

Free Adware & Spyware Detection/Cleaning Tips and Techniques

Francois Paget
McAfee AVERT
Senior Virus Research Engineer

With major contribution from :

Jimmy Kuo
McAfee AVERT

Prabhat K. Singh¹

I - Introduction

Adware and Spyware have become the bane of computer users, probably even more than viruses. There's money behind their development. So, they will progress and become more complex.

Use of Anti-Spyware (and/or Anti-Adware) products is increasing as more large mainstream companies join the arena with offerings of their own. However, many products are deficient in their ability to completely clean a machine and rid it of all the unsolicited components.

In the first part of this document, we will define these two specific categories of software more precisely. Through some examples, we will show the methods they use to enter our machines.

In the main second part, we will offer a guide for detecting and removing such programs from your system. Wherever necessary, we will reference commonly available, free, and user-friendly tools that will help you with such tasks. And even if you don't need it, and hopefully you won't, we hope this explanation is educational and will help you to better understand the machine you use. The aim is to help you do your job. Let the machine be your tool, and not the tool of someone else on the Internet.

II - Preliminary definitions

From an etymological point of view, the terms Adware (*Advertising Software*) and Spyware (*Spying Software*) are English acronyms that refer to specific categories of software.

At McAfee, both belong to the *PUP* family. At a high level, Potentially Unwanted Programs are *"any piece of software which a reasonably security- or privacy-minded computer user may want to be informed of, and, in some cases, remove. Potentially Unwanted Programs are usually made by legitimate corporate entities for specific beneficial purposes (to whom they may be beneficial is debatable), but so alter the security state of the computer on which they are installed, or the privacy posture of the user using the computer, that most users will want to be aware of them."*

¹ All people interested in adware and spyware have to read the paper available in the December 2004 Virus Bulletin issue :

How 'dare' you call it spyware!

Prabhat K. Singh, Fraser Howard and Joe Telafici McAfee AVERT

This 2004 paper must be considered as the original gist of my own work.

II.1 Adware

An Adware program is a program designed to monitor an end user and present ads to that user. It observes the browsing habits of the Internaut (Internet User) in order to make offers adapted to his profile. This task is generally implemented after the user has granted initial consent. An Adware package is not a self-reproducing program; it is often installed in exchange for another service, such as the right to use another program without paying for it. The user confirms his agreement by means of a dialogue box by responding positively to a question that he does not always have time to read in full.

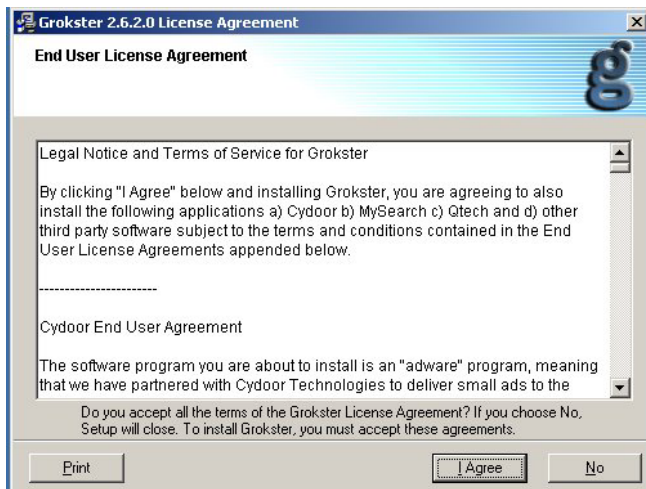


Figure 1: End User License Agreement for the free v2.6.2.0 version of *Grokster*

Certain *small* free utility programs (*freeware* or *shareware*) for which no authority is requested may resemble adware. Their authors maintain the associated copyright and receive financial compensation for the presentation of advertising banners that pop up periodically during use.

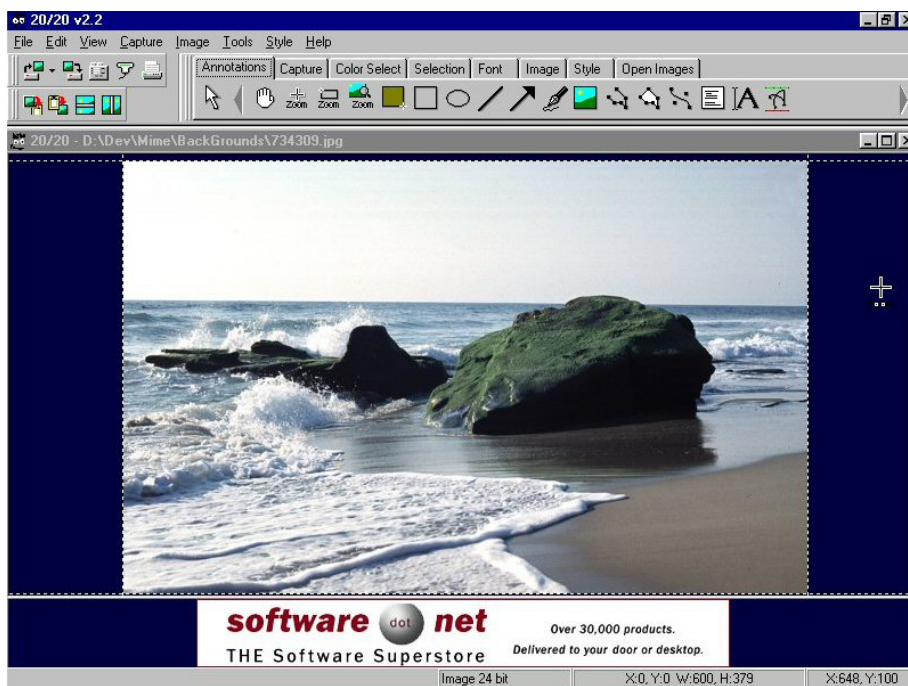


Figure 2: 20/20 (<http://www.hotfreeware.com/2020/2020.htm>) is a freeware package with an advertising banner. It displays images in a multitude of formats. It can also be used to create screenshots, add texts to images, modify images, etc.

It should also be pointed out that adware does not normally collect personal information [though unintentional, sometimes personal information is included into an URL and thus captured. We shall stand by the condition that Adware does not intentionally collect personal information] and is often in the form of binary files (.EXE or .DLL).

II.2 Spyware

Steve Gibson provided one of the first definitions of spyware on his site <http://grc.com/optout.htm>. It indicates that spyware is software that makes use of a user's Internet connection as a background task without their knowledge or explicit permission.



Figure 3: One of the first definitions of spyware, written/proposed by Steve Gibson

This definition is no longer sufficiently restrictive. In addition to all the elements described above, spyware now tends to include certain worms or Trojan horses.

McAfee currently defines *spyware*² as *a legitimate, non-replicating program designed to monitor the computer or browsing habits of a user. This might include monitoring keystrokes, tracking Internet history, uploading confidential information and the like*. It collects personal data and sends it to its creator or a third party via the Internet without first obtaining the clear, informed authorisation of the user.

Spyware is not to be confused with adware, BHO (Browser Help Objects, see next section), or redirection tools. Although they are sometimes unpleasant, the latter do not transmit personal data. Spyware may also be mistakenly confused with cookies and Web bugs, which are not programs but micro-image or data files, the access of which may be redirected or tracked and might sometimes violate privacy.

There are 2 types of spyware that differ according to their objective:

- Commerce: these programs are similar to adware packages. They do not simply redirect the targeted advertisements; they also transmit specific personal information in order to follow up their marketing approach by email, post or telephone. The collecting companies can then cash in on the gathered information and subsequent database.
- Information: this involves spying software that secretly records and/or transmits operations carried out on a computer. It might have legitimate uses such as for parental control but is then distorted from its original intent.

² McAfee – Virus Glossary of Terms
<http://www.nai.com/us/security/resources/glossary.htm>

Figure 4: Spyware tool to capture and send out information without the user's knowledge

II.3 BHO – Browser Helper Objects

A BHO is an additional program associated with Internet Explorer. It is integrated with the browser and allows third parties to personalise or add additional functionalities.

Its installation, voluntary or otherwise, is often based on ActiveX technology. BHOs generally provide an additional toolbar, added to the standard menu. The normal use of BHOs is to provide an additional functionality or added convenience.

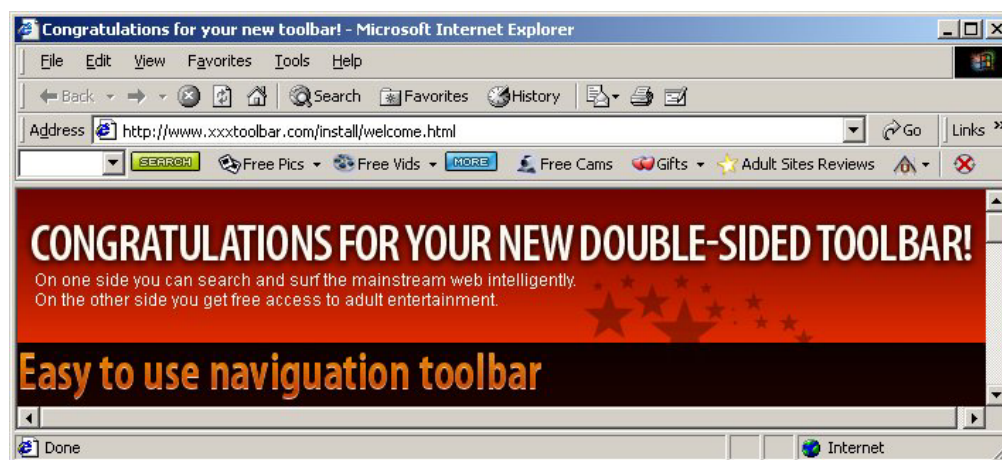


Figure 5: BHO

II.4 Hijackers

These are routines that alter the behaviour and/or navigation of the Internaut's browser. The alterations made are based on ActiveX controls or scripts languages such as JavaScript and are generally harmless but annoying if they are undesired. For example, a hijacker could add a number of new entries into the list of favourites, or change the start page itself.

III - Some dangers

Spyware carries obvious dangers. These range from theft of personal information and users' browsing habits including the lists of URLs visited and items searched on the web. Sometimes encoded into the URL is login information for a site, enabling someone else to access those sites using your identity. Some spyware also aid in eavesdropping and logging users' activities on the computer while others help advertising companies to display ad related pop-ups on the screen even when not actively surfing the Internet. Some of these programs further install backdoors that then allow hackers to attack the computers or make use of them to attack other computers.

Another concern with spyware programs is that most of these programs are very poorly written, especially with the consideration for portability. They might run on a specific version of Windows while crash on other versions. They may be resource hogs with memory leaks that consistently eat up system memory and CPU cycles. This leads to sluggish PC performance. The spyware programs may have a buggy or malicious uninstall feature (e.g. **HungryHands**) where upon uninstalling the spyware, it downloads and installs more unwanted programs or may just render the system unusable or unstable. Sometimes the adware or spyware programs alter the TCP/IP stack of the operating system (using LSPs bundled in **NewDotNet**) to redirect all the TCP traffic through the "sniffing" site. This would allow them to capture any pages you access, including any bank accesses or any other financial institution. Some of the financial institutions have already taken to banning any communication from such sites, to help protect its users from loss of private information. And further, if these LSP (Layered Service Provider) DLLs are improperly deleted, the PC's winsock stack would become damaged resulting in the need for a complete OS reinstall.

IV - Installation

Many software packages associated with free utility programs or online games will install additional programs. Users must accept these additional programs in order to get the "supposed" free versions. Producers of these additional programs pay the share/freeware author for this connection. It is thus inaccurate to refer to such software as "freeware" as the use of the software is actually exchanged for services. However, this is one of the most frequent methods by which adware and spyware are introduced. They can come along in these ways:

- Integrated with routines that form an integral part of the main application,
- Externalised as software packages in their own right to be installed alongside the main application,

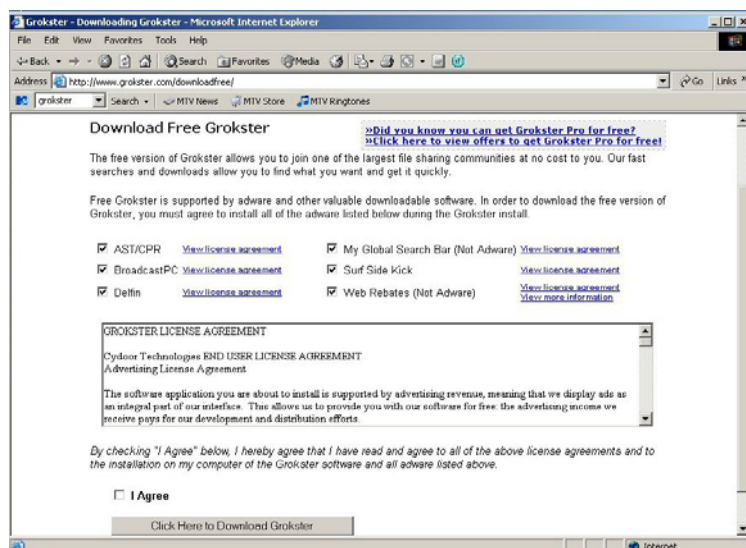


Figure 6 : Grokster installation example

The installation of particular software might also force you to access certain specific Internet sites. Or sometimes, access to certain sites might require that particular programs be already installed. These requirements are usually stated, but often in ambiguous terms and/or in a foreign language. Installations sometimes also take place silently, without prior agreement when dubious sites are visited (pornography, revisionism, sects, hacking, etc.). These aspects depend on the intentions and ethics of the site developer and the software designer. This characteristic, which is highly subjective, constitutes one of the differences between adware and spyware.

In summary, the most common methods used to infiltrate onto machines involve:

- Free software or demonstration versions downloaded from the Internet or distributed on media provided with certain reviews. Particular attention should be paid to:
 - o Utility programs for downloading and browsing assistance,
 - o Resource sharing software (between workstations),
 - o Screensavers,
 - o Games.
- Browsing sites that contain ActiveX controls that your browser does not block. The following sites should be avoided:
 - o Pornographic sites,
 - o Sites offering games,
 - o Sites linked to the underground world,
 - o Computer security sites offering hacking tools or other tools that bypass security.
- Unsolicited electronic mail (spam) or mail that is read in discussion forums. These might contain attachments asking you to click on them, may be in HTML format and contain links or elements that allow them to silently install.
- Online registration procedures for software licences or grants to access private browser zones.
- Resource sharing software (between workstations).
- Viruses and Trojan horses that might also install adware or spyware on the machine.

IV.1 Grokster example

Grokster, described by their own website, is an "advanced peer to peer file sharing program that enables users to share any digital file including images, audio, video, reports, documents, etc. Content developers and owners may now easily broadcast their files through the Grokster software".

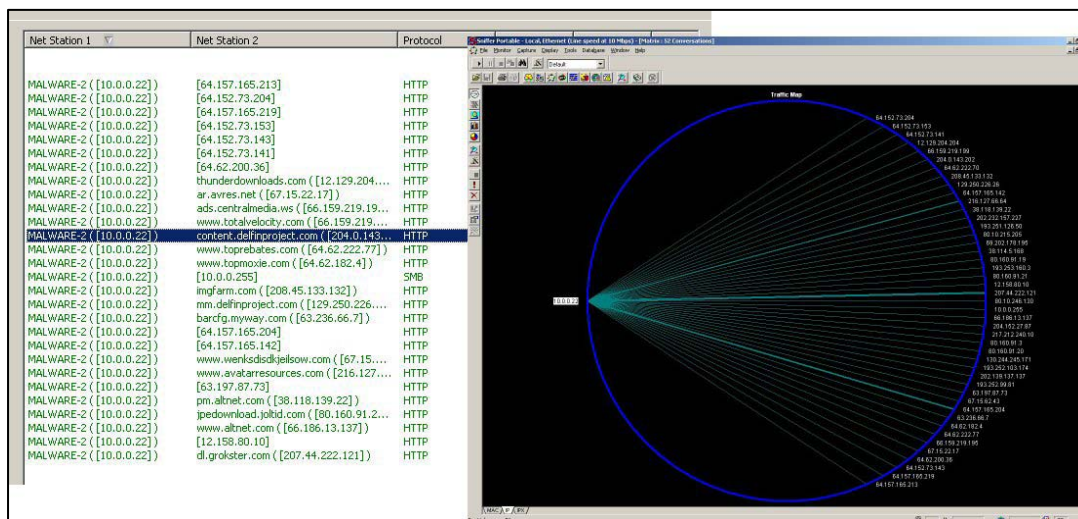


Figure 7 : A sniffer program recorded connections to more than 100 distinct sites.

The clean system used for this test is a minimal VMWARE W2000 temporary disk with:

- 1 icon on the desktop,
- 6 applications listed in the Add/Remove Programs facility,
- 30 processes in memory according to the Task Manager.

After installation, the machine dramatically slowed down. There were:

- 8 new icons,
- 15 new applications in the Add/Remove Programs facility,
- 10 new processes in memory according to the Task Manager.

There were also:

- 2 new Browser Helper Objects,
- 2 new favorites,
- 1177 keys added in the system registry and, 1579 values added or changed,
- 96 new directories in the folders tree and, 649 new files.

And finally:

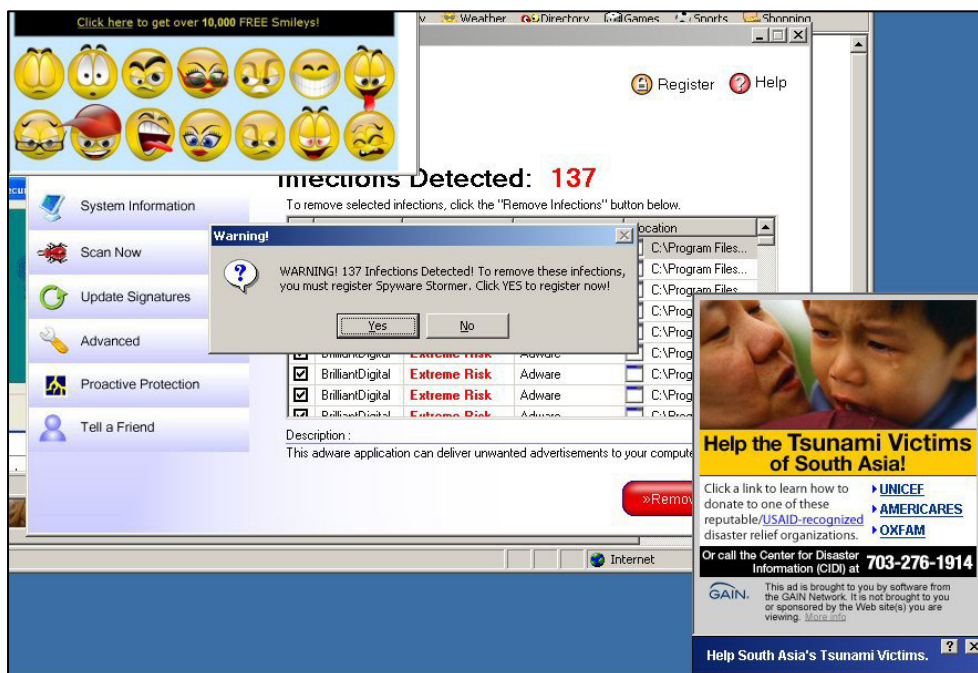


Figure 8: The installation resulted with the gift of a product declaring “137 Infections Detected.”

Consider, before the installation of *Grokster*, there was no adware or spyware nor even an antispysware product. Now, not only was there an antispysware product, it proclaimed that there were 137 infections on the machine. Before, none. Afterwards, “Buy this product to get rid of the 137 infections.” Should *Grokster* be responsible for perpetrating a scam?

It is now time to understand where these intruders are hidden.

V - Tools used to track PUPs

Many freeware tools are available to help track adware and spyware. For this paper I test and describe 6 of them.

V.1 InCtrl5 (<http://www.pcmag.com/article2/0,4149,25126,00.asp>)

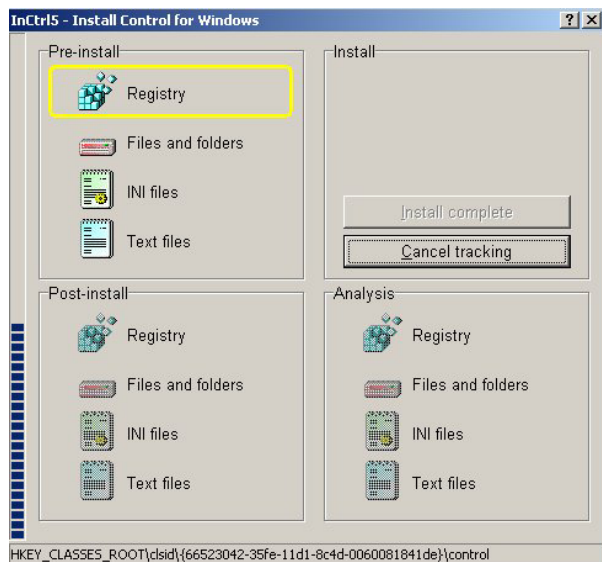


Figure 9: InCtrl5 takes snapshots of the system

InCtrl5 records snapshots of the system in order to determine any difference between snapshots. It runs on all Windows operating systems and is offered by PC Magazine. The first time, it takes a snapshot of the system. It is customizable to individual registry branches and .ini files. Afterwards, additional static snapshots can be created. And an analysis can be created to report the differences.

InCtrl5 offers a GUI for all aspects of its operation.

V.2 LspFix (<http://www.cexx.org/lspfix.htm>)

LspFix is a freeware utility for repairing corrupted Winsock stacks on the Windows system.

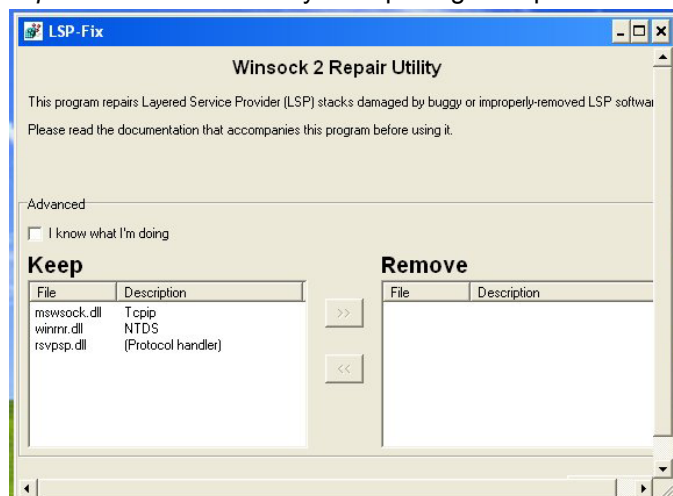


Figure 10: LspFix capture in a "clean" environment (Windows XP)

With a convenient interface, it attempts to correct Internet connection problems resulting from buggy or improperly removed *Layered Service Provider (LSP)* software³. It reads the list of LSP modules from the Windows registry and verifies that each module is present. If a module is missing, it is placed on the *Remove* list for removal. In case of adware infection, advanced users can override suggested removals and removes undesired entries. The result is that the remaining entries in the registry are renumbered to make them consecutive. The total module count is also updated.

V.3 ProcExp (<http://www.sysinternals.com/Utilities/ProcessExplorer.html>)

Process Explorer shows information about handles and DLL processes that are opened or loaded in the system. This tool comes in very handy when we need to find, suspend, or terminate an adware process. It is also helpful in searching for adware DLLs within all processes executing on a system.

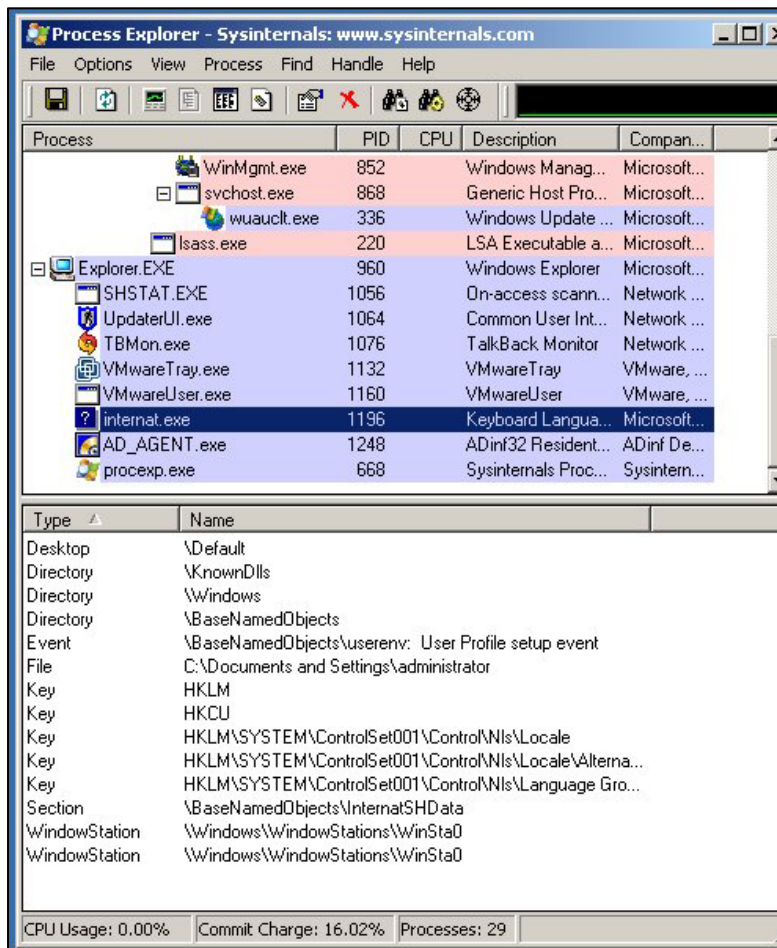


Figure 11: ProcExp capture in a "clean" environment

The display consists of two sub-windows. The top always shows a list of the currently active processes, including the names of their parent processes. The bottom window display, which you can choose to close, depends on the mode that the Process Explorer is in. If it is in handle mode,

³ A *Layered Service Provider* is a system driver linked deep into the networking services of Windows. It has access to every data entering and leaving the computer, as was as the ability to modify this data. A few such LSPs are necessary to allow Windows to connect to other computers, including the Internet. Spyware may also install itself as an LSP, thus having access to all the transmitted data. LSP are currently used by *CommonName*, *New.Net*, *NewtonKnows* and *WebHancer*.

you will see the handles that the processes in the upper window has opened. If *Process Explorer* is in DLL mode, you will see the DLLs and memory-mapped files that the process has loaded.

It needs *Microsoft Debugging Tools for Windows*
(<http://www.microsoft.com/whdc/devtools/debugging/default.mspx>)

V.4 RegMon (<http://www.sysinternals.com/ntw2k/source/regmon.shtml>)

Regmon is a Registry Monitoring Utility that displays the programs that are accessing the Registry and the sections they access. It will also report (in real time) the keys being accessed and the Registry Data being read or written.

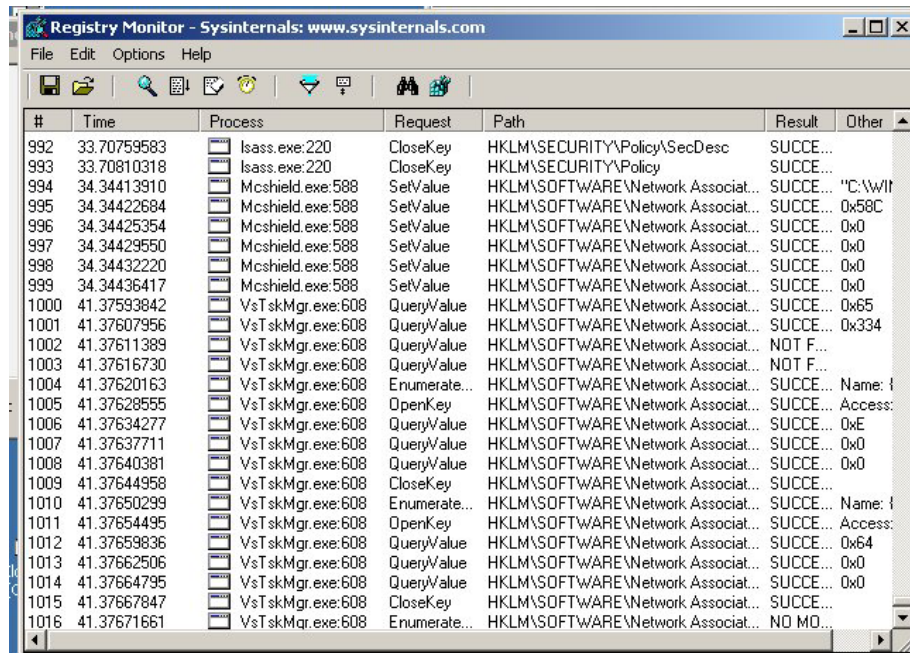


Figure 12: A Regmon screenshot

V.5 StartupRun (<http://www.nirsoft.net/utills/strun.html>)

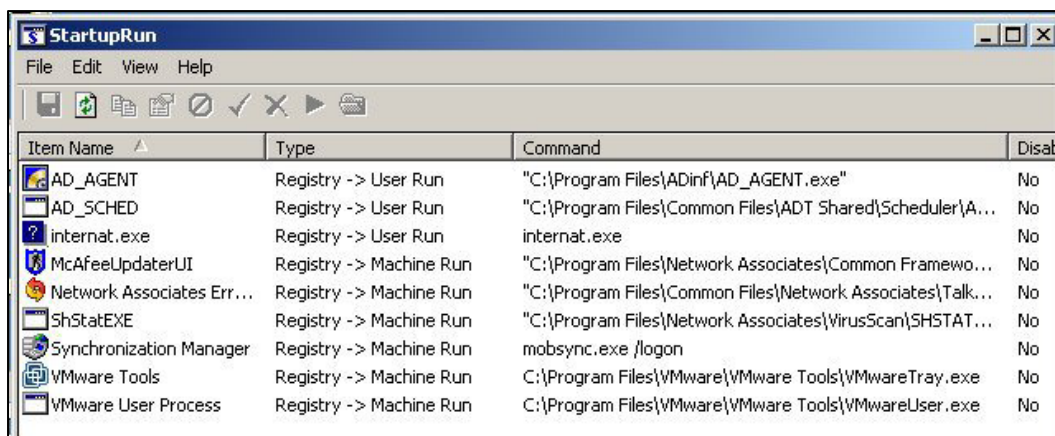


Figure 13: StartupRun capture in a "clean" environment

The *StartupRun* utility displays the list of all applications that are loaded automatically when Windows starts. For each application, additional information is displayed (Product Name, File

Version, Description, and Company Name) in order to easily identify the applications that are loaded at Windows startup.

If *StartupRun* identifies a spyware or adware program that runs at startup, it automatically paints it in pink. In addition, it supports editing, disabling, enabling and deleting the selected startup entries.

V.6 Sporder.exe

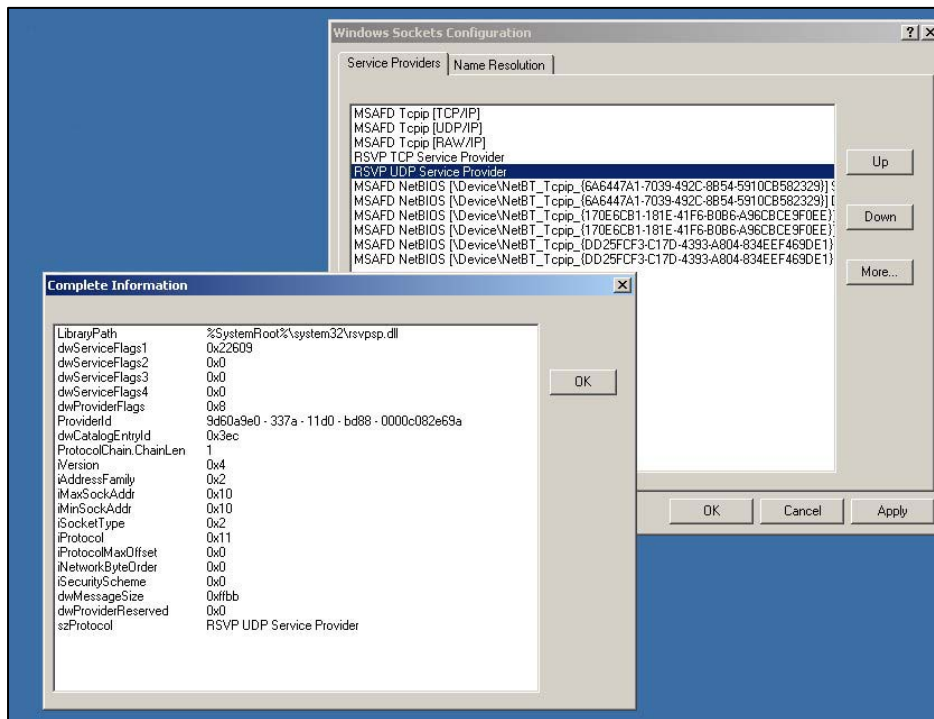


Figure 14: An *Sporder* screenshot

Sporder is a Microsoft utility available for download from MSDN. It gives information on all the *Layered Service Providers* installed on the system.

VI - Finding Intruders

It is not possible to provide a comprehensive list of all the possible indications of the presence of adware. However, these are likely symptoms:

- Popups are displayed on the user's PC,
- Browsing of the Internet slows down,
- Other URLs being accessed before the desired URL is accessed.

Signature based detection is the most accurate method of detection and identification of adware/spyware threats. Hence, professional scanners are the most suitable tools for detecting and identifying such elements. Most scanners provide information about the adware files installed and the registry entries made by them. This next section provides a "researcher approach" for the administrator, or simply the curious, who want to make their own investigations.

With this information, it will be also be possible to facilitate a manual cleaning process.

VI.1 Programs Loaded at Windows Boot

All the screenshots have been obtained by simply browsing the Internet without due care. (Appendix B shows how it was possible to catch all these intruders in only 10 minutes.)

To start, use the *StartupRun* utility to display the applications that are loaded when Windows starts. This shows us the *Browser Help Object (BHO)* DLLs installed on the PC.

After getting the BHO list, we need to check the *Properties* section of the suspicious DLL and EXE files. Looking first at the **Type**, unwanted programs can be noticed based on their **CompanyName** or **ProductName** values. Unfortunately, there is no readily available list of known values that would tell us whether it belongs to an Adware program or a "valid" program. So the user's judgment is the ultimate basis for any decisions made here. The names can be used to search the Internet to find out more about the company and/or the product. If the fields are empty, the program is clearly suspicious.

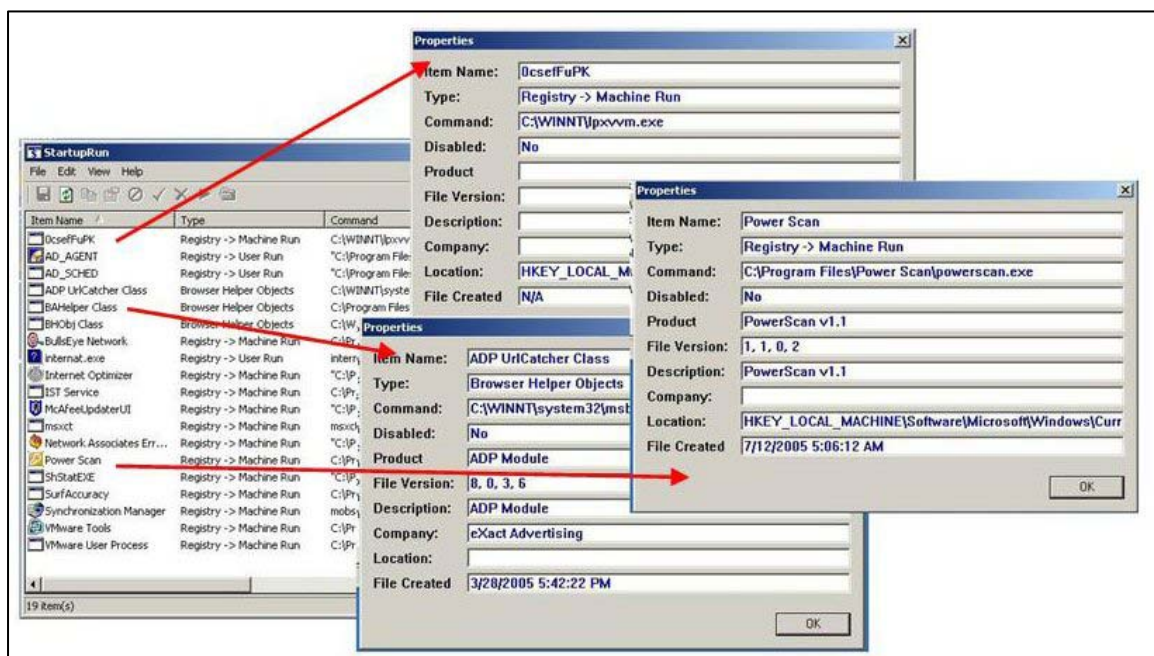


Figure 15 : SartupRun capture after we install the MTV BHO

VI.2 Registry Run and RunOnce keys

Adware/spyware and malware programs frequently use the `Software\Microsoft\Windows\CurrentVersion\{RunOnce, Run}` branch of `HKEY_LOCAL_MACHINE` and the `HKEY_CURRENT_USER` registry hives to store state information related to their execution across system reboots or user logins. By using the Run or Runonce keys, the software is executed when either the machine is powered on, or when the user logs into his account.

The StartupRun tools in the previous section read these keys and display the data.

A typical **RunOnce** entry will look like this:

```
Software\Microsoft\Windows\CurrentVersion\RunOnce
KeyValue = "tvs_re"
KeyData = C:\ProgramFiles\Java\tvs_re_inst.exe
```

A program added under the **RunOnce** entry will be executed only once, and after execution, its related **KeyValue** (here `"tvs_re"`) will be automatically deleted from the system.

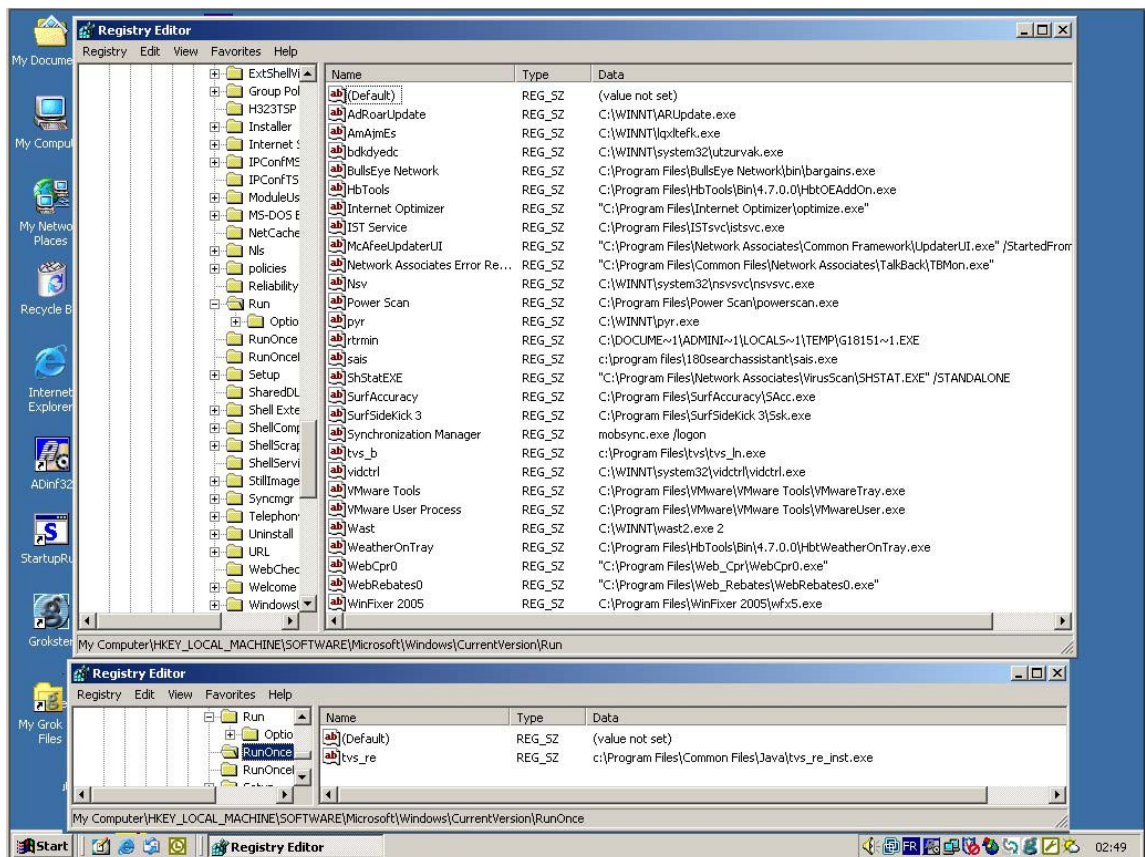


Figure 16: Regedit display

A typical **Run** entry will look like this:

```
Software\Microsoft\Windows\CurrentVersion\Run
KeyValue = "sais"
KeyData = "C:\ProgramFiles\180searchassistant\sais.exe"
```

The command under the **Run** key will always be executed when a user logs in or boots the system. Both these key branches can be under the `HKEY_LOCAL_MACHINE` and `HKEY_CURRENT_USER` hives. The former will execute the when the system is booting up while the latter will execute when the specific user has logs in.

VI.3 What is a CLass Identifier?

In COM world, you have to identify different pieces such as type libraries and applications. These pieces must be unique in the world. COM uses the **Globally Unique Identifier (GUID)** to define these different IDs.

A GUID is a 128-bit number, usually represented in hexadecimal, which is guaranteed "to be unique across space and time". In the following example, this number is a GUID:

```
{21B4ACC4-8874-4AEC-AEAC-F567A249B4D4}
```

COM borrows this identify system to the Distributed Computing Environment (DCE) naming scheme. Based on the GUID, the main COM IDs are:

- LIBID: the Type Library ID.
- APPID: the Application ID.
- CLSID: the globally unique identifier that identifies a COM class object.
- IID: the Interface ID.

Frequently, spyware/adware applications use COM objects to achieve the software reusability and adaptability benefits. An understanding of how the relevant parts of the system may be altered during a COM objects installation is important to for the detection and removal of spyware/adware programs.

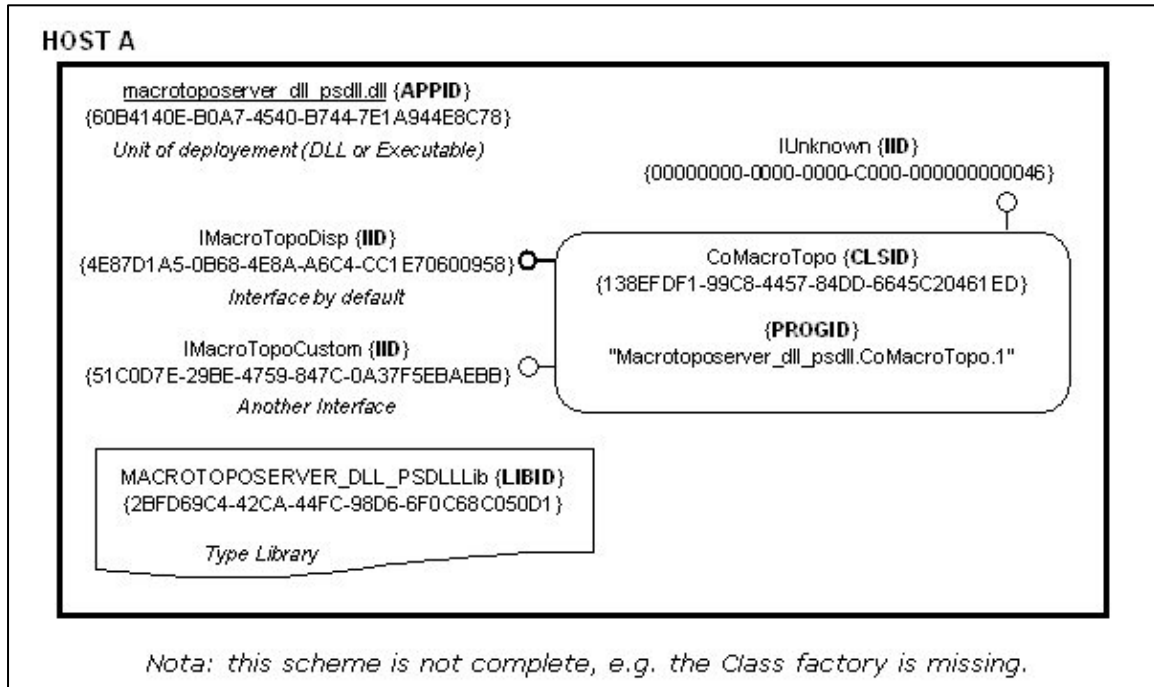


Figure 17: The COM GUID scheme⁴

A COM server is a binary file that contains the code for the methods used by one or more COM classes. This server can be packaged as either a DLL or an executable file. Two types of activation requests for an object exist:

- *In-process* activation request requires a DLL-based version of the COM server to load it into the client's memory space;
- *Out-of-process* activation request requires an executable to start the server process.

⁴ Source : The COM Project :

http://www.codeproject.com/com/mmtopo_comid.asp?msg=580696#SubSections

To allow client programs to activate objects without concern for what type of package is to be used or where the executables or DLLs are located, COM stores configuration information in the registry that maps the CLSIDs onto the server that implements that class. Whenever an activation request is made for a CLSID in a given machine the registry is consulted. If the configuration information is not available in the registry, the registry may redirect the request to a remote host from which the relevant code may be downloaded and installed.

On a local workstation, CLSIDs are stored in the following registry keys:

- o HKEY_CLASSES_ROOT, for backward compatibility.
- o HKEY_LOCAL_MACHINE\Software\Classes on Windows NT 4.0.
- o HKEY_CURRENT_USER\Software\Classes on Windows 2000.

Machine-wide information related to CLSIDs is available in the following registry keys:

- o HKEY_CLASSES_ROOT\CLSID\{GUID}
- or
- o HKEY_LOCAL_MACHINE\Software\Classes\CLSID\{GUID}
- o HKEY_CURRENT_USER\Software\Classes, \CLSID\{GUID}

When the *MTV BHO* was installed onto a VMWARE testing machine, 139 keys were created and 406 values added. The *InCtrl5* tool was able to list these changes.

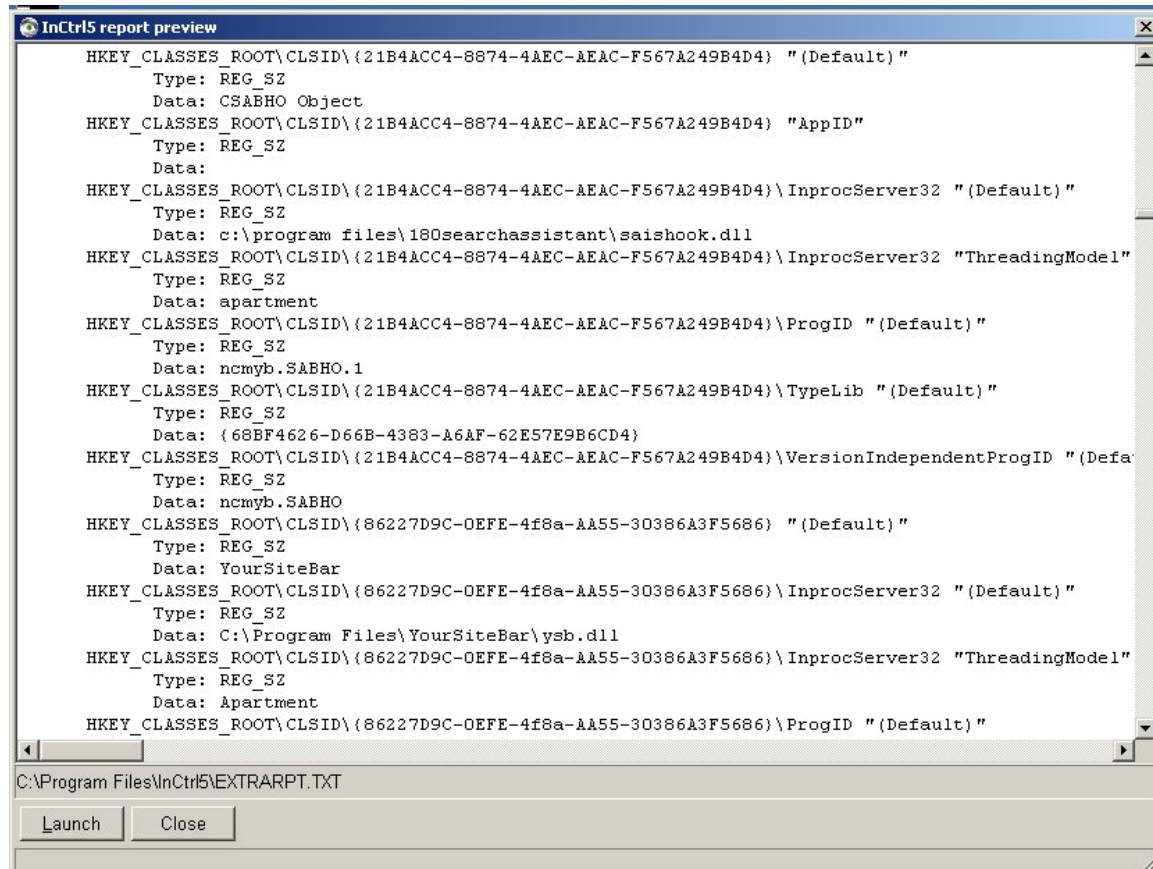


Figure 18: *InCtrl5* report (size of TXT file is greater than 130 Kbytes)

Among the changes, one class is used to implement BHO functionality. The registry entry was:

```
HKEY_CLASSES_ROOT\CLSID\{21B4ACC4-8874-4AEC-AEAC-F567A249B4D4} "(Default)"  
Type: REG_SZ  
Data: CSABHO Object
```

The class is named CSABHO.

To allow local activation of CSABHO objects, the CSABHO's CLSID entry in the registry has a subkey that indicates the file that contains the executable code for the CSABHO class' methods. Here, the COM server is packaged as a DLL, an **InprocServer32** key entry is required in the registry:

```
HKEY_CLASSES_ROOT\CLSID\{21B4ACC4-8874-4AEC-AEAC-F567A249B4D4}\InprocServer32
"(Default)"
Type: REG_SZ
Data: c:\program files\180searchassistant\saishook.dll
```

If the COM server had a package using an EXE file, it would use a **LocalServer32** context name like this:

```
HKEY_CLASSES_ROOT\CLSID\{21B4ACC4-8874-4AEC-AEAC-F567A249B4D4}\LocalServer32
"(Default)"
Type: REG_SZ
Data: c:\program files\180searchassistant\saishook.exe
```

RemoteServer32 is the third common context name. It would specify that the component is located on a remote machine.

To facilitate activation calls, programmers may use ProgIDs. The CLSID to ProgID translation is achieved through the following registry entry:

```
HKEY_CLASSES_ROOT\CLSID\{21B4ACC4-8874-4AEC-AEAC-F567A249B4D4}\ProgID "(Default)"
Type: REG_SZ
Data: ncmyb.SABHO.1
```

Conversely, to support the reverse mapping (ProgID to CLSID), we have:

```
HKEY_CLASSES_ROOT\ncmyb.SABHO\CurVer "(Default)"
Type: REG_SZ
Data: ncmyb.SABHO.1
HKEY_CLASSES_ROOT\ncmyb.SABHO.1 "(Default)"
Type: REG_SZ
Data: CSABHO Object
HKEY_CLASSES_ROOT\ncmyb.SABHO.1\CLSID "(Default)"
Type: REG_SZ
Data: {21B4ACC4-8874-4AEC-AEAC-F567A249B4D4}
```

VI.4 ShellServiceObjectDelayLoad registry key

This is similar to a Run key but instead of pointing to a file, it points to the CLSID's InProcServer32. The InProcServer32 contains the information about the particular DLL file to be used.

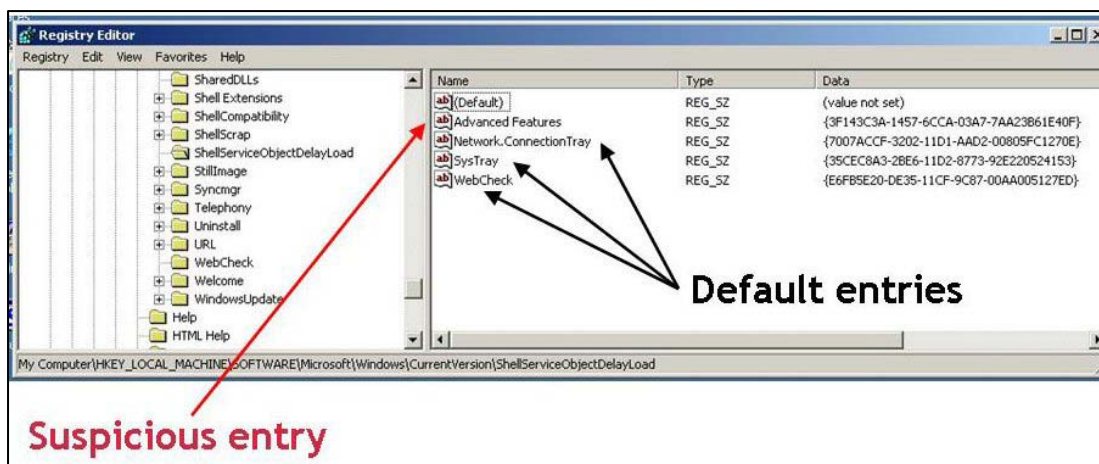


Figure 19: ShellServiceObjectDelayLoad infection (Proxy-Thunker)

This key is in registry as:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellService
ObjectDelayLoad, {CLSID-Value}
```

The files under this key are loaded automatically by Explorer.exe when the computer starts. Because Explorer.exe is the shell for the computer, it will always start, thus the files under this key will always be loaded, and early in the startup process.

Most normal setups will usually only have the following 3 entries:

- Network.ConnectionTray
- Systray
- WebCheck

VI.5 Internet Explorer Start & Search registry keys

Appendix A lists the default keys for some Internet Explorer versions. These are the same for the English and French version of IE. The reader should manually compare these keys in the registry with his registry values to determine if the browser is clean. Adware and spyware are in the habit of changing the following entries:

- o Start,
- o Search
- o Default_Page
- o Default_Search_URL

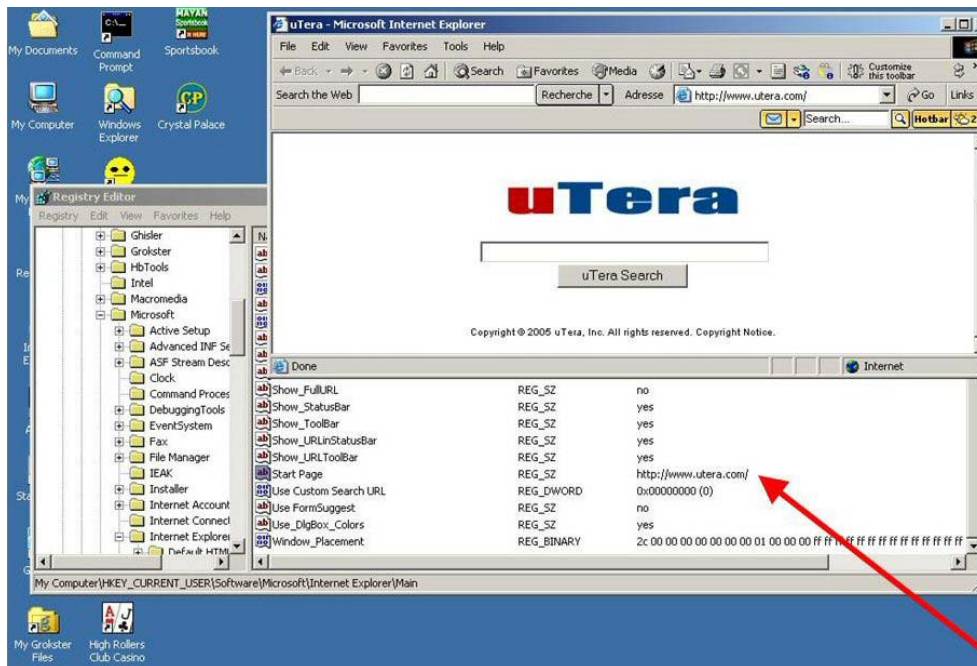


Figure 20 : How adware can change the Registry

VI.6 Internet Explorer Toolbar registry keys

The key `HKLM\SOFTWARE\Microsoft\Internet Explorer\Toolbar` contains the CLSIDs for the toolbars that are invoked by Internet Explorer.

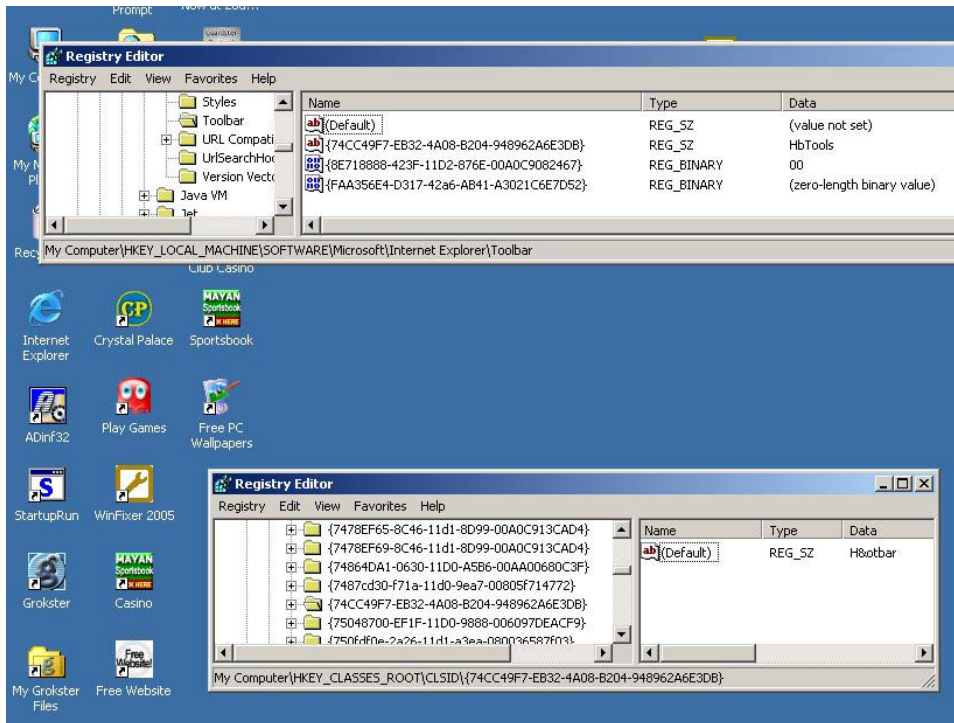


Figure 21: Regedit is able to show the presence of a toolbar

Each of these CLSIDs is mapped to the respective DLLs implementing the toolbar and this mapping information is present in the `HKCR\CLSID\<clsid-values>` branch of the registry.

VI.7 Internet (Explorer) Advanced Options

Adware like *CommonName* adds a section *CommonName* with options in the *Advanced* tab of MSIE options. This is achieved by creating subkeys under the registry key:

```
HKKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\AdvancedOptions
```

The next figure shows an example of a *CommonName* infection and the section added to the MSIE options window. These are present in the registry under the following keys:

```
HKKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet
Explorer\AdvancedOptions\CommonName
HKKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet
Explorer\AdvancedOptions\CommonName\BrowserAgent
HKKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet
Explorer\AdvancedOptions\CommonName\ResolveBookmarkName
HKKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet
Explorer\AdvancedOptions\CommonName\ResolveIntranetName
HKKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet
Explorer\AdvancedOptions\CommonName\Tooltip
```

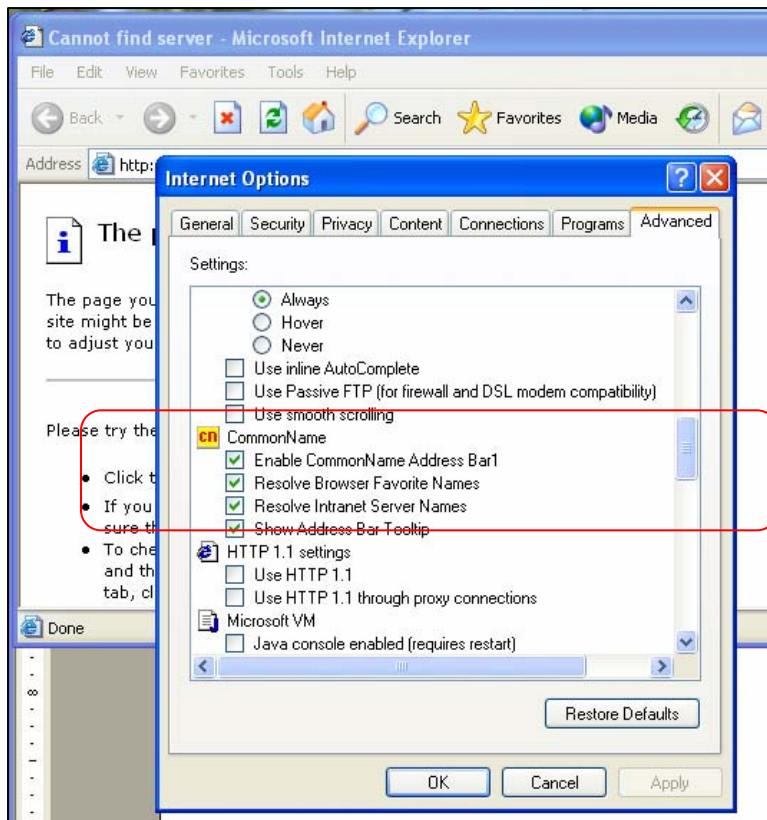



Figure 22: CommonName adds entries in the MSIE advanced folder

VI.8 Extra items in Internet Explorer Tools menu

Some adware create extra items under the IE Tools Menu. This place is used for adding information for branding purposes or after a Windows Update (MSN Messenger Button). The registry key implementing this is:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Extensions\{CLSID=Value}
```

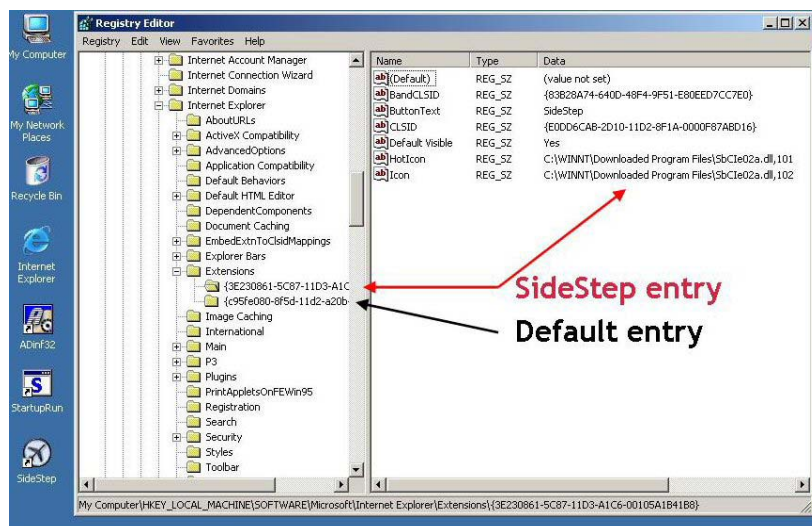


Figure 23: IE extra item infection (Adware-SideStep)

VI.9 User StyleSheet Hijacking

Internet Explorer supports the option of user-defined style sheet for all pages viewed in the browser. This method involves the overwriting of an existing style sheet (added by the PC user) or the implementation of a new style sheet. This feature can cause a change in the Default Home Page, popups and may also carry system slowdowns.

The key related to the style sheet is stored in:

```
HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Styles
```

No style is defined in a standard configuration.

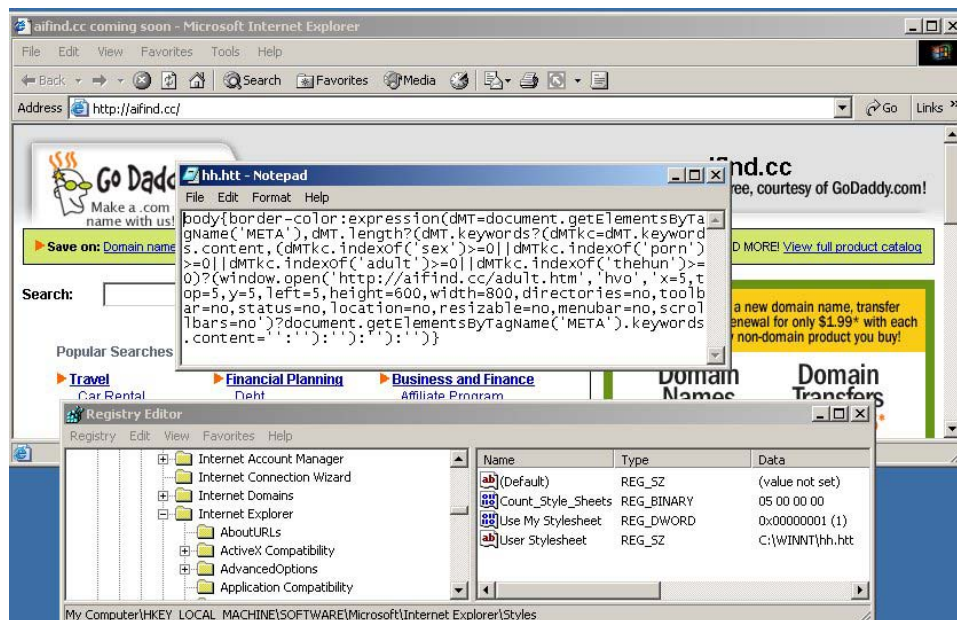


Figure 24: StyleSheet infection (Startpage-AT)

VI.10 DLL Injection

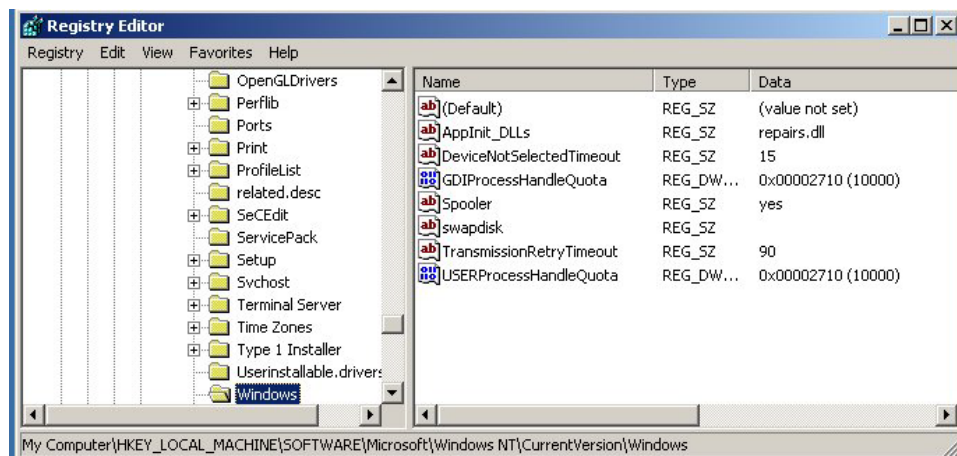


Figure 25: Adware SurfSideKick uses the DLL injection feature

This is achieved by adding the DLL pathname as a data value for a subkey in the following registry key:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs
```

During system initialization when *user32.dll* is mapped into a process' address space it automatically reads these registry key entries and loads any DLL mentioned in the data value field. Few legitimate programs use it. Mostly, it is used by Trojans or browser hijackers.

VI.11 Trusted Zone entries

The websites defined in the **Trusted Zone** entries are trusted by Internet Explorer. Any scripts from these sites will be automatically executed. The following registry key defines this:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet  
Settings\ZoneMap\Domains\{the trusted domain name}
```

This location is empty in many standard configurations.

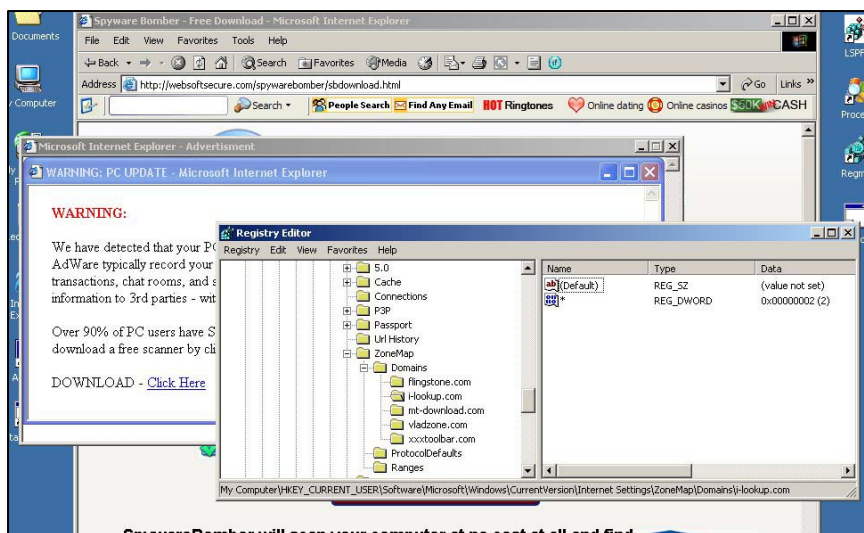


Figure 26: 5 Trusted sites are added after running *Downloader-NC* (ISTbar installation)

VI.12 Internet Protocol Hijack

Internet Explorer provides filtering mechanism through plugins. Filters are content types accepted by the browser.

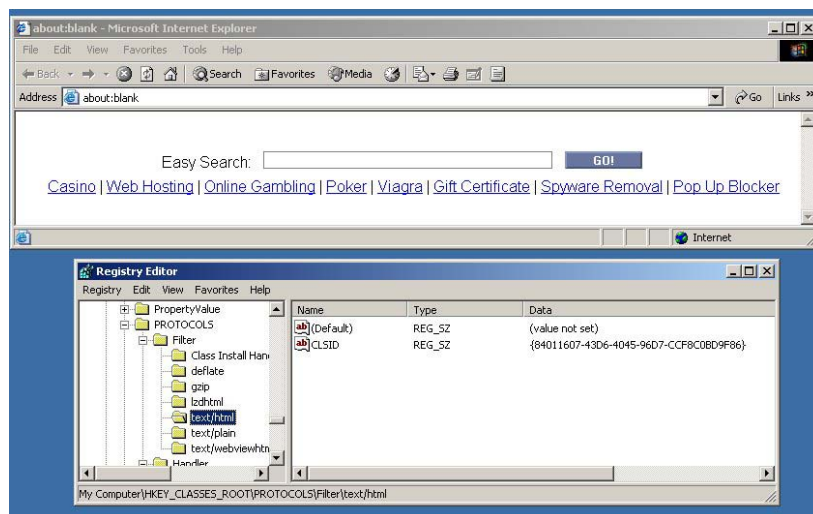


Figure 27: Protocol Hijacking after running *StartPage-EF*

If a filter exists for a particular content type, IE passes the content through the handler. *Startpage-EF* adds *text/html* and *text/plain* filters allowing them to hook all the webpage content passing through the browser.

The key containing this information is at:

HKEY_CLASSES_ROOT\PROTOCOLS\Filter

VI.13 New Favorites

Internet Favorites are easily visible via the dedicated button. New entries appearing without your knowledge are suspicious and are indicative of adware infection.

It is also possible to manage these entries via Windows Explorer in the *Favorites\Links* subdirectory (in *Document and Setting*).

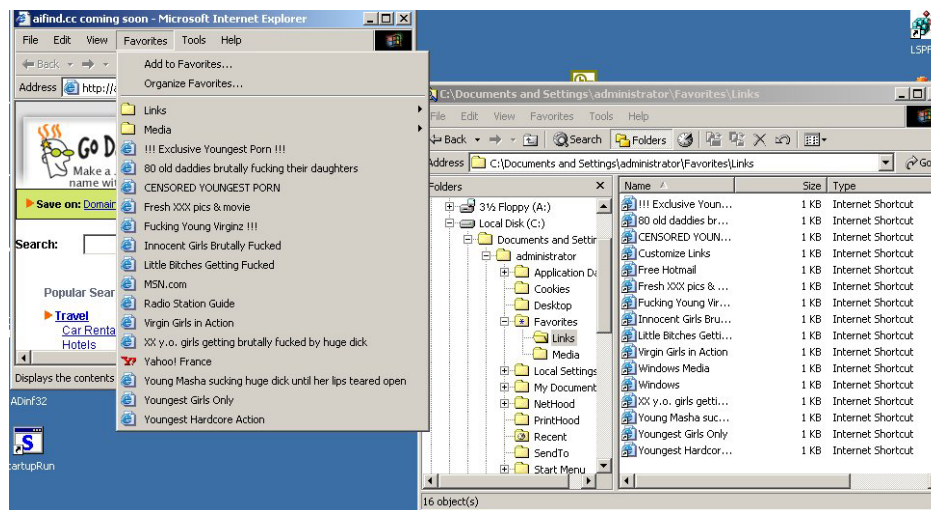


Figure 28: New entries appear under Favorites after *Startpage-AT* is introduced

VI.14 CLSID lists

When a suspicious entry is found in the registry, the thing to do would be to do some research on the filename, company name, or whatever information is provided in that registry entry. Of course, Google is a very good way to go about this. Additionally, there are some dedicated sites on the Internet that lists CLSIDs and their associated uses, as well as who, what, where and how they are involved.



Figure 29 : The *Castlecops* CLSID list

You can find this information at sysinfo.org⁵ or castlecops.com⁶.

VII - Cleaning Adware and Spyware

To remove an adware infection, the CLSID entries in:

- HKEY_CLASSES_ROOT
- HKEY_LOCAL_MACHINE\Software\Classes
- HKEY_CURRENT_USER\Software\Classes

must first be deleted. They are installed directly by the COM server.

Recall, we also noted other CLSID entries added inside the following registry keys:

- HKLM\SOFTWARE\Microsoft\Internet Explorer\Toolbar
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellServiceObjectDelayLoad, {CLSID-Value}
- HKEY_CLASSES_ROOT\PROTOCOLS\Filter (Plugin entries)

Before removing all these keys (and all the associated files), the researcher needs to have studied and determined the legitimacy of each key. Experience and CLSID lists are good ways for achieving this task.

Depending on the adware type, we may also need to delete or reinstate various entries noted previously. Particularly,

- The IE Start & Search registry keys (Appendix A lists the default values of some registry keys)
- The IE Advanced Option subkeys added in:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\AdvancedOptions (these would be deleted)
- The extra items added in the IE Tools menu:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Extensions\{CLSID-Value} (these would be deleted)
- The IE style sheet in use. If no style sheet is required (default), we would uncheck the *User style sheet* tab.
- The DLL injection subkey: HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs (this should be deleted).
- Unwanted **Trusted** sites present in the section:
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\Domains\{the trusted domain name} (these must be deleted)
- The StyleSheet entries added in HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Styles (these must be deleted)
- Additional entries inserted under Favorites.

VII.1 Need for booting in Safe Mode

Adware and spyware programs are composed of multiple component files with EXE, DLL or OCX extensions. While the machine is running, the adware process might have some of these files locked. They could also be injected into a system process (and thus can't be removed) such as EXPLORER.EXE, as shown in the next figure.

⁵ <http://www.sysinfo.org/bholist.php?type=text&subtype=bho>

⁶ <http://castlecops.com/CLSID.html>

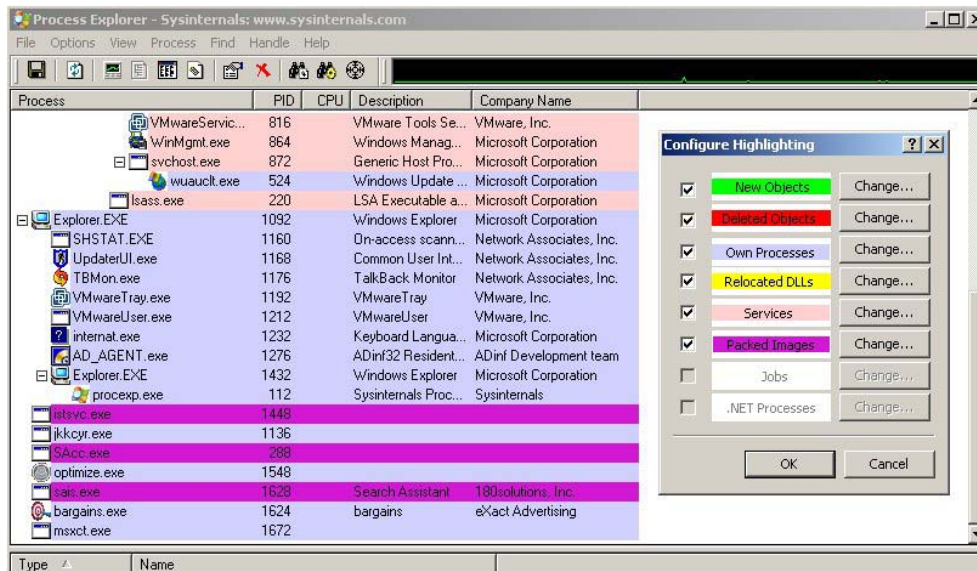


Figure 30: ProcExp screenshot showing AD_AGENT as part of Explorer, after the installation of the MTV BHO.

In order to delete the files encountering the “file in use” problem, we need to eliminate the processes from being “in use.” We accomplish this by booting the OS in safe-mode. This assures that the minimum required system components are loaded. Also, any program defined in the Run keys will not be executed in safe-mode. Consequently, the DLL or EXE component of the adware will not be in use and can be safely deleted from the system (inside the registry and physically).

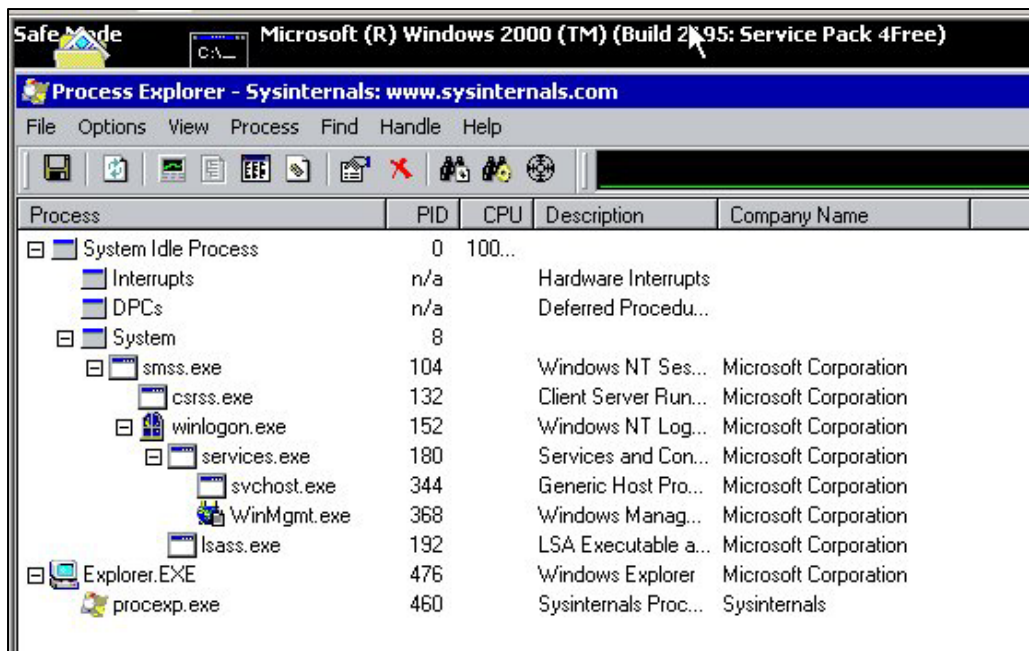


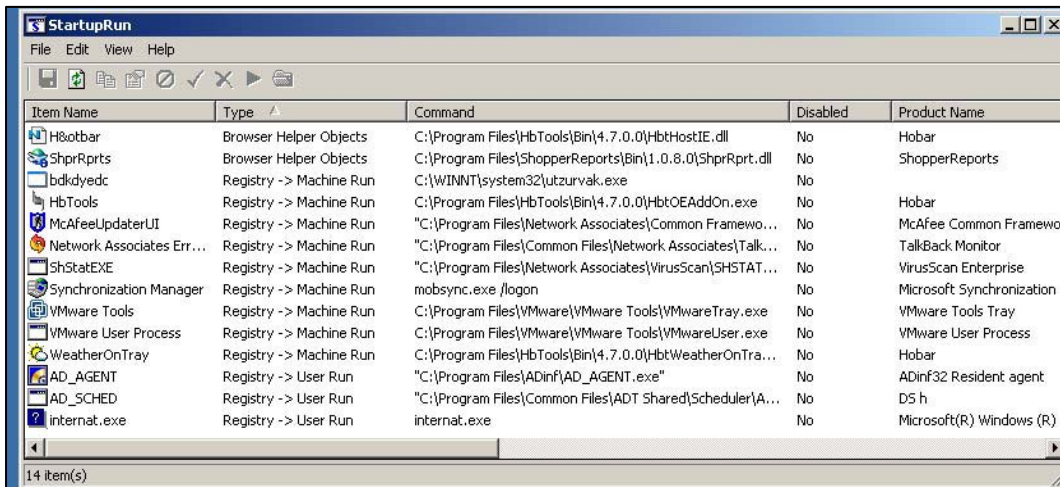
Figure 31: ProcExp screenshot showing W2000 safe mode, even though MTV BHO is installed.

VII.2 BHO Example

Browser Helper Objects (BHOs) are frequently used by adware programs to display pop-ups on infected PCs. They are implemented as DLLs and loaded by Internet Explorer (EXPLORE.EXE) whenever the web browser starts.

When an extra toolbar is added to IE (like the Yahoo or Google toolbar), it is implemented by using BHOs. At any moment, IE may have multiple BHOs, with many of them valid.

To unregister an Adware BHO, we need to first identify the DLL and its location. The *StartupRun* tool can be used for this.



Item Name	Type	Command	Disabled	Product Name
HbToolbar	Browser Helper Objects	C:\Program Files\HbTools\Bin\4.7.0.0\HbtHostIE.dll	No	Hobar
ShprRpts	Browser Helper Objects	C:\Program Files\ShopperReports\Bin\1.0.8.0\ShprRprt.dll	No	ShopperReports
bdkdydc	Registry -> Machine Run	C:\WINNT\system32\utzurvak.exe	No	
HbTools	Registry -> Machine Run	C:\Program Files\HbTools\Bin\4.7.0.0\HbtOEAddOn.exe	No	Hobar
McAfeeUpdaterUI	Registry -> Machine Run	"C:\Program Files\Network Associates\Common Framework...	No	McAfee Common Framework
Network Associates Err...	Registry -> Machine Run	"C:\Program Files\Network Associates\Talk...	No	TalkBack Monitor
ShStatEXE	Registry -> Machine Run	"C:\Program Files\Network Associates\VirusScan\SHSTAT...	No	VirusScan Enterprise
Synchronization Manager	Registry -> Machine Run	mobsync.exe /logon	No	Microsoft Synchronization
VMware Tools	Registry -> Machine Run	C:\Program Files\VMware\VMware Tools\VMwareTray.exe	No	VMware Tools Tray
VMware User Process	Registry -> Machine Run	C:\Program Files\VMware\VMware Tools\VMwareUser.exe	No	VMware User Process
WeatherOnTray	Registry -> Machine Run	C:\Program Files\HbTools\Bin\4.7.0.0\HbtWeatherOnTra...	No	Hobar
AD_AGENT	Registry -> User Run	"C:\Program Files\ADInf\AD_AGENT.exe"	No	ADInf32 Resident agent
AD_SCHED	Registry -> User Run	"C:\Program Files\Common Files\ADT Shared\Scheduler\A...	No	DS h
internat.exe	Registry -> User Run	internat.exe	No	Microsoft(R) Windows(R)

Figure 32: Tracking Adware-Hotbar with StartupRun

Finding the right file(s) is sometimes difficult. It may require analysis of many suspicious DLLs (unpacking the file, searching typical strings, etc.). But in most cases, we can find them by looking at the *CompanyName* field. And experience certainly helps. Here you quickly see 5 intruders.

- 3 files with *Hotbar.com Inc* as the company name
- 1 file with *ShopperReports* as the company name
- 1 file with no valid properties (an a highly suspicious filename)

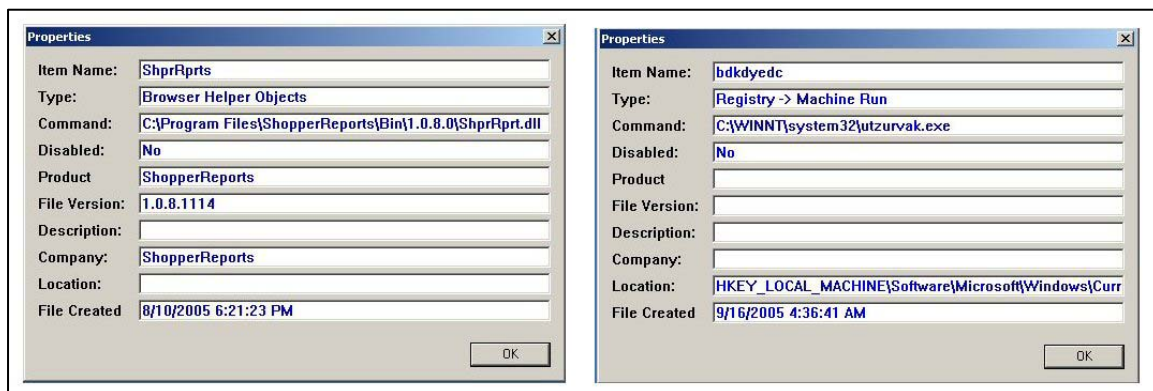


Figure 33: 1 file from ShopperReports and 1 unknown but suspicious file

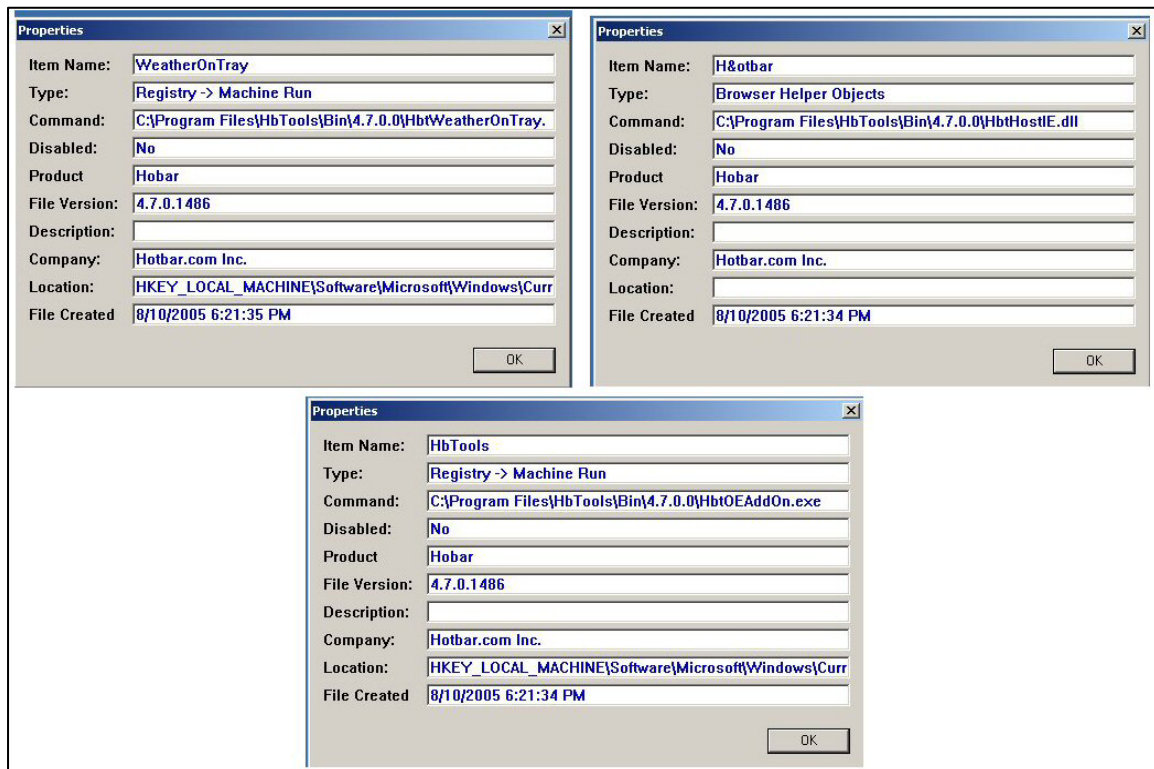


Figure 34: 3 files from Hotbar.com Inc.

Once files are identified, we have to carefully note their names and locations. Both pieces of information are available in the Command field.

Let continue this exercise by using the following example:

- Name : HbHostIE.dll
- Path : C:\Program Files\HbTools\Bin\4.7.0.0\

After booting the system in Safe Mode, we search the CLSID value(s) in the registry branch:

HKEY_CLASSES_ROOT\CLSID

Using **regedit**, we search for any mention of the path. The search could yield multiple instance, as shown in the next figure.

We advise you to save these CLSID values in a TXT file before deleting them.

The search and delete process must be executed in all 3 of the branches we noted before:

- HKEY_CLASSES_ROOT\CLSID
- HKEY_LOCAL_MACHINE\Software\Classes\CLSID
- HKEY_CURRENT_USER\Software\Classes\CLSID

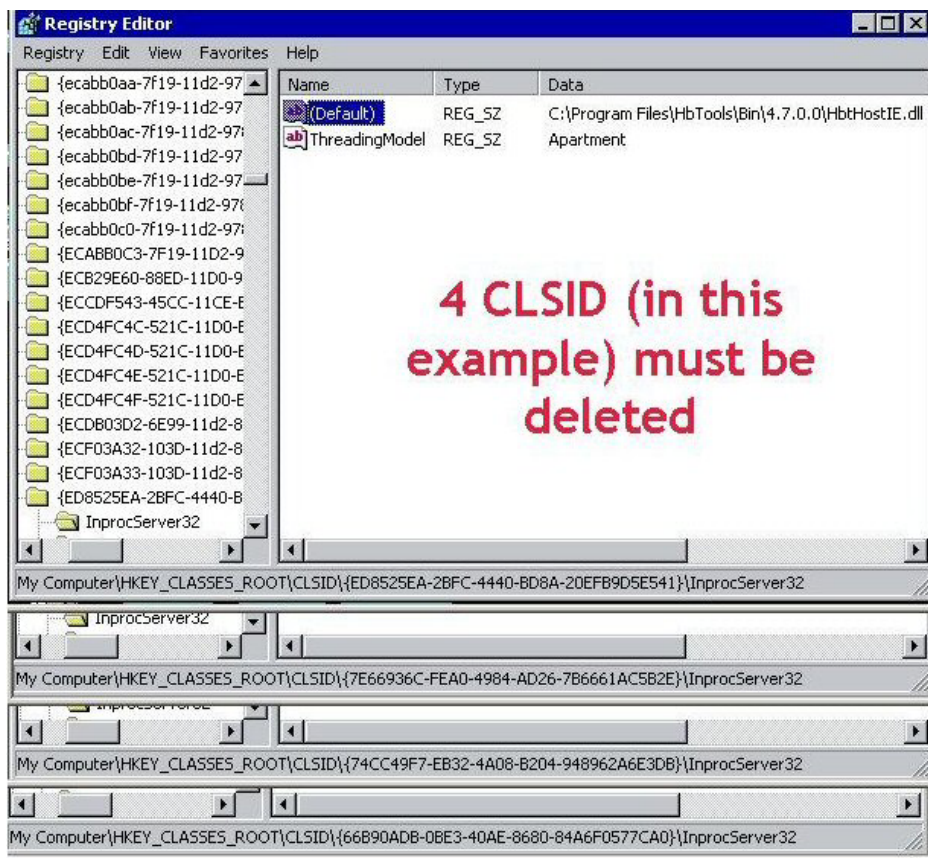


Figure 35 : These CLSID keys related to our *Hotbar* example must be deleted.

Now, run another search using each of the CLSID keys to find:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects\{the found CLSID value(s)}

Delete the key as shown in the next figure.

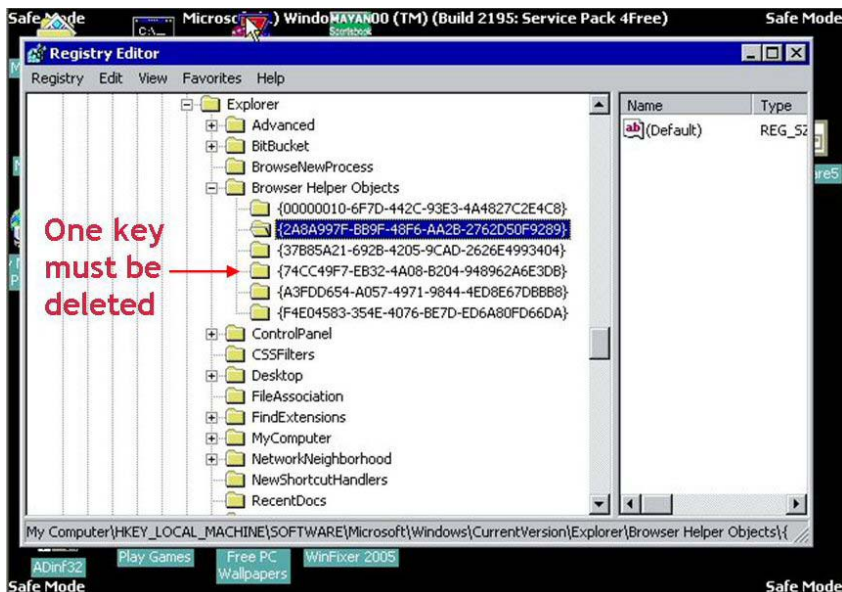


Figure 36: The BHO key(s) linked with the previous CLSID must be deleted

Finally, it is safe to delete the DLL, "C:\Program Files\HbTools\Bin\4.7.0.0\HbHostIE.dll", from the system.

VII.3 Cleaning LSPs

Some spyware and adware installations also use the Winsock 2 (L)ayered and (N)etwork (S)ervice (P)rovider implementations to redirect visits to specific sites. The network data emanating from the machine can be sent to a "central" site and then sent to the user's application to achieve "transparency" at the TCP/IP stack level. For example, the user types www.google.com and the browser first connects "silently" to www.marketscore.com, sends some data, and then connects to the actual google.com site. This is achieved by inserting a layer of service providers above the base service provider layer in the TCP/IP stack.

The base service provider implements the actual details of the transport protocol like setting up and tearing down the connections and doing the actual data transfer and flow control. Winsock2 LSP/NSPs are implemented as standard windows DLLs that export exactly one function named WSPStartup(). This is quite complicated. So as not to delve into details here, readers may refer to the MSDN resources for an exhaustive treatment of this topic⁷. Removal of spyware and adware that use LSPs may be a problem area. If we merely delete the registry entries and files entered during the spyware LSP installation, there is a fair chance that the Winsock2 settings may get clobbered or the LSP chain is broken and hence the network and Internet connections in the machine are lost. For this task a sequence of LSP unregistration functions need to be executed before the deletion of files and related registry entries.

We will use two free tools to clean LSPs. The first is *sporder.exe*. It can be used to view a system's LSP stack. The next figure displays the stack for an uninfected system. The following displays the stack for a system containing an Adware LSP (from www.friend.fr).

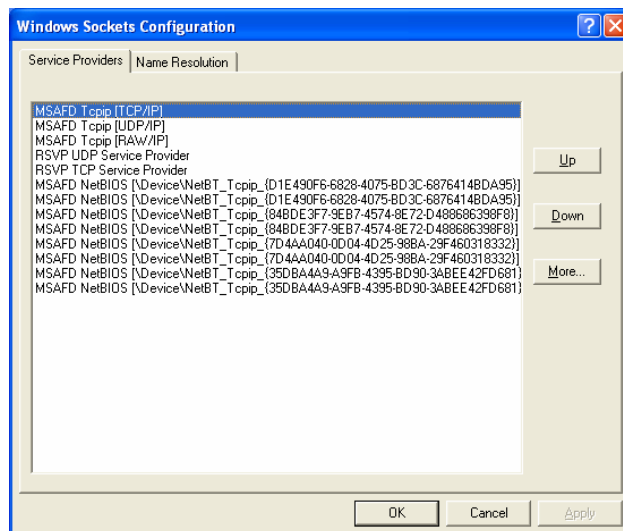


Figure 37 : Stack from an uninfected system

⁷ Unraveling the Mysteries of Writing a Winsock 2 Layered Service Provider
<http://www.microsoft.com/msj/0599/LayeredService/LayeredService.aspx>

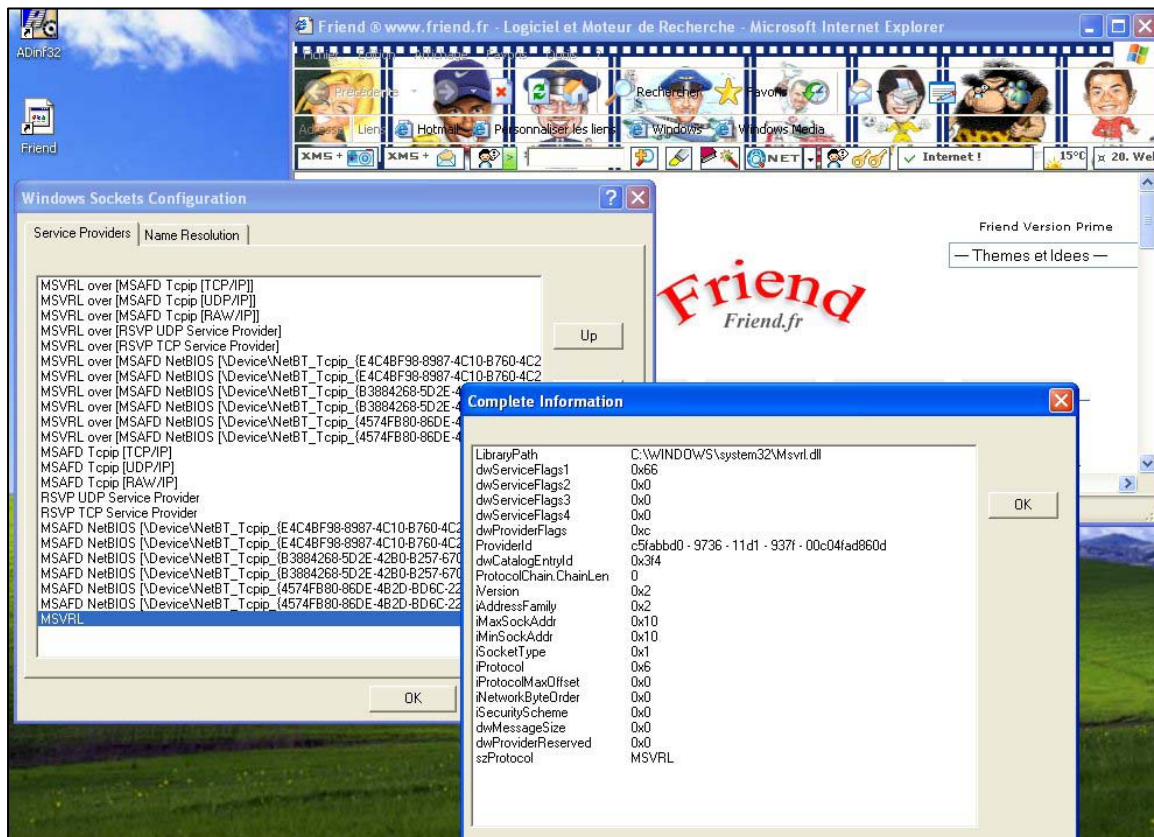


Figure 38: Stack from an infected system.

The second tool is *LSPFix*. We will use it to remove the protocol handler added by the adware and found by *Sporder*. This is a dangerous manipulation that must be done by a competent person. To avoid mistakes, we have to check the box stating "*I know what I am doing*" before removing the file and the related entries in the LSP stacks.

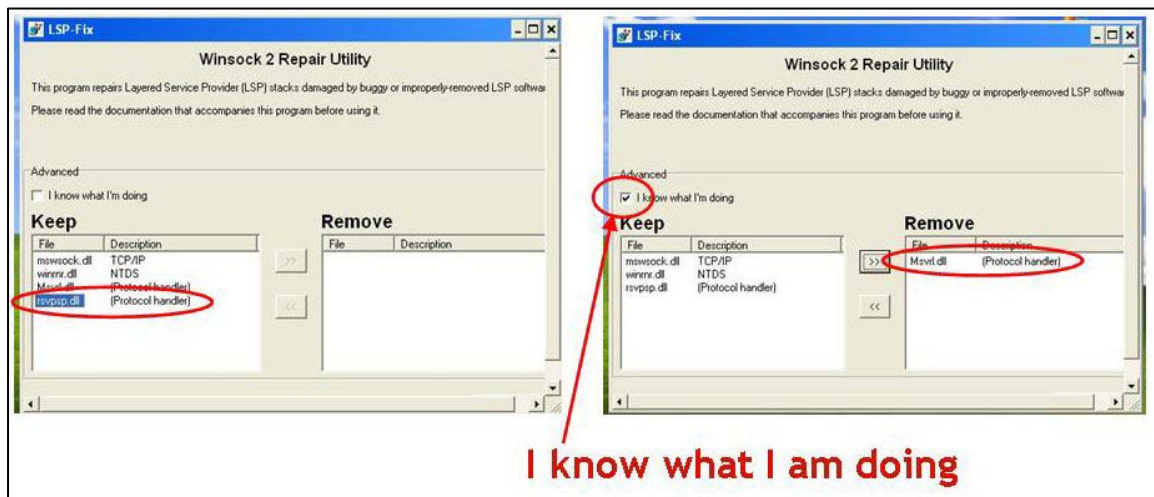


Figure 39: LSPFix: the Winsock 2 repair Utility

The work is done when we click the *Finish* button and a window displays the entries we removed. Another execution of *Sporder* confirms the removal when the display of the stack looks like that of a clean machine.

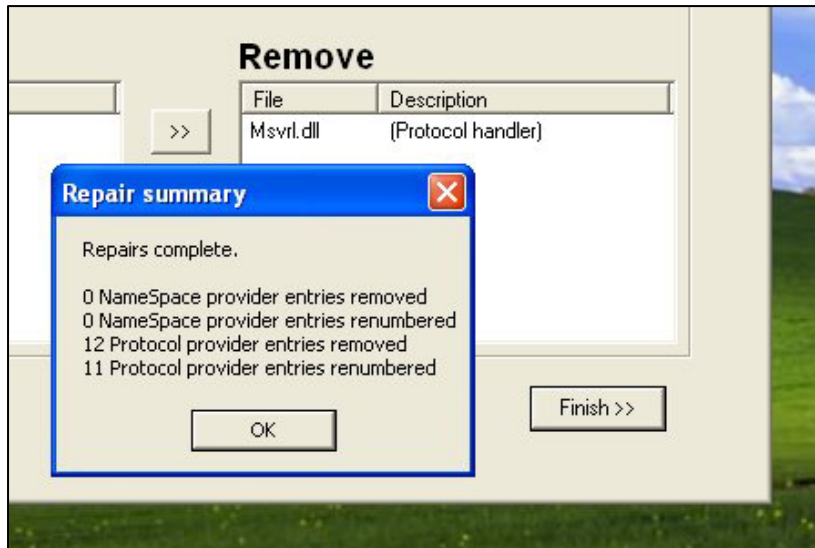


Figure 40 : LSPFix is able to remove the unfriendly entries

Cleaning the LSP stack is only the first part of the job. We now have to clean the Internet Explorer environment using the method already noted in paragraph VII.2

VIII - CONCLUSION

The main impression I am left with after working on this paper is already known to all in the anti-virus profession : it was very easy to clean most of the viruses and Trojans we encountered some years ago. But now some of the new Trojans are more complicated. And adwares and spywares are incredibly complex.

I could have given you very simple examples. I considered using adware that would only be one unique file. But you know as well as I do, that behind one adware, there lie at least half a dozen more binary files that interfere with each other. Adware is developed as a professional software package now and the uninstall process is consequently intricate.

In the past, we were mostly concerned with the increase in work due to the number of new files we had - and we still have - to study detection. For them cleaning was often a simple "kill process and delete" rule.

As adware makers will continue to improve their production, we are now faced with an emerging problem. It is - and it will be - linked to cleaning which will be more and more complex in the future.

APPENDIX A

Lists the default keys for some Internet Explorer versions.

IE 5.0

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Main]
    Local_Page="SystemRoot%\system32\blank.htm"
    Default_Page_URL="http://www.msn.com"
    Default_Search_URL="http://www.microsoft.com/isapi/redir.dll?prd=ie&ar=iesearch"
    Search_Page="http://www.microsoft.com/isapi/redir.dll?prd=ie&ar=iesearch"
    Start_Page="http://www.microsoft.com/isapi/redir.dll?prd={SUB_PRD}&clcid={SUB_CLSID}&pver={SUB_PVER}&ar=home"
[HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main]
    Local_Page="C:\\WINNT\\System32\\blank.htm"
    Search_Page="http://www.microsoft.com/isapi/redir.dll?prd=ie&ar=iesearch"
    Start_Page="http://www.microsoft.com/isapi/redir.dll?prd={SUB_PRD}&clcid={SUB_CLSID}&pver={SUB_PVER}&ar=home"
```

IE 5.5

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Main]
    Local_Page="SystemRoot%\system32\blank.htm"
    Default_Page_URL="http://www.microsoft.com/isapi/redir.dll?prd=ie&pver=5.5&ar=msnhome"
    Default_Search_URL="http://www.microsoft.com/isapi/redir.dll?prd=ie&ar=iesearch"
    Search_Page="http://www.microsoft.com/isapi/redir.dll?prd=ie&ar=iesearch"
    Start_Page="http://www.microsoft.com/isapi/redir.dll?prd={SUB_PRD}&clcid={SUB_CLSID}&pver={SUB_PVER}&ar=home"
[HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main]
    Local_Page="C:\\WINNT\\system32\\blank.htm"
    Search_Page="http://www.microsoft.com/isapi/redir.dll?prd=ie&ar=iesearch"
    Start_Page="http://www.msn.com"
```

IE 6.0

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main]
    Local_Page="C:\\WINDOWS\\SYSTEM\\blank.htm"
    Default_Page_URL="http://www.microsoft.com/isapi/redir.dll?prd=ie&pver=6&ar=msnhome"
    Default_Search_URL="http://www.microsoft.com/isapi/redir.dll?prd=ie&ar=iesearch"
    Search_Page="http://www.microsoft.com/isapi/redir.dll?prd=ie&ar=iesearch"
    Start_Page="http://www.microsoft.com/isapi/redir.dll?prd={SUB_PRD}&clcid={SUB_CLSID}&pver={SUB_PVER}&ar=home"
[HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main]
    Local_Page="C:\\WINDOWS\\SYSTEM\\blank.htm"
    Search_Page="http://www.microsoft.com/isapi/redir.dll?prd=ie&ar=iesearch"
    Start_Page="http://www.microsoft.com/isapi/redir.dll?prd={SUB_PRD}&clcid={SUB_CLSID}&pver={SUB_PVER}&ar=home"
```

APPENDIX B

Test environment

A VMWARE station with W2000 SP4

- 1) BHO installation via: [www . slotch . com / softwares /v4.0 / istdownload.exe](http://www.slotch.com/softwares/v4.0/istdownload.exe) (MTV toolbar)
- 2) Grokster installation via: [www . grokster . com](http://www.grokster.com) (FREE Grokster install - ad supported - grokster_installer.exe)
- 3) Install "Crystal Palace Casino"
- 4) Install "WinFixer 2005"
- 3) Browsing the Internet for 10 minutes, install the offer for "emoticons" add-in ([hotbar . com](http://hotbar.com))
Double click on icons installed on the desktop. When asked to install, I respond "yes".
- 5) Visit [www . shortcutbucks . com](http://www.shortcutbucks.com)
- 6) Visit [www . browserwise . com](http://www.browserwise.com)
- 7) Visit www.digitalnames.net
- 8) Run pop.exe (VirusScan detects this file as Downloader-NC, 2 984 bytes)
- 9) Run wsyc.exe (VirusScan detects this file as Proxy-Thunker, 23 072 bytes)
- 10) Run at.exe (VirusScan detects this file as Startpage-AT, 5 632 bytes)
- 11) Run start-fw.exe (VirusScan detects this file as Startpage-EF, 17 408 bytes)

The LSP experiment was conducted on a Windows XP environment. The adware used was found on [www . friend . fr](http://www.friend.fr)