GIAC Intrusion Detection In Depth

InfoPeople Security Solutions (IPSS) June 2002 Ottawa
GCIA Practical Assignment v3.3

by Michael Dawson

Dec 7, 2002

Assignment 1 The State of Intrusion Detection

Introduction

One of the biggest challenges a Security Analyst will face is the correlation of data from different types of log formats. The disparity of logs from different vendors is one of the main reasons for this. Some of the benefits of normalizing data are the ability to store data from various sources, IDSs, firewalls and system logs in one database. This would make it possible for data analysis to be performed on all of the data not just parts of it. Another advantage would be that an event correlation system could process alerts from a variety of IDS systems allowing better cross correlation. Normalization of data would also allow a Security Analyst view alerts from a single screen and not have to become familiar with many different vendor GUIs. Normalization of data would also allow easier sharing of data between users, vendors, response teams and law enforcement agencies. The intent of this paper is to introduce XML (Extensible Markup Language) and show how XML is being used in the Security Community. With the ability to normalized data an Enterprise IDS solution as depicted in Figure 1 could be realized.

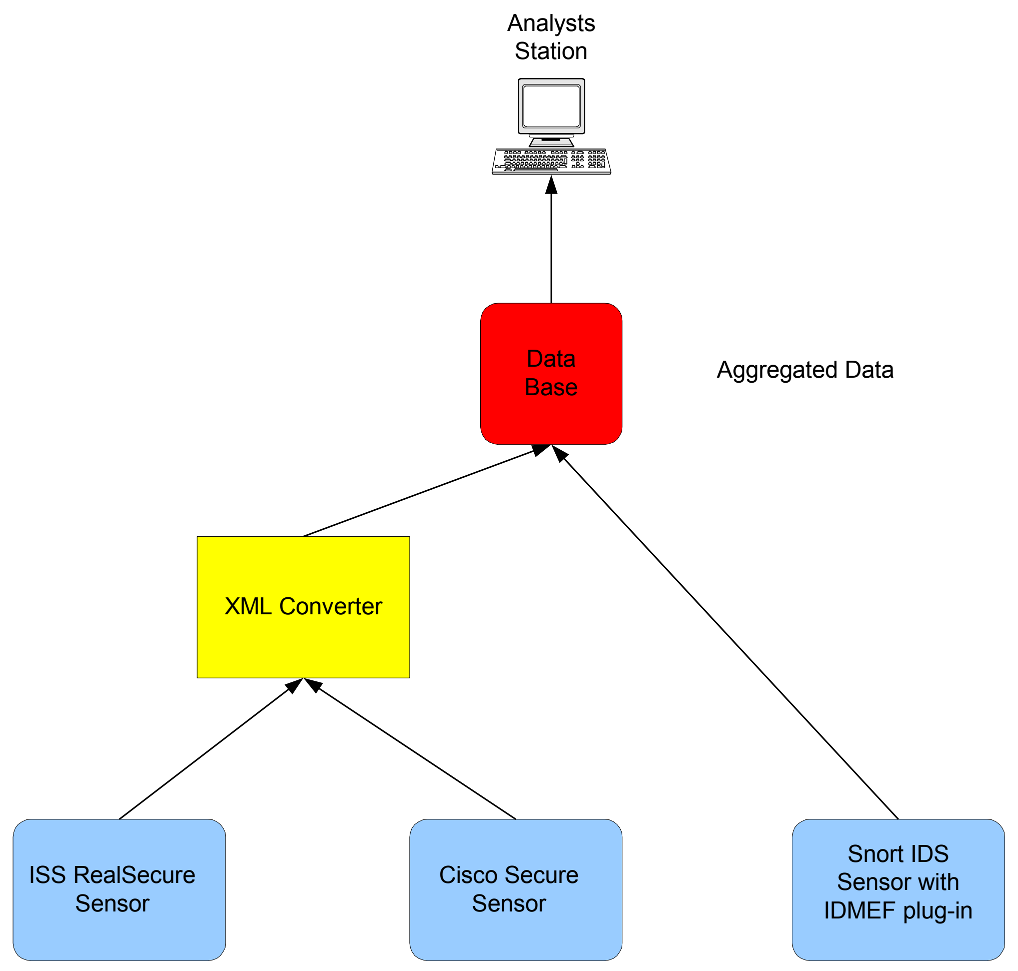


Figure 1

What is XML

XML is a language defined by the World Wide Web Consortium, which is the body that sets the standards for the Web. Markup languages are used to describe how the contents of a document should be interpreted, for example HTML will dictate how a Web site will display in your browser. Both HTML and XML use tags within a document that will be used to interpret the way the document will be presented. HTML tags are predefined and there are only about 100 available with HTML 4.01. Using XML one can define their own tags, the benefits of being able to great your own tags are that you can create as many different tags as needed and tags may be named in such a way as to make it easier to understand what the tags intent is (self documenting).

Here is an Example of HTML vs. XML.

HTML:
```
<HTML>
  <HEAD>
    <TITLE>Hello HTML</TITLE>
  </HEAD>
  <BODY>
    <CENTER>
      <H1>
        Hello HTML
      </H1>
    </CENTER>
    Welcome to the world of HTML.
  </BODY>
</HTML>
```

XML:
```
<?xml version="1.0" encoding="iso-8859-1"?>
<DOCUMENT>
   <GREETING>
       Hello XML
   </GREETING>
   <MESSAGE>
       Welcome to world of XML.
   </MESSAGE>
</DOCUMENT>
```

Here is an explanation of the XML document shown above.

The first line is the prolog, **<?xml version="1.0" encoding="iso-8859-1"?>**states that XML processing will use version 1.0 and that iso-8859-1 encoding is used which will support many foreign character. If no encoding is specified UTF-8 is assumed.

Next line **<DOCUMENT>** is the first opening tag. It could be named anything as long as these rules are followed, the name must start with a letter or underscore (_) followed by letters, underscore, digits, dots (.) or hyphen (-), no spaces are allowed. One other rule that must be followed is that each opening tag must have a closing tag **</DOCUMENT>**, notice the / in the closing tag. In XML the term Element consists of the opening tag, closing tag and the content in between the tags. The **<DOCUMENT>** in this case is called the *root element,* the reason is that every XML document must enclose the entire document except for processing instructions, in one element.

Two other elements have been created **<GREETING>** and **<MESSAGE>** which both contain text. The root element **<DOCUMENT>** contain two other elements **<GREETING>** and **<MESSAGE>.** All XML documents must be well formed and valid which means they must follow the rules laid out by the World Wide Web Consortium, the rules can be found at this W3C site. There are many tools available in order to check for well formedness and validity. James Clark provides a handy parser, called NSGMLS or and XML editor like XML writer can be down loaded for a 30 Day Evaluation from XML writer. There are Web sites that can validate an XML document online, the one used for the XML document created above can be found at w3school's. Two more things need to be covered in order to understand the basics of XML, DTD (Document Type Definition) and Stylesheets.

DTD (Document Type Definition)

A DTD is a document that defines the syntax and structure of the elements that make up an XML document. DTD can be created within the XML doc or can be an external document that is referenced within the XML document. Using the XML document already created a DTD will be added.

assign1.xml

```
<?xml version="1.0" encoding="iso-8859-1"? standalone="yes"?>
<!DOCTYPE DOCUMENT [
<!ELEMENT DOCUMENT (GREETING, MESSAGE)>
<!ELEMENT GREETING (#PCDATA)>
<!ELEMENT MESSAGE (#PCDATA)>
]>
        <DOCUMENT>
            <GREETING>
                 Hello XML
             </GREETING>
             <MESSAGE>
                 Welcome to world of XML.
             </MESSAGE>
        </DOCUMENT>
```

All text added to the XML document has been highlighted for clarity, the new entries will now be explained. Looking at the prolog standalone="yes" states that the XML document does not reference any external documents. The first line of the DTD <!DOCTYPE DOCUMENT [. Is the document type declaration and specifies the root element. Between the opening and closed brackets [ ]is the DTD.

The first entry <!ELEMENT DOCUMENT (GREETING, MESSAGE)> states the Element DOCUMENT can contain two other elements GREETING and MESSAGE.

The next two lines in the DTD <!ELEMENT GREETING (#PCDATA)> and <!ELEMENT MESSAGE (#PCDATA)> specify that the Elements GREETING and MESSAGE will store PCDATA and only PCDATA, which is text. For more information on DTD go to w3schools DTD Tutorial.

Stylesheets
If you were to open the XML document above in a browser, the document would be displayed exactly as typed above tags and all. In order to display the data in a readable format a Style Sheets must be used. Two types of Style Sheets may be used CSS (Cascading Style Sheets) and XSL (Extensible Style Sheets). Only XSL will be covered here because it not only deals with formatting a document but it is also able to transform a document before formatting it. The transformation language of XSL is often called XSLT. Here is the style sheet I created for the XML document created above.

assign1.xsl

```
<?xml version="1.0" encoding="iso-8859-1"?>
  <xsl:stylesheet version="1.0" mlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="DOCUMENT">
     <html>
      <head>
        <title>Today's greeting</title>
      </head>
     <body>
           <p><xsl:value-of select="GREETING"/></p>
           <p><xsl:value-of select="MESSAGE"/></p>
     </body>
     </html>
  </xsl:template>
    </xsl:stylesheet>
```

The XSL style sheet above will now be explained. The second line in the prolog states what XML version it is and mlns:xsl=http://www.w3.org/1999/XSL/Transform is what XSLT standard is being used. XSLT uses templates that need to be matched to specific nodes, in line three the node being matched is DOCUMENT. Once a node is matched the rest of the XSLT instructions will be applied to that node. Next we see what HTML tags will be created in the new document created. For instance the <title> tag will be added to the new document with “Today's greeting” as content. “Today’s greeting” can be seen in the screenshot below as the Browser title. The <p> </p> are the opening and closing tags for a paragraph in HTML. The XSL instruction between the <p> tags is <xsl:value-of select="GREETING"/>. This XML instruction states that whatever value is found in the GREETING node will be copied between the <p> </p> tags. In this case the contents are “Hello XML” as can be seen in the screenshot below. The next line <p><xsl:value-of select="MESSAGE"/></p> is the same as the one above it except the contents of the MESSAGE node are copied in the <p> </p> tags.

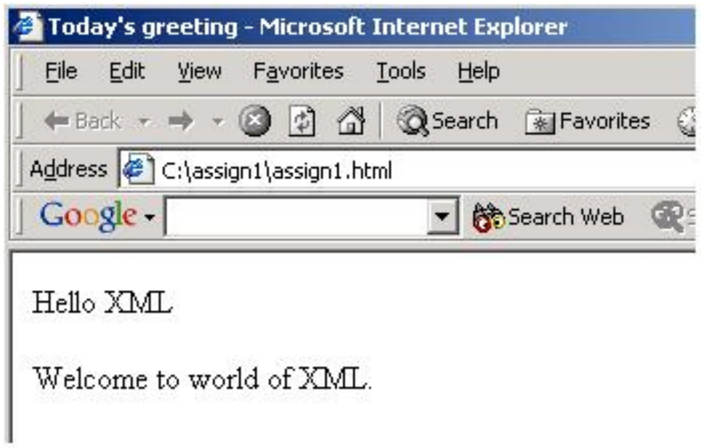
After creating the XSL Style Sheet above it was applied to the XML document, how this was done will now be explained. An XML parser is needed to parse both the XML and XSL document and apply the changes. Many parsers are available on the Internet; the one used for this paper is Instant Saxon and is available for free at Michael Kay’s Web site. Once Saxon is installed, the following command needs to be run from the command prompt in order to apply the XSL Style Sheet.

```
C:\assign1>saxon -o assign1.html assign1.xml assign1.xsl
```

The result of applying the Style Sheet are shown below:

```
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
      <title>Today's greeting</title>
```

```
  </head>
  <body>
    <p>
    Hello XML
    </p>
    <p>
     Welcome to world of XML.
    </p>
  </body>
</html>
```


Today's greeting - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\assign1\assign1.html
Google Search Web

Hello XML

Welcome to world of XML.

This concludes the introduction to XML. The purpose of this introduction was to show and demonstrate some of the basic concept with regards to XML. XML is a huge topic and it is recommended that the following sites be visited for more information. http://www.w3c.org and http://www.w3schools.com

How XML is being used in the Security Community

There are presently three different formats at different states of development the **Incident Object Description Exchange Format (IODEF), Intrusion Detection Message Exchange Format (IDMEF) and Simple Network Markup Language (SNML).**

Incident Object Description Exchange Format (IODEF) is a format that represents data for describing and exchanging data between Computer Security Incident Response Teams (CSIRT). One of the design principles in the IODEF is compatibility with the Intrusion Detection Message Exchange Format (IDMEF). IODEF is based on the IDMEF and provides compatibility with it. An Internet-Draft document can be viewed at the Internet Engineering Task Force (IETF) Web site. The draft lists the following benefits:

- XML provides all the necessary features to define a specific markup language for describing security incidents. It also defines a standard way to extend this language, either for later revisions ("standard" extensions), or for vendor-specific use ("non- standard" extensions).

- Software tools for processing XML documents are widely available in commercial and open source forms. Numerous tools and APIs for parsing and/or validating XML are available in a variety of languages, including Java, C, C++, Tcl, Perl, and Python. Widespread access to tools will make the adoption of the IODEF by product developers easier, and hopefully, faster.

- XML meets IODEF Requirement 4.1 that message formats support full internationalization and localization. The XML standard requires support for both the UTF-8 and UTF-16 encodings of ISO/IEC 10646 (Universal Multiple-Octet Coded Character Set, "UCS") and Unicode, making all XML applications (and therefore all IODEF-compliant applications) compatible with these common character encodings.

- XML also provides support for specifying on a per-element basis, the language in which the element's content is written, making IODEF easy to adapt to local languages in which CSIRTs and their constituency work.

- XML meets IODEF Requirement 4.2 that message formats must support modularity, filtering and aggregation. XML's integration with XSL, a style language, allows messages to be combined, discarded, and rearranged.

- XML is free (no license, license fees or royalties).

Simple Network Markup Language (SNML) is a format to represent TCP, UDP, and ICMP network traffic. The SNML is sometimes called "Snort Markup Language" when used with the snort IDS or as the "Simple Network Markup Language" when used in multi-vendor IDS environments. An XML plug-in is available for Snort at www.snort.org., when you download the source is available in the snapshots download area. More information is available at CERT Knowledgebase site. The following XML formatted message from Incident.org is an example of the output from logging in the SNML format..

Example of output:

```
<report>
  <event version="1.0">
    <sensor encoding="hex" detail="full">
      <interface>eth0</interface>
      <ipaddr version="4">10.0.0.1</ipaddr>
      <hostname>samplesensor.net</hostname>
    </sensor>
    <signature>PING-ICMP Destination Unreachable</signature>
    <timestamp>2000-11-27 03:12:51-04</timestamp>
    <packet>
      <iphdr saddr="10.0.0.1" daddr="10.0.0.2"
             proto="1" ver="4" hlen="5" tos="192" len="140"
             id="27894" ttl="255" csum="63475"
        <icmphdr type="3" code="3" csum="61863">
          <data>000000004500005C9CE900007E3149F18003040D801251730800
               FF25020086DA000000000000000000000000000000000000000
               000000000000000000000000000000000000000000000000000
               000000000000000000000000000000000004500003902000000
               40068CFB8002514518405F1D</data>
        </icmphdr>
      </iphdr>
    </packet>
  </event>
</report>
```

Intrusion Detection Message Exchange Format (IDMEF) is a format to represent data generated by an Intrusion Detection System (IDS) using XML (Extensible Markup Language). The Internet Engineering Task Force (IETF) has a working group called the Intrusion Detection Exchange Format Working Group (IDWG). IDWG's mandate is "to define data formats and exchange procedures for sharing information of interest to intrusion detection and response systems, and to management systems which may need to interact with them." IDWG has proposed Intrusion Detection Exchange Protocol (IDXP) as the communication protocol, although the IDMEF message format is independent of the communication protocol. XML is an extensible format and will let vendors specify additional types of data that go beyond what is specified by the IDMEF DTD (Document Type Definition).

Although Snort has been a leader in accepting IDMEF and has compiled in IDMEF output support since Version 1.8.7, the IDMEF plug-in for snort is available at Silicon Defense here. Many other vendors are using the IDMEF format as well. The VigilEnt Intrusion Manager (VIM) using APIs (Application Programming Interfaces) captures events on leading security devices such as Check Point FW-1/VPN-1, Cisco PIX and SecureIDS and ISS RealSecure. The events are collected at a proxy that normalizes alerts into an IDMEF format for central correlation and analysis. See Information Security magazine article Automating Policies. Intellitactics also uses XML, to make customizing NSM (Network security Manager easier which provides interoperability with other XML-compliant security and network devices. See Information Security magazine article Intellitactics' Network Security Manager. Example of XML document below obtained from a presentation given by Stuart Staniford and Joe McAlerney.

XML Document:

```
<IDMEF-Message version="0.1">
 <Alert alertid="329440" impact="unknown" version="1">
  <Time>
   <ntpstamp>0x3a2d8b3a.0x0</ntpstamp>
   <date>2000-12-05</date>
   <time>16:41:30</time>
  </Time>
  <Analyzer ident="IDS1">
   <Node category="dns">
    <location>San_Diego_Network</location>
    <name>supersnort</name>
    <Address category="ipv4-addr">
     <address>123.234.123.12</address>
    </Address>
   </Node>
  </Analyzer>
  <Classification origin="vendor-specific">
   <name>IDS297/http-directory-traversal1</name>
   <url>http://www.whitehats.com/IDS/IDS297</url>
  </Classification>
<Source spoofed="unknown">
   <Node>
    <Address category="ipv4-addr">
     <address>222.222.111.11</address>
    </Address>
   </Node>
  </Source>
  <Target decoy="unknown">
   <Node category="dns">
    <location>San_Diego_Network</location>
    <Address category="ipv4-addr">
     <address>123.234.123.7</address>
    </Address>
   </Node>
   <Service ident="0">
    <dport>80</dport>
    <sport>1397</sport>
   </Service>
  </Target>
  <AdditionalData meaning="Packet Payload" type="string">
   GET ../../stuff/I/should/not/be/seeing</AdditionalData>
 </Alert>
</IDMEF-Message>
```

The top of the IDMEF message is IDMEF-Message. Below the IDMEF-Message are two types of messages, Alert class and Heartbeat class. A Data Model for the IDMEF message format is shown in Figure 2. For more detail information with regards to the IDMEF see IDWG (Intrusion Detection Working Group) draft.

The Alert Class consists of what information a sensor would send when triggered by an event. The Classes that make up Alerts are:

Analyzer:
Identifies the analyzer that generated the alert.

CreateTime:
The time the alert was created and is required the next two times are optional.

DetectTime:
The time the event(s) leading up to the alert was
detected. In the case of more than one event, the time the first
event was detected. In some circumstances, this may not be the
same value as CreateTime.

AnalyzerTime:
The current time on the analyzer.

Source:
The source(s) of the event(s) leading up to the alert.

Target:
The target(s) of the event(s) leading up to the alert.

Classification:
The name of the alert, or other information that will help the manager determine the alert is.

Assessment:
Information about the impact of the event, actions
taken by the analyzer in response to it, and the analyzer's
confidence in its evaluation.

AdditionalData:
Information included by the analyzer that does not fit into the data model. This may be an atomic piece of data, or a large amount of data provided through an extension to the IDMEF

Alert is represented in the XML DTD as follows:

```
<!ELEMENT Alert                    (
    Analyzer, CreateTime, DetectTime?, AnalyzerTime?, Source*,
    Target*, Classification+, Assessment?, (ToolAlert |
    OverflowAlert | CorrelationAlert)?, AdditionalData*
  )>
<!ATTLIST Alert
    ident          CDATA           '0'
  >
```

The Heartbeat Class consists of information that will relay the status of a sensor to the managers. Heartbeats are sent at regular periods, for example every 15 minutes or hourly. The classes that make up Heartbeats are:

Analyzer:
Identifies the analyzer that generated the heartbeat.

CreateTime:
The time the heartbeat was created.

AnalyzerTime:
The current time on the analyzer.

AdditionalData:
Information included by the analyzer that does not fit into the data model.

```
<!ELEMENT Heartbeat                (
    Analyzer, CreateTime, AnalyzerTime?, AdditionalData*
  )>
<!ATTLIST Heartbeat
    ident          CDATA           '0'
  >
```

The following figure depicts the IDMEF Data Model:

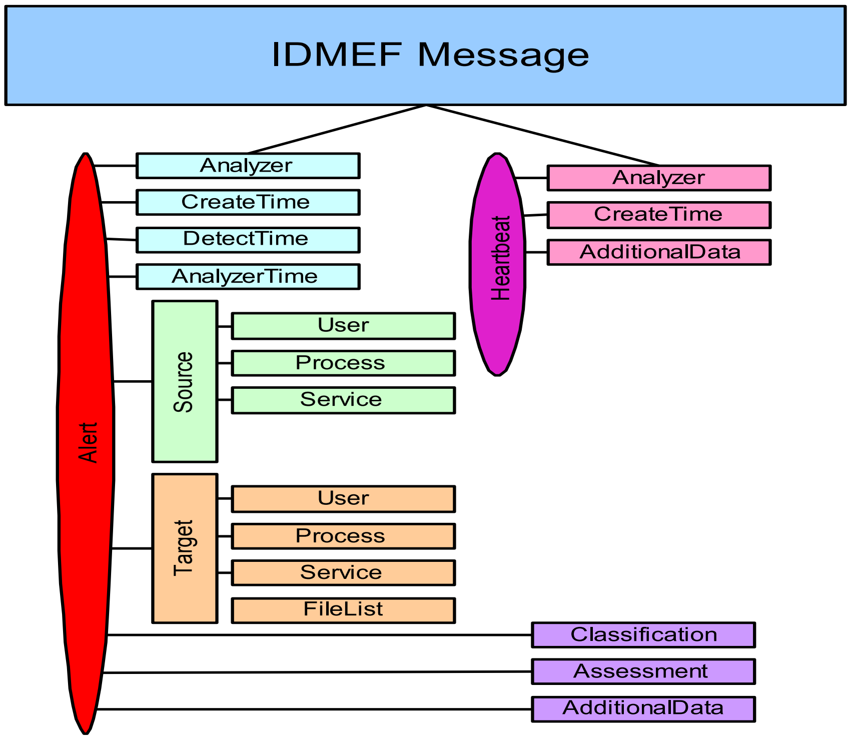


Figure 2

Summary

A standard way of exchanging intrusion detection alert information has been a long time coming, but there is evidence that this is changing. Many vendors have accepted the IDMEF format and are implementing this format in security management solutions. As more and more vendors get on boards the benefits realized will be great, as correlation of data from various sources will allow designs that include the best tool for the job.

References

The World Wide Web Consortium (W3C) 19 Nov. 2002 URL:
http://www.w3c.org (19 Nov 2002)

"DTD Tutorial." w3schools URL:
http://www.w3schools.com/dtd/ (23 Nov. 2002)

"DOM Validate XML." w3schools URL:
http://www.w3schools.com/dom/dom_validate.asp (23 Nov. 2002)

Kay, Michael. "About SAXON". 3 July 2001. URL:
http://users.iclway.co.uk/mhkay/saxon/instant.html (23 Nov. 2002)

Kay, Michael "How does XSLT transform XML." 20 Feb. 2001 URL:
http://tutorials.freeskills.com/read/category/84/id/143/p/2 (24 Nov. 2002)

Core Working Group "Extensible Markup Language (XML) 1.0 (Second Edition)" 6 Oct. 2000 URL: http://www.w3.org/TR/REC-xml#sec-well-formed (24 Nov. 2002)

Clark, James "James Clark's Home Page" URL:
http://www.jclark.com/. (24 Nov. 2002)

Wattle Software "Welcome to XMLwriter!" 1 Nov. 2002. URL:
http://xmlwriter.net/. (24 Nov. 2002).

IDWG "Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition" 20 June 2002 URL:
http://www.silicondefense.com/idwg/draft-ietf-idwg-idmef-xml-07.txt (24 Nov 2002)

Incident Object Description and Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition, IETF April 2002. URL
http://www.ietf.org/internet-drafts/draft-meijer-inch-iodef-00.txt. (24 Nov. 2002).

Pickel, Jed. "Enabling Automated Detection of Security Events that affect Multiple Administrative Domains" URL:
http://www.incident.org/thesis/x591.html#AEN607. (25 Nov 2002)

Briney, Andrew "Automating Policies" Oct. 2002 URL:
http://www.infosecuritymag.com/2002/oct/policytools.shtml (28 Nov. 2002)

Sidel, Scott "Intellitactics' Network Security Manager" Oct. 2020 URL:
http://www.infosecuritymag.com/2002/oct/testcenter.shtml (28 Nov. 2002)

Shipley, Greg "Dragon Claws its Way to the Top" 20 Aug. 2001 URL: http://www.networkcomputing.com/1217/1217f28.html?ls=NCJS_1217rt (29 Nov. 2002)

IDWG "Intrusion Detection Exchange Format 9 Oct. 2002 URL: http://www.ietf.org/html.charters/idwg-charter.html (29 Nov 2002).

Kothari, Pravin "Intrusion Detection Interoperability and Standardization" 19 Feb. 2002 URL: http://rr.sans.org/intrusion/interop.php (29 Nov 2002).

Staniford, Stuart; McAlerney, Joe "Libidmef and the IDMEF XML plugin for Snort" 15 Dec. 2000. URL: http://www.silicondefense.com/idwg/IDWG_122k.ppt (28 Nov. 2002)

Assignment 2 Network Detects

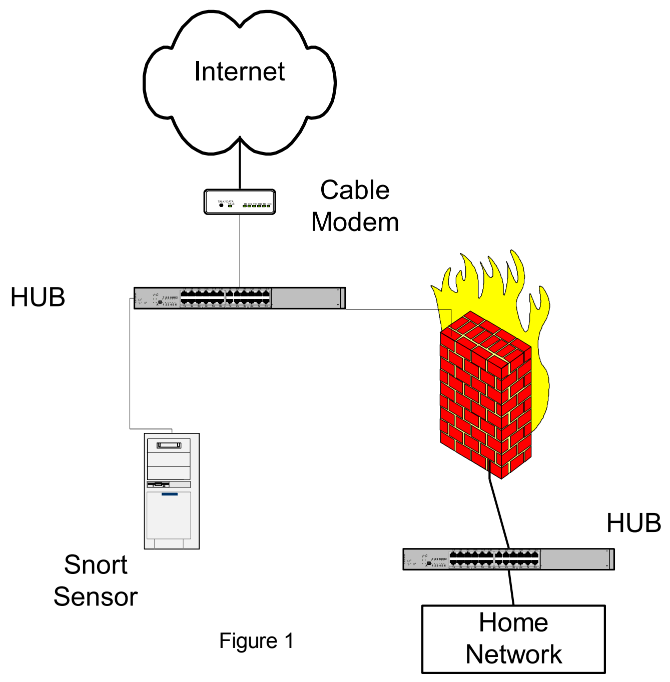


Figure 1

Detect 1

```
[**] [1:1256:7] WEB-IIS CodeRed v2 root.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/23-21:42:32.485164 24.42.15.56:4994 -> MY.NET.HOME.223:80
TCP TTL:120 TOS:0x0 ID:4587 IpLen:20 DgmLen:112 DF
***AP*** Seq: 0xABD15B62  Ack: 0xF0B38DAE  Win: 0x4470  TcpLen: 20
[Xref => http://www.cert.org/advisories/CA-2001-19.html]

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/23-21:42:32.776540 24.42.15.56:3003 -> MY.NET.HOME.223:80
TCP TTL:120 TOS:0x0 ID:4633 IpLen:20 DgmLen:120 DF
***AP*** Seq: 0xABD90A13  Ack: 0xF0B56491  Win: 0x4470  TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/23-21:42:32.955088 24.42.15.56:3013 -> MY.NET.HOME.223:80
TCP TTL:120 TOS:0x0 ID:4667 IpLen:20 DgmLen:120 DF
***AP*** Seq: 0xABE0A431  Ack: 0xF0B6CC1F  Win: 0x4470  TcpLen: 20

[**] [1:970:5] WEB-IIS multiple decode attempt [**]
[Classification: Web Application Attack] [Priority: 1]
09/23-21:42:33.070739 24.42.15.56:3032 -> MY.NET.HOME.223:80
TCP TTL:120 TOS:0x0 ID:4686 IpLen:20 DgmLen:136 DF
***AP*** Seq: 0xABED6558  Ack: 0xF0B7EB04  Win: 0x4470  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0333]

[**] [1:970:5] WEB-IIS multiple decode attempt [**]
[Classification: Web Application Attack] [Priority: 1]
09/23-21:42:33.198620 24.42.15.56:3037 -> MY.NET.HOME.223:80
TCP TTL:120 TOS:0x0 ID:4709 IpLen:20 DgmLen:157 DF
***AP*** Seq: 0xABF19473  Ack: 0xF0B97949  Win: 0x4470  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0333]

[**] [1:970:5] WEB-IIS multiple decode attempt [**]
[Classification: Web Application Attack] [Priority: 1]
09/23-21:42:33.348615 24.42.15.56:3041 -> MY.NET.HOME.223:80
TCP TTL:120 TOS:0x0 ID:4735 IpLen:20 DgmLen:157 DF
***AP*** Seq: 0xABF59DC1  Ack: 0xF0BA93D6  Win: 0x4470  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0333]

[**] [1:970:5] WEB-IIS multiple decode attempt [**]
[Classification: Web Application Attack] [Priority: 1]
09/23-21:42:33.476767 24.42.15.56:3047 -> MY.NET.HOME.223:80
TCP TTL:120 TOS:0x0 ID:4762 IpLen:20 DgmLen:185 DF
***AP*** Seq: 0xABFAE94B  Ack: 0xF0BBB763  Win: 0x4470  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0333]

[**] [1:1287:5] WEB-IIS scripts access [**]
[Classification: access to a potentially vulnerable web application]
[Priority: 2]
09/23-21:42:33.619095 24.42.15.56:3052 -> MY.NET.HOME.223:80
TCP TTL:120 TOS:0x0 ID:4788 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0xABFFA80E  Ack: 0xF0BCFBAC  Win: 0x4470  TcpLen: 20
```

```
[**] [1:1287:5] WEB-IIS scripts access [**]
[Classification: access to a potentially vulnerable web application]
[Priority: 2]
09/23-21:42:33.765345 24.42.15.56:3058 -> MY.NET.HOME.223:80
TCP TTL:120 TOS:0x0 ID:4809 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0xAC0457A6  Ack: 0xF0BDEEFA  Win: 0x4470  TcpLen: 20

[**] [1:1287:5] WEB-IIS scripts access [**]
[Classification: access to a potentially vulnerable web application]
[Priority: 2]
09/23-21:42:33.892489 24.42.15.56:3067 -> MY.NET.HOME.223:80
TCP TTL:120 TOS:0x0 ID:4834 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0xAC09BB4C  Ack: 0xF0BF1630  Win: 0x4470  TcpLen: 20

[**] [1:1287:5] WEB-IIS scripts access [**]
[Classification: access to a potentially vulnerable web application]
[Priority: 2]
09/23-21:42:34.016556 24.42.15.56:3074 -> MY.NET.HOME.223:80
TCP TTL:120 TOS:0x0 ID:4855 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0xAC0FC31F  Ack: 0xF0C0C91B  Win: 0x4470  TcpLen: 20

[**] [1:970:5] WEB-IIS multiple decode attempt [**]
[Classification: Web Application Attack] [Priority: 1]
09/23-21:42:34.136865 24.42.15.56:3081 -> MY.NET.HOME.223:80
TCP TTL:120 TOS:0x0 ID:4878 IpLen:20 DgmLen:138 DF
***AP*** Seq: 0xAC153345  Ack: 0xF0C22175  Win: 0x4470  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0333]

[**] [1:970:5] WEB-IIS multiple decode attempt [**]
[Classification: Web Application Attack] [Priority: 1]
09/23-21:42:34.258395 24.42.15.56:3085 -> MY.NET.HOME.223:80
TCP TTL:120 TOS:0x0 ID:4901 IpLen:20 DgmLen:136 DF
***AP*** Seq: 0xAC18A16F  Ack: 0xF0C341CA  Win: 0x4470  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0333]

[**] [1:970:5] WEB-IIS multiple decode attempt [**]
[Classification: Web Application Attack] [Priority: 1]
09/23-21:42:34.378373 24.42.15.56:3090 -> MY.NET.HOME.223:80
TCP TTL:120 TOS:0x0 ID:4916 IpLen:20 DgmLen:140 DF
***AP*** Seq: 0xAC1C7689  Ack: 0xF0C43B16  Win: 0x4470  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0333]

[**] [1:1287:5] WEB-IIS scripts access [**]
[Classification: access to a potentially vulnerable web application]
[Priority: 2]
09/23-21:42:34.510590 24.42.15.56:3093 -> MY.NET.HOME.223:80
TCP TTL:120 TOS:0x0 ID:4931 IpLen:20 DgmLen:136 DF
***AP*** Seq: 0xAC1F1ED3  Ack: 0xF0C5B7AB  Win: 0x4470  TcpLen: 20
```

1. Source of Trace:

This trace comes from a sensor I have on my cable modem as shown in Figure 1. On the same machine I have Back Officer Friendly (BOF) running and listening for FTP, Telnet, SMTP, HTTP, POP3, IMAP2 connections. BOF was setup to respond with fake replies. WinDump captures all data for use when more in depth analyses of traces are needed.

2. Detect was generated by:

The trace was generated by a Snort Sensor Version 1.8.7beta5-ODBC-WIN32 (Build 128). The sensor was using the latest stable release of the rule sets available at http://www.snort.org/dl/signatures/ . As a front-end to Snort I am using IDScenter. The 4 signatures that generated these alerts are listed below:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
IIS CodeRed v2 root.exe access"; flags:A+; uricontent:"/root.exe";
nocase; classtype:web-application-attack;
reference:url,www.cert.org/advisories/CA-2001-19.html; sid:1256;
rev:7

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
IIS cmd.exe access"; flags:A+; content:"cmd.exe"; nocase;
classtype:web-application-attack; sid:1002;  rev:5;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
IIS multiple decode attempt"; flags:A+; uricontent:"%5c";
uricontent:".."; reference:cve,CAN-2001-0333; classtype:web-
application-attack; sid:970;  rev:5;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
IIS scripts access"; flags:A+; uricontent:"/scripts/"; nocase;
classtype:web-application-activity; sid:1287;  rev:5;)
```

Explanation of the signature format that triggered the first alert:

```
alert – Alert and log packet when triggered.
tcp - Type of network traffic rule applies to.
$EXTERNAL_NET any  - Source IP and port.
-> - direction of traffic
$HTTP_SERVERS $HTTP_PORTS – Destination IP and port, in this case
they are both variables, which are set to ANY for $HTTP_SERVERS
and 80 for $HTTP_PORTS in the snort.conf file.
(msg:"WEB-IIS CodeRed v2 root.exe access" – message to be printed
in alerts and logs.
flags:A+ - Look for TCP flags, in this case an ACK field combined
with any other flag (+) sign.
uricontent:"/root.exe" – Specifies to look for /root.exe in the
URI portion of a packet. URI (Uniform Resource Identifier).
nocase – ignore case of string to be matched.
classtype:web-application-attack – A priority 1 attack that is
specified in the classification.config file.
reference:url,www.cert.org/advisories/CA-2001-19.html – location
of advisory posted at the CERT Coordination Center’s website.
sid:1256 – Snort rule ID
rev:7 – revision number of the signature.
```

3. Probability the source was spoofed:

LOW

1. The source address is probably not spoofed because:

As shown in the WinDump trace below the three-way handshake was established before the attack occurred, this would be very difficult to do with a spoofed IP.

2. TraceRoute to the source IP shows 8 hops, Windows by default has a TTL of 128 so a value of 120 for the TTL shown in the trace indicates that there are indeed 8 routers between the attacker and the sensor.
3. The CodeRed worm is very noisy and does not try to hide, when attacking random IP addresses.

```
21:42:32.436982 24.42.15.56.4994 > MY.NET.HOME.223.80: S
2882624353:2882624353(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
21:42:32.437185 MY.NET.HOME.223.80 > 24.42.15.56.4994: S
4038299053:4038299053(0) ack 2882624354 win 17520 <mss
1460,nop,nop,sackOK> (DF)
21:42:32.475838 24.42.15.56.4994 > MY.NET.HOME.223.80: . ack 1 win 17520
(DF)
21:42:32.485164 24.42.15.56.4994 > MY.NET.HOME.223.80: P 1:73(72) ack 1
win 17520 (DF)
```

4. Description of the attack:

The Nimda Worm seeks vulnerable systems, by attempting to connect to port 80 using random IPs. The crafted HTTP get requests exploit a buffer-overflow vulnerability CERT Advisory CA-2001-13 Buffer Overflow In IIS Indexing Service DLL, which allows the worm to run on your computer. CERT Advisory CA-2001-26 Nimda Worm describes this vulnerability. Nimda also attempts to exploit systems that have previously been infected with CodeRed II. CERT Advisory CA-2001-19 "Code Red" Worm describes this exploit.

5. Attack mechanism:

There are many steps to the attack, which will be explained below. The effects of this attack depends on what the system is running and on what day the system is attacked.

Once a three-way handshake is completed, crafted HTTP Get requests are sent as seen below from the log file created by Back Officer Friendly:

```
Mon Sep 23 21:42:32    HTTP request from 24.42.15.56: GET /scripts/root.exe?/c+dir
Mon Sep 23 21:42:32    HTTP request from 24.42.15.56: GET /MSADC/root.exe?/c+dir
Mon Sep 23 21:42:32    HTTP request from 24.42.15.56: GET
/c/winnt/system32/cmd.exe?/c+dir
Mon Sep 23 21:42:32    HTTP request from 24.42.15.56: GET
/d/winnt/system32/cmd.exe?/c+dir
```

```
Mon Sep 23 21:42:33    HTTP request from 24.42.15.56: GET
/scripts/..%255c../winnt/system32/cmd.exe?/c+dir
Mon Sep 23 21:42:33    HTTP request from 24.42.15.56: GET
/_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
Mon Sep 23 21:42:33    HTTP request from 24.42.15.56: GET
/_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
Mon Sep 23 21:42:33    HTTP request from 24.42.15.56: GET
/msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/system32/
cmd.exe?/c+dir
Mon Sep 23 21:42:33    HTTP request from 24.42.15.56: GET
/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir
Mon Sep 23 21:42:33    HTTP request from 24.42.15.56: GET
/scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir
Mon Sep 23 21:42:33    HTTP request from 24.42.15.56: GET
/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir
Mon Sep 23 21:42:34    HTTP request from 24.42.15.56: GET
/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir
Mon Sep 23 21:42:34    HTTP request from 24.42.15.56: GET
/scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir
Mon Sep 23 21:42:34    HTTP request from 24.42.15.56: GET
/scripts/..%%35c../winnt/system32/cmd.exe?/c+dir
Mon Sep 23 21:42:34    HTTP request from 24.42.15.56: GET
/scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir
Mon Sep 23 21:42:34    HTTP request from 24.42.15.56: GET
/scripts/..%252f../winnt/system32/cmd.exe?/c+dir
```

The first four entries in the BOF log show attempts to connect to the backdoor left by Code Red II, while the remaining log entries are examples of exploit attempts for the Directory Traversal vulnerability. Directory Traversal is the ability to execute code in a directory that was not intended to be accessed by a remote user. The code is executed with higher privileges. For example in the case of an Internet Information Servers (IIS), the code would be executed with the privileges of the IUSR_*machinename* account. Vulnerability Note VU#111677 explains this in greater detail. The CVE group has assigned this identifier to this vulnerability, CVE-2000-0884.

This attack would be considered a stimulus since the Nimda code has randomly targeted my system. The service that is being targeted is port 80. This service has known vulnerabilities and can be exploited when it is an unpatched default installation of IIS.

6. Correlation:

CERT® Advisory CA-2001-26 Nimda Worm. This advisory reports that Nimda has been around since 18 Sept, 2001. The footprint shown in the advisory is same as what my sensor and Back Officer Friendly logged.

Footprint:

```
GET /scripts/root.exe?/c+dir
GET /MSADC/root.exe?/c+dir
GET /c/winnt/system32/cmd.exe?/c+dir
GET /d/winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET
/msadc/..%5c../..%5c../..%5c/..\xc1\x1c../..\xc1\x1c../..\xc1\x1c../winnt/system32/cmd.exe
?/c+dir
GET /scripts/..\xc1\x1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0/../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0\xaf../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\x9c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%2f../winnt/system32/cmd.exe?/c+dir
```

A search on the source IP at DShield's IP Info site returned the following info:

IP Address: 24.42.15.56

HostName: CPE014400123433.cpe.net.cable.rogers.com

DShield Profile:

Country:	US
Contact E-mail:	abuse@rogers.com
Total Records against IP:	393
Number of targets:	195
Date Range:	2002-11-07 to 2002-11-10

Ports Attacked (up to 10):

Port	**Attacks**
80	13

7. Evidence of active targeting:

It is well known that Nimda targets random IP's. The rough probabilities stated in CERT® Advisory CA-2001-26 are:

- 50% of the time, an address with the same first two octets will be chosen
- 25% of the time, an address with the same first octet will be chosen
- 25% of the time, a random address will be chosen

With this said the Destination IP was probably generated by the code of an already infected system. Therefore this is not likely active targeting.

8. Severity:

Formula used to calculate Severity:
Severity = (criticality + lethality) – (system countermeasures + network countermeasures)

Criticality	This my personal system, explicitly built for gathering network traffic on outside of my firewall.	1
Lethality	The lethality would be high, if the system had been compromised. Allowing a remote user to access files and folders outside of the web folder.	4
System Countermeasure	There are no system countermeasures in place except for Service Pack 2	2
Network Countermeasure	There are no network countermeasures in place. (Outside of the firewall)	1

Severity = 2

9. Defense recommendation:

For the defense recommendations I will assume that I was running IIS 5.0.

1) Place the system running IIS behind a firewall for example in a DMZ.
2) A tool for removing Nimda is available at Symantec's site.
3) Install the latest security patches.
4) Install antivirus software on all systems.

10. Multiple choice question:

What is Directory Traversal? Select the best answer.

a) Moving from one directory to another.
b) Accessing files and folders that reside on the same logical drive as the web folders
c) Renaming a Directory
d) Deleting a Directory

Answer: b
An explanation of this can be found at Microsoft's Security Bulletin (MS00-078).

References

Erdelyi Gergely, Rautiainen Sami & Hypponen Mikko "F-Secure Virus Descriptions" July-August, 2001]
URL http://www.europe.f-secure.com/v-descs/bady.shtml (30 Sept. 2002)

Chien, Eric "CodeRed Worm" 19 July 2002. URL:
http://securityresponse.symantec.com/avcenter/venc/data/codered.worm.html (30 Sept 2002)

CERT® Advisory CA-2001-26 Nimda Worm 25 Sept 2001 URL:
http://www.cert.org/advisories/CA-2001-26.html (01 Sept. 2002)

CERT® Advisory CA-2001-13 Buffer Overflow In IIS Indexing Service DLL 17 Jan 2002 URL: http://www.cert.org/advisories/CA-2001-13.html (01 Sept. 2002)

CERT® Advisory CA-2001-19 "Code Red" Worm 17 Jan 2002 URL:
http://www.cert.org/advisories/CA-2001-19.html(01 Sept. 2002)

Vulnerability Note VU#111677 "Microsoft IIS 4.0 / 5.0 vulnerable to directory traversal via extended unicode in url" 18 Sept 2001 URL:
http://www.kb.cert.org/vuls/id/111677 (08 Oct 2002)

CVE-2000-0884 Common Vulnerabilities and Exposures 22 Jan 2001 URL:
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0884 (08 Oct 2002)

DShield.org
http://www.dshield.org/ipinfo.php (01 Nov 2002)

Microsoft Security Bulletin (MS00-078) 17 Oct 2000 URL:
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms00-078.asp (17 Nov 2002)

Ferrie, Peter W32.Nimda.A@mm Removal Tool25 July 2002 URL:
http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.a@mm.removal.tool.html (18 Nov. 2002)

Detect 2

```
[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/28-06:07:52.771195 0:0:77:94:7F:40 -> 0:40:5:E2:B8:9B type:0x800
len:0x3E
211.102.105.87:1988 -> MY.NET.HOME.212:8080 TCP TTL:99 TOS:0x0 ID:38467
IpLen:20 DgmLen:48 DF
******S* Seq: 0x1EA8EB Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

```
[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/28-06:07:53.545843 0:0:77:94:7F:40 -> 0:40:5:E2:B8:9B type:0x800
len:0x3E
211.102.105.87:1988 -> MY.NET.HOME.212:8080 TCP TTL:99 TOS:0x0 ID:43331
IpLen:20 DgmLen:48 DF
******S* Seq: 0x1EA8EB Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

```
[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/28-06:07:54.343015 0:0:77:94:7F:40 -> 0:40:5:E2:B8:9B type:0x800
len:0x3E
211.102.105.87:1988 -> MY.NET.HOME.212:8080 TCP TTL:99 TOS:0x0 ID:47683
IpLen:20 DgmLen:48 DF
******S* Seq: 0x1EA8EB Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

```
[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/28-06:07:55.147298 0:0:77:94:7F:40 -> 0:40:5:E2:B8:9B type:0x800
len:0x3E
211.102.105.87:1988 -> MY.NET.HOME.212:8080 TCP TTL:99 TOS:0x0 ID:54595
IpLen:20 DgmLen:48 DF
******S* Seq: 0x1EA8EB Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

1. Source of Trace:

This trace comes from a sensor I have on my cable modem as shown in Figure 1 show above. On the same machine I have Back Officer Friendly (BOF) running and listening for FTP, Telnet, SMTP, HTTP, POP3, IMAP2. I had also setup BOF to respond with fake replies. I then ran SnortSnarf on the alert.ids file created by Snort to generate the output shown above. I then sanitized my IP to MY.NET.HOME.

2. Detect was generated by:

The trace was generated by a Snort Sensor Version 1.8.7beta5-ODBC-WIN32 (Build 128). The sensor was using the latest stable release of the rule sets available at http://www.snort.org/dl/signatures/ . As a front-end to Snort I am using IDSCenter. The signature that generated this alert is shown below:

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

Explanation of the signature format that triggered the alert:

```
alert - Alert and log packet when triggered.
tcp - Type of network traffic rule applies to.
$EXTERNAL_NET any  - Source IP and port.
-> - direction of traffic
$HOME_NET 8080 - Destination IP and port,
(msg:"SCAN Proxy \(8080\) attempt" - message to be printed in
alerts and logs.
flags:S - Look for TCP flags, in this case the SYN field.
classtype:attempted-recon - A priority 2 attack that is specified
in the classification.config file.
sid:615 - Snort rule ID
rev:3 - revision number of the signature.
```

3. Probability the source was spoofed:

LOW

The source address is probably not spoofed because:

1) The initiator of this traffic is looking for a response.
2) I also believe that it is not spoofed since this is TCP traffic, which usually requires a three-way handshake.
3) TraceRoute to the source IP shows 27 hops, Windows by default has a TTL of 128 so a value of 99 for the TTL shown in the trace indicates that there are roughly 27 routers between the attacker and the sensor.

4. Description of the attack:

This scan may be probing for proxy servers for example WinGate. If Proxy servers are not configured properly, they might allow attackers to use them to access the Internet. If an attacker is able to use someone else's proxy he/she can perform other attacks anonymously. It would appear to a victim that the attack is coming from the proxy and not the attacker's system, thus making it difficult to track the attacker. An attacker may even chain several proxies making it even more difficult to trace the attack. There are many sites listing Open Proxies, just perform a search using key words **proxy** and **list.** IRC servers will perform an Open Proxy Scan on any system that attempts to connect to it, in order to prevent this type of anonymous connection.

More information on Open Proxy Scans by IRC can be found at help.undernet.org/. The attack could also be a reconnaissance scan looking for vulnerable systems that may be used in the future.

The RingZero Trojan is also known to scan ports 80, 8080 and 3128 (Squid Proxy). More information can be found at CA Virus information Center. I have ruled out RingZero as a possible source as there are no indications of scans of ports 80 or 3128. If that was the case, then the following Snort rule would have triggered an alert:

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

5. Attack mechanism:

Although looking for Open HTTP Proxies can be done manually using telnet, it is highly unlikely that this attack was executed in this manner. The trace shows 4 alerts occurring within 1 second of each other; this would be nearly impossible to pull off at the keyboard.
The steps provided below show how one might scan for Open Proxies manually: Proxies:

From a command line type:

- telnet open.proxy 8080
- GET http://www.somesite.com HTTP/1.
- Press enter twice (If www.somesite.com exists and open.proxy is indeed open www.somesite.com will be returned.

If the above method was used it was most likely scripted.

Another method for this type of scan would be to use an Open Proxy Scanning tool. There are many such tools available on the Internet such as CPT (cum proxy toolkit). A list of Open Proxies compiled with CPT can be found here. Another such tool is ProxyCheck. ProxyCheck is a command line tool that allows you to specify one or more ports to scan as well as one or more IPs. Another popular scanner is wGateScan v2.2, which is a really easy GUI tool to use and can be found here. Within the GUI you can specify a string to send once connected to a port Example: **GET http://www.yahoo.com HTTP/1.0.** This is a scan reconnaissance attack and should be considered a stimulus.

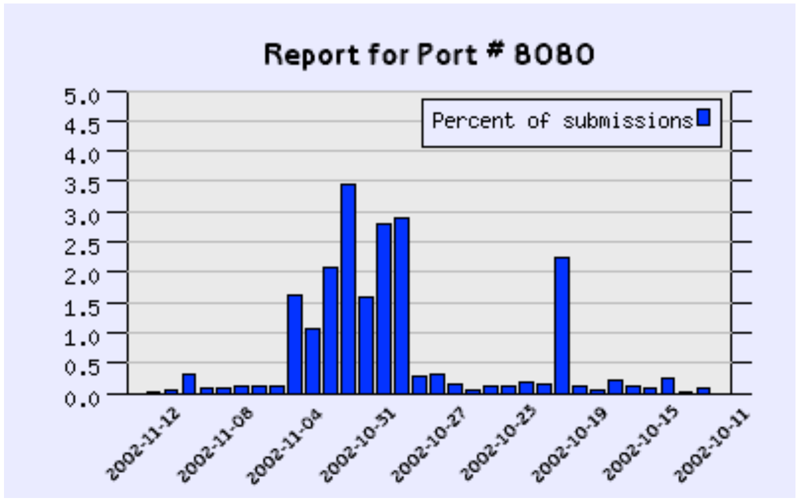
6. Correlation:

As well as the alert that Snort generated, Back Officer Friendly also logged the following alert:

Mon Oct 28 06:07:53 HTTP request from 211.102.105.87: GET http://www.yahoo.com/

Joanne Treurniet has reported similar traffic in her practical.

A search on the Destination port 8080 at DShield.org Port Information returned the following info:



A search on the source IP at DShield's IP Information site returned the following info:

IP Address: 211.102.105.87

HostName: 211.102.105.87

DShield Profile:

Country:	CN
Contact E-mail:	mchen_AT_capitalnet.com.cn (bounced)
Total Records against IP:	2
Number of targets:	1
Date Range:	2002-10-27 to 2002-10-27

Ports Attacked (up to 10):

Port	Attacks

7. Evidence of active targeting

I do not believe that the alert was the result of active targeting, since this type of scan is quite common nor am I able find any other evidence of other attacks against my network from the same IP. There is also evidence that attack was scripted and that this was simply a reconnaissance scan looking for Open Proxies.

8. Severity:

Formula used to calculate Severity:
Severity = (criticality + lethality) – (system countermeasures + network countermeasures)

Criticality	This my personal system, explicitly built for gathering network traffic on outside of my firewall.	1
Lethality	The lethality would be medium. Since it could appear that malicious traffic is coming from this system.	3
System Countermeasure	There are no system countermeasures in place.	2
Network Countermeasure	There are no network countermeasures in place. (Outside of the firewall)	1

Severity = 1

9. Defense recommendation:

1) It is recommended that port 8080 as well as any other unnecessary ports be blocked from incoming traffic.
2) Proxy servers that are running should be configured to only allow calls from systems residing inside of your network. Instructions on how to secure WinGate can be found here.
3) It is also recommended that a scan of your own network be performed looking for Open Proxies, to make sure no other proxies are open.

10. Multiple choice question:

What Trojan(s) are often associated with port 8080? Select one or more.

a) Deep Throat
b) WinHole
c) SpySender
d) RingZero

Answer: d

Deep Throat is usually associated with Ports 41, 999, 2140 (UDP), 3150 (UDP), 6670, 6771 and 60000.
WinHole is usually associated with Ports 808, 1080, 1081, 1082, 1083 and 2080.
SpySender is usually associated with Port 1807.

A great listing of Ports and which Trojan(s) use them can be found here.

References

DShield.org 11 Nov 2002 URL:
http://www.dshield.org/ipinfo.php (11 Nov 2002)

DShield.org 13 Nov 2002 URL:
http://www.dshield.org/port_report.php (13 Nov 2002)

Computer Associates Virus Information Center 7 Nov 2002 URL:
http://www3.ca.com/virusinfo/Virus.asp?ID=9790 (8 Nov 2002)

Undernet.org Undernet Scans for Insecure Proxies 2000 URL:
http://help.undernet.org/proxyscan/

BlackHat.be Cum Proxy Toolkit URL:
http://www.blackhat.be/cpt/

BlackHat.be Proxy List URL:
http://www.blackhat.be/cpt/proxy.lst

Von Braun, Joakim. Sans Institute What port numbers do well-known Trojan horses use.
URL: http://www.sans.org/newlook/resources/IDFAQ/oddports.htm

Detect 3

```
[**] DNS zone transfer [**]
07/08-09:45:05.504488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x5E
211.21.238.234:1026 -> 46.5.180.250:53 TCP TTL:44 TOS:0x0 ID:15422
IpLen:20 DgmLen:80 DF
***AP*** Seq: 0xAEFBEA36  Ack: 0x9616BE35  Win: 0x7D78  TcpLen: 32
TCP Options (3) => NOP NOP TS: 6464873 622332321
00 1A 19 8F 00 00 00 01 00 00 00 00 00 00 04 58  ...............X
58 58 58 03 63 6F 6D 00 00 FC 00 01              XXX.com.....
```

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

[**] DNS zone transfer [**]
07/08-09:45:07.594488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800
len:0x5E
211.21.238.234:1026 -> 46.5.180.250:53 TCP TTL:44 TOS:0x0 ID:15423
IpLen:20 DgmLen:80 DF
***AP*** Seq: 0xAEFBEA36  Ack: 0x9616BE35  Win: 0x7D78  TcpLen: 32
TCP Options (3) => NOP NOP TS: 6465083 622332321
00 1A 19 8F 00 00 00 01 00 00 00 00 00 00 04 58  ...............X
58 58 58 03 63 6F 6D 00 00 FC 00 01              XXX.com.....

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

1. Source of Trace:

This trace was obtained from http://www.incidents.org/logs/Raw/2002.6.8 .

2. Detect was generated by:

The tcpdump binary log files I downloaded were generated by a Snort ruleset. I then ran Snort Sensor Version 1.8.7beta5-ODBC-WIN32 (Build 128) on the binary log. Snort was ran using the latest stable release of the rule sets available at http://www.snort.org/dl/signatures/ . I then parsed the alert.ids with SnortSnarf. The signature that generated these alerts is:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS zone
transfer"; flags:A+; content: "|00 00 FC|"; offset:13;
reference:cve,CAN-1999-0532; reference:arachnids,212;
classtype:attempted-recon; sid:255;  rev:6;)
```

Explanation of the signature format that triggered the alerts:

alert – Alert and log packet when triggered.
tcp - Type of network traffic rule applies to.
$EXTERNAL_NET any - Source IP and port.
-> - direction of traffic
(msg:"DNS zone transfer" message to be displayed in alerts and logs.
flags:A+ - Look for TCP flags, in this case an ACK field combined with any other flag.
content: "|00 00 FC|" – search for this content in the payload.
offset:13 – sets the offset, where to begin pattern matching.
reference:cve,CAN-1999-0532; reference:arachnids,212 – for more info refer to these references.
classtype:attempted-recon – A priority 2 attack that is specified in the classification.config file.
sid:255 – Snort rule ID
rev:6 - revision number of the signature.

Note about search pattern:

Request for Comment 1035 (RFC 1035) states that a Query type of 252 is a request for a transfer of an entire zone. The pattern “00 00 FC” is 252 in hex.

3. Probability the source was spoofed:

LOW
The source address is probably not spoofed because:
The source IP is most likely not spoofed. The transfer of zone information requires an established TCP session (three way handshake), which would be difficult to do with a spoofed IP.
The source IP is part of a range that is registered to APNIC (Asia Pacific Network Information Centre) and is therefore valid, a search at www.apnic.net on IP 211.21.238.234 returned the following information:

```
inetnum:      211.21.0.0 - 211.21.255.255
netname:      HINET-TW
descr:        CHTD, Chunghwa Telecom Co.,Ltd.
descr:        Data-Bldg.6F, No.21, Sec.21, Hsin-Yi Rd.
descr:        Taipei Taiwan 100
country:      TW
admin-c:      HN27-AP
tech-c:       HN28-AP
remarks:      This information has been partially mirrored by APNIC
from
remarks:      TWNIC. To obtain more specific information, please use
the
remarks:      TWNIC whois server at whois.twnic.net.
mnt-by:       TWNIC-AP
changed:      hostmaster@twnic.net 20000707
status:       ALLOCATED PORTABLE
source:       APNIC
person:       HINET Network-Adm
address:      CHTD, Chunghwa Telecom Co., Ltd.
address:      Data-Bldg. 6F,  No. 21, Sec. 21, Hsin-Yi Rd.,
address:      Taipei Taiwan 100
country:      TW
phone:        +886 2 2322 3495
phone:        +886 2 2322 3442
phone:        +886 2 2344 3007
fax-no:       +886 2 2344 2513
fax-no:       +886 2 2395 5671
e-mail:       network-adm@hinet.net
nic-hdl:      HN27-AP
remarks:      same as TWNIC nic-handle HN184-TW
mnt-by:       TWNIC-AP
changed:      hostmaster@twnic.net 20000721
source:       APNIC
person:       HINET Network-Center
address:      CHTD, Chunghwa Telecom Co., Ltd.
address:      Data-Bldg. 6F,  No. 21, Sec. 21, Hsin-Yi Rd.,
address:       Taipei Taiwan 100
country:       TW
phone:         +886 2 2322 3495
phone:         +886 2 2322 3442
phone:         +886 2 2344 3007
fax-no:        +886 2 2344 2513
fax-no:        +886 2 2395 5671
e-mail:        network-center@hinet.net
nic-hdl:       HN28-AP
remarks:       same as TWNIC nic-handle HN185-TW
mnt-by:        TWNIC-AP
changed:       hostmaster@twnic.net 20000721
source:        APNIC
inetnum:       211.21.238.232 - 211.21.238.239
netname:       ASE-TEST.-IN-KH-NET
descr:         Ase Test. Inc.
descr:         No. 10,  Shi 5th St., Kaohsiung
descr:         Kaohsiung Taiwan
country:       TW
admin-c:       SC100-TW
tech-c:        SC100-TW
mnt-by:        TWNIC-AP
remarks:       This information has been partially mirrored by APNIC
from
remarks:       TWNIC. To obtain more specific information, please use
the
remarks:       TWNIC whois server at whois.twnic.net.
changed:       network-adm@hinet.net 20010802
source:        TWNIC
person:        Sidney Chen
address:       Ase Test. Inc.
address:       No. 10,  Shi 5th St., Kaohsiung
address:       Kaohsiung Taiwan
country:       TW
phone:         +886-7-336-0349
fax-no:        +886-7-366-0352
e-mail:        sidneychen@smsc.com
nic-hdl:       SC100-TW
remarks:       This information has been partially mirrored by APNIC
from
remarks:       TWNIC. To obtain more specific information, please use
the
remarks:       TWNIC whois server at whois.twnic.net.
changed:       hostmaster@twnic.net 20010802
source:        TWNIC
```

4) Description of the attack:

This is a reconnaissance attack that is attempting to transfer a zone file. The zone file can then be used to map a network. The Common Vulnerabilities and Exposure group has assigned this vulnerability identification number CAN-1999-0532 but it is still under review. A description of the attack can be found at URL: http://whitehats.com/IDS/IDS212.

5) Attack mechanism:

Once an attacker has determined that a system is a DNS server, the attacker simply requests all of the records for a given zone. Finding a DNS servers is easy, all one has to do is use whois at InterNIC. There are many tools available for this kind of requests such as nslookup, dig and host to name a few of the more popular ones. Examples of the various methods of querying a DNS server can found at URL:
http://docsrv.caldera.com/NET_tcpip/dnsC.nslook.html . This is a reconnaissance attack and should be considered a stimulus. The attack targets port 53 and unless the proper safeguards are in place a complete mapping of all host to IP's can be obtained for a given DNS domain.

6) Correlation:

Mark Thyer has submitted a similar attack in the In the Intrusion Signatures and Analysis book page 159. J.D. Baldwin has analyzed a similar attack as shown in his practical. http://www.giac.org/practical/JD_Baldwin.html . This attack is also described in the Intrusion Detection In-Depth course material.

7) Evidence of active targeting:

I believe that this specific host was being targeted, as there were no other systems with zone transfer requests logged. The logs I acquired cover these times 07/07-20:04:16 - 07/08-19:56:29, almost 24 hrs in fact there was only one other zone transfer request logged during this period from one other host to the same IP.

8) Severity:

Formula used to calculate Severity:
Severity = (criticality + lethality) – (system countermeasures + network countermeasures)

Criticality	The criticality of a DNS server usually high.	5
Lethality	Having all DNS records for a given domain could be very damaging. Seeing as this is a reconnaissance attack.	5
System Countermeasure	There are no records of a successful zone transfer in the trace, therefore the system most likely has countermeasures in place.	1

Network Countermeasure	Because there needs to be a TCP three way handshake established for a zone transfer takes place it is not likely that network countermeasures are in place.	2

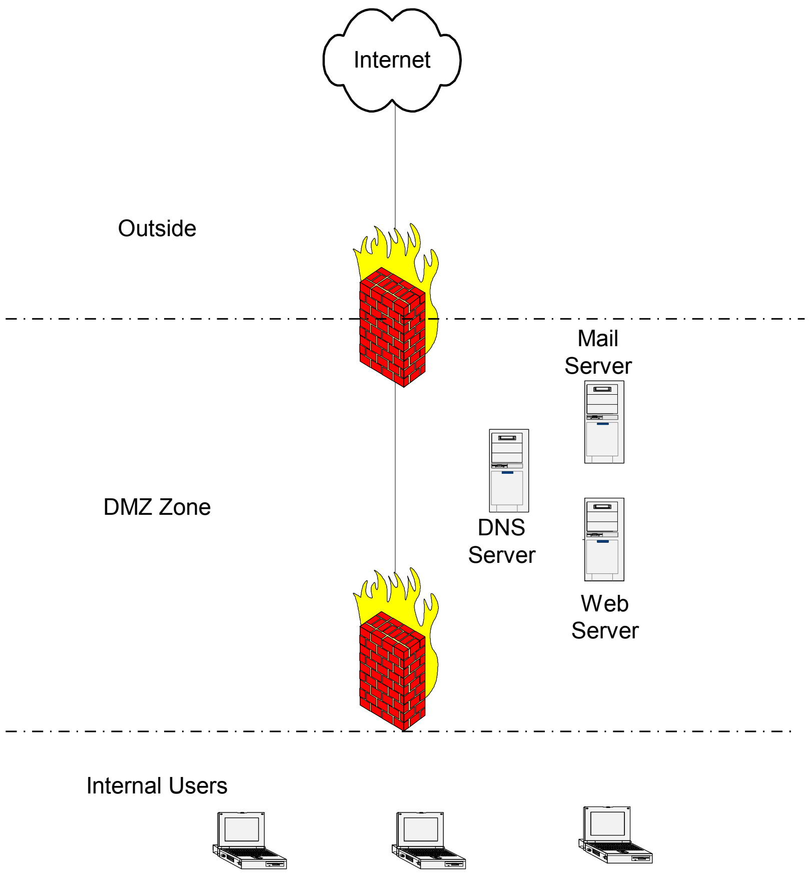
Severity=8

9) Defense recommendation:

1. It is recommended that the DNS server only allow zone transfers with trusted hosts. You may need to upgrade to a newer version of to take advantage of this option, where you can specify trusted secondary DNS servers. Bind Version 4.9.3 and above have the option to restrict what systems or networks are permitted perform successful zone transfers. Windows NT/2000 DNS servers can both be configured to restrict what systems are permitted to request zone file transfers; this is controlled by listing secondary DNS servers in the Notify list. Note by default Window NT allows zone transfer requests from any clients.

2. One other defense mechanism is to block TCP port 53 traffic from coming in via a firewall. DNS queries are UDP unless the data being returned is greater than 484 bytes. The 484 bytes limit of data come from the fact that a DNS UDP response is limited to 512 bytes total (512 – 20 IP header – 8 UDP header = 484 bytes). This may prevent large queries from being successful but the added security will probably out weight this problem. "If this is your only option, it is preferable to prevent the zone transfer even at the expense of blocking other legitimate data." Judy Novak.

3. Another recommended defense it to implement split DNS. Split DNS means to split the functionality of DNS on two different servers. One name server would reside in the DMZ (Demilitarized Zone) and would service name resolution requests from the internet and another name server on the inside of the network that would provide name resolution for internal users. As shown in the drawing below this design would allow External users name resolution in order to access services like Mail and Web servers and keep the name server that Internal users would use in a more secure area.



1. Multiple choice question:

Looking at the TTL 44 from the trace above, what is the most likely OS from which the attack was generated? Assume that the packet is not crafted.

a) Window 98
b) Windows 2000
c) Linux 2.4
d) Solaris 7.0

Answer: c

Most sites or services on the Internet are rarely more that 30 to 40 hops away. With this said, the Linux 2.4 kernel has a TTL of 64, Windows have a TTL of 128 and Solaris 7 has a TTL of 255 therefore the most likely the source of the attack was using Linux 2.4.

Top Three Questions:

Analysis was submitted 27 October 2002.

1. How did the server respond to the request?

The DNS server did not respond or reply to the zone transfer request or the Snort ruleset were not triggering on successful zone transfers. Using the raw data log I ran Windump with the following command:

```
windump -n -X -r c:\2002.6.8 ip src 46.5.180.250 and ip dst 211.21.238.234> dns.txt .
```

Running this command returned no record of any other communication with the source of the zone transfer.

2. What should be the source port be when requesting zone transfers between DNS servers?

   The answer to your question depends on what version of Bind the name server is running. Looking at the table below shows that newer versions of bind do use ports numbers above 1023. I would also like to add that Bind Version 8.x.x can be configured to use a specific port for example port 53 like previous version of Bind.

	Proto	Source	Destination	Use
Bind Version 4.x.x	UDP	53	53	Queries and replies between Name servers
	TCP	53	53	Zone Transfers and queries with large replies
Bind Version 8.x.x	UDP	>1023	53	Queries and replies between Name servers
	TCP	>1023	53	Zone Transfers

3. How much of the network can the attacker map?

   The attacker would only be able to map systems that have DNS records. With this said often the systems that do have DNS records are mission critical. For example mail servers and web servers. Many networks use Wins servers for most of their name resolutions but still have a need for DNS name resolution for critical systems.

References

RFC 1035 “DOMAIN NAMES IMPLEMENTATION AND SPECIFICATION” Nov. 1987 URL: http://www.ietf.org/rfc/rfc1035.txt?number=1035. (08 Oct 2002)

Whitehats IDS212 "DNS-ZONE-TRANSFER” URL: http://www.whitehats.com/IDS/212 (08 Oct 2002)

Northcutt Stephen, Cooper Mark, Fearnow Matt, Frederick Karen. Intrusion Signatures and Analysis. Indianapolis: New Riders Publishing, 2001.

Northcutt Stephen. Network Intrusion Detection An Analyst’s Handbook. Indianapolis: New Riders Publishing, 2000.

Using nslookup, dig, and host
http://docsrv.caldera.com/NET_tcpip/dnsC.nslook.html (08 Oct 2002)

InterNIC 14 Sept 2001 URL:
http://www.internic.net/cgi-bin/whois (10 Oct 2002)

CVE-1999-0532 Common Vulnerabilities and Exposures 26 Jul 1999 URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0532 (10 Oct 2002)

MPRM Group Limited “DNS ports” URL:
http://screamer.mobrien.com/Manuals/MPRM_Group/dns_notes.html (11 Oct 2002)

Microsoft Knowledge Base “Windows NT 4.0 DNS Server Default Zone Security Settings” 24 Oct 2002 URL:
http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q193837&. (2 Nov 2002)

Asia Pacific Network Information Centre
http://www.apnic.net/apnic-bin/whois.pl. (2 Nov 2002)

Assignment 3 Analyze This

Executive Summary:

We have been tasked by GIAC University to investigate and produce a Security Analysis due to the high number of alerts generated between the 11th and 15th of November. The data was obtained at www.incidents.org/logs , the logs provided consisted of three Snort log types Alert, Out of Spec and scan logs. Since the mandate of this audit was to investigate the increase of alerts during this period, the focus of the analysis was on the Top Talkers. Once the Top Talkers had been identified with respect to Alerts, the attackers were further investigated by looking for correlating data in the OOS and Scan logs. It is interesting to note that the top talker IP 24.59.33.240 (**possible Red Worm – traffic)** that generated the most alerts over the 5-day period has dropped significantly during the last two days of this period. We have provided a detailed analysis for each of the Top 6 alerts with recommended actions with in each analysis.

List of Files for the Analysis:

The dates of the log files are from Oct 11th to Oct 15th 2002. The following 3 types of logs were received:

Scan logs:

scans.021011
scans.021012
scans.021013
scans.021014
scans.021015

Scan logs are generated by Snort's preprocessor "Portscan Detector" and are triggered by X number of ports scans within Y number seconds. The snort.conf configuration file allows the setting of X and Y values. The logs show source and destination IP as well as type of scan.

Alert logs:

alert.021011
alert.021012
alert.021013
alert.021014
alert.021015

The Snort Intrusion Detection System generates alert logs. Alerts are triggered whenever network traffic matches Snort's signature.

Out of Spec (OOS) logs:

OOS_Report_2002_10_11_21861.txt
OOS_Report_2002_10_12_21861.txt
OOS_Report_2002_10_13_21861.txt
OOS_Report_2002_10_14_21861.txt
OOS_Report_2002_10_15_21861.txt

(OOS) Out Of Spec packets

Out of spec traffic generated by the following three methods

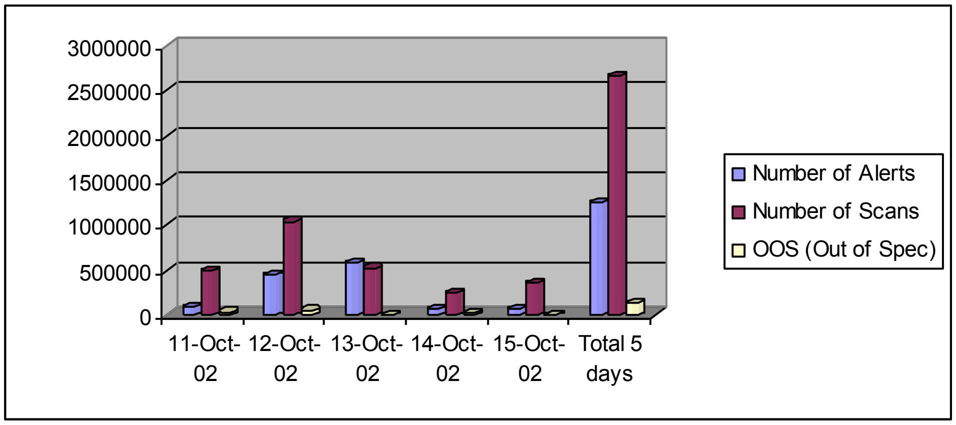
1. Corrupted packets which are corrupted at the source or enroute by intermediary systems I.E. routers
2. Crafted packets with the intent of port scanning and OS fingerprinting, or
3. The use of the Explicit Congestion Notification (ECN) see (**RFC2481**) for more information on ECN.

Note: The log files were generated by the Snort with the Alert mode set to **Fast** Which generates much less data than if the mode has been set to **Full.** One disadvantage is though is less detail is available in the logs for analysis.

The following table is a breakdown of the total Alerts, Scans and Out of Specs detected for each day, as well as the totals over a 5 day period for each type of detect.

Date	Number of Alerts	Number of Scans	OOS (Out of Spec)
11-Oct-02	94255	498899	4585
12-Oct-02	441945	1044981	7892
13-Oct-02	581052	524993	554
14-Oct-02	67352	245330	4028
15-Oct-02	63009	355095	509
Total 5 days	1247613	2669298	17568

The following graph is a graphical representation of the Alerts, Scans and Out of Specs detected for each day, as well as the totals over a 5 day period for each type of detect.



List of Detects:

List of detects prioritized by number of occurrences, with a brief description for each detect:

Signature	# Alerts	Short Description
High port 65535 tcp - possible Red Worm - traffic	752075	Possible Red Worm tcp Red Worm traffic.
PORTSCAN DETECTED	343979	Any TCP or UDP sent to more than X number of ports in Y amount of time.
spp_http_decode: IIS Unicode attack detected	64169	Suspicious code sent to a Web Server
CS WEBSERVER - external web traffic	20890	HTTP Connection attempt to CS WEBSERVER from external IPs
Watchlist 000220 IL-ISDNNET-990517	20000	Alerts on Traffic from certain IPs in this case they are IPs from Israel.
SMB Name Wildcard	19252	Source is attempting to get NetBIOS Info.
SUNRPC highport access!	5487	Attempted to connect to Sun RPC older versions of Sun would allow viewing of files.
FTP DoS ftpd globbing	3734	A remote attacker may be attempting to crash the ftpd server software by sending a wildcard request to create a denial of service on vulnerable ftp servers.
SYN-FIN scan!	3063	SYN-FIN scan looking for active TCP ports
spp_http_decode: CGI Null Byte attack detected	2540	Suspicious commands sent (%00) to IIS Web server for malicious purposes.
Queso fingerprint	2353	User in using the Queso tool to fingerprint a system
IDS552/web-iis_IIS ISAPI Overflow ida nosize	2186	An attempt to exploit a vulnerability in IIS
High port 65535 udp - possible Red Worm - traffic	1934	Possible Red Worm UDP Red Worm traffic.
Incomplete Packet Fragments Discarded	1032	Fragmented packets be sent to internal systems from the external sources
External RPC call	964	An attempted RPC call from and external source
Watchlist 000222 NET-NCFC	962	Alerts on Traffic from certain IPs in this case its an IP assigned to the Computer Center Chinese Academy of Sciences.

EXPLOIT x86 NOOP	618	This indicate that a string of the character 0x90 was detected. This usually indicates the NOP operation in x86 machine code.
Port 55850 tcp - Possible myserver activity - ref. 010313-1	302	MyServer is a DDOS agent that binds to UDP 55850, and the rootkit installs trojans of ls and ps.
MYPARTY - Possible My Party infection	190	A worm that use address book to mail itself and drops a backdoor Trojan that allows a hacker to control the system
connect to 515 from outside	184	Could be an attempt to exploit This event indicates a format string vulnerability in use_syslog() function in LPRng 3.6.24. This allows remote attackers to execute arbitrary commands.
Null scan!	182	Packets without any flags set in order to avoid IDS detection
Tiny Fragments - Possible Hostile Activity	160	Most likely a scan that is trying to avoid IDS detection.
CS WEBSERVER - external ftp traffic	149	FTP Connection attempt to CS WEBSERVER from external IPs
EXPLOIT x86 setuid 0	136	A source sending a setuid(0) system call to an internal x86 machine.
IRC evil - running XDCC	123	XDCC is a bot client backdoor allowing remote user to have administrative control of a machine.
NMAP TCP ping!	104	Attacker using NMAP a portscanning tool to probe and internal machine
SMB C access	94	An attempt to access the default administrative share C$. If allowed, the attacker can access the C: filesystem.
TCP SRC and DST outside network	68	Source and Destination IPs are both external IPs
Port 55850 udp - Possible myserver activity - ref. 010313-1	48	MyServer is a DDOS agent that binds to UDP 55850, and the rootkit installs trojans of ls and ps.
EXPLOIT x86 setgid 0	46	An event that indicates an exploit attempt by sending a setgid(0) system call to x86 system.
Possible trojan server activity	43	Connection from port 27374 possible Sub-7 or Ramen traffic
TFTP - Internal UDP connection to external tftp server	32	UDP connection from internal source to an external TFTP server.
External FTP to HelpDesk 172.201.70.49	17	Internal connection to and external TFTP server.

External FTP to HelpDesk 172.201.70.50	13	Internal connection to and external TFTP server.
Bugbear@MM virus in SMTP	13	A mass-mailing worm that can also spread through network shares. It has keystroke-logging and backdoor capabilities. The worm also attempts to terminate the processes of various antivirus and firewall programs.
TFTP - External TCP connection to internal tftp server	13	TCP connection from external source to an internal TFTP server.
RFB - Possible WinVNC - 010708-1	10	Possible external attemp to a VNC server.
Attempted Sun RPC high port access	7	Attempted to connect to Sun RPC older versions of Sun would allow viewing of files.
TFTP - External UDP connection to internal tftp server	6	UDP connection from external source to an internal TFTP server.
HelpDesk 172.201.70.50 to External FTP	4	Internal connection to and external TFTP server.
HelpDesk 172.201.70.49 to External FTP	4	Internal connection to and external TFTP server.
EXPLOIT NTPDX buffer overflow	3	Indication that a buffer overflow exploit was attempted against the ntpd network time daemon. Some versions of ntpd and xntpd are vulnerable to remote root access.
HelpDesk 172.201.83.197 to External FTP	2	Internal connection to and external TFTP server.
ICMP SRC and DST outside network	2	Both Destination and Source IP are not internal addresses IP is probably spoofed

Top Talkers:

Top 10 Talking Alerts listed by IP:

# Alerts	Source IP
379760	24.59.33.240
371288	MY.NET.83.146
12388	212.179.83.64
3166	128.8.120.85
3063	152.101.81.195
2805	66.77.73.144
2027	212.179.97.145
2010	129.6.153.67
1811	66.77.73.236
1545	64.52.4.180

# Alerts	Destination IP
379760	MY.NET.83.146
371288	24.59.33.240
21057	MY.NET.100.165
12377	MY.NET.114.88
3740	MY.NET.100.158
3167	MY.NET.99.205
2128	MY.NET.104.204
2011	MY.NET.109.85
568	MY.NET.88.165
560	MY.NET.84.147

Top 10 Talking Scans listed by IP:

# Scans	Source IP
610490	MY.NET.70.176
566210	MY.NET.84.147
254247	MY.NET.165.24
152509	MY.NET.91.240
135804	MY.NET.83.146
125375	MY.NET.198.204
98767	MY.NET.150.113
92032	MY.NET.88.165
76805	MY.NET.111.214
68729	MY.NET.70.207

# Scans	Destination IP
10979	204.183.84.240
7617	24.120.194.178
5620	12.220.145.126
3936	12.245.31.155
3694	68.39.48.75
2863	MY.NET.70.207
2636	151.204.131.129
2486	146.115.121.119
2238	141.149.54.140
2225	200.52.195.1

Top 10 Talking OOS (Out of Spec) listed by IP:

# OOS	Source IP
7186	152.101.81.195
3638	MY.NET.28.2
3557	64.52.4.180
814	209.116.70.75
542	MY.NET.70.183
485	200.221.192.194
206	MY.NET.165.20
68	200.221.192.245
63	148.65.203.115
55	148.63.246.3

# OOS	Destination IP
7186	MY.NET.199.255
3638	MY.NET.198.218
3557	MY.NET.159.93
814	MY.NET.100.217
542	MY.NET.1.4
485	MY.NET.91.81
206	MY.NET.90.114
68	MY.NET.91.81
63	MY.NET.150.133
55	MY.NET.84.245

List of Attacks that will be investigated in more depth due to the large number of alerts:

1. **High port 65535 TCP - possible Red Worm – traffic.**
2. **CS WEBSERVER - external web traffic**
3. **spp_http_decode: IIS Unicode attack detected**
4. **Watchlist 000220 IL-ISDNNET-990517**
5. **SMB Name Wildcard**
6. **SUNRPC highport access!**

1. **High port 65535 TCP - possible Red Worm – traffic.**

Source of attack:
The tables below show the majority of the TCP Red Worm Traffic is between two IP. The 24.59.33.240 IP is registered with an OrgName of ROADRUNNER-NYC a Cable Access Provider.

Source IP	# Alerts	Total	% of Total
24.59.33.240	379763	752040	50.5
MY.NET.83.146	371302	752040	49.4

# Alerts	Source IP	Destination IP
379724	24.59.33.240:65535	MY.NET.83.146:1379
371250	MY.NET.83.146:1379	24.59.33.240: 65535

Spoofed:

Low

1) The initiator of this traffic is looking for a response.
2) I also believe that it is not spoofed since this is TCP traffic, which usually requires a three-way handshake.
3) Because of the volume of traffic it would be difficult to do this with a spoofed IP

Attack Description:

Red Worm is now called Adore and is similar to Ramen and Lions worms. Adore scans the Internet checking Linux hosts to determine whether they are vulnerable to any of the following well-known exploits: LPRng, rpc-statd, wu-ftpd and BIND.

Adore worm replaces the ps binary with it own version, which will not show the worms processes when ran. The ps binary when ran shows what processes are running. It installs the files in /usr/lib/lib. And sends an email to the following addresses: adore9000@21cn.com, adore9000@sina.com, adore9001@21cn.com and adore9001@sina.com.

It attempts to send the following information:

/etc/ftpusers
ifconfig
ps -aux (using the original binary in /usr/bin/adore)
/root/.bash_history
/etc/hosts
/etc/shadow

The script also creates a backdoor by replacing /sbin/klogd with a version that has a backdoor. The backdoor activates when it receives a ping packet with correct size, and opens a shell in the port 65535. The original klogd will be saved to /usr/lib/klogd.o.

Note: klogd gleans from the /proc file system and from system calls to syslogd and display them on the console depending on the messages priority. By default, no messages appear on the console. Messages are sorted into 8 levels, 0-7, and the level number is prepended to each message.

Sam Spade return this info for 21cn.com :

```
  21cn corporation limited dns@21cn.net 85201919
  21cn corporation limited
  11C,109,Tiyu Rd. West
  Guangzhou,Guangdong,China 510620
```

http://www.internic.net returned this info for sina.com.

```
  Domain Name: SINA.COM
```

```
Registrar: NETWORK SOLUTIONS, INC.
Whois Server: whois.networksolutions.com
Referral URL: http://www.networksolutions.com
Name Server: RESOLVER.SINA.COM
Name Server: TOMAHAWK.SINA.COM
Updated Date: 07-aug-2002
```

Was it a false positive:
I believe this to be a false positive as the suspicious port 65535 is on the external host 24.59.33.240. Therefore, these are likely to be false positives, although investigations of the internal host involved in this alert is warranted.

Was this the only event the attacker triggered:
This is the only event triggered by the attacker. This was determined by using grep to parse alert, scan and OOS logs for other events triggered by IP 24.59.33.240. Though looking at the Figure 3. it is seen that communication is taking place in both directions, with one particular IP MY.NET.83.146. The Alert Top Talker listing also shows that IP 24.59.33.240 and MY.NET.83.146 are communicating with each other.

Correlation:
Both **Robert Turner** and **Jeff Zahr** have noted the same alerts in their practicals, although not as not nearly as many as seen in these traces.

Defensive Recommendations:
It is highly recommended that system MY.NET.83.146 be investigated, because of the large amount of alerts indicating Red Worm traffic. A tool that will detect Adore may be downloaded here. It is also recommended that the 4 email addresses mentioned in the attack description, be blocked. Access rules for this server should be reviewed to determine if external sources need to access these system. If no external access is needed the firewall rules should reflect this. MY.NET.83.146 should also have the latest security patches applied a listing of vendors may be found here as well as more information with regards to Red Worm.

Note: Figure 3. Also shows that two-way communication it taking place between one other IP 68.14.128.176 port 69 and MY.NET.83.146. Port 69 is used for TFTP and is not very secure. This should also be investigated to confirm whether this is legitimate traffic or not.

2. **CS WEBSERVER - external web traffic**
   Source of attack:
   All traffic is generated by one External source IP going to 1 IP
External -> 172.201.100.165:80.

SnortSnarf returned the following info:

Signature	# Alerts	Source	Destination
CS WEBSERVER - external web traffic	20890	1176	1

Spoofed:
   Low
1) The initiator of this traffic is looking for a response.
2) I also believe that it is not spoofed since this is TCP traffic, which usually requires a three-way handshake.
3) And most Web alerts have the A+ signature, which would indicate an established session. A+ flag in a signature indicates an ACK with the possibility of any other flags must be matched.

Attack Description:
This alerts generated by this signature appear to be legitimate HTTP traffic.

Is it a false positive:
This traffic appears to be legitimate traffic; using Arin I looked for more info on several of the source IPs and they are all valid companies. As I do not have access to the signature that is triggering this event, I am not positive for what type of traffic it is looking, but I suspect that the intent of the signature is to trigger on all http requests from External sources for this particular Web server.

Was this the only event the attacker triggered?
These were the only event triggered by the top 5 attackers. This was determined by using grep to parse alert, scan and OOS logs for other events triggered by top 5 attackers IPs.

Correlation:
The following people have both noted the same traffic in their practicals Wade Walker and Hee So.

Defensive Recommendations:
The only recommendation for the **CS WEBSERVER** is that the latest security patches be applied in order to make it less vulnerable and that the firewall rules ensure that external traffic only has access to port 80 on the **CS WEBSERVER.**

3. spp_http_decode: IIS Unicode attack detected

Source of attack:
SnortSnarf returned the following info:

Signature	**# Alerts**	**Source**	**Destination**
spp_http_decode: IIS Unicode attack detected	64169	565	1065

Looking at the Top 5 source IPs we see that most of the Alerts are being caused by Internal systems communicating with External IPs

Top Five Sources:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.85.74	24612	24614	5	5
MY.NET.84.133	4571	4571	16	16
MY.NET.152.22	2583	2583	1	1
MY.NET.53.33	1523	1523	15	15
MY.NET.53.93	1394	1394	1	1

Top Five Destination:

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
207.200.86.66	12399	12400	8	8
207.200.86.97	12394	12396	5	6
218.55.184.152	6893	6893	7	7
211.115.212.150	2247	2247	9	9
211.115.212.173	1174	1174	7	7

Note: These Alerts are generated by the HTTP Decode Preprocessor. The Preprocessor function is to convert HTTP URI strings to non-obfuscated ASCII strings. Allowing the analysis of HTTP traffic for suspicious activity.

Spoofed:
Low

1) The initiator of this traffic is most likely looking for a response.
2) I also believe that it is not spoofed since this is TCP traffic, which usually requires a three-way handshake.

Attack Description:
This is an indication that an attacker has attempted to send UNICODE representations of shell metacharacters that could possibly compromise an IIS 4 or IIS 5 WEBSERVER. The UNICODE in the HTTP request are crafted in order to execute code in directories residing outside of the sites webroot directory, this is called Directory Traversal. This can be done with a regular browser with an HTTP request such as this:
(Data from Back Officer Friendly)
HTTP request from 24.42.61.10: GET /msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/system32/cmd.exe?/c+dir

The Common Vulnerability and Exposure group has assigned this vulnerability number CVE-2000-0884. A really good paper on how to evade IDS and take advantage of Unicode Directory Traversal Vulnerability can be found here. As well as the CERT Coordination Center has a Vulnerability Note VU#789543 that talks to this vulnerability

Is it a false positive:
I do believe these alerts to be False Positives, most alerts have likely been triggered by foreign characters when accessing Asian web sites.

Was this the only event the attacker triggered:
Looking at the Top 5 IPs that generated IIS Unicode attacks, which accounts for 52% of the total IIS Unicode attacks, it was found that the only other alerts associated with the Top 5 are:

```
SYN-FIN scans to MY.NET.53.33, MY.NET.53.93 from External IP
152.101.81.195

Watchlist 000222 NET-NCFC communication between IP 159.226.236.23 to
MY.NET.152.22

And

EXPLOIT x86 setuid 0 traffic From External IP 211.39.156.67 to
MY.NET.53.33,
```

Correlation:
The following people have reported seeing similar traffic in their practicals
Michael Holstein, **Paul Crutchfield** and **Steven Drew**

Meaningful Analysis:
Alerts of this type are often generated by Nimda, which attempts to copy itself to vulnerable Microsoft IIS servers that are either unpatched or that have previously been infected with CodeRed II, see CERT Advisory CA-2001-26 Nimda Worm for more information.

Looking at the source of the attacks it was found that the majority of them were from Internal IPs. Using Sam Spade it was found that most alerts were generated when communicating with IP addresses registered in foreign countries. As **Steven Drew** mentioned in his practical, alerts can be attributed to the use of foreign language character set.

Defensive Recommendations:

1. It is recommended that all Internal systems that have generated these alerts be investigated for the Nimda worm. A list of IPs with more than 100 has been provided below. A tool for removing Nimda is available at Symantec's site.

2. It is also recommended that all IIS servers have the latest patches applied. Information for the patches can be found at the following Microsoft's sites ms00-078, MS01-020, MS01-044, as well as installing antivirus software on all critical systems.

3. It is also recommended that traffic from Internal IIS systems destined for External systems on TCP port 80 be blocked at the firewall. Unless there is a business need for Internal IIS server to communicate with External Web servers.

List of Internal IPs with more than 100 alerts:

MY.NET.85.74	MY.NET.104.117	MY.NET.153.189	MY.NET.153.176
MY.NET.84.133	MY.NET.53.36	MY.NET.88.186	MY.NET.153.203
MY.NET.152.22	MY.NET.106.105	MY.NET.88.246	MY.NET.53.220
MY.NET.53.33	MY.NET.183.25	MY.NET.112.204	MY.NET.87.193
MY.NET.53.93	MY.NET.153.177	MY.NET.140.33	MY.NET.152.184
MY.NET.168.181	MY.NET.91.101	MY.NET.91.96	MY.NET.84.216
MY.NET.153.146	MY.NET.91.109	MY.NET.91.95	MY.NET.153.193
MY.NET.104.117	MY.NET.153.197	MY.NET.88.139	MY.NET.145.27
MY.NET.106.106	MY.NET.153.143	MY.NET.91.2	MY.NET.53.172
MY.NET.88.242	MY.NET.153.126	MY.NET.153.124	MY.NET.153.163
MY.NET.104.121	MY.NET.153.148	MY.NET.153.168	MY.NET.53.40
MY.NET.152.215	MY.NET.153.167	MY.NET.153.199	MY.NET.153.165
MY.NET.91.92	MY.NET.153.164	MY.NET.153.174	MY.NET.153.211
MY.NET.153.190	MY.NET.183.15	MY.NET.153.127	MY.NET.53.72
MY.NET.153.196	MY.NET.91.100	MY.NET.107.74	MY.NET.183.59
MY.NET.88.228	MY.NET.91.104	MY.NET.53.160	
MY.NET.53.56	MY.NET.153.46	MY.NET.116.84	

4. Watchlist 000220 IL-ISDNNET-990517

Source of attack:
Source of the attacks were all from IPs that are registered in Israel.

SnortSnarf returned the following info:

Signature	# Alerts	Source	Destination
Watchlist 000220 IL-ISDNNET-990517	20000	98	69

Spoofed:
Low

1) The initiator of this traffic is looking for a response.
2) I also believe that it is not spoofed since this is TCP traffic, which usually requires a three-way handshake.

Attack Description:
Although the signature was designed to alert on traffic from the IPs in the Watchlist, the meaningful analysis will focus on what type of traffic is occurring between MY.NET and the IPs in the Watchlist.

Most of the alerts show that traffic was directed at ports 2939 and 1214, 1214 is a well-known port used for P2P (Peer to Peer) file sharing software such as Kazaa. Systems running Kazaa can be easily exploited. A description of an exploit for Kazaa can be found at Morpheus/Kazaa Exploit.

Was it a false positive:
This is not a false positive, as the intent of the signature is to alert on traffic from the IPs in the Watchlist.

Was this the only event the attacker triggered:
Looking at the Top 5 Talkers of this event see Figure 2 we see that three of them have triggered alerts upon communicating to more that one system.

Correlation:
The following people have reported traffic from the Watchlist 000220 IL-ISDNNET-990517 IPs Hee So, George Bakos and Brian Coyle. Correlating data pertaining to iMesh Russell Meyer

Meaningful Analysis:
Looking at Figure 2. we see that many systems are communicating with 212.179.35.118 and 212.179.66.17, these are both web servers. IP 212.179.66.77 is of special interest as it is a web site that makes available a file sharing software called iMesh Site. Looking at the destination ports it was noted that 1214 and 2939 were used the most. Port 1214 is often used by file sharing software and a search at Treachery Unlimited returned that port 2939 is registered for SM-PAS-2. The large amount of traffic using port 2939 may be explain by the fact that iMesh does not use a specific port and may very well be using port 2939.

Defensive Recommendations:
Alerting on the Watchlist 000220 IL-ISDNNET-990517 has proven useful and the University should continue logging and monitoring these alerts. It is recommended that the University remind the students about the security policy with regards to using file-sharing software and enforce these policies. The University should consider blocking the offending addresses at the firewall if it would not impact the Universities operations too much, the benefits of blocking Port 1214 may outweigh the problems that this will cause I.E. stopping some legitimate traffic. Blocking ports like 1214 would certainly help reduce file-sharing activities but as mentioned before iMesh does not use a specific port. Below is a list of ports that the University should review and consider blocking.

List of known Ports used by file sharing software:

MORPHEUS	1214
NAPSTER	6699, 8888, 8875
EDONKEY	4661, 4662, 4663, 4664, 4665
GNUTELLA	6345, 6346, 6347,6348,6349
AUDIO GALAXY	41000-42000
AIMSTER(AOL)	5190

It was also noticed during the analysis of this alert that port 1095 as shown in Figure 2 was used as a destination port for IP MY.NET.153.147, this port known to be used by RAT (Remote Administration Tool), this should be investigated.

5. SMB Name Wildcard

Source of attack:

SnortSnarf returned the following info:

Signature	# Alerts	Source	Destination
SMB Name Wildcard	19252	530	896

All of the sources IPs for this attack are External IPs.

Spoofed:
Low
IP is not likely spoofed, even though the traffic is UDP and could easily be spoofed, the attacker is expecting information to be returned and therefore not likely spoofed.

Attack Description:
This attack is typically a reconnaissance looking for NetBIOS name table information such as workstation name, domain, and a list of currently logged in users Windows systems or Unix/Linux systems running Samba. CERT's Vulnerability Note VU#32650 describes this vulnerability. This may be a prelude to an attack. This detect is a result of a machine trying to connect to port 137/UDP (NetBIOS Name Service). The "wildcard" indicates a request for all records, and can also be initiated with the command "nbtstat –a [IP address]" from the command line. Another indication that this is not legitimate traffic is that Window to Window traffic uses port 137 for both Source and Destination here is an example showing a port above 1024 being used by source. There is evidence that this is a scripted attack. Looking at the time we see that very little time between alerts and when looking at the destination IP we see that the IP is incremented by one for each alert till it reaches IP xxx.xxx.xxx.47 then next octet in incremented by 1 xxx.xxx.xx1.1 and so on.

Note: I have omitted some alerts to conserve space.

```
10/15-06:01:08.525863 [**] SMB Name Wildcard [**] 211.38.60.177:1026 ->
MY.NET.132.0:137

10/15-06:01:08.579223 [**] SMB Name Wildcard [**] 211.38.60.177:1026 ->
MY.NET.132.1:137

10/15-06:01:08.627508 [**] SMB Name Wildcard [**] 211.38.60.177:1026 ->
MY.NET.132.2:137

10/15-06:01:08.685422 [**] SMB Name Wildcard [**] 211.38.60.177:1026 ->
MY.NET.132.3:137
```

Omitted alerts

……………………………………………………………………………………………
……………………………………………………………………………………………

```
10/15-06:01:11.936239 [**] SMB Name Wildcard [**] 211.38.60.177:1026 ->
MY.NET.132.47:137
```

```
10/15-06:01:27.483482 [**] SMB Name Wildcard [**] 211.38.60.177:1026 ->
MY.NET.133.1:137
```

Here is a packet that has been captured by Snort depicting what the traffic should look like:

```
[**] SMB Name Wildcard [**]
05/10-18:08:05.359797 badguy.com:137 -> goodguy.com:137
UDP TTL:119 TOS:0x0 ID:45361
Len: 58
00 D4 00 00 00 01 00 00 00 00 00 00 20 43 4B 41 ............ CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21 AAAAAAAAAAAAA..!
00 01 ..
```

Trace was obtained at http://www.sans.org/newlook/resources/IDFAQ/port_137.htm.

Was it a false positive:
This is not a false positive as all of the alerts are from External source. There are no alerts for Internal-to-Internal system. Therefore the rule has most likely been configured to only alert on External to Internal SMB Name Wildcard traffic.

Was this the only event the attacker triggered:
Looking at the Top 5 Talkers of this event (SMB Name Wildcard) it was found that they generated two types of scans as well.

Top 5 Talkers for SMB Name Wildcard:
211.38.60.177
210.58.252.159
211.221.142.60
61.36.119.19
64.144.9.1

Example Scans:
Oct 15 06:01:08 211.38.60.177:1026 -> 130.85.132.0:137 UDP
Oct 15 06:01:10 211.38.60.177:2402 -> 130.85.132.20:139 SYN ******S*

Correlation:
The following people have reported SMB Name Wildcard traffic **Brian Sheffler, Tod Beardsley** and **Michael Holstein .**

Meaningful Analysis:
This traffic is a part of normal Microsoft networking and should not be considered suspicious if the source is another internal system, but the Scan logs show that the top 5 External source IPs that produced SMB Name Wildcard alerts, also had scanning activities.

Defensive Recommendations:

It is highly recommended that the following ports be blocked at the perimeter (firewall) for ingress traffic NetBIOS Name Service, 137/tcp and 137/udp, NetBIOS Datagram Service, 138/tcp and 138/udp and NetBIOS Session Service, 139/tcp and 139/udp and 445/tcp which is used by Windows 2000 systems by default. As mentioned in the CERT Vulnerability Note VN-2000-03 this would prevent external sources from sending NetBIOS service traffic to internal machines. This does not prevent internal users from attacking other internal machines and therefore it is also recommended that users be trained on how to share resource in a secure manner. CERT has provided a best practices Configuration Guide.

6. SUNRPC highport access!
Source of attack:

SnortSnarf returned the following info:

Signature	# Alerts	Source	Destination
SUNRPC highport access!	5487	41	41

All of the sources IPs for this attack are External IPs and all of the Destination IPs are Internal.

Spoofed:

The sources IPs are not likely spoofed. A packet that is part of established TCP session normally causes this event. The attacker is also looking for a response; this is difficult with a spoofed IP. Another reason to believe that the IPs are not spoofed is if the university is using stateful inspection firewalls and the OSs being attack are not vulnerable to sequence number prediction.

Attack Description:
The event is triggered by an attempt to connect to Unix systems, which may be running Portmap at a high port. In this case it is port 32771. This is of concern because Portmap keeps track of where RPC services are located (port). It is relatively easy to obtain this information once connected to a port by running rpcinfo –p. This will dump a list of RPC services and what ports offer them. Here is an example from Stephen Northcutt's paper The trouble with RPCs**.**

```
rpcinfo -p | grep 32772
   100024    1   udp  32772  status
   100002    2   tcp  32772  rusersd
   100002    3   tcp  32772  rusersd
```

Once the attacker has this information known exploit may be tried on these services. The **rusersd** daemon is a server that responds to queries and returns a list of users currently on the network. More information about this event can be found at arachNIDS. A CVE (Common Vulnerability and Exposure) is still under review CAN-1999-0632. Most of the events (96%) generated are using source Port 22 (SSH). The other source ports included ports 80,21,443.

Was it a false positive:
This does not appear to be a false positive.

Was this the only event the attacker triggered:
Using the Top 5 Talkers IP of this event. Alert, Scan and OOS log were parsed for other events, none were found.

Correlation:
Crist Clark has also reported SUNRPC highport access! Traffic in his practical.

Meaningful Analysis:
As mentioned above most of the connections to port 32771 have a source port of 22, which indicates that the initial connection was made from an internal machine. I did not find any other information in the logs to confirm this though. The other possibility is that an attacker is using a tool that can specify local source port in order to get past the firewall.

Defensive Recommendations:
It is recommended that the following systems be investigated to determine if they initiated the SSH connections. MY.NET.99.205, MY.NET.109.85 and MY.NET.149.14. This needs to be done in order to determine whether this is malicious traffic. If it is found that the traffic is indeed malicious, blocking traffic at the firewall that is destined for port 32771 is highly recommended.

Note: Blocking inbound traffic may prevent legitimate traffic from passing through the firewall, but the added security will outweigh this issue.

5 External sources:

The criteria I used for selecting these 5 external IP addresses for further investigation was I first looked at IPs that triggered on both Alerts and OOS. The remaining IPs selected were Top Talkers in the Top Talker Alert category.

External IP	# Alerts
152.101.81.195	3063
24.59.33.240	379760
212.179.83.64	12388
128.8.120.85	3166
66.77.73.144	2805

Sam Spade was used to gather information for each of these IP addresses, Sam Spade returned IP address 212.179.83.64 as being registered with www.ripe.net . I then ran a query at www.ripe.net in order to get more detailed information for this IP.

```
IP Address: 152.101.81.195
OrgName:    Hong Kong Internet & Gateway Services Ltd.
OrgID:      HKIGSL
NetRange:   152.101.0.0 - 152.101.255.255
CIDR:       152.101.0.0/16
NetName:    HKNET
NetHandle:  NET-152-101-0-0-1
Parent:     NET-152-0-0-0-0
NetType:    Direct Assignment
NameServer: HK.NET
NameServer: HKIGS.HK.NET
Comment:
RegDate:    1993-09-23
Updated:    2001-07-10
TechHandle: ZP69-ARIN
TechName:   CPCNet Hong Kong Ltd. NOC
TechPhone:  +852-2331-8123
TechEmail:  hostinfo@cpcnet-hk.com

IP Address: 24.59.33.240
OrgName:    ROADRUNNER-NYC
OrgID:      RRNY
```

```
NetRange:   24.58.0.0 - 24.59.255.255
CIDR:       24.58.0.0/15
NetName:    RR-NYS-3BLK
NetHandle:  NET-24-58-0-0-1
Parent:     NET-24-0-0-0-0
NetType:    Direct Allocation
NameServer: DNS1.RR.COM
NameServer: DNS2.RR.COM
NameServer: DNS3.RR.COM
NameServer: DNS4.RR.COM
Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate:    2001-11-02
Updated:    2002-08-30
TechHandle: ZS30-ARIN
TechName:   ServiceCo LLC
TechPhone:  +1-703-345-3416
TechEmail:  abuse@rr.com
OrgAbuseHandle: ABUSE10-ARIN
OrgAbuseName:   Abuse
OrgAbusePhone:  +1-703-345-3416
OrgAbuseEmail:  abuse@rr.com
OrgTechHandle: IPTEC-ARIN
OrgTechName:   IP Tech
OrgTechPhone:  +1-703-345-3416
OrgTechEmail:  abuse@rr.com

IP Address: 212.179.83.64
inetnum:      212.179.80.0 - 212.179.94.255
netname:      CABLES-CONNECTION
mnt-by:       INET-MGR
descr:        CABLES-CUSTOMERS-CONNECTION
country:      IL
admin-c:      MR916-RIPE
tech-c:       ZV140-RIPE
status:       ASSIGNED PA
remarks:      please send ABUSE complains to abuse@bezeqint.net
remarks:      INFRA-AW
notify:       hostmaster@bezeqint.net
changed:      hostmaster@bezeqint.net 20021029
source:       RIPE
route:        212.179.64.0/18
descr:        ISDN Net Ltd.
origin:       AS8551
notify:       hostmaster@bezeqint.net
mnt-by:       AS8551-MNT
changed:      hostmaster@bezeqint.net 20020618
```

```
source:      RIPE
person:      Miri Roaky
address:     bezeq-international
address:     40 hashacham
address:     petach tikva 49170 Israel
phone:       +972 1 800800110
fax-no:      +972 3 9203033
e-mail:      hostmaster@bezeqint.net
nic-hdl:     MR916-RIPE
changed:     hostmaster@bezeqint.net 20021027
source:      RIPE
person:      Zehavit Vigder
address:     bezeq-international
address:     40 hashacham
address:     petach tikva 49170 Israel
phone:       +972 1 800800110
fax-no:      +972 3 9203033
e-mail:      hostmaster@bezeqint.net
nic-hdl:     ZV140-RIPE
changed:     hostmaster@bezeqint.net 20021027
source:      RIPE

IP Address: 128.8.120.85
OrgName:    University of Maryland
OrgID:      UNIVER-262

NetRange:   128.8.0.0 - 128.8.255.255
CIDR:       128.8.0.0/16
NetName:    UMDNET
NetHandle:  NET-128-8-0-0-1
Parent:     NET-128-0-0-0-0
NetType:    Direct Assignment
NameServer: NOC.UMD.EDU
NameServer: NS1.UMD.EDU
NameServer: NS2.UMD.EDU
NameServer: MX.NSI.NASA.GOV
Comment:
RegDate:
Updated:    1998-10-06
TechHandle: UM-ORG-ARIN
TechName:   University of Maryland DNS Administration
TechPhone:  +1-301-405-3003
TechEmail:  dnsadmin@noc.umd.edu

IP Address: 66.77.73.144
Qwest Cybercenters QWEST-CYBERCENTER-2 (NET-66-77-0-0-1)
```

66.77.0.0 - 66.77.207.255
Fast Search, Inc. QWEST-MCC-FASTSRCH3 (NET-66-77-73-0-1)
66.77.73.0 - 66.77.73.255

Link graph and analysis of data relationship:

Below is a graph plotting the total Alerts, Scans and OOS over a 5 day period. Looking at the graph we can see that with an increase in Scan activity that Alerts increase as well. It is also interesting that there is a significant increase in both Scans and Alerts over the weekend. This leads me to believe that attackers have more time during this period for malicious activities and that the attackers may believe that the University may not have enough staff on Saturdays and Sundays to respond to attacks. I would advise that the administrators compare weekday traffic vs. weekend traffic over a longer period of time to confirm this theory.

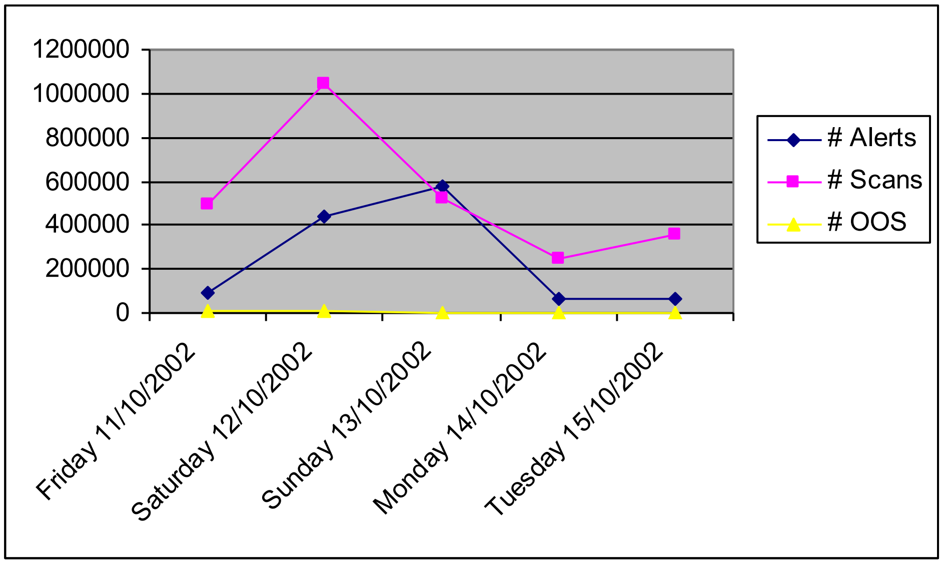
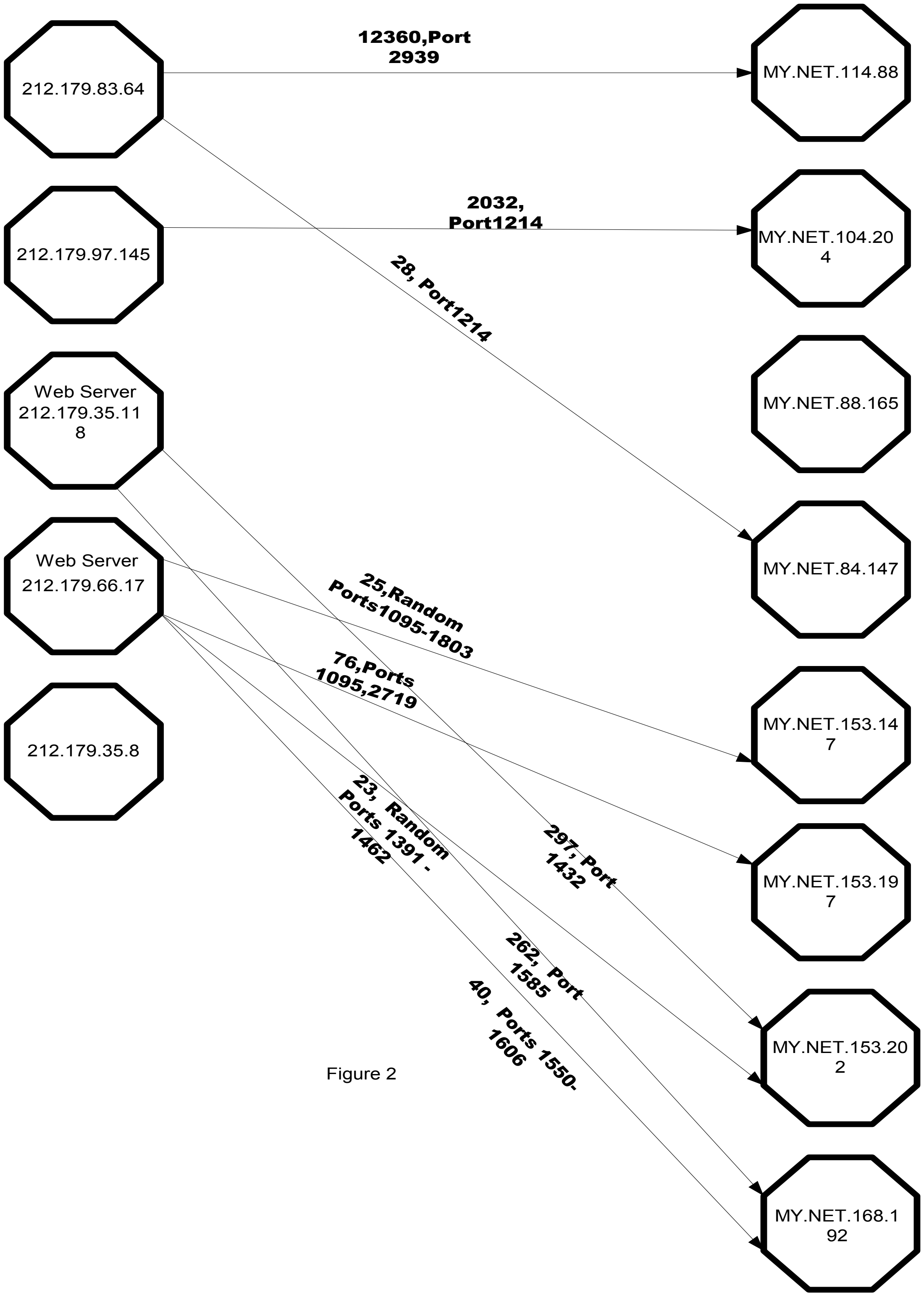


Figure 1

Below is a graph of traffic from the **Watchlist 000220 IL-ISDNNET-990517** alerts. What is shown in the graph on the left are the top talkers from external sources (Systems registered in Israel) and on the right are internal system. The arrows show direction of traffic and each line has #alerts, Destination Ports.



212.179.83.64
212.179.97.145
Web Server 212.179.35.118
Web Server 212.179.66.17
212.179.35.8
MY.NET.114.88
MY.NET.104.204
MY.NET.88.165
MY.NET.84.147
MY.NET.153.147
MY.NET.153.197
MY.NET.153.202
MY.NET.168.192
12360,Port 2939
2032, Port1214
28, Port1214
25,Random Ports1095-1803
76,Ports 1095,2719
23, Random Ports 1391 - 1462
297, Port 1432
262, Port 1585
40, Ports 1550-1606

Figure 2

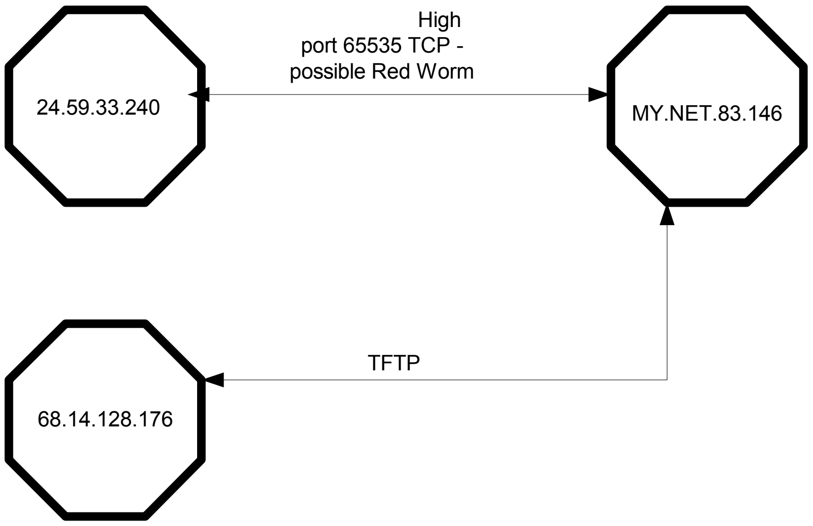


Figure 3

TFTP Traffic that generated alerts:

```
10/12-15:46:05.326555 [**] TFTP - Internal UDP connection to external tftp server [**]
68.14.128.176:69 -> 172.201.83.146:6257

10/13-03:29:36.426505 [**] TFTP - Internal UDP connection to external tftp server [**]
172.201.83.146:6257 -> 68.14.128.176:69
```

Analysis Process:

1. Ran grep on each file looking for 172.201.? IPs to make sure that none existed. # `grep 172.201 <name of file>`
2. I then opened the files and replaced the sanitized IP's that had MY.NET in the first two octets with 172.201. Here is the vi command used # **:1,$s/MY.NET/172.201**. It was necessary to convert these sanitized IPs so that snortsnarf.pl would parse the IPs properly. One of the alert files was a little over 70 megs and after 36 hours the conversion still was not complete. I used sed in order to do the search and replace. The tool I used to do this is ssed, which can be found at URL: `http://www.student.northpark.edu/pemente/sed/#ssed`.
3. Using two Perl scripts that were created by Tod A. Beardsley and included in his practical, I was able to summarize the scan log into meaningful data. Scripts can be found in Appendix A
4. Graphs and tables were created with Excel 2000
5. The biggest challenge with the data was the getting snortsnarf.pl to parse all to the data without running out of memory. The way I managed to deal with the problem was to remove all of the data pertaining to the biggest hitter from the logs.

Steps for trimming logs:

1) Using OC File Merger I merged all of the Alert logs into one file.
2) Using grep I then parsed the merged file for all of the lines pertaining to possible Red Worm and redirected it to a new file.

Example: #**grep merge "Red Worm" > RedWormOnly.txt**

- Using sed I then removed all lines with Red Worm in it with the following command # **sed "/Red/d" merge > reduced**

Listing of Tools and Utilities used:

1. grep
2. sed
3. vi
4. wc
5. Word 2000
6. Excel 2000
7. Active Perl

References:

Northcutt Stephen, Cooper Mark, Fearnow Matt, Frederick Karen,. Intrusion Signatures and Analysis. Indianapolis: New Riders Publishing, 2001.

Northcutt Stephen. Network Intrusion Detection An Analyst's Handbook. Indianapolis: New Riders Publishing, 2000.

Duke University Instructions on Cleaning IRC bot & backdoor: XDCC 24 May 2002 URL:
http://security.duke.edu/cleaning/xdcc.html (04 Nov 2002).

Network Coordination Centre URL:
http://www.ripe.net/. (04 Nov 2002)

Global Incident Analysis Center 12 April 2001 URL:
http://www.sans.org/y2k/adore.htm (05 Nov 2002).

LeChat "Morpheus Exploit" 15 Dec 2001 URL:
http://users.pandora.be/lechat/Morpheus%20Exploit.htm (30 Oct 2002).

CIAC "Linux worm Adore" 5 April 2001 URL:
http://www.ciac.org/ciac/bulletins/l-067.shtml (2 Nov 2002).

Meyer, Russell. Practical Assignment 30 July 2002 URL:
http://www.giac.org/practical/Russell_Meyer_GSEC.doc (1 Nov 2002)

Treachery Unlimited 21 June 2002 URL:
http://www.treachery.net/tools/ports/lookup.cgi

Pittard, Phil. BLOCKING FILE SHARING APPS 6 April 2002 URL:
http://www.internal.schools.net.au/listserver/sina-sa/msg05018.html (2 Nov 2002)

CERT® Vulnerability Note VN-2000-03 "Denial of Service Attack in NetBIOS Services" 10 Aug 2000 URL:
http://www.cert.org/vul_notes/VN-2000-03.html.

CERT® Coordination Center "Windows NT Configuration Guidelines" 28 March 2002 URL: http://www.cert.org/tech_tips/win_configuration_guidelines.html#IV (4 Nov 2002)

Northcutt, Stephen. "The trouble with RPCs" 6 Jan 2000 URL:
http://www.sans.org/y2k/trouble_RPCs.htm (4 Nov 2002)

CVE "CAN-1999-0632 (under review)" URL:
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0632

Hacker, Eric. IDS Evasion with Unicode 3 Jan 2001 URL:
http://online.securityfocus.com/infocus/1232 (17 Nov 2002)

CERT® Vulnerability Note, VU#789543, 18 Sept. 2001 URL:
http://www.kb.cert.org/vuls/id/789543 (18 Nov 2002)

CERT® Advisory CA-2001-26 Nimda Worm 25 Sept 2001 URL:
http://www.cert.org/advisories/CA-2001-26.html (18 Nov. 2002)

Ferrie, Peter W32.Nimda.A@mm Removal Tool25 July 2002 URL:
http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.a@mm.removal.tool.html (18 Nov. 2002)

Appendix A:

Perl Script for converting log files to comma delimited file.

```
#!/cygdrive/c/Perl/bin/perl.exe -w
# Name: csv.pl
# Reads in a Snort -A Fast style alert log which for some
#
# Usage: csv.pl infile [outfile]

unless ($ARGV[0]) {
  print "Need an input file!\n";
  die "(Hint: go to http://www.research.umbc.edu/~andy and get one)\n";
}

unless ($ARGV[1]) {
  $outfile = "$ARGV[0].csv";
} else {
  $outfile = "$ARGV[1]";
}

open(INFILE,"$ARGV[0]") || die "Can't open $ARGV[0] for reading!\n";
open(OUTFILE,">$outfile") || die "Can't open $ARGV[1] for writing!\n";

print "Transforming $ARGV[0] into $outfile.\n";
print "Just a moment.";

@calendar=qw(Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec);



while (<INFILE>) {
  next unless /(\w{1,3}\.){2}(\d{1,3}\.\d{1,3})/;      # Skip lines missing IPv4 IPs.
  next if /spp_portscan/;                               # Skip portscan notifications.
  chomp;
  if (/ \[\*\*\] /) {                                    # Alert report.

  ($date_and_time,$alert,$src_and_dst) = split(/\s+\[\*\*\]\s/);
  ($date,$time) = split(/-/,$date_and_time);
  ($month_number,$day) = split(/\//,$date);
  $month = $calendar[$month_number-1];
  ($src,$dst) = split(/\s-\>\s/,$src_and_dst);
  ($src_ip,$src_port) = split(/:/,$src);
  ($dst_ip,$dst_port) = split(/:/,$dst);
  $snort_entry="ALERT" ;
  } else {                                                   # Scan report.
  ($month,$day,$time,$src,$arrow,$dst,$alert,$flags) = split;
  undef $arrow;
  ($src_ip,$src_port) = split(/:/,$src);
  $alert = "$alert scan (Internally-based)" if $src_ip =~ /^MY\.NET/;
  $alert = "$alert scan (Externally-based)" unless $src_ip =~ /^MY\.NET/;
  ($dst_ip,$dst_port) = split(/:/,$dst);
  $snort_entry="SCAN" ;
}

  print OUTFILE "$snort_entry,";
  print OUTFILE "$month,$day,$time,$alert,";
  print OUTFILE "$src_ip,";
  print OUTFILE "$src_port" if $src_port;
  print OUTFILE "None" unless $src_port;
  print OUTFILE ",";
  print OUTFILE "$dst_ip";
  print OUTFILE ",";
  print OUTFILE "$dst_port" if $dst_port;
  print OUTFILE "," if $flags;
  print OUTFILE "None," unless $dst_port;
  print OUTFILE "$flags" if $flags;
  print OUTFILE "\n";

    $happydots++;
    print "." if $happydots % 100 == 0; # if $happydots == 100;
    print "Just a moment." if $happydots % 46600 == 0;
}
```

Script that summarize alert files that have been converted to a CSV file.

```
#!/cygdrive/c/Perl/bin/perl.exe

# Name: summarize.pl

# Take a source file (generated by csv.pl) and summarize the contents,
# grouping alerts in a variety of ways we care about. This code absolutely
# could be and should be optomized by a real perl hacker.

# Usage: summarize.pl infile [outfile]

unless ($ARGV[0]) {
  print "Need an input file!\n";
  print "(Hint: go to http://www.research.umbc.edu/~andy and get one)\n";
  die "(Hint2: Don't forget to turn it into CSV and drop the portscans.)\n";
```

```
}

unless ($ARGV[1]) {                    # Check for a specified output file.
  if ($ARGV[0] =~ /\.csv$/ ) {         # If it's *.csv, autogenerate the output
    $outfile = $`."-summary.txt";      # filename. (Could be seen as unfriendly.)
  }
  } else {
  $outfile = "$ARGV[1]";
}

open(INFILE,"$ARGV[0]")  || die "Can't open $ARGV[0] for reading!\n";
open(OUTFILE,">$outfile") || die "Can't open $outfile for writing!\n";

print "Counting up all the Events of Interest in $ARGV[0].\nJust a moment.";

while (<INFILE>) {
chomp;
if ( (split(/\,/,$_))[0] eq "ALERT") {

 ($snort_type,$month,$day,$time,$alert,
  $src_ip,$src_port,$dst_ip,$dst_port) = (split(/\,/,$_));
 $date = "$month/$day";
} else {
 ($snort_type,$month,$day,$time,$alert,
  $src_ip,$src_port,$dst_ip,$dst_port,$flags) = (split(/\,/,$_));
 $date = "$month/$day";
}

# Frequency analysis on all that junk up there.


$date_counter{"$date"}++;
$alert_counter{"$alert"}++;


  if ($src_ip =~ "^MY\.NET") {
          $internal_src_ip_counter{"$src_ip"}++;
          $internal_src_port_counter{"$src_port"}++;

          if ($dst_ip =~ "^MY\.NET") {
                  $internal_internal_relationship_counter{"$src_ip"."->"."$dst_ip"}++;
          } else {
                  $internal_external_relationship_counter{"$src_ip"."->"."$dst_ip"}++;
          }
  } else {
          $external_src_ip_counter{"$src_ip"}++;
```

```
        $external_src_port_counter{"$src_port"}++;
        if ($dst_ip =~ "^MY\.NET") {
                $external_internal_relationship_counter{"$src_ip"."->"."$dst_ip"}++;
        } else {
                $external_external_relationship_counter{"$src_ip"."->"."$dst_ip"}++;
                # Hopefully, this case never happens.
        }
  }

if ($dst_ip =~ "^MY\.NET") {
        $internal_dst_ip_counter{"$dst_ip"}++;
        $internal_dst_port_counter{"$dst_port"}++;
} else {
        $external_dst_ip_counter{"$dst_ip"}++;
        $external_dst_port_counter{"$dst_port"}++;
}

# Assure the user that something's happening, and we're not hung.

   $happydots++;
   print "." if $happydots % 100 == 0; # if $happydots == 100;
   print "Just a moment." if $happydots % 46600 == 0;
}

foreach $key ( keys(%date_counter) ) {
        push (@dates, "$date_counter{$key},$key");
}
foreach $key ( keys(%alert_counter) ) {
        push (@alerts, "$alert_counter{$key},$key");
}
foreach $key ( keys(%internal_src_ip_counter) ) {
        push (@internal_src_ips, "$internal_src_ip_counter{$key},$key");
}
foreach $key ( keys(%internal_src_port_counter) ) {
        push (@internal_src_ports, "$internal_src_port_counter{$key},$key");
}
foreach $key ( keys(%internal_dst_port_counter) ) {
        push (@internal_dst_ports, "$internal_dst_port_counter{$key},$key");
}
foreach $key ( keys(%internal_dst_ip_counter) ) {
        push (@internal_dst_ips, "$internal_dst_ip_counter{$key},$key");
}
foreach $key ( keys(%external_src_ip_counter) ) {
        push (@external_src_ips, "$external_src_ip_counter{$key},$key");
}
foreach $key ( keys(%external_src_port_counter) ) {
	push (@external_src_ports, "$external_src_port_counter{$key},$key");
}
foreach $key ( keys(%external_dst_ip_counter) ) {
	push (@external_dst_ips, "$external_dst_ip_counter{$key},$key");
}
foreach $key ( keys(%external_dst_port_counter) ) {
	push (@external_dst_ports, "$external_dst_port_counter{$key},$key");
}
foreach $key ( keys(%internal_internal_relationship_counter) ) {
	push (@internal_internal_relationships,
"$internal_internal_relationship_counter{$key},$key");
}
foreach $key ( keys(%internal_external_relationship_counter) ) {
	push (@internal_external_relationships,
"$internal_external_relationship_counter{$key},$key");
}
foreach $key ( keys(%external_internal_relationship_counter) ) {
	push (@external_internal_relationships,
"$external_internal_relationship_counter{$key},$key");
}
foreach $key ( keys(%external_external_relationship_counter) ) {
	push (@external_external_relationships,
"$external_external_relationship_counter{$key},$key");
}

# Group everything up in a sensible order:

@things_we_care_about = (
	[@dates],
	[@alerts],
	[@external_src_ips],
	[@external_src_ports],
	[@external_internal_relationships],
	[@external_external_relationships],
	[@internal_src_ips],
	[@internal_src_ports],
	[@internal_internal_relationships],
	[@internal_external_relationships],
	[@internal_dst_ips],
	[@internal_dst_ports],
	[@external_dst_ips],
	[@external_dst_ports],
	);

# Write it all down.
```

```
print "\nWriting the report to $outfile.";
undef $happydots;

foreach $report_item (@things_we_care_about) {

# print OUTFILE "\n\@$report_item\n";      # Uncomment this for light debugging

if ($report_item eq @things_we_care_about[0]) {
        $title = "EOIs by Date";
  } elsif ($report_item eq @things_we_care_about[1]) {
        $title = "EOIs by Alert Message";
  } elsif ($report_item eq @things_we_care_about[2]) {
        $title = "EOIs by Source IP (External Only)";
  } elsif ($report_item eq @things_we_care_about[3]) {
        $title = "EOIs by Source Port (External Only)";
  } elsif ($report_item eq @things_we_care_about[4]) {
        $title = "EOIs by Relationship (External->Internal Only)";
  } elsif ($report_item eq @things_we_care_about[5]) {
        $title = "EOIs by Relationship (External->External Only)";
  } elsif ($report_item eq @things_we_care_about[6]) {
        $title = "EOIs by Source IP (Internal Only)";
  } elsif ($report_item eq @things_we_care_about[7]) {
        $title = "EOIs by Source Port (Internal Only)";
  } elsif ($report_item eq @things_we_care_about[8]) {
        $title = "EOIs by Relationship (Internal->Internal Only)";
  } elsif ($report_item eq @things_we_care_about[9]) {
        $title = "EOIs by Relationship (Internal->External Only)";
  } elsif ($report_item eq @things_we_care_about[10]) {
        $title = "EOIs by Destination IP (Internal Only)";
  } elsif ($report_item eq @things_we_care_about[11]) {
        $title = "EOIs by Destination Port (Internal Only)";
  } elsif ($report_item eq @things_we_care_about[12]) {
        $title = "EOIs by Destination IP (External Only)";
  } elsif ($report_item eq @things_we_care_about[13]) {
        $title = "EOIs by Destination Port (External Only)";

}

print OUTFILE "    ";
for ($i = -1; $i <= length($title); $i++) {print OUTFILE "_" ; }

print OUTFILE "\n";
print OUTFILE " __/ $title \\";
for ($i = 0; $i+8+length($title) <= 70; $i++) { print OUTFILE "_" ; }
print OUTFILE "\n";
printf OUTFILE "| %-68s|\n";
undef $eoi_unique_count;
undef $eoi_total_count;
unless (@$report_item) {
      printf OUTFILE "| %-68s|\n","No events of interest for this category (usually a Good
Thing)" ;
}

foreach $item ( reverse(sort{ $a <=> $b }(@$report_item))) {
          ($count,$entry) = split(/\,/,$item);

          # Assure the user we're doing stuff (ie, not hung or anything)...
             $happydots++;
             print "." and $happydots = 0 if $happydots == 100;

          $eoi_unique_count++;
          $eoi_total_count = $eoi_total_count + $count;

          if (length($entry) <= 58) {
                    printf OUTFILE "| %-8d %-58s |\n",$count,$entry;
          } elsif (length($entry) > 65 ) {
                    printf OUTFILE "| %-8d %-55s... |\n",$count,substr($entry,0,55);
          }
}

printf OUTFILE "| %-68s|\n";
printf OUTFILE "| %-20s%8d%31s%8d |\n ",
          "Total Uniques: ",
          $eoi_unique_count,
          "Total EOIs: ",
          $eoi_total_count;

for ($i = 0; $i <= 68; $i++) { print OUTFILE "-" ; }
print OUTFILE "\n";

}

print "\nDone!\n";
```