Global Information Assurance Certification Paper

"MOM, 3 Detects, and 5 Days of Crunched Data - A SANS/GCIA Practical Submission".

GIAC Certified Intrusion Analyst (GCIA) Version 3.3

In conjunction with Mary Washington College and the James Monroe Center.
Spring 2003 Semester.

Candidate:
Don Murdoch

Submission Date: June 11, 2003

Table of Contents

Conventions and Administrivia

Footnotes: Supporting information for points made, facts expressed, and web site / web articles in this paper are listed in footnotes and in the references section. This prevents disruption in reading the paper.

Microsoft Email: In Assignment 2, Detect 3 a reference to email with a Microsoft representative is mentioned; the sender did not grant or deny permission to use the email and was specifically asked (no answer from sender). This email will be provided to SANS/GIAC if necessary.

RIAA/MPAA/DMCA: Assignment 3 makes reference to the Recording Industry Association of America (www.riaa.org), the Motion Pictures Association of America (www.mpaa.org), and the Digital Millennium Copyright Act as acronyms.

Output from Commands: Commands and their output are shown in
`Courier Type, 10 Point.`

References to GIAC/GCIA candidates: There are several references to SANS GIAC candidates in this paper. Specific URL's are all listed in the References section, if there is not a footnote reference. Every effort has been made to credit sources.

Assignment One: Intrusion Detection with MOM - Going Above the Wire

Introduction

There are several areas, or layers, where intrusions into a system can occur. At the "wire" or network layer, there are several tools that can successfully discern the nature of traffic for most commercial protocols. But how do you respond to the challenge of knowing what happens when you need to analyze "above the wire", at the operating system and application layers? What about when traffic is properly formed and does not trigger IDS rules? By focusing on the WAN/LAN layer traffic and looking for "exception traffic" – signatures within packets that are indicative of malicious intent - properly formed, legal traffic is virtually ignored. With attackers getting more sophisticated, the analyst needs to respond with tools that can be used above the wire at the application and operating system level.

In this paper, Microsoft Operations Manager 2000 (hence, MOM) will be discussed as a tool to aid the analyst in understanding what occurs within the operating system and the application level.

Recent Statistics

Over the past several years, a variety of studies have revealed that while attacks from outside an organization have increased, greater financial loss has occurred from deliberate actions by staff within an organization. Some studies conducted during 2001 indicated that as much as 80% of the identified financial loss is from insiders, not outsiders[1], while others clearly indicate that their Internet connection is responsible for 2/3 of attacks[2]. Unauthorized insider access has varied between 15% and 25% over 1997 to 2002, with losses ranging from a low of $1000 to $50M for the same period[3]. These statistics emphasize the point that an organization needs to look both within and without for intrusions, anomalies, and violations of computer usage policy.

Introducing Microsoft Operations Manager 2000

Microsoft Operations Manager 2000 (hence MOM[4]) is Microsoft's solution for event management, centralized reporting, and automated event response for the Windows NT/2000/2003 operating system and most of Microsoft's BackOffice product line. There are many capabilities of MOM that are beyond the scope of this paper; emphasis here is on features that aid and assist the intrusion analyst in identification and examination of Events Of Interest (EOI) particular to the Windows environment.

[1] Source URL: http://www.all.net/journal/netsec/2001-05.html
[2] Source URL: http://www.gocsi.com/press/20020407.html
[3] Computer Security Institute. "2002 Computer Security Institute/FBI Computer Crime & Security Survey", p. 10. URL: http://www.gocsi.com/press/20020407.html
[4] MOM2000, for the purposes of this paper, is running on Windows 2000 Service Pack 3, with Active Directory. MOM's features URL: http://www.microsoft.com/mom/evaluation/features/default.asp

MOM Architecture

The major components of MOM reside in one of three tiers (following Microsoft's three tier component architecture model) as illustrated in the figure.

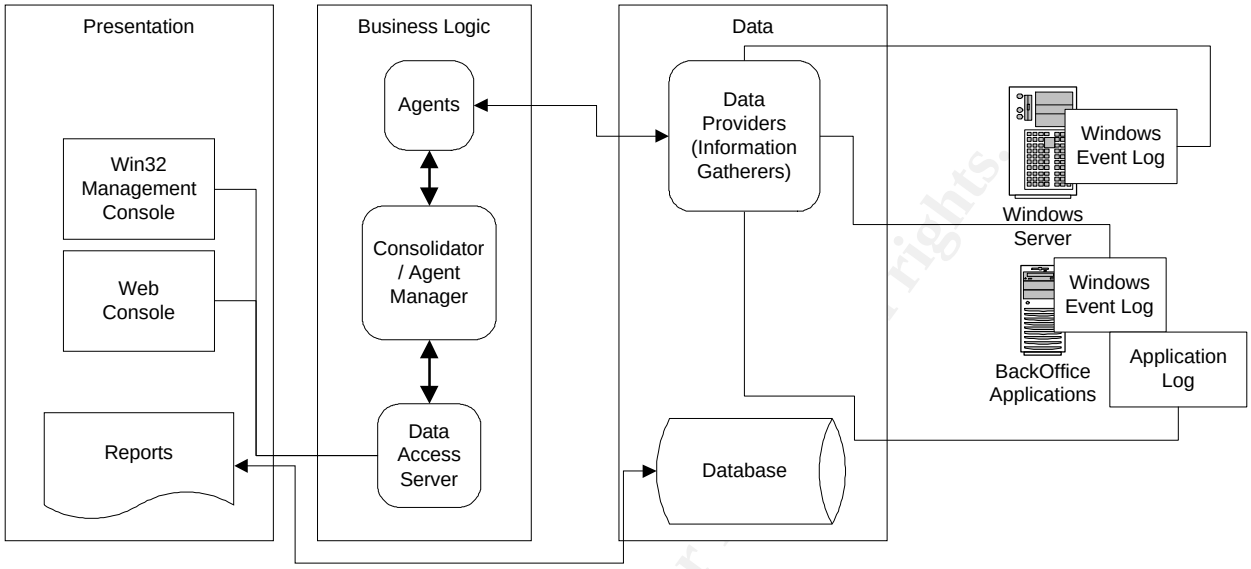


Figure 1: Basic MOM Architecture

The presentation tier is composed of user interface applications – whether via the web through a browser or with the Win32 MOM console. At the business logic layer, the agents monitor the systems and take action based on processing rules defined using the console. The consolidator is the focal point for all reported information and it provides data confidentiality by using encrypted channels. It collects and processes all data from agents. MOM's agents run on managed servers in the enterprise, and are configured with processing rules. These agents read the Windows event log and process data. There are a variety of other data providers designed to appeal to the enterprise. These include a syslog interface for receiving information from UNIX systems, a generic log file provider for applications that write single line log files, SNMP traps, and performance counters embedded within Windows.

MOM2000 Processing Rules

MOM has three classes of processing rules. These rules define how MOM collects and responds to information generated by monitored systems. The three processing rule classes are event, alert, and performance.

Within the event processing rule category, there are five distinct actions that MOM performs as its agents report information. First, it can either alert or respond to a specific event. Second, it can search for missing events for a given time period. This is an example of process by exception – if a specific event is not seen, then something deserves some attention. Third, MOM can also consolidate and summarize events. Fourth, MOM can filter out insignificant events, essentially discarding events that are reported from monitored systems. This feature allows you to audit for events on a

computer, but not present them for reporting and alerting within MOM. Last, MOM the system can specify that specific data should be collected from specific sources.

Alert processing rules are the second set of processing rules. MOM can take a variety of actions based on the collected data. It can perform a variety of actions in order to communicate with operations staff, such as paging and email notification.

The last group of processing rules is performance counter monitoring rules. These are based on Windows Management Instrumentation (WMI), which is Microsoft's implementation of the DTMF standards. Here, MOM monitors for specific measurement statistics on system and application performance. MOM can also generate an alert if a specific performance counter passes a threshold - say a disk reaches 90% capacity.

Windows Events of Interest

There are numerous Events of Interest (EOI's) that the Windows OS will record in the various event logs and application log files. MOM monitors the Windows event logs and takes action based on events or event characteristics posted to the event logs (System, Security, Application, DNS, File Replication, and Directory Service). MOM is delivered with several Management Packs designed to monitor event logs. Depending on how auditing configured throughout Windows, highly granular information can be reported to the event logs. Event monitoring can be enabled or disabled as needed. Below are some representative Management Packs provided with MOM that relate to intrusion detection.

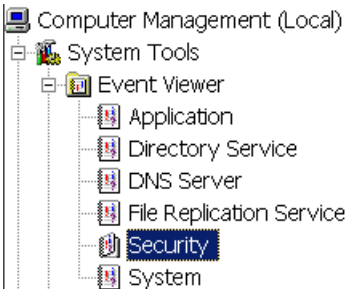


Figure 2: Domain Controller Event Logs

Sample Management Pack	EOI's and Area of Functionality
Default Event Collector	Monitors all event logs and provides comprehensive monitoring.
Domain Name Service	All DNS events.
Internet Information Service	Monitors IIS (web server), FTP, SMTP, and NNTP services. There are also sample scripts to determine server responsiveness.
Routing and Remote Access	Monitor dial up devices, VPN client access failures and bad logon attempts, and capacity issues.
Active Directory	Access to the directory service, replication, and security if auditing is configured within the directory service.

Microsoft also supplies Application Packs that monitor specific products such as Exchange, SQL Server, and Systems Management Server. These contain additional rules configured to monitor for events specific to the monitored application.

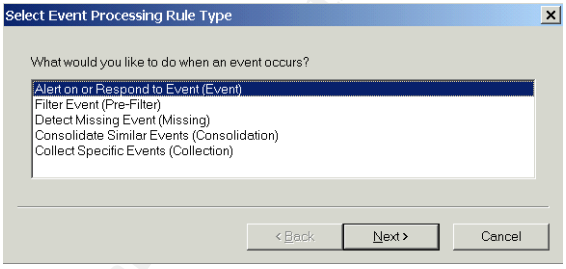
Event Processing Rule Setup and Example

In this example, a rule will be configured to monitor for failed logon attempts. Failed logon attempts qualify as an EOI, since they indicate an access was attempted without knowing proper authentication credentials.

First, open the MOM console and survey the list of Processing Rule Groups under the Rules node. Note that there are a variety of groups, which have logically associated processing rules. Also, if a processing rule group is not assigned to a particular computer group type, the defined rules will not be processed. Therefore, in order to activate a processing rule group, one must right click on the processing rule group, select Properties, the Computer Group tab, and add the appropriate computer group. Servers that are in this group type will process these rules.

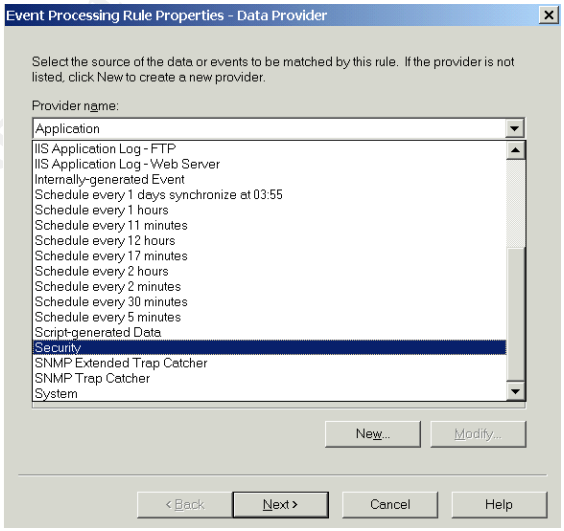
Step One: Determine rule type, and choose "Alert on or Respond to Event (Event)".

Then press Next.



Step Two: Chose the "Security" provider from the list.

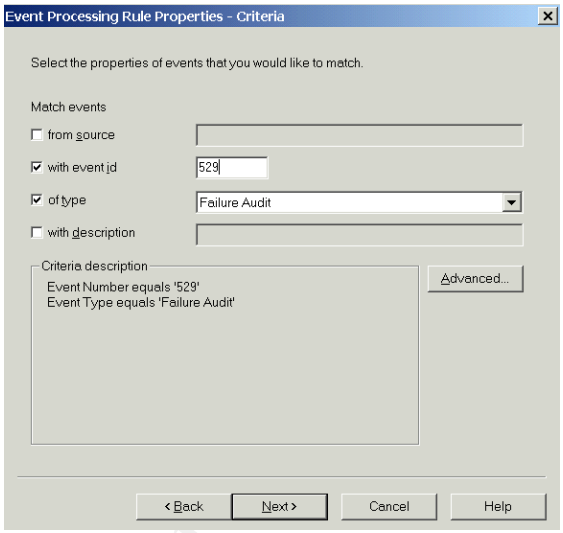
Then press Next.



Step Three: Enter in the specific criteria that this rule needs to match against.

Here, the Event ID 529 corresponds a Logon/Logoff audited event from the Security subsystem. The criteria are further limited to Failure Audits, because the alert should not fire for successful logon events. Note that this event is not generated unless auditing is enabled.

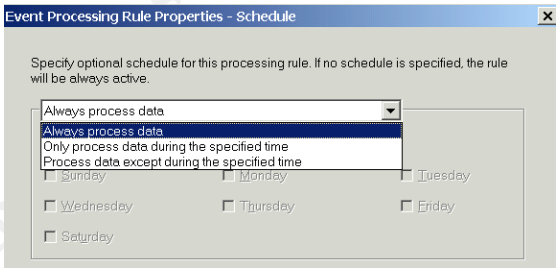
Then press Next.



Step Four: Select "Always process data" from the menu.

This type of event monitoring should be round the clock.

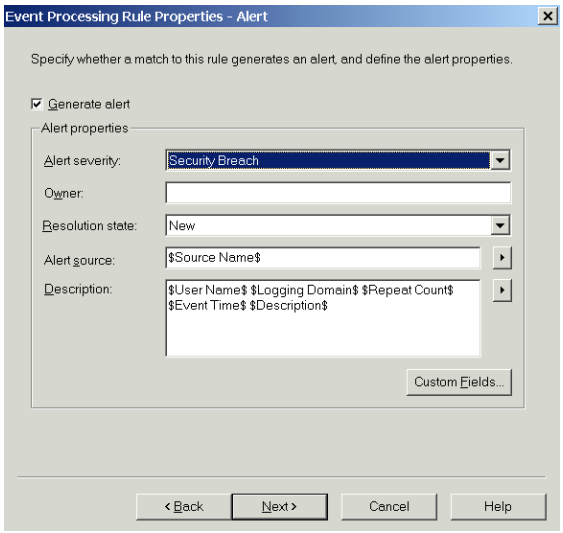
Then press Next.



Step Five: Check the Generate Alert Checkbox. Then, for this alert, select Security Breach for the Severity.

In order to get the additional data fields to be reported in the Description, press the arrow button and select the fields from the popup list. MOM will retrieve additional data from the reported event.

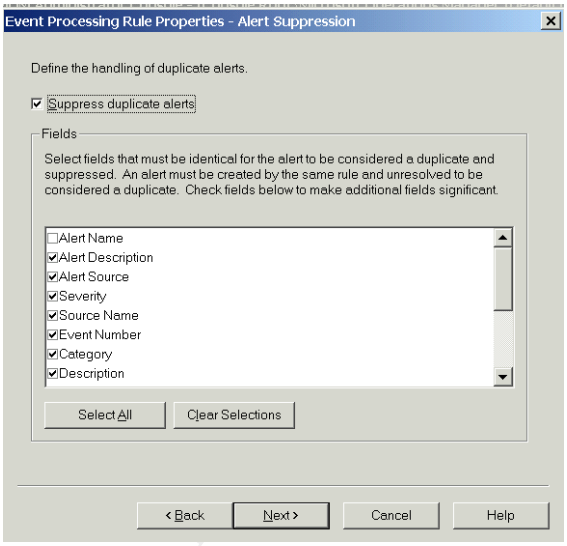
Then press Next.



Step Six: Check the Suppress duplicate alerts box.

By default, MOM will attempt to roll up multiple matching alerts into one alert. This prevents an alert storm from occurring. Duplicate suppression is based on matching fields, as shown in the dialog.

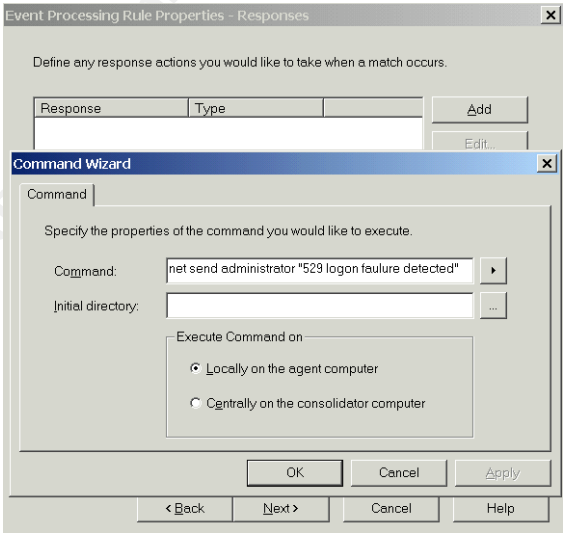
Then press Next.



Step Seven: If desired, define an automated response (as shown here) by pressing the Add button.
There are five types of responses.

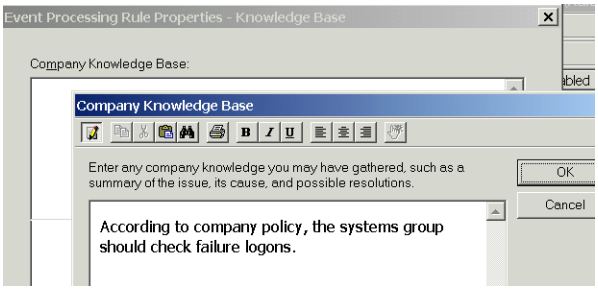
- Launch a Script
- Send an SNMP Trap
- Send a notification to a notification group
- Execute a command or batch file
- Update state variable

The response here is to execute the "net send" command (windows popup on the console).



Then press Next.
Step Eight: If desired, text can be entered in the Knowledge base tab (the Edit button is covered).

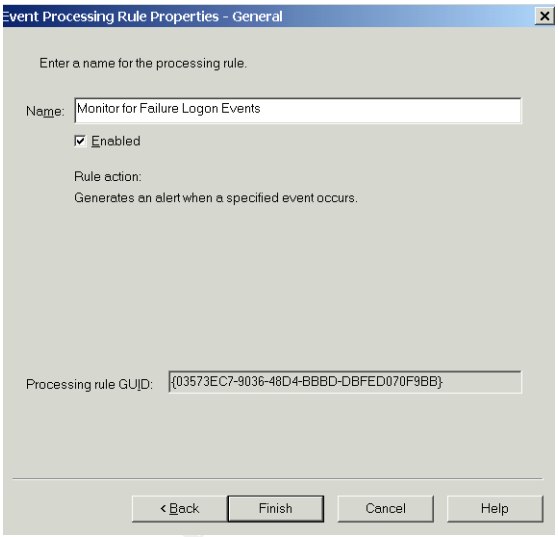
This area can be used for a variety of purposes, and is editable when responding to an alert - useful for site-specific information.



Then press Next.

Step Nine: Enter in an appropriate name for the event-processing rule.

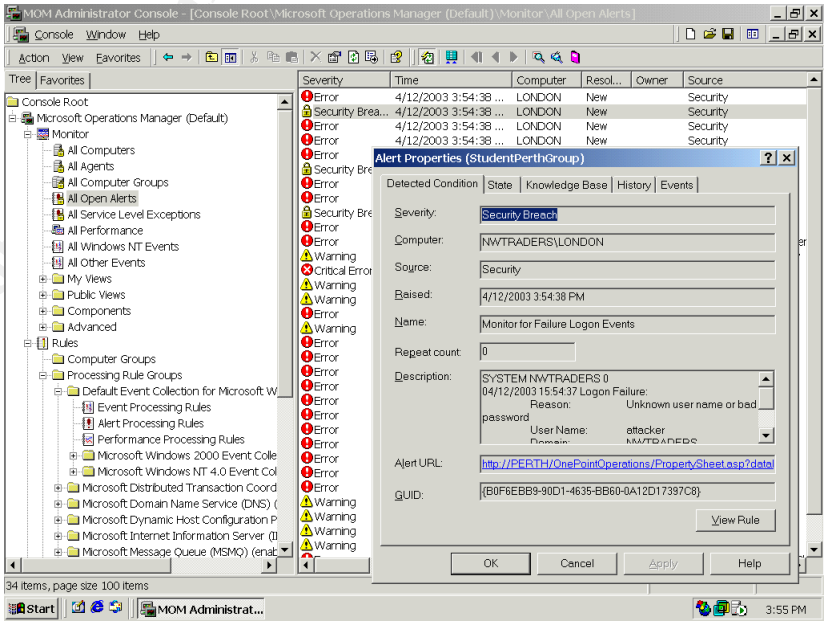
Then press Finish, and MOM will save the event definition.



Once the event-processing rule is defined, the changes will need to be committed to the MOM agents. In the MOM console, right click on the Rules node and select "Commit Configuration Change". MOM will inform agents that there are new rules, and this update will occur in a few minutes. Changes are committed to the MOM server when Event 21241 appears in the Application log, and Event ID 21240 appears on the agents.

Note in this screen capture the details on the alert. There are also other alerts from the domain, and a variety of tabs in the alert dialog.

This screen capture shows the MOM console responding to the alert configured above.



Considerations in Deployment

There are several considerations one should consider when deploying MOM in a manner that can best support an Information Security role within the organization. In particular, the needs of an intrusion analyst – having a reference point for EOI correlation from an IDS – require quite a bit of planning for MOM deployment.

Events to Monitor

Microsoft has documented about 6500 events that Windows produces. Below is a non-exhaustive list of some examples of security events to monitor for intrusion detection on systems or across the enterprise (candidates for Windows EoI's).

Event	Description
517	The audit log was cleared by a specific user.
528/529	Successful/unsuccessful logon by specified user.
531	Logon attempted to disabled account
532	Logon attempted to expired account (accounts can be time limited)
538	Logon attempted to a locked out account
546	Internet Key Exchange (IKE) session establishment failed
547	Internet Key Exchange (IKE) session negotiation failed
564	Object deleted by process
576	Special privileges applied to user
612-615	Events for IPSec policy changes
1309	Impersonation attempted on a thread not associated with a client
1317,1319	Specified user, group does not exist
1326-1331	Logon failures for specific events relating to account status

As can be seen from this representative list, there are a variety of highly granular events that Windows can be configured to produce which help the intrusion analyst. Note that often auditing is not enabled for a specific service by default – so if monitoring is desired, then auditing will need to be configured using the specific management tool for that service.

Example Usage of MOM for Intrusion Detection

Exploring MalWare: There are several worms and viruses that attempt to explore and make use of network shares. If specific systems are configured to audit for "failure" events when a Windows share is accessed, MOM can inform an operator within a few minutes. Recent examples of this behavior include the SoBigB (May 2003) and Nimda (2001) virus's which explore network drives on all possible machines.

Repeated Administrator Logon Attempts: Since the "Administrator" account cannot be locked out by default in Windows NT4 and Windows 2000, an operator can be informed in a few minutes about a dictionary or brute force password attack attempt.

Anomalous Disk Usage: One of the malicious uses of compromised systems is to use them as file servers for movies, MP3's, and pirated software. An attacker may be smart enough to not fill up a disk, but the performance monitor counters can be configured to monitor for logical disk space (as opposed to physical) usage and if it reaches a threshold inform an operator.

Providing Event Correlation

One of the more difficult challenges in intrusion analysis is event correlation. It is highly valuable to be able to correlate EOI's raised by a network Intrusion detection system with events raised in an enterprise from its managed servers. By using an authoritative centralized timeserver for all servers, correlation between the networks' IDS and MOM can reliably be made. Since MOM cam use UNIX syslog data, an IDS or other UNIX processes can post information to MOM, improving event correlation chances[5].

Configuring Auditing

In order to actually receive audit events, auditing must be configured – it is not enabled by default for Windows NT/2000. The best place to configure domain-wide auditing policies is by using the "Default Domain Group Policy", and enable auditing. By doing this, a system administrator would not have to modify each server's local security policy.

At an absolute minimum, the domain should be configured to monitor for Failure evens for account logon, logon events, and system events. Windows 2000 and Active Directory differentiates local interactive logon from over the network logon - thus two different of logon event types. As show in the accompanying figure, Failure auditing for everything, and Success auditing for at least logon, account logon and system events[6] should be configured.

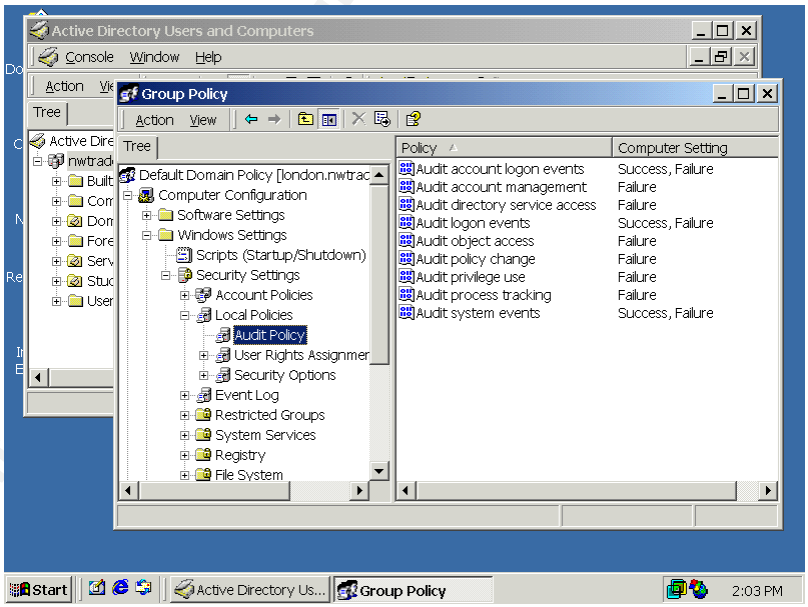


Figure 3: Windows Auditing with Group Policy

As mentioned earlier, most services under Windows have their own management tool. Using service specific tools is required in order to configure auditing related to a specific service. For example, if auditing Active Directory is required, the container nodes need to have specific auditing configured.

Deploying MOM may impact operating system performance and the network in general. MOM must be deployed in stages in order to make sure that each managed node functions properly and reports what it needs to report. Not all of MOM event reporting is

[5] For details on setting up MOM to use UNIX syslogs, see Microsoft Support article 297443. URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;297443

[6] Current consensus guidelines on auditing can be found in the SANS "Securing Windows 2000 Professional – Using the Gold Template Standard" and "Securing Windows 2000 Step by Step" books.

enabled by default. Site-specific event categories for processing rule groups can (and should) be set for individual or groups of systems as needed. DCAM reporting of events across a firewall boundary also require special deployment considerations and don't "just work" by default, as explained in the Installation Guide[7].

Summary

The intrusion analyst needs to be better armed and informed about what is occurring on the network. By concentrating on the packet layer, the analyst can miss valuable EoI's at the operating system and application level. MOM can be brought to bear on this space, thus increasing the quality of "above the wire" data which can help to detect intrusion, misuse, and violations of security policy within an organization and help better respond to incidents.

References: Assignment One

Microsoft Corporation. "Microsoft Operations Manager 2000 Product Information". 7 June 2003. URL: http://www.microsoft.com/mom/evaluation/default.asp

Microsoft Corporation. "Microsoft Operations Manager 2000 Deployment and Migration", 7 June 2003. URL: http://www.microsoft.com/mom/techinfo/deployment/default.asp

Microsoft Corporation. "Microsoft Operations Manager 2000 Product Documentation", 7 June 2003. URL: http://www.microsoft.com/mom/techinfo/productdoc/default.asp (the Users Guide, Installation Guide, and online Help are all available).

Jeff Shawgo, ed. "Securing Windows 2000 Step by Step (Ver 1.5)". The SANS Institute, July 1, 2001. Chapter 3.

Ben Bower, Dean Farrimgton, Chris Weber. "Security Windows 2000 Professional Using the Gold Standard Security Template". SANS Press, 2002.

Assignment Two – Three Network Detects.

Detect One: Proxy Scan Attempts

Source of Trace

The tcpdump binary files for this trace came from www.incidents.org website[8], specifically the "2002.4.30" log file. These packets were generated because they triggered an alert from a Snort instance running in binary logging mode[9].

Network Topology

Before analysis, one fact must be established: which is the home or monitored network? The answer is either 226.185.0.0/16 (unlikely), or a collection of 226.185.0.0/24 to 226.185.255.0/24 networks (what we have). How and why can this decision be made?

[7] Deployment of MOM through a firewall for servers on a DMZ or perimeter network is beyond the scope. See the MOM Installation Guide, Ch. 5 and Ch. 8 for more details. URL: http://www.microsoft.com/mom/docs/installg.pdf
[8] Source URL: http://www.incidents.org/logs/Raw/
[9] Source URL: http://www.incidents.org/logs/Raw/README

By closely following the excellent example set forth in André Cormier's GCIA practical[10], network layout can be determined.

One assumption must be made – that during the obfuscation process used to hide the source IP's, the network addresses were modified to look like multicast addresses in the 226.0.0.0 reserved range but the source addresses should be real enough.

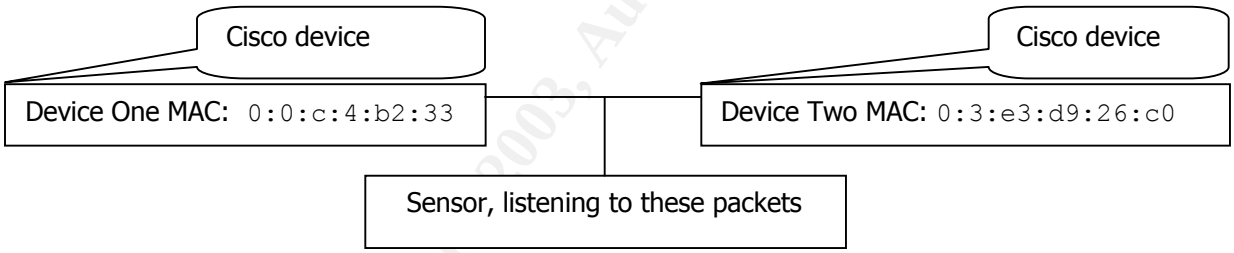
Step One. Determine source MAC addresses by having tcpdump report them and then determine uniqueness (second field). The command and results are:

```
$ tcpdump -neqr 2002.4.30 | cut -d ' ' -f2 | sort | uniq
0:0:c:4:b2:33
0:3:e3:d9:26:c0
```

Step Two. Determine destination MAC addresses by having tcpdump report them and determine uniqueness (third field). The command and results are:

```
$ tcpdump -neqr 2002.4.30 | cut -d ' ' -f3 | sort | uniq
0:0:c:4:b2:33
0:3:e3:d9:26:c0
```

Step Three. Network Layout: the network looks like this (Cisco hardware[11] in use):



Step Four. Determine source IP's from MAC address "0:0:c:4:b2:33" (fifth field).

```
$ tcpdump -neqr 2002.4.30 ether src 0:0:c:4:b2:33 | cut -d ' ' -f5 | cut -d \.
-f 1-4 | sort -t \. -n | uniq

226.185.106.176
```

Step Five. Determine destination IP's for MAC address "0:0:c:4:b2:33" (seventh field).

```
tcpdump -neqr 2002.4.30 ether src 0:0:c:4:b2:33 | cut -d ' ' -f 7 | cut -d \. -
f 1-4 | sort -t \. -n | uniq
```

The output is abbreviated to conserve space – there were 211 addresses produced with representative addresses listed below to show the span of IP's.

[10] Note: The analysis follows André Cormier's and the sequence of instructions in the order he presents. Source URL: http://cert.uni-stuttgart.de/archive/intrusions/2003/01/msg00121.html
[11] As determined from the IEEE MAC registrations: URL: http://standards.ieee.org/regauth/oui/oui.txt

```
12.219.192.109,   12.219.60.66,     12.230.85.251,    12.231.29.229
12.246.46.45,     12.250.207.52,    12.252.146.125,   12.26.84.145
80.14.177.160,    80.9.170.88,      128.200.144.60,   130.161.165.71
141.225.28.32,    142.59.25.232,    146.145.124.89,   147.97.7.118
204.253.104.15,   204.253.104.80,   205.138.230.129,  205.138.3.22
```

Step Six. Determine source IP's from MAC addres "0:3:e3:d9:26:c0" (field five):

```
$ tcpdump -neqr 2002.4.30 ether src 0:3:e3:d9:26:c0 | cut -d ' ' -f 5 | cut -d
\. -f 1-4 | sort t \. -n | uniq
```

The output is abbreviated to conserve space – there were 108 addresses produced with representative addresses listed below to show the span of IP's.

```
4.42.11.16,       12.18.157.202,    12.88.196.239,    24.45.15.143
63.118.236.100,   63.211.17.228,    63.240.15.5,      63.240.213.201
203.107.138.88,   203.122.47.137,   203.17.162.33,    203.197.101.17
203.69.227.10,    204.202.148.16,   205.252.49.1,     206.102.126.101
217.133.20.131,   226.196.64.17,    255.255.255.255
```

Step Seven. Determine destination IP's from MAC address "0:3:e3:d9:26:c0" (field seven):

```
$ tcpdump -neqr 2002.4.30 ether src 0:3:e3:d9:26:c0 | cut -d ' ' -f 7 | cut -d
\. -f 1-4 | sort -t \. -n | uniq
```

The output is abbreviated to conserve space – there were 97 addresses produced. Representative addresses include:

```
226.185.101.183,  226.185.105.131,  226.185.106.176,  226.185.106.59
226.185.232.57,   226.185.234.179,  226.185.235.80,   226.185.236.106
226.185.7.94,     226.185.91.234,   226.185.9.222,    226.185.93.166
```

Step Eight. Perform sanity checks to make sure the above steps are valid. These commands produced no results. First, check to see if there is anything coming from the "0:3:e3:d9:26:c0" MAC address with the IP address that is believed to be the protected (home) network of 226.185.0.0 (either /16 or /24).

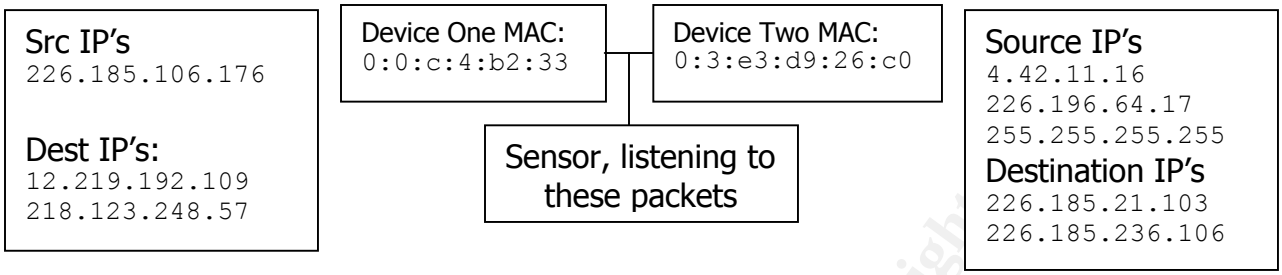
```
$ tcpdump -neqr 2002.4.30 ether src 0:3:e3:d9:26:c0 | cut -d ' ' -f 5 | grep
"^226\.185\."
```

Next, check to see if any combination of MAC addresses that could invalidate these steps (no results produced):

```
$ tcpdump -neqr 2002.4.30 ether src 0:3:e3:d9:26:c0 and ether dst not
0:0:c:4:b2:33
$ tcpdump -neqr 2002.4.30 ether src  0:0:c:4:b2:33  and ether dst not
0:3:e3:d9:26:c0
```

Revised Network Diagram

Below is a revised network diagram from Step Three with representative example IP addresses based on the steps performed and the information determined.



Note that the address range 226.185.0.0 to 226.185.255.255 did not appear as a destination IP for the "0:0:c:4:b2:33" with the single exception of one address. It is safe to conclude that the segment between the routers is on the 226.185.106.0/24 network, which explains the single source IP.

Detect Traces

The Snort alerts for this detect are:

```
[**] [1:618:2] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/30-03:49:22.644488 194.108.153.205:4750 -> 226.185.177.57:3128
TCP TTL:100 TOS:0x20 ID:16525 IpLen:20 DgmLen:48 DF
******S* Seq: 0xF2828EF0  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:618:2] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/30-03:49:25.894488 194.108.153.205:4750 -> 226.185.177.57:3128
TCP TTL:100 TOS:0x20 ID:16580 IpLen:20 DgmLen:48 DF
******S* Seq: 0xF2828EF0  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:618:2] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/30-03:49:32.454488 194.108.153.205:4750 -> 226.185.177.57:3128
TCP TTL:100 TOS:0x20 ID:16688 IpLen:20 DgmLen:48 DF
******S* Seq: 0xF2828EF0  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/30-03:49:47.584488 194.108.153.205:4911 -> 226.185.177.57:8080
TCP TTL:100 TOS:0x20 ID:16870 IpLen:20 DgmLen:48 DF
******S* Seq: 0xF325A8DD  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/30-03:49:50.834488 194.108.153.205:4911 -> 226.185.177.57:8080
TCP TTL:100 TOS:0x20 ID:16916 IpLen:20 DgmLen:48 DF
```

```
******S* Seq: 0xF325A8DD  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/30-03:49:57.394488 194.108.153.205:4911 -> 226.185.177.57:8080
TCP TTL:100 TOS:0x20 ID:17002 IpLen:20 DgmLen:48 DF
******S* Seq: 0xF325A8DD  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

Windump (tcpdump for Windows) output of traffic between these two systems:

```
23:49:22.644488 IP 194.108.153.205.4750 > 226.185.177.57.3128: S
4068642544:4068642544(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
0x0000       4520 0030 408d 4000 6406 565e c26c 99cd  E..0@.@.d.V^.l..
0x0010       e2b9 b139 128e 0c38 f282 8ef0 0000 0000  ...9...8........
0x0020       7002 4000 2329 0000 0204 05b4 0101 0402  p.@.#)..........

23:49:25.894488 IP 194.108.153.205.4750 > 226.185.177.57.3128: S
4068642544:4068642544(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
0x0000       4520 0030 40c4 4000 6406 5627 c26c 99cd  E..0@.@.d.V'.l..
0x0010       e2b9 b139 128e 0c38 f282 8ef0 0000 0000  ...9...8........
0x0020       7002 4000 2329 0000 0204 05b4 0101 0402  p.@.#)..........

23:49:32.454488 IP 194.108.153.205.4750 > 226.185.177.57.3128: S
4068642544:4068642544(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
0x0000       4520 0030 4130 4000 6406 55bb c26c 99cd  E..0A0@.d.U..l..
0x0010       e2b9 b139 128e 0c38 f282 8ef0 0000 0000  ...9...8........
0x0020       7002 4000 2329 0000 0204 05b4 0101 0402  p.@.#)..........

23:49:47.584488 IP 194.108.153.205.4911 > 226.185.177.57.8080: S
4079331549:4079331549(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
0x0000       4520 0030 41e6 4000 6406 5505 c26c 99cd  E..0A.@.d.U..l..
0x0010       e2b9 b139 132f 1f90 f325 a8dd 0000 0000  ...9./...%......
0x0020       7002 4000 f49f 0000 0204 05b4 0101 0402  p.@.............

23:49:50.834488 IP 194.108.153.205.4911 > 226.185.177.57.8080: S
4079331549:4079331549(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
0x0000       4520 0030 4214 4000 6406 54d7 c26c 99cd  E..0B.@.d.T..l..
0x0010       e2b9 b139 132f 1f90 f325 a8dd 0000 0000  ...9./...%......
0x0020       7002 4000 f49f 0000 0204 05b4 0101 0402  p.@.............

23:49:57.394488 IP 194.108.153.205.4911 > 226.185.177.57.8080: S
4079331549:4079331549(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
0x0000       4520 0030 426a 4000 6406 5481 c26c 99cd  E..0Bj@.d.T..l..
0x0010       e2b9 b139 132f 1f90 f325 a8dd 0000 0000  ...9./...%......
0x0020       7002 4000 f49f 0000 0204 05b4 0101 0402  p.@.............
```

This is a application specific scan – an attempt to locate a Squid proxy server. Looking one layer deeper into the data the scanner is attempting to target the same machine and is making a (possibly educated) guess that there may be more than one proxy product installed on the same system - or a proxy on a different, common port.

Exact dates do not correlate one-to-one in this data file and with the above output from Snort, Windump and file posting time. Specifically, the file name leads one to believe it contains data on April 30. The date stamp on the web site has "Tue Jun 4 04:12:40 2002". Snort reports 5/29 to 5/30 dates. Using Snort's "-U" option, converts time to UTC time, which would help normalize the data. The output generated from Snort and Windump do not agree with respect to timestamp, which would present a challenge if this evidence were ever used in open court in establishing legal venue[12]. One should be aware of these challenges. Further, the checksums do not match since the IP addresses are obfuscated for use in the educational/certification setting.

Detect Generated By

Running Snort Ver. 1.9.1 on the source tcpdump file generated the information presented here for this detect. The default Snort rule set was used. The specific command line used was:

```
c:\snort\snort -q -U -X -c snort.conf -h "226.185.106.0/24" -k none –r
c:\snort\practical\2002.4.30 -l c:\snort\practical\2002.4.30.log
```

Snort Options on this command line:

-q: Quiet mode, do not report statistics to the user
-U: Convert to UTC time
-X: Hex display output desired
-c: Use the specified configuration file
-h: Home network identifier (derived from running tcpdump, discussed above)
-k: Using "none " turns off checksum checking
-r: Use the specified input file for reading and analysis
-l: Use the fully qualified path for output files

The Snort rules that generated these detects is from the default "scan.rules" file that comes with Snort and is as follows:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 3128 (msg:"SCAN Squid Proxy attempt";
flags:S; classtype:attempted-recon; sid:618; rev:2;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 8080 (msg:"SCAN Proxy \(8080\)
attempt"; flags:S; classtype:attempted-recon; sid:620; rev:2;)
```

Both of these perform similar functions, so they will be explained together. The rule will produce an alert for any traffic from the external network destined for the internal network to a given port – here, 3128 (rule 1) or 8080 (rule 2). The TCP flags must be a SYN indicating an initial connection attempt.

Probability the Source Address was Spoofed

It is unlikely that the sources are spoofed addresses. Spoofed, here, means that the source address is modified to be an address other than the actual sending address. Note that with the proliferation of the Internet, it is conceivable that the attacker has a

[12] **Venue**: a legal term meaning "the place where the alleged events from a legal action", which is critical to establishing jurisdiction and authority of a given court of law.

compromised machine and could be using that machine to relay data back to their real machine.

In any event, this type of an attack is part one of a two-stage attack. The first part is to determine if there is something exploitable. The second part is to use the exploit. In order to either a) execute a denial of service or b) to use the Squid proxy as a method of hiding the attackers true address, the attacker would need to confirm proxy availability.

As an aside: In a recent interview posted to Slashdot.org[13], the author of nmap (Fyodor) reported that he really liked how Chinese students have used nmap to search for open proxies in order to get past the "Great China Firewall".

Description of Attack

This proxy scan attempts to determine the existence of a Squid proxy server running on the target host on port 3128, or a similar product on port 8080. Squid, available from http://www.squid-cache.org/, is an Open Source web proxy cache designed for Windows and Unix/Linux, and is delivered on at least RedHat 8.0[14] installation media. This type of software is ideal for organizations who wish to conserve bandwidth, improve website response time for commonly accessed sites, and/or have small pipelines to the Internet. According to the Squid online documentation it listens for HTTP requests from clients on port 3128[15] by default. It is also a common practice to run a proxy server on port 8080. An example is Microsoft's ISA Server[16] which listens on port 8080 by default.

Attack Mechanism

The packets in this detect are active reconnaissance for open proxy servers. Here, the absence of specific scans to other potential targets lends – although admittedly does not prove – that this system was under reconnaissance. If one can determine that the proxy is available and responding, then one can either a) use the proxy and hide their true IP address (and identity) or b) attempt one of the exploits referred to below.

As an example, CVE-2002-0068 indicates that an attacker can execute their own FTP commands – potentially allowing for system binaries to be replaced. This opens the door to things like system binary replacements or even Squid being replaced with a Trojan version. If the exploit discussed in CAN-2002-0715 can be achieved, then an attacker can learn user credentials of users authenticating to the proxy. If an attacker learns the naming scheme for user accounts they have valuable information needed to attempt penetration. If the machine at 226.185.177.57 were a reachable proxy server then an attacker would be able to use that proxy server for his or her own ends. Since Squid will proxy HTTP, HTTPS, and FTP protocols the server could be used to hide an attackers IP.

[13] Source URL: http://interviews.slashdot.org/interviews/03/05/30/1148235.shtml?tid=126&tid=172&tid=95
[14] There is a Squid initialization script in /etc/rd3.d/k25squid. Presence determined by the install type.
[15] See: http://squid.visolve.com/squid24s1/network.htm#http_port for further information.
[16] Supporting URL's:
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/isa/proddocs/isadocs/cmt_upgradprx2indep.asp and http://msdn.microsoft.com/library/default.asp?url=/library/en-us/isa/isaabout_3yum.asp

Correlations

The Common Vulnerabilities and Exploits (CVE) website hosted by MITRE Corp (cve.mitre.org), lists several entries relating to exploits in the Squid proxy server (Ver 2.4 and prior). They are[17]:

Name	Description
CVE-2001-0843	Squid proxy server 2.4 and earlier allows remote attackers to cause a denial of service (crash) via a mkdir-only FTP PUT request.
CVE-2002-0067	Squid 2.4 STABLE3 and earlier does not properly disable HTCP, even when "htcp_port 0" is specified in squid.conf, which could allow remote attackers to bypass intended access restrictions.
CVE-2002-0068	Squid 2.4 STABLE3 and earlier allows remote attackers to cause a denial of service (core dump) and possibly execute arbitrary code with an ftp:// URL with a larger number of special characters, which exceed the buffer when Squid URL-escapes the characters.
CVE-2002-0069	Memory leak in SNMP in Squid 2.4 STABLE3 and earlier allows remote attackers to cause a denial of service.
CVE-2002-0714	FTP proxy in Squid before 2.4.STABLE6 does not compare the IP addresses of control and data connections with the FTP server, which allows remote attackers to bypass firewall rules or spoof FTP server responses.
CAN-2002-0715	Vulnerability in Squid before 2.4.STABLE6 related to proxy authentication credentials may allow remote web sites to obtain the user's proxy login and password.

In examining the description of these entries, one can see that there is a variety of exploits possible. If the targeted system (226.185.177.57) had this software installed, and was vulnerable (not patched or running a more updated version), then any one or all of these exploits can be attempted.

A variety of other SANS GCIA candidates have reported on these particulars detects. None of them articulated a dual attack approach. These include:

Candidate	Date	Source document URL
Tony Adams	Apr 2000	http://www.giac.org/practical/Tony_Adams.doc
Mark Limesand	Sep 2000	http://www.giac.org/practical/Mark_Limesand.doc
Mike Poor	Nov 2001	http://www.giac.org/practical/Mike_Poor_GCIA.doc
Kevin Timm	Mar 2002	http://www.giac.org/practical/Kevin_Timm_GCIA.doc

Evidence of Active Targeting

The activity here indicates directed reconnaissance (open proxy scan), which normally occurs early on in the attack process. In order to use a proxy, one must reliably establish a connection with it by using a real address.

[17] The specific search URL is: http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=Squid+Proxy+attempt+

Depending on the intelligence (meaning gathered information) of the attacker, they likely have discerned that the target is likely to run Squid - perhaps by analyzing web server logs (requests) from an inside user. The attacker went one step farther and guessed that the system administrator configured a web proxy server at an alternate port. However – this is only one 24 hour period – so at best one can conclude that specific hosts on the inside network are being targeted.

The command '`windump -r 2002.4.30 -n -X "dst 226.185.177.57"`' revealed only six packets. The command '`windump -r 2002.4.30 -n "dst port 3128"`' only shows 17 distinct packets – 6 identified above, and the remaining 11 appear to be normal TCP/IP conversations from 226.185.106.176.62365 to/from 140.138.246.48.3128 (a different system). By examining the payloads, this secondary traffic is from the Gnutella file-sharing program. A representative packets of Gnutella is shown for completeness:

```
16:51:01.964488 IP 226.185.106.176.62365 > 140.138.246.48.3128: P
2402308671:2402308725(54) ack 2561862541 win 17520 (DF)
0x0000   4500 005e dd78 4000 7c06 c16c e2b9 6ab0        E..^.x@.|..l..j.
0x0010   8c8a f630 f39d 0c38 8f30 523f 98b2 eb8d        ...0...8.0R?....
0x0020   5018 4470 b078 0000 474e 5554 454c 4c41        P.Dp.x..GNUTELLA
0x0030   2043 4f4e 4e45 4354 2f30 2e36 0d0a 5573        .CONNECT/0.6..Us
0x0040   6572 2d41 6765 6e74 3a20 476e 7563 6c65        er-Agent:.Gnucle
0x0050   7573 2031 2e36 2e30 2e30 0d0a 0d0a             us.1.6.0.0....
```

The command '`windump -r 2002.4.30 -n "dst port 8080"`' shows 10 packets:

```
23:49:47.584488 IP 194.108.153.205.4911 > 226.185.177.57.8080: S
4079331549:4079331549(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
23:49:50.834488 IP 194.108.153.205.4911 > 226.185.177.57.8080: S
4079331549:4079331549(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
23:49:57.394488 IP 194.108.153.205.4911 > 226.185.177.57.8080: S
4079331549:4079331549(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
07:44:38.254488 IP 62.46.84.116.51861 > 226.185.106.59.8080: S
971328212:971328212(0) win 32768 <mss 1460,nop,wscale 0> (DF)
07:44:44.244488 IP 62.46.84.116.51861 > 226.185.106.59.8080: S
971328212:971328212(0) win 32768 <mss 1460,nop,wscale 0> (DF)
07:44:44.844488 IP 62.46.84.116.51862 > 226.185.106.59.8080: S
972660542:972660542(0) win 32768 <mss 1460,nop,wscale 0> (DF)
07:44:50.754488 IP 62.46.84.116.51862 > 226.185.106.59.8080: S
972660542:972660542(0) win 32768 <mss 1460,nop,wscale 0> (DF)
12:29:07.124488 IP 194.108.153.205.4990 > 226.185.232.57.8080: S
2869095998:2869095998(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
12:29:10.404488 IP 194.108.153.205.4990 > 226.185.232.57.8080: S
2869095998:2869095998(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
12:29:16.934488 IP 194.108.153.205.4990 > 226.185.232.57.8080: S
2869095998:2869095998(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
```

Note that the window size and TCP options are different for 194.108.153.205 and 62.46.84.116, indicating the machines are likely to have different TCP/IP protocol suites (and thus, be different operating systems). Also notice that there is a subsequent attempt from 194.108.153.205 on a different port number, about 12 hours later. The

attacker is attempting to be stealthy by allowing for a large time lag between attempted probes; or they might be checking to see if the server were down. This further confirms the probability that the address is not spoofed.

Severity

In order to assess the severity of the attack the following formula is applied:[18]

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

Criticality = 2. If this system were indeed functioning as a proxy server, then the system would be a valuable asset.

Lethality = 4. Squid (depending on version) is vulnerable to a DoS attack (e.g.: CVE-2001-0843) and to allow a web server to learn a proxy users credentials (CAN-2002-0715). A Trojaned version of Squid could be used to send users to a web server functioning as a proxy itself.

System Countermeasures = 4. There is no evidence to indicate that the server responded to the attempted connection.

Network Countermeasures = 4. There is no evidence that the perimeter defenses allowed traffic inside, and evidence (above) that indicates perimeter defenses stopped the traffic because of the retries.

The result is a –2 for Severity ((2+4) – (4+4)).

Defensive Recommendation

There are numerous ways to defend against this type of exploit. First, if possible, the outermost (or "choke") router should be configured to only accept inbound connections to information services that should be accessible from the Internet. Examples include web servers (HTTP on port 80), file transfer servers (FTP on port 20 and 21), and messaging servers (SMTP on port 25). Since a web proxy cache product like Squid is generally intended for internal users who are connecting outbound through the proxy, inbound connections to this service should not be permitted (blocked/closed by default) unless it is intended to be used by an outsider.

A firewall should be employed between the protected network and the Internet. This device should use NAT and should prevent inbound connections to protected resources unless necessary, and approved by the sites' management (a policy issue first).

[18] Source used: Stephen Northcutt, Judy Novak; "Network Intrusion Detection, Third Edition", © 2002 New Riders, pp 300 –306.

If at all possible, migrate away from using publicly routable IP addresses and use a private address scheme for internal hosts. Incorporate NAT into the site's network.

Should the targeted system actually be running Squid, then update to 2.5.

Lastly, if a site is using a proxy product like Squid or ISA server, use a nonstandard port – meaning do not use the documented port in the manufacturer's documentation.

Multiple choice Test Question and Answer

Question: You see SYN packets from the outside destined for an inside server to port 3128 or 8080 for a particular server. There are no responses for these packets, and the timing for the inbound packet follows a regular incrementing pattern. What does this indicate?

A. The DNS entries for the proxy server are incorrect.
B. This is an active attempt to probe for proxy server such as Squid.
C. This is a new ICMP Smurf variant attack.
D. A local user has traveled off of the network and their web browser is incorrectly configured.

Answer: B.

Explanation: DNS does not provide port numbers to a querying user; it provides hostname to IP address resolution. ICMP does not use ports. While a local user who traveled off to a different location might trigger traffic like this, the user would not produce both traces – they would produce one trace since it is unusual to configure two proxy servers on the same system at different ports (in at least the authors' experience). The traffic described is likely to be from an attacker.

Questions/Comments from the Intrusions list

Andrew Jones directed the author to X's paper on network configuration.
Q. Is it possible that the attacker knows nothing about the target system, and is merely trying a few different possible ports (Andrew Rucker Jones, 4/22/03)?
A. Certainly – however, the lack of evidence of other systems being targets of the same directed scan leads one to believe that there is a probability of directed targeting.

Q. For a simple proxy scan, is it really worth the effort to do prior reconnaissance (Andrew Rucker Jones, 4/22/03**)**?
A. The answer is based on attacker a) motivation, b) knowledge, c) skills, and d) evidence in logs. In short – yes – a stealthy attacker would want to minimize their footprint in a network. There are attackers that use automated techniques with little skill that can produce this signature. Looking at the CVE entries, if the attacker can gain information from a perimeter device that is supposed to have access to the Internet,

then it would allow them to glean information about the user community without penetrating further. In military terms, one might call this scouting.

Detect Two: IIS CMD.EXE Access Attempt

Source of Trace

The tcpdump binary files for this trace came from www.incidents.org website[19], specifically the "2002.4.30" log file.

This detect analyzes fifteen (15) occurrences of an attempt to exploit an IIS server by running **cmd.exe**. The first trace appears three times, and the second trace appears 12 times. Representative packets will be shown. The packets and alerts are related in content and purpose; both will be considered together.

Representative trace for group one, beginning with the Snort alert:

```
[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
05/30-09:41:35.814488 211.112.169.129:1592 -> 226.185.222.84:80
TCP TTL:104 TOS:0x0 ID:43867 IpLen:20 DgmLen:136 DF
***AP*** Seq: 0x11ED25F1 Ack: 0x93D0377E Win: 0x4470 TcpLen: 20
```

Using tcpdump, this representative packet detail shows:

```
tcpdump -nqX -r 2002.4.30 "src 211.112.169.129 and dst 226.185.222.84"

05:41:35.814488 211.112.169.129.1592 > 226.185.222.84.http: tcp 96 (DF)
0x0000   4500 0088 ab5b 4000 6806 9984 d370 a981        E....[@.h....p..
0x0010   e2b9 de54 0638 0050 11ed 25f1 93d0 377e        ...T.8.P..%...7~
0x0020   5018 4470 4b35 0000 4745 5420 2f73 6372        P.DpK5..GET./scr
0x0030   6970 7473 2f2e 2e25 3563 2e2e 2f77 696e        ipts/..%5c../win
0x0040   6e74 2f73 7973 7465 6d33 322f 636d 642e        nt/system32/cmd.
0x0050   6578 653f 2f63 2b64 6972 2072 2048  5454        exe?/c+dir.r.HTT
0x0060   502f 312e 300d 0a48 6f73 743a 2077 7777        P/1.0..Host:.www
0x0070   0d0a 436f 6e6e 6e65 6374 696f 6e3a 2063        ..Connnection:.c
0x0080   6c6f 7365 0d0a 0d0a                             lose....
```

This packet sequence appears three times, as follows:

```
tcpdump -tttt -nq -r 2002.4.30 "src 211.112.169.129 and dst 226.185.222.84"

05/30/2002 09:41:35.814488 211.112.169.129.1592 > 226.185.222.84.http: tcp 96
(DF)
05/30/2002 09:41:42.724488 211.112.169.129.1959 > 226.185.222.84.http: tcp 117
(DF)
05/30/2002 09:41:45.944488 211.112.169.129.1959 > 226.185.222.84.http: tcp 117
(DF)
```

Next, a check is made for response traffic. The following command returned no records, (which follows because the log data only contains packets that generated an alert):

```
tcpdump -tttt -nq -r 2002.4.30 "src 226.185.222.84 and dst 211.112.169.129"
```

[19] Actual URL: http://www.incidents.org/logs/Raw/

By using Ethereal, the data payload for the first packet can be seed. The text in bold shows the attempted UNICODE exploit, which may be effective against an unpatched Windows NT/2000 system running PWS/IIS[20]:

```
Hypertext Transfer Protocol
    GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir r HTTP/1.0\r\n
    Host: www\r\n
    Connnection: close\r\n
    \r\n
```

In a nutshell

Consolidating these techniques, the HTTP data payload of the three packets is:

```
05/30/2002 09:41:35.814488 211.112.169.129.1592 > 226.185.222.84.http:
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir r HTTP/1.0\r\n
    Host: www\r\n
    Connnection: close\r\n
    \r\n
05/30/2002 09:41:42.724488 211.112.169.129.1959 > 226.185.222.84.http:
    GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir c+dir
HTTP/1.0\r\n
    Host: www\r\n
    Connnection: close\r\n
    \r\n
05/30/2002 09:41:45.944488 211.112.169.129.1959 > 226.185.222.84.http:
    GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir c+dir
HTTP/1.0\r\n
    Host: www\r\n
    Connnection: close\r\n
    \r\n
```

Representative trace for group two, beginning with the Snort alert:

```
[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
05/30-18:02:46.004488 226.196.64.17:2012 -> 226.185.106.71:80
TCP TTL:108 TOS:0x0 ID:52525 IpLen:20 DgmLen:136 DF
***AP*** Seq: 0xCF477F1  Ack: 0xDD86B97F  Win: 0x2238  TcpLen: 20
```

Tcpdump output for the first packet in the group:

```
tcpdump -tttt -nXq -r 2002.4.30 "src 226.196.64.17 and dst 226.185.106.71"

05/30/2002 18:02:46.004488 226.196.64.17.2012 > 226.185.106.71.http: tcp 96
(DF)
0x0000   4500 0088 cd2d 4000 6c06 b14e e2c4 4011        E....-@.l..N..@.
0x0010   e2b9 6a47 07dc 0050 0cf4 77f1 dd86 b97f        ..jG...P..w.....
0x0020   5018 2238 90a6 0000 4745 5420 2f73 6372        P."8....GET./scr
0x0030   6970 7473 2f2e 2e25 3563 2e2e 2f77 696e        ipts/..%5c../win
0x0040   6e74 2f73 7973 7465 6d33 322f 636d 642e        nt/system32/cmd.
0x0050   6578 653f 2f63 2b64 6972 2072 2048 5454        exe?/c+dir.r.HTT
0x0060   502f 312e 300d 0a48 6f73 743a 2077 7777        P/1.0..Host:.www
0x0070   0d0a 436f 6e6e 6e65 6374 696f 6e3a 2063        ..Connnection:.c
```

[20] Personal Web Server / Internet Information Server.

```
0x0080   6c6f 7365 0d0a 0d0a                          lose....
```

This packet sequence appears twelve times, all within the same second, as follows[21]:

```
tcpdump -tttt -s n -r 2002.4.30 "src 226.196.64.17 and dst 226.185.106.71"

05/30/2002 18:02:46.004488 226.196.64.17.2012 > 226.185.106.71.http: P
217348081:217348177(96) ack 3716594047 win 8760 (DF)
05/30/2002 18:02:46.084488 226.196.64.17.2019 > 226.185.106.71.http: P
217348180:217348297(117) ack 3716594172 win 8760 (DF)
05/30/2002 18:02:46.124488 226.196.64.17.2019 > 226.185.106.71.http: P
217348297:217348413(116) ack 0 win 0 [tos 0x10]
05/30/2002 18:02:46.204488 226.196.64.17.2022 > 226.185.106.71.http: P
217348223:217348340(117) ack 3716594249 win 8760 (DF)
05/30/2002 18:02:46.294488 226.196.64.17.2027 > 226.185.106.71.http: P
217348281:217348426(145) ack 3716594313 win 8760 (DF)
05/30/2002 18:02:46.344488 226.196.64.17.2027 > 226.185.106.71.http: P
217348426:217348570(144) ack 3716594831 win 0 [tos 0x10]
05/30/2002 18:02:46.714488 226.196.64.17.2056 > 226.185.106.71.http: P
217348556:217348654(98) ack 3716594749 win 8760 (DF)
05/30/2002 18:02:46.794488 226.196.64.17.2061 > 226.185.106.71.http: P
217348620:217348716(96) ack 3716594885 win 8760 (DF)
05/30/2002 18:02:46.864488 226.196.64.17.2068 > 226.185.106.71.http: P
217348716:217348816(100) ack 3716594921 win 8760 (DF)
05/30/2002 18:02:46.904488 226.196.64.17.2068 > 226.185.106.71.http: P
217348816:217348915(99) ack 3716595439 win 0 [tos 0x10]
05/30/2002 18:02:46.944488 226.196.64.17.2073 > 226.185.106.71.http: P
217348781:217348877(96) ack 3716594969 win 8760 (DF)
05/30/2002 18:02:46.984488 226.196.64.17.2073 > 226.185.106.71.http: P
217348877:217348972(95) ack 0 win 0 [tos 0x10]
```

Note the sequence numbers and data sizes in bold – this indicates a conversation between the two machines, showing that the web server at least processed the requests.

Next, a check is made for response traffic. The following command returned no records:

```
tcpdump -tttt -nq -r 2002.4.30 "src 226.185.106.71 and dst 226.196.64.17"
```

The data payload for this packet can readily be seed (as above):

```
Hypertext Transfer Protocol
    GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir r HTTP/1.0\r\n
    Host: www\r\n
    Connnection: close\r\n
    \r\n
```

In a nutshell

The first three HTTP data payloads of the twelve packets are as follows:

```
05/30/2002 18:02:46.004488 226.196.64.17.2012 > 226.185.106.71.http:
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir r HTTP/1.0\r\n
    Host: www\r\n
```

[21] In having this detect reviewed by the incidents.org mail list, Andrew Jones encouraged an investigation of sequence numbers. The TCPDUMP trace was revised accordingly.

```
    Connnection: close\r\n
    \r\n
05/30/2002 18:02:46.084488 226.196.64.17.2019 > 226.185.106.71.http:
 GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir c+dir
HTTP/1.0\r\n
    Host: www\r\n
    Connnection: close\r\n
    \r\n
05/30/2002 18:02:46.124488 226.196.64.17.2019 > 226.185.106.71.http:
GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir c+dir
HTTP/1.0\r\n
    Host: www\r\n
    Connnection: close\r\n
    \r
. . .
```

Network Configuration

The network configuration is the same as used for Detect One, above.

Detect Generated By

Snort 1.9.1 default rules provided this detect. The specific Snort rule was:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS cmd.exe
access"; flow:to_server,established; content:"cmd.exe"; nocase; classtype:web-
application-attack; sid:1002;  rev:5;)
```

Assuming the variables are configured matching their names, this rule will produce an alert if any system on the external network attempts to connect to a web server on port 80 and if the phrase "cmd.exe" appears in the content to the server (not the reverse).

In an attempt to determine how many occurrences of this particular exploit were attempted, a specific rule was reapplied to the data file. This rule used was:

```
alert tcp any any -> 226.185.0.0/16 80 (msg:"WEB-IIS cmd.exe access";
flow:to_server,established; content:"cmd.exe"; nocase; )
```

There were fifteen occurrences of matching packets. Source address 211.112.169.129 had 3 occurrences and 226.196.64.17 had 12 occurrences.

Probability the Source Address was Spoofed

The address is legitimate (TCP conversation based on the **sequence numbers**), which means they were not spoofed. In order to take advantage of this particular exploit, the attacker would need to see the results of the HTTP request in order to see the results of the directory traversal command or a command that they attempted to execute.

Description of Attack

The attacker attempted to connect to the web server and send a "malformed" request to the web server that the server would misinterpret. When unpatched IIS receives specific

UNICODE characters it will actually traverse directories and run the file specified in the URL. The attempt was to run the default Windows command processor (cmd.exe). In the first group of attempted exploits, the attacker uses knowledge of the default Windows NT/2000 installation directory to run "cmd.exe".

From the second group, there is a pattern of these attempted exploits coming very close to one another. In this group, the attacker used different UNICODE characters and attempted to take advantage of default FrontPage Server Extensions being used on the system. The second group could also trigger a "FrontPage Exploit" alert from Snort depending on the order of rules in the configuration files. By examining CERT advisory CA-2001-26, referring to the System Footprint section, the HTTP GET requests nearly match signatures of Nimda.

Also for the second group, a reserved Class D multicast IP address as defined in RFC 3171[22] was used. Multicast addresses are from 224.0.0.0 to 239.255.255.255, with the address range 225.0.0.0 - 231.255.255.255 currently being reserved and the RFC states they should not be used. Therefore the attacker would be attempting to hide their source by using routable traffic that cannot be traced back to a single machine.

Attack Mechanism

According to CERT Advisory CA-2001-26[23], this worm can be delivered through several vectors including email with a MIME attachment of type "multipart/alternative" with two attachments in an attempt to fool the reader. The first is an empty "text/html" – giving the appearance that the email is blank, or empty. The second part of the message is a misidentified Base-64 encoded attachment that is executable on a Windows system (depending on the email client). Another vector is browser propagation – the worm already having affected pages on a web server, an unsuspecting site visitor can activate the worm that attempts to deliver executables to the victim. In some cases, the victim runs the code. A third vector, the worm may also propagate itself through the file system and create infected email files.

There is yet another vector. A client can send requests to an IIS server and attempt to take advantage of the directory traversal vulnerability in unpatched IIS 4/5. In order to take advantage of this exploit under IIS, an attacker (person or code) sends a HTTP GET request and types in a variety of UNICODE characters in the URI. The ".." in the request identified above is an attempt to navigate above the default IIS directory, which is installed by default to "c:\inetpub\wwwroot". The attackers in both groups of packets attempt to take advantage default Windows NT/200 configuration. The second group attacks attempts to take advantage of software (FrontPage Server Extensions) that are likely to be installed on a system.

[22] Source URL: http://www.ietf.org/rfc/rfc3171.txt?number=3171
[23] Source URL http://www.cert.org/advisories/CA-2001-26.html

This operating system specific and web server specific attack is intended to give the attacker command level execution. An attacker would want to know the results and then perform specific tasks against the server – such as running the default TFTP server to bring on their own executable (root kit?).

Correlations

This exploit is documented on several security specific websites. Examples include:

- Miter's CVE entry is CVE-2000-0884[24].
- Microsoft Security bulletin ID: MS00-078[25].
- CERT: The CERT advisory CA-2001-26 discusses the Nimda worm in depth.
- IDC: In the whitepaper titled "Emerging Threats to the Employee Computing Environment[26]", the authors of WebSense explain how blended viruses/worms like Nimda use automated, high-speed techniques to attack indiscriminately.
- GFI has an excellent write up on the Nimda worm[27].

A few SANS GCIA candidates have discussed this vulnerability:

Candidate	Date	Document URL
Mike Poor	Mar 2002	http://www.giac.org/practical/Mike_Poor_GCIA.doc
Thomas M. Rodriguez	Oct 2001	http://www.giac.org/practical/Thomas_Rodriguez_GCIA.doc

Evidence of Active Targeting

These traces are from Nimda or a variant. Therefore, the system is not likely to be under direct attack by an individual – rather automated attack by the worm. According to the Internet Storm Center (www.incidents.org), there are a few IP's known to attack from the network 211.112.169.0/24 (.155 and .240) recently[28]. The source network is Korean, and lately has been the source of numerous attacks.

The second group is a Nimda like worm (see Correlations for more information). This worm is highly automated, and attempts to "seek and destroy" as quickly as it can and can be delivered by multiple vectors. Nimda is an example of a "blended threat". The particular server is not likely to be an active target from a given individual – it is much more likely that an infected server just happened to hit this specific machine during the time period the packet capture was conducted.

Note that the source IP for group two (226.196.64.17) is an IP multicast address in a reserved range; this is an attempt to hide the source.

[24] Source URL: http://cve.mitre.org/cve/downloads/full-cve.html
[25] Source URL: http://www.microsoft.com/technet/treeview/default.asp?url= /technet/security/bulletin/ms00-078.asp
[26] Note that WebSense sponsored this paper. The full citation is: "Emerging Threats to the Employee Computing Environment: Expanding EIM Beyond the Browser An IDC White Paper", Sponsored by Websense Analysts: Brian Burke, Chris Christiansen, and Charles Kolodgy. 5 Speen Street, Framingham, MA 01701 USA (from www.idc.com). Websense URL: http://www.websense.com/products/resources/wp/emergingthreats_idc.pdf
[27] Available from URL: http://www.gfi.com/press/nimdaworm.htm
[28] Source URL: http://isc.sans.org/source_report.html?order=&subnet=211.112.169

Severity

In order to assess the severity of the attack the following formula is applied:[29]

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

Group One Target: 226.185.222.84

Note: For the purposes of discussion and grading, the second group should be used. The score would be the same for both groups.

Group Two Target: 226.185.106.71

Criticality = 4. Following the network layout in Detect One, the network segment between the outer router and the inner router is 226.185.106.0/24. Since the IP for the system in question is on this network but there is no corresponding MAC address, this must be a publicly available web server whose traffic is forwarded inside - this system is an accessible web server (given the TCP sequence numbers). Publicly exposed web servers generally support the sites outward facing presence - an important asset for their public image.

Lethality = 4. The specific attacks here are not "lethal" – running a command interpreter and seeing a directory is not "lethal" to the system. However, the very nature of this exploit – what can be done with a system program - is clearly "lethal". Nimda also attacks indiscriminately - it is an equal opportunity worm, and will attack inside systems as well as outside.

System Countermeasures = 3. As shown in the TCP sequence numbers, the system did have a proper conversation – therefore the system did actually evaluate every request and may have become vulnerable (we just don't know – there are no response packets, no web server logs, and no incident report).

Network Countermeasures = 1. The traffic was allowed in to the network, where this server is located and not filtered by the outermost device.

The result is a 4 for Severity: ((4+4) – (3+1)) = 4.

Defensive Recommendation

Microsoft has updated Windows with Hot fixes and Service Packs that can address this issue. Microsoft has also provided the IIS Lockdown tool that will dramatically improve IIS security. This tool should be used to initially secure an IIS server before it is

[29] Source used: Stephen Northcutt, Judy Novak; "Network Intrusion Detection, Third Edition", © 2002 New Riders, pp 300 –306.

deployed in production. Microsoft also has provided a URL Scan tool that is an ISAPI filter. This tool will check the validity of URL's passed into IIS and block ones with UNICODE characters. These tools are discussed in Microsoft support article 325864[30]. If there is a need to run scripts under IIS, then make sure that the correct permissions are applied to the directories (web server execution set to "scripts only") and NTFS permissions applied correctly for the environment.

One recommendation that may seem odd on the surface is to install Windows NT/200 and IIS in a non-default path. Microsoft's IIS server cannot be installed into a different directory or on a different drive unless the unattended installation options are used when installing the base operating system. Microsoft support article 259671[31] discusses how to install IIS components to a non-default path and drive letter during unattended setup. IIS should be installed into a non-default path and drive letter if at all possible.

If at all possible, a content sensitive filtering firewall product that can interrogate the HTTP data stream for UNICODE characters should be used. There are several products that are available. One is a combination of CheckPoint VPN1 and WebSense. Another is Microsoft ISA Server, which has content filtering built into the firewall itself. Multicast addresses should be blocked at the outermost point on the network if possible, as per CERT guidelines[32].

Multiple choice Test Question and Answer

Question: You want to check your IIS server logs and determine if the system has been attacked with the Nimda worm lately. What type of requests would you look for in the IIS logs?

A. HTTP POST requests for "admin.dll".
B. HTTP GET requests for cmd.exe.
C. HTTP PUT requests for cmd.exe.
D. HTTP GET requests with odd "%" characters and attempts to run the system shell, "cmd.exe".

Answer: D.

Explanation: Although B is close; the better answer is D since the Nimda worm uses HTTP GET requests as follows:

```
GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir c+dir
HTTP/1.0\r\n
    Host: www\r\n
    Connnection: close\r\n
    \r\n
```

[30] Support article URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;325864
[31] Support article URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;259671
[32] Source URL: http://www.cert.org/security-improvement/practices/p082.html

Questions/Comments from the Intrusions list

(all from Andrew Jones in 4/03 and 5/03):
Q: Who are the attackers? Are they defined in a known database?
A. The specific IP is not listed at www.incidents.org - however, IP's from the same Class C network are[33] listed.

Q: Can more analysis be done to attempt to determine if the targets are Microsoft or another operating system?
A. Yes, one can make a very good guess at the operating system. One can examine the specific packets in question and compare IP and TCP options to lists of common characteristics for various known operating systems. One such list is maintained at honeynet.org[34]. Checking TTL, Window, DF, and TOS the target matches Windows 2000.

Q. What are those characters [used to exploit IIS]?
A. Various Unicode characters will do – these include "%5C", "%2F", and "%35C" are examples.

Detect Three – UPNP Coming At Me (WinXP ICS)

Note: This detect is from a production network the author routinely works with; organizational policy and network configuration was changed as a result of this analysis. Email with Jeffrey Schlimmer of Microsoft confirms that the traffic identified here is Windows XP Internet Connection Sharing.

Source of Trace

Several of these packets were observed May 6, 2003 (a representative day within the last month). The Snort sensor, running on a server in the perimeter network detected these packets. On this particular day, there were 159,657alerts of which 11,142 are for this detect. This machine is between the outermost uplink router to the Internet and the main enterprise firewall/router. A representative alert is shown below:

```
[**] [1:1917:3] SCAN UPNP service discover attempt [**]
[Classification: Detection of a Network Scan] [Priority: 3]
05/03-00:00:19.592437 MY.LOCAL.NET.248:4904 -> 24.15.190.1:1900
UDP TTL:125 TOS:0x0 ID:18789 IpLen:20 DgmLen:161 Len: 141
```

"MY.LOCAL.NET" represents the production network (addresses are changed to prevent disclosure). This as interesting in and of itself, so IP addresses for these packets were produced with tcpdump and representative packets were collected:

```
MY.LOCAL.NET.248, MY.LOCAL.NET.182, 68.70.164.85,     24.15.190.1
172.128.60.10,    172.168.90.37,    65.177.81.125,    67.201.78.5
67.30.233.124,    67.30.246.9,      68.10.248.252,    68.35.69.96
```

[33] Source URL: http://isc.sans.org/source_report.html?order=&subnet=211.112.169
[34] Source URL: http://project.honeynet.org/papers/finger/traces.txt

The command '`tcpdump -r 2003.05.04.snort-pcap.log.1051934402 -n -X -vvv "src MY.LOCAL.NET.248 and dst 24.15.190.1 and port 1900"`' showed:

```
00:00:19.592437 MY.LOCAL.NET.248.4904 > 24.15.190.1.1900:  [udp sum ok] udp 133
(ttl 125, id 18789, len 161)
0x0000   4500 00a1 4965 0000 7d11 908c xxxx xxxx        E...Ie..}.......
0x0010   180f be01 1328 076c 008d b646 4d2d 5345        .....(.l...FM-SE
0x0020   4152 4348 202a 2048 5454 502f 312e 310d        ARCH.*.HTTP/1.1.
0x0030   0a48 4f53 543a 2032 3339 2e32 3535 2e32        .HOST:.239.255.2
0x0040   3535 2e32 3530 3a31 3930 300d 0a4d 414e        55.250:1900..MAN
0x0050   3a20 2273 7364 703a 6469 7363 6f76 6572        :."ssdp:discover
0x0060   220d 0a4d 583a 2033 0d0a 5354 3a20 7572        "..MX:.3..ST:.ur
0x0070   6e3a 7363 6865 6d61 732d 7570 6e70 2d6f        n:schemas-upnp-o
0x0080   7267 3a73 6572 7669 6365 3a57 414e 5050        rg:service:WANPP
0x0090   5043 6f6e 6e65 6374 696f 6e3a 310d 0a0d        PConnection:1...
0x00a0   0a                                             .
...
```

Using Ethereal, the relevant data portion from this packet is:

```
Hypertext Transfer Protocol
    M-SEARCH * HTTP/1.1\r\n
    HOST: 239.255.255.250:1900\r\n
    MAN: "ssdp:discover"\r\n
    MX: 3\r\n
    ST: urn:schemas-upnp-org:service:WANPPPConnection:1\r\n
    \r\n
```

Detect Generated By:

There is an instance of Snort running on a sensor in the perimeter network. This sensor listens using promiscuous mode, analyzing every packet between the Enterprise network and the Internet. Specifically, the rule that produced this packet is:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 1900 (msg:"SCAN UPNP service discover
attempt"; content:"M-SEARCH "; offset:0; depth:9; content:"ssdp\:discover";
classtype:network-scan; sid:1917; rev:3;)
```

This rule means normally means that any external network will cause the alert to fire if the packet has these characteristics: it must be UDP, directed to port 1900, contain the text "M-SEARH" and "ssdp:discover" early in the data portion of the packet. This rule fired for both inbound and outbound packets because the variable $EXTERNAL_NET and $HOME are defined as "any" - the site tracks inbound and outbound alerts (site policy).

The recent rule bases for Snort include three UPNP rules in order to narrow down specific types of UPNP traffic. Also, the UPNP specification allows for traffic to be on UDP port 1900 and 5000; this rule is by no means inclusive for this type of traffic.

Probability the Address is Spoofed

Investigation and correlation of addresses from MY.LOCAL.NET proved they are genuine addresses. The nature of UPNP traffic is stimulus and response. In order for UPNP to function for network device discovery, traffic must actually go to and from live hosts.

Description of an Attack:

First, consider this information on the data in order to understand how it can be used against the site. According to the UPNP documentation for the Internet Gateway Specification from www.upnp.org[35], the service type identified in the data payload corresponds to a UPNP Internet Gateway Device. An interpretation of the HTTPMU (HTTP via Multicast) can be found in the proposed specification document at ww.upnp.org[36]. The header for the data in the packets indicates that the system in question is searching using multicast for any host (the "*" in the M-SEARCH option) that will offer a PPP connection (the "ST" option). The address space 239.255.255.250 is the local administrative domain that SSDP uses according to the draft SSDP document[37]. If a client has such a connection, it will respond with a random delay of "0 to MX in seconds", which will prevent a flood of replies. Assuming that there is a system at the other end of this conversation (24.15.190.1), it would see this properly formatted request and answer.

Attack Mechanism:

The best explanations of attacks are taken directly from the Microsoft Security bulletin, MS01-059[38] states:

> "An attacker could send a NOTIFY directive to a UPnP-capable computer, specifying that the device description should be downloaded from a particular port on a particular server. If the server was configured to simply echo the download requests back to the UPnP service (e.g., by having the echo service running on the port that the computer was directed to), the computer could be made to enter an endless download cycle that could consume some or all of the systems availability.
>
> An attacker could craft and send this directive to a victim's machine directly, by using the machine's IP address. Or, he could send this same directive to a broadcast and multicast domain and attack all affected machines within earshot, consuming some or all of those systems' availability.
>
> An attacker could specify a third-party server as the host for the device description in the NOTIFY directive. If enough machines responded to the directive, it could have the effect of flooding the third-party server with bogus requests, in a distributed denial of service attack. As with the first scenario, an attacker could either send the directives to the victim directly, or to a broadcast or multicast domain."

[35] Ulhas Warrier and Prakash Iyer of Intel Corporation along with Frédéric Pennerath and Gert Marynissen of AlcatelSource have written the "WANPPPConnection:1 Service Template Version 1.01" which defines the discussed UPNP standard. Source URL: http://www.upnp.org/standardizeddcps/igd.asp

[36] Yaron Y. Goland of CrossGain and Jeffrey C. Schlimmer of Microsoft: "Multicast and Unicast UDP HTTP Messages", which is currently in draft status with the UPNP committee. Source URL: http://www.upnp.org/download/draft-goland-http-udp-04.txt

[37] Yaron Y. Goland and others from Microsoft Corporation along with Shivaun Albright from HP, "Simple Service Discovery Protocol/1.0", Source URL: http://www.upnp.org/download/draft_cai_ssdp_v1_03.txt

[38] Source URL: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-059.asp

As stated above in the excerpt from the Microsoft bulletin, an attacker can cause a buffer overflow and/or a Denial of Service on the target. Here, the specific senders are sending out "ssdp:discovery" packets which announce their presence. Anyone listening between the sender and the receiver can learn that the sender is vulnerable; and that the sender would respond to a discovery attempt from an attacker. The attacker could send malformed requests to the sender and redirect it to a malicious server using example source code provided in Chip Calhoun's GCIH practical[39] (single packet DoS).

This particular trace represents "discovery", and on a production network is should be sued to prevent a possible exploit by using this trace to locate and configure the client. If there were actual attacks going on, Snort would report "MISC UPNP malformed advertisement" and "MISC UPNP Location overflow" alerts (these alerts are contained in the misc.rules file).

The other attack type (buffer overflow), allows the attacker to send arbitrary code to the victim. The observed behavior of a victim computer is quite varied. Examples include remote installation of a root kit, disabling anti viral software, or depositing malicious script code on the system.

Correlations:

SANS GCIH candidate Chip Calhoun has an excellent write up and demonstration of this exploit. His GCIH practical includes a general discussion, source code for the exploit, and the complete incident handling process discussion.

The Microsoft TechNet Security Bulletin MS01-059. Further, Microsoft Knowledge Base articles Q314757, Q314941, Q315000 and Q315056 discuss these vulnerabilities and are accessible from the http://upport.nicrosoft.com/ website.

Extensive information about UPNP in general is available at the primary UPNP website, http://www.upnp.org/.

The CVE has these entries[40] about the UPNP service:

Name	Description
CVE-2001-0876 [41]	Buffer overflow in Universal Plug and Play (UPnP) on Windows 98, 98SE, ME, and XP allows remote attackers to execute arbitrary code via a NOTIFY directive with a long Location URL.
CVE-2001-0877 [42]	Universal Plug and Play (UPnP) on Windows 98, 98SE, ME, and XP allows remote attackers to cause a denial of service via (1) a spoofed SSDP advertisement that causes the client to connect to a service on another

[39] Source URL: http://www.giac.org/practical/Chip_Calhoun_GCIH.doc
[40] Source URL: http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=upnp
[41] Source URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0876
[42] Source URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0877

	machine that generates a large amount of traffic (e.g., chargen), or (2) via a spoofed SSDP announcement to broadcast or multicast addresses, which could cause all UPnP clients to send traffic to a single target system.
CAN-2001-0721 [43]	Universal Plug and Play (UPnP) in Windows 98, 98SE, ME, and XP allows remote attackers to cause a denial of service (memory consumption or crash) via a malformed UPnP request.

Evidence of Active Targeting:

With over 11,000 packets in one day from the source to the destination listed above (MY.LOCAL.NET.248:4904 -> 24.15.190.1:1900), it is highly likely that one of the systems on the organizations' network has some sort of software installed that warrants investigation. Further, there are numerous copies of Windows XP installed on the network; there are only a few source IP's with this type of traffic exiting the network. The command used to determine these metrics is:

```
tcpdump -r 2003.04.26.snort-pcap.log.1051243201 "udp and src MY.LOCAL.NET.248
and dst port 1900" | wc -l
```

There are also eleven (11) distinct IP's sending this type of traffic into the network - the single suspect address accounted for more than 99% of the traffic. The other addresses from inbound UPNP traffic are not likely be spoofed; rather, they are more likely to be misconfigured PC's.

Severity

Note: Microsoft has rated this issue "critical" for Windows XP systems.
In order to assess the severity of the attack the following formula is applied:[44]

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

This event scores a 0, as outlined below (0 = (1+2)-(1+2)).

Criticality (1): UPNP vulnerabilities are directed against desktop systems, which are not normally critical to a network's infrastructure. It was also found that this system was not a "production asset".

Lethality (2): UPNP malicious and malformed traffic can result in arbitrary code being executed on a target (CVE-2001-0876). Since a Windows XP system is most likely to be connected to either a NetWare NDS based network or a Microsoft Windows NT/2000 domain at an enterprise network, a victim system can be used to further damage a network. Lethality is rated above 1 because the source is reliably advertising.

[43] Source URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0721
[44] Source used: Stephen Northcutt, Judy Novak; "Network Intrusion Detection, Third Edition", © 2002 New Riders, pp 300 –306.

System Countermeasures (1): Given the amount of traffic (>11,000 packets), there is enough information to warrant an investigation and to suspect the victim system. The system is known by the site to be an end user, non-critical asset. The antithesis of a defense mechanism was in place here - the service was on and communicating.

Network Countermeasures (2): Traffic was allowed out of the source network. The firewall did not block inbound/outbound UPNP. (Note: This issue no longer exists…).

Defensive Recommendation:

The current recommendation from Microsoft and others is to block the UPNP ports 1900/UDP and 5000/UDP. For this network, the recommendation is to block packets from exiting and entering the network at the outermost router. Next, this type of traffic needs to be blocked from traveling through the enterprise (entire site) network. If users start installing UPnP devices on the network and widespread discovery were possible, one user elsewhere on the network could accidentally turn off a refrigerator or increase the volume on a television (this analogy follows the purpose of UPNP).

When end user systems are installed, the "SSDP Discovery Service" should be disabled on behalf of the user. Microsoft provides a downloadable patch for this purpose.

Multiple Choice Question and Answer

Question: Your intrusion detection system reports excessive traffic to a variety of systems at using source port 1900/UDP. You examine this data and see "ssdp:discovery" and what appears to the a multicast web request. What is this traffic indicative of?

A. Nothing; it is most likely users browsing Internet sites.
B. You have a system that is advertising UPNP and may become compromised with a UPNP vulnerability
C. Nothing; UDP 1900 traffic is normal SMB broadcast queries.
D. You have a system that is transmitting Video on Demand with NetMeeting, and therefore this is a false positive.

Answer: B.

Explanation: More than likely, you have a system that has had some sort of UPNP vulnerability exploited, or is misconfigured in some way. UDP 1900 traffic is indicative of the UPNP network discovery protocol. It extremely unlikely that systems would all use source ports of 1900 for web browsing; further, they would be using TCP. SMB and NetMeeting traffic is carried on different ports.

Assignment Three: Analyze This!

Assumptions

Raw data files are provided without a network map. Therefore one must make some educated guesses about the structure of the network. First, following industry guidelines and best practices it is assumed that the Snort sensor is placed between the Internet uplink router and the site firewall. Second, it is assumed that a recent version of Snort (1.9 or 1.9.1) is in use; therefore alerts should match the recent rules posted on the Snort web site when the name of the rule is comparable. Last – the monitored network is labeled "MY.NET." in the source data.

> For the purposes of analysis by various tools, "MY.NET." was renumbered to "123.123." (This address did not appear in the original source data).

Executive Summary

This analysis was conducted at the request of GIAC University. Provided were fifteen log files - one was out of synch with the others (explained below) so an additional file was requested. It is understood that the Snort Intrusion Detection System rule base is "fairly standard"; meaning that alerts and behaviors of the IDS should be similar to the currently released version of Snort. The University has requested that compromised systems be identified if possible, and that correlations be made of internal systems with external systems. This deliverable includes:

- A description of the data files and a discussion of how damaged they were.
- An executive summary (this section).
- Alerts logically arranged by descending frequency, for alerts that can be understood.
- Top 10 "talkers" - systems occurring most frequently in the provided data.
- Highlighted external source addresses, Internet lookup information, and a justification why they were chosen.
- Correlations with other analysts from the www.giac.org web site.
- A link graph data that demonstrates a specific issue - file sharing over Internet Relay Chat (IRC) - of which the University should be aware.
- Recommendations on how network security can be improved.
- A description of the process of how the data was analyzed and the tools used.

This deliverable satisfies each of these requests. Of the issues identified, one recurring pattern appeared - questionable message/file traffic using IRC. The University is monitoring IRC application usage that is being used for file sharing. Given the recent DMCA legislation and the pressure brought on Internet Service Providers and Universities by the RIAA and the MPAA, upper management at GIAC University should be aware of this traffic and the trend that it indicates. IRC has become much more than real time text message exchange. Particularly, the "XDCC" file sharing protocol is often used for questionable file sharing - current research indicates Universities are often the source of copyrighted media files being shared on IRC - against the expressed wishes of the copyright holder.

Data (files) Analyzed

The following data files were downloaded from http://www.incidents.org/logs:

Alert	Out of Spec	Scans
alert.030501.gz	OOS_Report_2003_05_01_31055	scans.030501.gz
alert.030502.gz	OOS_Report_2003_05_02_28431	scans.030502.gz
alert.030503.gz	OOS_Report_2003_05_03_7239	scans.030503.gz
alert.030504.gz	OOS_Report_2003_05_04_21395	scans.030504.gz
alert.030505.gz	OOS_Report_2003_05_05_25821	scans.030505.gz
	OOS_Report_2003_05_06_4938	

These logs represent five consecutive days worth of data from GIAC University. The "alert" log files contain both alerts and port scans status records. Many of the records in the alert files were incomplete - as if the disk drive could not keep up with the IDS. Code needed to be written that "repaired" the files by removing records that did not contain enough correct information to be a properly formatted record from the IDS. The IDS should be checked to make sure it has sufficient capacity for the task at hand. There we 901,622 alerts total.

The 'scans' files contain detailed information about port scans directed against the network.

The Out of Spec (OOS) files contain packets that had some sort of anomaly that the IDS detected. The data in the OOS files indicated it was collected on the day before – the 2003_05_01 file contained events of interest all dated 04/31 with the exception of the last entry. Therefore the 05_02 to 05_06 files were used. It is assumed that there was some problem with the rollover script, or that these files were generated a day later by post analysis of the binary captured data from Snort.

Much of the alert data was improperly formatted. On a file-by-file basis, the analysis showed:

- alert.030501 had 69648 lines of data with 32527 lines being improperly formatted (46.7%).
- alert.030502 had 141594 lines of data with 55068 lines being improperly formatted (38.8%).
- alert.030503 had 530515 with 58218 being improperly formatted (10.9%).
- alert.030504 had 399037 with 48738 being improperly formatted (12.2%).
- alert.030505 had 136937 with 41106 being improperly formatted (30%)

Consolidated Alerts by Frequency (Statistics, Part One)

On the next few pages is the consolidated list of attacks. This data is arranged highest occurrence to lowest. Following these tables is detailed analysis of the specific alerts with the top five source and destination IP's encountered.

The peak alert time during the period covered was 5/3 from 11AM to 12PM, with a total of 213,309 alerts. Therefore, 23% of alerts were in this single hour.

Presentation in this format is based on some of the best GIAC papers - Honors papers by Hee So and Les Gordon.

Attack / Alert	Count	Ext -> Ext	Ext -> Int	Int -> Int	Int -> Ext	Uniq Ext Src	Uniq Int Src	Uniq Int Dst	Uniq Ext Dst
Incomplete Packet Fragments Discarded	355297	0	462	1	354834	99	2	66	6
TCP SRC and DST outside network	208305	208305	0	0	0	198915	0	0	2196
SMB Name Wildcard	174119	16	174103	0	0	22474	0	40910	2
spp_http_decode: IIS Unicode attack detected	30427	0	673	0	29754	275	528	174	705
High port 65535 udp - possible Red Worm - traffic	27260	0	12121	0	15139	163	78	109	222
CS WEBSERVER - external web traffic	24935	3	24932	0	0	5319	0	1	1
High port 65535 tcp - possible Red Worm - traffic	23632	0	11822	0	11810	83	64	67	92
Tiny Fragments - Possible Hostile Activity	13532	2	4369	0	9161	21	1	21	900
TFTP - Internal TCP connection to external tftp server	9337	0	4527	0	4810	33	11	12	31
EXPLOIT x86 NOOP	6019	0	6019	0	0	168	0	147	0
connect to 515 from outside	5033	0	5033	0	0	3	0	4873	0
[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC	5023	0	0	0	5023	0	12	0	21
spp_http_decode: CGI Null Byte attack detected	5020	0	69	0	4951	35	119	6	127
Null scan!	2474	1	2473	0	0	115	0	109	1
Queso fingerprint	1577	1	1576	0	0	328	0	123	1
[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.	1562	0	1562	0	0	74	0	63	0
123.123.30.4 activity	1343	0	1343	0	0	294	0	1	0
Possible trojan server activity	921	0	351	0	570	48	21	179	24
123.123.30.3 activity	804	0	804	0	0	44	0	1	0
CS WEBSERVER - external ftp traffic	781	0	781	0	0	147	0	1	0
[UMBC NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC	746	0	0	0	746	0	3	0	11
IDS552/web-iis_IIS ISAPI Overflow ida nosize	717	0	717	0	0	455	0	580	0
SUNRPC highport access!	520	0	520	0	0	30	0	23	0
TFTP - Internal UDP connection to external tftp server	395	0	38	0	357	14	31	19	33
[UMBC NIDS IRC Alert] User joining Warez channel detected. Possible XDCC bot	271	0	271	0	0	9	0	5	0

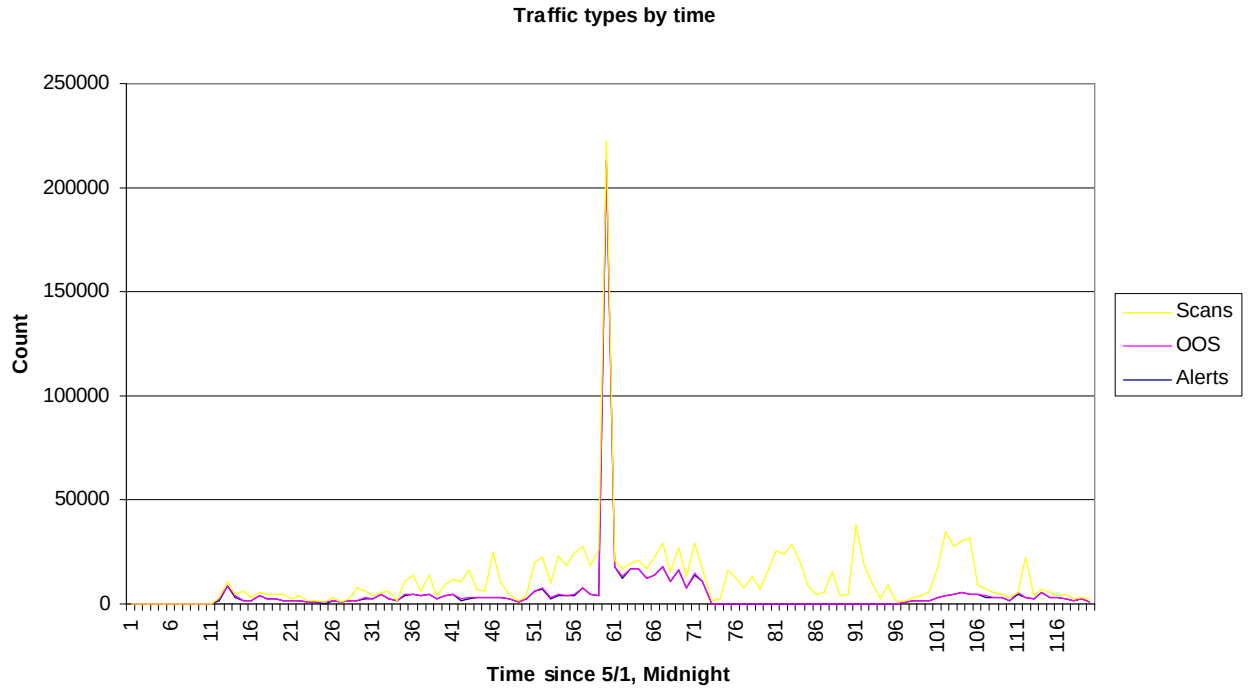
Attack / Alert	Count	Ext -> Ext	Ext -> Int	Int -> Int	Int -> Ext	Uniq Ext Src	Uniq Int Src	Uniq Int Dst	Uniq Ext Dst
[UMBC NIDS IRC Alert] Possible Incoming XDCC Send Request Detected.	194	0	194	0	0	11	0	11	0
IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize	188	0	0	0	188	0	2	0	167
IRC evil - running XDCC	168	0	0	0	168	0	15	0	14
External RPC call	149	0	149	0	0	4	0	149	0
[UMBC NIDS IRC Alert] User joining XDCC channel detected. Possible XDCC bot	149	0	149	0	0	6	0	5	0
NMAP TCP ping!	145	0	145	0	0	41	0	60	0
EXPLOIT x86 setuid 0	128	0	128	0	0	118	0	106	0
SNMP public access	98	0	98	0	0	5	0	9	0
NIMDA - Attempt to execute cmd from campus host	60	0	0	0	60	0	2	0	58
EXPLOIT x86 setgid 0	53	0	53	0	0	50	0	48	0
EXPLOIT x86 stealth noop	51	0	51	0	0	12	0	6	0
TCP SMTP Source Port traffic	34	0	34	0	0	3	0	12	0
Back Orifice	26	0	26	0	0	2	0	26	0
Notify Brian B. 3.54 tcp	26	0	26	0	0	21	0	1	0
Notify Brian B. 3.56 tcp	22	0	22	0	0	20	0	1	0
SMB C access	13	0	13	0	0	11	0	9	0
Probable NMAP fingerprint attempt	12	0	12	0	0	8	0	9	0
Attempted Sun RPC high port access	10	0	10	0	0	3	0	3	0
RFB - Possible WinVNC - 010708-1	8	0	4	0	4	4	4	4	4
FTP passwd attempt	7	0	7	0	0	3	0	2	0
[UMBC NIDS IRC Alert] K:line'd user detected, possible trojan.	7	0	7	0	0	6	0	4	0
TFTP - External UDP connection to internal tftp server	6	0	5	0	1	4	1	4	1
DDOS shaft client to handler	4	0	4	0	0	2	0	2	0
NIMDA - Attempt to execute root from campus host	3	0	0	0	3	0	1	0	3
TFTP - External TCP connection to internal tftp server	3	0	2	0	1	2	1	2	1
EXPLOIT x86 NOPS	2	0	2	0	0	1	0	1	0

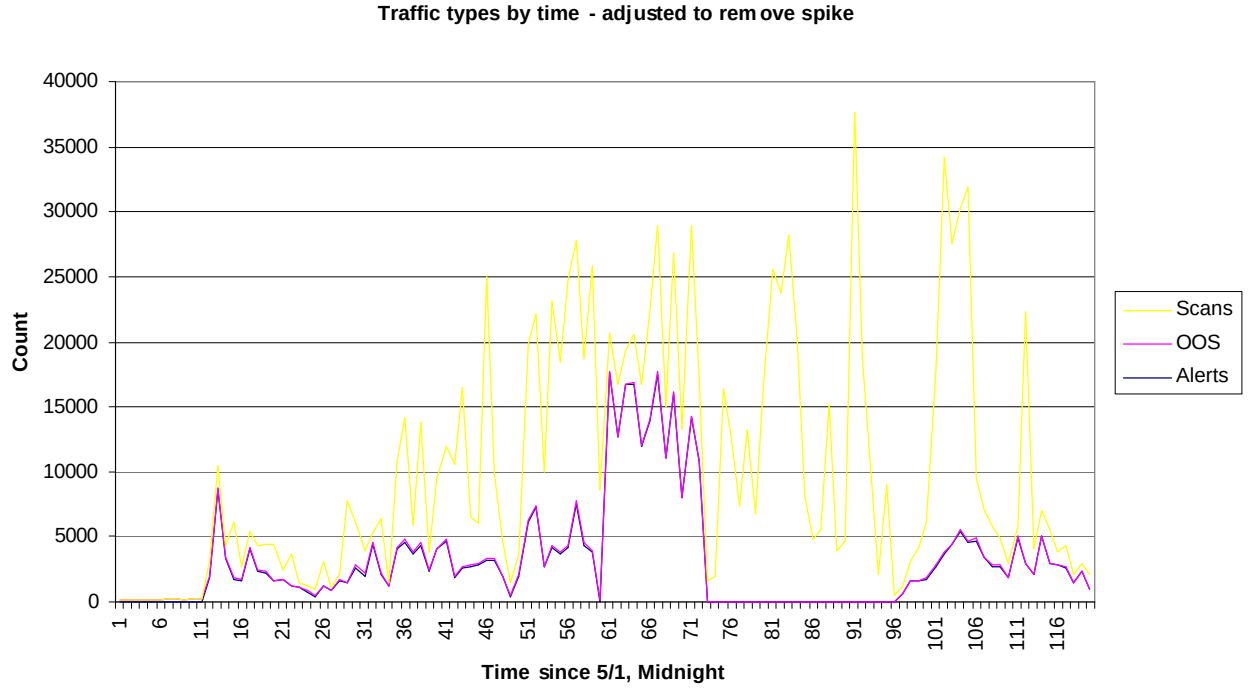
Attack / Alert	Count	Ext -> Ext	Ext -> Int	Int -> Int	Int -> Ext	Uniq Ext Src	Uniq Int Src	Uniq Int Dst	Uniq Ext Dst
SYN-FIN scan!	2	0	2	0	0	2	0	2	0
Bugbear@MM virus in SMTP	1	0	1	0	0	1	0	1	0
DDOS TFN Probe	1	0	1	0	0	1	0	1	0
[UMBC NIDS IRC Alert] Possible trojaned machine detected	1	0	1	0	0	1	0	1	0
site exec - Possible wu-ftpd exploit - GIAC000623	1	0	1	0	0	1	0	1	0

Legend	Explanation
Attack	The name of the attack or alert signature from the data set. Many of these names are customized and are not "normal" Snort rules.
Count	Simple frequency of the specific alert.
Ext -> Ext	This lists the occurrence of an attack whose source IP and destination IP is not "123.123.X.X". This means that the sensor picked up a packet who's source address is most likely forged. Forged source IP's are common in DoS/DdoS attacks when there is a compromised host on the internal network.
Ext -> Int	This lists traffic from outside the network inbound to internal hosts. These represent what one would think of as "normal attacks" from the Internet (the sort of things that firewalls are supposed to help protect from).
Int -> Int	This lists traffic from inside the network to other hosts inside the network. When this activity is seen by a sensor, it likely means a false positive, an internal host is attacking other hosts, or a forged source address.
Int – Ext	The last possible combination, this is when an internal host is sending a packet outbound that triggered an alert. This may indicate a false positive, a subverted internal host, or a host is being monitored for some reason.
Uniq Ext Src	This is the count of distinct source IP addresses from the external network that triggered the alert.
Uniq Int Src	This is the count of distinct source IP addresses from the internal network that triggered the alert.
Uniq Int Dst	This is the count of distinct destination (target) IP addresses on the internal network.
Uniq Ext Dst	This is the count of distinct destination (target) IP addresses on the external network that triggered the alert.

Scan/Alert Activity Chart (Statistics, Part Two)

It is necessary to use two charts that show activity. The first has all data, and the second has the single spike (5/3, 11AM to 12PM, 22% of traffic) removed. The X-Axis represents hours since day one (5/1 at midnight).





The second chart is more revealing and snows a better picture of activity. By removing the single spike, the data is no longer "flattened".

Identifying Relationships

Most of the alerts will be discussed in this section. The more critical and interesting individual alerts that highlight issues that the University should consider will be discussed in some depth; others will have a few comments, and a few have no comments. Alerts are listed in descending order of occurrence. The number of occurrences is in parenthesis. Alerts indicating scanning or alerts that cannot be effectively analyzed are listed separately. Four lists of unique addresses are given. These represent internal unique source/destination and external unique source/destination addresses. The top five are shown. For each address, the count of alerts for that address will be listed in parentheses. Where possible, a SANS/GCIA candidate correlation will be listed[45], with details for each paper listed in "References".

Incomplete Packet Fragments Discarded (355297)

Ext'l Source: 99	Int'l Sources: 2	Int'l Destinations: 66	Ext'l Destination: 6
12.129.72.164 (78)	123.123.210.114 (354834)	123.123.221.138 (76)	213.97.198.23 (354768)
12.129.72.165 (76)	123.123.203.98 (1)	123.123.168.105 (75)	213.97.198.2305 (34)
12.129.72.172 (61)		123.123.207.30 (64)	233.2.171.1 (27)
64.12.56.35 (42)		123.123.224.138 (47)	233.2.171.105/0 (3)
141.151.63.124 (11)		123.123.211.26 (13)	192.168.1.4 (1)

Normally, fragmentation occurs with a packet is too large for one of the networks between sender and receiver. For example, Token Ring packets are larger than Ethernet packets. Fragmentation has become a source of attack over the past several years, thus it deserves some attention. For example, an attacker may be sending in packets designed to elude an IDS. There are a variety of attacks that can cause a system to crash from malicious fragmentation. An attacker may send in fragments where one subsequent fragment is "inside" a previous fragment (known as a teardrop attack). There is a specific attack where a total packet size that exceeds the TCP/IP maximum packet size (65,535 bytes) is sent – this often causes the host system to crash (known as Ping of Death). Lastly, a fragment group may be sent with part of the group missing (deliberately or accidentally), which will tie up system resources and then normally elicit an ICMP error message. This error message provides an attacker with valuable information – the system is up, the system listened, and the system informed the sender of "bad data". This last pattern can be reconnaissance designed to look "normal" to an IDS or firewall.

Note that host 123.123.210.114 produced 354,834 alerts (39% of the total alerts and over 99% of this alert type). The network support software on this machine may need to be reconfigured and should be checked for compromise. An example of misconfiguration would be an older Linux system with a 2.0 kernel using the Andrew File System (AFS)[46], which recently caused excessive fragmentation for the author.

[45] Not all alerts have GIAC correlations - many are just "observed", or have no significant discussion in practicals.
[46] This specific example occurred recently at the author's site. An older RedHat client with a 2.0 kernel was connecting to a new AFS Cell and was not closing files properly. Using tcpdump, we found that the client was sending fragmented packets that were about 200 bytes in size.

Issue: Fragmentation should occur rarely if at all on modern networks. Effort should be made to determine why fragmentation is occurring and eliminate it.

SANS/GCIA Correlation: Cory Steers, Edward Peck, David Jenkins

TCP SRC and DST outside network (208305)

Ext'l Source: 198915	Int'l Sources: 2196	Int'l Destinations: 0	Ext'l Destination: 2196
192.168.1.100 (486)			64.202.103.12 (106932)
0.0.0.0 (43)			65.116.88.75 (43804)
192.168.8.17 (29)			146.100.53.56 (29559)
10.0.1.2 (12)			216.200.173.18 (25217)
169.254.163.42 (10)			200.140.153.140 (458)

This alert occurs when the IDS sees traffic that both originates from and is destined off the home network (123.123.X.X). Network traffic should have at least one of the two addresses on the home network. Alert counts of magnitude (23% of all total alerts) normally indicate one thing – some sort of compromised system that is likely running a Denial of Service attack. This list here includes "private" addresses[47] (192.168.X.X, 10.X.X.X) and automatically configured addresses (169.254.X.X). Therefore, one must dig a littler deeper to validate/confirm the impression that sources are "private". There were 207,716 alerts with source address not on private IP networks with 198,893 unique source addresses with no clear distribution pattern.

With some understanding of a University's mission, the private IP addresses are most likely labs, internal networks, and APIPA[48] addresses come from Windows 9X/2000/XP PC's that are not centrally managed or received and address from campus DHCP[49]. Alerts in these address ranges are "noise", and can be discounted. Alerts whose source and destination are genuinely off of the network are worthy of further investigation, and their occurrence on a well-managed network indicates some sort of compromise.

Issue: Source addresses that an IDS sees outbound should be known at all times – the site should be able to track down any source address of any packet leaving the network. If not, the packet is a strong candidate for compromise, spoofing, or crafted scanning.

SANS/GCIA Correlation: Michael Wilkinson

SMB Name Wildcard (174119)

Ext'l Source: 22474	Int'l Sources: 0	Int'l Destinations: 40910	Ext'l Destination: 2
133.82.241.150 (8412)		123.123.24.34 (1797)	233.2.171.1 (15)
216.78.180.128 (2639)		123.123.194.13 (820)	233.2.171.105/0 (1)
195.167.225.233 (2032)		123.123.249.134 (750)	
143.248.115.88 (1898)		123.123.222.166 (659)	
66.1.191.80 (1503)		123.123.24.44 (646)	

The vast majority of this traffic is connection attempts from the outside seeking to locate listening/reachable Windows (or Samba configured on a Unix/Linux) systems. It is entirely

[47] Private addresses are defined and described in RFC 1918. URL: http://www.ietf.org/rfc/rfc1918.txt?number=1918
[48] APIPA: Automatic Private Internet Protocol Addressing.
[49] DHCP: Dynamic Host Configuration Protocol

possible to make a normal connection to a Windows share that is improperly configured for security (group Everyone has "Full Control" by default), which can result in an attacker silently stealing or modifying data on the system. Therefore if there are file systems shared and the user did not secure the share permissions, their system is vulnerable to file system manipulation (replacement, modification, theft, etc.).

SANS/GCIA Correlation: Brent Deterding, Marilyn Morris, Martin Kinwan

spp_http_decode: IIS Unicode attack detected (30427)

Ext'l Source: 275	Int'l Sources: 528	Int'l Destinations: 174	Ext'l Destination: 705
130.125.82.27 (65)	123.123.153.143 (2388)	123.123.222.166 (130)	218.153.6.197 (2482)
211.90.88.43 (42)	123.123.97.213 (2200)	123.123.217.206 (39)	211.233.29.9 (2162)
194.80.238.42 (24)	123.123.153.176 (1825)	123.123.201.218 (36)	218.153.6.229 (1997)
211.97.161.70 (20)	123.123.153.165 (1701)	123.123.252.251 (28)	210.219.197.11 (1573)
130.60.65.102 (17)	123.123.153.149 (1321)	123.123.249.18 (24)	218.153.6.244 (1483)

This set of alerts appears to be a generalized alert of UNICODE exploits that are directed against IIS servers.

This particular alert is most often associated with a hybrid (or blended) exploit – a Sun system running "sadmind" which has been compromised and is attacking Microsoft IIS hosts. There is a general class of Microsoft exploits known as UNICODE exploits, with a variety of signatures denoting the specific exploit (attack, worm, virus, what have you). What is important to note here is the preponderance of internal hosts causing this alert to be triggered – it would appear that a recent rash of these exploits have hit the network.

Raw statistics don't tell the whole story, however. There were 528 University sources. It seems odd that a University would have this number of compromised Solaris systems sending this type of traffic over a five-day period. Most likely, this rule is functioning as a "catch all" rule. This rule should be investigated and a more thorough analysis on the rule and the data it produces to make sure it is valid for the University.

This attack was seen 673 times directed at internal hosts[50], with all of these alerts coming originating from the outside network. These attacks are automated (note the times below). Most received the attack 3 – 6 times, and were attempted by the same host twice in a row as the following data set reveals:

```
| DATE                | Source          | Destination and   Port }
| 05/05-21:28:48.982432 | 61.242.154.194  | 123.123.221.138 | 80   |
| 05/05-21:28:48.982432 | 61.242.154.194  | 123.123.221.138 | 80   |
| 05/05-21:28:48.982432 | 61.242.154.194  | 123.123.221.138 | 80   |
| 05/05-21:52:30.070209 | 12.213.238.102  | 123.123.200.78  | 80   |
| 05/05-22:00:06.300526 | 156.17.168.1    | 123.123.222.166 | 80   |
| 05/05-22:09:51.273832 | 156.17.168.1    | 123.123.222.166 | 80   |
| 05/05-22:12:04.002331 | 80.5.219.171    | 123.123.222.166 | 80   |
| 05/05-22:16:05.532169 | 80.5.219.171    | 123.123.222.166 | 80   |
```

[50] The query was "select date, src, dst, dstp from alert where attack like "spp_http_decode: IIS Unicode attack%" and dst like "123.123.%" order by date"

Issue: Microsoft IIS servers should never be deployed without the current patches to prevent them from being exploited by this type of an attack.

SANS/GCIA Correlation: Potheri Mohan (actually a GCIH practical), Joe Rayford
Other Correlation: URL: http://www.unl.edu/security/virus_alerts/sadmind.htm

High port 65535 udp - possible Red Worm - traffic (27260)

Ext'l Source: 163	Int'l Sources: 78	Int'l Destinations: 109	Ext'l Destination: 222
65.120.111.17 (1839)	123.123.201.58 (13423)	123.123.201.58 (10628)	65.120.111.17 (1992)
64.118.111.251 (1469)	123.123.240.62 (190)	123.123.207.230 (135)	66.42.68.210 (1678)
66.42.68.210 (1045)	123.123.207.230 (129)	123.123.206.70 (124)	64.118.111.251 (1604)
62.75.136.123 (945)	123.123.228.50 (100)	123.123.240.62 (119)	12.235.90.8 (1114)
12.235.90.8 (838)	123.123.233.10 (98)	123.123.201.38 (85)	62.75.136.123 (918)

Port 65535 is frequently the listening port for a variety of Trojan applications. An example is the Adore Worm, which can be activated with a specific ICMP packet. The internal source machines generating this traffic should be investigated to make sure that they are not running Trojans.

SANS/GCIA Correlation: Matthew Fiddler

High port 65535 tcp - possible Red Worm - traffic (23632)

Ext'l Source: 83	Int'l Sources: 64	Int'l Destinations: 67	Ext'l Destination: 92
67.161.246.193 (3294)	123.123.201.38 (3945)	123.123.201.38 (3294)	67.161.246.193 (3944)
218.141.54.99 (2549)	123.123.226.250 (3454)	123.123.226.250 (2549)	218.141.54.99 (3454)
213.161.3.60 (1697)	123.123.226.206 (1320)	123.123.226.206 (1697)	213.161.3.60 (1320)
217.127.167.6 (1214)	123.123.233.134 (846)	123.123.233.134 (1214)	217.127.167.6 (846)
65.161.73.251 (830)	123.123.206.130 (403)	123.123.218.18 (830)	198.248.222.157 (402)

As above – traffic to and from port 65535 is often indicative of a Trojan on the system.

SANS/GCIA Correlation: Doug Kite, Matthew Fiddler

Tiny Fragments - Possible Hostile Activity (13532)

Ext'l Source: 21	Int'l Sources: 1	Int'l Destinations: 21	Ext'l Destination: 900
12.207.10.226 (4292)	123.123.235.110 (9161)	123.123.234.82 (4290)	141.158.2.187 (474)
212.194.174.202 (15)		123.123.71.164 (25)	200.44.28.208 (443)
213.23.15.177 (14)		123.123.204.26 (14)	200.168.70.146 (404)
68.36.90.84 (8)		123.123.237.254 (7)	24.61.80.253 (360)
212.194.100.196 (7)		123.123.204.78 (4)	200.77.81.95 (336)

See "Incomplete Packet Fragments Discarded" above for related discussion.

TFTP - Internal TCP connection to external tftp server (9337)

Ext'l Source: 33	Int'l Sources: 11	Int'l Destinations: 12	Ext'l Destination:
64.12.28.99 (1028)	123.123.201.42 (1721)	123.123.201.42 (1535)	64.12.30.224 (1113)
64.12.30.224 (937)	123.123.223.114 (1071)	123.123.223.114 (1013)	64.12.28.99 (1101)
64.12.25.166 (887)	123.123.235.206 (746)	123.123.189.41 (670)	64.12.25.166 (970)
160.75.92.16 (586)	123.123.238.114 (615)	123.123.235.206 (602)	160.75.92.16 (372)
64.12.27.86 (249)	123.123.189.41 (489)	123.123.238.114 (584)	64.12.27.86 (290)

The **T**rivial **F**ile **T**ransfer **P**rotocol (TFTP) is an unauthenticated, completely open file transfer application. Historically it has been used for loading operating systems, uploading

and downloading Cisco router configurations, and other applications where a file may be moved. Notice the high number of external sources shown in red that appear as both source and destination addresses (64.12.29.99 and 64.12.25.166). The majority of this traffic in and out of the network – no traffic was recorded within the network.

An ARIN query shows IP 64.12.28.99, 64.12.25.166 and 64.12.27.86 belong to America Online. It is highly doubtful that a dialup or broadband subscriber would need to use TFTP in and out of the network - most end users would be using a browser or a FTP based program to move data on and off of University systems. The example below explains why this alert deserves attention.

Example Attack Using TFTP: One of the common attacks generated against Microsoft IIS servers is to send specially crafted HTTP strings with UNICODE characters and run commands. If the web server were vulnerable then the attacker would attempt to run the native Windows TFTP application to retrieve a Trojan application from the attackers system. Next the attacker would use the UNICODE exploit to run their code, which would install a Trojan.

SANS/GCIA Correlation: Doug Kite, Edward Peck
Other Correlation: The example exploit discussed was demonstrated in the Microsoft Security Clinic, Section One, in over 20 cities in the USA during 2002/2003.

EXPLOIT x86 NOOP (6019)

Ext'l Source: 168	Int'l Sources: 0	Int'l Destinations: 147	Ext'l Destination: 0
24.45.157.41 (2966) 198.144.65.56 (1087) 12.16.131.99 (699) 80.148.9.10 (325) 24.107.25.179 (219)		123.123.190.93 (414) 123.123.227.86 (249) 123.123.86.19 (224) 123.123.130.64 (217) 123.123.228.198 (166)	

This type of traffic is often a "false positive", because the machine code "NO OP" character sequence can occur in a variety of normal traffic. Often times it is part of image data in web pages (an MPEG, MP3, or JPEG for example).

SANS/GCIA Correlation: David Obom, Jeff Zahr

connect to 515 from outside (5033)

Ext'l Source: 3	Int'l Sources: 0	Int'l Destinations: 4873	Ext'l Destination: 0
128.46.117.76 (4872) 68.49.94.97 (160) 152.1.193.6 (1)		123.123.70.199 (160) 123.123.132.16 (2) 123.123.1.202 (1) 123.123.1.203 (1) 123.123.1.208 (1	

This alert is an example of a directed attempt to exploit a known problem with the BSD Line Printer Daemon (LPD). There are fourteen (14) known vulnerabilities in the LPD software defined in the CVE database[51]. Several of them allow an attacker to gain "super

[51] Source URL: http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=LPD

user" access (root in Unix/Linux), which would allow for all kinds of dangerous activity. Notice that there are almost 4900 distinct internal addresses for this alert – someone from the outside is "trolling" for a vulnerable system running LPD.

SANS/GCIA Correlation: David Singer

[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC (5023)

Ext'l Source: 1	Int'l Sources: 12	Int'l Destinations:	Ext'l Destination: 21
	123.123.198.221 (3926) 123.123.83.100 (501) 123.123.132.27 (262) 123.123.84.250 (201) 123.123.101.42 (79)		205.188.149.12 (3925) 208.194.163.37 (294) 205.160.101.121 (269) 157.156.254.222 (201) 66.252.30.186 (153)

This specific alert is not in the current rule set – but ones that are very close are. Assuming that this rule is the close to rule 1639 "CHAT IRC DCC file transfer request" and/or rule 1640 "CHAT IRC DCC chat request", this alert indicates that X-DCC file transfer activity is occurring – or perhaps an XDCC "bot" is advertising itself in a channel. IRC is documented in RFC 1459[52]. There is an excellent article by Gibson Research, which explains in great depth and detail, how IRC was used by a thirteen-year-old hacker to perform a Distributed Denial of Service attack against grc.com.[53] The article explains how automated "bot's" from compromised machines can be used to launch attacks against anyone connected to the Internet. Essentially, once some sort of minimal Trojan software is installed on the Windows PC, the program uses IRC to announce its presence and provide its remote controller with its IP, user name, and password.

This particular alert will be used in order to generate a link graph, showing the relationship of source to destination. This alert would indicate that a potentially compromised client is attempting to share files over IRC using the X-DCC protocol, which is often used for nefarious activities.

In the following table shows details for hosts highlighted in red text on the graph. The senders all had multiple IRC alerts. All of the 123.123.X.X machines were involved in an IRC "kill" - which is used to remove a user from a chat room. An administrator or an automated program could have done this upon seeing XDCC advertisements.

Direction	IP Address	Attack Signature
DST	123.123.83.100	[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.
SRC	123.123.83.100	[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC
DST	123.123.83.100	Queso fingerprint
DST	123.123.83.100	SMB Name Wildcard
DST	123.123.105.48	[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.
DST	123.123.105.48	[UMBC NIDS IRC Alert] K:line'd user detected, possible trojan.
SRC	123.123.105.48	[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC
DST	123.123.105.48	SMB Name Wildcard

[52] Source URL: http://www.ietf.org/rfc/rfc1459.txt?number=1459
[53] Source URL: http://grc.com/dos/grcdos.htm

DST	123.123.194.125	[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.
SRC	123.123.194.125	[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC
DST	123.123.194.125	Queso fingerprint
DST	123.123.194.125	SMB Name Wildcard
DST	123.123.223.78	[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.
SRC	123.123.223.78	[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC
SRC	123.123.223.78	IRC evil - running XDCC
DST	123.123.223.78	SMB Name Wildcard
SRC	38.192.23.234	[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.
SRC	38.192.23.234	[UMBC NIDS IRC Alert] User joining Warez channel detected. Possible XDCC bot
DST	38.192.23.234	[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC
DST	38.192.23.234	IRC evil - running XDCC
SRC	198.163.214.2	[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.
SRC	198.163.214.2	[UMBC NIDS IRC Alert] Possible Incoming XDCC Send Request Detected.
DST	198.163.214.2	[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC
DST	198.163.214.2	IRC evil - running XDCC
DST	233.2.171.1	[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC
DST	233.2.171.1	CS WEBSERVER - external web traffic
DST	233.2.171.1	High port 65535 tcp - possible Red Worm - traffic
DST	233.2.171.1	High port 65535 udp - possible Red Worm - traffic
DST	233.2.171.1	Incomplete Packet Fragments Discarded
DST	233.2.171.1	Null scan!
DST	233.2.171.1	Queso fingerprint
DST	233.2.171.1	SMB Name Wildcard
DST	233.2.171.1	TCP SRC and DST outside network
DST	233.2.171.1	Tiny Fragments - Possible Hostile Activity

SANS/GCIA Correlation: none identified.
Other Correlation: There is an excellent and frequently reproduced article by TonikGin titled "XDCC – An .EDU Admin's Nightmare"[54] which discusses XDCC and how to exploit Windows systems in a campus environment in great detail. Another good website is hosted by Balduz who has written software titled "XDCC Packet Catcher"[55]. Balduz's license is such that his (?) software cannot be used by law enforcement.

[54] Source URL: http://www.russonline.net/tonikgin/EduHacking.html
[55] Source URL: http://catcher.home.dhs.org/

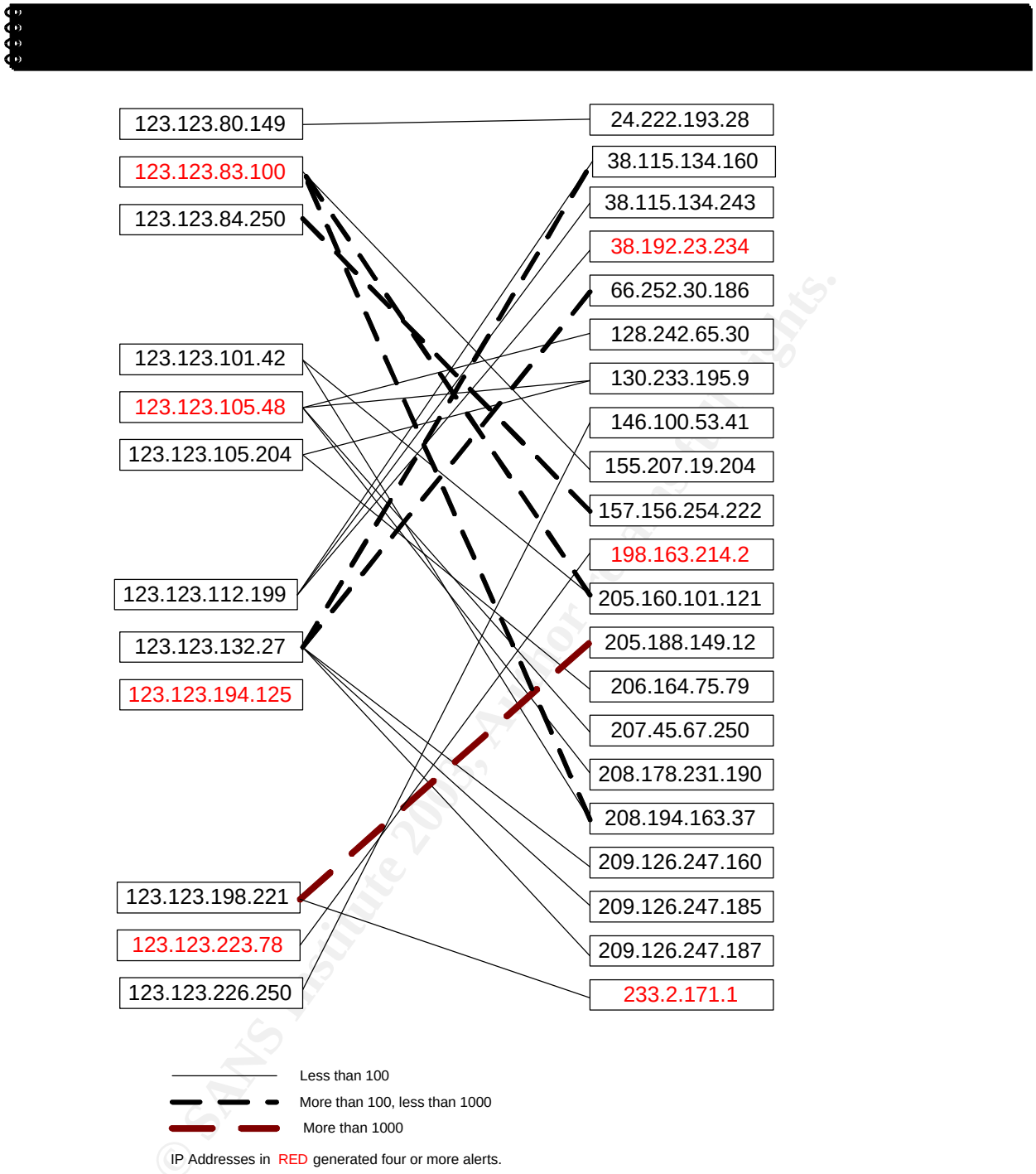


Figure 4: Link Graph of " XDCC client detected attempting to IRC"

spp_http_decode: CGI Null Byte attack detected (5020)

Ext'l Source: 35	Int'l Sources: 119	Int'l Destinations: 6	Ext'l Destination: 127
194.80.238.42 (26)	123.123.236.134 (1282)	123.123.222.166 (27)	216.241.219.14 (1282)
213.66.193.81 (4)	123.123.53.122 (388)	123.123.91.245 (26)	216.241.219.22 (905)
219.40.184.46 (3)	123.123.252.22 (344)	123.123.204.26 (7)	192.151.53.10 (438)
67.249.232.209 (2)	123.123.53.198 (246)	123.123.217.206 (6)	193.149.126.3 (343)
149.159.46.33 (2)	123.123.97.96 (205)	123.123.24.34 (2)	66.135.192.226 (186)

Another alert this is often a false positive. The majority of destination ports for this traffic matches web servers (port 80 and 8080).

SANS/GCIA Correlation: Cory Steers

[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan. (1562)

Ext'l Source: 74	Int'l Sources: 0	Int'l Destinations: 63	Ext'l Destination:
216.90.96.115 (241) 216.152.64.155 (220) 208.178.231.190 (136) 216.32.207.207 (132) 160.94.151.137 (129)		123.123.97.128 (319) 123.123.223.78 (309) 123.123.227.246 (179) 123.123.249.250 (155) 123.123.97.146 (142)	

This particular signature is not in the default Snort 1.9 rule base, and the IRC alerts that are defined to not include a "kill" option. The "/Kill" command will disconnect an IRC user from an IRC server. There are some Trojans that will advertise themselves through an IRC server; a channel administrator may send a "kill" to the user in order to curtail the activity. Note that these alerts are all inbound - meaning that internal users are doing something to get their sessions killed.

Possible trojan server activity (921)

Ext'l Source: 48	Int'l Sources: 21	Int'l Destinations: 179	Ext'l Destination: 24
80.199.219.86 (93) 141.151.82.206 (71) 80.161.122.170 (35) 131.164.194.210 (31) 12.235.154.197 (24)	123.123.220.50 (461) 123.123.201.234 (55) 123.123.208.46 (10) 123.123.220.54 (7) 123.123.12.4 (6)	123.123.220.50 (93) 123.123.220.54 (24) 123.123.201.234 (19) 123.123.12.4 (6) 123.123.25.22 (6)	80.199.219.86 (461) 68.114.228.18 (55) 62.202.71.181 (10) 12.235.154.197 (7) 218.145.25.46 (4)

This rule is highly ambiguous, and does not correspond directly to current Snort rules. Source ports for the 123.123.X.X network include: 80, 110, 1214, 143, 1528, 2304, 2959, 2986, 3038, 3162, 3877, and 6346. Destination ports for destinations on the 123.123.X.X network include 80, 110, 143, 1214, 1382, 1433, 1455, 1801, 1867, 27374, 2986, 3038, 3162, 3315, 3516, 3877, and 4662. Destination ports for hosts' incoming traffic (hosts originating off of the 123.123.X.X network) were limited to one: 27374, the port number for the Trojan SubSeven. There were 150 distinct source ports for hosts not on the 123.123.X.X network.

Issue: If possible, the 21 internal sources should be checked for the SubSeven Trojan.

[UMBC NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC (746)

Ext'l Source: 0	Int'l Sources: 3	Int'l Destinations: 0	Ext'l Destination: 11
	123.123.97.128 (390) 123.123.97.146 (208) 123.123.195.99 (148)		216.90.96.115 (313) 216.152.64.155 (284) 209.176.110.218 (51) 12.30.169.39 (37) 64.62.150.55 (34)

There are about ten variations on the "sdbot" Trojan. This Trojan allows a remote user to control a Windows PC using IRC. These 3 internal hosts should be checked for

compromise by looking in the local TCP/IP connection table and seeing if they are running a local ident server, or a bot connection to an IRC server. As sighted in the Gibson Research article, the Windows commands are:

```
netstat -an | find ":113 "
netstat -an | find ":6667"
```

Other Correlation: Symantec Corporation.[56] Gibson Research[57].

IDS552/web-iis_IIS ISAPI Overflow ida nosize (717)

Ext'l Source: 455	Int'l Sources: 0	Int'l Destinations: 590	Ext'l Destination: 0
61.170.226.22 (54)		123.123.249.240 (6)	
211.97.104.57 (53)		123.123.130.27 (5)	
211.93.40.29 (34)		123.123.75.9 (4)	
130.132.187.227 (20)		123.123.197.27 (4)	
130.160.201.90 (19)		123.123.211.94 (4)	

This alert is a similar other IIS related alerts - this one is a specific attempt against the index server and is mitigated by current Service Packs for Windows NT/2000/XP.

SANS/GCIA Correlation: Joe Ellis

SUNRPC highport access! (520)

Ext'l Source: 30	Int'l Sources: 0	Int'l Destinations: 23	Ext'l Destination: 0
64.12.24.27 (353)		123.123.194.187 (353)	
128.8.10.18 (88)		123.123.24.8 (88)	
131.118.254.39 (8)		123.123.221.2 (12)	
66.187.232.100 (8)		123.123.99.11 (8)	
64.12.26.98 (7)		123.123.221.74 (8)	

This is a particular attack that, if successful, can allow an attacker to gain super user access on a Sun Solaris system. There is a particular standard range for Sun RPC services, and this alert is registered when an attacker attempts to connect to the registered RPC service port range.

SANS/GCIA Correlation: David Singer, Andrew Siske
Other Correlations: The University should be aware that Sun RPC services are the number one vulnerability on the SANS/FBI Critical Internet Security Vulnerabilities list[58].

Issue: Any of the 23 destination hosts that are Sun's should be checked for compramise.

TFTP - Internal UDP connection to external tftp server (395)

Ext'l Source: 14	Int'l Sources: 31	Int'l Destinations: 19	Ext'l Destination: 33
68.14.128.176 (18)	123.123.189.41 (61)	123.123.207.230 (9)	217.234.140.155 (73)
63.250.195.10 (3)	123.123.197.70 (44)	123.123.197.70 (4)	217.234.142.60 (58)
217.234.137.29 (3)	123.123.237.170 (37)	123.123.211.154 (3)	217.234.134.19 (48)
217.125.139.175 (2)	123.123.71.164 (35)	123.123.234.102 (3)	217.234.137.29 (43)

[56] Source URL: http://securityresponse.symantec.com/avcenter/venc/data/backdoor.sdbot.html
[57] Source URL: http://grc.com/dos/grcdos.htm(although referenced in a previous footnote).
[58] URL: http://www.sans.org/top20/#U1

63.250.205.57 (2)	123.123.226.206 (21)	123.123.244.182 (2)	217.234.131.238 (41)

Similar comments for the TFTP alert above apply here.

SANS/GCIA Correlation: Daniel A. Russell II, Jim Hurst

[UMBC NIDS IRC Alert] User joining Warez channel detected. Possible XDCC bot (271)

Ext'l Source: 9	Int'l Sources: 0	Int'l Destinations: 5	Ext'l Destination: 0
209.221.61.43 (246) 209.126.201.242 (8) 38.115.134.51 (7) 64.83.108.187 (2) 63.98.19.242 (2)		123.123.217.194 (262) 123.123.196.23 (3) 123.123.210.134 (2) 123.123.243.62 (2) 123.123.234.210 (2)	

A user joining a "warez" channel discussion often indicates something potentially illegal or nefarious is going on. The "hacker" community is notorious for using a "z" instead of an "s" to identify itself. Often, the location of freely available copyrighted software openly discussed in these channels. Alternately, malicious code such as rootkits may be posted or located. It is also probable that a Trojan may be announcing itself in these channels.

Issue: Computers should be checked to make sure that they are not involved in illegal activity. Universities are continually being notified by the RIAA for potential DMCA violations – and with the RIAA winning lawsuits lately, a University should take this class of alert seriously as it may expose them to direct legal challenge and financial liability.

SANS/GCIA Correlation: Les Gordon

[UMBC NIDS IRC Alert] Possible Incoming XDCC Send Request Detected. (194)

Ext'l Source: 11	Int'l Sources: 0	Int'l Destinations: 11	Ext'l Destination: 0
195.159.0.85 (67) 134.33.33.33 (44) 206.167.75.79 (38) 194.78.213.3 (17) 206.84.2.2 (8)		123.123.105.204 (87) 123.123.201.34 (45) 123.123.241.246 (32) 123.123.206.238 (17) 123.123.80.209 (6)	

As discussed for other XDCC traffic an IRC client or a bot is involved in file transfers.

IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize (188)

Ext'l Source: 0	Int'l Sources: 2	Int'l Destinations:	Ext'l Destination: 167
	123.123.97.181 (184) 123.123.97.48 (4)		130.223.114.111 (3) 130.63.106.222 (3) 130.63.174.86 (3) 130.63.182.132 (2) 130.223.137.44 (2)

The vulnerability identified by this alert is in Microsoft IIS where an attacker can cause a buffer overflow and execute arbitrary program code on the web server. Given that two specific internal machines generated this traffic and it is directed outbound it is highly possible these machines are compromised with some sort of worm.

SANS/GCIA Correlation: Donald Gregory

IRC evil - running XDCC (168)

Ext'l Source: 0	Int'l Sources: 15	Int'l Destinations:	Ext'l Destination: 14
	123.123.80.209 (65) 123.123.241.246 (44) 123.123.206.238 (20) 123.123.217.94 (10) 123.123.249.250 (8)		134.33.33.33 (88) 206.167.75.79 (40) 194.78.213.3 (18) 216.152.65.144 (4) 63.98.19.244 (3)

There are some hosts on the network that are functioning as XDCC servers using an IRC channel.

SANS/GCIA Correlation: Rick Yuen, Doug Kite

[UMBC NIDS IRC Alert] User joining XDCC channel detected. Possible XDCC bot (149)

Ext'l Source: 6	Int'l Sources:	Int'l Destinations: 5	Ext'l Destination:
209.221.61.43 (143) 209.126.201.242 (2) 80.247.205.219 (1) 66.28.62.223 (1) 132.232.0.10 (1)		123.123.217.194 (145) 123.123.226.166 (1) 123.123.243.62 (1) 123.123.194.213 (1) 123.123.201.214 (1)	

As discussed previously, XDCC is an automated file sharing facility that is used by many IRC servers. There is a companion utility (Iroffer) to help aid in publishing and posting file sharing from an IRC server. More frequently these types of servers are use for nefarious purposes.

Issue: Automated file sharing "systems" such as XDCC can be used in a highly automated fashion from system compromise. Often, these IRC channels may contain copyrighted materials such as video and music.

SANS/GCIA Correlation: Al Williams

External RPC call (149)

Ext'l Source: 4	Int'l Sources: 0	Int'l Destinations: 149	Ext'l Destination: 0
213.140.12.216 (93) 213.140.12.215 (54) 141.157.67.253 (1) 152.1.193.6 (1)		123.123.2.2 (1) 123.123.2.4 (1) 123.123.2.35 (1) 123.123.2.38 (1) 123.123.2.43 (1)	

As with the previous RCP alert, this alert signifies a directed attempt for systems running RPC's (normally Sun systems). The attacker 213.140.12.215 generated 38 scan attempts over a 2 minute period on 5/4 about 9:46 AM. These scans were all directed to "130.85.81.X, port 111", which is off of the University's network.

SANS/GCIA Correlation: Mark Mekne

EXPLOIT x86 setuid 0 (128)

Ext'l Source: 118	Int'l Sources: 0	Int'l Destinations: 106	Ext'l Destination: 0
64.237.43.146 (4) 213.100.47.243 (3) 138.87.209.10 (2) 129.237.246.11 (2) 193.110.91.3 (2)		123.123.132.23 (4) 123.123.211.138 (3) 123.123.206.238 (3) 123.123.54.37 (3) 123.123.24.8 (2)	

SANS/GCIA Correlation: Matthew Fiddler

SNMP public access (98)

Ext'l Source: 5	Int'l Sources: 0	Int'l Destinations: 9	Ext'l Destination: 0
132.250.134.33 (43) 147.46.56.20 (32) 130.206.173.31 (19) 158.123.183.25 (3) 131.118.250.215 (1)		123.123.154.26 (32) 123.123.163.43 (23) 123.123.162.31 (20) 123.123.190.55 (7) 123.123.190.200 (7)	

This is an application level scan searching for poorly configured SNMP[59]. The attacker is attempting to find SNMP servers with a community string using default of "public" - the community string only authentication method between SNMP servers. It's unlikely but these packets might be accidental, from a misconfigured console or server.

SANS/GCIA Correlation: This is listed on the SANS Top Twenty list of Critical Internet Security Vulnerabilities[60].

NIMDA - Attempt to execute cmd from campus host (60)

Ext'l Source: 0	Int'l Sources: 2	Int'l Destinations:	Ext'l Destination: 58
	123.123.97.181 (57) 123.123.97.48 (3)		130.158.21.162 (2) 130.39.232.52 (2) 130.233.165.4 (1) 130.158.175.253 (1) 130.161.164.219 (1)

NIMDA is an automated worm that is continually searching for vulnerable IIS servers from a compromised server. Here, two systems are infected with this worm. System 123.123.97.181 also generated a different alert. NIMDA is well understood and can be circumvented with Microsoft patches/service packs and current antiviral software. These alerts were clustered between 5/3 from 22:00 hrs to 5/4 00:45 hrs. It would appear that University countermeasures were effective in limiting exposure.

SANS/GCIA Correlation: Donald Merchant

EXPLOIT x86 setgid 0 (53)

Ext'l Source: 50	Int'l Sources: 0	Int'l Destinations: 48	Ext'l Destination: 0
131.118.254.130 (3) 198.64.135.190 (2) 66.149.100.116 (1)		123.123.24.8 (3) 123.123.252.134 (2) 123.123.227.6 (2)	

[59] Simple Network Management Protocol
[60] Source URL: http://www.sans.org/top20/

216.32.202.192 (1) 208.163.53.25 (1)		123.123.208.30 (2) 123.123.250.210 (1)	

SANS/GCIA Correlation: Frequently counted, rarely discussed.

EXPLOIT x86 stealth noop (51)

Ext'l Source: 12	Int'l Sources: 0	Int'l Destinations: 6	Ext'l Destination: 0
131.118.254.130 (21) 129.165.254.6 (15) 63.105.19.40 (3) 24.169.246.137 (2) 68.11.218.33 (2)		123.123.24.8 (22) 123.123.163.143 (15) 123.123.234.250 (11) 123.123.234.78 (1) 123.123.224.14 (1)	

SANS/GCIA Correlation: Frequently counted, rarely discussed.

TCP SMTP Source Port traffic (34)

Ext'l Source: 3	Int'l Sources: 0	Int'l Destinations: 12	Ext'l Destination: 0
65.61.132.218 (19) 211.147.25.99 (14) 61.235.215.130 (1)		123.123.100.230 (13) 123.123.24.22 (6) 123.123.163.5 (4) 123.123.154.194 (2) 123.123.230.44 (2)	

Inbound traffic on port 25 should occur to messaging servers using the Simple Mail Transfer Protocol (SMTP). It is doubtful that the internal destinations are mail servers; if so, there would be tens of thousands of these alerts. By using a well-known port for an information service that is normally allowed in through a firewall, an attacker is attempting to elude an IDS and scan the inner network by eliciting ICMP error messages.

SANS/GCIA Correlation: Jeff Holland, Loraine Weaver

Back Orifice (26)

Ext'l Source: 2	Int'l Sources: 0	Int'l Destinations: 26	Ext'l Destination: 0
81.77.146.65 (25) 61.152.209.21 (1)		123.123.12.101 (1) 123.123.12.233 (1) 123.123.17.227 (1) 123.123.18.135 (1) 123.123.55.199 (1)	

The attacker seems to be "trolling" for a Windows Trojan application. The top IP, 81.77.146.65, is registered to the ISP "Energis UK" in the Netherlands[61].

SANS/GCIA Correlation: Loraine Weaver

SMB C access (13)

Ext'l Source: 11	Int'l Sources: 0	Int'l Destinations: 9	Ext'l Destination:
63.89.120.133 (2) 200.195.24.4 (2) 210.197.112.84 (1) 67.112.40.29 (1) 219.65.43.171 (1)		123.123.137.34 (3) 123.123.190.100 (2) 123.123.190.102 (2) 123.123.137.36 (1) 123.123.132.24 (1)	

[61] A query was made of .ARIN for the IP, and referred to RIPE.

As with the previous SMB alert, this appears to be outsiders looking for vulnerable or open Windows hosts.

Attempted Sun RPC high port access (10)

Ext'l Source: 3	Int'l Sources: 0	Int'l Destinations: 3	Ext'l Destination: 0
63.250.195.10 (6) 63.250.207.62 (2) 129.6.15.29 (2)		123.123.53.49 (6) 123.123.84.173 (2) 123.123.162.64 (2)	

These may be false positives. The port range for high order RPC ports can also be used for client ephemeral ports (ports for a user application). There are more than 100 RPC specific rules defined for RPC traffic - this rule should have a more descriptive name assigned to it.

RFB - Possible WinVNC - 010708-1 (8)

Ext'l Source: 4	Int'l Sources: 4	Int'l Destinations: 4	Ext'l Destination: 4
151.196.51.184 (1) 68.55.196.211 (1) 12.234.194.144 (1) 68.50.16.210 (1)	123.123.111.51 (1) 123.123.217.194 (1) 123.123.150.86 (1) 123.123.168.180 (1)	123.123.90.201 (1) 123.123.111.51 (1) 123.123.205.162 (1) 123.123.168.180 (1)	68.55.196.211 (1) 24.225.200.75 (1) 68.55.34.178 (1) 68.50.16.210 (1)

WinVNC is a remote management tool that is designed to provide a user with complete access to the console as if they were sitting in front of the console. While a remote management tool is not inherently dangerous, usage of WinVNC over the public Internet should be carefully investigated to make sure security is at the utmost. Earlier versions of VNC transmitted the password in clear text. The host numbers highlighted in red are examples of someone externally attempting to connect to and/or successfully managing a system with VNC.

SANS/GCIA Correlation: Edward Peck, Jim Hurst

[UMBC NIDS IRC Alert] K:line'd user detected, possible trojan. (7)

Ext'l Source: 6	Int'l Sources: 0	Int'l Destinations: 4	Ext'l Destination: 0
65.35.129.149 (2) 158.36.176.47 (1) 208.15.4.42 (1) 213.114.30.234 (1) 206.167.75.78 (1)		123.123.234.250 (4) 123.123.206.206 (1) 123.123.105.48 (1) 123.123.224.222 (1)	

FTP passwd attempt (7)

Ext'l Source: 3	Int'l Sources: 0	Int'l Destinations: 2	Ext'l Destination: 0
212.16.216.3 (5) 151.92.176.5 (1) 68.86.248.127 (1)		123.123.24.27 (5) 123.123.24.47 (2)	

TFTP - External UDP connection to internal tftp server (6)

Ext'l Source: 4	Int'l Sources: 1	Int'l Destinations: 4	Ext'l Destination: 1
63.250.205.60 (2) 63.250.207.52 (1) 63.250.207.57 (1) 63.208.170.220 (1)	123.123.210.114 (1)	123.123.114.54 (2) 123.123.208.62 (1) 123.123.117.155 (1) 123.123.114.44 (1)	213.97.198.23 (1)

Comment above for the preceding TFTP alert apply here.

DDOS shaft client to handler (4)

Ext'l Source: 2	Int'l Sources:	Int'l Destinations: 2	Ext'l Destination:
66.135.208.200 (3) 218.145.28.15 (1)		123.123.84.235 (3) 123.123.97.90 (1)	

TFTP - External TCP connection to internal tftp server (3)

Ext'l Source: 2	Int'l Sources: 1	Int'l Destinations: 2	Ext'l Destination: 1
12.207.10.226 (1) 152.1.193.6 (1)	123.123.132.26 (1)	123.123.234.82 (1) 123.123.132.26 (1)	152.1.193.6 (1)

NIMDA - Attempt to execute root from campus host (3)

Ext'l Source:	Int'l Sources: 1	Int'l Destinations:	Ext'l Destination: 3
	123.123.97.181 (3)		130.223.24.233 (1) 130.223.39.181 (1) 130.223.104.68 (1)

Previous comments for the NIMDA alert apply.

EXPLOIT x86 NOPS (2)

Ext'l Source: 1	Int'l Sources:	Int'l Destinations: 1	Ext'l Destination:
204.174.19.31 (2)		123.123.241.186 (2)	

As above for the other "No Op" exploit attempt, this may be a false positive because No Op code bytes are often seen in graphic image data.

DDOS TFN Probe (1)

Ext'l Source: 1	Int'l Sources:	Int'l Destinations: 1	Ext'l Destination:
159.153.176.243 (1)		123.123.16.13 (1)	

This is an example of a Distributed Denial of Service (DDoS) trolling for machines potentially compromised with the Tribal Flood Network Trojan.

Other Correlations: Tribal Flood Network (TFN) is discussed in the SANS GCIA curriculum.

site exec - Possible wu-ftpd exploit - GIAC000623 (1)

Ext'l Source: 1	Int'l Sources:	Int'l Destinations: 1	Ext'l Destination:
24.191.90.120 (1)		123.123.222.30 (1)	

Bugbear@MM virus in SMTP (1)

Ext'l Source: 1	Int'l Sources:	Int'l Destinations: 1	Ext'l Destination:
160.94.128.49 (1)		123.123.6.47 (1)	

Alerts Indicating Scanning

The Snort intrusion detection system is very good at detecting a variety of scans against the network. Their importance are not to be minimized; rather, since scanning is so common details on these alerts are condensed in this analysis. Due to the open nature of a University campus, these types of intrusions are commonplace.

- **Null scan! (2474):** 115 distinct sources and 109 distinct destinations. Detailed discussion is in "Network Intrusion Detection", Ch 14.
- **Queso Fingerprint (1577):** 328 distinct sources with 123 distinct destinations.
- **NMAP TCP ping! (145):** with 41 unique sources and 60 unique destinations.
- Probable NMAP fingerprint attempt (12): with 8 unique sources and 9 destinations.
- **SYN-FIN scan! (2):** Two unique sources and two unique destinations.

Alerts With Insufficient Information

There are several alerts that cannot be analyzed without the rule base, or the alert information provided did not provide enough raw data. These include:

- **CS WEBSERVER External web traffic (24935):** SANS/GCIA Correlation's by ade Walker, Mike Poor, Scott Baird explains this is a custom rule.
- **123.123.30.4 activity (1343):** with 294 distinct sources. Destination ports are common for a Microsoft server (80, 111, 135, 137, 445, 1433, 8009, 17300, 56464) and it appears the Novell client is in use (port 524).
- **123.123.30.3 activity (804):** with 44 distinct sources. The port list for was missing NetBIOS ports (destination ports are 80, 515, 524, 1433, 3019, 8009, and 17300).
- **CS WEBSERVER - external ftp traffic (781)**.
- **Notify Brian B. 3.54 tcp (26).**
- **Notify Brian B. 3.56 tcp (22).**
- **[UMBC NIDS IRC Alert] Possible trojaned machine detected (1)**

Top Talkers (Still More Statistics)

There are a variety of ways to qualify and quantify the "top talkers". Raw quantity information represents counts of senders and receivers by alert, scan, and OOS packets. Qualified information attempts to look at a specific alert, to eliminate noise or false positive packets and determine sender/receiver, time, responses, or other types of quality data. Examples are presented below.

Top Talkers – Alerts Data

First are the top 10 source IP addresses, regardless. Last are the top 10 destination IP addresses. In between these two lists are Filtered Source IP's, which are IP address from the external network and the IP's of the second highest alert ("TCP SRC and DST outside network") filtered out. By using this criteria the source addresses that are not spoofed and from outside the network directed inbound to the home network are shown.

Top 10 Source IP's	Count
123.123.210.114	354839
216.39.48.127	14011
123.123.201.58	13423
123.123.235.110	9161
133.82.241.150	8412
12.207.10.226	4966
128.46.117.76	4873
123.123.201.38	4027
123.123.198.221	3926
123.123.226.250	3457

Filtered Source IP	Count
216.39.48.127	14011
133.82.241.150	8412
12.207.10.226	4966
128.46.117.76	4873
67.161.246.193	3294
24.45.157.41	2966
216.78.180.128	2639
218.141.54.99	2551
195.167.225.233	2032
143.248.115.88	1898

Top 20 Destination IP	Count
213.97.198.23	354775
64.202.103.12	106932
65.116.88.75	43804
146.100.53.56	29559
123.123.100.165	25839
216.200.173.18	25217
123.123.201.58	10637
123.123.234.82	4972
67.161.246.193	3944
205.188.149.12	3926

There is another layer of analysis to be performed. More telling statistics can be show if one can show the count of distinct targeted hosts and vice versa.

The first table is the top destination with unique source IP's. Put another way, there are 176 unique source addresses when the destination is 123.123.194.13. The second table is the number of unique destinations for the source IP's. In other words, there are 182 distinct destination IP's when the source is 218.20.156.204.

Destination	Count
64.202.103.12	98171
65.116.88.75	43804
146.100.53.56	29559
216.200.173.18	25217
123.123.100.165	5518
123.123.222.166	442
123.123.24.44	304
123.123.30.4	298
123.123.24.34	284
123.123.194.13	176

Source	Count
133.82.241.150	7862
128.46.117.76	4872
216.78.180.128	2639
195.167.225.233	2032
143.248.115.88	1898
66.1.191.80	1504
123.123.235.110	899
213.48.36.57	226
123.123.97.181	215
218.20.156.204	182

Top Talkers – Scan Data

Next, the top 10 IP scanning source addresses and destinations were calculated. A second query was ran against the database to make sure that there were no scans from the internal network (source IP = 123.123.X.X.) and there were none.

Source Scan	Count
130.85.210.114	64664
130.85.240.62	39800
130.85.87.50	32605
130.85.250.98	29293
130.85.97.190	26833
130.85.1.3	21850
130.85.234.158	20913
130.85.205.150	16744
152.1.193.6	15962
130.85.153.152	15298

Destinations	Count
213.97.198.23	64602
130.85.132.26	15967
64.39.186.133	1779
66.66.126.241	1737
66.167.144.245	1624
24.42.0.66	1620
68.165.25.243	1570
68.13.93.150	1219
12.245.31.155	1212
68.81.50.22	1186

Top Talkers – OOS Data

Next, the top 10 IP scanning source addresses were calculated. A second query was ran against the database to determine how many IP's were on the home network – there were 41 packets summarized below.

Top OOS Source	Count
209.123.49.137	1564
68.54.93.181	1319
213.197.10.95	463
81.218.114.59	365
64.28.101.9	338
210.233.23.128	318
81.218.109.79	270
66.140.25.157	250
148.64.48.213	214
210.253.214.117	201

Top OOS Dest	Count
123.123.6.7	1374
123.123.218.254	942
123.123.194.13	573
123.123.235.202	559
123.123.226.178	495
123.123.113.4	435
123.123.6.47	387
123.123.24.22	379
123.123.194.125	375
123.123.24.23	357

This table shows the count of OOS packets from internal addresses to external addresses and the number of occurrences.

Internal Source	Destination	Count
123.123.104.113	211.95.129.136	1
123.123.12.2	194.125.183.44	4
123.123.12.4	205.244.232.133	20
123.123.17.30	80.12.56.12	1
123.123.194.179	64.12.151.110	2
123.123.252.14	152.163.208.249	3
123.123.40.11	218.109.210.40	2

Internal machines sending OOS packets require attention by system administrators – OOS packets should not be generated on modern operating systems using well-written software.

External Sources with Justifications

First, the most popular hosts for the alert "TCP SRC and DST outside network" should be investigated. These are 64.202.103.12 and 65.116.88.75. Since these are the top destinations for traffic that is spoofed (forged source address), it would be useful to determine the target of this type of traffic and attempt corrective action.

Host: 64.202.103.12	Host: 65.116.88.75
Initial Query: OzShells Internet Solutions SCNET-CHG-OZSHELLS1 (**NET-64-202-103-0-1**) 64.202.103.0 - 64.202.103.255 Server Central Network SCN-CHG-1 (NET-64-202-96-0-1) 64.202.96.0 - 64.202.127.255	**Initial Query:** Qwest Communications NET-QWEST-BLKS-4 (NET-65-112-0-0-1) 65.112.0.0 - 65.127.255.255 CREATIVE INTERNET TECHNIQUES QWST-65-116-88 (**NET-65-116-88-0-1**) 65.116.88.0 - 65.116.95.255
Specific details: OrgName: OzShells Internet Solutions OrgID: OIS-41 Address: P.O. Box 6006 City: Waikiki StateProv: Western Australia PostalCode: 6169	**Specific Details:** OrgName: CREATIVE INTERNET TECHNIQUES OrgID: CRTV Address: 3982 POWELL ROAD Address: SUITE 225 City: POWELL StateProv: OH PostalCode: 43065

```
Country:    AU

NetRange:   64.202.103.0 -
64.202.103.255
CIDR:       64.202.103.0/24
NetName:    SCNET-CHG-OZSHELLS1
NetHandle:  NET-64-202-103-0-1
Parent:     NET-64-202-96-0-1
NetType:    Reassigned
NameServer: NS1.OZSHELLS.COM
NameServer: NS2.OZSHELLS.COM
Comment:
RegDate:    2003-02-11
Updated:    2003-02-11

TechHandle: KBU8-ARIN
TechName:   Butler, Kevin
TechPhone:  +61 409 108608
TechEmail:  admin@ozshells.com

OrgTechHandle: KBU7-ARIN
OrgTechName:   Butler, Kevin
OrgTechPhone:  +61 409 108608
OrgTechEmail:  admin@ozshells.com
```

```
Country:    US

NetRange:   65.116.88.0 -
65.116.95.255
CIDR:       65.116.88.0/21
NetName:    QWST-65-116-88
NetHandle:  NET-65-116-88-0-1
Parent:     NET-65-112-0-0-1
NetType:    Reallocated
Comment:
RegDate:    2002-03-12
Updated:    2002-03-12

TechHandle: CA544-ARIN
TechName:   Admin, CIT
TechPhone:  +1-740-881-0323
TechEmail:  ip-admin@foonet.net

OrgTechHandle: CA544-ARIN
OrgTechName:   Admin, CIT
OrgTechPhone:  +1-740-881-0323
OrgTechEmail:  ip-admin@foonet.net
```

For the first address, 64.202.103.12, dshield.org does not have any attack information. Dshield does report that the IP corresponds to the host name "giving.head.for-money.net", which does not sound like a place to visit. Using Internet Explorer and typing in the IP address while keeping the Alt-F4 key combination ready, a simple web page appeared saying "enigma.ozshells.com" (a placeholder). Putting the second IP address (65.116.88.75) into IE for a plain Apache install page appears. This IP does not have a domain name (nslookup revealed nothing). Dshield had no entries for this IP either.

Another address that warrants investigation is the most popular host for the alert "Incomplete Packet Fragments Discarded ", 213.97.198.23. Based on knowing the target, perhaps some determination can be made why the fragmentation is occurring.

Host: 213.97.198.23

```
inetnum:      213.97.0.0 - 213.97.255.255
netname:      RIMA
descr:        Telefonica De Espana SAU (NCC#2000013794)
descr:        Red de servicios IP
descr:        Spain
country:      ES
admin-c:      LJP5-RIPE
tech-c:       FLT14-RIPE
rev-srv:      scmrro3.nombres.ttd.es
rev-srv:      scmrro4.nombres.ttd.es
rev-srv:      ns.ripe.net
status:       ASSIGNED PA
remarks:      For ABUSE/SPAM/INTRUSION issues
remarks:      PLEASE CONTACT THROUGH LINK
remarks:      http://www.telefonicaonline.com/nemesys/
remarks:      or send mail to nemesys@telefonica.es
```

```
remarks:      any mail to adminis.ripe@telefonica.es will be ignored
notify:       adminis.ripe@telefonica.es
mnt-by:       MAINT-AS3352
changed:      adminis.ripe@telefonica.es 20000302
changed:      adminis.ripe@telefonica.es 20020530
changed:      administracion.ripe@telefonica-data.com 20030121
source:       RIPE
route:        213.97.0.0/16
descr:        TTDNET (Red de servicios IP)
origin:       AS3352
mnt-by:       MAINT-AS3352
mnt-routes:   MAINT-AS3352
mnt-lower:    MAINT-AS3352
changed:      administracion.ripe@telefonica-data.com 20010308
source:       RIPE
person:       L Jimenez
address:      TELEFONICA DE ESPANA
address:      Emilio Vargas, 4
address:      28043-MADRID
address:      SPAIN
phone:        +34 91 5846497
fax-no:       +34 91 5842650
e-mail:       adminis.ripe@telefonica.es
nic-hdl:      LJP5-RIPE
remarks:      For ABUSE/SPAM/INTRUSION issues
remarks:      PLEASE CONTACT THROUGH LINK
remarks:      http://www.telefonicaonline.com/nemesys/
remarks:      or send mail to nemesys@telefonica.es
remarks:      any mail to adminis.ripe@telefonica.es will be ignored
notify:       ah@telefonica.es
changed:      adminis.ripe@telefonica.es 20020530
source:       RIPE
person:       Francisco Lorenzo de Tuero
address:      TELEFONICA DE ESPANA
address:      Emilio Vargas, 4
address:      28043-MADRID
address:      SPAIN
phone:        +34 91 5194446
fax-no:       +34 91 5846936
remarks:      For ABUSE/SPAM/INTRUSION issues
remarks:      PLEASE CONTACT THROUGH LINK
remarks:      http://www.telefonicaonline.com/nemesys/
remarks:      or send mail to nemesys@telefonica.es
remarks:      any mail to adminis.ripe@telefonica.es will be ignored
e-mail:       tecnic.ripe@telefonica.es
nic-hdl:      FLT14-RIPE
notify:       ah@telefonica.es
changed:      ah@telefonica.es 20020225
source:       RIPE
```


An initial search at whois.arin.net indicated the RIPE Network Coordination Centre maintained this address. This address does not have a listing at www.dshield.org, so it is not known for being an attacker.

While there are other good candidates for investigation, the top two external hosts involved in the XDCC traffic from above should be investigated. As discussed previously,

IRC and XDCC traffic is often used for less than legal file transfer activity (not always, and use of a file sharing application does not prove guilt). They are

Host: 205.188.149.12	Host: 205.160.101.121
OrgName: America Online, Inc OrgID: AMERIC-59 Address: 22080 Pacific Blvd City: Sterling StateProv: VA PostalCode: 20166 Country: US NetRange: 205.188.0.0 - 205.188.255.255 CIDR: 205.188.0.0/16 NetName: AOL-DTC NetHandle: NET-205-188-0-0-1 Parent: NET-205-0-0-0-0 NetType: Direct Assignment NameServer: DNS-01.NS.AOL.COM NameServer: DNS-02.NS.AOL.COM Comment: RegDate: 1998-04-18 Updated: 1998-04-27 TechHandle: AOL-NOC-ARIN TechName: America Online, Inc. TechPhone: +1-703-265-4670 TechEmail: domains@aol.net	**Initial Query:** Sprint SPRINT-BLKE (NET-205-160-0-0-1) 205.160.0.0 - 205.163.255.255 Randolph Macon Academy RANDOLPH-MACON-BLK1 (**NET-205-160-101-96-1**) 205.160.101.96 - 205.160.101.127 Sprint Midatlantic Telecom SPRINT-CDA067 (NET-205-160-96-0-1) 205.160.96.0 - 205.160.103.255 **Specific Details:** OrgName: Randolph Macon Academy OrgID: RMA-10 Address: 200 Academy Drive City: Front Royal StateProv: VA PostalCode: 22630 Country: US NetRange: 205.160.101.96 - 205.160.101.127 CIDR: 205.160.101.96/27 NetName: RANDOLPH-MACON-BLK1 NetHandle: NET-205-160-101-96-1 Parent: NET-205-160-96-0-1 NetType: Reassigned Comment: RegDate: 2000-04-19 Updated: 2000-04-19 TechHandle: SL370-ARIN TechName: Lewellen, Stu TechPhone: +1-540-636-5420 TechEmail: stu@rma.edu OrgTechHandle: SL370-ARIN OrgTechName: Lewellen, Stu OrgTechPhone: +1-540-636-5420 OrgTechEmail: stu@rma.edu

Randolph Macon Academy is a boarding school with a military program. Perhaps there are some students with to much time on their hands?

Defensive Recommendation

Most University's want to maintain an "open and free" environment for the pursuit of academics. In order to "lock down" (disable) access from the Internet to inside campus resources, a justification must exist which can provide the network administrators with sufficient ammunition to challenge the prevailing desire of academia. Given sufficient

information about the risks, people should understand that there are prudent reasons to control traffic. The defensive recommendations here are based on these principles. If this were a corporate environment, the network would be configured differently (block everything, allow in only what is absolutely necessary and don't allow everything out).

There are several configurations that can be made to improve site security and to improve the quality of alert reporting from the IDS. Each will be discussed in turn.

Ingress and Egress Filtering: implement ingress and egress filtering on the network. Ingress filtering drops the packet if the source address is on the inside (University) network. For example, Ingress filtering would drop any packet whose source address is "123.123.X.X". Egress filtering drops any packet whose source address is not on the interior (University) network. The downside is that the IDS may not see IP spoofing. The upside is that liability (its systems attacking other systems) can be curtailed.

Content Filtering for IRC Traffic: There is a preponderance of XDCC traffic in the alerts. From the customized alerts the University has chosen to monitor for XDCC traffic over IRC, and IRC in general. Since there is a tendency to use this traffic for illegal file sharing (movies, MP3, copyrighted software), the University would be well advised to know its potential liability. Recent pushes by the RIAA to enforce the DMCA through lawsuits put a University environment at risk. The University IT staff needs to communicate the General Counsel in order to determine the liability issues, limits, and considerations.

Some recent examples of RIAA/MPAA targets of the DMCA include[62]:

- Harvard University
- University of Connecticut
- Secour sued
- IcraveTV.com sued
- RecordTV.com sued
- Verizon forced to turn over more than 450 subscribers names (June 2003)

Stop NetBIOS At the Perimeter: Windows file sharing is designed for workgroup and departmental usage; there are much more secure ways of allowing files to be shared without enabling Windows file sharing, and the associated protocols that it needs. In short - block inbound traffic to TCP and UDP ports 135-139.

Locate Servers on networks w/o Workstations: Server systems should be located on a network segment (address space, physical switch fabric and Virtual LAN) from client workstations. There are numerous exploits that can be used to get Windows Administrator passwords by sniffing the local network and this should be curtailed.

[62] Articles and source URL's include:

- http://news.com.com/2100-1023-255961.html?tag=rn
- http://www.linuxjournal.com/article.php?sid=5992
- http://news.com.com/2100-1025_3-1013154.html

Second, traffic to and from University production support servers should be controlled such that inside networks can access servers, and outside networks cannot. By locating servers in their own address block, analysis of IDS can occur more quickly. Analysts will be able to better identify servers from workstations from NAT'd address spaces.

Supplemental IDS: Because of the "openness" that a University desires, it would be a wise investment in time and resources to implement inner IDS systems configured to monitor for alerts that indicate compromise (Nimda, port 65535, port traffic to common Trojan ports, SRC/DEST not on network, etc.). This IDS would need to capture more of the packet, particularly the MAC addresses. By having the physical address of a network adapter on the University network a compromised system should be much easier to locate. During analysis, no MAC addresses were provided in the source data, making it impossible to determine the source of spoofed packets.

Consider deployment of Personal Firewalls: There are several products available for the desktop that can detect, defend against, and disable most of the current threats, which a University is likely to face. The University should perform a cost benefit / justification analysis in order to determine if the cost of software purchase, maintenance and the often hidden cost of deployment of a individual desktop personal firewall and/or Internet Security product are warranted. The cost of deployment should be weighted against the cost of openness - and the protection that a desktop firewall can provide. For example, deploying a $50.00 product to 3000 desktops costs $150,000. With a technician visiting the desktop and spending two hours to configure and test the product, the costs would be $348,000 assuming that the "fully loaded costs" of a help desk staff worker are about $58.00/hour. Labor is, and is likely always to be, the higher cost – especially since the $50/unit cost would be more like $22-$25/unit when buying in volume. In contrast, a campus wide firewall and content filter would be less – especially when factors such as annual maintenance, PC reinstallation, new unprotected PC's being installed, and visitors with unprotected systems coming into the network are considered.

Increase IDS sensor resolution: As cited in this paper, a University is a current "high value" target for the RIAA/MPAA and the DMCA. There were no alerts over the five-day period for other common file sharing applications such as Gnutella and KaZaa. Also, thousands of records in the alert files produced were incomplete indicating that the system is not capable of accommodating the traffic flowing past the sensor.

Tightly Secure Microsoft Machines: Due to their visibility and market share, Microsoft Windows and various products are high on the attack list. Microsoft has published guidance on securing their products - Windows NT, 2000, IIS, SQL Server, etc. These guides should be followed at a minimum for any Microsoft servers being used on the Campus.

Limit Bandwidth: There are a variety of rate limiting methods that can be built up which can be used to control or curtail file sharing activity. One example is to collect the amount of data transmitted through a Cisco PIX and if that activity goes above a "reasonable"

threshold, put the offending IP in a rate limit pool. This method will allow someone to continue to use Internet resources, but make large file sharing not nearly as effective and discourage its use.

Data Analysis Process

First, a summary of the data construction process is presented. Next, the analysis is presented. Details of the process are in an appendix in this practical.

Tools Used:

- RedHat Linux 8, Windows 2000 SP3, VMware 3
- Snort Ver 1.91
- Perl5
- MySql V.3
- Microsoft Excel and Visio
- Various Snort Digest tools – SnortSnarf, Snortalog, SnortSort

The main steps in the analysis process included:

1. Inspection of the data format that lead to an initial clean up of data. There were thousands of lines in the alert files that were incomplete alert fragments.
2. Parse "scan" data from the "alert" data in an attempt to prepare it for SnortSnarf.
3. Run SnortSnarf on a dedicated dual CPU system with 1.0GB RAM – more than 4 days were used with over 2.0 GB of disk space for the produced reports. Data was separated by type (scan, alert) and by day.
4. Run SnortSort on a semi-dedicated system running under Vmware.
5. Revised clean up of data based on looking at the results.
6. Reanalyze with SnortSnarf, SnortaLog, and SnortSort.
7. Revamp thinking; decided to use a database (great personal thanks and credit to Brandon Newport for providing a solid staring place in his practical).
8. Configure mysql and perl DBI to support assignment on both systems.
9. Reformat the data to support easier data loading – alert, scan, and OOS data were reformatted using a delimiter. Data cleaned one last time in this process.
10. Write a variety of customized Perl programs to produce reports for this assignment
    a. Consolidated error report program (based Hee So's format).
    b. I used a basic alert counting program (based on Chris Kuethe's and Chris Calabrese code)
    c. Detailed alert and address report program in order to produce the details of alert, count, internal/external IP and count information
    d. Specialized report program for XDCC traffic in order to build the link graph
    e. I also wrote a program to generate the counts by time for alerts and scans in order to have input for MS Excel for the timeline charts
11. I felt a strategic investment in coffee, Half and Half, and pure bleached white sugar was prudent at this point. Folgers has a new blend.
12. Use Mysql and write various select queries to determine additional statistics, port information, "distinct" information, counts, and other data relationships.
13. Fumbled around with MS Excel to create link graph's – decided on Visio.

Once the consolidated chart of alert traffic was created, it was apparent that there were several alerts that could not be successfully analyzed due to lack of supplemental information (like the rules files). I spent about two weeks trying to determine which tool was the best to analyze the data – SnortSnarf, SnortALog, SnortSort were all attempted but they proved to cumbersome. Realizing that there was sufficient guidance in Brandon Newport's practical to "get me going" with using MySql and a table layout, I decided to put my RDMBS and Perl skills to the test.

These commands are taken from Brandon Newport's GCIA practical (May 8, 2001). Many of these commands were used; and they provided a model for other commands to run against the database.

```
select count(*) from alert;
select count(distinct src) from alert;
select count(*) from spp;
select count(distinct src) from spp;
select src, count(*) as count from alert group by src order by count desc limit
25;
select dst, count(*) as count from alert group by dst order by count desc limit
25;
select count(*) as count, dstp from alert group by dstp order by count desc
limit 25;
select src, count(*) as count from alert where src like "%MY.NET.%" group by src
order by count desc limit 25;
select count(src) as count from alert where src like "%MY.NET.%";
select count(distinct attack) as count from alert;
select count(distinct attack) as count from spp;
select src, dst, count(distinct src) as count from alert group by dst order by
count desc limit 25;
select src, dst, count(distinct dst) as count from alert group by src order by
count desc limit 25;
```

In order to determine the attack sources for a given signature/alert, A query like the one below can be used: This query retrieves the source IP and the count of those source IP's from outside the home network (123.123.X.X) which match a TFTP attempt. The output is grouped by the source IP, ordered highest to lowest, and limited to five result addresses.

```
mysql> select src, count(src) as count from alert
    -> where src not like "123.123.%"
    -> and attack like "TFTP%"
    -> group by src order by count desc limit 5;
+--------------+-------+
| src          | count |
+--------------+-------+
| 64.12.28.99  |  1028 |
| 64.12.30.224 |   937 |
| 64.12.25.166 |   887 |
| 160.75.92.16 |   586 |
| 64.12.27.86  |   249 |
+--------------+-------+
5 rows in set (1.93 sec)
```

Another query follows. This one selects the destination addresses.

```
mysql> select dst, count(dst) as count from alert
    -> where dst like "123.123.%"
    -> and attack like "TFTP%"
    -> group by dst order by count desc limit 5;
+-----------------+-------+
| dst             | count |
+-----------------+-------+
| 123.123.201.42  |  1535 |
| 123.123.223.114 |  1013 |
| 123.123.189.41  |   670 |
| 123.123.235.206 |   602 |
| 123.123.238.114 |   584 |
+-----------------+-------+
5 rows in set (2.12 sec)
```

I don't mind saying that I struggled a bit with a "link graph". Not all candidates included one, and there isn't a great deal of consistency in presentation. I elected to generate one based on XDCC traffic because my research showed that IRC and XDCC are frequently being used to traffic in illegal media files and Universities are constantly coming under fire from the MPAA and RIAA. The detailed analysis of IRC traffic and XDCC traffic also represented learning – I have only used a chat client one time, and avoid them like the plague. The effort here was to learn something new and attempt to show traffic relationships that indicate potential liability for the University.

References

Print and Web Articles

Burke, Brian. Christiansen, Chris. Kolodgy, Charles. "Emerging Threats to the Employee Computing Environment: Expanding EIM Beyond the Browser An IDC White Paper". IDC Corporation. URL: http://www.websense.com/products/resources/wp/emergingthreats_idc.pdf (April 23, 2003).

Jeff Shawgo, ed. "Securing Windows 2000 Step by Step (Version 1.5)". The SANS Institute, July 1, 2001. Chapter 3.

Balduz. "XDCC Catcher". URL: http://catcher.home.dhs.org/ (June 1, 2003): Note legal warning on site.

Bower, Ben. Farmington, Dean; Weber, Chris. "Security Windows 2000 Professional Using the Gold Standard Security Template". SANS Press, 2002.

Bowman, Lisa. "Broadband fans busted over Gnutella". Apr 17, 2001. URL: http://news.com.com/2100-1023-255961.html?tag=rn (June 7, 2003).

Bowlan, Lisa; Hansen, Evan. "Verizon to hand names over to RIAA". Jun 4, 2003. URL: http://news.com.com/2100-1025_3-1013154.html (June 8, 2003).

Computer Security Institute. "2002 Computer Security Institute/FBI Computer Crime & Security Survey". Apr 7, 2002. p. 10. URL: http://www.gocsi.com/press/20020407.html (Apr 15, 2003).

GFI Corporation. "Nimda Worm: Description". URL: http://www.gfi.com/press/nimdaworm.htm (Apr 23, 2003).

Goland, Yaron; Schlimmer, Jeffery. "Multicast and Unicast UDP HTTP Messages". URL: http://www.upnp.org/download/draft-goland-http-udp-04.txt(May 7, 2003). (Currently in draft status with the UPNP committee).

Goland, Yaron; Cai, Ting; Leach, Paul; Gu, Ye Albright, Shivaun. "Simple Service Discovery Protocol/1.0", Oct 28, 1999. URL: http://www.upnp.org/download/draft_cai_ssdp_v1_03.txt (5/12/03)

Hayes, Bill. " VIRUS ALERT - Solaris/sadmind.worm (sadmind/IIS) Internet Worm". URL: http://www.unl.edu/security/virus_alerts/sadmind.htm (April 28, 2003)

Microsoft Corporation. "Microsoft Operations Manager 2000 Product Information". Jun 7, 2003. URL: http://www.microsoft.com/mom/evaluation/default.asp

Microsoft Corporation. "Microsoft Operations Manager 2000 Deployment and Migration", Jun 7, 2003. URL: http://www.microsoft.com/mom/techinfo/deployment/default.asp

Microsoft Corporation. "Microsoft Operations Manager 2000 Product Documentation", June 7, 2003. URL: http://www.microsoft.com/mom/techinfo/productdoc/default.asp (the Users Guide, Installation Guide, and online Help are all available from this main URL).

National Laboratory for Applied Network Research (NLANR), "NLANR Beacon". URL: http://dast.nlanr.net/Projects/Beacon/ (May 23, 2003).

Roblimo, "Fyodor Answers Your Network Security Questions", May 30, 2003. URL: http://interviews.slashdot.org/interviews/03/05/30/1148235.shtml?tid=126&tid=172&tid=95 (Jun 6, 2003).

Northcutt, Stephenn; Cooper, Mark; Fearrow, Matt; Fredrick, Matt: "Intrusion Signatures and Analysis, 3rd Ed.", New Riders, 2001.

Northcutt, Stephenn; Novak, Judy; McLachlan, Donald. "Network Intrusion Detection: An Analyst's Handbook.2nd Ed". Indianapolis: New Riders, 2000.

SANS Institute. "The Twenty Most Critical Internet Security Vulnerabilities (Updated) ~ The Experts' Consensus". Version 3.23 May 29, 2003 URL: http://www.sans.org/top20/. June 11. 2003

Stevens, W. Richard. "TCP/IP Illustrated, Volume 1". Reading: Addison Wesley Longman, Inc, 1994.

Streck, Phillip. "Conversations: The Case Against the DMCA" (Apr 11, 2002). URL: http://www.linuxjournal.com/article.php?sid=5992(June 7, 2003).

TonikGin. " XDCC – An .EDU Admin's Nightmare". Sept. 11 2002. URL: http://www.russonline.net/tonikgin/EduHacking.html (Jun 4, 2003)

Wall, Larry; Tom Christiansen; Schwartz, Randal L. "Programming Perl 2nd ed." Sebastopol: O'Reilly & Associates, 1991

Warrior,Ulhas; Iyer, Prakash. "WANPPPConnection:1 Service Template Version 1.01". URL: http://www.upnp.org/standardizeddcps/igd.asp(May 4, 2003).

SANS Certification Candidates

Adams, Tony. "10 Detects for SANS GIAC Intrusion Analyst Certification". April 8, 2000. URL: http://www.giac.org/practical/Tony_Adams.doc (June 2, 2003).

Baird, Scott. "Intrusion Detection In Depth GCIA Practical Assignment". May 24, 2002. URL: http://www.giac.org/practical/Scott_Baird_GCIA.doc (Jun 2003)

Calhoun, Chip. "Windows XP UPnP Exploits GCIH Practical Assignment". (Paper not dated; assume certification date of 19 Feb, 2004). URL: http://www.giac.org/practical/Chip_Calhoun_GCIH.doc (Jun 2003)

Cormier, André. "LOGS: GIAC GCIA Version 3.3 Practical Detect(s)". URL: http://cert.uni-stuttgart.de/archive/intrusions/2003/01/msg00121.html (Note: this URL is for a list server, and Mr. Cormier's practical was not published at the time of this submission). (Jun 2003)

Deterding, Brent. GCIA Practical (no title). Dec 8, 2000. URL: http://www.giac.org/practical/Brent_Deterding_GCIA.doc (Jun 2003)

Ellis, Joe. "GCIA Practical Assignment, v3.0 Intrusion Detection In Depth". May 28, 2002. URL: http://www.giac.org/practical/Joe_Ellis_GCIA.doc (Jun 3, 2003).

Gordon, Les. " Intrusion Analysis - The Director's Cut!". May 20, 2002. URL: http://www.giac.org/practical/GCIA/Les_Gordon_GCIA.doc(Jun 2003)

Fiddler, Matthew. "Intrusion detection in depth". Feb 22, 2002. URL: http://www.giac.org/practical/Matthew_Fiddler_GCIA.doc (June 2, 2003)

Gregory, Donald. "SANS GIAC Intrusion Detection in Depth". URL: www.giac.org/practical/GCIA/Donald_Gregory_GCIA.pdf (Jun 3,2003).

Holland, Jeff. "SANS GIAC Level Two: Intrusion Detection in Depth". May 22, 2002. URL: http://www.giac.org/practical/Jeff_Holland_GCIA.doc (Jun 2003)

Hurst, Jim. "GCIA Practical Assignment". Mar 10, 2002. URL: http://www.giac.org/practical/Jim_Hurst_GCIA.doc (June 3, 2003)

Jenkins, David. " Intrusion Detection In Depth GCIA Practical Assignment Version 3.0 (revised August 13, 2001)". URL: http://www.giac.org/practical/David_Jenkins_GCIA.doc (Jun 2003)

Kite, Doug. "Doug Kite. Intrusion Detection in Depth". July 2002. URL: http://www.giac.org/practical/GCIA/Doug_Kite_GCIA.pdf (Jun 5, 2003).

Kirwan, Martin. "GIAC Intrusion Detection Curriculum Practical Assignment for SANS Security DC 2000 July 5 - 10, 2000". URL: http://www.giac.org/practical/Martin_Kirwan.doc (Jun 2003)

Kuenthe, Chris. "Chris Kuethe: GCIA Practical Assignment". 24 Jun 2000. URL: http://www.sans.org/y2k/practical/chris_kuethe_gcia.html (May 2001).

Limesand, Mark. "Analysis of Laurie's Network from January to August". Sep 15, 2000. URL: http://www.giac.org/practical/Mark_Limesand.doc (May 22, 2003).

Menke, Mark. "GIAC INTRUSION DETECTION CURRICULUM PRACTICAL ASSIGNMENT Version 2.2.5". Nov 22, 2000. URL: http://www.giac.org/practical/Mark_Menke_GCIA.doc (May 2003)

Merchant, Donald. "GIAC Certified Intrusion Analysts (GCIA)". Oct 7, 2002. URL: http://www.giac.org/practical/GCIA/Donald_Merchant_GCIA.doc (June 3, 2003)

Mohan, Potheri. "SANS 2000 Certification Practical". June 5, 2000. URL: http://www.giac.org/practical/Potheri_Mohan.doc (Jun 2003).

Morris, Marilyn. (No title). June 10, 2000. URL: http://www.giac.org/practical/Marilyn_Morris.doc Jun 2003).

Peck, Edward. "SANS GIAC GCIA Practical Version 3.0 ". Nov 25, 2001. URL: http://www.giac.org/practical/Edward_Peck_GCIA.doc (Jun 2003).

Poor, Mike. "Mike Poor: Intrusion Detection in Depth". Nov 7, 2001. URL: http://www.giac.org/practical/Mike_Poor_GCIA.doc (May 23, 2003).

Rayford, Joe. "Intrusion Detects and Analysis ". April 4, 2001. URL: http://www.giac.org/practical/Joe_Rayford_GCIA.doc (Jun 2003)

Rodriguez, Thomas M. "Intrusion Detection In Depth". Oct 2001. URL: http://www.giac.org/practical/Thomas_Rodriguez_GCIA.doc (May 13, 2003).

Russell II, Daniel A. "GCIA Intrusion Detection In Depth". Mar 19, 2002. URL: http://www.giac.org/practical/Daniel_Russell_GCIA.doc(Jun 2003)

Singer, David. "GCIA Practical". April 5, 2002. URL: http://www.giac.org/practical/David_Singer_GCIA.doc (June 2003).

Siske, Andrew. "GIAC Intrusion Detection Practical Assignment for SANS Security DC". July 5, 2000. URL: http://www.giac.org/practical/Andy_Siske_GCIA.htm (Jun 2003)

Steers, Cory. "GCIA Practical". May 15, 2002. URL: http://www.giac.org/practical/GCIA/Cory_Steers_gcia.doc (Jun 1, 2003).

Timm, Kevin. "GCIA Version 3.1". Mar 10, 2002. URL: http://www.giac.org/practical/Kevin_Timm_GCIA.doc (May 23, 2003).

Weaver, Loraine. (no title). Jul 4, 2002. URL: http://www.giac.org/GCIA_500.php (Actual URL is a Zip file with four specific MSFT word files). (Jun 2003)

Walker, Wade. "GIAC Certified Intrusion Analyst (GCIA Candidate Wade Walker". Jan 31, 2002. URL: http://www.giac.org/practical/Wade_Walker_GCIA.doc (Jun 2003)

Wilkinson, Michael. " GCIA Practical for SANS Darling Harbour". Mar 27, 2002. URL: http://www.giac.org/practical/michael_wilkinson_gcia.doc (Jun 2003)

Yuen, Rick. "SANS Parliament Hill 2001 GIAC Intrusion Detection, Practical Assignment". Oct 15, 2002. URL: http://www.giac.org/practical/Rick_Yuen_GCIA.doc (Jun 2003)

Williams, Al. "SANS GCIA Practical". Aug 2002. URL: http://www.giac.org/practical/GCIA/Al_Williams_GCIA.pdf (Jun 2003)

Web Sites Common to this Body of Knowledge

cve.mitre.org. "Common Vulnerabilities and Exploits". URL's inclue:

- "Search", http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=Squid+Proxy+attempt (Apr 22, 2003).
- "CVE-2000-0884": http://cve.mitre.org/cve/downloads/full-cve.html (Apr 20, 2003).
- "Search". http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=upnp (May 4, 2003).
- "CVE-2001-0876". URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0876 (May 6, 2003).
- "CVE-2007-0876". URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0876 (May 22, 2003).
- " CAN-2001-0721". URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0721 (May 26, 2003).
- " CVE-2001-0877". URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0877 (May 26, 2003).

msdn.microsoft.com. Microsoft Corporation, MSDN. "MSDN". URL's

- "Platform SDK: Internet Security and Acceleration Server 2000". http://msdn.microsoft.com/library/default.asp?url=/library/en-us/isa/isaabout_3yum.asp (Apr 12, 2003).

project.honeynet.org. " Lists of fingerprints for passive fingerprint monitoring". (May 2000). URL: http://project.honeynet.org/papers/finger/traces.txt (May 23, 2003).

support.microsoft.com. "Microsoft Product Support Services". URL's

- "HOW TO: Install and Use the IIS Lockdown Wizard". http://support.microsoft.com/default.aspx?scid=kb;en-us;325864 (May 22, 2003).
- "How to Change the Default Installation Paths for FTP and the Web". http://support.microsoft.com/default.aspx?scid=kb;en-us;259671 (May 22, 2003)
- "How to Configure MOM to Monitor and Collect UNIX Syslogs" (Aug 29, 2001). URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;297443 (Jun 11, 2003).

www.cert.org. "CERT Coordination Center". URL's

- "Protect your Web server against common attacks". http://www.cert.org/security-improvement/practices/p082.html (May 22, 2003).
- "CA-2001-26". http://www.cert.org/advisories/CA-2001-26.html (May 23, 2003).

www.iana.org. "Internet Number Authority". URL's:

- "Internet Protocol IP V4 Address Space", URL http://www.iana.org/assignments/ipv4-address-space

www.ieee.org. "OUI List". URL: http://standards.ieee.org/regauth/oui/oui.txt (May 19, 2003)

www.ietf.org. "The Request for Comments (RFCs)". URL's:

- "RFC 1459: Internet Relay Chat Protocol", May 1993. URL: http://www.ietf.org/rfc/rfc1459.txt?number=1459(June 12, 2003)
- "RFC 1918 - Address Allocation for Private Internets" (Feb 1996). URL: http://www.ietf.org/rfc/rfc1918.txt?number=1918(May 5, 2003).
- "RFC 3171: IANA Guidelines for IPv4 Multicast Address Assignments" (Aug 2001). URL: http://www.ietf.org/rfc/rfc3171.txt?number=3171 (May 18, 2003).
- "RFC 3179: Address Allocation for Private Internets" (Feb 1996). URL: http://www.ietf.org/rfc/rfc1918.txt?number=1918 (April, May 2003)
- "RFC 3330: Special-Use IPv4 Addresses" (Sep 2002). URL: http://www.ietf.org/rfc/rfc3330.txt?number=3330 (Jun 1, 2003)

www.microsoft.com/technet. "Microsoft TechNet". URL's

- "Migrating Microsoft Proxy Server 2.0". http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/isa/proddocs/isadocs/cmt_upgradprx2indep.asp (April 12, 2003).
- "Security Bulletin MS00-0078". http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms00-078.asp (April 19, 2003).
- "IIS 5 Security Checklist". 2001. URL: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/chklist/iis5cl.asp (June 11, 2003)
- "Security Bulletin MS01-059". http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-059.asp (May 6, 2003).

www.securityfocus.com. "Security Focus". URL: "Microsoft IIS and PWS Extended Unicode Directory Traversal Vulnerability". http://www.securityfocus.com/bid/1806/info/ (May 13, 2003)

www.squid.org. "Squid Configuration Manual". (May 15, 2002). URL. http://squid.visolve.com/squid24s1/network.htm#http_port (April 12, 2003).

securityresponse.symantec.com. Symantec Corporation. URL's:

- "Backdoor.sdbot" May 29, 2003. URL: http://securityresponse.symantec.com/avcenter/venc/data/backdoor.sdbot.html (June 11, 2003)