The Dark side of the Internet: Introduction to Malware

By: Brian Pohlman

Master of Science in Information Security

Lewis University

December 2009

(Page intentionally left blank)

ABSTRACT

The Internet can now be accessed almost anywhere by various means, especially through mobile devices. These devices allow users to connect to the Internet from anywhere there is a wireless network supporting that device's technology. Services of the Internet, including email and the web, may be available. The Internet has also become a large market for companies [1]; some of the biggest companies today have grown by taking advantage of the efficient nature of low-cost advertising and commerce through the Internet. It is the fastest way to spread information to a vast number of people simultaneously. The Internet has also revolutionized shopping. For example, a person can order a CD online and receive it in the mail within a couple of days, or they can download it directly and have it in a matter of minutes. The Internet has also greatly facilitated personalized marketing which allows a company to market a product to a specific person or a specific group of people more so than any other advertising medium. Examples of personalized marketing include online communities such as MySpace, Facebook, and Twitter. The low-cost and nearly instantaneous sharing of ideas, knowledge, and skills has made collaborative work much easier. Not only can a group cheaply communicate and share ideas, but the wide reach of the Internet allows such groups to easily form in the first place. The Internet also allows users to remotely access other computers and information easily, wherever they may be located in the world. They may do this with or without the use of security, authentication, and encryption technologies, depending on the requirements [2]. This encourages new ways of working from home or while on the road. While the Internet is a fantastic place to shop, social network, and work from home there is a much darker side to the Internet that the average person rarely sees, until it is too late that is. And while they realize

their PC may have been infected with malicious software, otherwise known as malware, they may have no idea of how it got there and seemingly how easily it installed on their PC in the first place. In this paper, I will attempt to explain what malware is as well as its purpose. I will then show how simple it can be to become infected and actually download the malware for the purpose of reverse engineering. I will walkthrough using tools and techniques needed to breakdown malware to demonstrate how it acts on a live system. This will be done within the confines of a locked down virtual machine built specifically for this type of analysis.

Introduction to Malware

In short, malware can be defined as any unintended and unsolicited installation of software on a system without the user knowing or wanting it [10]. Malware has become the greatest external threat to most systems, causing damage and requiring extensive recovery efforts within most organizations [3]. Malware is divided into the following major categories:

- **Viruses.** A virus self-replicates by inserting copies of itself into host programs or data files. Viruses are often triggered through user interaction, such as opening a file or running a program. Viruses can be divided into the following two subcategories:
    - **Compiled Viruses.** A compiled virus is executed by the operating system. Types of compiled viruses include file infector viruses, which attach themselves to executable programs; boot sector viruses, which infect the master boot records of hard drives; and multipartite viruses, which combine the characteristics of file infector and boot sector viruses [3].
    - **Interpreted Viruses.** Interpreted viruses are executed by an application. An example of this would be a macro virus. Macro viruses take advantage of the capabilities of the applications' macro programming language to infect the application documents as well as the document templates [3].
- **Worms.** A worm is a self-replicating program that usually executes itself without user intervention. Worms are divided into two categories:

  - **Network.** A network worm takes advantage of vulnerabilities in a network service to propagate itself and infect other systems.
  - **Mass Mailing.** A mass mailing worm is similar to an email virus but is self-contained, rather than infecting an existing file [3].
- **Trojan horse.** A Trojan horse is a malicious piece of software that performs an evil deed on behalf of an attacker without the user knowing it is there. As the name implies, some Trojan horses make their way onto a system embedded within another piece of software. Pirated software has been known to contain Trojan horses.
- **Blended Attacks.** A blended attack uses multiple infection or transmission methods. For example, a blended attack could possibly combine the methods of viruses and worms [3].
- **Tracking Cookies.** A tracking cookie is a persistent cookie that is accessed by many web sites, allowing a third party to create a profile of the user's surfing behavior. Tracking cookies are often used in conjunction with web bugs, which are tiny graphics on web sites that are within the HTML content of a Web page or e-mail.
- **Attacker Tools.** Attacker tools can be delivered to a system and used as part of a malware infection or other system compromise. Some popular types include:
  - **Backdoors.** A backdoor is a malicious program that listens for commands on a specified TCP or UDP port. Most backdoors allow an attacker to

perform a specified set of actions on a system, such as acquiring passwords or executing commands.

    - **Key Loggers.** A key logger monitors and records keystrokes made at a keyboard. The keystrokes can be written to a log file which the attacker can then use to discern the username and password used for websites, important documents, etc.
    - **Rootkits.** A rootkit commonly refers to a piece of malicious software that hides itself from system administrators in order to perform some sort of evil task. A good rootkit will survive reboots and hide processes, files, registry entries, and network connections.
    - **Browser Helper Object.** A Browser Helper Object (BHO) is designed as a plug-in for Microsoft's Internet Explorer to provide added functionality. Attackers create malicious BHO's that act as spyware. When installed, the BHO can monitor all uses of the browser and report this information back to the attacker.
- **Attacker Toolkits.** While not necessarily considered malware, many attackers use a wide variety of tools in order to monitor and attack systems. Once a system has been compromised through malware or some other means, an attacker may load a toolkit on the system. The toolkit can aid in further compromising the system or be used to attack other systems. The following is a list of programs that may be found in an attacker toolkit [3]:

- **Packet Sniffer.** Sniffers are designed to monitor network traffic on a wired or wireless interface. They can typically be used to decode communications and steal data.
- **Port Scanner.** A port scanner scans remote systems to determine which ports are open. This is a way to determine potential targets.
- **Vulnerability Scanner.** A vulnerability scanner is a program that looks for vulnerabilities on either local or remote systems. This is also another way to identify potential targets.
- **Password Cracker.** A password cracker is generally used to crack operating system or application passwords using the brute force technique. By doing this, an attacker can gain root or administrator privileges.
- **Remote Login Programs.** A remote login program is typically used by a system administrator to log in and have full control of another system while working remotely. An attacker could abuse this method and use a remote login client to control compromised systems and transfer data back and forth between them.

History of Malware

Now that we have an idea of what malware is, I would like to take a step back and look at the history of malware. The idea of a computer virus first came about in the early days of computing. The earliest viruses were created simply just as benign pranks. Malicious viruses did not come about until the early 80’s. The first worms, created in the 1970s, were also benign and intended to aid administrators with system maintenance. Malware became common during the 1980s with the most common form being compiled viruses, specifically boot sector viruses. In 1988, the Morris work was released, disrupting thousands of networked computers. Trojan horses also begin to surface in the mid-1980s [3].

During the early 1990s, the malware situation remained the same with compiled viruses continuing to the most common form of malicious code. However, toward the end of the 1990s, several important changes in computing allowed new opportunities for malware. The first factor was that the number of personal computers increased greatly. In addition, the use of e-mail and software with macro languages such as word processors and spreadsheets became widespread. Virus writers began creating viruses and spreading them through e-mail, as well as creating self-contained worms with similar capabilities. Two malware attacks, the Melissa virus (1999) and the LoveLetter worm (2000), each affected millions of systems. Trojan horse and RAT (Remote Access Trojan) combinations, such as BackOrifice and NetBus, also became popular in the late 1990s [3].

Since 2000, worms have been the most common form of malware. Worms are often favored by virus writers because they can spread very quickly. Boot sector viruses have become a thing of the past because of the declining usage of the floppy disk. In 2001, the first major blended attack, Nimda, was released causing a large disruption. Nimda had characteristics of viruses, worms, and malicious mobile code. It was so effective at the time because, unlike other malware, it used five different infection methods:

- via email
- via network shares
- via browsing of compromised web sites
- exploitation of various Microsoft IIS 4.0/5.0 directory traversal vulnerabilities
- via back doors left behind by the Code Red II and sadmind/IIS worms

Recently, mobile code attacks have become more common, mainly because of the prevalence of web browsers and HTML-based e-mail. Another trend is that more instances of malware, including worms, Trojan horses, and malicious mobile code, deliver attacker tools, such as rootkits, keyloggers, and backdoors, to infected systems [3].

Malware Capture Methods

There are a few different ways to get a hold of malware if you plan to do analysis. The first and easiest is to simply use the Internet to track down websites that contain live samples. http://www.offensivecomputing.net is one of the more well known sites that contain live samples of malware [4]. Simply register for a free account and you can begin studying the malware posted there. There are also a handful of sites just like the one mentioned that serve the same purpose. Also, there are many underground hacking sites full of users who are willing to trade malware. This would be another option. Another way of potentially collecting malware is to “spam yourself.” Sign up for multiple email addresses through Yahoo!, Gmail, Hotmail, or any of your liking and you are likely to fall into the claws of a spammer.

The most in depth method of collecting malware is to utilize honeypots. Honeypots are unique in that they don’t solve a particular security problem. Instead, they are highly flexible tools with many different information security supplications [5]. This contrasts with technologies such as firewalls and intrusion detection systems, which are easier to define as they solve specific problems. Firewalls are a prevention technology; they are host or network solutions which keep attackers out. IDSs are a detection technology; an IDS serves the purpose if detecting and alerting security professionals of malicious activity. Honeypots are harder to define because they can be involved in aspects of prevention, detection, information gathering, and much more. The Honeypot mail list developed the definition of a honeypot and it as follows:

> *“A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource”* [5]

As the Honeypot mail list says, “The definition was difficult to develop, as honeypots can come in so many different shapes and sizes. As a result, the definition is very broad in scope, as it has to cover many different applications of honeypots. The definition of a honeypot does not indicate how a honeypot works nor what its purpose is. Instead, its definition refers to how a honeypot generates its value” [5]. Simply put, honeypots are a technology whose value depends on the bad guys interacting with it. All honeypots work on the same concept: Nobody should be using or interacting with them. Any transactions or interaction with a honeypot are by definition unauthorized.

It should be noted that honeypots do not replace other traditional security systems; they are just an additional layer to overall security of the network. Honeypots can be setup externally, internally, in the DMZ, or even in all of the locations. Figure 1 shows a honeypot setup in all three locations:

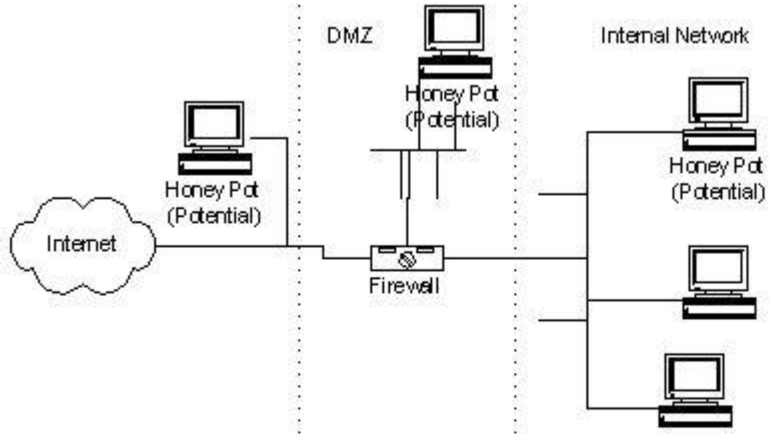


Figure 1. [13] Honeypot topology

Again, the above setup is just an example and can be scaled to fit the requirements of the network. Honeypots are similar to an Intrusion Detection System (IDS) but with a focus on gathering information as well as deception.

Malware Analysis

Once you have a piece of malware that you want to analyze, the first question you should ask yourself is, “Where will the analysis be done?” The answer that usually comes to mind is VMware. VMware allows for the simulation of multiple computers running simultaneously on a single physical system [6]. There are several advantages to adopting this route for malware analysis, compared to building a full blown lab made of up physical infrastructure components:

- It is very beneficial to have multiple systems running so malware can interact with components of the simulated network. With VMware, it's possible to build a multi-component lab without needing multiple machines.
- A great feature of VMware is its ability to take s snapshot of the system's state before infecting it and talking periodic snapshots throughout the analysis to save time. This allows for an easy means of reverting back to the desired system state almost instantly.
- VMware's host-only networking option is convenient for connecting virtual systems using a simulated network without additional hardware. The host-only network allows any virtual system to see all traffic on the simulated network when listening in promiscuous mode. This makes monitoring the specimen's network interactions easy.

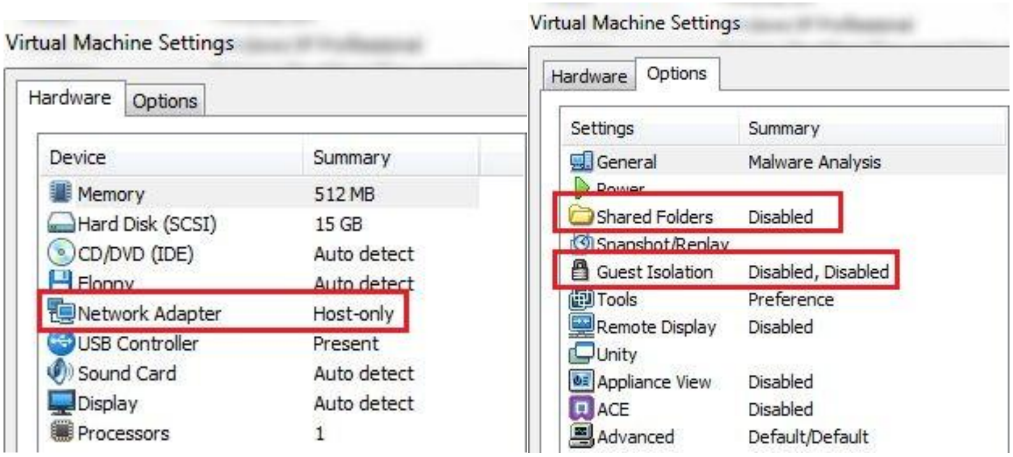
Preparing a VMware-based analysis lab is relatively simple. You need a system with plenty of RAM and disk space that will act as the physical host. You also need the necessary software: VMware Workstation (not free but you can use free of charge for 30 days) and the installation media for the OS you want to run the analysis on. The specifications of the machine I'm using are 3 GHz processor, 2GB of RAM and a 320GB hard disk. The OS used by the virtual machine will be a fully patched Windows XP SP3. I've allocated 512MB of ram and 15GB to the virtual OS, which is plenty for this analysis.

Now that the virtual machine has been created, the necessary tools will need to be added for the analysis phase. It's worth noting that not all of these tools will be used but worth having as some have features that others do not. These are ones that I have worked with and have a good handle on what benefits they provide. Feel free to add any tools that you feel will assist best in the analysis. Below is a list of tools I chose and a brief description of what they do:

- Tcpview – shows detailed listing of all TCP and UDP endpoints on the system, including the owning process name, remote address and state of TCP connections
- Process Explorer – Shows detailed information about a process including its icon, command-line, full image path, memory statistics, user account, security attributes, and more
- Psfile – Shows a list of files on a system that are opened remotely
- HookExplorer – Designed to scan a process looking for IAT or detours style hooks
- PEiD – The PE format is a structure that contains the necessary information for the Windows OS loader to manage the wrapped executable code. PEiD can detect most of the common packers and compilers for the PE file
- RegShot – Allows you to take a snapshot of the system registry then compare it against a second one – done after system changes or installing a new product
- InstallRite – Allows you to see all changes during a software installation process
- Upx – High-performance executable packer
- GMER – Rootkit detector and remover
- OllyDBG – 32-bit level analyzing debugger
- IDA Pro – Multi-processor disassemble and debugger

- FileInsight – Integrated tool environment for web site and file analysis
- Malcode Analyst Pack – A series of utilities that assist in doing malcode analysis
- SysAnalyzer – Automated malcode run time analysis application that monitors various aspects of system and process states
- Wireshark – Packet sniffer
- ProcNetMonitor – Tool that monitors network activity of all running processes in the system
- FileAlyzer – Allows basic analysis of files, showing file properties and file contents in hex dump form. Is also able to interpret common file structures like text, graphics, HTML, media, and PE

Now that we have the virtual machine built and malware analysis tools installed, we need to take a look at and address two concerns when using VMware to analyze malware. The first concern being since the virtual machine runs on the host, there is a chance of cross contamination. In order to minimize the chances of our host being infected, three setting changes to contain malware to the virtual machine are made. The first change is to set VMware's virtual network adapter to ***Host-only.*** Host-only networking provides a network connection between the virtual machine and the host computer. This will be useful in isolating the malware to our virtual machine. The second setting change is to disable ***Shared Folders***. Shared folders allow the virtual machine and host computer to easily share files amongst one another. By disabling this feature, malware will not be able to spread via open shares. The third and final setting change is to disable ***Guest Isolation***. By having Guest isolation enabled, it will allow dragging and dropping as well as copying and pasting between the virtual machine and host computer. This will be disabled to prevent any possibility of malware spreading via these means. Refer to **Figure 1** below to see how to set this up.

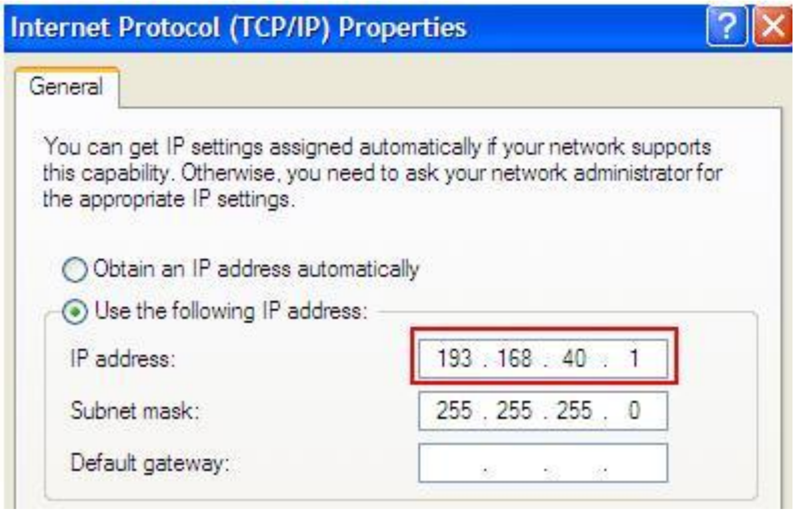


Device	Summary
Memory	512 MB
Hard Disk (SCSI)	15 GB
CD/DVD (IDE)	Auto detect
Floppy	Auto detect
Network Adapter	Host-only
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect
Processors	1

Settings	Summary
General	Malware Analysis
Power	
Shared Folders	Disabled
Snapshot/Replay	
Guest Isolation	Disabled, Disabled
Tools	Preference
Remote Display	Disabled
Unity	
Appliance View	Disabled
ACE	Disabled
Advanced	Default/Default

Figure 2. VMware Lockdown Screenshot

As another precaution, it is recommended to assign an invalid IP address to the virtual machine. This precaution will protect the host system from becoming infected while allowing network activity to be monitored. To set an invalid IP address, go to ***Network Connections*** → right-click on VMware Network Adapter VMnet1 and select ***Properties***. Highlight ***Internet Protocol (TCP/IP)*** and select ***Properties***. Select ***Use the following IP address*** radio button and enter in an invalid IP address. See **Figure 3** below for details:



Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 193 . 168 . 40 . 1

Subnet mask: 255 . 255 . 255 . 0

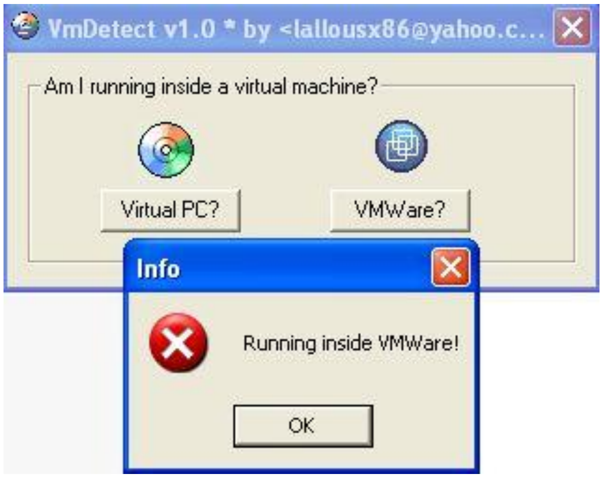
Default gateway: . . .

Figure 3. Invalid IP address assigned

The second concern to address is that of VMware detection technologies. Attackers are always looking for ways to detect VMware and other virtualization technologies. By utilizing virtualization detecting techniques, attackers can create malware that impossible to reverse engineer within a virtual machine. This means that the malware would need to be reverse engineered on a live system, increasing the chances of an outbreak. Below are tools that a hacker can integrate into their malware to detect whether or not virtualization is being used and essentially render malware analysis in a virtual environment impossible:

- redPill – Stored Interrupt Descriptor Table (SIDT) command retrieves the Interrupt Descriptor Table (IDT) address and analyzes the address to determine whether VMware is used
- Scoopy – Builds on SIDT/IDT technique of redPill by checking the Global Descriptor Table (GDT) and the Local Descriptor Table (LDT) address to verify the results of redPill

- Doo – Included with Scoopy tool, it checks for clues in the registry keys, drivers, and other differences between the VMware hardware and real hardware
- Jerry – Some of the normal x86 instruction set is overridden by VMware and slight differences can be detected by checking the expected result of normal instruction with the actual result
- VmDetect – VirtualPC introduces instructions to the x86 instruction set. VMware uses existing instructions that are privileged. VmDetect uses techniques to see if either of these situations exists. This is the most effective method. Here is what happens when VmDetect is run inside the virtual machine I just setup:



VmDetect v1.0 * by <lallousx86@yahoo.c...

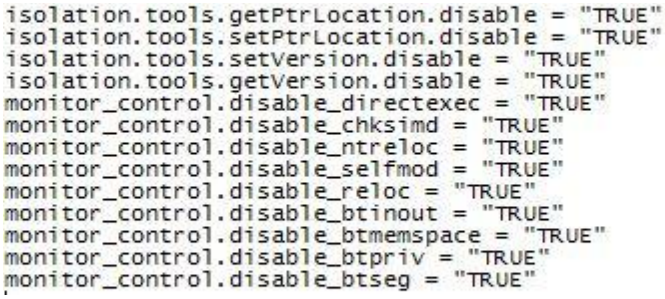
Am I running inside a virtual machine?

Info

Running inside VMWare!

Figure 4. VmDetect Screenshot 1

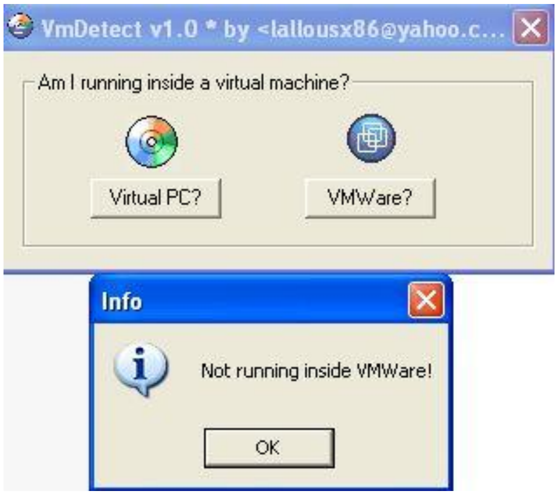
An attacker could create a VmDetect module for their malware to nullify any chance of it being reverse engineered within a virtual machine. Once the malware executes on the system, the module determines that it's running inside a virtual machine and stops the malware from executing any further, making it impossible for live analysis to take place. Fortunately, there are undocumented features in VMware that are effective at eliminating the most commonly used signatures of a virtual environment. I added the following lines to the .vmx file of my halted virtual machine [7]:



```
isolation.tools.getPtrLocation.disable = "TRUE"
isolation.tools.setPtrLocation.disable = "TRUE"
isolation.tools.setVersion.disable = "TRUE"
isolation.tools.getVersion.disable = "TRUE"
monitor_control.disable_directexec = "TRUE"
monitor_control.disable_chksimd = "TRUE"
monitor_control.disable_ntreloc = "TRUE"
monitor_control.disable_selfmod = "TRUE"
monitor_control.disable_reloc = "TRUE"
monitor_control.disable_btinout = "TRUE"
monitor_control.disable_btmemspace = "TRUE"
monitor_control.disable_btpriv = "TRUE"
monitor_control.disable_btseg = "TRUE"
```

Figure 5. Lines added to .vmx file

Figure 6 now shows VmDetect not detecting the presence of a virtual machine:



VmDetect v1.0 * by <lallousx86@yahoo.c...

Am I running inside a virtual machine?

Not running inside VMWare!

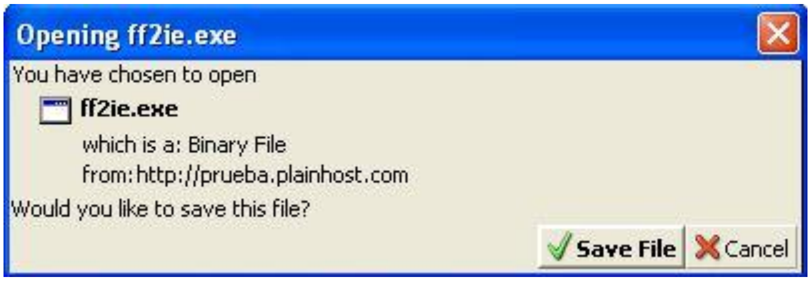
Figure 6. VmDetect Screenshot 2

To any piece of malware, it would appear as if it were running on a normal PC without some kind of virtualization technology. While adding the above lines to the .vmx file of a virtual machine are effective at neutralizing the virtual detection techniques mentioned above, take note that by doing this, it will break some of the functionality of the virtual machine such

VMware Tools, drag and drop, file sharing, clipboard, and more. Since these settings are not documented by VMware, it is important to use at your own risk.

The virtual machine is now built and customized to handle malware analysis. The malware analysis tools have also been installed. Before starting the analysis, a snapshot of the virtual machine is taken so that we can revert back to a clean state at any time during the analysis. With the VMware virtual machine running, click ***VM → Snapshot → Take Snapshot…*** and a clean snapshot is taken. We can now start the analysis phase.

For the first phase, we will do a static analysis of the binary. Static analysis is generally safer than dynamic analysis because the code isn't actually running. There is no need to worry about it deleting files, calling its command and control center, or stealing data. Usually, the only risk involved in static analysis is the risk of accidentally running the malware. To aid in obtaining a piece a malware to analyze, I went to http://www.malwareurl.com. This website is dedicated to fighting malware and other web related threats [8]. I was able to find a malicious link and upon clicking on it, I was offered a piece of malware:


Opening ff2ie.exe

You have chosen to open

ff2ie.exe

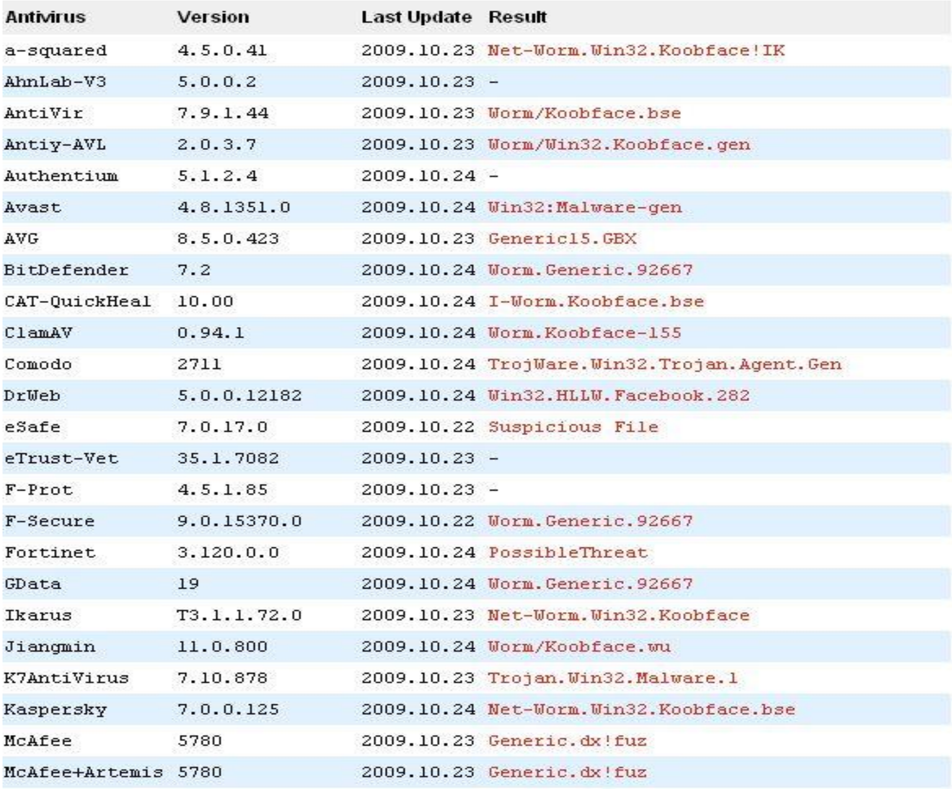
which is a: Binary File

from: http://prueba.plainhost.com

Would you like to save this file?

Figure 7. Malware Screenshot

I saved the file and wanted to know what I was dealing with. I submitted it to http://www.virustotal.com, which is malware analysis website that offers free virus scan services, for further analysis [9]. **Figure 8** shows the results:

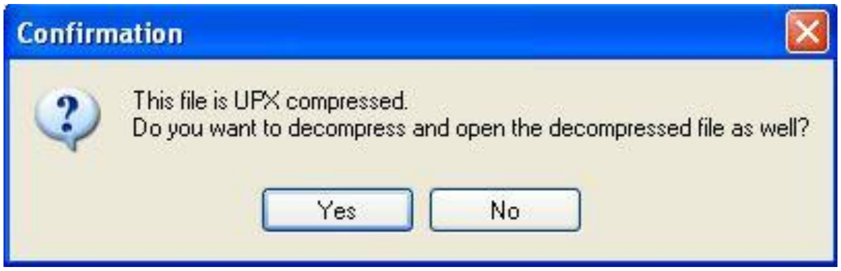


Antivirus	Version	Last Update	Result
a-squared	4.5.0.41	2009.10.23	Net-Worm.Win32.Koobface!IK
AhnLab-V3	5.0.0.2	2009.10.23	-
AntiVir	7.9.1.44	2009.10.23	Worm/Koobface.bse
Antiy-AVL	2.0.3.7	2009.10.23	Worm/Win32.Koobface.gen
Authentium	5.1.2.4	2009.10.24	-
Avast	4.8.1351.0	2009.10.24	Win32:Malware-gen
AVG	8.5.0.423	2009.10.23	Generic15.GBX
BitDefender	7.2	2009.10.24	Worm.Generic.92667
CAT-QuickHeal	10.00	2009.10.24	I-Worm.Koobface.bse
ClamAV	0.94.1	2009.10.24	Worm.Koobface-155
Comodo	2711	2009.10.24	TrojWare.Win32.Trojan.Agent.Gen
DrWeb	5.0.0.12182	2009.10.24	Win32.HLLW.Facebook.282
eSafe	7.0.17.0	2009.10.22	Suspicious File
eTrust-Vet	35.1.7082	2009.10.23	-
F-Prot	4.5.1.85	2009.10.23	-
F-Secure	9.0.15370.0	2009.10.22	Worm.Generic.92667
Fortinet	3.120.0.0	2009.10.24	PossibleThreat
GData	19	2009.10.24	Worm.Generic.92667
Ikarus	T3.1.1.72.0	2009.10.23	Net-Worm.Win32.Koobface
Jiangmin	11.0.800	2009.10.24	Worm/Koobface.wu
K7AntiVirus	7.10.878	2009.10.23	Trojan.Win32.Malware.1
Kaspersky	7.0.0.125	2009.10.24	Net-Worm.Win32.Koobface.bse
McAfee	5780	2009.10.23	Generic.dx!fuz
McAfee+Artemis	5780	2009.10.23	Generic.dx!fuz

Figure 8. VirusTotal Screenshot

By the looks of it, it appears we are dealing with the Koobface worm. Koobface is a computer worm that targets the users of social networking sites Facebook, MySpace, hi5, Bebo, Friendster, and Twitter. Upon infection, it attempts to gather sensitive information from the victims such as passwords to banking sites. Once the attacker gets the login information, they can then login to the account just as if they were the user and have full access to all the information, including credit card numbers.

I wanted to determine what type of file it was. I used FileAlyzer to determine that it was compressed and packed with UPX:



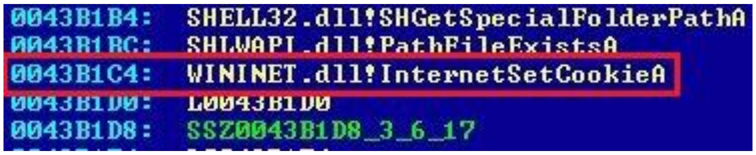
This file is UPX compressed.
Do you want to decompress and open the decompressed file as well?

Figure 9. UPX Screenshot

UPX is one of the most common packers used by malware authors today. It achieves an excellent compression ratio and offers very fast decompression. Rarely does an antivirus product detect the packer until it is too late. Now that the file is decompressed, we can continue the analysis.

To get a better idea of what this piece of malware is going to do once executed, I'll run it through PE Explorer Disassembler. Below are some ideas of what may happen once the binary is executed:

- There will be network activity

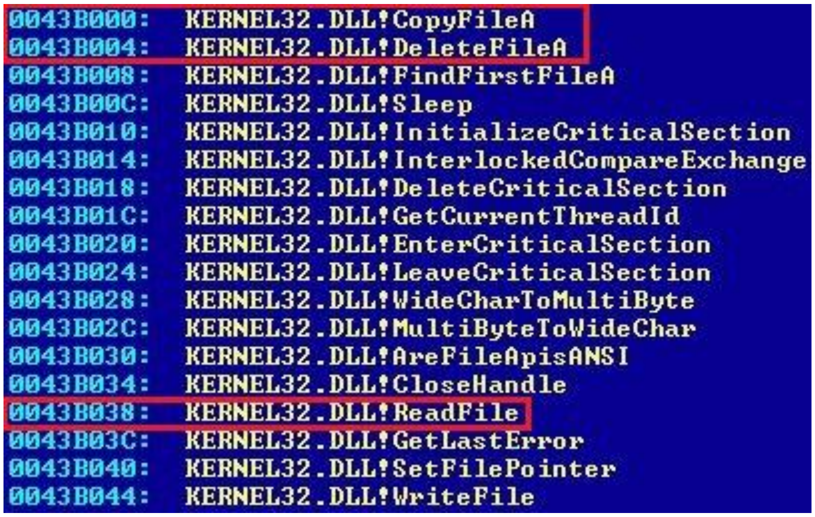


```
0043B1B4:   SHELL32.dll!SHGetSpecialFolderPathA
0043B1BC:   SHLWAPI.dll!PathFileExistsA
0043B1C4:   WININET.dll!InternetSetCookieA
0043B1D0:   L0043B1D0
0043B1D8:   SSZ0043B1D8_3_6_17
```

Figure 10. PE Explorer Screenshot 1

WININET.dll is a module that contains Internet-related functions used by Windows applications. InternetSetCookie is a function that creates a cookie associated with a specified URL.

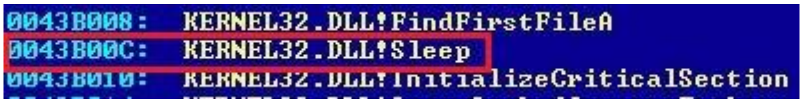
- There is file activity (writing and deleting)



```
0043B000:   KERNEL32.DLL!CopyFileA
0043B004:   KERNEL32.DLL!DeleteFileA
0043B008:   KERNEL32.DLL!FindFirstFileA
0043B00C:   KERNEL32.DLL!Sleep
0043B010:   KERNEL32.DLL!InitializeCriticalSection
0043B014:   KERNEL32.DLL!InterlockedCompareExchange
0043B018:   KERNEL32.DLL!DeleteCriticalSection
0043B01C:   KERNEL32.DLL!GetCurrentThreadId
0043B020:   KERNEL32.DLL!EnterCriticalSection
0043B024:   KERNEL32.DLL!LeaveCriticalSection
0043B028:   KERNEL32.DLL!WideCharToMultiByte
0043B02C:   KERNEL32.DLL!MultiByteToWideChar
0043B030:   KERNEL32.DLL!AreFileApisANSI
0043B034:   KERNEL32.DLL!CloseHandle
0043B038:   KERNEL32.DLL!ReadFile
0043B03C:   KERNEL32.DLL!GetLastError
0043B040:   KERNEL32.DLL!SetFilePointer
0043B044:   KERNEL32.DLL!WriteFile
```

Figure 11. PE Explorer Screenshot 2

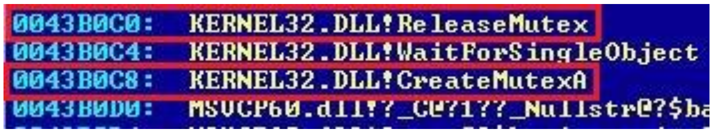
- System time check and sleep for undefined period



```
0043B008:   KERNEL32.DLL!FindFirstFileA
0043B00C:   KERNEL32.DLL!Sleep
0043B010:   KERNEL32.DLL!InitializeCriticalSection
```

Figure 12. PE Explorer Screenshot 3

- Creation of a mutex to ensure only one copy of the worm runs at a time

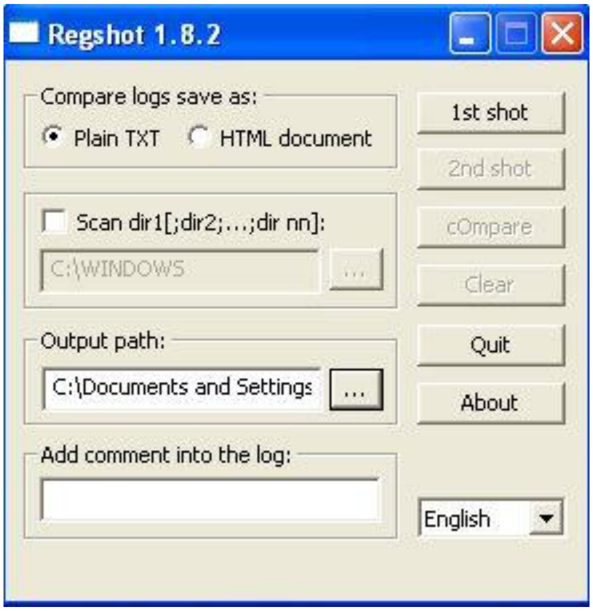


```
0043B0C0:   KERNEL32.DLL!ReleaseMutex
0043B0C4:   KERNEL32.DLL!WaitForSingleObject
0043B0C8:   KERNEL32.DLL!CreateMutexA
0043B0D0:   MSVCP60.dll!?_C@?1??_Nullstr@?$ba
```

Figure 13. PE Explorer Screenshot 4

The above observations are just assumptions and quite vague at that, but we at least have some idea of what to expect. We will not fully know what this piece of malware intends to do until it can be put through dynamic or live analysis. During the live analysis phase, we can expect malicious network activity, malicious files being created and deleted, system down time or sleeping, and mutex creation. Since we are now going to execute the live binary, we can feel safer knowing that we took the proper steps to thwart cross contamination. During this phase, we will also be using VMware's snapshot capability. Since we will be executing the malware numerous times to monitor its actions using various tools, we will need to revert back to a clean state.

Before executing the binary, we will take a snapshot of the registry with Regshot:



Compare logs save as: ● Plain TXT ○ HTML document

☐ Scan dir1[;dir2;...;dir nn]: C:\WINDOWS

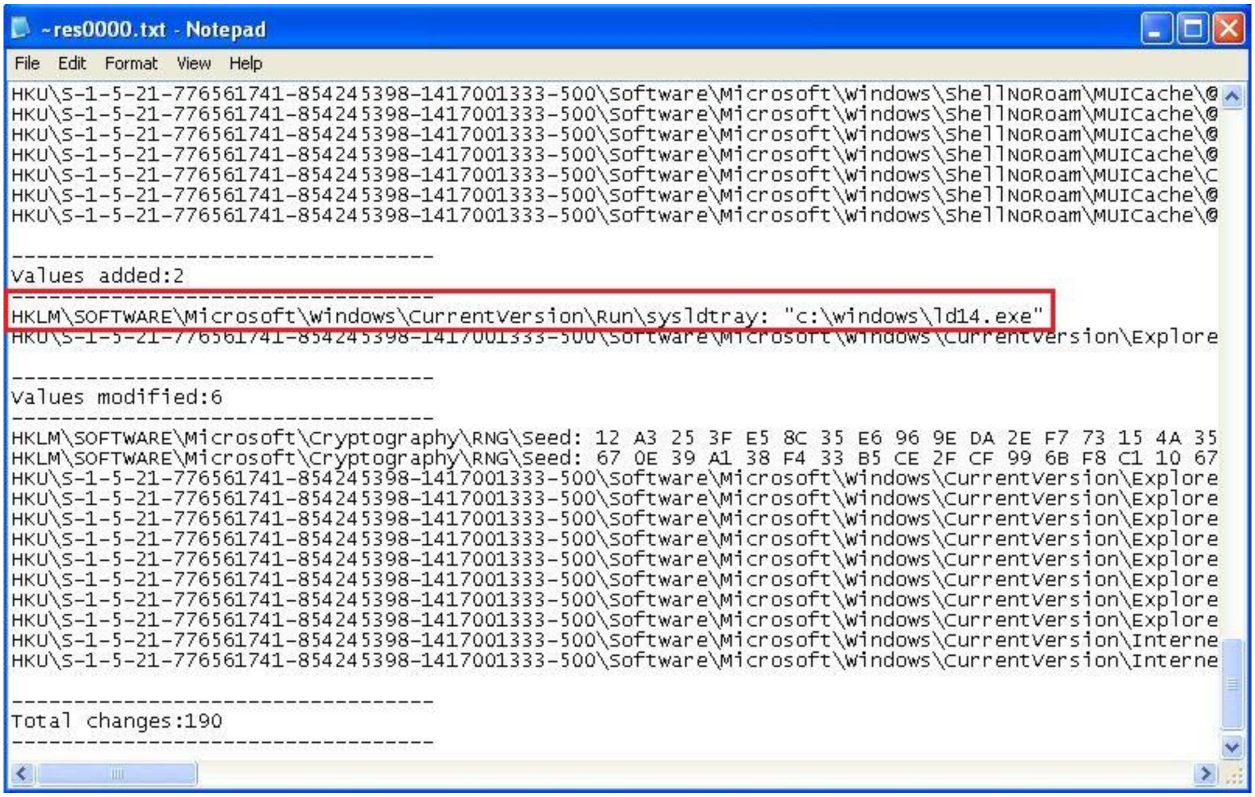
Output path: C:\Documents and Settings

Add comment into the log:

English

Figure 14. Regshot Screenshot

After executing the binary, we will take another snapshot of the registry by clicking on the ***2nd shot*** button then compare both snapshots by clicking the ***cOmpare*** button. The results appear after the compare is done:

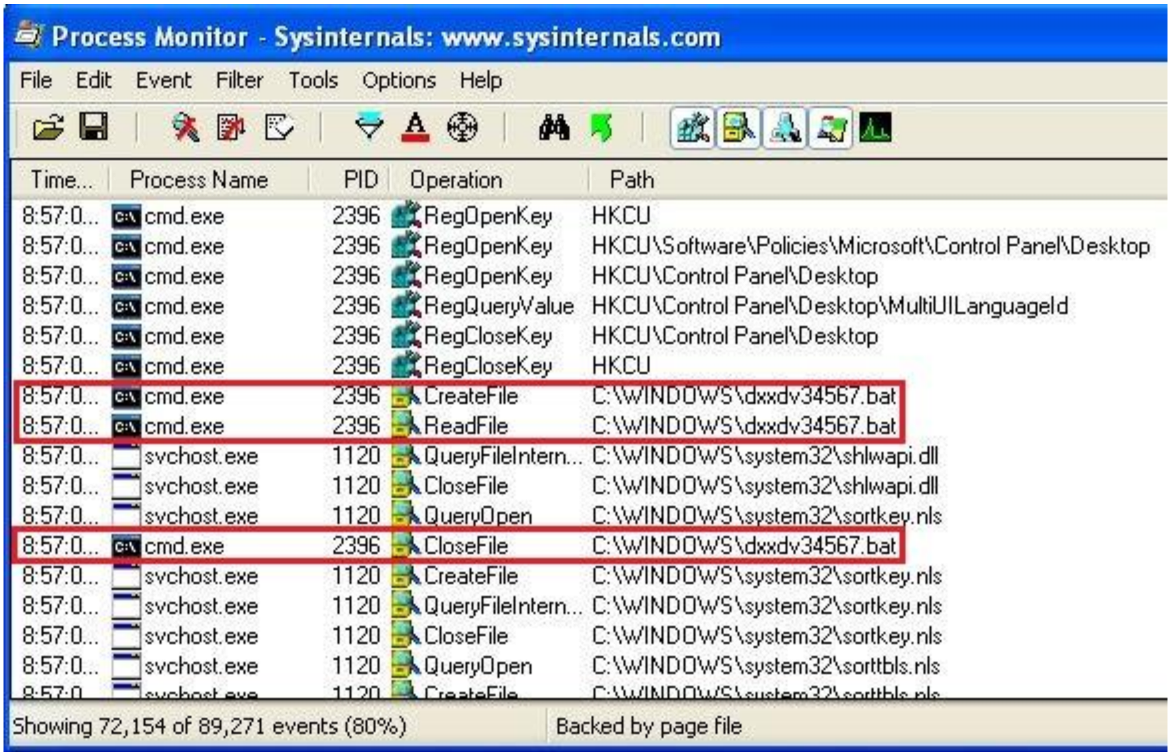


```
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\ShellNoRoam\MUICache\@
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\ShellNoRoam\MUICache\@
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\ShellNoRoam\MUICache\@
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\ShellNoRoam\MUICache\@
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\ShellNoRoam\MUICache\C
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\ShellNoRoam\MUICache\@
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\ShellNoRoam\MUICache\@

----------------------------------
Values added:2
----------------------------------
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\sysldtray: "c:\windows\ld14.exe"
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\CurrentVersion\Explore

----------------------------------
Values modified:6
----------------------------------
HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed: 12 A3 25 3F E5 8C 35 E6 96 9E DA 2E F7 73 15 4A 35
HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed: 67 0E 39 A1 38 F4 33 B5 CE 2F CF 99 6B F8 C1 10 67
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\CurrentVersion\Explore
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\CurrentVersion\Explore
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\CurrentVersion\Explore
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\CurrentVersion\Explore
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\CurrentVersion\Explore
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\CurrentVersion\Explore
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\CurrentVersion\Explore
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\CurrentVersion\Explore
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\CurrentVersion\Interne
HKU\S-1-5-21-776561741-854245398-1417001333-500\Software\Microsoft\Windows\CurrentVersion\Interne

----------------------------------
Total changes:190
----------------------------------
```

Figure 15. Regshot Results

Regshot now shows what values were added and modified as well as the amount of total changes made to the registry. From here we see that the binary will place an entry in the registry HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run. The key name sysldtray points to the file C:\WINDOWS\ld14.exe. This will ensure that the malware will survive reboots because everything placed in this registry location will run automatically when the PC is rebooted.

As we move further into the analysis, we will turn to Process Monitor which is a tool that is very useful in examining running processes on a system. By utilizing this tool, we can see if our malware spawns other processes. While going through the logs, we come across something interesting:

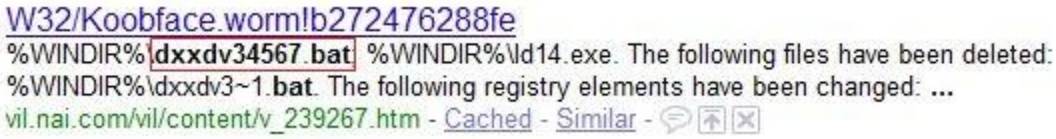


Time...	Process Name	PID	Operation	Path
8:57:0...	cmd.exe	2396	RegOpenKey	HKCU
8:57:0...	cmd.exe	2396	RegOpenKey	HKCU\Software\Policies\Microsoft\Control Panel\Desktop
8:57:0...	cmd.exe	2396	RegOpenKey	HKCU\Control Panel\Desktop
8:57:0...	cmd.exe	2396	RegQueryValue	HKCU\Control Panel\Desktop\MultiUILanguageId
8:57:0...	cmd.exe	2396	RegCloseKey	HKCU\Control Panel\Desktop
8:57:0...	cmd.exe	2396	RegCloseKey	HKCU
8:57:0...	cmd.exe	2396	CreateFile	C:\WINDOWS\dxxdv34567.bat
8:57:0...	cmd.exe	2396	ReadFile	C:\WINDOWS\dxxdv34567.bat
8:57:0...	svchost.exe	1120	QueryFileIntern...	C:\WINDOWS\system32\shlwapi.dll
8:57:0...	svchost.exe	1120	CloseFile	C:\WINDOWS\system32\shlwapi.dll
8:57:0...	svchost.exe	1120	QueryOpen	C:\WINDOWS\system32\sortkey.nls
8:57:0...	cmd.exe	2396	CloseFile	C:\WINDOWS\dxxdv34567.bat
8:57:0...	svchost.exe	1120	CreateFile	C:\WINDOWS\system32\sortkey.nls
8:57:0...	svchost.exe	1120	QueryFileIntern...	C:\WINDOWS\system32\sortkey.nls
8:57:0...	svchost.exe	1120	CloseFile	C:\WINDOWS\system32\sortkey.nls
8:57:0...	svchost.exe	1120	QueryOpen	C:\WINDOWS\system32\sorttbls.nls
8:57:0	svchost.exe	1120	CreateFile	C:\WINDOWS\system32\sorttbls.nls

Showing 72,154 of 89,271 events (80%)   Backed by page file

Figure 16. Process Monitor in action

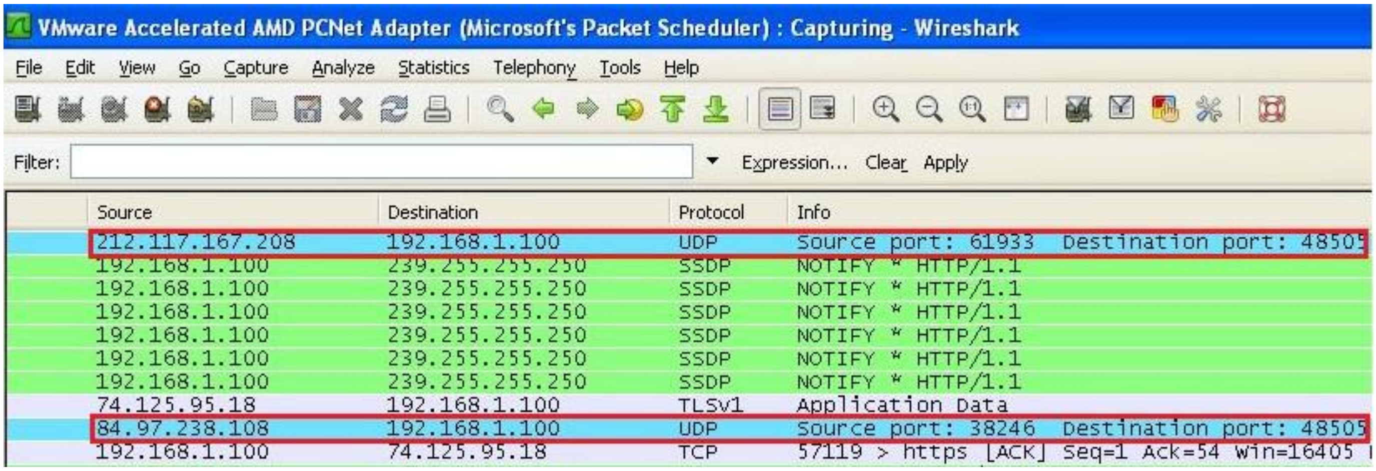
A batch file called ***dxxdv34567.bat*** appears to be created, executed, and then closed. The unconventional name chosen for this file should be enough to throw up a red flag as to whether or not it is legitimate. A search of the system for this file turns up nothing, meaning it is likely hidden. A quick Google search confirms this is the Koobface worm:



W32/Koobface.worm!b272476288fe
%WINDIR%**dxxdv34567.bat** %WINDIR%\ld14.exe. The following files have been deleted: %WINDIR%\dxxdv3~1.**bat**. The following registry elements have been changed: **...**
vil.nai.com/vil/content/v_239267.htm - Cached - Similar -

Figure 17. Google Search Results

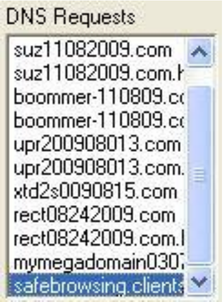
Lastly, to determine what network activity, if any, is taking place, we'll run Wireshark to capture and review the network traffic and use SysAnalyzer to see what DNS requests have been made. In the first few seconds of sniffing packets with Wireshark, we see suspicious traffic:


VMware Accelerated AMD PCNet Adapter (Microsoft's Packet Scheduler) : Capturing - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression... Clear Apply

```
Source            Destination        Protocol  Info
212.117.167.208   192.168.1.100      UDP       Source port: 61933  Destination port: 48505
192.168.1.100     239.255.255.250    SSDP      NOTIFY * HTTP/1.1
192.168.1.100     239.255.255.250    SSDP      NOTIFY * HTTP/1.1
192.168.1.100     239.255.255.250    SSDP      NOTIFY * HTTP/1.1
192.168.1.100     239.255.255.250    SSDP      NOTIFY * HTTP/1.1
192.168.1.100     239.255.255.250    SSDP      NOTIFY * HTTP/1.1
192.168.1.100     239.255.255.250    SSDP      NOTIFY * HTTP/1.1
74.125.95.18      192.168.1.100      TLSv1     Application Data
84.97.238.108     192.168.1.100      UDP       Source port: 38246  Destination port: 48505
192.168.1.100     74.125.95.18       TCP       57119 > https [ACK] Seq=1 Ack=54 Win=16405
```

Figure 18. Wireshark in action

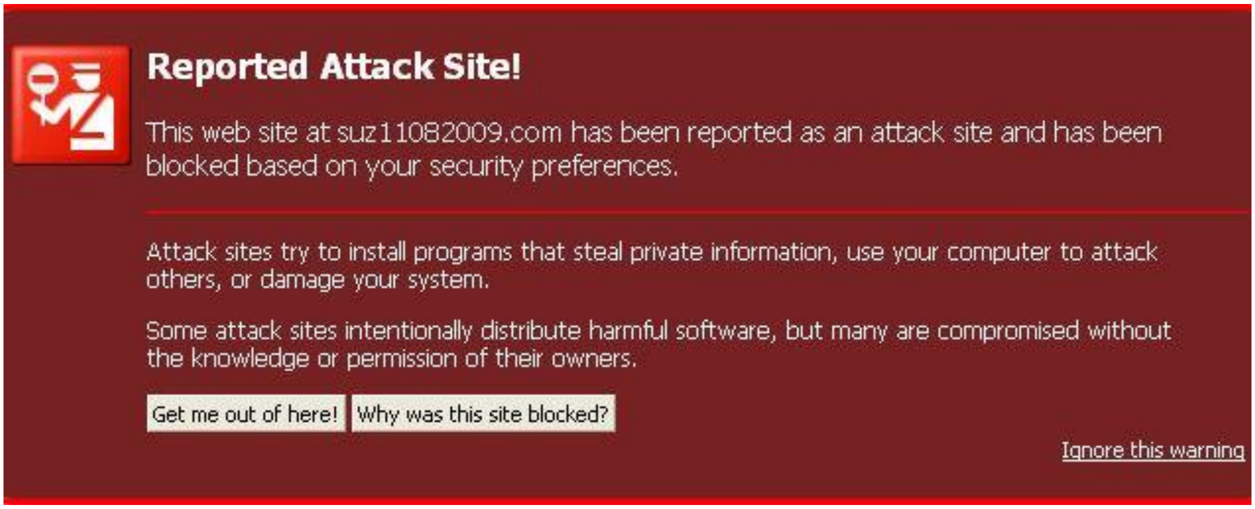
A Whois lookup of both Source IP addresses determines that the traffic is coming from Luxembourg and France. It is a good possibility that these IP addresses are the command and control center for this particular piece of malware. Also take note of the high port numbers being used. Worms are notorious for generating traffic on higher, lesser known, random port numbers. SysAnalyzer will show what DNS requests have been made:



```
suz11082009.com
suz11082009.com.
boommer-110809.c
boommer-110809.c
upr200908013.com
upr200908013.com.
xtd2s0090815.com
rect08242009.com
rect08242009.com.
mymegadomain030
safebrowsing.client
```

Figure 19. SysAnalyzer showing DNS requests

Incidentally, navigate to any of the above sites and you will receive the following warning:



Reported Attack Site!

This web site at suz11082009.com has been reported as an attack site and has been blocked based on your security preferences.

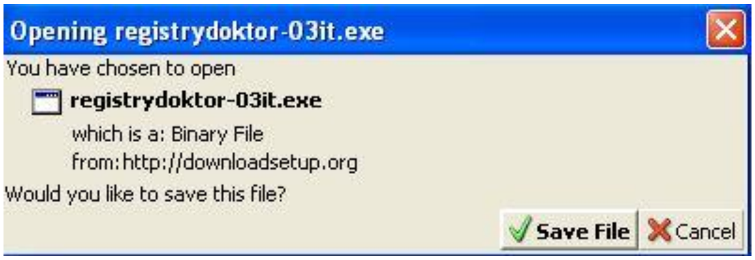
Attack sites try to install programs that steal private information, use your computer to attack others, or damage your system.

Some attack sites intentionally distribute harmful software, but many are compromised without the knowledge or permission of their owners.

Figure 20. Mozilla Firefox reporting an Attack Site

Koobface is likely trying to connect to any one of these sites to download exploits to continue its attack.

I would like to walkthrough the analysis of one more piece of malware to get a good feel for the tools being used. Most of these tools can be used in the everyday use of computers so it should be important to have an understanding of how they work. First, we will navigate back to http://www.malwareurl.com to find a malicious link. Upon clicking on the link, we get the following offering:



Opening registrydoktor-03it.exe

You have chosen to open

registrydoktor-03it.exe

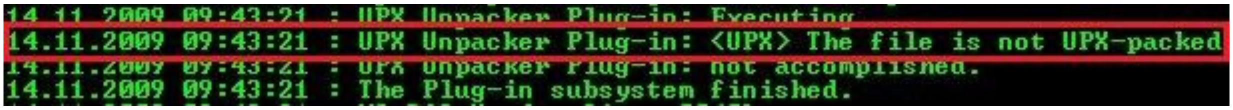
which is a: Binary File

from: http://downloadsetup.org

Would you like to save this file?

Figure 21. Malware being offered

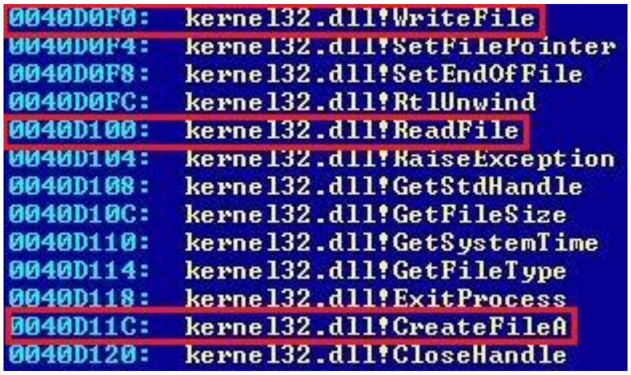
This is then saved to the desktop for analysis. Using PE Explorer, we want to determine if this file has been packed. Below, Figure 22 confirms that it is not:



```
14.11.2009 09:43:21 : UPX Unpacker Plug-in: Executing
14.11.2009 09:43:21 : UPX Unpacker Plug-in: <UPX> The file is not UPX-packed
14.11.2009 09:43:21 : UPX Unpacker Plug-in: not accomplished.
14.11.2009 09:43:21 : The Plug-in subsystem finished.
```

Figure 22. File not UPX-packed

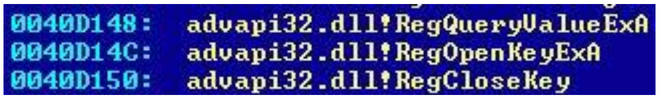
Going back to PE Explorer, let’s see if we can determine what this file may do once executed. From our analysis in the screenshot in Figure 23 below, we can expect to files created, read, and written to:



```
0040D0F0:   kernel32.dll!WriteFile
0040D0F4:   kernel32.dll!SetFilePointer
0040D0F8:   kernel32.dll!SetEndOfFile
0040D0FC:   kernel32.dll!RtlUnwind
0040D100:   kernel32.dll!ReadFile
0040D104:   kernel32.dll!RaiseException
0040D108:   kernel32.dll!GetStdHandle
0040D10C:   kernel32.dll!GetFileSize
0040D110:   kernel32.dll!GetSystemTime
0040D114:   kernel32.dll!GetFileType
0040D118:   kernel32.dll!ExitProcess
0040D11C:   kernel32.dll!CreateFileA
0040D120:   kernel32.dll!CloseHandle
```

Figure 23. PE Explorer screenshot 1

We can also expect this piece of malware to make changes to the registry as shown below in Figure 24:



```
0040D148:   advapi32.dll!RegQueryValueExA
0040D14C:   advapi32.dll!RegOpenKeyExA
0040D150:   advapi32.dll!RegCloseKey
```

Figure 24. PE Explorer screenshot 2

Since we have very little information on what to expect, we'll turn to live analysis. To record the entire installation, a program called InstallRite will be used. InstallRite records every step of the installation process. It will tell you exactly what files were created, modified, and deleted as well as doing the same for the registry. Before installing, InstallRite will create a snapshot of the current system state. Figure 25 shows InstallRite just before snapshot creation and Figure 26 shows the process in action:

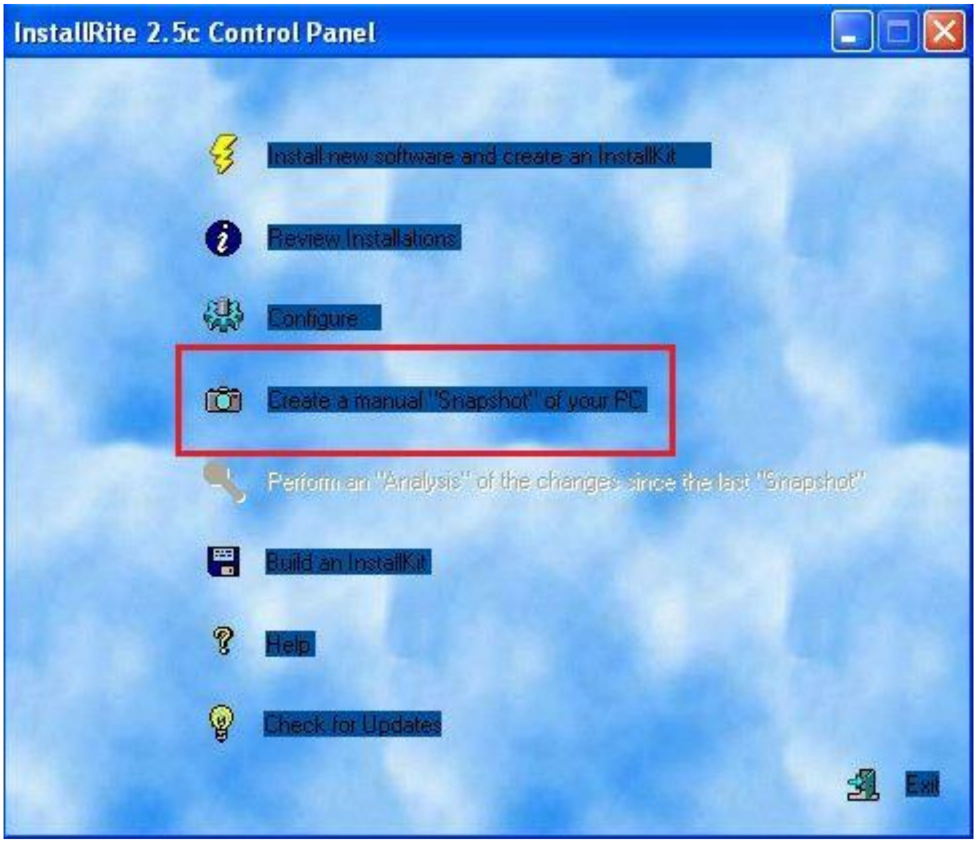
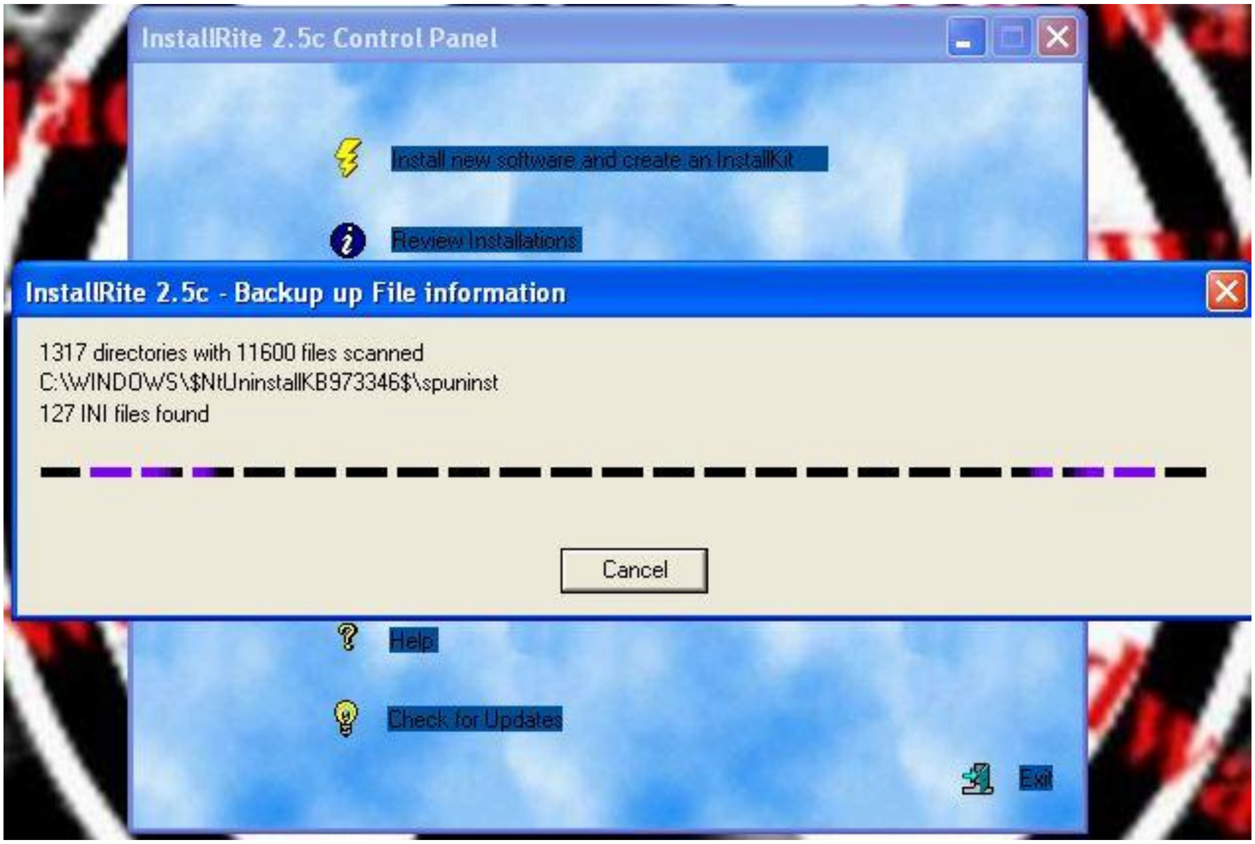

InstallRite 2.5c Control Panel
Install new software and create an InstallKit
Review Installations
Configure
Create a manual "Snapshot" of your PC
Perform an "Analysis" of the changes since the last "Snapshot"
Build an InstallKit
Help
Check for Updates
Exit

Figure 25. InstallRite just before snapshot creation



InstallRite 2.5c Control Panel
Install new software and create an InstallKit
Review Installations
Help
Check for Updates
Exit

InstallRite 2.5c - Backup up File information

1317 directories with 11600 files scanned
C:\WINDOWS\$NtUninstallKB973346$\spuninst
127 INI files found

Figure 26. InstallRite creating a system snapshot

We can now install the questionable executable. Figures 27 and 28 document this process:

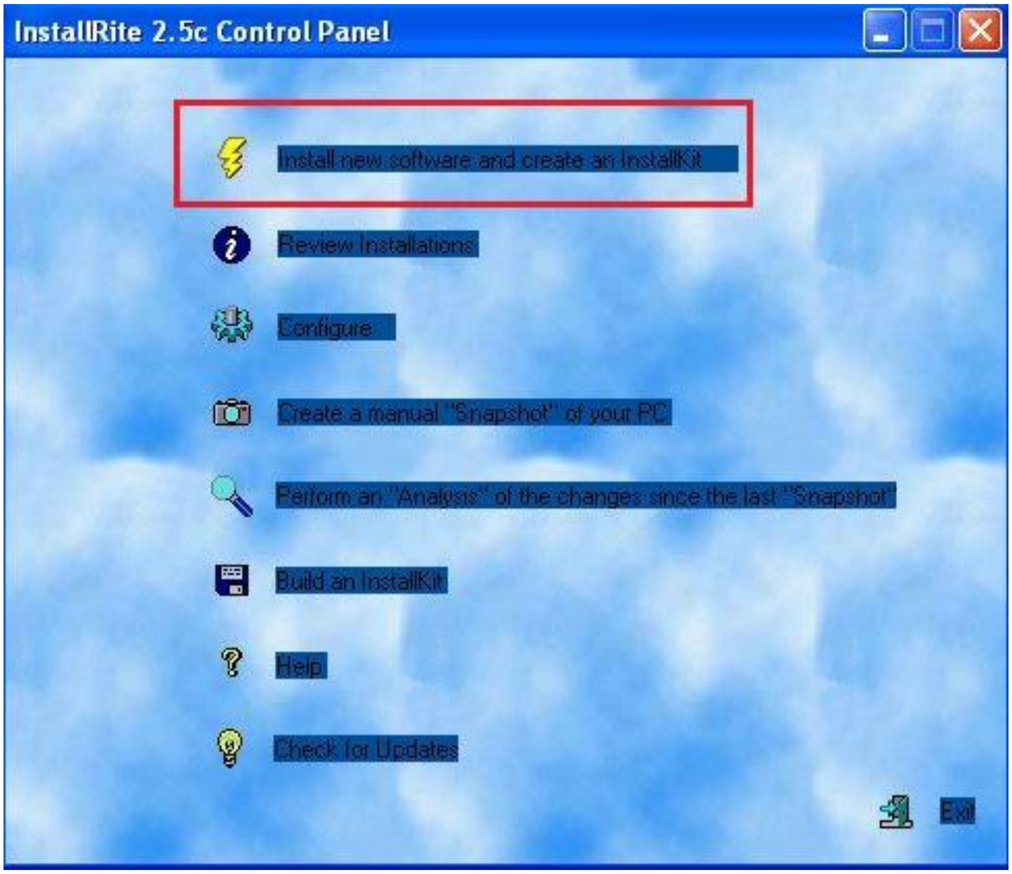
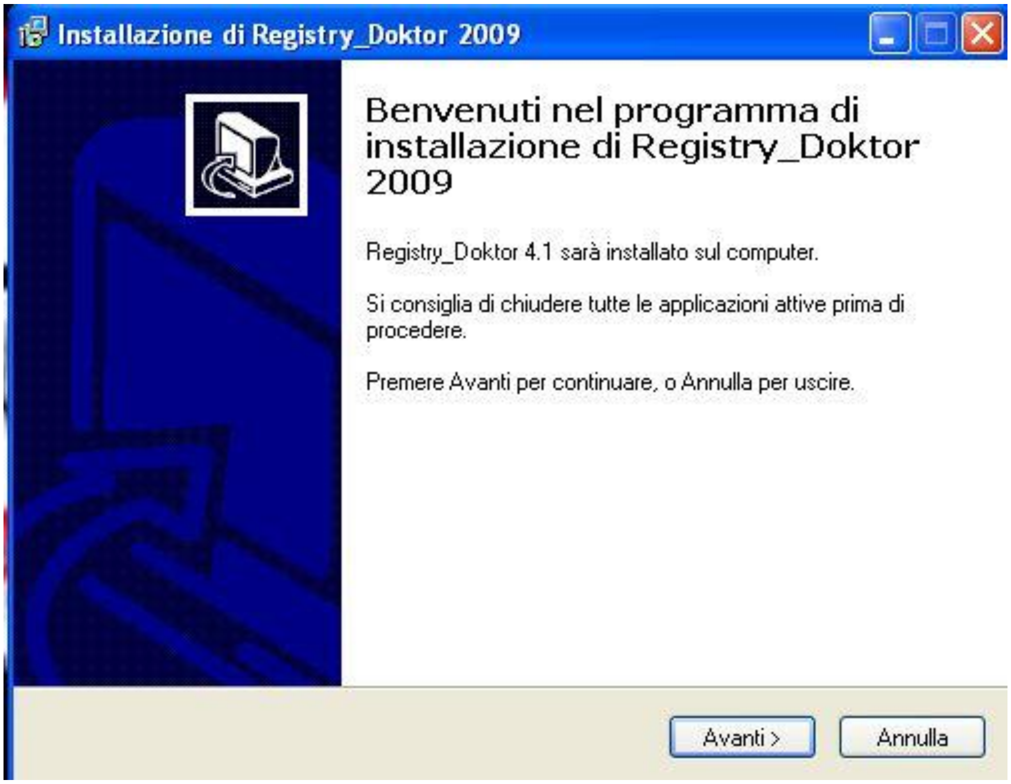

InstallRite 2.5c Control Panel
Install new software and create an InstallKit
Review Installations
Configure
Create a manual "Snapshot" of your PC
Perform an "Analysis" of the changes since the last "Snapshot"
Build an InstallKit
Help
Check for Updates
Exit

Figure 27. Selecting option to install executable



Installazione di Registry_Doktor 2009

Benvenuti nel programma di installazione di Registry_Doktor 2009

Registry_Doktor 4.1 sarà installato sul computer.

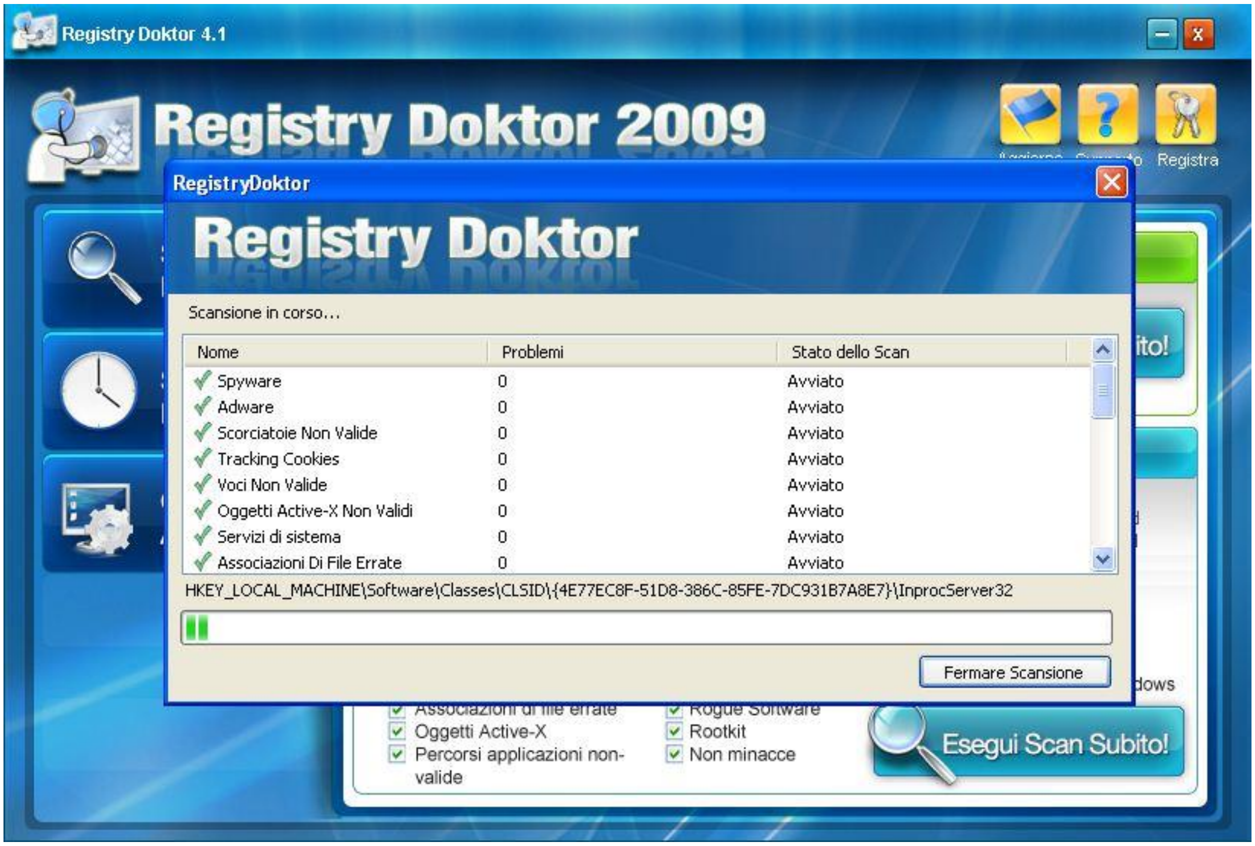
Si consiglia di chiudere tutte le applicazioni attive prima di procedere.

Premere Avanti per continuare, o Annulla per uscire.

Figure 28. Installation process of questionable executable

The first warning flag is raised during the installation process. As you can see above in Figure 28, make note of the different language being used.

As soon as the installation process finishes, we begin to see a popup of what appears to be “Registry Doktor 2009” scanning our PC:



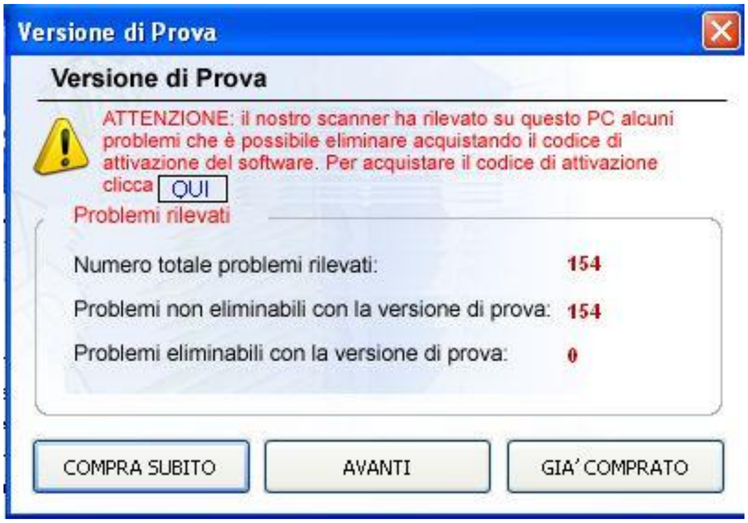
Scansione in corso...

Nome	Problemi	Stato dello Scan
✓ Spyware	0	Avviato
✓ Adware	0	Avviato
✓ Scorciatoie Non Valide	0	Avviato
✓ Tracking Cookies	0	Avviato
✓ Voci Non Valide	0	Avviato
✓ Oggetti Active-X Non Validi	0	Avviato
✓ Servizi di sistema	0	Avviato
✓ Associazioni Di File Errate	0	Avviato

HKEY_LOCAL_MACHINE\Software\Classes\CLSID\{4E77EC8F-51D8-386C-85FE-7DC931B7A8E7}\InprocServer32

Figure 29. Registry Doktor 2009 scanning

The scan finishes and warns that errors were found:



Versione di Prova

ATTENZIONE: il nostro scanner ha rilevato su questo PC alcuni problemi che è possibile eliminare acquistando il codice di attivazione del software. Per acquistare il codice di attivazione clicca QUI

Problemi rilevati

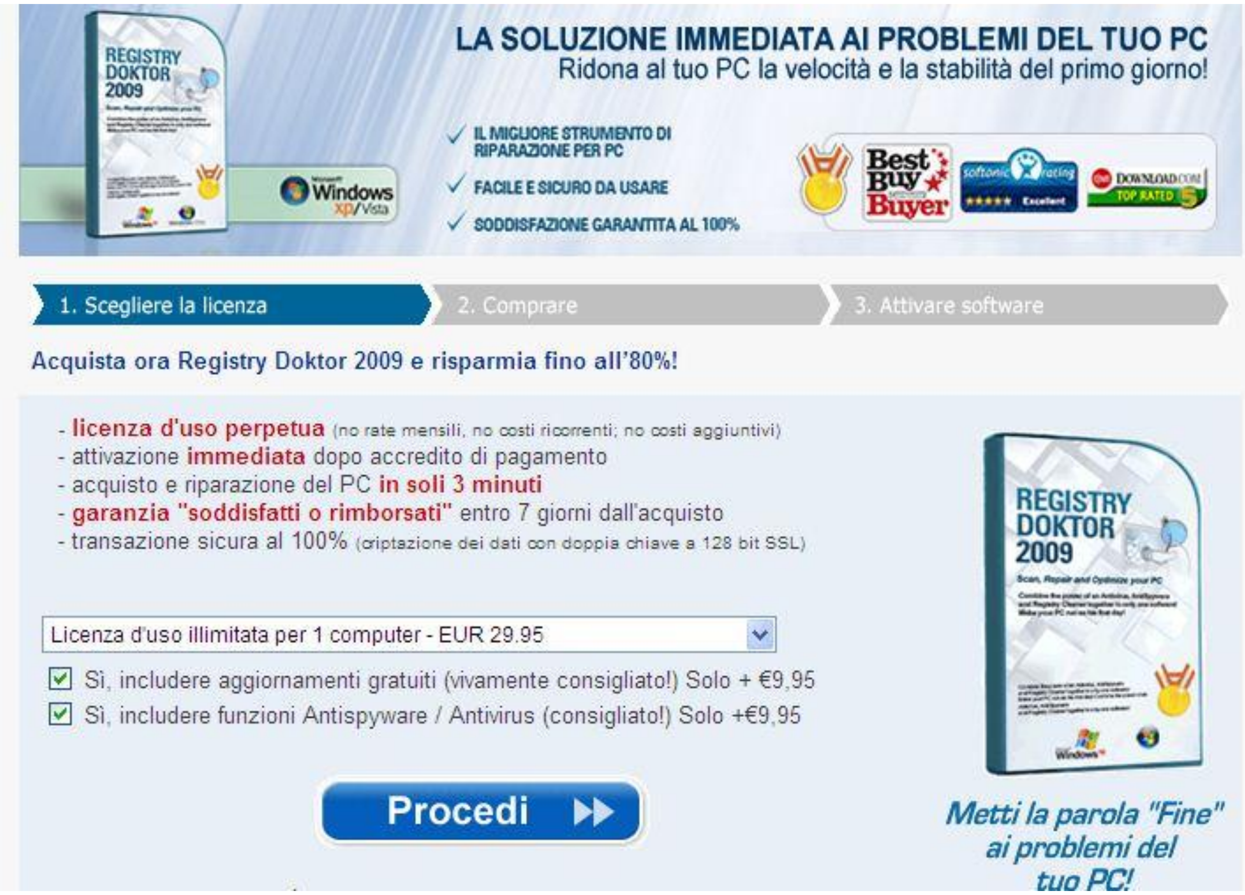
Numero totale problemi rilevati: 154

Problemi non eliminabili con la versione di prova: 154

Problemi eliminabili con la versione di prova: 0

Figure 30. Registry Doktor 2009 error warning

Upon clicking on any of the buttons in Figure 30, we are redirected to the Registry Doktor 2009 activation page in Figure 31 where we are told we can enter a credit card number to purchase Registry Doktor 2009. Once Registry Doktor 2009 is purchased, it will then fix the error messages:



LA SOLUZIONE IMMEDIATA AI PROBLEMI DEL TUO PC
Ridona al tuo PC la velocità e la stabilità del primo giorno!

✓ IL MIGLIORE STRUMENTO DI RIPARAZIONE PER PC
✓ FACILE E SICURO DA USARE
✓ SODDISFAZIONE GARANTITA AL 100%

1. Scegliere la licenza 2. Comprare 3. Attivare software

Acquista ora Registry Doktor 2009 e risparmia fino all'80%!

- licenza d'uso perpetua (no rate mensili, no costi ricorrenti; no costi aggiuntivi)
- attivazione immediata dopo accredito di pagamento
- acquisto e riparazione del PC in soli 3 minuti
- garanzia "soddisfatti o rimborsati" entro 7 giorni dall'acquisto
- transazione sicura al 100% (criptazione dei dati con doppia chiave a 128 bit SSL)

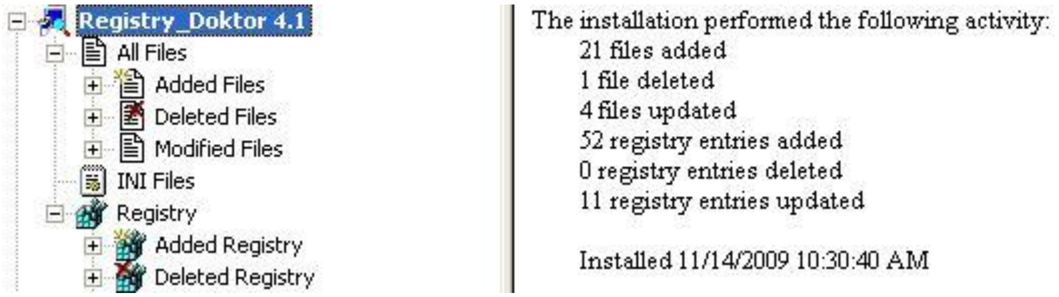
Licenza d'uso illimitata per 1 computer - EUR 29.95

☑ Sì, includere aggiornamenti gratuiti (vivamente consigliato!) Solo + €9,95
☑ Sì, includere funzioni Antispyware / Antivirus (consigliato!) Solo +€9,95

Figure 31. Registry Doktor 2009 registration page

By now it should be obvious that rogueware has been installed. Rogueware is typically thought of as fake antivirus or antispyware that attempts to trick users into purchasing the product to remove nonexistent threats. Essentially, users are being scammed and losing money as well as having their credit card number stolen when entered into the rogueware's website.

Using InstallRite, we can see what changes are made when this rogueware is installed:



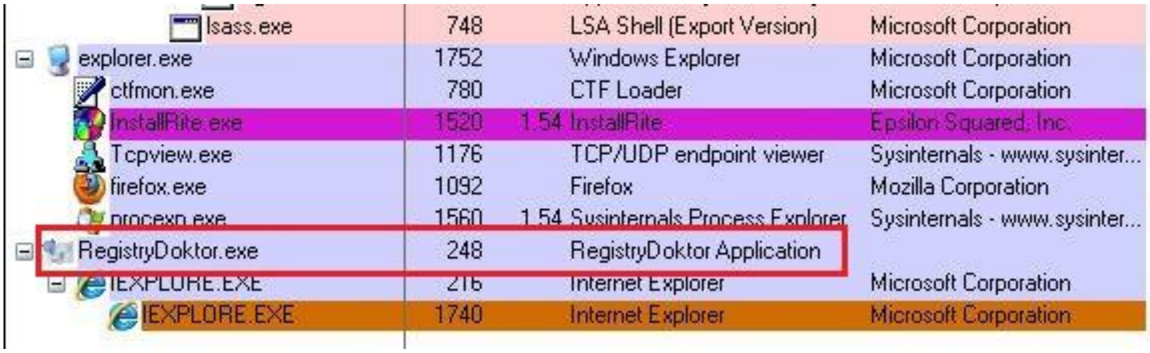
```
Registry_Doktor 4.1
    All Files
        Added Files
        Deleted Files
        Modified Files
    INI Files
    Registry
        Added Registry
        Deleted Registry
```

The installation performed the following activity:
21 files added
1 file deleted
4 files updated
52 registry entries added
0 registry entries deleted
11 registry entries updated

Installed 11/14/2009 10:30:40 AM

Figure 32. InstallRite showing changes made by the Rogueware

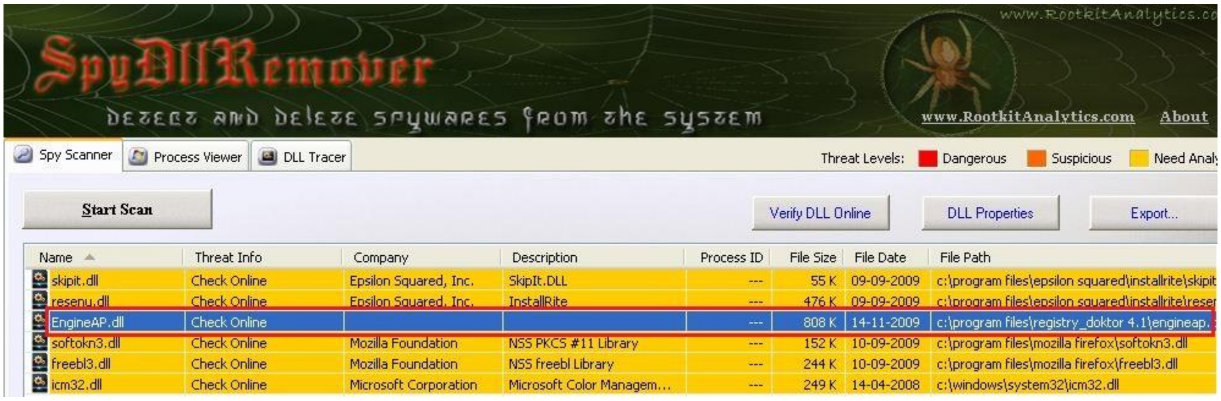
We can then use Process Explorer to kill any malicious processes created by Registry Doktor 2009:



```
        lsass.exe          748         LSA Shell (Export Version)   Microsoft Corporation
explorer.exe              1752         Windows Explorer             Microsoft Corporation
    ctfmon.exe             780         CTF Loader                   Microsoft Corporation
    InstallRite.exe       1520   1.54  InstallRite                  Epsilon Squared, Inc.
    Tcpview.exe           1176         TCP/UDP endpoint viewer      Sysinternals - www.sysinter...
    firefox.exe           1092         Firefox                      Mozilla Corporation
    procexp.exe           1560   1.54  Sysinternals Process Explorer Sysinternals - www.sysinter...
RegistryDoktor.exe         248         RegistryDoktor Application
    IEXPLORE.EXE           216         Internet Explorer            Microsoft Corporation
        IEXPLORE.EXE      1740         Internet Explorer            Microsoft Corporation
```

Figure 33. Process Monitor showing malicious process

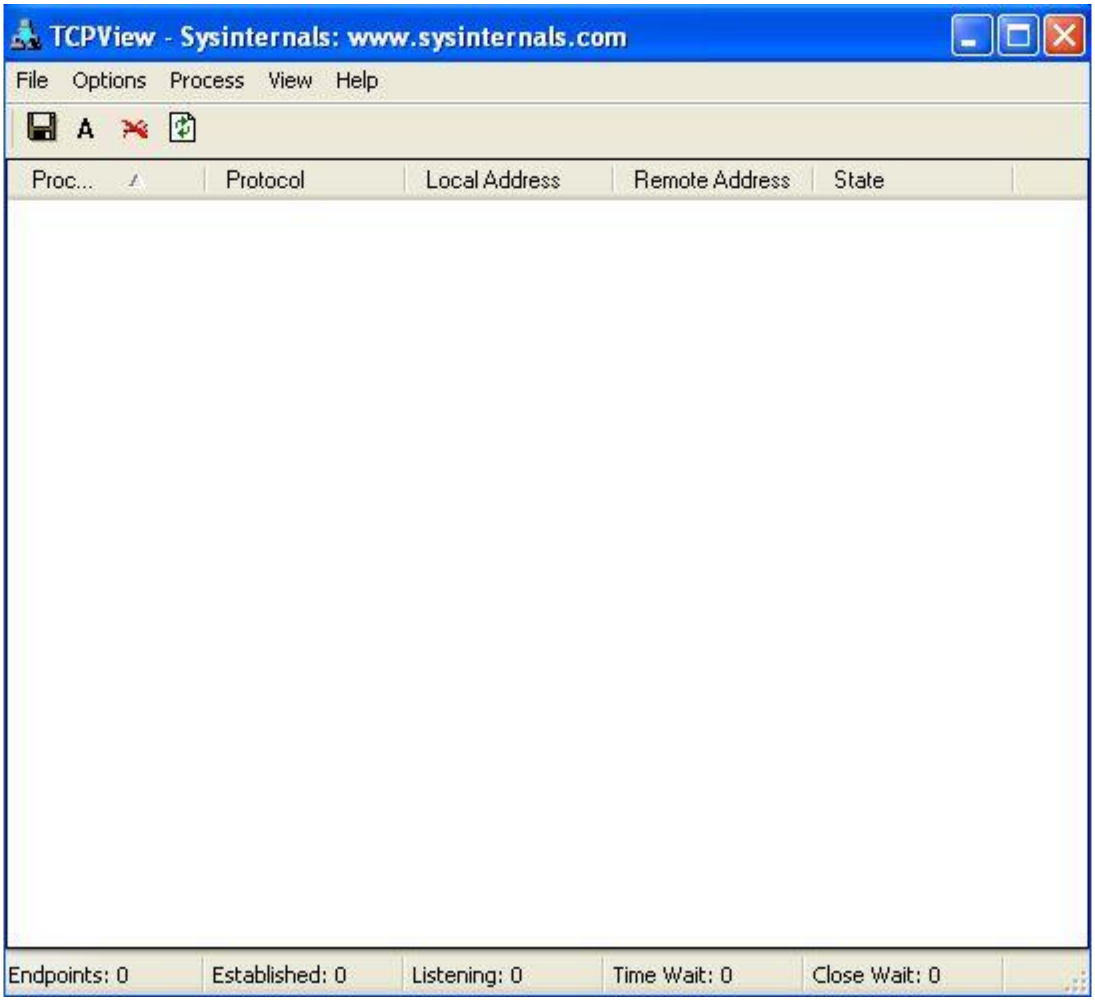
SpyDLLRemover will show us any malicious DLL's left behind by Registry Doktor 2009:



Name	Threat Info	Company	Description	Process ID	File Size	File Date	File Path
skipit.dll	Check Online	Epsilon Squared, Inc.	SkipIt.DLL	---	55 K	09-09-2009	c:\program files\epsilon squared\installrite\skipit
resenu.dll	Check Online	Epsilon Squared, Inc.	InstallRite	---	476 K	09-09-2009	c:\program files\epsilon squared\installrite\reser
EngineAP.dll	Check Online			---	808 K	14-11-2009	c:\program files\registry_doktor 4.1\engineap.
softokn3.dll	Check Online	Mozilla Foundation	NSS PKCS #11 Library	---	152 K	10-09-2009	c:\program files\mozilla firefox\softokn3.dll
freebl3.dll	Check Online	Mozilla Foundation	NSS freebl Library	---	244 K	10-09-2009	c:\program files\mozilla firefox\freebl3.dll
icm32.dll	Check Online	Microsoft Corporation	Microsoft Color Managem...	---	249 K	14-04-2008	c:\windows\system32\icm32.dll

Figure 34. SpyDLLRemover showing a malicious DLL

TCPView shows no network activity present once the rogueware installed. This is a good sign as we can be confident our PC is not being controlled by an attacker:



Proc...	Protocol	Local Address	Remote Address	State

Endpoints: 0 Established: 0 Listening: 0 Time Wait: 0 Close Wait: 0

Figure 35. TCPView showing no network activity

Once the malicious processes and DLL's are removed, we can then begin the process of manually removing any changes made to the registry by Registry Doktor 2009. While it can be a painful process to parse the registry, it is often necessary to ensure complete removal of the rogueware.

Conclusion

The Internet is generally thought to be a fantastic cyber environment to be a part of. It allows people to stay in touch when separated by large distances. It can be accessed almost anywhere, any time. Anyone with access can stay in touch with friends and family via social networking sites like Facebook, MySpace, and Twitter. The Internet also provides an endless supply of knowledge helping young children to adults learn about whatever interests them. From a business standpoint, the Internet allows for greater flexibility in working hours and location, especially with high-speed connections. Users can work from home or a coffee shop or even check emails while on vacation. Unfortunately, there is a darker side to the Internet, one that most users don't see until it is too late. Virus infections are becoming an increasing problem. Backdoor Trojans are infecting user PCs and stealing personal information such as credit card numbers and passwords, which could lead to identity theft. The focus of this paper was mainly on the reverse engineering aspect, to show exactly what a piece of malware can do before and after it was run on a live system. Unfortunately, this happens to many thousands of users on a daily basis without them knowing. The best way to combat malware is to never connect your PC to the Internet in the first place but if you do ensure you have a good antivirus / anti-malware client with up-to-date virus definitions. It is also a good idea to have a router with firewall capability so that you can filter the bad traffic away from your network. Should you do online banking, it is best to use a separate PC that only connects to the Internet for this purpose. Once finished, unplug it from the Internet so as not to expose it to any further risks. And most importantly, keep all PCs up to date with the latest patches. These are not sure fire ways to keep your system clean and in fact there is no complete or perfect solution. But by taking the proper precautions, you will more than likely be able to enjoy all the Internet has to offer.

References

[1] Internet. Wikimedia Foundation, Inc. Retrieved October 25, 2009 from: http://en.wikipedia.org/wiki/The_internet

[2] History of the Internet. Wikimedia Foundation, Inc. Retrieved October 25, 2009 from: http://en.wikipedia.org/wiki/History_of_the_Internet

[3] Mell, P., Kent, K., & Nusbaum, J. (2005). Guide to Malware Incident Prevention and Handling. Retrieved (2009, November 1) from http://www.csrc.nist.gov/publications/nistpubs/800-83/SP800-83.pdf

[4] Quist, D. (n.d.). *Offensive computing | Community malicious code research and analysis.* Retrieved from http://www.offensivecomputing.net/

[5] Spitzner, L. (2004). *Know your enemy: Learning about security threats / the Honeynet Project*. Boston: Addison-Wesley.

[6] (n.d.). *VMware workstation - Run Multiple OS including Linux on Windows on Virtual Machines*. Retrieved from http://www.vmware.com/products/workstation/

[7] Ormandy, T. (2007). An Empirical Study into the Security Exposure to Hosts of Hostile Virtualized Environments. Retrieved (2009, November 1) from http://taviso.decsystem.org/virtsec.pdf

[8] (n.d.). *MalwareURL - URL listing*. Retrieved from http://www.malwareurl.com/listing-urls.php?urls=off

[9] (n.d.). *VirusTotal - Free Online Virus and Malware Scan*. Retrieved from http://www.virustotal.com/

[10] Harris, S., Harper, A., Eagle, C., & Ness, J. (2008). *Gray Hat Hacking: The Ethical Hacker's Handbook*. New York: McGraw-Hill.

[11] Hyslip, T. S. (2007). Malware Response and Analysis. Retrieved (2009, November 1) from http://www.infosecwriters.com/text_resources/pdf/THyslip_Malware.pdf

[12] Distler, D. (2007). Malware Analysis: An Introduction. Retrieved (2009, November 8) from http://www.sans.org/reading_room/whitepapers/malicious/malware_analysis_an_introduction_2103?show=2103.php&cat=malicious

[13] (2000, July 12). *Sans: Intrusion Detection FAQ: What is a Honeypot?* Retrieved from http://www.sans.org/security-resources/idfaq/honeypot3.php

Appendix

TCPView	www.sysinternals.com
Process Explorer	www.sysinternals.com
Psfile	www.sysinternals.com
HookExplorer	http://labs.idefense.com/files/labs/releases/previews/hookexplorer
PEiD	www.peid.info
RegShot	http://sourceforge.net/projects/regshot/
InstallRite	http://www.epsilonsquared.com/installrite.htm
UPX	http://upx.sourceforge.net/
GMER	http://www.gmer.net/
OllyDBG	http://www.ollydbg.de/
IDA Pro	http://www.hex-rays.com/idapro/
FileInsight	http://www.avertlabs.com/research/blog/index.php/2009/09/10/new-version-of-mcafee-fileinsight/
Malcode Pack	http://labs.idefense.com/software/malode.php#more_malcode+analysis+pack
SysAnalyzer	http://labs.idefense.com/software/malcode.php
Wireshark	http://www.wireshark.org/
ProcNetMonitor	http://securityxploded.com/procnetmonitor.php
FileAlyzer	http://www.safer-networking.org/en/filealyzer/index.html
Google Search	www.google.com
Whois Lookup	http://cqcounter.com/whois/
VmDetect	http://www.codeproject.com/KB/system/VmDetect.aspx
Scoopy	http://www.trapkit.de/research/vmm/scoopydoo/index.html
Red Pill	http://www.invisiblethings.org/papers/redpill.html
PE Explorer	http://www.heaventools.com/overview.htm
SpyDLLRemover	http://securityxploded.com/spydllremover.php