Fight crime.
Unravel incidents... one byte at a time.

MALWARE ADVENTURE

GIAC Reverse Engineering Malware
Pratical Assignment 1.0
Prepared by: Russell Elliott
September 17, 2004

Abstract

This practical assignment reviews the steps taken to reverse engineer a malware specimen. Before the analysis is started, the methodology used to step up a test laboratory sutiable for reverse engineering malware is discussed. The necessary tools that are need are reviewed. The reverse engineering is accomplished through three steps, physical characteristics of the specimen, behavioral analysis, and code analysis. Finally, through what is learned about the specimen and combining the steps, defensive and prevent measures are discussed.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

MALWARE ADVENTURE

INTRODUCTION

It will be an adventure in analyzing the malware specimen, msrll.exe. As with any great adventure, the preparation for the adventure is with the laboratory setup needed for the analysis of the malware specimen and will be discussed first. Once the laboratory is setup and explained, the adventure begins with a discussion of the properties of the malware specimen. Next, we start the adventure with an examination the behavior by running msrll.exe. This behavioral analysis will give us clues for what direction to go in when we begin looking at the code. After the code analysis, the findings will be wrapped up.

Commands are typed are indicated in bold characters. After each command, an explanation will be given on what the command will do.

LABORATORY SETUP

The laboratory setup must meet the needs of reverse-engineering. The major problem with reverse-engineering is isolating the laboratroy computer(s) from production networks and the internet. If not, there is a risk of not only purposely infecting the laboratory computer(s) but also other computers on the network. Therefore, to eliminate this risk a standalone computer is used with VMware Workstation 4.0 software, so multiple physical computers, physical hubs/switches, etc. are not needed. The resources required to run multiple operating systems consists of three items, processing speed, memory, and disk space. The host computer used is configured with a Pentium 4 2.6 GHz processor, 1,024 MB of RAM, and 250 GB hard drive. This configuration will allow for an acceptable level of performance for emulating an actual network. Once the host computer was backed up, the next step will be installing and configuring the guest operating systems.

Windows Operating System

The guest Windows operating system was initially installed in VMware with a bridge connection to the internet. This allowed for activation of the Windows guest and for running the Windows Automatic Update service to insure that the operating system was fully patched. Once the activation and update were completed, the host computer was physically removed from the network, the bridge connection was removed, and the host only connection implemented. The Windows guest computer's final configuration consisted of one 8 GB IDE hard drive with two partitions, C and D, 128 MB RAM, one network interface card configured as a host only connection and to use DHCP, a IDE CD-ROM drive, a USB controller, and of course a floppy drive.

The Windows software from the course materials were copied to and installed on the guest system. A list of the initially installed software along with a description, anticipated use, and the area of analysis the software will be used for are given in Table 1 – Windows Guest Software. The area of use on for some of the tools can be used either during the behavioral analysis phase, code analysis phase, or both. Table 1 – Windows Guest Software lists predominate area that the tool is normally used in. When preparing to analyze an unknown malware specimen, the exact tools required is not known. Therefore, I rather have most tools installed and ready to use before the actual analysis begins. Even with the tools listed in Table 1 – Windows Guest Software, there is no guaranty that all of the tools will be used or that additional tools will be needed. Finally, the msrll.zip file is copied to the Windows guest operating.

The final step for setting up the Windows guest operating system is to obtain a starting point or baseline. This will aid in starting over if needed during the analysis with a fresh installation. Two options are used. First, a snapshot of the current guest system is taken. This will be a quick way to restore the operating system. The second is zipping the VMware files for the Windows guest operating system. Next, we will look at the Linux setup.

Software	Description	Use	Area of Analysis
Bintext	Detects strings and ASCII characters it the code	Determine strings and characters used in the malware program	Behavioral
IDAPro (free version)	Used to dissemble code.	Determine how the program is written and to identify what the program does.	Code
Jad	Utility to convert java class files into java source files.	Analyze any java class files.	Code
LordPE	Edits PE files and can dump the memory.	Use to modify the header for memory dump files. This will insure that the program will run.	Code
IEController-2.0	Controls and monitors connections with Internet Explorer.	Use when investigating web sites. Creates a sandbox for Internet Explorer.	Code
Netcat	Utility to transfer files across a network.	Used to move files from the Linux guest machine to the host machine.	Behavioral
OllyDbg	A debugger with a disassembler.	Used to analysis code and change the code as a program is running.	Code
Plugin-OllyDump	Dumps process memory for debugging.	Used to help in analysis of code and debugging.	Code
RegShot	Creates a copy of the registry and will compare consecutive shots.	Used to see what registry changes are made.	Behavioral
FileMon	Logs the access of files on the system.	Used to determine where, when, and why the malware specimen accesses files.	Behavioral
RegMon	Logs the access to the registry.	Used to determine where and when malware specimen accesses the registry and for what purpose.	Behavioral
TDIMon	Logs and monitor TCP and UDP traffic at the Transport Drive Interface level.	Used to see what connections occur while running a program or malware specimen.	Behavioral
upx	Compresses executable files.	Can be used to uncompress upx files.	Code
Md5sums	Generated the md5 hash file.	Used to generate md5 hash for comparison purposes. Will be able to tell if a file has changed.	Behavioral
WinZip	A file compression utility.	Unzip files and to compress files on the guest operating system.	

Table 1 – Windows Guest Software

Linux Operating System

The guest Linux operating system is considerably easier to install in VMware. Part of the course material included a VMware image of a scaled down Red Hat Linux operating system with preinstalled software. The configuration of the Linux guest computer consisted of a 2 GB SCSI hard drive, 64 MB RAM, only one network interface card configured as a host only connection and to use DHCP, an IDE CD-ROM drive, a USB controller, and an audio device. A list of the initially installed software along with a description, anticipated use, and the area of analysis the software will be used for are given in Table 2 – Linux Guest Software.

Software	Description	Use	Area of Analysis
Snort	Sniffs network traffic.	Capture network packets to determine how machines are talking (TCP/UDP/ARP and Protocol)	Both
IRC	IRC server	Provide an IRC server to malware and attempt to communicate with malware.	Behavioral
Honeyd	A virtual honey pot.	Provide services and protocols that the malware may attempt to communicate with.	Behavioral

TABLE 2 – Linux Guest Software

The final laboratory setup step was to start both guest operating systems and verify the networking settings. Since both are set for DHCP an ipconfig was performed on the Windows guest and an ifconfig was performed on the Linux guest. The Windows guest computer IP address is 192.168.252.128, the Linux guest computer is 192.168.252.129, and the host computer IP address is 192.168.252.1. Finally, pings were performed to each of the guest systems. This ensured that the two machines could talk with each other and recognize each other. Figure 1 – Virtual Laboratory Network, illustrates the virtual laboratory and how the host and guest computers are connected. Now, we can start analyzing msrll.exe.

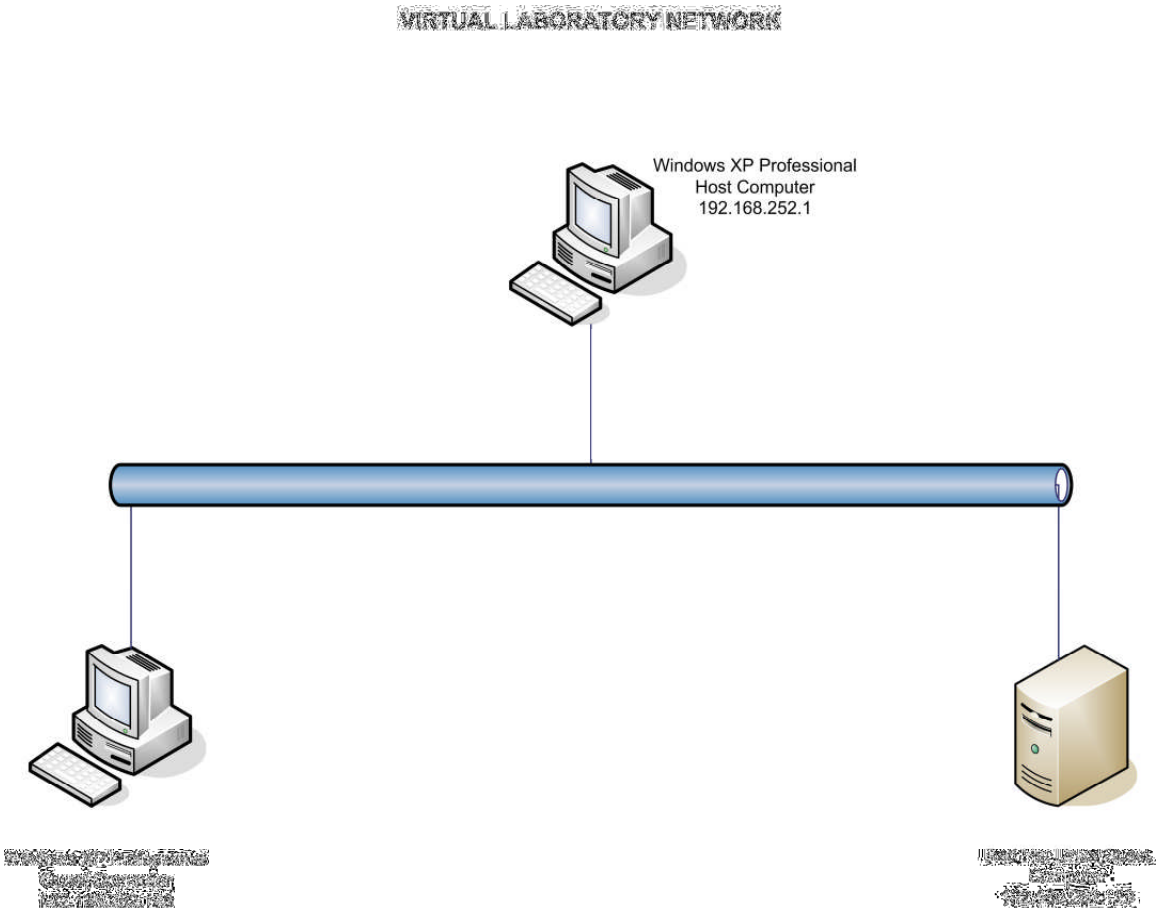

Figure 1 – Virtual Laboratory Network

PROPERTIES OF THE MALWARE SPECIMEN

Let the adventure begin by looking at the properties of the malware specimen. From now on we will be working in the laboratory with both the Windows guest computer and the Linux guest computer. Even though the host computer is a standalone computer, we still must take care in the transferring any files from the computer.

The msrll.zip file was copied to its own directory, d:\msrll, during the preparation phase. Now we double click on the file, msrll.zip, and extract it to c:\msrll\. Now we can look at the file, in a windows explorer window, to determine two of the properties, namely, the type of file, exe, and the size of the file, 41 Kb. The file size reported by PEInfo confirms this by reporting a file size of 41,984 bytes.

PEInfo.exe is a command line tool. I placed PEInfo in the same directory with msrll.exe. Then open a command prompt and change to the directory containing both PEInfo and msrll. The command used was **peinfo.exe msrll.exe > peinformsrll.txt**. This will run PEInfo using the msrll.exe file and place the

output into the peinfomsrll.txt file. Reviewing the peinfomsrll.txt file, PEInfo also provided additional information. The type of machine that the malware specimen will run on is listed as “Machine: 014C, Translation--> Intel 80386 Processor.”

PEInfo also provides a listing of strings in the code. The interesting strings are listed below. A portion of the listing omitted here due to being unreadable. The complete output from PEInfo, including strings, is provided in Appendix A – PEInfo. The first obvious string is “!This program cannot be run in DOS mode.” You will also note that the specimen is packed with Aspack as indicated by the string “.aspack”. This indicates that the information gather on the specimen itself will not be accurate. Finally, note that various dlls are also listed.

PEInfo listing for msrll.exe

```
!This program cannot be run in DOS mode.
.idata
.aspack
.adata
6>HBId
Y^nk•K
.     .
.     .
.     .
Z/rA'`
ga1YAx
kN2$6|[x
VirtualAlloc
VirtualFree
kernel32.dll
ExitProcess
user32.dll
MessageBoxA
wsprintfA
LOADER ERROR
The procedure entry point %s could not be located in the dynamic
link library %s
The ordinal %u could not be located in the dynamic link library
%s
 (08@P`p
kernel32.dll
GetProcAddress
GetModuleHandleA
LoadLibraryA
advapi32.dll
msvcrt.dll
msvcrt.dll
shell32.dll
user32.dll
version.dll
```

```
wininet.dll
ws2_32.dll
AdjustTokenPrivileges
__getmainargs
ShellExecuteA
DispatchMessageA
GetFileVersionInfoA
InternetCloseHandle
WSAGetLastError
```

Now, let's get some additional information about the specimen by looking at the MD5 hash. The MD5 hash requires executing md5sum in a command prompt window. Open a command prompt and change to the directory where md5sum is located. Us the following command to generate the MD5 hash: **md5sum c:\msrll\msrll.exe > md5.txt**. This will execute the md5sum program on the msrll.exe file and output the MD5 hash to a text file called md5.txt. Opening the md5.txt file in notepad we find that the MD5 hash is 84acfe96a98590813413122c12c11aaa. This is useful information in case the malware specimen creates another msrll.exe file elsewhere. If this occurs, we will compute the MD5 hash of the new file and compare it to the original MD5 hash. This will tell us if any changes in the file occur when it copies itself.

Property	Characteristic
Type of file	exe
Size of file	41 Kb
MD5 hash of file	84acfe96a98590813413122c12c11aaa
Operating system(s) it runs on	Windows – W9x, Windows 2000, Windows XP
Interesting strings embedded in it	!This program cannot be run in DOS mode. .aspack various dlls

Table 3 – Summary of Properties of Malware Specimen

BEHAVIORAL ANALYSIS

Now that we have basic property information on the malware specimen we can begin our adventure. First, let's get a shot of the current registry. This is accomplished by running RegShot and I saved this shot into an hiv file.

Next, let's get the monitoring software fired up. We start FileMon, RegMon, and TDIMon. You'll notice that when these programs are started they will start monitoring their respective system area. In each program stop the capturing, clear the screen, and have the log files use the system time. Using the system

time to record the records will aid us in comparing the time events between the logs. This may be important to determine what order events happened. Change over to the Linux guest machine. I created a directory, /ilotx, and subdirectory /snort1 in order for snort to output the captured network traffic in the /ilotx/snort1 directory. The command used to fire up snort is **snort –dev –l /ilotx/snort1**. This will start snort, direct the packets to the screen, use verbose mode and record the packets in the directory /ilotx/snort1. Snort will create subdirectories in /ilotx/snort1 by origin IP address. Within each of these subdirectories, snort will create files based on type of traffic, TCP, UDP, and ARP and also by origin port and destination port. With each running of snort, I create another directory by incrementing the number at the end. Therefore, the second execution of snort will record its log in the /ilotx/snort2 directory.

Now switch back to the Windows guest machine and open Windows Explore and change to the directory containing msrll.exe, namely c:\msrll\. Also, open the task manager so we can be prepared to stop any processes or programs. Start monitoring with FileMon, RegMon, and TDIMon. Finally, double click on msrll.exe. Sit back and watch for approximately 30 seconds.

OH NO! The final disappeared from the c:\msrll directory. What has it done? Quick, switch to task manager, processes tab and find msrll.exe. Right click the process msrll.exe and click on end process. Once msrll.exe process stops, go to each of the monitoring tools and stop the capturing. After this, I save each of the log files for future use. Switch to the Linux guest machine and stop snort by pressing Ctrl B.

Before we start the analysis, take another RegShot and save the file. Click on difference in order for RegShot to generate a listing with the differences in the first and second RegShot. Now, we can start analyzing the information generated by FileMon, RegMon, TDIMon, and RegShot.

File Changes

Now, we can begin analyzing the behavior of msrll.exe. While the program was running we noted that it deleted itself from the c:\msrll directory. What other files has it deleted, changed or added. A listing from FileMon showing the files that are created, deleted, written to, and changed is shown below. This first listing shows msrll.exe copies or writes itself to C:\WINDOWS\System32\mfm\msrll.exe. Of course, msrll.exe created the subdirectory mfm first before writing a copy there. Then msrll.exe will go back and delete the original file in the C:\msrll directory and it will go back to verify that the file was deleted. You will also notice the PID of msrll.exe changes from 1372 to 1408. This was observed in FileMon, RegMon, and TDIMon listings.

462	7:32:30 PM	msrll.exe:1372	WRITE	C:\WINDOWS\System32\mfm\msrll.exe		
	SUCCESS	Offset: 0 Length: 41984				
967	7:32:32 PM	msrll.exe:1408	OPEN	C:\msrll\msrll.exe	SUCCESS	
	Options: Open	Access: All				
968	7:32:32 PM	msrll.exe:1408	DELETE	C:\msrll\msrll.exe	SUCCESS	
969	7:32:32 PM	msrll.exe:1408	CLOSE	C:\msrll\msrll.exe	SUCCESS	
971	7:32:32 PM	msrll.exe:1408	OPEN	C:\msrll\msrll.exe	FILE NOT FOUND	
	Options: Open	Access: All				

This next listing shows msell.exe writing to the file jtram.conf in the directory C:\WINDOWS\system32\mfm. The writing to jtram.conf starts after we stop the process msrll.exe. There appears to be a shutdown routine whenever the process is stopped.

1767	7:33:58 PM	msrll.exe:1408	WRITE	C:\WINDOWS\system32\mfm\jtram.conf		
	SUCCESS	Offset: 0 Length: 53				
1788	7:33:58 PM	msrll.exe:1408	WRITE	C:\WINDOWS\system32\mfm\jtram.conf		
	SUCCESS	Offset: 53 Length: 53				
2169	7:33:59 PM	msrll.exe:1408	WRITE	C:\WINDOWS\system32\mfm\jtram.conf		
	SUCCESS	Offset: 1006 Length: 77				
2170	7:33:59 PM	msrll.exe:1408	WRITE	C:\WINDOWS\system32\mfm\jtram.conf		
	SUCCESS	Offset: 1083 Length: 1				
2171	7:33:59 PM	msrll.exe:1408	CLOSE	C:\WINDOWS\system32\mfm\jtram.conf		
	SUCCESS					
2810	7:34:09 PM	msrll.exe:1408	CLOSE	C:\WINDOWS\System32\mfm	SUCCESS	

Looking at the jtram.conf file, it appears to be encrypted as the listing shows.

```
+P8RAA/6BcPcW82IaTZZvToQ18PaG8Fq0tDEOxoCIbLEN5tvPA==
P/8RAHl7FhUzkoXA9NBub9e0IsoRisbnebTfr3uakWK6Cw8CDA==
HgIRAI9184TgGftDpgr8Pl9JoFH8cAolG1BOgvg8D3n6NJ5oEg==
Ov4RAItTSttxPN2zgQ4FZ4TE01WZxcHS1ukv8+Ql/RbFRWiFmQ==
Yf8RABSSa6ugVlWoFe4UfmN/w5qm7x3A5USXq+8PlkXHTulvfA==
YQARAJ2ptgxFufalmnh0OUzB/Yh8lMBkQzbesUHdbECSYjomBA==
UAIRAFBpoHN8LeTK1a3qOH5vZ22WndptKYhyYI6i6TLm+3cj9A==
7QARAOZADgWXZNOmMsEmAvo6gkOoN7qLxTQnEn9rvWFE8m1iuA==
RgBKAEwmhsd+7zs2/v8hzRAadzxXAAbmxS+p1D/v5OmWHrLPiV90+tX5A+MaEXyDt+RpPbSa
k4fnZe9VMy5OnQ83jAvn4VgaguAbAikFjwCSWYBQx0EJR+b7DvleTg==
RQARAIU6QW5MOziIG6mAEf9qWKQNo+Yg2JE8U0zua/Dis1iEEQ==
4P0RABV6i/ME2XgmanIXm7Wf5qyDhS/N0GwADqm8l/6/Ro7vCQ==
e/0RAJC0MOnOW8eaigCatn+m3ERLOJ3Zcq8ErjLRlKV3tlnd+Q==
yfsRAPjglZH+96wJtbmgjuduvn+VnoyVlEuJrGxpJYhzwb6lYQ==
Ff4RAPAKN69Eg5at+feJxx1S3TFtolA+8usqBSSrxOKKW+Kyxg==
GwIjAAkWFXw0vN9Fy71ARJ8JUmI2EXIFU4/hgleOPmZE+Bl7gAFHvbFuzh1MDsrQ/A9M/Ojkzg
==
```

```
JQIRANHLt0GVbG0xV/o2sUAOf2Dk7mNRENPjH83Mu1tCNqcAQw==
DQIRAOTKxXbisTLp0gAscTf2kQUgenBxdpWFFmKzVk5lZJquSA==
kgljADZJUpfGMVOG2TMlqnKoKXzpKijiPXcYNuObiHjqXGMI1uqiLehZeGG0JRmnPI4i2+/0Yw==
```

Other interesting files are the files that msrll attempted to open but could not find or had other problems are included the list below. The failures included file not found for files and dlls, path not found, and buffer overflow. These file reads along with other interesting file reads are listed in Appendix B – Interesting FileMon Failures.

```
1640    7:33:58 PM      msrll.exe:1408  OPEN   C:\dev\random  PATH NOT FOUND
        Options: Open  Access: All

363     7:32:30 PM      msrll.exe:1372  QUERY INFORMATION          C:\msrll\msrll.exe
        BUFFER OVERFLOW   FileNameInformation

368     7:32:30 PM      msrll.exe:1372  OPEN   C:\WINDOWS\system32\wininet.dll.123.Manifest
        FILE NOT FOUND       Options: Open  Access: All
369     7:32:30 PM      msrll.exe:1372  OPEN   C:\WINDOWS\system32\wininet.dll.123.Config
        FILE NOT FOUND       Options: Open  Access: All
410     7:32:30 PM      msrll.exe:1372  QUERY INFORMATION          C:\msrll\msrll.exe.Local\
        FILE NOT FOUND       Attributes: Error
```

Finally, I did another md5sum on the new msrll.exe file. This is to verify that it is the same file or if the file is changed. The file has the same md5 hash as the original file. Next, let's focus our attention to the results of RegMon and RegShot.

Registry Changes

First, let's examine the results of RegMon for changes made by the process msrll.exe. After examining the changes listed by RegMon, we will examine the results of RegShot. This will be helpful in that it will confirm the changes reported by RegMon. There are numerous registry keys and values that were read by msrll.exe. The keys we are most interested in are the created keys by msrll.exe. All created keys by msrll.exe are listed in Appendix C – Created Keys. One interesting observation is that numerous keys were repeatedly created. The registry keys that were created are listed below with duplicate creates deleted.

```
HKLM\SOFTWARE\Microsoft\Cryptography\RNG
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
```

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{f7035723-f4b0-11d6-
b29f-806d6172696f}\
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{a99e5349-f4f7-11d6-
93ec-005056400081}\
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{f7035721-f4b0-11d6-
b29f-806d6172696f}\
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{091136c0-7a8e-
11d8-941f-000c29e6fb6b}\
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\Z
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
HKLM\Software\Microsoft\Tracing
HKCU\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
HKCU\Software\Microsoft\windows\CurrentVersion\Internet Settings
HKCU\Software\Microsoft\windows\CurrentVersion\Internet Settings\Connections
HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
```

Another interesting observation is the attempted deletion of key values as shown. These proxy settings were not deleted since a proxy server was not setup, therefore the values were not there. This indicates that msrll.exe does not what the use of a proxy server.

```
HKCU\Software\Microsoft\windows\CurrentVersion\Internet Settings\ProxyServer
        NOTFOUND
HKCU\Software\Microsoft\windows\CurrentVersion\Internet Settings\ProxyOverride
        NOTFOUND
HKCU\Software\Microsoft\windows\CurrentVersion\Internet Settings\AutoConfigURL
        NOTFOUND
```

Compare these results with the RegShot results. RegShot will show the number of keys added, number of values added, and the number of values modified. This is only a before and after comparison of the registry and will not show which process added or changed the key or value. Since I only focused on the process msrll.exe with RegMon, only those values that msrll.exe changed are examined. RegShot does show some additional information as shown.

```
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\mfm
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\mfm\Security
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\mfm
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\mfm\Security
```

The significant of looking at changes to the registry is to determine what and how msrll is impacting our system. Obviously, msrll is added as a process and will

start and stop when the system is turned on and off. Now let's turn our attention to the network connections that msrll.exe attempts to make.

Network Connections

The network connections were monitored by TDIMon. Unfortunately, the TDIMon log did not provide much useful information expect that msrll was interested in TCP ports 113 and 2200. To get the full implications of what network connects are being made we need to look at the packet sent while msrll is running. Along to the rescue is our good friend snort, running on the Linux guest machine.

This first listing form the snort logs shows that the Windows guest computer is attempting to resolve the address for collective7.zxy0.com by attempting to connect to a DNS server on port 53.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/26-00:10:45.068204 0:C:29:B8:DB:B4 -> 0:50:56:C0:0:1 type:0x800 len:0x50
192.168.252.128:1033 -> 192.168.252.1:53 UDP TTL:128 TOS:0x0 ID:32 IpLen:20 DgmLen:66
Len: 38
00 02 01 00 00 01 00 00 00 00 00 00 0B 63 6F 6C  .............col
6C 65 63 74 69 76 65 37 04 7A 78 79 30 03 63 6F  lective7.zxy0.co
6D 00 00 01 00 01                                m.....
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

The msrll process is stopped and the host file on the Windows machine is changed to make the msrll think collective7.zxy0.com address is 192.168.252.129, the Linux machine. Start snort on the Linux machine and then switch over to the Windows machine and start msrll.exe. The new location of msrll.exe is C:\Windows\System32\mfm. Let the process run for about 30 seconds and then stop the process through the task manager. Switch to the Linux machine and stop snort. Now, let's look at the log files.

Here are several attempts to connect to ports which did not show up in the previous snort log file. Once the collective7.zxy0.com address was resolved, attempts were made to connect to ports 6667, 9999, and 8080. These ports are bolded in the listing below. Port 6667 is normally used for IRC, port 8080 is normally used as a web or http, and we are not sure about port 9999.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/24-06:12:02.012874 192.168.252.128:1046 -> 192.168.252.129:6667
TCP TTL:128 TOS:0x0 ID:454 IpLen:20 DgmLen:48 DF
******S* Seq: 0xF7CCD404  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/24-06:12:02.014171 192.168.252.129:6667 -> 192.168.252.128:1046
```

```
TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF
***A*R** Seq: 0x0  Ack: 0xF7CCD405  Win: 0x0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/24-06:12:33.040674 192.168.252.128:1047 -> 192.168.252.129:9999
TCP TTL:128 TOS:0x0 ID:467 IpLen:20 DgmLen:48 DF
******S* Seq: 0xF843D62B  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/24-06:12:33.040987 192.168.252.129:9999 -> 192.168.252.128:1047
TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF
***A*R** Seq: 0x0  Ack: 0xF843D62C  Win: 0x0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/24-06:12:59.077109 192.168.252.128:1048 -> 192.168.252.129:8080
TCP TTL:128 TOS:0x0 ID:470 IpLen:20 DgmLen:48 DF
******S* Seq: 0xF8A7F3AF  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/24-06:12:59.077437 192.168.252.129:8080 -> 192.168.252.128:1048
TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF
***A*R** Seq: 0x0  Ack: 0xF8A7F3B0  Win: 0x0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Since msrll is attempting to connect to an IRC server, I started the IRC and restarted snort on the Linux machine. On the Windows machine, I restarted msrll.exe. The snort log files below shows msrll.exe connecting with the IRC server with a user name of GVmMMTUJVT and then joins the channel mils. I also noticed that the user name is randomly generated. Each time msrll.exe is started, it will connect using a different user name. Yet each time it connects, it will always join the channel mils.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/12-01:32:12.108986 0:C:29:B8:DB:B4 -> 0:C:29:56:DA:DF type:0x800 len:0x86
192.168.252.128:1073 -> 192.168.252.129:6667 TCP TTL:128 TOS:0x0 ID:6009 IpLen:20 DgmLen:120 DF
***AP*** Seq: 0xA1A8E80A  Ack: 0x49C21F4E  Win: 0x4421  TcpLen: 20
55 53 45 52 20 54 68 48 51 55 59 76 46 64 61 62     USER ThHQUYvFdab
20 6C 6F 63 61 6C 68 6F 73 74 20 30 20 3A 4A 54   localhost 0 :JT
6B 4E 71 58 63 61 63 54 48 57 6A 73 52 77 74 54    kNqXcacTHWjsRwtT
74 49 63 44 74 4D 6C 6B 52 42 6F 43 77 4D 68 0A   tIcDtMlkRBoCwMh.
4E 49 43 4B 20 47 56 6D 4D 4D 54 55 4A 56 54 0A  NICK GVmMMTUJVT.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/12-01:32:44.080649 0:C:29:B8:DB:B4 -> 0:C:29:56:DA:DF type:0x800 len:0x43
192.168.252.128:1073 -> 192.168.252.129:6667 TCP TTL:128 TOS:0x0 ID:6028 IpLen:20 DgmLen:53 DF
***AP*** Seq: 0xA1A8E86D  Ack: 0x49C225BE  Win: 0x4422  TcpLen: 20
4A 4F 49 4E 20 23 6D 69 6C 73 20 3A 0A          JOIN #mils :.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

I tried numerous times to communicate with msrll with no success. Since msrll is listening on port 2200, I attempted to telnet to the port as well as using netcat. It appeared that I would connect, but I could not get a response. I also attempted to communicate through the IRC and again I would not get a response.

Now would be a good time to start the code analysis. What are some of the secrets that we will look for? First, what is the jtram.conf file used for. Since encryption appears to be used, what is the encryption routine used. Can we decrypt the file? How do you communicate with msrll? What other connections does msrll attempt to make?

CODE ANALYSIS

The previous behavioral analysis showed an interesting string in the msrll.exe file, namely .aspack. Executable files are packed for several reasons, to make it compressed for quicker and easier distribution over the internet, to make it more difficult to unpack and analyze, or to make it impossible to analyze the code. First, an attempt was made to analysis the code with IDAPro. These results were not encouraging. IDAPro produced unusable disassemble of the code. The hex view in IDAPro has sections filled with "?" marks as the listing shows and thus not useable.

```
.idata:0051CCA0  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ?? "????????????????"
.idata:0051CCB0  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ?? "????????????????"
.idata:0051CCC0  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ?? "????????????????"
.idata:0051CCD0  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ?? "????????????????"
.idata:0051CCE0  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ?? "????????????????"
.idata:0051CCF0  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ?? "????????????????"
.idata:0051CD00  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ?? "????????????????"
.idata:0051CD10  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ?? "????????????????"
.idata:0051CD20  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ?? "????????????????"
```

Off to the internet we go to search for an Aspack unpacker. This is accomplished by using another computer that is connected to the internet and doing a Google search on "disassemble aspack" to find what tools are available to unpack the msrll.exe file focusing mainly on AspackDie. Why AspackDie? In the course material, I found a reference to AspackDie which stated "Here is an example of one tool, called AspackDie, which worked well when I tried to unpack a malicious executable protected by the Aspack utility."[1] I found several sites with AspackDie. It is important that the unpacker be compatible with the version of the packer. This may be a trail and error exercise. The site that I downloaded AspackDie 1.41, written by yoda, from is http://protools.anticrack.de/unpackers.htm.

I then transferred the zipped file to the laboratory Windows guest machine. I unzipped it into its own directory on the d:\ drive. The readme.tXt file that came with the program states that this will unpack all Aspack since 2000[2]. I ran the

[1] Zeltser, p 1-38.
[2] yoda.

program, AspackDie, and opened msrll.exe. When AspackDie was finished I save the file as unpacked.exe in its own directory, d:\unpack. The test to see if this unpacked code is correct, I executed unpack.exe. If the code is the code that was packed with Aspack, then it should show the same behavior as msrll.exe. The behavior of the unpacked code was the same as the pack code. Therefore, I will examine this code.

First, examine the code using bintext. Bintext reveals more information than the previous string searches. We are able to find the servers that msrll tries to communicate with as the listing shows. The listing indicates that it uses collective7.zxy.com.

```
0000BD6E   0040BD6E     0   servers
0000BD80   0040BD80     0
collective7.zxy0.com,collective7.zxy0.com:9999!,collective7.zxy0.com:8080
```

Running the unpacked code through IDAPro provides more useful information. There are numerous jumps, subroutines, and functions that msrll uses. There are also numerous calls to the following dlls, ADVAPI32, KERNEL32, msvcrt, SHELL32, USER32, VERSION, WINNET, and WS2_32. Sections of the code pertain to irc, SSL, and reporting statistics. Msrll also reads from the file c:\dev\random. Initially, I thought that the reference to /dev/random was just a mistake in the coding. In that the writer had mixed some Unix/Linux code with Windows code. I created the file, C:\dev\random, with just 0's and then ran the program. It appears msrll uses this file to encrypt the file jtram.conf. By changing the contents of C:\dev\random and rerunning msrll, the contents of jtram.conf changes. I have not observed this behavior before.

Possible commands used to communicate with msrll where located. When I tried the various commands on the irc channel, I still did not get a response from the Windows computer. The commands found are listed below.

?clone	?clones	?login
?uptime	?reboot	?status
?jump	?nick	?echo
?hush	?wget	?join
?akick	?part	?dump
?md5p	?free	?update
?hostname	?!fif	?play
?copy	?move	?sums
?rmdir	?mkdir	?exec
?kill	?killall	?crash
?sklist	?unset	?uattr
?dccsk	?killsk	

Let's turn to using debugged, namely OllDbg. Load the unpacked version of msrll into OllDbg. After it is loaded, OllDbg will pause on the entry point. Let's see if we can determine if there is a login command with a password using the irc server. Start the Linux machine and the irc server. Next, search the code for string compare commands. The first one that is found is actually a call command jumping to msvcrt._stricmp function, as shown by the following code: **00401462 |. E8 990B0100 |CALL <JMP.&msvcrt._stricmp> ; _stricmp.** Instead of setting a breakpoint on line 0040162, I found the function **msvcrt._stricmp 00412000 $-FF25 F0B45100 JMP DWORD PTR DS:[<&msvcrt._stricmp>] ; msvcrt._stricmp**, and set the breakpoint there. This way any calls to msvcrt._stricmp will cause the program to pause. While looking at the functions, I find the following lines, **00412280 $-FF25 90B55100 JMP DWORD PTR DS:[<&msvcrt.strcmp>]** ; msvcrt.strcmp, and **004122A0 $-FF25 A0B55100 JMP DWORD PTR DS:[<&msvcrt.strncmp>] ; msvcrt.strncmp**, and set breakpoints at these lines as well. To set a breakpoint, highlight the line and press the F2 key. This will toggle on and off the breakpoint.

Once these breakpoints are set, they will show in the Breakpoints window. Using the Breakpoints window, highlight the breakpoints and toggle them off. Start running the program and switch to the Linux machine. Monitor the Linux machine until you see a random user log into the irc channel. Switch back to the Windows machine and pause OllDbg by pressing the F12 key. Now, toggle the breakpoints on and continue running the program in OllDbg by pressing the Ctrl and the F2 keys. Each time a string is compared the program will pause. Now, try logging into the Windows machine by typing **?login password**. Switch back to the Windows machine and wait for a breakpoint to respond by pausing. Look for the values for the strings being compared. If you see password then the string that it is being compared with should be the password to login with. Unfortunately, I was not able to obtain the login password.

I attempted to determine what is contained in the jtram.conf file. I set breakpoints for fread, file reads, and ran the program. I was hoping to find when msrll would read the c:\dev\random file and then step through commands one at a time by pressing the F7 key. Stepping through the commands should reveal where this data is used. Again, I was unable to find anything significant. Rats!!

ANALYSIS WRAP-UP

It has been a long adventurous journey. First, time was taken to discuss the laboratory setup. Whenever you deal with malware, utmost care must be taken to insure that the laboratory is isolated from the network. The worst that can happen, when the laboratory is isolated, is having to discard or rebuild a machine. Even though, VMware is used for the laboratory, care still must be taken. I have accepted the risk that the malware may infect the host computer. This is a manageable risk.

The setup of the actual virtual machines was also discussed. I find it to be easier to have all the possible tools that could be used preinstalled. This way, in the heat of the analysis, you do not have to stop and search for any particular tool. Although this was not the case, since I had to search for an unpacking utility to unpack the malware specimen. I did not anticipate that the specimen was packed using Aspack and most people may not anticipate it. Yet, I was able to find an unpacking utility that worked and even found a web site that has many unpacking utilities for future use.

The behavior analysis was able to tell us a lot about the malware specimen. We found where it installed itself to, the registry keys that were created, modified, and accessed. We were also able to analyze the networking desire of the malware specimen through the use of snort. We were also able to provide reasonable accommodations of the networking requests from the malware specimen.

The code analysis yielded more information. We were able to find the actual computer name that the malware specimen wanted to communicate with. The specimen only wanted to communicate with one machine on multiple ports.

Based on what has been observed and found, the malware specimen loads a backdoor on the computer, gathers information, and prepares a file. This file probably contains information useful for identifying the computer that would be useful for further mischievous activities. Either the backdoor or the irc channel can be used for communication.

There are several ways to defend against such malware. The rule of least privilege for users should be used. Users should only be users and not have the privileges to install software, access the registry, or write to the system area. The use of firewalls on the corporate network as well as on the PC to block undesirable inbound and outbound ports will help to protect unsuspecting users. The use of antivirus software that is regularly updated will help in protecting the user from malware. Antivirus software is only as good as the virus definitions. A newly released virus, Trojan, backdoor, worm, etc might not be detected by the antivirus software. Finally, some sort of intrusion detection system may also be used to detect and prevent backdoors. This would produce a defense in depth.

APPENDIX A - PEINFO

```
Path: D:\Msrll\msrll.exe
     File size:   41984
    Image size:   1179648
File Alignment:   512
Resources account for 0.00% of the executable

Issues:
=======
String: GetProcAddress                       (1)
String: LoadLibrary                          (1)

***** HEADER *****


Machine:                      014C
     Translation--> Intel 80386 Processor
NumberOfSections    :         0006
TimeDateStamp       :         40790135
     Created (GMT): Sun Apr 11 08:26:29 2004

PointerToSymbolTable:         00000000
NumberOfSymbols:              00000000
SizeOfOptionalHeader:         00E0
Magic:                        010B
SizeOfCode:                   00011800
SizeOfInitializedData:        00014600
SizeOfUninitializedData:      00105C00
AddressOfEntryPoint:          0011D001
BaseOfCode:                   00001000
BaseOfData:                   00013000
ImageBase:                    00400000
SectionAlignment:             00001000
FileAlignment   :             00000200
LinkerVersion:                2.56
OperatingSystemVersion:       4.00
ImageVersion:                 1.00
SubsystemVersion:             4.00
Win32VersionValue:            00000000
SizeOfImage                   00120000
SizeOfHeaders:                00000400
CheckSum:                     00017803
Subsystem:                    0002
     Translation--> Windows GUI
DllCharacteristics:           0000
SizeOfStackReserve:           00200000
SizeOfStackCommit:            00001000
SizeOfHeapReserve:            00100000
SizeOfHeapCommit:             00001000
LoaderFlags:                  00000000
NumberOfRvaAndSizes:          00000010
Characteristics:              020F
     (non-32-bit-word machine)
```

```
      Bytes of machine word are not reversed
      Relocation info stripped
      Line numbers stripped
      Local symbols stripped
      Debugging info stripped into .dbg file
      Need not copy to swapfile if run from removable media
      Need not copy to swapfile if run from network
      Runs on MP or UP machine
      Working set trimmed normaly
      Executable file
      Not a system file
      Not a DLL


***** DATA DIRECTORY *****


                    VAddress    Size
                   --------    --------
       Export:         00000000      00000000
       Import:         0051dfac      000001F8
       Resource:       00000000      00000000
       Exception:      00000000      00000000
       Security:       00000000      00000000
       Relocation:     0051df54      00000008
       Debug:          00000000      00000000
       Architecture:   00000000      00000000
       GlobalPtr:      00000000      00000000
       TLS:            00000000      00000000
       LoadConfig:     00000000      00000000
       BoundImport:    00000000      00000000
       IAT:            00000000      00000000

Section Name:     .text
VirtualAddress:   00401000
VirtualSize:      00012000 (73728)
SizeOfRawData:    00008000 (32768)
PointerToRawData: 00000400
Section characteristics:
       Contains initialized data
       Default alignment (16 bytes)
       Is readable
       Is writeable

Section Name:     .data
VirtualAddress:   00413000
VirtualSize:      00002000 (8192)
SizeOfRawData:    00000600 (1536)
PointerToRawData: 00008400
Section characteristics:
       Contains initialized data
       Default alignment (16 bytes)
       Is readable
       Is writeable

Section Name:     .bss
VirtualAddress:   00415000
```

```
VirtualSize:      00105B70 (1071984)
SizeOfRawData:    00000000 (0)
PointerToRawData: 00000000
Section characteristics:
      Contains initialized data
      Default alignment (16 bytes)
      Is readable
      Is writeable

Section Name:     .idata
VirtualAddress:   0051B000
VirtualSize:      00002000 (8192)
SizeOfRawData:    00000800 (2048)
PointerToRawData: 00008A00
Section characteristics:
      Contains initialized data
      Default alignment (16 bytes)
      Is readable
      Is writeable

Section Name:     .aspack
VirtualAddress:   0051D000
VirtualSize:      00002000 (8192)
SizeOfRawData:    00001200 (4608)
PointerToRawData: 00009200
Section characteristics:
      Contains initialized data
      Default alignment (16 bytes)
      Is readable
      Is writeable

Section Name:     .adata
VirtualAddress:   0051F000
VirtualSize:      00001000 (4096)
SizeOfRawData:    00000000 (0)
PointerToRawData: 0000A400
Section characteristics:
      Contains initialized data
      Default alignment (16 bytes)
      Is readable
      Is writeable

Import Name: kernel32.dll
Name:                         0051DF6C
Characteristics:              00400000
TimeDateStamp:                00000000
Not bound

Thunk       Ordinal    Name
--------    -------    ---------------
0011DF64        0      LoadLibraryA
0011DF60        0      GetModuleHandleA
0011DF5C        0      GetProcAddress

Import count: 3

Import Name: advapi32.dll
```

```
Name:                           0051E074
Characteristics:                00400000
TimeDateStamp:                  00000000
Not bound

Thunk       Ordinal    Name
--------    -------    ---------------
0011E0D1        0      AdjustTokenPrivileges

Import count: 1

Import Name: msvcrt.dll
Name:                           0051E081
Characteristics:                00400000
TimeDateStamp:                  00000000
Not bound

Thunk       Ordinal    Name
--------    -------    ---------------
0011E0D9        0      _itoa

Import count: 1

Import Name: msvcrt.dll
Name:                           0051E08C
Characteristics:                00400000
TimeDateStamp:                  00000000
Not bound

Thunk       Ordinal    Name
--------    -------    ---------------
0011E0E1        0      __getmainargs

Import count: 1

Import Name: shell32.dll
Name:                           0051E097
Characteristics:                00400000
TimeDateStamp:                  00000000
Not bound

Thunk       Ordinal    Name
--------    -------    ---------------
0011E0E9        0      ShellExecuteA

Import count: 1

Import Name: user32.dll
Name:                           0051E0A3
Characteristics:                00400000
TimeDateStamp:                  00000000
Not bound

Thunk       Ordinal    Name
--------    -------    ---------------
0011E0F1        0      DispatchMessageA
```

```
Import count: 1

Import Name: version.dll
Name:                           0051E0AE
Characteristics:                00400000
TimeDateStamp:                  00000000
Not bound

Thunk       Ordinal    Name
--------    -------    ---------------
0011E0F9        0      GetFileVersionInfoA

Import count: 1

Import Name: wininet.dll
Name:                           0051E0BA
Characteristics:                00400000
TimeDateStamp:                  00000000
Not bound

Thunk       Ordinal    Name
--------    -------    ---------------
0011E101        0      InternetCloseHandle

Import count: 1

Import Name: ws2_32.dll
Name:                           0051E0C6
Characteristics:                00400000
TimeDateStamp:                  00000000
Not bound

Thunk       Ordinal    Name
--------    -------    ---------------
0011E109        0      WSAGetLastError

Import count: 1

Strings:
--------
!This program cannot be run in DOS mode.
.idata
.aspack
.adata
6>HBId
Y^nk•K
X•l?%A
\+VS`%
Y8EoM,
gPtL7S
YQ(W;n
oukd••
3#b5pHo
A^[jK<
w3i5Y-
[u)aH=
/0mo0^
```

```
Bj3K7%(
yko`w+r
•TN9x0
U[{*\4
m&8NRM
8e47xW
EAe4xIpO
r8cy!/
127$9v
zYX[[T
PS=,sdVQ
UZKSU,
5OUS</
%XjBZnu
:|gs3~3
s&+*uX
L,HvCy
wZMFN_
y!]zqZ
s$ILIEK
'.gcH(
PQiqGt
?Q~)Qv
Y|5S(K
0]2%I^
>~g[f!Unl
xaa11K
d{fB0d•^G
s$OY5-
s*9r\sN
Z3O-,;
kvK@~G
ek^{}P
}vlt•&E?
PAPD;-
xCd4!c
.`gmRx[
 M'L s
I$^!%8
xq,p:j
bn;&%y
y[:BaV_
Yqc*Jam
GMZid+K
bI4x+Za
Z/rA'`
ga1YAx
kN2$6|[x
VirtualAlloc
VirtualFree
kernel32.dll
ExitProcess
user32.dll
MessageBoxA
wsprintfA
LOADER ERROR
```

```
The procedure entry point %s could not be located in the dynamic link
library %s
The ordinal %u could not be located in the dynamic link library %s
 (08@P`p
kernel32.dll
GetProcAddress
GetModuleHandleA
LoadLibraryA
advapi32.dll
msvcrt.dll
msvcrt.dll
shell32.dll
user32.dll
version.dll
wininet.dll
ws2_32.dll
AdjustTokenPrivileges
__getmainargs
ShellExecuteA
DispatchMessageA
GetFileVersionInfoA
InternetCloseHandle
WSAGetLastError
```

APPENDIX B – INTERESTING FILEMON FAILURES

```
1640  7:33:58 PM  msrll.exe:1408    OPEN  C:\dev\random     PATH NOT
FOUND Options: Open  Access: All
1526  7:32:48 PM  msrll.exe:1408    OPEN  C:\Documents and
Settings\Administrator\Application
Data\Microsoft\Network\Connections\Pbk\   PATH NOT FOUND    Options:
Open Directory  Access: All
1493  7:32:48 PM  msrll.exe:1408    DIRECTORY   C:\Documents and
Settings\All Users\Application Data\Microsoft\Network\Connections\Pbk\
      NO SUCH FILE      FileBothDirectoryInformation: *.pbk
611   7:32:31 PM  msrll.exe:1372    QUERY INFORMATION
      C:\msrll\CLBCATQ.DLL    FILE NOT FOUND    Attributes: Error
615   7:32:31 PM  msrll.exe:1372    QUERY INFORMATION
      C:\msrll\COMRes.dll     FILE NOT FOUND    Attributes: Error
363   7:32:30 PM  msrll.exe:1372    QUERY INFORMATION
      C:\msrll\msrll.exe      BUFFER OVERFLOW   FileNameInformation
971   7:32:32 PM  msrll.exe:1408    OPEN  C:\msrll\msrll.exe      FILE
NOT FOUND   Options: Open  Access: All
243   7:32:30 PM  msrll.exe:1372    QUERY INFORMATION
      C:\msrll\msrll.exe.Local      FILE NOT FOUND    Attributes: Error
302   7:32:30 PM  msrll.exe:1372    QUERY INFORMATION
      C:\msrll\msrll.exe.Local\     FILE NOT FOUND    Attributes: Error
410   7:32:30 PM  msrll.exe:1372    QUERY INFORMATION
      C:\msrll\msrll.exe.Local\     FILE NOT FOUND    Attributes: Error
671   7:32:31 PM  msrll.exe:1372    QUERY INFORMATION
      C:\msrll\msrll.exe.Local\     FILE NOT FOUND    Attributes: Error
482   7:32:30 PM  msrll.exe:1372    QUERY INFORMATION
      C:\msrll\netapi32.dll   FILE NOT FOUND    Attributes: Error
675   7:32:31 PM  msrll.exe:1372    QUERY INFORMATION
      C:\msrll\Secur32.dll    FILE NOT FOUND    Attributes: Error
494   7:32:31 PM  msrll.exe:1372    QUERY INFORMATION
      C:\msrll\SETUPAPI.dll   FILE NOT FOUND    Attributes: Error
477   7:32:30 PM  msrll.exe:1372    QUERY INFORMATION
      C:\msrll\UxTheme.dll    FILE NOT FOUND    Attributes: Error
250   7:32:30 PM  msrll.exe:1372    QUERY INFORMATION
      C:\msrll\ws2_32.dll     FILE NOT FOUND    Attributes: Error
254   7:32:30 PM  msrll.exe:1372    QUERY INFORMATION
      C:\msrll\WS2HELP.dll    FILE NOT FOUND    Attributes: Error
732   7:32:31 PM  msrll.exe:1372    OPEN
      C:\WINDOWS\AppPatch\systest.sdb     FILE NOT FOUND    Options:
Open  Access: All
1227  7:32:47 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\libssl32.dll FILE NOT FOUND    Attributes: Error
1230  7:32:47 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\libssl32.dll FILE NOT FOUND    Attributes: Error
782   7:32:31 PM  msrll.exe:1408    OPEN
      C:\WINDOWS\Prefetch\MSRLL.EXE-03966588.pf FILE NOT FOUND
      Options: Open  Access: All
241   7:32:30 PM  msrll.exe:1372    OPEN
      C:\WINDOWS\Prefetch\MSRLL.EXE-2C7795E2.pf FILE NOT FOUND
      Options: Open  Access: All
1226  7:32:47 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\system\libssl32.dll      FILE NOT FOUND    Attributes:
Error
```

```
1225  7:32:47 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\libssl32.dll    FILE NOT FOUND    Attributes:
Error
1229  7:32:47 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\system32\libssl32.dll    FILE NOT FOUND    Attributes:
Error
1010  7:32:32 PM  msrll.exe:1408    CREATE      C:\WINDOWS\System32\mfm
      NAME COLLISION    Options: Create Directory  Access: All
1532  7:32:48 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\DNSAPI.dll  FILE NOT FOUND    Attributes:
Error
1232  7:32:47 PM  msrll.exe:1408    OPEN
      C:\WINDOWS\system32\mfm\jtram.conf  FILE NOT FOUND    Options:
Open  Access: All
1636  7:33:58 PM  msrll.exe:1408    OPEN
      C:\WINDOWS\system32\mfm\jtram.conf  FILE NOT FOUND    Options:
Open  Access: All
224   7:32:47 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\libssl32.dll      FILE NOT FOUND
      Attributes: Error
1228  7:32:47 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\system32\mfm\libssl32.dll      FILE NOT FOUND
      Attributes: Error
917   7:32:32 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\msrll.exe   BUFFER OVERFLOW
      FileNameInformation
784   7:32:32 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\msrll.exe.Local   FILE NOT FOUND
      Attributes: Error
839   7:32:32 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\msrll.exe.Local\  FILE NOT FOUND
      Attributes: Error
964   7:32:32 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\msrll.exe.Local\  FILE NOT FOUND
      Attributes: Error
1456  7:32:48 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\msrll.exe.Local\  FILE NOT FOUND
      Attributes: Error
778   7:32:31 PM  msrll.exe:1372    OPEN
      C:\WINDOWS\System32\mfm\msrll.exe.Manifest      FILE NOT FOUND
      Options: Open  Access: All
1396  7:32:48 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\NETAPI32.dll      FILE NOT FOUND
      Attributes: Error
1632  7:33:58 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\rasadhlp.dll      FILE NOT FOUND
      Attributes: Error
1388  7:32:48 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\RASAPI32.DLL      FILE NOT FOUND
      Attributes: Error
1392  7:32:48 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\rasman.dll  FILE NOT FOUND    Attributes:
Error
1404  7:32:48 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\rtutils.dll FILE NOT FOUND    Attributes:
Error
```

```
1315  7:32:48 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\Secur32.dll FILE NOT FOUND    Attributes:
Error
1459  7:32:48 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\sensapi.dll FILE NOT FOUND    Attributes:
Error
1400  7:32:48 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\TAPI32.dll  FILE NOT FOUND    Attributes:
Error
1408  7:32:48 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\WINMM.dll   FILE NOT FOUND    Attributes:
Error
785   7:32:32 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\ws2_32.dll  FILE NOT FOUND    Attributes:
Error
789   7:32:32 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\mfm\WS2HELP.dll FILE NOT FOUND    Attributes:
Error
1496  7:32:48 PM  msrll.exe:1408    DIRECTORY
      C:\WINDOWS\System32\Ras\      NO SUCH FILE
      FileBothDirectoryInformation: *.pbk
261   7:32:30 PM  msrll.exe:1372    OPEN
      C:\WINDOWS\system32\shell32.dll.124.Config      FILE NOT FOUND
      Options: Open  Access: All
797   7:32:32 PM  msrll.exe:1408    OPEN
      C:\WINDOWS\system32\shell32.dll.124.Config      FILE NOT FOUND
      Options: Open  Access: All
260   7:32:30 PM  msrll.exe:1372    OPEN
      C:\WINDOWS\system32\shell32.dll.124.Manifest    FILE NOT FOUND
      Options: Open  Access: All
796   7:32:32 PM  msrll.exe:1408    OPEN
      C:\WINDOWS\system32\shell32.dll.124.Manifest    FILE NOT FOUND
      Options: Open  Access: All
1415  7:32:48 PM  msrll.exe:1408    OPEN
      C:\WINDOWS\System32\TAPI32.dll.124.Config FILE NOT FOUND
      Options: Open  Access: All
1414  7:32:48 PM  msrll.exe:1408    OPEN
      C:\WINDOWS\System32\TAPI32.dll.124.Manifest     FILE NOT FOUND
      Options: Open  Access: All
630   7:32:31 PM  msrll.exe:1372    OPEN
      C:\WINDOWS\system32\urlmon.dll.123.Config FILE NOT FOUND
      Options: Open  Access: All
629   7:32:31 PM  msrll.exe:1372    OPEN
      C:\WINDOWS\system32\urlmon.dll.123.Manifest     FILE NOT FOUND
      Options: Open  Access: All
1231  7:32:47 PM  msrll.exe:1408    QUERY INFORMATION
      C:\WINDOWS\System32\Wbem\libssl32.dll     FILE NOT FOUND
      Attributes: Error
369   7:32:30 PM  msrll.exe:1372    OPEN
      C:\WINDOWS\system32\wininet.dll.123.Config      FILE NOT FOUND
      Options: Open  Access: All
922   7:32:32 PM  msrll.exe:1408    OPEN
      C:\WINDOWS\system32\wininet.dll.123.Config      FILE NOT FOUND
      Options: Open  Access: All
368   7:32:30 PM  msrll.exe:1372    OPEN
      C:\WINDOWS\system32\wininet.dll.123.Manifest    FILE NOT FOUND
      Options: Open  Access: All
```

```
921   7:32:32 PM  msrll.exe:1408    OPEN
      C:\WINDOWS\system32\wininet.dll.123.Manifest    FILE NOT FOUND
      Options: Open  Access: All
321   7:32:30 PM  msrll.exe:1372    OPEN
      C:\WINDOWS\WindowsShell.Config      FILE NOT FOUND    Options:
Open  Access: All
861   7:32:32 PM  msrll.exe:1408    OPEN
      C:\WINDOWS\WindowsShell.Config      FILE NOT FOUND    Options:
Open  Access: All
```

APPENDIX C – CREATED KEYS

```
989   7:32:30 PM  msrll.exe:1372    CreateKey
      HKLM\SOFTWARE\Microsoft\Cryptography\RNG  SUCCESS     Key:
0xE1914B30
1027  7:32:30 PM  msrll.exe:1372    CreateKey
      HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings
      SUCCESS     Key: 0xE15C2450
1299  7:32:30 PM  msrll.exe:1372    CreateKey
      HKLM\SOFTWARE\Microsoft\Cryptography\RNG  SUCCESS     Key:
0xE1546020
1302  7:32:30 PM  msrll.exe:1372    CreateKey
      HKLM\SOFTWARE\Microsoft\Cryptography\RNG  SUCCESS     Key:
0xE1546020
1305  7:32:30 PM  msrll.exe:1372    CreateKey
      HKLM\SOFTWARE\Microsoft\Cryptography\RNG  SUCCESS     Key:
0xE1546020
1308  7:32:30 PM  msrll.exe:1372    CreateKey
      HKLM\SOFTWARE\Microsoft\Cryptography\RNG  SUCCESS     Key:
0xE1546020
1311  7:32:30 PM  msrll.exe:1372    CreateKey
      HKLM\SOFTWARE\Microsoft\Cryptography\RNG  SUCCESS     Key:
0xE1546020
1314  7:32:30 PM  msrll.exe:1372    CreateKey
      HKLM\SOFTWARE\Microsoft\Cryptography\RNG  SUCCESS     Key:
0xE1546020
1317  7:32:30 PM  msrll.exe:1372    CreateKey
      HKLM\SOFTWARE\Microsoft\Cryptography\RNG  SUCCESS     Key:
0xE1546020
1506  7:32:31 PM  msrll.exe:1372    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User
Shell Folders     SUCCESS     Key: 0xE1FCC1B0
1509  7:32:31 PM  msrll.exe:1372    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell
Folders     SUCCESS     Key: 0xE1FCC1B0
1842  7:32:31 PM  msrll.exe:1372    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoin
ts2\{f7035723-f4b0-11d6-b29f-806d6172696f}\     SUCCESS     Key:
0xE1FCC1B0
1845  7:32:31 PM  msrll.exe:1372    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoin
ts2\{a99e5349-f4f7-11d6-93ec-005056400081}\     SUCCESS     Key:
0xE1FCC1B0
1848  7:32:31 PM  msrll.exe:1372    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoin
ts2\{f7035721-f4b0-11d6-b29f-806d6172696f}\     SUCCESS     Key:
0xE1FCC1B0
1851  7:32:31 PM  msrll.exe:1372    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoin
ts2\{091136c0-7a8e-11d8-941f-000c29e6fb6b}\     SUCCESS     Key:
0xE1FCC1B0
1854  7:32:31 PM  msrll.exe:1372    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoin
ts2\Z SUCCESS     Key: 0xE1FCC1B0
```

```
1905  7:32:31 PM  msrll.exe:1372    CreateKey
      HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\User
Shell Folders     SUCCESS     Key: 0xE1F8E660
1908  7:32:31 PM  msrll.exe:1372    CreateKey
      HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell
Folders     SUCCESS     Key: 0xE1F8E660
1916  7:32:31 PM  msrll.exe:1372    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User
Shell Folders     SUCCESS     Key: 0xE1FCC1B0
1919  7:32:31 PM  msrll.exe:1372    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell
Folders     SUCCESS     Key: 0xE1FCC1B0
1927  7:32:31 PM  msrll.exe:1372    CreateKey
      HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\User
Shell Folders     SUCCESS     Key: 0xE1F8E660
1930  7:32:31 PM  msrll.exe:1372    CreateKey
      HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell
Folders     SUCCESS     Key: 0xE1F8E660
2214  7:32:31 PM  msrll.exe:1372    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User
Shell Folders     SUCCESS     Key: 0xE118DD68
2217  7:32:31 PM  msrll.exe:1372    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell
Folders     SUCCESS     Key: 0xE118DD68
2220  7:32:31 PM  msrll.exe:1372    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User
Shell Folders     SUCCESS     Key: 0xE118DD68
2223  7:32:31 PM  msrll.exe:1372    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell
Folders     SUCCESS     Key: 0xE118DD68
2518  7:32:32 PM  msrll.exe:1408    CreateKey
      HKLM\SOFTWARE\Microsoft\Cryptography\RNG  SUCCESS     Key:
0xE1585288
2548  7:32:32 PM  msrll.exe:1408    CreateKey
      HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings
      SUCCESS     Key: 0xE1585288
3664  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User
Shell Folders     SUCCESS     Key: 0xE15864C8
3667  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell
Folders     SUCCESS     Key: 0xE15864C8
3702  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User
Shell Folders     SUCCESS     Key: 0xE15864C8
3705  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell
Folders     SUCCESS     Key: 0xE15864C8
3716  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User
Shell Folders     SUCCESS     Key: 0xE15864C8
3719  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell
Folders     SUCCESS     Key: 0xE15864C8
3926  7:32:48 PM  msrll.exe:1408    CreateKey
      HKLM\Software\Microsoft\Tracing     SUCCESS     Key: 0xE1B4A628
```

```
4297  7:32:48 PM  msrll.exe:1408    CreateKey
      HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\User
Shell Folders     SUCCESS     Key: 0xE1F8DDB0
4402  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCU\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
      SUCCESS     Key: 0xE1916C00
4433  7:32:48 PM  msrll.exe:1408    CreateKey
      HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell
Folders     SUCCESS     Key: 0xE1F8DDB0
4437  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User
Shell Folders     SUCCESS     Key: 0xE1F856A0
4508  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCU\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
      SUCCESS     Key: 0xE1AF4A08
4540  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell
Folders     SUCCESS     Key: 0xE1F8DDB0
4544  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCU\Software\Microsoft\windows\CurrentVersion\Internet Settings
      SUCCESS     Key: 0xE1F8DDB0
4553  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCU\Software\Microsoft\windows\CurrentVersion\Internet
Settings\Connections    SUCCESS     Key: 0xE1F8DDB0
4557  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCU\Software\Microsoft\windows\CurrentVersion\Internet
Settings\Connections    SUCCESS     Key: 0xE1F8DDB0
4561  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCU\Software\Microsoft\windows\CurrentVersion\Internet Settings
      SUCCESS     Key: 0xE1F8DDB0
4568  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCC\Software\Microsoft\windows\CurrentVersion\Internet Settings
      SUCCESS     Key: 0xE1F856A0
4571  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCU\Software\Microsoft\windows\CurrentVersion\Internet
Settings\Connections    SUCCESS     Key: 0xE1F856A0
4574  7:32:48 PM  msrll.exe:1408    CreateKey
      HKCU\Software\Microsoft\windows\CurrentVersion\Internet
Settings\Connections    SUCCESS     Key: 0xE1AF4A08
4578  7:32:48 PM  msrll.exe:1408    CreateKey
      HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
      SUCCESS     Key: 0xE1AF4A08
4658  7:32:48 PM  msrll.exe:1408    CreateKey
      HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
      SUCCESS     Key: 0xE1F856A0
4678  7:32:48 PM  msrll.exe:1408    CreateKey
      HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
      SUCCESS     Key: 0xE1AF4A08
4767  7:33:58 PM  msrll.exe:1408    CreateKey
      HKLM\SOFTWARE\Microsoft\Cryptography\RNG  SUCCESS     Key:
0xE1916C00
4770  7:33:58 PM  msrll.exe:1408    CreateKey
      HKLM\SOFTWARE\Microsoft\Cryptography\RNG  SUCCESS     Key:
0xE1916C00
4773  7:33:58 PM  msrll.exe:1408    CreateKey
      HKLM\SOFTWARE\Microsoft\Cryptography\RNG  SUCCESS     Key:
0xE1916C00
```

```
4776  7:33:58 PM  msrll.exe:1408    CreateKey
      HKLM\SOFTWARE\Microsoft\Cryptography\RNG  SUCCESS     Key:
0xE1916C00
4779  7:33:58 PM  msrll.exe:1408    CreateKey
      HKLM\SOFTWARE\Microsoft\Cryptography\RNG  SUCCESS     Key:
0xE1916C00
4782  7:33:58 PM  msrll.exe:1408    CreateKey
      HKLM\SOFTWARE\Microsoft\Cryptography\RNG  SUCCESS     Key:
0xE1916C00
4785  7:33:58 PM  msrll.exe:1408    CreateKey
      HKLM\SOFTWARE\Microsoft\Cryptography\RNG  SUCCESS     Key:
0xE1916C00
```

List of References

Baker, Andrew R., Caswell, Brian, Poor, Mike. Snort 2.1 Intrusion Detedtion, Second Edition. Rockland: Syngress Publishing, 2004.

Duntemann, Jeff. Assembly Language Step-by-Step: Programming with DOS and Linux, Second Edition. New York: John Wiley & Sons, 2000.

Hoglund, Greg, and McGraw, Gary. Exploiting Software: How to Break Code. Boston: Addison-Wesley, 2004.

Negus, Christopher. Red Hat Linux 8 Bible. Indianapolis: Wiley, 2002.

Skoudis, Ed. Malware: Fighting Malicious Code. Upper Saddle River: Prentice Hall PTR, 2004.

Yoda. Readme.tXt. 2002.

Zeltser, Lenny. Reverse-Engineering Malware, Section 3: Deeper Analysis. 2004.

Last Updated: February 25th, 2018

Upcoming SANS Forensics Training

SANS London March 2018	London, United Kingdom	Mar 05, 2018 - Mar 10, 2018	Live Event
SANS San Francisco Spring 2018	San Francisco, CA	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS Secure Singapore 2018	Singapore, Singapore	Mar 12, 2018 - Mar 24, 2018	Live Event
SANS Paris March 2018	Paris, France	Mar 12, 2018 - Mar 17, 2018	Live Event
Mentor Session - FOR500	Minneapolis, MN	Mar 13, 2018 - May 01, 2018	Mentor
SANS Northern VA Spring - Tysons 2018	McLean, VA	Mar 17, 2018 - Mar 24, 2018	Live Event
SANS Munich March 2018	Munich, Germany	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS Pen Test Austin 2018	Austin, TX	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS Secure Canberra 2018	Canberra, Australia	Mar 19, 2018 - Mar 24, 2018	Live Event
Mentor Session - FOR610	Milwaukee, WI	Mar 21, 2018 - May 02, 2018	Mentor
SANS Boston Spring 2018	Boston, MA	Mar 25, 2018 - Mar 30, 2018	Live Event
Community SANS Columbia FOR610	Columbia, MD	Mar 26, 2018 - Mar 31, 2018	Community SANS
SANS 2018	Orlando, FL	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS Abu Dhabi 2018	Abu Dhabi, United Arab Emirates	Apr 07, 2018 - Apr 12, 2018	Live Event
Community SANS Virginia Beach FOR508 @ SLAIT	Virginia Beach, VA	Apr 09, 2018 - Apr 14, 2018	Community SANS
SANS vLive - FOR578: Cyber Threat Intelligence	FOR578 - 201804,	Apr 10, 2018 - May 17, 2018	vLive
SANS London April 2018	London, United Kingdom	Apr 16, 2018 - Apr 21, 2018	Live Event
SANS Zurich 2018	Zurich, Switzerland	Apr 16, 2018 - Apr 21, 2018	Live Event
SANS Baltimore Spring 2018	Baltimore, MD	Apr 21, 2018 - Apr 28, 2018	Live Event
SANS Seattle Spring 2018	Seattle, WA	Apr 23, 2018 - Apr 28, 2018	Live Event
SANS vLive - FOR508: Advanced Digital Forensics, Incident Response, and Threat Hunting	FOR508 - 201804,	Apr 23, 2018 - May 30, 2018	vLive
SANS Riyadh April 2018	Riyadh, Saudi Arabia	Apr 28, 2018 - May 03, 2018	Live Event
Automotive Cybersecurity Summit & Training 2018	Chicago, IL	May 01, 2018 - May 08, 2018	Live Event
SANS vLive - FOR500: Windows Forensic Analysis	FOR500 - 201805,	May 08, 2018 - Jun 14, 2018	vLive
Security West 2018 - FOR610: Reverse-Engineering Malware: Malware Analysis Tools and Techniques	San Diego, CA	May 11, 2018 - May 16, 2018	vLive
Security West 2018 - FOR578: Cyber Threat Intelligence	San Diego, CA	May 11, 2018 - May 15, 2018	vLive
Security West 2018 - FOR500: Windows Forensic Analysis	San Diego, CA	May 11, 2018 - May 16, 2018	vLive
SANS Security West 2018	San Diego, CA	May 11, 2018 - May 18, 2018	Live Event
Security West 2018 - FOR508: Advanced Digital Forensics, Incident Response, and Threat Hunting	San Diego, CA	May 11, 2018 - May 16, 2018	vLive
Security West 2018 - FOR572: Advanced Network Forensics and Analysis	San Diego, CA	May 11, 2018 - May 16, 2018	vLive
Security West 2018 - FOR518: Mac Forensic Analysis	San Diego, CA	May 11, 2018 - May 16, 2018	vLive