosts are responsible for the majority of the activity. This is a clear indication that things are not going well in the network. Several compromises have already taken place, and the level of hostile activity within the network far outweighs the attacks from outside the network.

OOS Log

Destination Port	Quantity	Common Service
25	4354	SMTP
110	3745	POP3
80	1683	HTTP
4662	1487	
113	543	AUTH / IDENT
28053	214	
24842	174	
443	140	SSL
8080	136	HTTP
22	54	SSH or PC Anywhere
3247	39	DVT Datalink?
3970	24	
6346	22	Gnutella-svc
1214	21	Kazaa
21	17	FTP
143	16	Imap
3964	14	
6881	12	
4167	6	
2787	4	Piccolo Cornerstone Software?
53	4	DNS
3482	4	Vulture
1330	3	StreetPerfect
31678	3	
4665	3	

Alert Log		
Destination Port	**Quantity**	**Common Service**
Null	1628061	Not logged in IDS (scans)
80	29130	HTTP
524	12745	NCP
137	12172	Netbios name service
51443	7255	Novell HTTP Server (common example port)
65535	4965	Adore RC1 Sins trojans
4662	3295	
22	2693	Ssh, PcAnywhere, (shaft)
1025	1772	Blackjack
0	1051	
111	930	RPC
27374	851	Bad Blood, Ttfloader, The Saint, SubSeven Muie, SubSeven, Seeker, Ramen, Lion, Fake

		SubSeven, EGO, Webhead, Muerte
25	799	SMTP
53	654	DNS
32771	651	FileNET RMI
1330	613	StreetPerfect
5000	513	upnp-evnt
135	284	Epmap DCE
2745	216	URBISNET
3645	212	Cyc
110	206	Pop3
6129	154	
20432	142	shaft
21	131	FTP
4672	121	remote file access server

These listings of the most common destination ports have been included for reference while performing the audit.

The following external source addresses were noted as interesting during my analysis. Included is the registration information and a brief explanation for their inclusion on this list.

Ip Address	
62.42.66.52	**Active Mstream Client for internal Mstream handlers**

```
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html
inetnum:      62.42.0.0 - 62.42.127.255
netname:      ONO-SCOPES-4
descr:        Cableuropa - ONO
descr:        ONO net in whole Spain
country:      ES
admin-c:      OIM1-RIPE
tech-c:       OIM1-RIPE
status:       ASSIGNED PA
remarks:      mail spam reports: abuse@ono.com
remarks:      security incidents: security@ono.com
notify:       ripe-tech@ono.es
mnt-by:       ONO-MNT
changed:      ripe-tech@ono.es 20030318
source:       RIPE
route:        62.42.0.0/16
descr:        Cableuropa - Ono
descr:        Ono network in whole Spain
origin:       AS6739
remarks:      mail spam reports: abuse@ono.com
remarks:      security incidents: security@ono.com
mnt-by:       ONO-MNT
changed:      ripe-tech@ono.es 20020619
source:       RIPE
```

```
role:           ONO IP MANAGER
address:         C/ Basauri, 5
address:         Urbanizacion La Florida
address:         E-28023 Aravaca, Madrid
address:         SPAIN
phone:           +34911809300
fax-no:          +34911809245
e-mail:          ripe-tech@ono.es
admin-c:         JMD-RIPE
tech-c:          JMD-RIPE
tech-c:          JABM1-RIPE
tech-c:          MJS6-RIPE
tech-c:          MJC7-RIPE
tech-c:          AGG20-RIPE
tech-c:          FRL9-RIPE
nic-hdl:         OIM1-RIPE
changed:         ripe-tech@ono.es 20030422
source:          RIPE
```

213.180.193.68 | **Active Mstream Client for internal Mstream handlers**

```
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html
inetnum:        213.180.192.0 - 213.180.193.255
netname:         COMPTEK-NET1
descr:           CompTek International
descr:           3, Gubkina str., Moscow, 117809
country:         RU
admin-c:         YNDX1-RIPE
tech-c:          YNDX1-RIPE
status:          ASSIGNED PA
notify:          noc@yandex.net
mnt-by:          COMPTEK-MNT-RIPE
changed:         wawa@comptek.ru 20020607
source:          RIPE
route:          213.180.192.0/20
descr:           CompTek network / special
origin:          AS13238
notify:          noc@comptek.ru
mnt-by:          COMPTEK-MNT-RIPE
changed:         wawa@comptek.ru 20010123
source:          RIPE
role:           Yandex LLC Network Operations
address:         Yandex LLC
address:         40A Vavilova st.
address:         117333, Moscow, Russia
phone:           +7 095 9743555
fax-no:          +7 095 9743565
e-mail:          noc@yandex.net
trouble:         ------------------------------------------------------
trouble:         Points of contact for Yandex LLC Network Operations
trouble:         ------------------------------------------------------
trouble:         Routing and peering issues:  noc@yandex.net
trouble:         SPAM issues:                 abuse@yandex.ru
trouble:         Network security issues:     abuse@yandex.ru
trouble:         Mail issues:                 postmaster@yandex.ru
trouble:         General information:         info@yandex.ru
trouble:         ------------------------------------------------------
```

```
admin-c:      VLI1-RIPE
tech-c:       KBG2-RIPE
notify:       noc@yandex.net
nic-hdl:      YNDX1-RIPE
mnt-by:       COMPTEK-MNT-RIPE
changed:      wawa@comptek.ru 20020607
source:       RIPE
```

213.231.96.32	Active Mstream Client for internal Mstream handlers

```
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html
inetnum:      213.231.93.0 - 213.231.127.255
netname:      CANARIASTELECOM
descr:        AUNA TLC - CABLETELCA, S.A.
descr:        PROVIDER LIR
country:      ES
admin-c:      TA718-RIPE
tech-c:       TA718-RIPE
status:       ASSIGNED PA
notify:       techauna@auna.es
mnt-by:       AUNA-MNT
mnt-lower:    AUNA-MNT
changed:      techauna@auna.es 20031107
source:       RIPE
route:        213.231.64.0/18
descr:        CANARIASTELECOM
origin:       AS16040
notify:       techauna@auna.es
mnt-by:       AUNA-MNT
changed:      jsanchez@cabletelca.es 20020307
changed:      gestionripe@cabletelca.es 20021118
changed:      techauna@auna.es 20031020
source:       RIPE
role:         Techauna AUNA
address:      Avenida Diagonal, 579
address:      Barcelona 08014
address:      Spain
phone:        +34 93 502 0000
fax-no:       +34 93 502 2809
e-mail:       techauna@auna.es
admin-c:      TA718-RIPE
tech-c:       TA718-RIPE
nic-hdl:      TA718-RIPE
notify:       techauna@auna.es
mnt-by:       AUNA-MNT
remarks:      ------------------------------------------------
remarks:      for net abuse questions please contact:
remarks:      abuse@auna.es
remarks:      ------------------------------------------------
changed:      techauna@auna.es 20031119
source:       RIPE
```

212.195.102.30	Active Mstream Client for internal Mstream handlers

```
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html
inetnum:      212.195.64.0 - 212.195.255.255
```

```
netname:        T-ONLINEFRANCE-ADSL
descr:          Pools for ADSL customers
country:        FR
admin-c:        NOCT1-RIPE
tech-c:         NOCT1-RIPE
status:         ASSIGNED PA
notify:         ripe@t-online.fr
mnt-by:         T-ONLINEFRANCE
changed:        vox@t-online.fr 20021015
source:         RIPE
route:          212.194.0.0/15
descr:          T-Online France - Club Internet
origin:         AS5410
notify:         ripe@t-online.fr
mnt-by:         T-ONLINEFRANCE
changed:        vox@t-online.fr 20040112
source:         RIPE
role:           Network Operation Centre T-ONLINE FRANCE
address:        T-Online France - Club Internet
address:        11 rue de Cambrai
address:        75019 Paris
address:        France
phone:          +33 1 55 45 45 00
fax-no:         +33 1 55 45 47 78
e-mail:         ripe@t-online.fr
admin-c:        AV-RIPE
tech-c:         AV-RIPE
tech-c:         OB346-RIPE
tech-c:         DA3757-RIPE
tech-c:         OT1274-RIPE
nic-hdl:        NOCT1-RIPE
mnt-by:         T-ONLINEFRANCE
changed:        vox@t-online.fr 20040504
source:         RIPE
```

213.189.89.109	(Source of majority of "Possible Trojan Server" traffic)

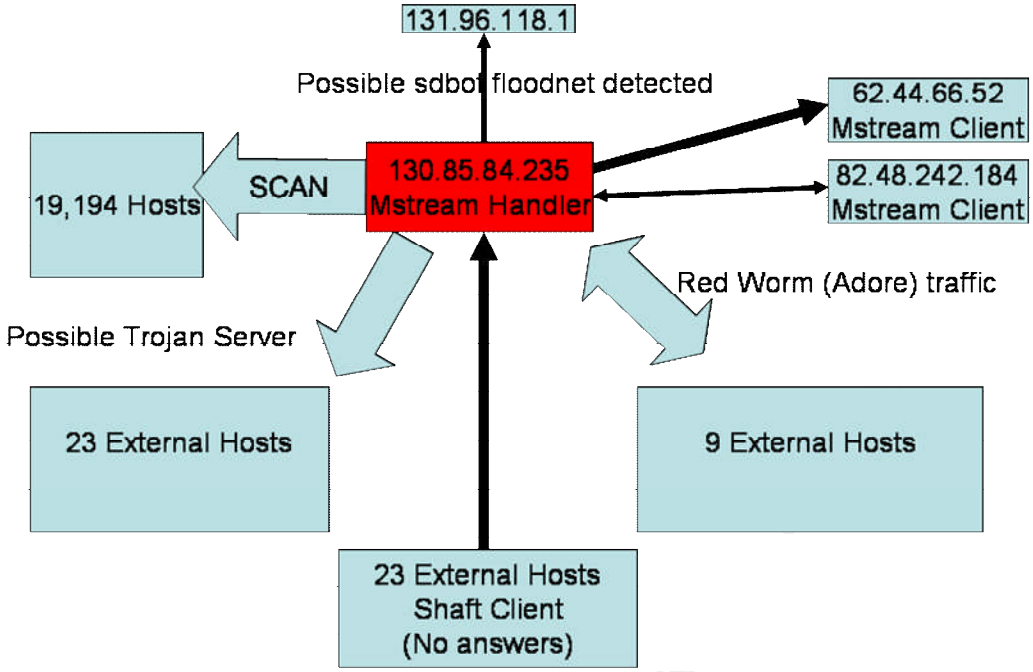
```
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html
inetnum:        213.189.89.0 - 213.189.89.255
netname:        STAFF-NET
descr:          STAFF SEGMENT
country:        KW
admin-c:        QNET1-RIPE
tech-c:         AA581-RIPE
status:         ASSIGNED PA
notify:         admin-c@qualitynet.net
mnt-by:         QNET-NOC
changed:        admin-c@qualitynet.net 20030611
source:         RIPE
route:          213.189.64.0/19
descr:          QualityNet Kwait
origin:         AS9155
member-of:      RS-QNET
mnt-by:         QNET-NOC
changed:        hia@qualitynet.net 20000401
source:         RIPE
person:         Qnet Admin Contact
```

address: Kuwait phone: +965 80 8888 e-mail: admin-c@qualitynet.net nic-hdl: QNET1-RIPE notify: stinger@qualitynet.net mnt-by: MOC-MNT changed: stinger@qualitynet.net 20030611 source: RIPE **person**: Abdulaziz Al-osaimi address: Ministry of Communications address: Po box 318 Safat, 1111 Kuwait phone: +965 481 1036 nic-hdl: AA581-RIPE notify: asr@itsq8.com mnt-by: MOC-MNT changed: asr@itsq8.com 19980115 source: RIPE
82.48.242.184 **Mstream Client (from Link Graph)**
% Rights restricted by copyright. % See http://www.ripe.net/ripencc/pub-services/db/copyright.html **inetnum**: 82.48.240.0 - 82.48.255.255 netname: TELECOM-ADSL-3 descr: Telecom Italia S.p.A. descr: E@sy.ip service descr: Wholesale service for ISP country: IT admin-c: BS104-RIPE tech-c: BS104-RIPE status: ASSIGNED PA remarks: Please send abuse notification to abuse@telecomitalia.it notify: net_ti@telecomitalia.it mnt-by: TIWS-MNT changed: net_ti@telecomitalia.it 20040101 source: RIPE **route**: 82.48.0.0/16 descr: INTERBUSINESS origin: AS3269 notify: network@cgi.interbusiness.it mnt-by: TIWS-MNT mnt-routes: INTERB-MNT changed: net_ti@telecomitalia.it 20031016 source: RIPE **person**: BBBEASYIP STAFF address: Via Val Cannuta, 250 address: I-00100 Roma address: Italy phone: +39 06 36881 e-mail: ripe-staff@telecomitalia.it nic-hdl: BS104-RIPE notify: ripe-staff@telecomitalia.it changed: net_ti@telecomitalia.it 20001019 source: RIPE
68.54.84.49 **Top Talker from OOS log**
CustName: Comcast Cable Communications, Inc. Address: 3 Executive Campus

```
Address:    5th Floor
City:       Cherry Hill
StateProv:  NJ
PostalCode: 08002
Country:    US
RegDate:    2003-03-19
Updated:    2003-03-19

NetRange:   68.54.80.0 - 68.54.95.255
CIDR:       68.54.80.0/20
NetName:    BALTIMORE-A-4
NetHandle:  NET-68-54-80-0-1
Parent:     NET-68-32-0-0-1
NetType:    Reassigned
Comment:    NONE
RegDate:    2003-03-19
Updated:    2003-03-19

TechHandle: IC161-ARIN
TechName:   Comcast Cable Communications Inc
TechPhone:  +1-856-317-7200
TechEmail:  cips_ip-registration@cable.comcast.com

OrgAbuseHandle: NAPO-ARIN
OrgAbuseName:   Network Abuse and Policy Observance
OrgAbusePhone:  +1-856-317-7272
OrgAbuseEmail:  abuse@comcast.net

OrgTechHandle: IC161-ARIN
OrgTechName:   Comcast Cable Communications Inc
OrgTechPhone:  +1-856-317-7200
OrgTechEmail:  cips_ip-registration@cable.comcast.com
```

Link graph and analysis

The system engr-84-235.pooled.umbc.edu (IP address 130.85.84.235 seemed to pop up in several places. In order to understand better what was happening to this system, a link graph has been drawn to show the relationships between this system and the exploits occurring on the network. From this diagram we can see that the system has definitely been compromised by more than one piece of malicious code, and that the system is actively scanning other hosts. (The intent of the scanning is likely to compromise additional hosts to become agents for the next DDOS attack) Fortunately this audit has not found evidence of Shaft or mstream agents installed on the network.



Compromised System List

IP Address	Event(s)	Recommendation
130.85.1.3	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.1.4	DDOS mstream Compromised	Analyze and Rebuild
130.85.10.79	NIMDA Compromised	Analyze and Rebuild
130.85.110.7	DDOS mstream Compromised	Analyze and Rebuild
130.85.110.82	MiMail Compromised	Analyze and Rebuild
130.85.111.34	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.112.152	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.112.163	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.12.4	Red Worm (Adore) Compromised Probable Trojan Server	Analyze and Rebuild

130.85.12.6	Red Worm (Adore) Compromised MiMail Attack Probable Trojan Server	Analyze and Rebuild
130.85.12.7	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.150.199	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.150.253	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.151.75	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.152.21	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.153.174	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.153.195	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.153.35	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.153.8	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.153.94	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.16.90	Possible Trojan Server	Analyze system
130.85.17.45	NIMDA Compromised	Analyze and Rebuild
130.85.190.1	Probable Trojan Server	Analyze and Rebuild if needed
130.85.190.102	Probable Trojan Server	Analyze and Rebuild if needed
130.85.190.202	Probable Trojan Server	Analyze and Rebuild if needed
130.85.190.203	Probable Trojan Server	Analyze and Rebuild if needed
130.85.190.93	Probable Trojan Server	Analyze and Rebuild if needed
130.85.190.95	Probable Trojan Server	Analyze and Rebuild if needed
130.85.190.97	Probable Trojan Server	Analyze and Rebuild if needed
130.85.24.20	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.24.33	Red Worm (Adore) Compromised	Analyze and Rebuild

130.85.24.34	Red Worm (Adore) Compromised Probable Trojan Server	Analyze and Rebuild
130.85.24.44	Red Worm (Adore) Compromised Probable Trojan Server	Analyze and Rebuild
130.85.24.74	Red Worm (Adore) Compromised Probable Trojan Server	Analyze and Rebuild
130.85.25.10	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.25.11	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.25.66	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.25.67	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.25.68	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.25.69	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.25.70	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.25.71	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.25.73	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.30.3	Watched System	Analyze system
130.85.30.4	Watched System	Analyze system
130.85.34.11	Red Worm (Adore) Compromised Probable Trojan Server	Analyze and Rebuild
130.85.34.14	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.34.5	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.42.2	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.43.10	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.5.100	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.53.29	FTP to Helpdesk	Analyze system
130.85.55.27	Possible Trojan Server	Analyze system
130.85.6.15	Probable Trojan Server	Analyze and

		Rebuild if needed
130.85.6.62	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.6.7	Red Worm (Adore) Compromised Probable Trojan Server	Analyze and Rebuild
130.85.60.14	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.60.16	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.60.17	DDOS mstream Compromised Red Worm (Adore) Compromised Probable Trojan Server	Analyze and Rebuild
130.85.66.56	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.69.22	Possible Trojan Server	Analyze system
130.85.70.197	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.70.225	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.70.49	FTP to Helpdesk	Analyze system
130.85.70.50	FTP to Helpdesk	Analyze system
130.85.70.96	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.71.248	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.80.224	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.80.28	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.80.5	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.82.43	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.82.79	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.82.8	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.84.23	DDOS mstream Compromised	Analyze and Rebuild
130.85.84.235	Red Worm (Adore) Compromised Probable Trojan Server	Analyze and Rebuild
130.85.97.104	Red Worm (Adore)	Analyze and

	Compromised	Rebuild
130.85.97.106	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.97.11	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.97.124	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.97.135	MiMail Compromised	Analyze and Rebuild
130.85.97.166	NIMDA Compromised	Analyze and Rebuild
130.85.97.175	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.97.180	NIMDA Compromised	Analyze and Rebuild
130.85.97.182	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.97.196	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.97.217	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.97.228	NIMDA Compromised	Analyze and Rebuild
130.85.97.25	NIMDA Compromised	Analyze and Rebuild
130.85.97.28	DDOS mstream Compromised	Analyze and Rebuild
130.85.97.36	NIMDA Compromised	Analyze and Rebuild
130.85.97.41	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.97.44	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.97.51	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.97.59	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.97.66	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.97.69	NIMDA Compromised	Analyze and Rebuild
130.85.97.70	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.97.74	NIMDA Compromised	Analyze and Rebuild

130.85.97.87	Red Worm (Adore) Compromised Probable Trojan Server	Analyze and Rebuild
130.85.97.88	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.97.92	Red Worm (Adore) Compromised Possible Trojan Server	Analyze and Rebuild
130.85.97.94	Red Worm (Adore) Compromised MiMail Compromised	Analyze and Rebuild
130.85.97.95	Red Worm (Adore) Compromised	Analyze and Rebuild
130.85.98.87	Red Worm (Adore) Compromised	Analyze and Rebuild

IP addresses highlighted in orange are also participating in active scanning.

In addition to these systems identified above, the following additional systems exhibit signs of compromise in that they are participating in Portscans.

Internal Hosts PortScanning			
130.85.111.5	130.85.81.77	130.85.43.5	130.85.69.24
130.85.81.39	130.85.97.46	130.85.97.75	130.85.97.83
130.85.53.16	130.85.97.81	130.85.83.91	130.85.97.15
130.85.97.79	130.85.53.22	130.85.97.14	130.85.112.2
130.85.97.55	130.85.97.35	130.85.97.43	130.85.11.13
130.85.84.22	130.85.98.61	130.85.153.7	130.85.97.23
130.85.97.30	130.85.43.3	130.85.43.2	130.85.97.53
130.85.5.44	130.85.97.77	130.85.97.68	130.85.97.38
130.85.53.41	130.85.84.14	130.85.152.1	

Defensive recommendations

Policy:
The appropriate use policy located at http://www.umbc.edu/oit/security/policy/2-UMBC/IT-01-final.html is currently more than 7 years old. While the basics of the document are still sound and valid, there are likely to be some technology issues which should be updated in that timeframe.

Some of the information, while highly useful within the Campus, should not be published to servers freely available off of the campus as it provides quite useful reconnaissance information to a potential hacker. Two examples are: http://www.umbc.edu/oit/sans/physnet/noc/approved_equip.html which provides a listing of the preferred routers, switches, and hubs in use on the campus, and an under construction page located at http://www.umbc.edu/oit/sans/physnet/noc/layoutinfo.html which promises to provide a network layout diagram. While none of these pages on their own will lead to a compromise, individuals who are not part of the campus do not have a need to know for this information.

Procedures and Implementation:
The centralized location for obtaining security information and tools located at http://www.umbc.edu/oit/security/ is an excellent method to help and encourage the entire campus to think securely. Once again, portions of these pages including references to an Urgent Microsoft Security alert are 18 months out of date. This information and service only provides a value added if it is up to date. Opportunities for improvement include adding pointers to MS Update, or adding a System Update Server (SUS) to the network. The advantage of running your own SUS server are that you can test updates prior to deploying them to your user community.

Upgrade the IDS sensors and move to storing alert / scan / OOS data in databases (instead of flat files to improve the efficiency of the analysis. This should also decrease the corruption of the datafiles and will also result in a consistent log structure. The additional tools available with the database implementations will improve analyst efficiency by eliminating much of the manual manipulation of data which is required to prepare the data for loading into a database now. Snort 2.1.3 provides recursive scanning to eliminate alert obfuscation (hiding the real attack behind a signature that the administrator cares less about.)

Review the border security policy with a goal of creating a list of ports that should NEVER pass in or out of the network. In addition, create a list of servers which are dedicated only to internal activities, and block those ip addresses from passing through the firewall. An example of this would be the Helpdesk.

Analysis Process

The analysis workstation was based on Windows XP with MySql, and Perl. Perl scripts were adapted from the work of Jason Lam, and from the snortsnarf perl scripts. Guidance on the cleanup of data files was obtained from the work of Ian Martin.

After preprocessing the data files into a more common structure the files were parsed and loaded into the MySql database. From here SQL queries and in some cases simple Grep processing on the original files were used to obtain the results.

The number of un-parseable log errors was well under 1%.

References:
The following sources provided background for the completion of this assignment.

[CSE1] Communications Security Establishment, "A Guide to Security Risk Management for Information Technology Systems," 1996. http://www.cse.dnd.ca/

[CSE2] Communications Security Establishment, "A Guide to Risk Assessment And Safeguard Selection for Information Technology Systems," 1996. http://www.cse.dnd.ca/

[ISO1] International Standard ISO/IEC 17799:2000 Code of Practice for Information Security Management, 2000.

[SANS1] Intrusion Detection Immersion Curriculum – SANS, (Track 3 Intrusion Detection In-Depth, 2001.

Prabhacker, Rajasingham, "Threat and Risk Assessments: Some Issues", April 2001, http://www.sans.org/infosecFAQ/audit/risk.htm

[i] CSE, "A Guide to Security Risk Management for Information Technology Systems," 1996.
[ii] CSE. "A Guide to Risk Assessment And Safeguard Selection for Information Technology Systems," 1996

Last Updated: November 18th, 2017

Upcoming Training

SIEM & Tactical Analytics Summit & Training	Scottsdale, AZ	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Munich December 2017	Munich, Germany	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event
Las Vegas 2018 - SEC503: Intrusion Detection In-Depth	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	vLive
SANS London February 2018	London, United Kingdom	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS Dallas 2018	Dallas, TX	Feb 19, 2018 - Feb 24, 2018	Live Event
Community SANS Pensacola SEC503	Pensacola, FL	Feb 19, 2018 - Feb 24, 2018	Community SANS
Community SANS Baltimore SEC503	Baltimore, MD	Mar 12, 2018 - Mar 17, 2018	Community SANS
SANS Northern VA Spring - Tysons 2018	McLean, VA	Mar 17, 2018 - Mar 24, 2018	Live Event
SANS Secure Canberra 2018	Canberra, Australia	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS 2018	Orlando, FL	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS Abu Dhabi 2018	Abu Dhabi, United Arab Emirates	Apr 07, 2018 - Apr 12, 2018	Live Event
SANS London April 2018	London, United Kingdom	Apr 16, 2018 - Apr 21, 2018	Live Event
SANS Baltimore Spring 2018	Baltimore, MD	Apr 21, 2018 - Apr 28, 2018	Live Event
Baltimore Spring 2018 - SEC503: Intrusion Detection In-Depth	Baltimore, MD	Apr 23, 2018 - Apr 28, 2018	vLive
SANS vLive - SEC503: Intrusion Detection In-Depth	SEC503 - 201805,	May 02, 2018 - Jun 14, 2018	vLive
SANS Security West 2018	San Diego, CA	May 11, 2018 - May 18, 2018	Live Event
SANS Minneapolis 2018	Minneapolis, MN	Jun 25, 2018 - Jun 30, 2018	Live Event
Community SANS Columbia SEC503	Columbia, MD	Aug 13, 2018 - Aug 18, 2018	Community SANS
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced