WHITE PAPER: SYMANTEC SECURITY RESPONSE

Techniques of Adware and Spyware

Eric Chien
Symantec Security Response

From the proceedings of the VB2005 Conference. Used with permission of the author.

Techniques of Adware and Spyware

Contents

Techniques of Adware and Spyware

Contents (continued)

Techniques of Adware and Spyware

Contents (continued)

Abstract

A whole class of threats commonly known as adware and spyware has proliferated over the last few years with very few impediments. These programs are security risks that are typically used to gather marketing information or display advertisements in order to generate revenue. Not only are these threats far more widespread than traditional malware, but they also utilize techniques that are far more advanced than those used in traditional threats. No doubt this is because adware and spyware programs are being created by registered corporations with professional developers rather than by hobbyist virus writers. This paper will examine the techniques used by adware and spyware in their attempts to remain resident on the system and examine the types of data being extracted from the user's system. These techniques will be compared to similar techniques being used by traditional malicious software and we will speculate at the point at which adware and spyware becomes more akin to a Trojan horse. Solutions will be discussed including exploring the necessity of full system repair including repairing the registry, process scanning, and address the removal of other advanced hooking concepts such as Winsock layered service providers.

This paper discusses adware and spyware programs that are typically used to facilitate advertising or gather data for market research. Because these programs are updated constantly and potentially produced by affiliates, each program is identified only by its common name within the paper. Appendix A contains MD5s and other identifying information about the actual samples analyzed.

Background

In 1987, the first publicly recorded use of the word 'adware' appeared on the Internet in the Usenet newsgroup comp.sys.mac. Amusingly, the post refers to a Macintosh application rather than a Windows application.

Example 1: Newsgroup posting referring to adware

```
Newsgroups: comp.sys.mac
Date: Tue, 21-Jul-87 15:39:54 EDT
Subject: FastEddie 3.1 announcement
…
FastEddie 3.1 has been released by Cottage Software as "AdWare" (as they
like to call it) without the restrictions of earlier demo-versions which
were limited to handle small files only. They call it AdWare because a
window comes up every so often which advertises Cottage Software.
```

However, adware would not be on the radar of security companies until 15 years later, when Permissioned Media, Inc. forced anti-virus companies to re-evaluate what was and wasn't a virus. In October 2002, Permissioned Media, Inc. released a software program that sent a link to itself to everyone in the

Microsoft Outlook contact list, just like a mass-mailing email worm. The difference was that this functionality was specified in an end user license agreement (EULA) on installation of the program.

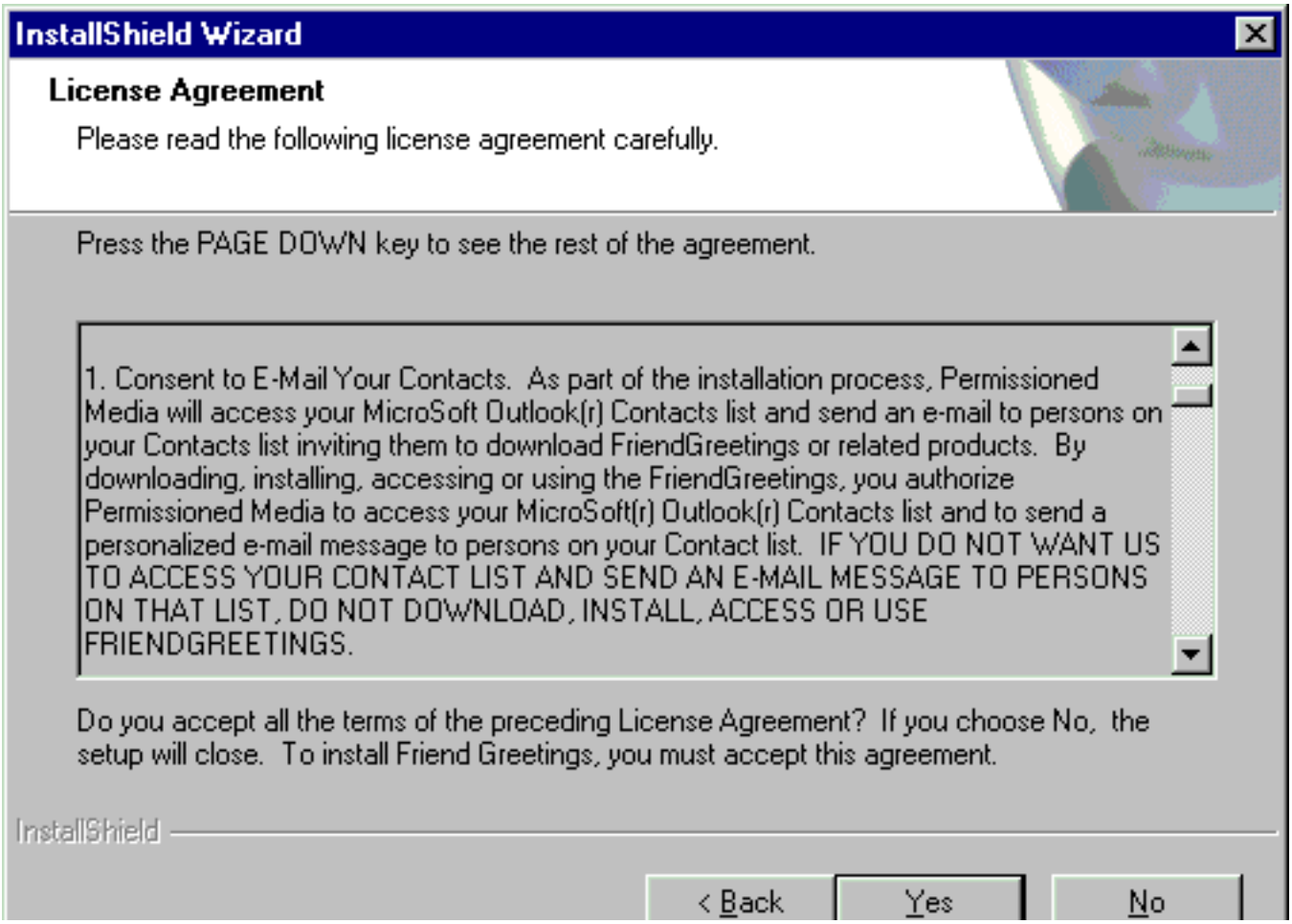

Figure 1: The mass-mailing functionality of W32.Friendgreet.worm was specified in a EULA.

While this program may have been intended to perform market research, due to the fact it self-replicated, it was categorized as a worm, irrespective of the EULA. Today, programs identified as adware and spyware by security vendors generally do not self-replicate, but do often exhibit behaviors more akin to malicious software – such as techniques to trick users into executing the application, avoid being noticed, and prevent removal.

This paper will discuss the common and potentially malicious delivery vectors, load points, anti-removal, and stealth techniques used by adware and spyware and also give a brief overview of the types of data gathered by these programs.

Delivery Vectors

Adware and spyware programs are installed on a system in a variety of ways, but rarely in a conspicuous and forthright manner. In a study by the Ponemon Institute, 47% of respondents had 'no idea' where such programs came from. In addition, 97% did not recall even seeing an end user license agreement.

Most adware and spyware programs are obtained initially by browsing the web or along with some unrelated ad-supported software. The programs are rarely installed from a conspicuous website, but rather through social engineering banner ads, drive-by-downloads, and through peer-to-peer networks with misleading filenames. Some adware and spyware programs are even installed by exploiting software vulnerabilities.

Social engineering banner ads

The first challenge for adware and spyware vendors is to get people to install their software. Virus writers face exactly the same challenge and solve it by using social engineering techniques to entice users into running their creation. They use email messages with message bodies like 'Check out this message' and then attach their virus rather than some legitimate content. Not surprisingly, similar techniques are used by adware and spyware vendors.

Figure 2: An example from the FastClick Ad Network.

Many websites utilize banner ad services where an advertising image is placed on their website. Unfortunately, a large number of these banner ads are completely misleading. Some banner ads utilize an image that mimics a Windows message box with an urgent message tricking computer users into clicking on the image. Once they click on the fake message box they are redirected to other sites that may initiate the installation of adware or spyware or further mislead the user.

For example, some of these fake message boxes will state the user's computer is infected or have some other system problem such as an inaccurate clock. When clicking the fake message box, the user is redirected to install software to correct the problem, when in fact the user was not infected or didn't have an inaccurate clock.

The example shown here comes from the FastClick Ad Network. The message box is not a message box at all, but really just a GIF image and potentially misleading.

Drive by downloads

Another confusion technique used by adware and spyware vendors is the use of drive by downloads. Drive by downloading is the action of prompting a user to install a program as they browse the web without the user actually requesting the installation of any program in the first place. A drive by download is usually invoked via automatic web page refresh or ActiveX control installers.

Automatic refresh

Automatic page refresh occurs when a web page simply redirects the browser to an EXE causing a dialog prompt to be displayed. This redirection can be achieved in a variety of ways including using simple HTML or Javascript.

Example 2: Page redirection using HTML and Javascript

```
HTML

<meta http-equiv="refresh" content="0;url='http://127.0.0.1/example.exe">

Javascript

<script>
location.href='http://127.0.0.1/example.exe'
</script>
```

Both of the above methods will result in an installation prompt being displayed immediately when visiting the web page (see Figure 3).

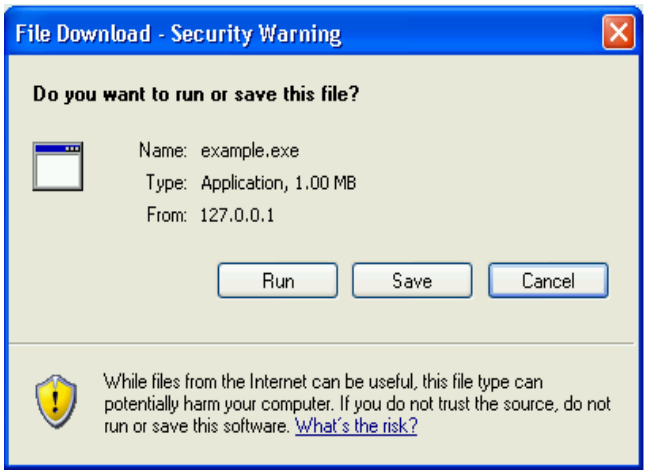

Figure 3: The installation prompt displayed on visiting the web page.

While the Internet Explorer prompt warns the user the application could contain malicious code, the same prompt is used for any program the user tries to download and thus, most users have learned simply to ignore the message and select 'Run'. To avoid this install prompt, many adware and spyware vendors have moved to using ActiveX control installers instead.

Active X

ActiveX is a set of technologies developed by Microsoft to allow programs to interoperate. Part of these technologies is an ActiveX control. ActiveX controls are designed as an extension to static HTML content, providing more dynamic web content.

When viewing a web page that requires an ActiveX control to invoke dynamic content, the browser will prompt the user to download and install the ActiveX control. Most adware and spyware programs don't use this method to present dynamic web content, but rather to trick the user into downloading and installing the ActiveX control, which is nothing more than the adware or spyware program.

The ActiveX delivery method is often more effective than the automatic refresh method because the vendor can control the text in the install prompt. The example shown in Figure 4 was an ActiveX control install prompt that was displayed when visiting spazbox.net. The text 'Website Access Click YES ...' is provided by the installer itself and was supposed to contain the product name. By being able to control this text within the dialog prompt, an ActiveX control can further social engineer a user into selecting 'Yes' and installing the executable.

By default, Internet Explorer only installs digitally-signed ActiveX controls. For example, the control shown in Figure 4 was signed by VeriSign.

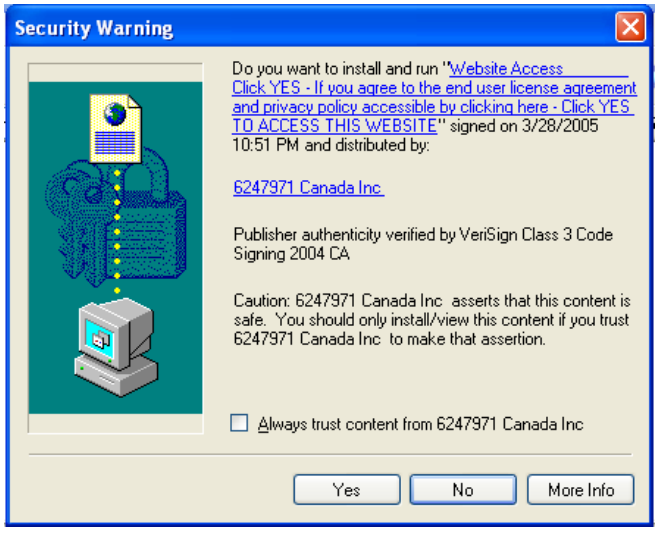

Figure 4: An ActiveX control install prompt displayed when visiting spazbox.net.

While VeriSign verifies that the company exists, it does not verify the content, nor does it verify the signature text, which is displayed in the install prompt. Thus, while signed content is theoretically supposed to be safer because it is traceable, in practice this is not the case and obtaining a signature for an application is not a major hurdle or deterrent.

Continual prompting

Unfortunately, declining an installation prompt isn't enough to prevent all adware and spyware programs from being installed on the system. Some vendors employ continual prompting, redisplaying the install prompt until the user gives up and consents to the installation.

For example, toolbarcash.com (Figure 5) provides code that will cause 'auto-prompt install and persistent retry'. Toolbarcash.com achieves this with JavaScript that continually prompts one to install the ActiveX installer.

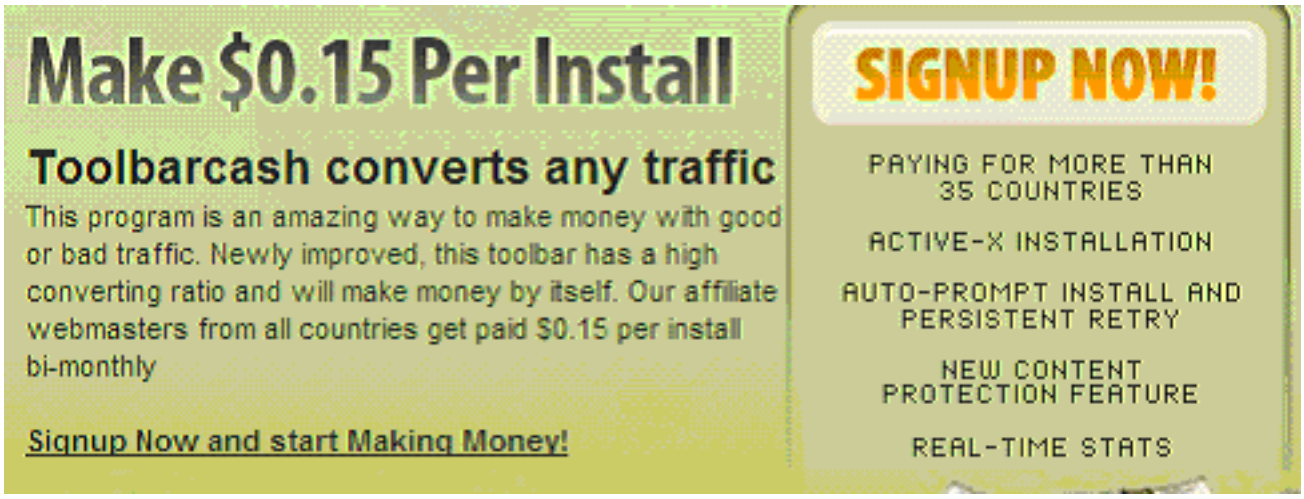

Figure 5: Toolbarcash.com provides code that will cause 'auto-prompt install and persistent retry'

Bundled and chained installs

Another common delivery vector is to bundle adware and spyware with some other third-party software. For example, when installing Kazaa, programs from Cydoor, GAIN Network, InstaFinder, and Rx Toolbar are also installed, all of which facilitate the delivery of advertising or gather marketing data.

While Kazaa currently provides clear notification that these programs will be installed, many software programs bury notification of the bundled program in the end user license agreement (EULA). Furthermore, once these programs are installed they often download additional programs or display advertisements that are misleading and entice users to download additional security risks. The method of downloading further programs is known as chained installs.

With adware and spyware detested by most computer users, one may wonder why third-party software companies would bundle these programs in their software. Money is the answer. For example, rates range from pennies to $0.25 per install, which can add up, especially if the software is as widely distributed as Kazaa. Appendix B contains some example emails with revenue rates.

Peer to peer installation

A relatively new delivery vector being used is seeding adware and spyware programs on peer to peer networks. These files are usually labeled with enticing filenames or are even bundled with pirated media such as television shows or movies.

For example, software that installs programs from vendors such as Direct Revenue, 180Solutions, and Exact Advertising has been discovered via popular BitTorrent tracking websites.

Some companies, such as the Marketing Metrix Group, even specialize in distributing programs on peer to peer networks.

Exploits

While most adware and spyware programs obtain user consent, albeit with misleading or inconspicuous disclosure, some are installed on systems without any user consent. This occurs by exploiting vulnerabilities in Internet Explorer that allow content to be automatically downloaded and executed.

Adware and spyware vendors distance themselves from these techniques, even though they may be aware the technique is used by an affiliate. The following emails were uncovered by the Federal Trade Commission (FTC) in their investigation against Seismic Media, a company that would place banner ads on websites via large online advertising companies. Instead of innocuous banner ads, they would insert code causing automatic installation of programs that would eventually install software from companies like 180Solutions and Integrated Search Technologies.

Example 3: Emails uncovered in the investigation against Seismic Media

```
From: MasterWebFanClub@aol.com
To: jared@optintrade.com
Date: Fri, Nov-28-2003 12:37 PM
Subject: strategy
I do my sneaky shit with adv.com today through
Sunday – everyone's off anyway…. You then send an email to your contact
early Monday AM saying the advertiser was unethical and pulled a switch and
you are no longer doing business with them… Then we stop buying adv.com
through you in any way.
```

```
From: MasterWebFanClub@aol.com
To: jared@optintrade.com
Date: Sat, Mar-6-2004 4:51 PM
Subject: I DID IT
I figured out a way to install an exe without any user interaction. This is
the time to make the $$$ while we can.
```

This practice continues today. IFrameDollars is an example of an affiliate that utilizes a variety of exploits to install and execute a program automatically. The program in turn downloads a wide variety of programs from adware to diallers, including advertising software from MediaTickets.

IFrameDollars exploits the MHTML URL Processing Vulnerability (CAN-2004-0380), the Microsoft Windows Cursor Vulnerability (CAN-2004-1049), and the Microsoft Java Virtual Machine Vulnerability (CAN-2003-0111), all of which allow an executable to run on the system without the user's knowledge.

The software installed includes programs that modify the Internet Explorer homepage, display advertising, modify search results, dial high-cost pay numbers, and track user computing habits. These types of program would traditionally be identified as adware, spyware, and diallers, but since they are installed via an exploit some security vendors identify them as malicious software. Unfortunately, the lack of installation prompts in these programs contributes to their ability to be used by rogue affiliates who install them surreptitiously. If they did not silently install, the user would be notified immediately of their execution.

Load Points

Once adware and spyware has been installed on the system, it needs to ensure that it restarts each time the computer starts.

Windows provides load points at different times during system startup. Load points exist for when Windows starts, when one logs in, when the shell (Explorer) starts, and when applications start.

Almost all malicious software uses at least one load point to ensure persistence across reboots. Adware and spyware use load points for the same reason, but also sometimes as a method to extract data.

The most common load point is the Run registry key, `HKEY_LOCAL_MACHINE\Software\Microsoft\CurrentVersion\Run`. Windows executes the file specified in this registry location on startup.

Another system startup load point utilized by adware and spyware is a winlogon notification package. Winlogon mainly provides the interactive logon prompt, but is able to send event notifications to

registered applications. Winlogon loads these registered DLLs in its address space and when logon or other events occur, winlogon calls the associated export in the notification DLL.

Programs can also be loaded into every running application by adding themselves to the registry key `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs`. When applications are executed, Windows checks this key, loads the DLL in the newly executed application, and calls the entry point of the registered DLL. This causes code in the DLL to run under the context of the application and ensures that code is always executing on the system.

Instead of being loaded in every running application, many adware and spyware programs target Internet Explorer (IE) specifically. Internet Explorer provides a framework for plug-ins known as toolbars and browser helper objects. Both are COM (Component Object Model) objects. COM is Microsoft's framework for reusable binary code. IE toolbars and browser helper objects are loaded by IE and run under the context of Internet Explorer.

When Internet Explorer starts, it iterates the registry for toolbars and browser helper objects and invokes them. The toolbar and browser helper in return can register themselves to receive event notifications from IE, such as a website being visited. Thus, a toolbar and BHO allows a program not only to start itself each time IE is executed, but also harness and gather data from IE.

Instead of loading with Internet Explorer, which only provides notification of browsing with IE, one can also load any time Winsock is started. Applications that perform network actions need to load Winsock. Winsock provides a plug-in architecture known as layered service providers that allow applications to insert themselves into Winsock. This allows them to start any time a network aware application is started and view all network traffic.

By using these common load points, adware and spyware not only start automatically, but also run under the context of other applications. This allows them to remain unnoticed and also gives them access applications such as Internet Explorer actions and all network traffic.

Anti-Removal & Stealth Techniques

Once adware and spyware has been installed on the system and is able to persist across reboots via a load point, it may employ techniques to prevent its removal. These programs typically protect their processes and threads, registry entries, and files via simple obfuscation to rootkit techniques that are otherwise seen only in malicious software.

Simple obfuscation

Many adware and spyware programs use random filenames and registry keys to avoid being noticed.

Others are even more devious and use filenames and registry names that are similar to legitimate processes to further confuse a user. For example, Elitebar will copy itself to the Windows system directory as `win<3 random letters>32.exe` rather than creating its own program directory in Program Files and using a filename that represents Elitebar.

FindWhatever toolbar creates the file svchost.exe in the Windows directory, whereas the legitimate svchost.exe file exists in the Windows system directory. This technique was also used by the Welchia worm.

Watchdog techniques

While obfuscation techniques can trick the average user, they do not impede security products from removing unwanted programs. Adware and spyware programs attempt to impede security products by using watchdog techniques. Watchdog techniques monitor the system and allow a program to reinstall itself if it is removed.

Files

Many adware and spyware programs monitor their own files and simply recreate them if they are missing. Common techniques include simply keeping a copy of the file within themselves or keeping a copy of the file in another location. If the file is deleted, the program just copies the file back. Some programs don't even check if they have been deleted and just reinstall their component files constantly. Instead of storing a second copy on disk, Aurora from Direct Revenue is able to restore itself by storing a copy of itself in memory in explorer.exe and writing that to disk.

Registry

Registry entries can likewise be polled and re-added upon removal. However, a more elegant approach is to utilize Windows registry notifications via the RegNotifyChangeKeyValue API. RegNotifyChangeKeyValue takes in a registry key handle to monitor an event handle. The event handle is signaled if the registry key is modified, as shown in the following example code.

Example 4: Monitoring registry key modifications

```
RegOpenKeyEx(hMainKey, "HKLM\\VirusBtn", 0, KEY_NOTIFY, &hKey);
hEvent = CreateEvent(NULL, TRUE, FALSE, NULL);
RegNotifyChangeKeyValue(hKey, TRUE, dwFilter, hEvent, TRUE);
WaitForSingleObject(hEvent, INFINITE);
...
```

Thus, a program can easily replace a registry key or value after the key or value has been removed by a security product. Look2Me utilizes this technique. This impedes the removal of Look2Me from the

registry. Fortunately, other methods exist to remove the registry key that does not trigger a notification.

Processes

Adware and spyware programs consider their processes to be their most important component since, once they are terminated, they are no longer providing data and can no longer protect their other components such as files and registry keys. A watchdog process monitors its own process and restarts the process if it is terminated. This prevents one from stopping and removing the application.

A naïve approach is to start a program twice and launch a thread that iterates the process list continually in search of the other copy. If the other copy is not found in the process list, the process is restarted.

Similar to the registry, a more elegant approach is to receive notification that the twin process has terminated rather than polling the process list constantly. WebSearch is a toolbar that employs a watchdog process in this manner. The application makes a copy of itself and both copies are executing in memory. Each process simply monitors the other process by using WaitForSingleObject on the handle of the other process. When WaitForSingleObject returns, it means the process has terminated. If this occurs, the process simply restarts the other process.

Often such watchdog techniques are not implemented by the same process, but by injecting the watchdog code into another existing process on the system. Typically, the host process is a process that is always actively running on the system. For example, explorer.exe is a common target. This ensures the watchdog code is always active and hides its presence as well.

Code injection

The ability to inject code into another process provides the program with a method to remain stealth, remain actively running, and gives it access to other processes' address space. This section details the common methods adware and spyware programs use to run under the context of another process and the benefits of doing so.

Direct memory write

Processes in Windows have the ability to access the memory space of other running processes. A process can read and write to the memory space of other processes as well as execute code in those processes. Malicious threats have used this technique to bypass security products such as desktop firewalls, to hide their presence from users, and to ensure they are always running. The email worm Lovgate used this technique to inject code that opened a backdoor listening port under the operating system process lsass.exe.

Adware and spyware use this technique for similar purposes. Aurora from Direct Revenue injects code into explorer.exe. The injected code checks the registry key
`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows`

NT\CurrentVersion\Winlogon\Shell constantly to see if it is set to 'explorer.exe <Aurora filename>'. This registry key is a load point and ensures Aurora runs every time Windows starts. In addition, Aurora injects a copy of itself into the memory space of explorer.exe and will replace itself on disk if it is removed.

This technique hides Aurora's actions in explorer.exe and prevents a user from easily removing Aurora from their system.

Application initialization registry key

Code injection can be also be achieved by having Windows load and execute code by modifying particular registry entries. Windows has a feature to cause all applications that are running in the current log on session to load a DLL when the application is started. This DLL can be defined in the registry key HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs. The DLL is loaded into each running application via LoadLibrary. Thus, the DLL entry point is executed under the context of each application.

Elkern, a virus dropped by the mass-mailing email worm Klez used this technique to ensure it was always executing on the system and adware and spyware have copied this technique as well.

SuperSpider uses this technique to ensure that its process is always running by restarting itself when its AppInit DLL entry point is called. In addition, in order to delete SuperSpider's DLL, one would need to terminate every running application because SuperSpider's DLL is loaded into every running application. In general, this would not be possible. Alternatively, one could remove the registry entry and reboot the system. Unfortunately, SuperSpider will rewrite the registry entry before the system is rebooted.

Windows hook

Windows also allows applications to add a hook into the system message handling mechanism of Windows. This hook can monitor Windows message traffic before the message reaches the target window procedure. In particular, the hook can be added to intercept messages for all the threads in the same desktop. The hook procedure must be implemented in a DLL and this DLL is effectively injected into the address space of other running processes. Since a Windows hook receives all system messages, it is commonly used by malicious keyboard loggers, but can also be used to persist on the system.

VirtuMonde takes advantage of a Windows hook to restart itself if it is terminated. Normally, when stopping an application using ExitProcess, all loaded DLL module entry points are called with the DLL_PROCESS_DETACH notification. This allows the DLL to clean up before being unloaded. However, when calling TerminateProcess all threads are suspended immediately, DLLs are not notified, and the process is killed. This allows system software to end programs and not allow them to prevent termination.

However, an undocumented action occurs when the process being terminated has installed a system-wide

Windows hook. When calling TerminateProcess on a process that has called SetWindowsHookEx, the DLL with the Windows hook procedure receives a DLL_PROCESS_DETACH notification contrary to the normal action of TerminateProcess. This notification is designed to give the DLL the opportunity to remove the Windows hook because removing the DLL from memory, but not removing the Windows hook would likely lead to a system crash. Unfortunately, adware and spyware use this notification to launch themselves again before being terminated.

VirtuMonde simply implements an empty Windows hook procedure that just calls the next hook in the chain via CallNextHookEx. VirtuMonde only utilizes the Windows hook to receive the DLL_PROCESS_DETACH message when being terminated and restarts itself via CreateProcess.

Winlogon notification packages

Winlogon is the component that provides interactive logon into Microsoft Windows. Winlogon provides technology for DLLs to be loaded in winlogon and receive events generated by Winlogon. Winlogon provides notification of system startup, shutdown, logon, logoff, workstation locking, workstation unlocking, when the screensaver starts, when the screensaver stops, and when the shell (explorer.exe) starts.

When winlogon starts, it checks the registry key HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify for subkeys. Each subkey represents a notification DLL and within each subkey is a set of values that specify the DLL name and the function names of the implemented event handlers.

Look2Me utilizes a winlogon notification DLL to receive logon, logoff, and shutdown notifications from winlogon. Each time these notifications are received, Look2Me ensures that it is still properly installed on the system. In combination with additional techniques, this prevents security software from removing Look2Me. For example, if references of Look2Me are removed from the registry, as soon as one reboots, Look2Me will add its entries again.

Since a winlogon notification package is a DLL that is running under the context of winlogon.exe, one can not simply terminate winlogon.exe. Terminating winlogon.exe is likely to result in an immediate system reboot. Suspending the threads of Look2Me is an option, except Look2Me has implemented another technique to prevent its threads from being suspended.

Permission modification

In addition to hooking the system, adware and spyware can avoid removal by preventing other applications from analyzing them and removing them by modifying account and object permissions on the system.

File locking

Files can be protected from being read or scanned by a variety of means. A program can simply call CreateFile on itself with the share permissions set to NULL. This prevents other applications from opening the file until it is closed by the program. Consider the following code, which obtains its own filename and then calls CreateFile.

Example 5: Locking a file with CreateFile

```
hHandle = GetModuleHandle(NULL);
GetModuleFileName(hHandle,szFilename,MAX_PATH);
CreateFile(szFilename, GENERIC_READ,
NULL, // NO SHARE PERMS
NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
```

In addition, files can be locked by region using the LockFile API. LockFile gives exclusive access to the calling process to a region of a file. Consider the following code, which locks the entire file.

Example 6: Locking a file by region with CreateFile, LockFile

```
hFile=CreateFile(szFilename, GENERIC_READ, FILE_SHARE_READ, NULL,
OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
dwSizeLo = GetFileSize(hFile,&dwSizeHi);
LockFile(hFile,0,0,dwSizeLo,dwSizeHi);
```

This code allows CreateFile to be called, but does not allow one to call ReadFile on the locked regions.

Object access control lists

Another method of preventing security software from opening a file is modifying the access control list of an object to remove all access from all users. This is achieved by using the SetSecurityInfo or SetNamedSecurityInfo API. Once all access is removed from the object, the object cannot be opened for reading unless the access control list is modified first to allow access.

Account privileges

Another technique used by Look2Me is simply to remove account privileges. The API OpenProcess requires the SE_DEBUG_PRIVILEGE, which is typically held by the Administrator. Removing this privilege from the Administrator prevents the account from utilizing OpenProcess and thus, many security tools from analyzing the process.

One must utilize LsaAddAccountRights to re-add the privilege first. However, these new privileges only take affect after a re-logon. Look2Me is able to remove the privileges again before a re-logon is

performed.

Rootkit techniques

The use of rootkit techniques to hide and prevent removal is an advanced hooking technique used by adware and spyware programs and was originally found in malicious software. Rootkit techniques involve hooking system APIs and modifying the data returned by those APIs. For example, if a program wishes to hide its own file, it may hook the FindFirstFile and the FindNextFile APIs and filter out any results that match its own filename. Such rootkit techniques can be used to both hide and prevent the removal of objects such as files, registry entries, and processes.

Usermode

Elitebar is a toolbar that uses user mode rootkit techniques. Elitebar mainly displays pop-up windows with advertisements. However, Elitebar will also hide its own files, services, and registry keys. This prevents someone and some security software from being able to find and remove Elitebar.

Elitebar uses a DLL patching technique where key functions are patched in memory when the DLL is loaded. In particular, Elitebar patches FindFirstFileExW and FindNextFileW to hide its own file; EnumServiceStatusA and EnumServiceStatusW to hide its services; RegEnumKeyExW and RegEnumKeyExA to hide its registry keys; and LdrGetDllHandle and NtResumeThread to receive notification of newly loaded DLLs and thus, the need to patch them in memory as well.

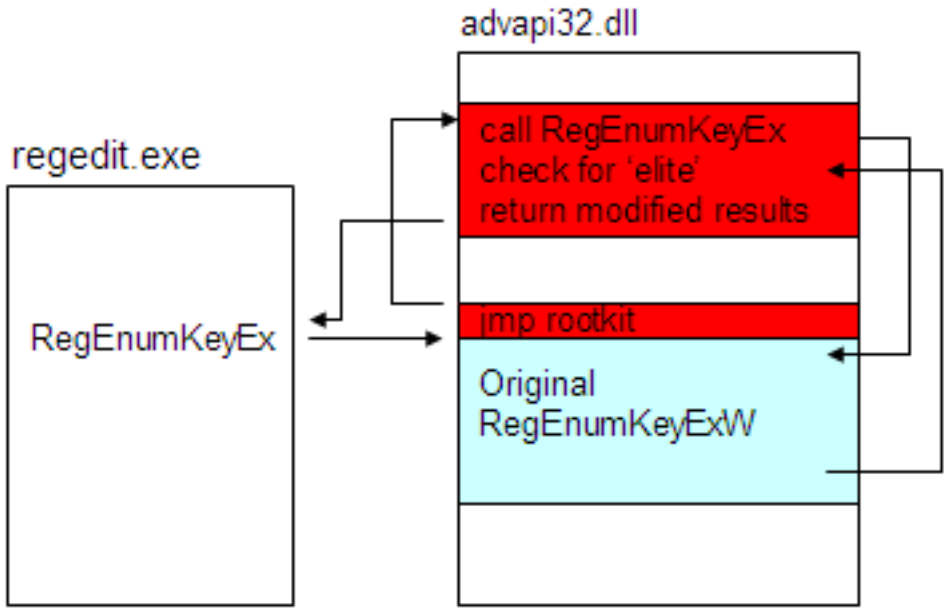

Figure 6: Elitebar uses a DLL patching technique where key functions are patched in memory when the DLL is loaded.

Elitebar enumerates all the current running processes on the system. The memory space of each process is searched for the desired DLLs and APIs. Elitebar injects code into the DLL module memory space just after the PE header overwriting the section table. Then, each desired API is patched in memory replacing the prologue with a jump to Elitebar's code. The injected code contains a routine for each hooked API, which calls the original API and alters the return data. In particular, any result with the string 'elite' is removed.

Kernel mode

Adware and spyware also utilize more robust kernel mode hooks. Many kernel mode techniques can be used to hide and prevent object removal, but most programs using a kernel mode rootkit simply hook the service descriptor table. The service descriptor table is a function pointer table in the kernel that contains all the relevant system APIs. By changing the values in the function pointer table, the program can redirect code execution to itself rather than the intended API.

CommonName contains kernel rootkit functionality that hides registry keys. CommonName installs a driver that patches the service descriptor table intercepting a variety of system APIs. The driver is supplied with a configuration file that contains strings to hide.

In particular, the driver hooks ZwEnumerateValueKey, which is called when viewing the registry. If ZwEnumerate ValueKey returns any strings in the configuration file, the hooked function returns the error code 'STATUS_NO_MORE_ENTRIES' instead. This action has the unfortunate side effect that any registry values that are added afterwards under the same key will not be visible either.

CommonName hides its Run registry key to prevent someone or some security products from uninstalling the program. Rootkit techniques can also actively prevent removal. ISearch installs a driver to prevent one from deleting their files or registry keys by hooking the kernel APIs ZwSetInformation, ZwWriteFile, ZwDeleteKey, ZwDeleteValueKey, ZwSetValueKey, and ZwCreateFile. When one attempts to delete one of ISearch's files or registry entries, ISearch's code is called first. ISearch checks whether the item being deleted belongs to itself and if so, returns access denied, preventing its removal. For adware and spyware, rootkit technology prevents removal by a user via the normal means. Furthermore, rootkit technology also prevents most security software from removing the security risk as well. While more security software products are being designed to be rookit-aware, at this time only a few power user tools are available for finding and removing rootkits with varying degrees of effectiveness.

Data gathering

Of course, adware and spyware programs have a purpose beyond just preventing their removal. Many track user computing habits for market research. This market research is sold to clients or used to display advertisements. The amount and type of data gathered varies, from those that attempt to send no personally-identifiable information, to others that gather and send confidential information to remote

servers.

Aggregate browsing habits

Many security risks gather web browsing habits. Typically, the URL of each website visited is sent to a remote server. With RX Toolbar installed, a visit to the website www.hertz.com, will result in the following request being sent to a remote server:

```
/webrdf/categories2?version=1,0,9,3&url=http://www.hertz.com
```

While the version of RX Toolbar is sent, no unique identifier other than the IP address is sent at that time. In response to the query, the server sends back potentially related links, which are displayed in the toolbar. Selecting one of these links causes an identifier of the link to be sent to the remote server. Presumably this information is used to contribute to aggregate marketing data of popular links.

Computing habit profiles

Other security risks perform similar actions, but also include a unique identifier. For example, GAIN, an advertising network by Claria, utilizes a 'subscriber id' so they can potentially build a profile of unique users. This profile is not limited to browsing habits such as the websites visited, how long each web page is viewed, whether purchases were made, and which advertisements were viewed and clicked, but also geographic information such as the country and city and information such as the first four digits of a credit card number used to make purchases. Additional demographic information can also be bound to the subscriber ID. Some GAIN-supported software may ask for information such as gender and income. If the user provides this information, the information will also be sent along with the subscriber ID to Claria. Furthermore, GAIN also may collect system information such as what software is installed on the system.

System information

BetterInternet is another example of a security risk that collects system information. BetterInternet will send a list of all running processes, registry entries, hostname, Windows serial number and product ID, network card MAC address, Windows version, and other software version information to a remote server.

Example 7: Information sent by BetterInternet

```
<os majorVersion="5" minorVersion="1"
buildNumber="2600" osPlatform="Win32 on Windows NT"
csdVersion=""/>
<ie version="6.0.2600.0000" product="Internet
Explorer 6 (Windows XP)"/>
…
<registryEntry hive="HKLM"
path="SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Browser
```

```
Helper Objects" subtree="yes">
…
<registryKey name="MPlayer2"/>
<registryKey name="NetMeeting"/>
<registryKey name="OutlookExpress"/>
…
</registryEntry>
<procList>
<proc PID="0" exe="[System Process]"/>
<proc PID="4" exe="System"/>
<proc PID="552" exe="smss.exe"/>
<proc PID="600" exe="csrss.exe"/>
…
</procList>
</custom>
</systemInformation>
<install>
…
<serialID value="D411CC54"/>
<productID value="55274-640-1934803-23789"/>
<MACaddress value="000F1FD80EBB"/>
<HostName value="s_d10422"/>
…
```

While system information may not be personally-identifiable, many users are unlikely to approve of third parties knowing which programs they have installed and are actively running. Unfortunately, the type of information being collected is not limited just to system information and aggregate browsing habits.

Personally-identifiable and confidential information

Personally identifiable and other confidential information is also gathered and sent to remote servers by a variety of adware and spyware programs. This information includes first and last names, credit card numbers, username and passwords, confidential transactions, and even instant messaging sessions. An example of a company that collects personally identifiable and confidential information is comScore Networks, Inc. comScore provides marketing data for numerous clients and obtain this data by distributing software known as Marketscore. Marketscore monitors browsing habits and collects information including information that is sent via SSL. Previous versions of Marketscore would proxy all HTTP traffic including HTTPS. Requests from the user would go to comScore first and comScore would then request the data from the destination server and pass the data back to the user. With comScore as man-in-the-middle, they are able to view all proxied network traffic including the HTTPS. Normally, this traffic would be encrypted, but comScore gets around this by using the same technique a malicious

hacker would use in conducting a man-in-the-middle attack – installing a trusted root certificate on the user's machine where HTTPS traffic is encrypted between the user and comScore using comScore's public key and between comScore and the bank using the bank's public key. comScore can then decrypt and view the traffic from the user and re-encrypt it before connecting to the bank on behalf of the user.

Current versions of Marketscore do not proxy, but instead mine outgoing and incoming data locally and send this data to comScore. Marketscore uses a Winsock Layered Service Provider (LSP) to monitor all network traffic. Marketscore monitors browsing habits and sends them to comScore's servers along with other data depending on the page being viewed. If the data received from specific domains contain specific keywords, the received data is then gzipped and MIME-encoded and sent to comScore. The data sent can contain usernames, passwords, credit card numbers, names, addresses, buying habits and includes data that was sent via HTTPS. comScore does not just monitor HTTP and HTTPS, but other protocols as well, such as instant messaging. comScore records when one sends and receives messages and has access to data such as screen names. Since comScore operates as an LSP, it is able to see the content of the instant messages as well, but during testing no instant message bodies were sent to comScore. This data is also bound to a specific user id. This user id is bound to demographic information including gender, age and geographic location, which aids comScore in building market research data. While some users may be willing to expose their non-personally-identifiable browsing habits, others would be uncomfortable with exposing their personal details and specific browsing and purchasing habits, let alone instant messaging conversations. Actions that range from exposing confidential information to the general nuisance factors of popping up advertising every five seconds results in consumers requesting their anti-virus products remove these programs. In addition to the legal quagmire of removing these programs, even well-behaved programs can modify the system so it cannot be restored to its previous state.

Removal

Most anti-virus products were designed originally to remove viruses and contain engines to modify files rather than to deal with system modifications such as new directories, created files, modified registry entries and running processes. Removing adware and spyware programs properly requires this type of full system repair and many vendors have updated their existing products as a result. Unfortunately, many of the system modifications made by adware and spyware programs are indirect, or else previous values are not stored and thus reverting back to previous settings may not always be possible.

For example, if adware modifies the Internet Explorer homepage, a security product may not know the previous value. Some solutions involve either changing the page to the default value, to a blank page, to a page which explains how to set the home page again, or to a user-defined value. Even knowing the previous values is problematic, especially in a situation where a system has many adware or spyware programs installed, all of which have changed the homepage setting. System modifications are not reserved to potentially benign settings such as the IE homepage, but can include other settings that are

critical to the operating system or security settings. Reverting these settings can lead to a loss of network connectivity or a system crash. Worse, a partial repair can render the system completely useless. For example, Windows layered server providers are placed in a chain with each LSP calling down to the LSP in the next layer. If one deletes the LSP file, but does not deregister the LSP from the chain, network connectivity will fail. Other similar changes can prevent the system from even booting, such as deleting a rogue GINA (Graphical Identification and Authentication) DLL.

Furthermore, partial repair forces other security tools to scan for remnant components, and vice versa, more aggressive products may perform a full repair and remove common components being used by other legitimate software installed on the system. At a minimum, security products need to be able to remove all files that have an associated load point and the load point itself. This includes files referenced in the Run registry key, browser helper objects, toolbars, and LSPs. While Windows has many load points, they are finite. Smart security products can employ a level of generic removal without removal signatures using only detection signatures and then scanning the default load points and removing them accordingly. Ideally, other side effects would also be removed, but as discussed previously, the original state of those objects may not be known or the original state may not even be appropriate any more. Thus, proper removal is still subjective today and many removal tests that simply rescan with another product to see what the tested product left behind aren't necessarily good indicators. Removing everything is easy, but may not leave the system in a better state.

Conclusion

The actions of many adware and spyware programs go beyond simply facilitating advertisements or gathering aggregate non-personally-identifiable data. Many adware and spyware programs use techniques akin to malicious threats from social engineering to exploiting vulnerabilities. Once installed on the system, they use techniques to hide themselves and prevent their removal. In addition, many adware and spyware programs gather personally-identifiable and confidential data and are able to correlate that data continually to build marketing profiles. The extent of the data is so large that many governments would probably love to have such a system.

While many adware and spyware vendors continue to paint themselves in a glowing light, few are well-behaved and many employ techniques commonly seen in malicious code. Security vendors need to examine behaviors of the software in addition to the general purpose of the software. Even if the general purpose of the software is to facilitate advertising, if it utilizes a malicious technique such as self-replication, the software should be categorized appropriately. Nevertheless, in the end consumers are demanding that their security products prevent these programs from being installed and that upon being installed, they have the ability to remove them properly.

References

[1] W32.Friendgreet.Worm,
http://securityresponse.symantec.com/avcenter/venc/data/w32.friendgreet.worm.html
[2] 2005 Spyware Study, Ponemon Institute, http://www.ponemon.org/.
[3] Personal communication with Glenn Jarvis.
[4] http://www.vitalsecurity.org/2005/06/aurora-installsource-revealed-and-175.html.
[5] Testimony of Ari Schwartz, The Senate Committee on Commerce, Science and Transportation on 'Spyware'.
[6] Federal Trade Commission vs Seismic Entertainment Complaint for Injunction and Other Equitable Relief, District of New Hampshire.
[7] Federal Trade Commission vs Seismic Entertainment Memorandum in Support of Plaintiff's Motion for a Temporary Restraining Order, District of New Hampshire.
[8] http://www.microsoft.com/technet/security/bulletin/MS04-013.mspx.
[9] http://www.microsoft.com/technet/security/Bulletin/MS05-002.mspx.
[10] http://www.microsoft.com/technet/security/bulletin/MS03-011.mspx.
[11] http://www.gainpublishing.com/help/privacy_statement.html.
[12] http://www.cit.cornell.edu/computer/security/marketscore/technical.html.
[13] Shevchenko, Sergei; 'Standing the privilege attack', Virus Bulletin, June 2005, p.4.
[14] Personal communication with Stephen Doherty, John
Park, Paul Mangan, Rowan Gallagher, Sean Kiernan.

Appendices

Appendix A

MD5s of samples analyzed.

IFrameDollars
adv510.php ef14f5df0cef41044190e497beecd218
sploit.anr fa13e863b5dbd74efee3f6031114372f
x.chm 77e491093722dade0df0515978c93585
loaderadv510.jar a74c2e2d1deba87b5c640c1b82ffe369
loadadv510.exe 75f1d7592f7e57614608dab64edf37de
Elitebar
protector_update.exe
39eb6705ef3936d61e168abde344bf98
CommonName
winik.sys 935055fff770f7192e79a0c558d7c2cb
ISearch
Delprot.sys 83564be28b3087346be74d28164934c7
VirtuMonde
Killhook.dll ced130e0e301e7e0110ed5ad838d2d34
SuperSpider
18c3cd089329c8315cedc9da0d857c5d
WebSearch
Pib.exe 872ca0e9ff6d69fb592143e7dabdde7c
MarketScore
opls.dll 6be86e215d9819d66ba98f62910f9209
opnsqr.exe fd959de3f5f3994f88e1d943f7a1b872
GAIN
GMT.exe 7474487f6cce10f730916a677c35a6b5
BetterInternet
27fc61e214be9724a407e9d68aadb40a
Aurora
Nail.exe d959377938f29d91ca1cd533fea2efbb
FindWhatever
7efb8953848d1f6cdb3b59eeabec87f9

Appendix B

Example emails soliciting bundled installs.

Hi, My name is <omitted> and I am a Sr. Media Planner for Vista Interactive Media. I came across your site and was interested in the various game products you have listed for download. My company has several applications we are looking to distribute across users desktops. We would pay you per install; and provide you with your own login to our system so you can view your revenue in real time. With the amount of downloads you are getting monthly, this could be a large source of additional money, while adding your users valuable tools to navigate the web better.
Here are the products I'm looking to distribute and the prices I can pay for each one.
1.) Mega Search toolbar w/ Highlighter feature
-$0.10 per install
-www.megasearchbar.com
-This toolbar offers a search box, as well as a build in pop up blocker and text highlighter to show relevant content on a users web page.
2.) Instafinder
-$0.04 per install
-www.instafinder.com
-This product will take dead URL pages and return relevant content as a user surfs the web. For instance, if someone was looking for Car insurance, and they incorrectly typed the word insurance in the web browser, they would be taken to a page full of potential sponsors instead of seeing a 404-Error page inside IE.
3.) Contextual Partner
-$0.11 per install
-This product will display various pop up ads as users surf the web If you include all 3 applications I can pay you $0.25 per install.
Payments are sent @ the end of every month via Wire, Paypal or Check. If you run our bundle for 1-day and aren't completely satisfied with results you can remove our .exe. We also provide you with login/password to view your installs and Revenue online @ www.vistainteractivemedia.com/cobundle.
You have nothing to lose, and everything to gain by giving our products a test. Please contact me @ your earliest convenience to continue discussing this option.
<omitted>
P.S All our products contain remove links inside Windows Add/Remove, and they've been submitted for the approval of C.O.A.S.T. (http://www.coastinfo.org/index.htm)
<omitted>
Sr. Media Buyer
Vista Interactive Media
<omitted>
5 Corporate Park Rd, Suite 160

Irvine, CA 92606
<omitted>

Hi, my name is <omitted>. I work for 180Solutions, a company that generates revenue for software developers like yourself. I am contacting you because I noticed that you work with <omitted> to distribute your software <omitted>. 180Solutions has developed n-CASE, which is the most widely distributed client-side comparison shopping engine in the world. Not only is n-Case
distributed all over the world, it is the easiest comparison shopping engine to use. We work with dozens of software developers just like you that bundle and distribute n-CASE with their existing software applications. I would like to offer you the opportunity to bundle n-CASE with your software, and we will pay you $.07 for every one of your users that opts-in to installing n-CASE. Typically, our distribution partners bundle n-CASE in their existing setup programs or as a part of an upgrade, and then give their users the option to install n-CASE during setup. Currently over 60% of users when prompted, opt to install n-CASE. If you are interested in developing another source of revenue, please feel free to contact me or follow this link: <omitted> I look forward to hearing from you.

About the Author

Eric Chien joined Symantec Corporation at the Symantec Security Response headquarters in Santa Monica, California in 1997. Eric graduated from the University of California, Los Angeles with a Bachelor of Science degree in Electrical Engineering and Molecular Genetics. Currently, Eric heads research in the Europe, Middle East, and Africa (EMEA) regions, analyzing current virus threats and researching new threats in the world of viruses and malicious software.

He has been a key developer in projects such as the Digital Immune System (DIS), Symantec's automated system of virus analysis, and the Seeker project, which proactively finds viruses on the Internet. Eric has spoken at various conferences and published a variety of papers addressing threats to computer security via malicious software.

About Symantec

Symantec is the global leader in information security, providing a broad range of software, appliances, and services designed to help individuals, small and mid-sized businesses, and large enterprises secure and manage their IT infrastructure. Symantec's Norton™ brand of products is the worldwide leader in consumer security and problem-solving solutions. Headquartered in Cupertino, California, Symantec has operations in 35 countries. More information is available at www.symantec.com.

Symantec has worldwide operations in 35 countries. For specific country offices and contact numbers, please visit our Web site. For product information in the U.S., call toll-free 1 800 745 6054.

Symantec Corporation
World Headquarters
20330 Stevens Creek Boulevard
Cupertino, CA 95014 USA
408 517 8000
800 721 3934
www.symantec.com

04/05 10406630