GCIH Practical Assignment – Version 2.1a
Option 2 – Support for the Cyber Defense Initiative

# Malicious Links

Tom Simcock

July 20th, 2003

## Summary

This paper was written in support of the Cyber Defense Initiative and discusses a serious vulnerability that affects all users running default installations of Windows XP. A vulnerability exists in the Windows XP Help and Support Center that allows files and directories on computers running Windows XP to be deleted by clicking on specially crafted hyperlinks. This is referred as the Windows XP HCP URI Handler Abuse Vulnerability.

The paper details the relationship between the WWW service, the port associated with this service and an exploit that can be used through this service to delete files on computers running a default installation of Windows XP. A section on ways to deal with this vulnerability is also included.

# Table of Contents

## List of Tables

## List of Figures

# 1. Targeted Port

## 1.1 Targeted Service

Port 80 is the well-known port for the World Wide Web (WWW) service used on the Internet[1] and consistently appears on Internet Storm Center's top ten attacked ports list (Table 1).

| Service Name | Port Number | Explanation |
|---|---|---|
| netbios-ssn | 139 | Windows File Sharing Probe |
| **www** | **80** | **HTTP Web server** |
| ms-sql-m | 1434 | Microsoft-SQL-Monitor |
| microsoft-ds | 445 | Win2k+ Server Message Block |
| ident | 113 | --- |
| netbios-ssn | 139 | NETBIOS Session Service |
| eDonkey2000 | 4662 | eDonkey2000 Server Default Port |
| --- | 0 | --- |
| itactionserver1 | 7280 | ITACTIONSERVER 1 |
| domain | 53 | Domain Name Server |

**Table 1**. Top Ten Attacked Ports for 17 / 6 / 2003 [2]

The WWW service is run via Web servers that generally utilise port 80. Web servers often contain vulnerabilities, which make them prime targets for attackers to use to gain unauthorised access to systems. *"Almost 70 percent of malicious activity occurs as a result of entry through port 80"* (Costello, Sam. "ISS: Worms overtake DoS as top attacks in 2002". April 03, 2002. URL: http://www.idg.net/idgns/2002/04/03/ISSWormsOvertakeDoSAsTop.shtml). The high numbers of port scans recorded by the Internet Storm Center (Figure 1) appear to support this claim.

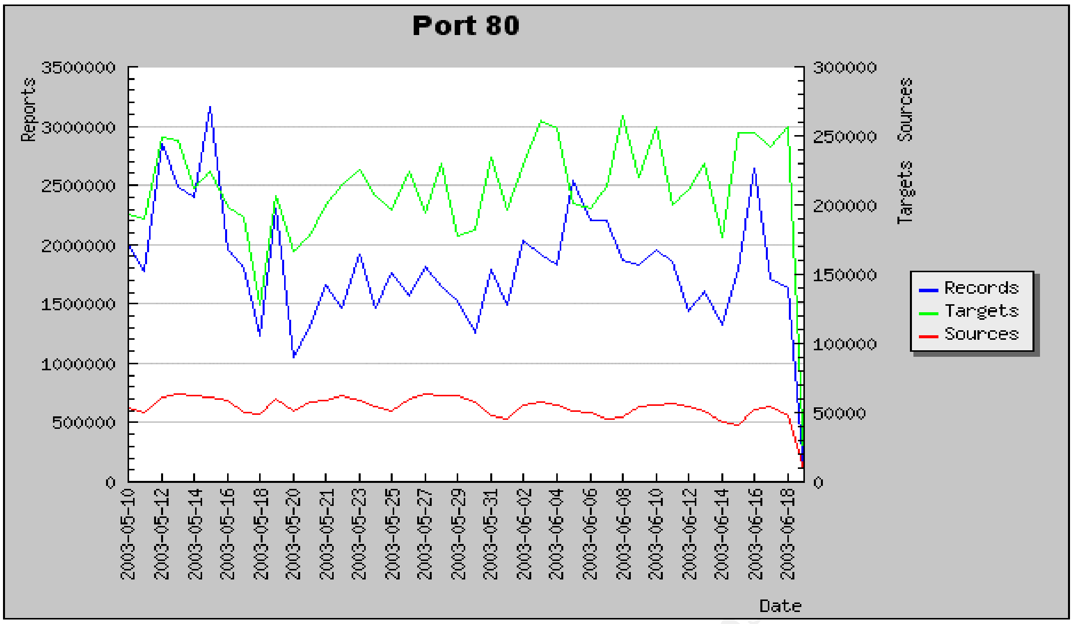


**Figure 1.** Records, Targets and Sources for Port 80 between 10/5/2003 – 18/6/2003[3]

According to Neohapsis (Table 2) a large number of Trojan horse programs also operate through port 80.

| Protocol | Service | Name |
|---|---|---|
| Tcp | 711Trojan | [Trojan] 711 Trojan (Seven Eleven) |
| Tcp | AckCmd | [Trojan] AckCmd |
| Tcp | AckCmd | [Trojan] AckCmd |
| Tcp | BackEnd | [Trojan] Back End |
| Tcp | BO2000Plug-Ins | [Trojan] Back Orifice 2000 Plug-Ins |
| Tcp | Cafeini | [Trojan] Cafeini |
| Tcp | CGIBackdoor | [Trojan] CGI Backdoor |
| Tcp | Executor | [Trojan] Executor |
| Tcp | GodMessage4Creator | [Trojan] God Message 4 Creator |
| Tcp | GodMessage | [Trojan] God Message |
| Tcp | Hooker | [Trojan] Hooker |
| Tcp | IISworm | [Trojan] IISworm |
| Tcp | MTX | [Trojan] MTX |
| Tcp | NCX | [Trojan] NCX |
| Tcp | Noob | [Trojan] Noob |
| Tcp | Ramen | [Trojan] Ramen |
| Tcp | ReverseWWWTunnel | [Trojan] Reverse WWW Tunnel Backdoor |
| Tcp | RingZero | [Trojan] RingZero |
| Tcp | RTB666 | [Trojan] RTB 666 |
| Tcp | Seeker | [Trojan] Seeker |
| Tcp | WANRemote | [Trojan] WAN Remote |
| Tcp | WebDownloader | [Trojan] WebDownloader |
| Tcp | WebServerCT | [Trojan] Web server CT |

**Table 2.** Known Services running on port 80 [4]

The number of Trojan horse programs written to operate through port 80 may correlate to the fact that Web servers must leave a listening port open (usually port 80), to allow incoming requests to reach the Web servers' HTTP daemon. If an attacker can successfully install a Trojan on a Web server, they can use this Trojan for continuous access to the Trojaned system.

## 1.2 Description of Service

The World Wide Web (WWW) is one of many services for sharing information over the Internet. The WWW serves a variety of functions including research, development, marketing, advertising, sales, support, and entertainment. It allows users to access resources residing on Web servers that support the HTTP protocol. These resources may range from HTML documents (Hypertext Markup Language), images, and music files to the output of cgi scripts or dynamically generated results from queries.

Web browsers are a common application, used to access and display information from Web servers. A web browser (HTTP client) is used to communicate with a Web server (HTTP server) via one or more TCP connections. The HTTP client establishes a TCP connection to the Web server and issues a request, and then listens for the server's response. The file returned by the server may contain hyperlinks that point to other files that may reside on other servers (Figure 2).

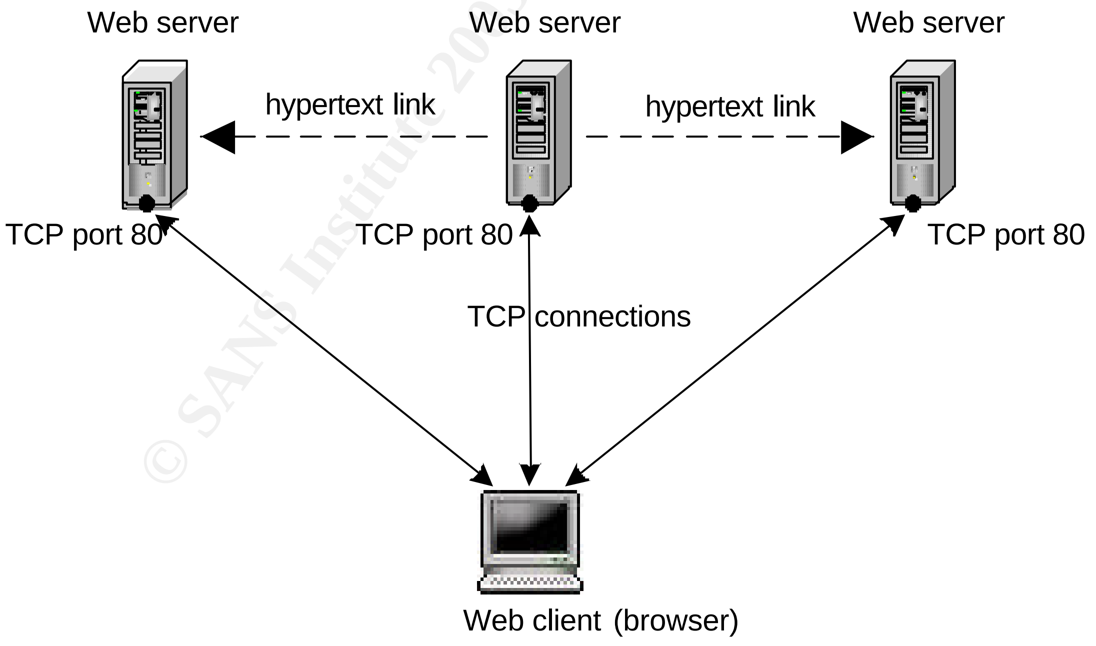


**Figure 2.** Simplified Model of WWW Architecture [5]

## 1.3 Protocol used by WWW

HTTP or Hypertext Transfer Protocol is a stateless, application-level network protocol and is the basis for the World Wide Web. The HTTP protocol is an Internet standard that specifies how an application can locate and acquire resources stored on another computer.[6] The HTTP protocol is based on the client - server model of communication and allows a computer (the client) to establish a TCP connection to a Web server, and make requests of the server. If the server accepts the request, it sends back the specified resource to the client.

Users usually make requests to Web servers by entering a URL in the address bar on a web browser or via a hyperlink on a web page, e.g. http://www.google.com. The HTTP protocol uses URLs (Uniform Resource Locators), a type of URI (Universal Resource Identifier), to point to resources on Web servers by location. A resource requested via a URL can comprise of these elements:

- protocol type (http, ftp, hcp, etc)
- machine name
- directory path
- filename

} http://host/path/file.html

The browser uses its URI handler to take http requests as parameters and attempts to establish a connection with the Web server specified in the URL. If the connection is successful the browser uses a HTTP method to interact with the Web server, e.g. GET /index.html HTTP/1.1, is a HTTP request for the index.html resource located on the Web server that the client has connected to.

The HTTP protocol is stateless which means after the Web server has responded to the client's request, the connection between client and server is severed and forgotten. Each time a user sends or receives data, a new connection to the Web server has to be established. Each new request is processed by the Web server without any knowledge of the previous resource requested. This structure makes it difficult to create interactive Web environments where maintaining a users' session data would be useful, e.g. online shopping and online banking websites. Various techniques have been implemented to address the shortcomings of the HTTP protocol which allow Web servers to track a users' session, e.g. cookies and session IDs.

HTTP has been in use since 1990 (version HTTP/0.9). The current version is HTTP/1.1 as defined in RFC2616. A related protocol used for HTTP transfer of sensitive information is called Secure Socket Layers (SSL). SSL creates an encrypted channel through which a client and server can communicate. SSL commonly uses port 443.

## 1.4 Vulnerabilities

The vulnerabilities commonly associated with the WWW service are those found in Web server software, and potentially vulnerable resources hosted by Web servers. The common vulnerabilities can be broken down into the following categories:

- *Cross-Site Scripting:*
  Embedding commands inside URLs e.g. Hyperlinks on websites hosted by Web servers can lead to code execution on the user's local machine.

- *Buffer Overflow:*
  Web servers may contain unchecked buffers that can be overflowed by sending requests that are larger than the buffer has been defined to hold. This can lead to arbitrary code being executed on the Web server.

- *Cookie Poisoning:*
  Unauthorised information may be extracted by manipulating data inside a cookie. This can trick a Web server into displaying information only authorised to other users.

- *Field Manipulation:*
  Manipulating hidden fields or inserting scripts into visible fields on web pages can result in the Web server behaving in ways that should not normally be allowed.

- *Web Server Misconfiguration:*
  Misconfigurations and default Web server installations often leave Web servers vulnerable to attack. Sample scripts included with Web servers may also contain vulnerabilities.

- *Backdoor and Debug Options:*
  Development or testing code that is not removed when a Web server is released sometimes contains vulnerable code that may allow attackers to perform unauthorised actions on the Web server.

- *SQL Injection:*
  An attacker may be able to manipulate Web servers running databases containing poorly designed input validation routines by injecting SQL code.

- *Directory Traversal Attack:*
  This attack attempts to bypass the Web server's directory access restrictions to view and/or manipulate files on the Web server.[7]

The array of vulnerabilities that exist manifest out of the competitive nature of software companies and the demands for new features by users. These factors often result in software being released without having been written with security in mind or thoroughly tested for vulnerabilities. When the vulnerabilities are discovered, the attacks commence. Vendors usually release a patch or updated version of the vulnerable software as they become aware of the new vulnerability. Once an update has been made available it is up to administrators of Web servers to patch their vulnerable machines. Often the vulnerable Web servers remain vulnerable long after the patches have been released due to Web Administrators not having kept up to date with patches and upgrades for their product. Attackers scan the WWW for Web servers with known vulnerabilities which when located often lead to the compromise of those servers. The lack of secure software writing practices and testing procedures coupled with the demand for new features and lack of attention to product updates ensure there will always be exploitable vulnerabilities in Web servers.

# 2. Specific Exploit

## 2.1 Exploit Details

| | |
|---|---|
| Name: | Windows XP HCP URI Handler Abuse Vulnerability |
| Bugtraq: | BID 5478 |
| CVE: | CAN-2002-0974 |
| CERT: | Vulnerability Note: VU#489721 |
| Microsoft: | MS02-060 |
| Operating Systems Impacted: | Microsoft Windows XP Home<br>Microsoft Windows XP Professional<br>Microsoft Windows XP 64-bit Edition |
| Protocols exploit uses: | HTTP, HCP |
| Services exploit uses: | WWW, Web Server, Internet Explorer, Help & Support Center, (can be used with Outlook Express) |
| Brief Description: | A vulnerability in the Windows XP Help and Support Center allows an attacker to delete a file or directory on a client machine via a hyperlink. |
| Variants: | CAN-2001-0909<br>Buffer overflow in helpctr.exe program in Microsoft Help and Support Center for Windows XP.<br>CAN-2003-0009<br>Cross-site scripting (XSS) vulnerability in Help and Support Center for Microsoft Windows Me. |

## 2.2 Description of variants

There are two variants of the Windows XP HCP URI Handler Abuse Vulnerability. The two variants including the exploit discussed in this paper all suffer from poor input parameter checking which can lead to the successful exploitation of vulnerable systems.

1. Buffer overflow in helpctr.exe program in Microsoft Help Center for Windows XP allows remote attackers to execute arbitrary code via a long hcp: URL.

An attacker can exploit this vulnerability by making a HCP request with an overly long string. This will trigger the overflow condition and may result in malicious attacker-supplied code being executed on the vulnerable system. This vulnerability is similar to the 'Windows XP HCP URI Handler Abuse Vulnerability" but is exploiting the vulnerable system via a buffer overflow.

*Additional Information:*

CAN-2001-0909
URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0909

Security Focus
URL: http://www.securityfocus.com/bid/6802

2. Cross-site scripting (XSS) vulnerability in Help and Support Center for Microsoft Windows Me.

This vulnerability is similar to the 'Windows XP HCP URI Handler Abuse Vulnerability'. This attack executes arbitrary scripts via a hcp:// URL with a malicious script in the topic parameter. Windows ME does not use the vulnerable Uplddrvinfo.htm used to exploit vulnerable Windows XP systems.

*Additional Information:*

CAN-2003-0009
URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0009

Microsoft Security Bulletin: MS03-006
URL:
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS03-006.asp

## 2.3 Protocol used by the exploit

The HTTP protocol is used initially for the communication between the victim's web browser and the attackers Web server. Once the web page containing the dangerous hyperlink has loaded into the victim's browser and has been clicked, the Help and Support Center's custom URL protocol HCP, is used to carry out the attack.

Custom URL protocols are designed to provide specialized behaviour to existing applications that are based on Microsoft's WebBrowser control. URI protocol handlers allow applications to register custom URL protocols. These protocols need to be registered with an application's URI protocol handler for the custom URL to be understood. When the custom protocol is invoked the associated application will launch automatically. The HCP (Help Center Pluggable Protocol), is a custom URL protocol and is used for navigation within the Windows Help and Support Center. HCP can also be used to launch the Help and Support Center from applications that support URL based functionality and custom URL protocols, e.g. Internet Explorer 4.0 and higher.[8]

Internet Explorer on Windows XP comes with a URI protocol handler for the Help and Support Center application. Typing "hcp:" into the address bar of Internet Explorer on Windows XP computers will cause the Help and Support Center application to launch. The HCP protocol can be specified in a hyperlink using the "hcp:" prefix, e.g.
"<a href="hcp://system/sysinfo/msinfo.htm">System Information</a>".
When the hyperlink is submitted via Internet Explorer the Help and Support Center application is launched and the appropriate web page is displayed.

## 2.4 How the exploit works

The Windows Help and Support Center includes a feature that executes after the “Found New Hardware Wizard” completes. The feature prompts a user to send their system’s hardware profile to Microsoft to obtain support for hardware installed on the user’s machine. If the user agrees to send this information to Microsoft, the Help and Support Center uses the Uplddrvinfo.htm file to send the user’s hardware profile to the Microsoft Driver Feedback server by using the Upload Manager service.[9]

The exploit takes advantage of an ActiveX control defined in the Uplddrvinfo.htm file (Figure 3).

```
var oFSO = new ActiveXObject( "Scripting.FileSystemObject" );
try
{
oFSO.DeleteFile( sFile );
```

**Figure 3.** Vulnerable ActiveX Control Found in Uplddrvinfo.htm File

This ActiveX control accepts filenames from the Help and Support Center’s URI handler where (**sFile**) is derived from a HCP URL. By using the “hcp://” prefix it is possible to pass a HCP URL to the ActiveX control’s (sFile) parameter. Specifying a file name or directory to the (sFile) parameter will cause the Help and Support Center to launch, and when terminated will delete the specified file or directory, e.g. typing “hcp://system/DFS/uplddrvinfo.htm?file://C:\windows\notepad.exe” (without quotations) into the Internet Explorer address bar and pressing enter, will cause the Help and Support Center to launch and then delete the notepad application when the Help and Support Center is terminated.

The HCP protocol can be embedded within HTML hyperlinks, which can be executed from a HTML based web page or email. Internet Explorer runs HCP requests with limited restrictions and its URI handler does not adequately validate HCP parameters, allowing the ActiveX control to execute without prompting the user. Sending a filename as a parameter to the ActiveX control via a hyperlink allows for the deletion of the file or directory specified, if the hyperlink is clicked. When the hyperlink is clicked the Help and Support Center is launched. On termination of the Help and Support Center application the files specified in the hcp hyperlink are deleted.

The exploit code to be used in a HTML document has four parts:

- the HCP protocol prefix:
  hcp:

- the directory path pointing to the vulnerable Uplddrvinfo.htm file:
  //system/DFS/Uplddrvinfo.htm

- the filename or directory to be deleted
  e.g. file://c:\test\* (a "*" can be used as a wildcard to specify all files in a directory.

- surrounding hyperlink tags and the text that the hyperlink will display
  e.g. <a href="   ">Click Me</a>

```
<a href="hcp://system/DFS/Uplddrvinfo.htm?file://c:\test\*">Click Me</a>
```

This code as an example will delete all files in C:\test. Though this directory does not exist by default, the filename or directory can be specified to delete any file or directory. This exploit can also be used to target computers running certain versions of Outlook and Outlook Express by sending the victim a HTML based email with an embedded HCP hyperlink.

Certain restrictions apply to the deletion of files:

1. All files can be deleted (including files with the hidden and archived attributes set), except files that have the read-only attributes set.

2. If a directory has been targeted for deletion, all files will be deleted until a read-only file is found. If the first file in a directory has been set to read-only then no files will be deleted.

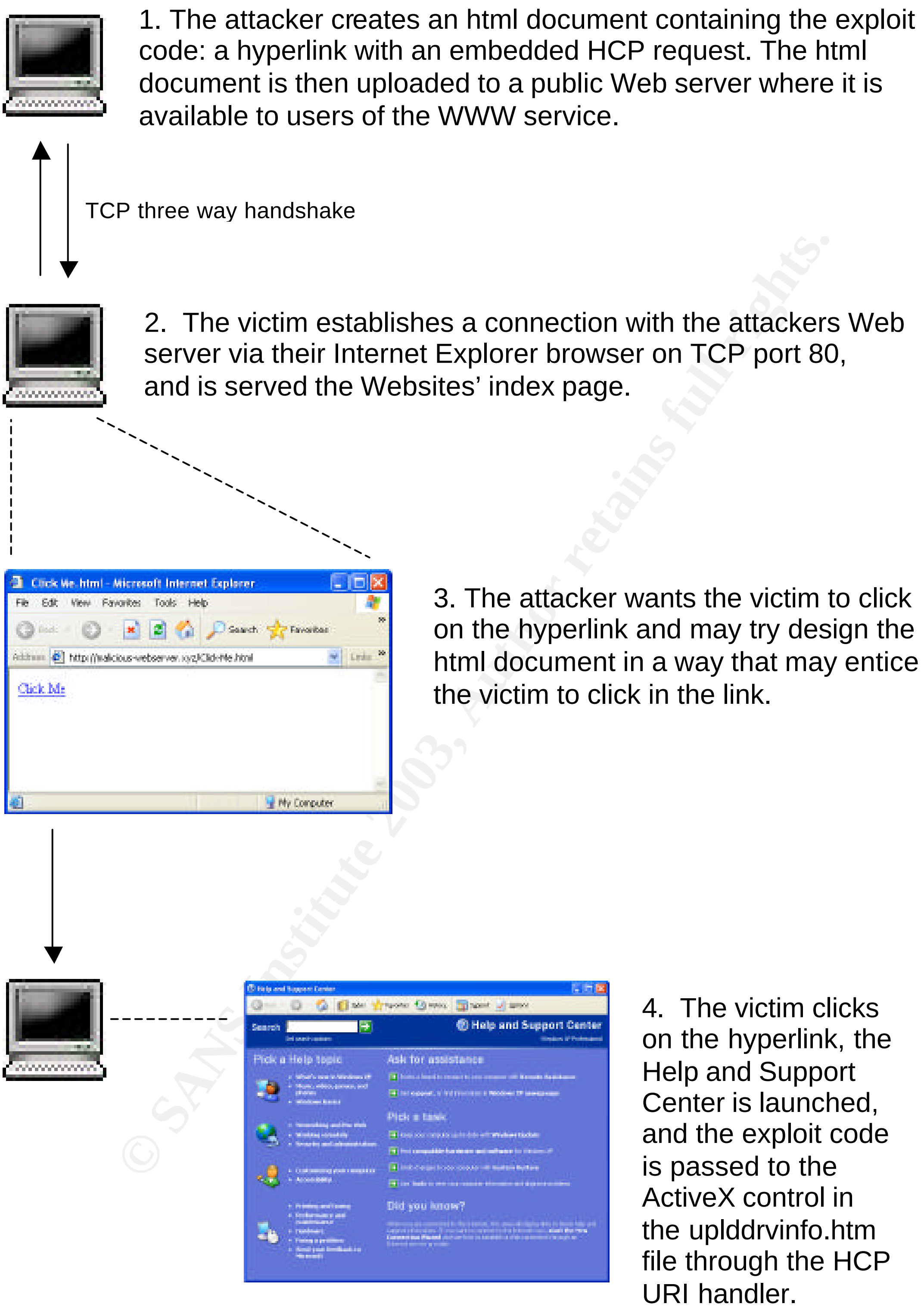
3. If a single file is targeted and appears after files that have been set to read-only, the file will be deleted.

4. Sub-directories of the deleted directory are unaffected.

5. Windows XP uses its Automatic System Recovery utility to automatically restore system files if they are deleted. However if certain system files such as c_1252.nls are deleted then this will prevent the system from rebooting successfully.

6. There have been various claims and a demonstration on TechTV that the windows systems directory (e.g. C:\WINDOWS) can be deleted.[10] After testing this a number of times, it doesn't appear to work. TechTV claimed that hidden files could not be deleted, but testing showed that only read-only files were not able to be deleted.

## 2.5 Exploit Diagram



1. The attacker creates an html document containing the exploit code: a hyperlink with an embedded HCP request. The html document is then uploaded to a public Web server where it is available to users of the WWW service.

TCP three way handshake

2. The victim establishes a connection with the attackers Web server via their Internet Explorer browser on TCP port 80, and is served the Websites' index page.

3. The attacker wants the victim to click on the hyperlink and may try design the html document in a way that may entice the victim to click in the link.

4. The victim clicks on the hyperlink, the Help and Support Center is launched, and the exploit code is passed to the ActiveX control in the uplddrvinfo.htm file through the HCP URI handler.

5. On termination of the Help and Support Center application, the ActiveX control deletes the files specified in the hyperlink.

## 2.6 How to use the exploit

1. Create an HTML document and a HCP Hyperlink. The hyperlink needs to specify the path to the Uplddrvinfo.htm file, which contains the ActiveX control. The hyperlink also needs to specify a file name or directory to pass to the ActiveX control as a parameter.

   *Example exploit hyperlink:*

   ```
   <a href="hcp://system/DFS/Uplddrvinfo.htm?file://C:\test\*">Click
   Me</a>
   ```

   This example code will delete all files in C:\test up to and excluding files with the read-only attribute set.

   *Basic HTML web page code containing the exploit hyperlink:*

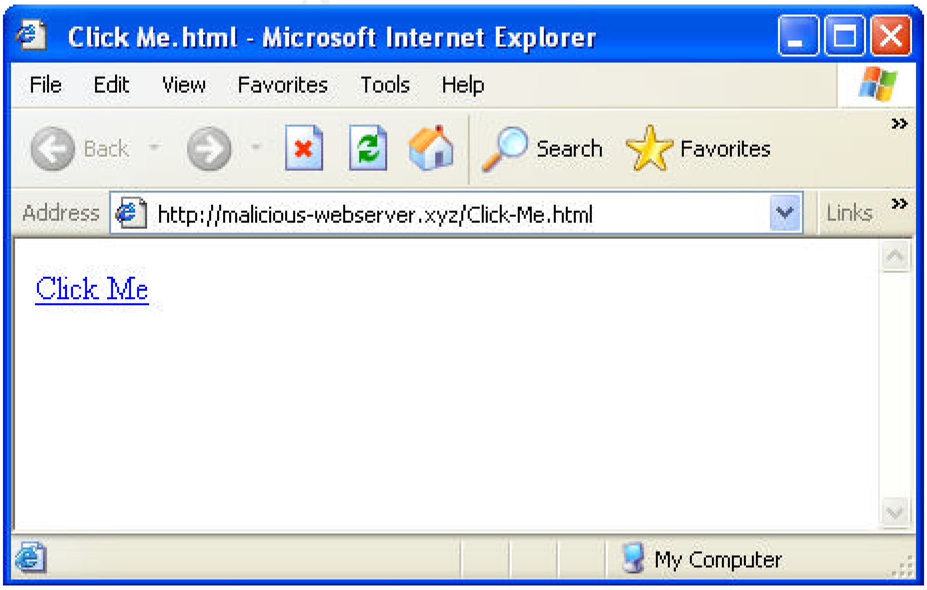
   E.g.

   ```
      <html>
        <body>

   <a href="hcp://system/DFS/Uplddrvinfo.htm?file://c:\test\*">Click
   Me</a>

        </body>
      </html>
   ```

   This code will create a simple HTML as shown in Figure 5.


Click Me.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Links
My Computer

Address: http://malicious-webserver.xyz/Click-Me.html

Click Me

**Figure 5.** Example Exploit Web Page

2. Upload the html document to a public Web server where it is available to users of the WWW service.

It is the attacker's desire for the victim to click on the hyperlink. The attacker may try to encourage the victim to visit the malicious website and click on a hyperlink by making the web page inviting to the user. An attacker may try to attract users by adding certain keywords to a web page's metadata so it will be picked up in search engines. Creating a front page for a web site that requires the user to "click to enter" is another method an attacker may employ to encourage the user to click on a malicious hyperlink (Figure 6).

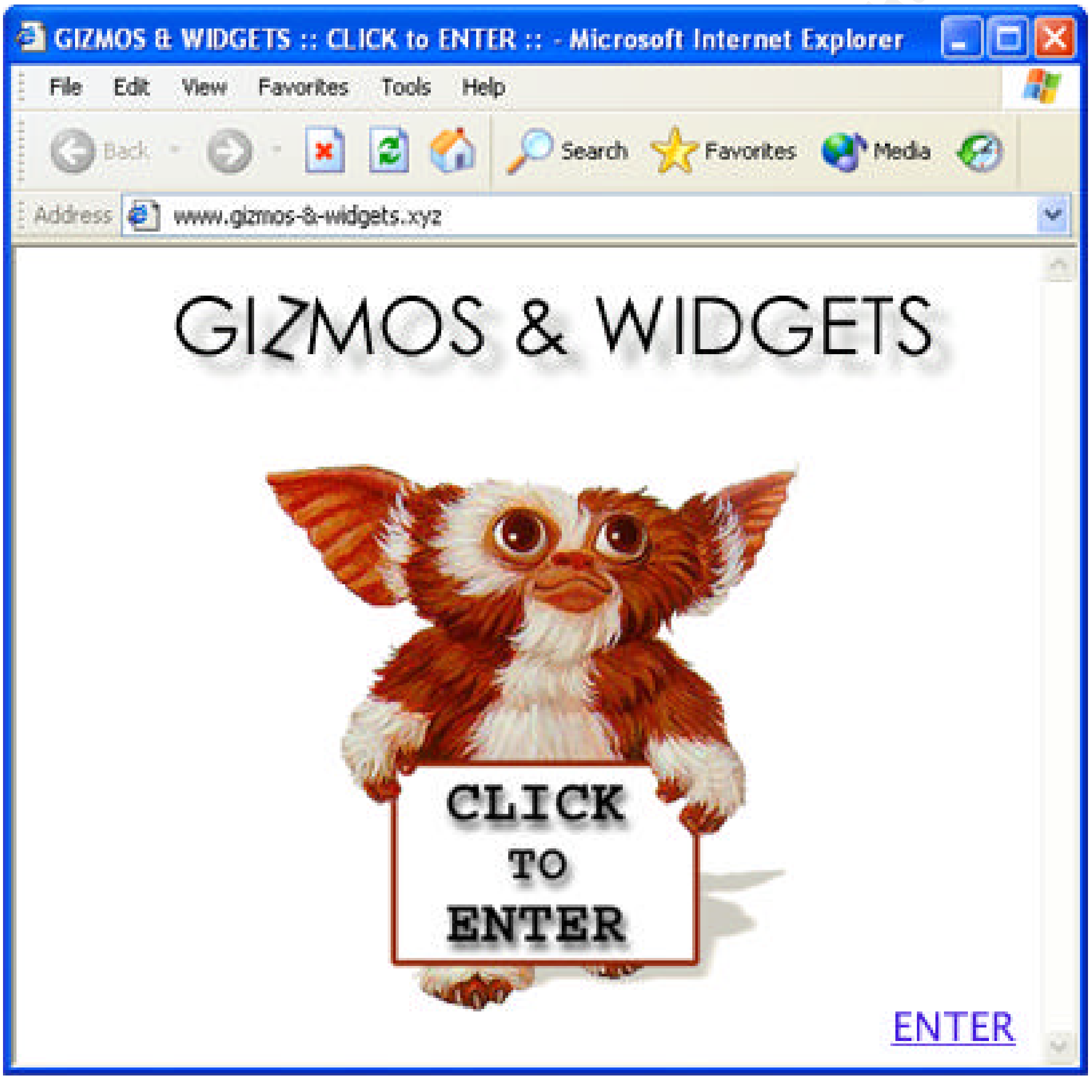
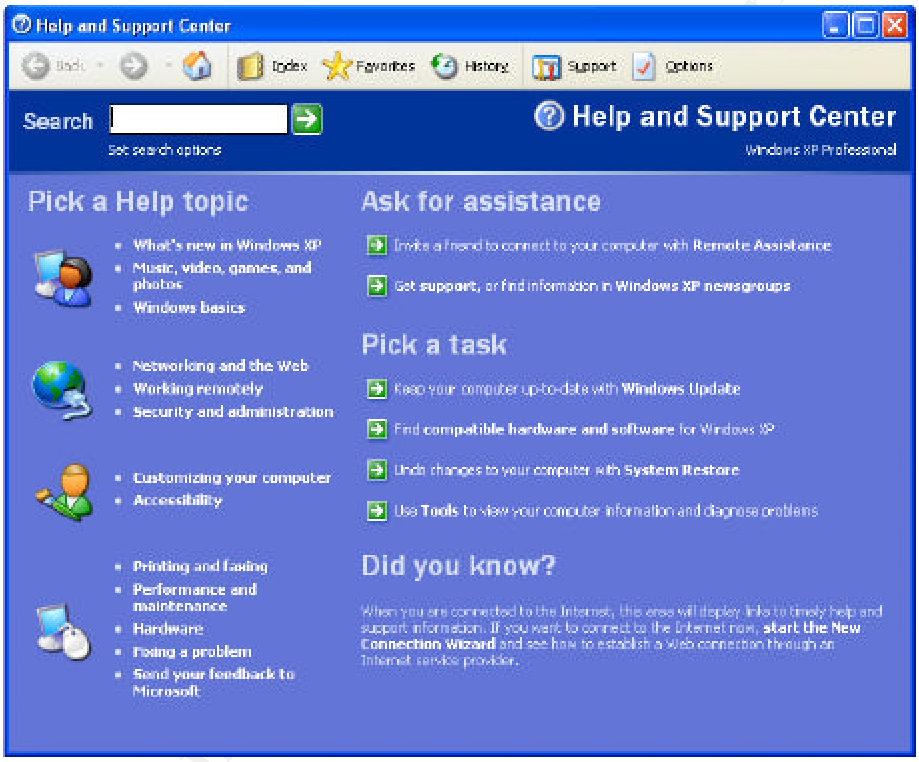


**Figure 6.** Website that Requires Users to Click on a Hyperlink
This is an example of a webpage that uses hyperlinks for entry into the website.
To enter the website a user must click on the hyperlinked text or picture as shown above.

## 2.7 Signature of the attack

Once the hyperlink has been clicked the Help and Support Center is launched (Figure 7). An unsuspecting user may find this abnormal. If the user knows about this vulnerability and has not patched their machine they can check which files the hyperlink specifies to delete by checking the browsers status bar while placing the cursor over the link or viewing the page's source code. Once the Help and Support Center application is terminated the files will be deleted. The user would need to copy or move the files that are going to be deleted into a different directory before the application is terminated. An alternative to moving or copying the files, would be to set the attributes of files in that directory to read-only before terminating the application.


Help and Support Center
Back
Index
Favorites
History
Support
Options
Search
Set search options
Help and Support Center
Windows XP Professional
Pick a Help topic
What's new in Windows XP
Music, video, games, and photos
Windows basics
Networking and the Web
Working remotely
Security and administration
Customizing your computer
Accessibility
Printing and faxing
Performance and maintenance
Hardware
Fixing a problem
Send your feedback to Microsoft
Ask for assistance
Invite a friend to connect to your computer with Remote Assistance
Get support, or find information in Windows XP newsgroups
Pick a task
Keep your computer up-to-date with Windows Update
Find compatible hardware and software for Windows XP
Undo changes to your computer with System Restore
Use Tools to view your computer information and diagnose problems
Did you know?

When you are connected to the Internet, this area will display links to timely help and support information. If you want to connect to the Internet now, **start the New Connection Wizard** and see how to establish a Web connection through an Internet service provider.

**Figure 7.** Help and Support Center Default Screen

During testing, network sniffers did not sniff the HCP attack because there was no traffic moving across the network. Once the browser has loaded a copy of the web page into memory it no longer requires resources from the originating Web server and therefore generates no TCP traffic. The link is essentially run locally and points to files on the local machine.

For the purposes of detection an external program or client-side plugin could be written to check for hcp content while browsing, that would alert and/or block hcp content prior to clicking on the hcp hyperlink. The most efficient way of blocking this sort of attack is to patch the vulnerable system using one of the methods provided in section 2.8.

## 2.8 How to protect against it

There are a number of ways to address this HCP vulnerability:

1. Download and apply Windows XP Service Pack 1, which includes a fix for the HCP vulnerability (max. 140 megabytes).

or

2. For users that do not wish to install the service pack a number of alternative options are available:

A. Download and apply Microsoft's official patch for this issue (1.35 megabytes).

- Microsoft Windows XP:
http://www.microsoft.com/Downloads/Release.asp?ReleaseID=43681

- Microsoft Windows XP 64-bit Edition:
http://www.microsoft.com/Downloads/Release.asp?ReleaseID=43676

or

B. Download and apply the “XPdite” exploit patcher from Gibson Research Corporation (30KB, URL: http://grc.com/files/xpdite.exe).

or

C. Deregister the HCP protocol.

- o Run the Windows Registry Editor (regedit.exe)
- o Locate the key:
  HKEY_CLASSES_ROOT\hcp\shell\open\command
- o Create a new string data item called DefaultBackup, and give it a value equal to that of the (Default) data item
- o Set the (Default) data item's value to the empty string

Note: Deregistering the hcp protocol will also disable parts of the Help and Support Center.

or

D: Rename or delete the Uplddrvinfo.htm file.

## 2.9 Source code / Pseudo Code

Example exploit code:

```
<html>
  <body>

<a href="hcp://system/DFS/Uplddrvinfo.htm?file://c:\test\*">Click Me</a>

  </body>
</html>
```

This code will delete all files in C:\test only limited by the rules specified in section 2.4.

Step 1:
The attacker has created a HTML document containing the exploit code, a hyperlink with an embedded HCP request. The HTML document is then uploaded to a public Web server where it is available to users of the WWW service.

Step 2:
The victim clicks on the “Click Me” hyperlink which executes a HCP request. The HCP request contains a call to the Uplddrvinfo.htm file which includes the ActiveX control that takes the file part of the HCP request file://C:\test\* as the ( **sFile** ) parameter (Figure 8).

```
var oFSO = new ActiveXObject( "Scripting.FileSystemObject" );
try
{
oFSO.DeleteFile( sFile );
```

**Figure 8.** Vulnerable ActiveX Control Found in Uplddrvinfo.htm File

Step 3:
The Help and Support Center application is launched to the victim’s surprise and file://C:\test\* is sent to the ( **sFile** ) parameter of the ActiveX control as specified in the hyperlink. Because the browser runs the HCP request with relaxed restrictions, the user is will not be prompted when the ActiveX control is executed.

Step 4:
The Help and Support Center application displays a web page in its interface stating “The wizard was unable to find the necessary software for your new hardware”. On terminating the Help and Support Center application the “oFSO.DeleteFile( **sFile** )” code of the ActiveX control is executed, deleting

the C:\test directory. The files in C:\test will be deleted in accordance with the rules specified in section 2.4.

*Link to source code:*

Bugtraq:
Delete arbitrary files using Help and Support Center [MSRC 1198dg]
URL: http://cert-uni-stuttgart.de/archive/bugtraq/2002/08/msg00224.html

## 2.10 Additional Information

*Description & Information:*

Security Focus:
Microsoft Windows XP HCP URI Handler Abuse Vulnerability
URL: http://www.securityfocus.com/bid/5478/info

CERT:
Vulnerability Note VU#489721
URL: http://www.kb.cert.org/vuls/id/489721

Common Vulnerabilities and Exposures:
CVE: CAN-2002-0974
URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0974

Microsoft:
MS02-060 Flaw in Windows XP Help and Support Center Could Enable File Deletion
URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;328940

GRC:
XPdite: Quickly replace a dangerous Windows XP file
URL: http://grc.com/xpdite/xpdite.htm

*Source Code:*

Bugtraq:
Delete arbitrary files using Help and Support Center [MSRC 1198dg]
URL: http://cert-uni-stuttgart.de/archive/bugtraq/2002/08/msg00224.html

*News Related:*

The Register
Win-XP Help Center request wipes your HD
URL: http://www.theregister.co.uk/content/4/27074.html

TechTV, Leo Laporte
Demonstration of the vulnerability
URL:
http://cgi.techtv.com/mediamodule?action=view&version=20020910095425&video_src=/thescreensavers/2002/ss020909c&width=320&height=240&vidsection=3200042&add_date=1031641200&start=&end=&duration=&bitrates=

# 3. Appendices

Appendix 1 shows an extensive list of vulnerabilities for port 80 recorded in the Common Vulnerabilities & Exposures List.

**Appendix 1.** Common Vulnerabilities & Exposures List, Port 80, as of 17 / 6 / 2003. [11]

| Name | Vulnerability |
|---|---|
| CVE-2001-0987 | Cross-site scripting vulnerability in CGIWrap before 3.7 allows remote attackers to execute arbitrary Javascript on other web clients by causing the Javascript to be inserted into error messages that are generated by CGIWrap. |
| CVE-2001-0805 | Directory traversal vulnerability in ttawebtop.cgi in Tarantella Enterprise 3.00 and 3.01 allows remote attackers to read arbitrary files via a .. (dot dot) in the pg parameter. |
| CVE-2001-0463 | Directory traversal vulnerability in cal_make.pl in PerlCal allows remote attackers to read arbitrary files via a .. (dot dot) in the p0 parameter. |
| CVE-2001-0021 | MailMan Webmail 3.0.25 and earlier allows remote attackers to execute arbitrary commands via shell metacharacters in the alternate_template paramater. |
| CVE-2001-0009 | Directory traversal vulnerability in Lotus Domino 5.0.5 web server allows remote attackers to read arbitrary files via a .. attack. |
| CVE-2000-1187 | Buffer overflow in the HTML parser for Netscape 4.75 and earlier allows remote attackers to execute arbitrary commands via a long password value in a form field. |
| CVE-2000-1171 | Directory traversal vulnerability in cgiforum.pl script in CGIForum 1.0 allows remote attackers to ready arbitrary files via a .. (dot dot) attack in the "thesection" parameter. |
| CVE-2000-1024 | eWave ServletExec 3.0C and earlier does not restrict access to the UploadServlet Java/JSP servlet, which allows remote attackers to upload files and execute arbitrary commands. |
| CVE-2000-1005 | Directory traversal vulnerability in html_web_store.cgi and web_store.cgi CGI programs in eXtropia WebStore allows remote attackers to read arbitrary files via a .. (dot dot) attack on the page parameter. |
| CVE-2000-0975 | Directory traversal vulnerability in apexec.pl in Anaconda Foundation Directory allows remote attackers to read arbitrary files via a .. (dot dot) attack. |
| CVE-2000-0922 | Directory traversal vulnerability in Bytes Interactive Web Shopper shopping cart program (shopper.cgi) 2.0 and earlier allows remote attackers to read arbitrary files via a .. (dot dot) attack on the newpage parameter. |
| CVE-2000-0869 | The default configuration of Apache 1.3.12 in SuSE Linux 6.4 enables WebDAV, which allows remote attackers to list arbitrary diretories via the PROPFIND HTTP request method. |
| CVE-2000-0853 | YaBB Bulletin Board 9.1.2000 allows remote attackers to read arbitrary files via a .. (dot dot) attack. |
| CVE-2000-0726 | CGIMail.exe CGI program in Stalkerlab Mailers 1.1.2 allows remote attackers to read arbitrary files by specifying the file in the $Attach$ hidden form variable. |
| CVE-2000-0677 | Buffer overflow in IBM Net.Data db2www CGI program allows remote |

| | |
|---|---|
| | attackers to execute arbitrary commands via a long PATH_INFO environmental variable. |
| CVE-2000-0671 | Roxen web server earlier than 2.0.69 allows allows remote attackers to bypass access restrictions, list directory contents, and read source code by inserting a null character (%00) to the URL. |
| CVE-2000-0670 | The cvsweb CGI script in CVSWeb 1.80 allows remote attackers with write access to a CVS repository to execute arbitrary commands via shell metacharacters. |
| CVE-2000-0638 | Big Brother 1.4h1 and earlier allows remote attackers to read arbitrary files via a .. (dot dot) attack. |
| CVE-2000-0630 | IIS 4.0 and 5.0 allows remote attackers to obtain fragments of source code by appending a +.htr to the URL, a variant of the "File Fragment Reading via .HTR" vulnerability. |
| CVE-2000-0628 | The source.asp example script in the Apache ASP module Apache::ASP 1.93 and earlier allows remote attackers to modify files. |
| CVE-2000-0627 | BlackBoard CourseInfo 4.0 does not properly authenticate users, which allows local users to modify CourseInfo database information and gain privileges by directly calling the supporting CGI programs such as user_update_passwd.pl and user_update_admin.pl. |
| CVE-2000-0538 | ColdFusion Administrator for ColdFusion 4.5.1 and earlier allows remote attackers to cause a denial of service via a long login password. |
| CVE-2000-0439 | Internet Explorer 4.0 and 5.0 allows a malicious web site to obtain client cookies from another domain by including that domain name and escaped characters in a URL, aka the "Unauthorized Cookie Access" vulnerability. |
| CVE-2000-0432 | The calender.pl and the calendar_admin.pl calendar scripts by Matt Kruse allow remote attackers to execute arbitrary commands via shell metacharacters. |
| CVE-2000-0382 | ColdFusion ClusterCATS appends stale query string arguments to a URL during HTML redirection, which may provide sensitive information to the redirected site. |
| CVE-2000-0381 | The Gossamer Threads DBMan db.cgi CGI script allows remote attackers to view environmental variables and setup information by referencing a non-existing database in the db parameter. |
| CVE-2000-0322 | The passwd.php3 CGI script in the Red Hat Piranha Virtual Server Package allows local users to execure arbitrary commands via shell metacharacters. |
| CVE-2000-0303 | Quake3 Arena allows malicious server operators to read or modify files on a client via a dot dot (..) attack. |
| CVE-2000-0282 | TalentSoft webpsvr daemon in the Web+ shopping cart application allows remote attackers to read arbitrary files via a .. (dot dot) attack on the webplus CGI program. |
| CVE-2000-0260 | Buffer overflow in the dvwssr.dll DLL in Microsoft Visual Interdev 1.0 allows users to cause a denial of service or execute commands, aka the "Link View Server-Side Component" vulnerability. |
| CVE-2000-0252 | The dansie shopping cart application cart.pl allows remote attackers to execute commands via a shell metacharacters in a form variable. |
| CVE-2000-0236 | Netscape Enterprise Server with Directory Indexing enabled allows remote attackers to list server directories via web publishing tags such as ?wp-ver-info and ?wp-cs-dump. |
| CVE-2000-0208 | The htdig (ht://Dig) CGI program htsearch allows remote attackers to read arbitrary files by enclosing the file name with backticks (`) in parameters to htsearch. |
| CVE-2000-0207 | SGI InfoSearch CGI program infosrch.cgi allows remote attackers to |

| | |
|---|---|
| | execute commands via shell metacharacters. |
| CVE-2000-0192 | The default installation of Caldera OpenLinux 2.3 includes the CGI program rpm_query, which allows remote attackers to determine what packages are installed on the system. |
| CVE-2000-0169 | Batch files in the Oracle web listener ows-bin directory allow remote attackers to execute commands via a malformed URL that includes '?&'. |
| CVE-2000-0169 | Batch files in the Oracle web listener ows-bin directory allow remote attackers to execute commands via a malformed URL that includes '?&'. |
| CVE-2000-0127 | The Webspeed configuration program does not properly disable access to the WSMadmin utility, which allows remote attackers to gain privileges. |
| CVE-2000-0057 | Cold Fusion CFCACHE tag places temporary cache files within the web document root, allowing remote attackers to obtain sensitive system information. |
| CVE-2000-0039 | AltaVista search engine allows remote attackers to read files above the document root via a .. (dot dot) in the query.cgi CGI program. |
| CVE-1999-1550 | bigconf.conf in F5 BIG/ip 2.1.2 and earlier allows remote attackers to read arbitrary files by specifying the target file in the "file" parameter. |
| CVE-1999-1011 | The Remote Data Service (RDS) DataFactory component of Microsoft Data Access Components (MDAC) in IIS 3.x and 4.x exposes unsafe methods, which allows remote attackers to execute arbitrary commands. |
| CVE-1999-0953 | WWWBoard stores encrypted passwords in a password file that is under the web root and thus accessible by remote attackers. |
| CVE-1999-0951 | Buffer overflow in OmniHTTPd CGI program imagemap.cgi allows remote attackers to execute commands. |
| CVE-1999-0947 | AN-HTTPd provides example CGI scripts test.bat, input.bat, input2.bat, and envout.bat, which allow remote attackers to execute commands via shell metacharacters. |
| CVE-1999-0937 | BNBForm allows remote attackers to read arbitrary files via the automessage hidden form variable. |
| CVE-1999-0936 | BNBSurvey survey.cgi program allows remote attackers to execute commands via shell metacharacters. |
| CVE-1999-0934 | classifieds.cgi allows remote attackers to read arbitrary files via shell metacharacters. |
| CVE-1999-0874 | Buffer overflow in IIS 4.0 allows remote attackers to cause a denial of service via a malformed request for files with .HTR, .IDC, or .STM extensions. |
| CVE-1999-0710 | The RedHat squid program installs cachemgr.cgi in a public web directory, allowing remote attackers to use it as an intermediary to connect to other systems. |
| CVE-1999-0612 | A version of finger is running that exposes valid user information to any entity on the network. |
| CVE-1999-0474 | The ICQ Webserver allows remote attackers to use .. to access arbitrary files outside of the user's personal directory. |
| CVE-1999-0407 | By default, IIS 4.0 has a virtual directory /IISADMPWD which contains files that can be used as proxies for brute force password attacks, or to identify valid users on the system. |
| CVE-1999-0346 | CGI PHP mlog script allows an attacker to read any file on the target server. |
| CVE-1999-0278 | In IIS. remote attackers can obtain source code for ASP files bv |

| | |
|---|---|
| | appending "::$DATA" to the URL. |
| CVE-1999-0276 | mSQL v2.0.1 and below allows remote execution through a buffer overflow. |
| CVE-1999-0276 | mSQL v2.0.1 and below allows remote execution through a buffer overflow. |
| CVE-1999-0270 | pfdispaly CGI program for SGI's Performer API Search Tool allows read access to files. |
| CVE-1999-0269 | Netscape Enterprise servers may list files through the PageServices query. |
| CVE-1999-0266 | The info2www CGI script allows remote file access or remote command execution. |
| CVE-1999-0264 | htmlscript CGI program allows remote read access to files. |
| CVE-1999-0262 | faxsurvey CGI script on Linux allows remote command execution via shell metacharacters. |
| CVE-1999-0260 | The jj CGI program allows command execution via shell metacharacters. |
| CVE-1999-0237 | Remote execution of arbitrary commands through Guestbook CGI program. |
| CVE-1999-0236 | ScriptAlias directory in NCSA and Apache httpd allowed attackers to read CGI programs. |
| CVE-1999-0233 | IIS allows users to execute arbitrary commands using .bat or .cmd files. |
| CVE-1999-0196 | The websendmail program in the Webgais program allows a remote user to access arbitrary files. |
| CVE-1999-0191 | IIS newdsn.exe CGI script allows remote users to overwrite files. |
| CVE-1999-0178 | The win-c-sample program in the WebSite web server has a buffer overflow that allows remote execution of commands. |
| CVE-1999-0177 | The uploader program in the WebSite web server allows a remote attacker to execute arbitrary programs. |
| CVE-1999-0176 | The Webgais program allows a remote user to execute arbitrary commands. |
| CVE-1999-0175 | The convert.bas program in the Novell web server allows a remote attackers to read any file on the system that is internally accessible by the web server. |
| CVE-1999-0174 | The view-source CGI program allows remote attackers to read arbitrary files via a .. (dot dot) attack. |
| CVE-1999-0172 | FormMail CGI program allows remote execution of commands. |
| CVE-1999-0149 | The wrap CGI program in IRIX allows remote attackers to view arbitrary directory listings via a .. (dot dot) attack. |
| CVE-1999-0148 | The handler CGI program in IRIX allows arbitrary command execution. |
| CVE-1999-0147 | The aglimpse CGI program of the Glimpse package allows remote execution of arbitrary commands |
| CVE-1999-0146 | The campas CGI program provided with some NCSA web servers allows an attacker to read arbitrary files. |
| CVE-1999-0070 | test-cgi program allows an attacker to list files on the server |
| CVE-1999-0067 | CGI phf program allows remote command execution through shell metacharacters. |
| CVE-1999-0066 | AnyForm CGI remote execution |
| CVE-1999-0045 | List of arbitrary files on Web host via nph-test-cgi script |
| CVE-1999-0039 | Arbitrary command execution using webdist CGI program in IRIX. |
| CVE-1999-0021 | Arbitrary command execution via buffer overflow in Count.cgi (wwwcount) cgi-bin program. |

| | |
|---|---|
| CAN-2002-0011 | Information leak in doeditvotes.cgi in Bugzilla before 2.14.1 may allow remote attackers to more easily conduct attacks on the login. |
| CAN-2001-1209 | Directory traversal vulnerability in zml.cgi allows remote attackers to read arbitrary files via a .. (dot dot) in the file parameter. |
| CAN-2001-1115 | generate.cgi in SIX-webboard 2.01 and before allows remote attackers to read arbitrary files via a dot dot (..) in the content parameter. |
| CAN-2001-1014 | eshop.pl in WebDiscount(e)shop allows remote attackers to execute arbitrary commands via shell metacharacters in the seite parameter. |
| CAN-2001-0997 | Textor Webmasters Ltd listrec.pl CGI program allows remote attackers to execute arbitrary commands via shell metacharacters in the TEMPLATE parameter. |
| CAN-2001-0871 | Directory traversal vulnerability in HTTP server for Alchemy Eye and Alchemy Network Monitor allows remote attackers to execute arbitrary commands via an HTTP request containing (1) a .. in versions 2.0 through 2.6.18, or (2) a DOS device name followed by a .. in versions 2.6.19 through 3.0.10. |
| CAN-2001-0821 | The default configuration of DCShop 1.002 beta places sensitive files in the cgi-bin directory, which could allow remote attackers to read sensitive data via an HTTP GET request for (1) orders.txt or (2) auth_user_file.txt. |
| CAN-2001-0780 | Directory traversal vulnerability in cosmicpro.cgi in Cosmicperl Directory Pro 2.0 allows remote attacker to gain sensitive information via a .. (dot dot) in the SHOW parameter. |
| CAN-2001-0561 | Directory traversal vulnerability in Drummond Miles A1Stats prior to 1.6 allows a remote attacker to read arbitrary files via a '..' (dot dot) attack in (1) a1disp2.cgi, (2) a1disp3.cgi, or (3) a1disp4.cgi. |
| CAN-2001-0476 | Multiple buffer overflows in s.cgi program in Aspseek search engine 1.03 and earlier allow remote attackers to execute arbitrary commands via (1) a long HTTP query string, or (2) a long tmpl paramater. |
| CAN-2001-0436 | dcboard.cgi in DCForum 2000 1.0 allows remote attackers to execute arbitrary commands by uploading a Perl program to the server and using a .. (dot dot) in the AZ parameter to reference the program. |
| CAN-2001-0400 | nph-maillist.pl allows remote attackers to execute arbitrary commands via shell metacharacters ("`") in the email address. |
| CAN-2001-0305 | Directory traversal vulnerability in store.cgi in Thinking Arts ES.One package allows remote attackers to read arbitrary files via a .. (dot dot) in the StartID parameter. |
| CAN-2001-0305 | Directory traversal vulnerability in store.cgi in Thinking Arts ES.One package allows remote attackers to read arbitrary files via a .. (dot dot) in the StartID parameter. |
| CAN-2001-0302 | Buffer overflow in tstisapi.dll in Pi3Web 1.0.1 web server allows remote attackers to cause a denial of service, and possibly execute arbitrary commands, via a long URL. |
| CAN-2001-0291 | Buffer overflow in post-query sample CGI program allows remote attackers to execute arbitrary commands via an HTTP POST request that contains at least 10001 parameters. |
| CAN-2001-0272 | Directory traversal vulnerability in sendtemp.pl in W3.org Anaya Web development server allows remote attackers to read arbitrary files via a .. (dot dot) attack in the templ parameter. |
| CAN-2001-0271 | mailnews.cgi 1.3 and earlier allows remote attackers to execute arbitrary commands via a user name that contains shell metacharacters. |
| CAN-2001-0253 | Directory traversal vulnerability in hsx.cgi program in iWeb Hyperseek 2000 allows remote attackers to read arbitrary files and directories via a |

| | |
|---|---|
| | .. (dot dot) attack in the show parameter. |
| CAN-2001-0251 | The Web Publishing feature in Netscape Enterprise Server 3.x allows remote attackers to cause a denial of service via the REVLOG command. |
| CAN-2001-0250 | The Web Publishing feature in Netscape Enterprise Server 4.x and earlier allows remote attackers to list arbitrary directories under the web server root via the INDEX command. |
| CAN-2001-0232 | newsdesk.cgi in News Desk 1.2 allows remote attackers to read arbitrary files via shell metacharacters. |
| CAN-2001-0223 | Buffer overflow in wwwwais allows remote attackers to execute arbitrary commands via a long QUERY_STRING (HTTP GET request). |
| CAN-2001-0217 | Directory traversal vulnerability in PALS Library System pals-cgi program allows remote attackers to read arbitrary files via a .. (dot dot) in the documentName parameter. |
| CAN-2001-0214 | Way-board CGI program allows remote attackers to read arbitrary files by specifying the filename in the db parameter and terminating the filename with a null byte. |
| CAN-2001-0212 | Directory traversal vulnerability in HIS Auktion 1.62 allows remote attackers to read arbitrary files via a .. (dot dot) in the menue parameter, and possibly execute commands via shell metacharacters. |
| CAN-2001-0211 | Directory traversal vulnerability in WebSPIRS 3.1 allows remote attackers to read arbitrary files via a .. (dot dot) attack on the sp.nextform parameter. |
| CAN-2001-0210 | Directory traversal vulnerability in commerce.cgi CGI program allows remote attackers to read arbitrary files via a .. (dot dot) attack in the page parameter. |
| CAN-2001-0210 | Directory traversal vulnerability in commerce.cgi CGI program allows remote attackers to read arbitrary files via a .. (dot dot) attack in the page parameter. |
| CAN-2001-0075 | Directory traversal vulnerability in main.cgi in Technote allows remote attackers to read arbitrary files via a .. (dot dot) attack in the filename parameter. |
| CAN-2001-0025 | ad.cgi CGI program by Leif Wright allows remote attackers to execute arbitrary commands via shell metacharacters in the file parameter. |
| CAN-2000-1110 | document.d2w CGI program in the IBM Net.Data db2www package allows remote attackers to determine the physical path of the web server by sending a nonexistent command to the program. |
| CAN-2000-0940 | Directory traversal vulnerability in Metertek pagelog.cgi allows remote attackers to read arbitrary files via a .. (dot dot) attack on the "name" or "display" parameter. |
| CAN-2000-0940 | Directory traversal vulnerability in Metertek pagelog.cgi allows remote attackers to read arbitrary files via a .. (dot dot) attack on the "name" or "display" parameter. |
| CAN-2000-0832 | Htgrep CGI program allows remote attackers to read arbitrary files by specifying the full pathname in the hdr parameter. |
| CAN-2000-0832 | Htgrep CGI program allows remote attackers to read arbitrary files by specifying the full pathname in the hdr parameter. |
| CAN-2000-0760 | The Snoop servlet in Jakarta Tomcat 3.1 and 3.0 under Apache reveals sensitive system information when a remote attacker requests a nonexistent URL with a .snp extension. |
| CAN-2000-0709 | The shtml.exe component of Microsoft FrontPage 2000 Server Extensions 1.1 allows remote attackers to cause a denial of service in some components by requesting a URL whose name includes a standard DOS device name. |

| | |
|---|---|
| CAN-2000-0590 | Poll It 2.0 CGI script allows remote attackers to read arbitrary files by specifying the file name in the data_dir parameter. |
| CAN-2000-0429 | A backdoor password in Cart32 3.0 and earlier allows remote attackers to execute arbitrary commands. |
| CAN-2000-0242 | WindMail allows remote attackers to read arbitrary files or execute commands via shell metacharacters. |
| CAN-2000-0153 | FrontPage Personal Web Server (PWS) allows remote attackers to read files via a .... (dot dot) attack. |
| CAN-2000-0122 | Frontpage Server Extensions allows remote attackers to determine the physical path of a virtual directory via a GET request to the htimage.exe CGI program. |
| CAN-2000-0079 | The W3C CERN httpd HTTP server allows remote attackers to determine the real pathnames of some commands via a request for a nonexistent URL. |
| CAN-2000-0071 | IIS 4.0 allows a remote attacker to obtain the real pathname of the document root by requesting non-existent files with .ida or .idq extensions. |
| CAN-2000-0066 | WebSite Pro allows remote attackers to determine the real pathname of webdirectories via a malformed URL request. |
| CAN-2000-0054 | search.cgi in the SolutionScripts Home Free package allows remote attackers to view directories via a .. (dot dot) attack. |
| CAN-1999-1479 | The textcounter.pl by Matt Wright allows remote attackers to execute arbitrary commands via shell metacharacters. |
| CAN-1999-1462 | Vulnerability in bb-hist.sh CGI History module in Big Brother 1.09b and 1.09c allows remote attacker to read portions of arbitrary files. |
| CAN-1999-1374 | perlshop.cgi shopping cart program stores sensitive customer information in directories and files that are under the web root, which allows remote attackers to obtain that information via an HTTP request. |
| CAN-1999-1232 | day5datacopier in SGI IRIX 6.2 trusts the PATH environmental variable to find the "cp" program, which allows local users to execute arbitrary commands by modifying the PATH to point to a Trojan horse cp program. |
| CAN-1999-1179 | Vulnerability in man.sh CGI script, included in May 1998 issue of SysAdmin Magazine, allows remote attackers to execute arbitrary commands. |
| CAN-1999-1178 | Sambar Server 4.1 beta allows remote attackers to obtain sensitive information about the server via an HTTP request for the dumpenv.pl script. |
| CAN-1999-1154 | LakeWeb Filemail CGI script allows remote attackers to execute arbitrary commands via shell metacharacters in the recipient email address. |
| CAN-1999-1081 | Vulnerability in files.pl script in Novell WebServer Examples Toolkit 2 allows remote attackers to read arbitrary files. |
| CAN-1999-1078 | WS_FTP Pro 6.0 uses weak encryption for passwords in its initialization files, which allows remote attackers to easily decrypt the passwords and gain privileges. |
| CAN-1999-1072 | Excite for Web Servers (EWS) 1.1 allows local users to gain privileges by obtaining the encrypted password from the world-readable Architext.conf authentication file and replaying the encrypted password in an HTTP request to AT-generated.cgi or AT-admin.cgi. |
| CAN-1999-1070 | Buffer overflow in ping CGI program in Xylogics Annex terminal service allows remote attackers to cause a denial of service via a long query parameter. |

| | |
|---|---|
| CAN-1999-1069 | Directory traversal vulnerability in carbo.dll in iCat Carbo Server 3.0.0 allows remote attackers to read arbitrary files via a .. (dot dot) in the icatcommand parameter. |
| CAN-1999-1069 | Directory traversal vulnerability in carbo.dll in iCat Carbo Server 3.0.0 allows remote attackers to read arbitrary files via a .. (dot dot) in the icatcommand parameter. |
| CAN-1999-1067 | SGI MachineInfo CGI program, installed by default on some web servers, prints potentially sensitive system status information, which could be used by remote attackers for information gathering activities. |
| CAN-1999-1063 | CDomain whois_raw.cgi whois CGI script allows remote attackers to execute arbitrary commands via shell metacharacters in the fqdn parameter. |
| CAN-1999-1050 | Directory traversal vulnerability in Matt Wright FormHandler.cgi script allows remote attackers to read arbitrary files via (1) a .. (dot dot) in the reply_message_attach attachment parameter, or (2) by specifying the filename as a template. |
| CAN-1999-1006 | Groupwise web server GWWEB.EXE allows remote attackers to determine the real path of the web server via the HELP parameter. |
| CAN-1999-0913 | dfire.cgi script in Dragon-Fire IDS allows remote users to execute commands via shell metacharacters. |
| CAN-1999-0885 | Alibaba web server allows remote attackers to execute commands via a pipe character in a malformed URL. |
| CAN-1999-0509 | Perl, sh, csh, or other shell interpreters are installed in the cgi-bin directory on a WWW site, which allows remote attackers to execute arbitrary commands. |
| CAN-1999-0477 | The Expression Evaluator in the ColdFusion Application Server allows a remote attacker to upload files to the server via openfile.cfm, which does not restrict access to the server properly. |
| CAN-1999-0467 | The Webcom CGI Guestbook programs wguest.exe and rguest.exe allow a remote attacker to read arbitrary files using the "template" parameter. |
| CAN-1999-0253 | IIS 3.0 with the iis-fix hotfix installed allows remote intruders to read source code for ASP programs by using a %2e instead of a . (dot) in the URL. |
| CAN-1999-0238 | php.cgi allows attackers to read any file on the system. |
| CAN-1999-0229 | Denial of service in Windows NT IIS server using .... |

Appendix 2 displays the full source code for the Help and Supports Center's vulnerable Uplddrvinfo.htm resource. The specific code used by the exploit is highlighted here also.

**Appendix 2.** Full source code of the vulnerable Uplddrvinfo.htm resource.

```
<html>
<head>
<object id="pchealth" classid="CLSID:FC7D9E02-3F9E-11d3-93C0-
00C04F72DAF7"></object>
<meta http-equiv="Content-Type" content="text/html; CHARSET=windows-1252" />
<meta http-equiv="PICS-Label" content='(PICS-1.1 "http://www.rsac.org/ratingsv01.html" l
comment "RSACi North America Server" by "inet@microsoft.com" r (n 0 s 0 v 0 l 0))' />
<meta http-equiv="MSThemeCompatible" content="Yes" />
<title id="eDFSTitle">Get Help with Your Hardware Device</title>
<style type="text/css">
body
{
font:messagebox;
padding:10px 10px 5px 10px;
}

h3
{
font-family:tahoma;
font-size:130%;
color:black;
}

.clsContent, .clsSubmit
{
padding-bottom: 5px;
}

.clsContent a
{
color:#114488;
text-decoration:underline;
}

ul
{
margin-top:10px;
list-style:circle outside;
}

a
{
color:blue;
text-decoration:underline;
}

div#eConfirm
{
font-size:100%;
}
```

```
button
{
font-size:8pt;
padding-left:10px;
padding-right:10px;
}

.clsHidden
{
display:none;
}

div#eShowPriv
{
width:450px;
border:2px outset;
background:window;
}

.clsNavTable
{
position:absolute;
bottom:5px;
right:30px;
}

div#eResContainer
{
overflow-y:auto;
display:none;
}

</style>
<script type="text/jscript" defer>

/*** Global vars ***/

var g_bDebugMode = false;
if( false == g_bDebugMode )
{
window.onerror = new Function( "return true" );
}
var g_sThisHREF = location.href;
var g_bSend = false;
var g_bConnect = false;
var g_bFromConnectUI = false;
var g_bDelayUpload = false;
var g_bFromSavedFav = false;
var g_sDrvIDs = "";
var g_bUploadSuccess = false;
var g_bFav_UploadSuccess = false;
var g_bSaveFavorite = false;
var g_bValidXMLFile = ValidateXMLFile();
var g_oWShell = new ActiveXObject( "WScript.Shell" );
var g_oXMLDlg, g_oPrivDlg, g_oDrag = null;
var g_sTestURL = "http://feedback.windows.com/TestDFS.asp";
var g_bNonRASConnectoid = false;
var g_bRanRasPhone = false;
var g_bUploadOnce = false;

//For localization
var L_DataFileError_Message = "The upload data file does not exist or is malformed.";
var L_FavoriteAdded_Message = "This page will be added to your Help and Support
Favorites list.";
var L_FavoriteDupe_Message  = "You already have a Favorite link to this topic.";
var L_DialogHeading_Text = "Missing Device Driver XML Profile";
var L_DialogClose_Text = "Close";
var L_Connecting_Text = "Connecting...";
var L_LaunchNCW_Message = "Windows cannot find an existing Internet connection on your
computer. To create an Internet connection now, click Yes.\n\nIf you have an existing
connection that Windows might not be detecting, click No, and then manually start the
connection. Important: Do not close Help and Support Center. After establishing the Internet
connection, come back to the connection page, and then click Next again.\n\nIf you know you
have an existing connection that is already open, click No, and then click Next again.";

/*** End Global vars ***/

/*** Error Handling Code ***/

function ASSERT( sObj )
{
try
{
eval( sObj );
}
catch( e )
{
if( true == g_bDebugMode )
{
alert( "Error running: " + sObj + " -> " + e.description + "\nNumber -> " + e.number );
}
}
}

/*** End Error Handling Code ***/

/*** Connection Code ***/

function oConnCheck_onCheckDone( cn, lStatus, hr, url, vCtx )
{
if( 4 == lStatus )
{
eConnectingH3.innerText = L_Connecting_Text;
ASSERT( "GetDriverName( false )" );
}
else
{
if( true == g_bFromConnectUI )
{
ASSERT( "Cancel( true )" );
}
else
{
eConnectingH3.style.display = "none";
h3Default.style.display = "block";
ASSERT( "ShowConnectionUI()" );
}
}
}

function RASInternetConnect()
{
var bHasConnectoid = pchealth.Connectivity.HasConnectoid;
var bAutoDialEnabled = pchealth.Connectivity.AutoDialEnabled;

if( true == bHasConnectoid )
{
if( true == bAutoDialEnabled || true == g_bRanRasPhone )
{
try
{
pchealth.Connectivity.AutoDial( false );
setTimeout( "TestConn( g_sTestURL )", 50 );
}
catch( e )
{
eNoConnect.style.display = "block";
eConnecting.style.display = "none";
h3Default.style.display = "block";
eConnectionUI.style.display = "block";
eBtnNav2.style.display = "block";
}
}
else
{
h3Default.style.display = "block";
eConnectionUI.style.display = "block";
eBtnNav2.style.display = "block";
eConnecting.style.display = "none";
OpenConnManager();
}
}
else
{
eConnecting.style.display = "none";
h3Default.style.display = "block";
eConnectionUI.style.display = "block";
eBtnNav2.style.display = "block";
ShowMessage( L_LaunchNCW_Message );
}
}

function InitConn( bContinue )
{
var bLAN = pchealth.Connectivity.IsALan;
var sIPs = GetIPAdresses();

if( true == bContinue )
{
h3Default.style.display = "none";
eMain.style.display = "none";
```

```
eConnectionUI.style.display = "none";
eBtnNav2.style.display = "none";
tblResources.style.display = "none";
eNoConnect.style.display = "none";
eConnecting.style.display = "block";

if( ( true == bLAN || true == g_bNonRASConnectoid ) && ( "" != sIPs && "0.0.0.0" != sIPs ) )
{
setTimeout( "TestConn( g_sTestURL )", 50 );
}
else
{
if( "" != sIPs && "0.0.0.0" != sIPs )
{
setTimeout( "TestConn( g_sTestURL )", 50 );
}
else
{
if( true == g_bFromConnectUI )
{
ASSERT( "RASInternetConnect()" );
}
else
{
eConnecting.style.display = "none";
ASSERT( "ShowConnectionUI()" );
}
}
}
}
else
{
eConnecting.style.display = "none";
tdThanks.style.display = "block";
ShowResults();
}
}

function TestConn( sConnTestURL )
{
var oPCH_ConnCheck = pchealth.Connectivity.CreateObject_ConnectionCheck();
oPCH_ConnCheck.onCheckDone = oConnCheck_onCheckDone;
oPCH_ConnCheck.StartUrlCheck( sConnTestURL, 0 );
}

/*** End Connection Code ***/

/*** Upload Process ***/

function ValidateXMLFile()
{
var sURL = unescape( g_sThisHREF.substring( g_sThisHREF.indexOf( "?" ) + 1 ) );
sURL = sURL.replace( "file://", "" ).replace( /\&.*/, "" );

try
{
var oXML = new ActiveXObject( "Msxml2.DOMDocument" );
```

```
oXML.async = false;
oXML.validateOnParse = false;
oXML.resolveExternals = false;
oXML.load( sURL );
if( "" == oXML.xml )
{
return false;
}
else
{
return true;
}
}
catch( e )
{
return false;
}
}

function SetUploadState( bStatus )
{
eNext.disabled = false;
if( true == bStatus )
{
g_bSend = true;
g_bConnect = true;
}
else
{
g_bSend = false;
g_bConnect = false;
}
}

function SetConnState( bConnStatus )
{
eConnNext.disabled = false;
g_bFromConnectUI = true;
if( true == bConnStatus )
{
g_bConnect = true;
g_bDelayUpload = false;
}
else
{
g_bConnect = false;
g_bDelayUpload = true;
}
}

function ShowConnectionUI()
{
if( true == g_bSend )
{
eAccess.scrollTop = 0;
eMain.style.display = "none";
h3Default.style.display = "block";
```

```
eConnectionUI.style.display = "block";
eBtnNav1.style.display = "none";
eBtnNav2.style.display = "block";
if( true == g_bFromSavedFav )
{
eChkSaveQuery.style.display = "none";
eLblSaveQuery.style.display = "none";
eConnBack.style.display = "none";
}
ASSERT( "MatchSize()" );
}
else
{
ASSERT( 'Cancel()' );
}
}

function Send()
{
if( true == g_bSend )
{
if( false == g_bFromSavedFav && false == g_bUploadOnce )
{
if( true == g_bValidXMLFile )
{
window.setTimeout( 'ASSERT( "UploadData()" )', 50 );
}
else
{
ASSERT( "ShowDataFileError()" );
return;
}
}
InitConn( g_bConnect );
}
else
{
Cancel();
}

if( true == g_bDelayUpload )
{
h3Default.style.display = "block";
tblCancel.style.display = "block";
tdThanks.style.display = "none";
tdDelayUpload.style.display = "block";
eTDContactMan.style.display = "block";
eNoConnect.style.display = "none";
tdNoRecord.style.display = "none";
}
}

function Cancel( bConnFailed )
{
eMain.style.display = "none";
eConnectionUI.style.display = "none";
eBtnNav2.style.display = "none";
```

```
eBtnNav3.style.display = "block";
tblResources.style.display = "block";
eAccess.scrollTop = 0;
if( false == g_bValidXMLFile )
{
Content_LI3.style.display = "none";
}

if( !bConnFailed )
{
Content_LI3.style.display = "none";
tblCancel.style.display = "block";
tdThanks.style.display = "none"
tdDelayUpload.style.display = "none";
eTDContactMan.style.display = "block";
tdNoRecord.style.display = "block";
}
else
{
eConnecting.style.display = "none";
tblCancel.style.display = "block";
tdThanks.style.display = "none"
tdDelayUpload.style.display = "none";
eTDContactMan.style.display = "block";
eNoConnect.style.display = "block";
}

if( true == g_bFromSavedFav )
{
eFinish.style.display = "none";
}

ResizeButtons( [eBack,eNext_D,eFinish] );
}

function Back()
{
eAccess.scrollTop = 0;
eResContainer.style.display = "none";
h3Searching.style.display = "none";
h3Default.style.display = "block";
tblCancel.style.display = "none";
tblResources.style.display = "none";
eNoConnect.style.display = "none";

if( true == g_bFromConnectUI )
{
eConnectionUI.style.display = "block";
eBtnNav1.style.display = "none";
eBtnNav3.style.display = "none";
eBtnNav2.style.display = "block";
eConnecting.style.display = "none";
ResizeButtons( [eConnBack,eConnNext,eConnCancel] );
}
else
{
eMain.style.display = "block";
tdNoRecord.style.display = "none";
eConnectionUI.style.display = "none";
eBtnNav3.style.display = "none";
eBtnNav2.style.display = "none";
eBtnNav1.style.display = "block";
ResizeButtons( [eNext,eCancel] );
}
}

function UploadData()
{
var sXMLFile = unescape( g_sThisHREF.substring( g_sThisHREF.indexOf( "?" ) + 1 ) );
sXMLFile = sXMLFile.replace( "file://", "" ).replace( /\&.*/, "" );
var vtExpDate = SetExpirationDate( 20 );
var oMPC_UL = new ActiveXObject( "UploadManager.MPCUpload" );
var oJob = oMPC_UL.CreateJob();
oJob.Sig = "";
oJob.Server = "http://feedback.windows.com/scripts/uploadserver.dll";
oJob.ProviderID = "NonEsc";
oJob.Mode = 1;
oJob.PersistToDisk = true;
oJob.History = 0;
oJob.ExpirationTime = vtExpDate;
oJob.GetDataFromFile( sXMLFile );
oJob.ActivateAsync();
g_bUploadOnce = true;
}

function Resize()
{
document.all.eResWindow.style.width = document.body.clientWidth - 50;
}

function CheckDriver( sDriver )
{
var iLCID = pchealth.UserSettings.CurrentSKU.Language;
var sURL = "http://go.microsoft.com/fwlink/?linkid=433&eDrvID=" + escape( sDriver ) +
"&lcid=" + iLCID;
document.frames["eResWindow"].location.replace( sURL );
h3Searching.style.display = "block";
ASSERT( "ShowResults()" );
}

function ShowResults()
{
if( true == g_bConnect )
{
eResContainer.style.display = "block";
document.all.eResWindow.style.width = document.body.clientWidth - 55;
}
eAccess.scrollTop = 0;
eConnecting.style.display = "none";
tblResources.style.display = "block";
tdThanks.style.display = "none";
Content_LI3.style.display = "block";
eConnectionUI.style.display = "none";
eBtnNav2.style.display = "none";
```

```
if( false == g_bFromSavedFav && false == g_bFromConnectUI )
{
eBtnNav3.style.display = "block";
}
ResizeButtons( [eBack,eFinish] );
if( true == g_bFromSavedFav )
{
tdNoRecord.style.display = "none";
Content_LI3.style.display = "none";
eFinish.style.display = "none";
}

setTimeout( "ASSERT( 'Resize_ResultsContainer()' );", 100 );
}

function Resize_ResultsContainer()
{
if( eResContainer.offsetHeight > 250 )
{
eResContainer.style.height = '250px';
}
}

function GetDriverName( bShow, bForSavedFav )
{
var docElem, aDriverID, iDriverIDLen;
var sXMLFile = unescape( g_sThisHREF.substring( g_sThisHREF.indexOf( "?" ) + 1 ) );
sXMLFile = sXMLFile.replace( "file://", "" ).replace( /\&.*/, "" );

if( true == g_bValidXMLFile )
{
var streamXML = pchealth.OpenFileAsStream( sXMLFile );
var oXML = new ActiveXObject( "Msxml2.DOMDocument" );
oXML.async = false;
oXML.validateOnParse = false;
oXML.resolveExternals = false;
oXML.load( streamXML );
docElem = oXML.documentElement;
aDriverID = docElem.selectNodes( "//hwid" );
iDriverIDLen = aDriverID.length
if( !bShow )
{
g_sDrvIDs = escape( aDriverID.item(0).text ) + "|";
for( var i=1; i<iDriverIDLen; i++ )
{
g_sDrvIDs += escape( aDriverID.item(i).text ) + "|";
}

if( !bForSavedFav )
{
eMain.style.display = "none";
h3Default.style.display = "none";
CheckDriver( g_sDrvIDs );
}
}
}
else
```

```
{
if( true == g_bFav_UploadSuccess )
{
var sDrvIDs = g_sThisHREF.substring( g_sThisHREF.indexOf( "&eDrvIDs=" ) ).replace(
"&eDrvIDs=", "" );
ASSERT( "CheckDriver('" + sDrvIDs + "')" );
}
else
{
eConnecting.style.display = "none";
ASSERT( "ShowDataFileError()" );
}
}
}

function ShowPopUp( sType )
{
if( "xml" == sType )
{
ASSERT( "ShowUploadXML()" );
}
else
{
ASSERT( "OpenPrivWindow()" );
}
}

function OpenPrivWindow()
{
if( null == g_oPrivDlg )
{
g_oPrivDlg = window.showModelessDialog( "hcp://system/dfs/Privacy.htm", window,
"status:no;help:no;resizable:no;dialogWidth:500px;dialogHeight:450px" );
}
}

function DisplayXML()
{
var sURL = unescape( g_sThisHREF.substring( g_sThisHREF.indexOf( "?" ) + 1 ) );
if( -1 != sURL.indexOf( "file://" ) )
{
sURL = sURL.replace( "file://", "" );
}
sURL = sURL.replace( /\&.*/, "" );
var oXML = new ActiveXObject( "Msxml2.DOMDocument" );
oXML.async = false;
oXML.validateOnParse = false;
oXML.resolveExternals = false;
oXML.load( sURL );
var oStyle = new ActiveXObject( "Msxml2.DOMDocument" );
oStyle.async = false;
oStyle.load( "xmldisplay.xsl" );
var sHTML = oXML.transformNode( oStyle );
sHTML = sHTML.replace( "</h5>", L_DialogHeading_Text + "</h5>" );
sHTML = sHTML.replace( "</button>", L_DialogClose_Text + "</button>" );
g_oXMLDlg.document.write( sHTML );
}
```

```
function ShowUploadXML()
{
if( null == g_oXMLDlg )
{
if( true == g_bValidXMLFile )
{
var sURL = "hcp://system/dfs/XMLDialog.htm";
g_oXMLDlg = window.showModelessDialog( sURL, window,
"status:no;help:no;dialogWidth:600px;dialogHeight:450px");
}
else
{
ASSERT( "ShowDataFileError()" );
}
}
}

/*** End Upload Process ***/

/*** Dialog Code ***/

function ShowMessage( sMessage )
{
var sButtonType = "YESNOCANCEL";
var sRetVal = pchealth.MessageBox( sMessage, sButtonType );
if( "YES" == sRetVal )
{
eNoConnect.style.display = "none";
OpenConnWizard();
}
else if( "NO" == sRetVal )
{
g_bNonRASConnectoid = true;
}
}

/*** End Dialog Code ***/
/*** Add To HSS Favorites ***/

function AddFavorite( bFromResourceLink )
{
var oFav = pchealth.UserSettings.Favorites;
var sThisURL = g_sThisHREF;
if( true == g_bSaveFavorite || true == oFav.IsDuplicate( sThisURL ) )
{
pchealth.MessageBox( L_FavoriteDupe_Message, "OK" );
return;
}
else
{
g_bSaveFavorite = true;
}
ASSERT( "GetDriverName( false, true )" );

if( true == bFromResourceLink )
{
pchealth.MessageBox( L_FavoriteAdded_Message, "OK" );
```

```
}
}

/*** End Favs ***/
/*** Misc ***/

function OpenConnWizard()
{
try
{
var oShell = new ActiveXObject( "WScript.Shell" );
var sShellCmd_NCW = "rundll32 netshell.dll,StartNCW 0";
oShell.Run( sShellCmd_NCW );
}
catch( e )
{

}
}

function OpenConnManager()
{
try
{
g_bRanRasPhone = true;
g_oWShell.Run( "rasphone.exe" );
}
catch( e )
{
g_bRanRasPhone = false;
}
}

function LaunchCPL_HW()
{
g_oWShell.Run( "control hdwwiz.cpl" );
if( false == g_bFromSavedFav )
{
setTimeout( "pchealth.close();", 500 );
}
}

function HSS_NavigateToTS()
{
var oFSO = new ActiveXObject( "Scripting.FileSystemObject" );
var sWinDir = oFSO.GetSpecialFolder(0);
var sHelpTopicURL = "";
if( true == pchealth.UserSettings.IsDesktopVersion )
{
sHelpTopicURL =
"hcp://services/subsite?node=TopLevelBucket_4/Fixing_a_problem&select=TopLevelBucket_
4/Fixing_a_problem/Home_Networking_and_network_problems";
if( false == g_bFromSavedFav )
{
g_oWShell.Run( sHelpTopicURL );
}
else
```

```
{
location.href = sHelpTopicURL;
}
}
else
{
sHelpTopicURL =
"hcp://services/subsite?node=Connections/Network_Connections&topic=ms-its:" + sWinDir +
"\\Help\\netcfg.chm::/trouble_all.htm&select=Connections/Network_Connections/Troubleshoot
ing/Troubleshooting_network_and_dial-up_connections";
if( false == g_bFromSavedFav )
{
g_oWShell.Run( sHelpTopicURL );
}
else
{
location.href = sHelpTopicURL;
}
}
}

function ResizeButtons( arr_oButtons )
{
var aBtnsLen = arr_oButtons.length;
var aWidths = new Array();
var iMax = 0;
for( var i=0; i<aBtnsLen; i++ )
{
aWidths[i] = arr_oButtons[i].offsetWidth;
}

for( var i=0; i<aBtnsLen; i++ )
{
if( aWidths[i] > iMax )
{
iMax = aWidths[i];
}
}
var iMaxBtnWidth = iMax;

for( var i=0; i<aBtnsLen; i++ )
{
arr_oButtons[i].style.width = iMaxBtnWidth;
}

}

function MatchSize()
{
if( "block" == eMain.currentStyle.display )
{
if( eNo.checked == false && eYes.checked == false )
{
eNext.disabled = true;
}
ResizeButtons( [eNext,eCancel] );
}
```

```
else if( "block" == eConnectionUI.currentStyle.display )
{
if( eNoConn.checked == false && eYesConn.checked == false )
{
eConnNext.disabled = true;
}
ResizeButtons( [eConnBack,eConnNext,eConnCancel] );
}
}

function NavigateTo_FullView( sURL )
{
ASSERT( 'g_oWShell.Run( "' + sURL + '" )' );
}

function ShowDataFileError()
{
eMain.style.display = "none";
eConnectionUI.style.display = "none";
eBtnNav1.style.display = "none";
eBtnNav2.style.display = "none";
eBtnNav3.style.display = "block";
eBack.disabled = true;
h3Default.style.display = "block";
eDataFileError.style.display = "block";
}

function SetExpirationDate( iDaysFromNow )
{
var oDate = new Date();
var vtDate = oDate.getVarDate();
var iMin_Msec = 1000 * 60;
var iHr_Msec = iMin_Msec * 60;
var iDy_Msec = iHr_Msec * 24;
var iCurrent_Msec = Date.parse( vtDate );
var iExp_Msec = iCurrent_Msec + ( iDaysFromNow * iDy_Msec );
var oExpDate = new Date( iExp_Msec );
var vtExpDate = oExpDate.getVarDate();
return vtExpDate;
}

function GetIPAdresses()
{
var sIPs = "";
try
{
sIPs = pchealth.Connectivity.IPAddresses;
}
catch( e )
{
try
{
var oSetting = new ActiveXObject( "rcbdyctl.Setting" );
sIPs = oSetting.GetIPAddress;
}
catch( e )
{
}
}
finally
{
return sIPs;
}
}

function DetectReadyState()
{
try
{
var eResWinDoc = document.frames["eResWindow"].document;
if( "complete" == eResWinDoc.readyState )
{
h3Searching.style.display = "none";
}
}
catch( e )
{
h3Searching.style.display = "none";
}
}

/*** End Misc ***/
/*** Event Handlers ***/

function window::onresize()
{
if( "block" == eResContainer.currentStyle.display )
{
ASSERT( "Resize()" );
}
}

function window::onload()
{
if( -1 != g_sThisHREF.indexOf( "saved=true" ) )
{
g_bSend = true;
g_bConnect = true;
g_bFromSavedFav = true;
if( -1 != g_sThisHREF.indexOf( "eDrvIDs=" ) )
{
g_bFav_UploadSuccess = true;
}
ASSERT( "g_bFromConnectUI=false;Send()" );
}
else
{
eNo.checked = false;
eYes.checked = false;
eBtnNav1.style.display = "block";
ASSERT( "MatchSize()" );
}

if( false == g_bValidXMLFile )
```

```
{
eDataLnk.disabled = true;
eDataLnk.style.cursor = "default";
eDataLnk.onclick = new Function( "return false" );
}
}

function window::onunload()
{
var oFav = pchealth.UserSettings.Favorites;
var sThisURL = g_sThisHREF;
var oDate = new Date();
var sDate = oDate.toLocaleDateString();
var sFavTitle = document.title + " -- " + sDate;
var sThisURL_New = sThisURL + "&saved=true";
var sThisURL_Exists = sThisURL.replace( /file:\/\/.*/i,"saved=true" );
sThisURL_Exists = sThisURL_Exists + "&eDrvIDs=" + g_sDrvIDs;
var sFile = unescape( sThisURL.substring( sThisURL.indexOf( "?" ) + 1 ) );
sFile = sFile.replace( "file://", "" ).replace( /\&.*/, "" );

-------------------------------------------- Code used by exploit --------------------------------------------

var oFSO = new ActiveXObject( "Scripting.FileSystemObject" );
try
{
oFSO.DeleteFile( sFile );
}
------------------------------------------------------------------------------------------------------------------------

catch( e ){ }
if( true == g_bSend )
{
if( true == g_bSaveFavorite )
{
for(var oEnumFavs = new Enumerator( oFav ); !oEnumFavs.atEnd(); oEnumFavs.moveNext()
)
{
var oThisFav = oEnumFavs.item();
if( sThisURL == oThisFav.URL || sThisURL_New == oThisFav.URL || sThisURL_Exists ==
oThisFav.URL )
{
oFav.Delete( oThisFav );
break;
}
}
oFav.Add( sThisURL_Exists, sFavTitle );
}
}
}

/*** End Event Handlers ***/
</script>
</head>
<body scroll="no" bgcolor="#ffffff" text="#000000" topmargin="0" leftmargin="10"
oncontextmenu="return false;">
<div id="eAccess" style="height:90%;overflow:auto;">
<h3 id="h3Default">Get Help with Your Hardware Device</h3>
```

```
<h3 id="h3Searching" class="clsHidden">Searching for device information...<br /><img
id="eConnImg" src="hcp://system/images/progbar.gif" border="no" /></h3>
<div id="eConnecting" class="clsHidden"
style="position:absolute;top:expression((document.body.offsetHeight -
eConnectingH3.offsetHeight)/2);left:expression((document.body.offsetWidth -
eConnectingH3.offsetWidth)/2);">
<h3 id="eConnectingH3">Checking your Internet connection...<br /><img id="eConnImg"
src="hcp://system/images/progbar.gif" border="no" /></h3>
</div>
<div id="eResContainer">
<iframe id="eResWindow" src="" frameborder="0" width="1" height="1" application="yes"
onreadystatechange="ASSERT( 'DetectReadyState()' );"></iframe>
</div>
<div id="eNoConnect" class="clsHidden">
<br />
<b style="color:red;">Internet connection could not be established</b>
<br />
<br />
There was a problem in connecting to the Internet. Click the <b>Back</b> button, and try to
connect again.
<br />
<br />
<a href="javascript:ASSERT( 'HSS_NavigateToTS()' );" id="eNoConnectLnk">Get
troubleshooting tips on connecting</a>
<br />
<br />
</div>
<div id="eDataFileError" class="clsHidden">
<b style="color:red;">Unable to continue the hardware information upload process</b>
<br />
<br />
A problem has occurred with your hardware information file and we are unable to process it.
This process cannot continue without this information. If you wish to complete this process,
you will need to try installing the device again by accessing the Add New Hardware wizard.
<br />
<br />
<a href="javascript:ASSERT( 'LaunchCPL_HW()' );">Start the Add New Hardware
wizard.</a>
</br>
</div>
<div id="tblCancel" class="clsHidden">
<div id="tdNoRecord" class="clsHidden">
You chose not to allow Microsoft to record your hardware information.
<br />
<br />
</div>
<div id="tdDelayUpload" class="clsContent">
Your hardware information will be sent automatically the next time you connect to the Internet.
<br />
<br />
</div>
<div id="eTDContactMan" class="clsContent">
You may want to contact the manufacturer and ask them to provide a driver for this hardware
device.
<br />
<br />
</div>
```

```
</div>
<div id="tblResources" class="clsHidden">
<div id="tdThanks" class="clsContent">
Thank you for submitting your missing driver information to Microsoft.
</div>

<div id="tdResources" class="clsContent">
You may find the following resources helpful:
</div>
<div class="clsContent">
<ul id="eUL1">
<li id="Content_LI1" style="padding-bottom:10px;">
<a href="javascript:NavigateTo_FullView( 'hcp://system/compatctr/CompatOffline.htm'
);">Compatibility Center</a> - Find out
which hardware and software is compatible with Windows XP.
</li>

<li id="Content_LI2" style="padding-bottom:10px;">
<a href="javascript:NavigateTo_FullView( 'hcp://system/sysinfo/sysinfomain.htm' );">My
Computer Information</a> - Read
information about hardware and software that your computer currently has installed.
</li>
<li id="Content_LI3">
<a href="javascript:void(0);" onclick="ASSERT( 'AddFavorite( true )' );" id="lnkAddFav">Save
this page to your Help Center Favorites list</a> -
You can submit your hardware information again and get manufacturer feedback.
</li>
</ul>
<br />
</div>
</div>
<div id="eMain">
<div id="eMainContent">
<div id="eTDMainContent" class="clsContent">
The wizard was unable to find the necessary software for your new hardware.
<br />
<br />
Help and Support Center can now record your hardware profile to assist with future support.
This is done automatically by reading the identification number that the manufacturer
has encoded in your hardware. If you want, you can see the contents of <a id="eDataLnk"
href="javascript:ASSERT( 'ShowPopUp( \'xml\' )' );">the file that contains this information</a>.
This information will remain confidential--see Microsoft's <a id="ePrivLnk"
href="javascript:ASSERT( 'ShowPopUp()' );">privacy policy</a>. Using this
information, Microsoft can query for any available details on the hardware manufacturer's
Web site.
<br />
<br />
<div id="eConfirm" cellpadding="0" cellspacing="0" style="margin:10px 20px 10px
20px;vertical-align:abstop;">
<input type="radio" name="eRad" id="eYes" onclick="ASSERT( 'SetUploadState( true )'
);"><label id="eLblYes" for="eYes">Yes, record my hardware profile.</label>
<br />
<input type="radio" name="eRad" id="eNo" onclick="ASSERT( 'SetUploadState( false )'
);"><label id="eLblNo" for="eNo">No, I prefer not to participate.</label>
</div>
</div>
<div id="eMainList" class="clsContent">
```

```
This information will help us provide:
<ul id="eUL2">
<li id="Content_LI4">
Improved manufacturer support for new hardware installations in
future versions of Windows.
</li>
<li id="Content_LI5">
More supporting software for you to download from Microsoft Web
sites.
</li>
<li id="Content_LI6">
Links to other Internet sites that carry software or support information
for your new hardware.
</li>
</ul>
</div>
<div id="eCancelDesc" class="clsContent">
Click <b>Cancel</b> to end the wizard. You may want to contact the manufacturer
and ask them to provide a driver(software) for this hardware device.
</div>
</div>
</div>
<div id="eConnectionUI" class="clsHidden">
<div id="eConnConntent" class="clsContent">
To submit your hardware information now and receive the latest manufacturer information
back, please connect to the Internet. Or, if you want,
your hardware information can be submitted automatically the next time you connect to the
Internet, but you won't receive the latest information in return.
<br />
<br />
If you choose to save this page to your Help and Support Center Favorites list, then you can
access this page again to submit your hardware information
and get manufacturer feedback.
<br />
<br />
<div id="eConfirmConn" cellpadding="0" cellspacing="0" style="margin:0px 20px 0px 20px;">
<input type="radio" name="eRadConn" id="eYesConn" onclick="ASSERT( 'SetConnState(
true )' );"><label id="eLblYesConn" for="eYesConn">Attempt to connect to the Internet
now</label>
<br />
<input type="radio" name="eRadConn" id="eNoConn" onclick="ASSERT( 'SetConnState(
false )' );"><label id="eLblNoConn" for="eNoConn">Send my hardware information the next
time I connect</label>
</div>
<br />
<input type="checkbox" id="eChkSaveQuery" onclick="if(this.checked) ASSERT(
'AddFavorite( false )' );"><label id="eLblSaveQuery" for="eChkSaveQuery">Save this page to
my Help and Support Center Favorites list</label>
</div>
</div>
</div>
<div id="eBtnNav1" class="clsNavTable clsHidden" cellpadding="0" cellspacing="0"
align="right" style="margin:0px">
<button id="eBack_D" disabled>&lt;&nbsp;Back</button>
<button id="eNext" onclick="ASSERT( 'g_bFromConnectUI=false;Send()'
);">Next&nbsp;&gt;</button>
<button id="eCancel" onclick="ASSERT( 'g_bFromConnectUI=false;Cancel()'
);">Cancel</button>
</div>
<div id="eBtnNav2" class="clsNavTable clsHidden" cellpadding="0" cellspacing="0"
align="right" style="margin:0px">
<button id="eConnBack" onclick="g_bFromConnectUI=false;ASSERT( 'Back()'
);">&lt;&nbsp;Back</button>
<button id="eConnNext" onclick="ASSERT( 'g_bFromConnectUI=true;Send()'
);">Next&nbsp;&gt;</button>
<button id="eConnCancel" onclick="ASSERT( 'Cancel()' );">Cancel</button>
</div>
<div id="eBtnNav3" class="clsNavTable clsHidden" align="right" style="margin:5px 0px 0px
0px;">
<button id="eBack" onclick="ASSERT( 'Back()' );">&lt;&nbsp;Back</button>
<button id="eNext_D" disabled>Next&nbsp;&gt;</button>
<button id="eFinish" onclick="ASSERT( 'pchealth.close()' );">Finish</button>
</div>
</body>
</html>
```

# 4. References

1 IANA Ports Database. “Port Numbers.” 18 June 2003.
URL: http://www.iana.org/assignments/port-numbers

2 Internet Storm Center. “Top ten ports.” 17/6/2003.
URL: http://isc.incidents.org/top10.html

3 Internet Storm Center. “Records, Targets and Sources for Port 80.” 10/5/2003 - 18/6/2003.
URL: http://isc.incidents.org/port_details.html?port=80

4 Neohapsis. “Port Assignments.”
URL: http://www.neohapsis.com/neolabs/neo-ports/neo-ports.html

5 Adapted from Stevens, W. Richard. TCP/IP Illustrated, Volume 3. Reading: Addison Wesley Longman, Inc, 1996. 162.

6 Network Working Group. “Hypertext Transfer Protocol – HTTP/1.1” Request for Comments: 2616.
ftp://ftp.isi.edu/in-notes/rfc2616.txt

7 OWASP. “Overview.” URL: http://www.owasp.org/asac/

8 Microsoft. “About Asynchronous Pluggable Protocols.”
URL: http://msdn.microsoft.com/workshop/networking/pluggable/overview/overview.asp

9 Microsoft. “Flaw in Windows XP Help and Support Center Could Enable File Deletion.” 22 April 2003.
URL: http://support.microsoft.com/default.aspx?scid=KB;en-us;328940&

10 TechTV, Leo Laporte. “Demonstration of the Windows XP vulnerability”
URL: http://cgi.techtv.com/mediamodule?action=view&version=20020910095425&video_src=/thescreensavers/2002/ss020909c&width=320&height=240&vidsection=3200042&add_date=1031641200&start=&end=&duration=&bitrates=

11 CVE. “Common Vulnerabilities & Exposures List.'' 17 June 2003.
URL: http://cve.mitre.org/cve/