



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

## Interested in learning more?

Check out the list of upcoming events offering  
"Reverse-Engineering Malware: Malware Analysis Tools and Techniques (Forensics)"  
at <http://www.giac.org/registration/grem>



**Michael Mauch**

**GIAC Reverse Engineering Malware (GREM)**

**Practical Assignment, Version 1.0**

**GREM Certification Attempt**

**October, 1<sup>st</sup> 2004**

© SANS Institute. Author retains full rights.

## Table of Contents

<b>1</b>	<b><u>ABSTRACT.....</u></b>	<b><u>3</u></b>
<b>2</b>	<b><u>LABORATORY SETUP.....</u></b>	<b><u>4</u></b>
<b>3</b>	<b><u>PROPERTIES OF THE MALWARE SPECIMEN.....</u></b>	<b><u>7</u></b>
<b>4</b>	<b><u>BEHAVIORAL ANALYSIS.....</u></b>	<b><u>10</u></b>
4.1	MONITORING REGISTRY AND FILE SYSTEM ACCESS.....	10
4.2	MONITORING / REDIRECTING NETWORK CONNECTIONS.....	13
4.3	MONITORING PROCESSES ON THE SYSTEM.....	16
<b>5</b>	<b><u>CODE ANALYSIS.....</u></b>	<b><u>17</u></b>
5.1	LORDPE / ASPACKDIE.....	17
5.2	IDA PRO.....	19
5.3	OLLYDBG.....	20
<b>6</b>	<b><u>ANALYSIS WRAP-UP.....</u></b>	<b><u>21</u></b>
<b>7</b>	<b><u>APPENDIX.....</u></b>	<b><u>23</u></b>
7.1	TABLE OF FIGURES.....	23
7.2	REFERENCES.....	23
7.3	BINTEXT OUTPUT OF THE WITH ASPACKDIE UNCOMPRESSED EXECUTABLE.....	24
7.4	SNORT CAPTURE OF IRC LOGIN AND IDENT REQUEST.....	36

## 1 Abstract

Intelligent malware spreads more and more, one can read this in the news nearly every day. But what to do if you think you have an infected system and your AV scanner doesn't find anything?

With the documentation of the practical part of the REM certification attempt I will show a flexible way how it is possible to analyse such malware in case of an incident.

Within this document the method and the systems used for analysis of the given malware is described.

Because of having no deep programming background I started with behavioral analysis and went then over to the code analysis. The other way round would have been possible, too.

© SANS Institute 2004, Author retains all rights.

## 2 Laboratory Setup

There are a few possibilities for setting up a laboratory for analysing malicious code (using virtual or physical machines, and so on). If interested you can find more information in the book "Malware: Fighting Malicious Code" by Ed Skoudis and Lenny Zeltser (see also [references](#) at the appendix).

I used vmware for analysing the malware. This has the following advantages:

- **Isolated environment:** As network configuration I used the host-only mode for all my virtual machines to keep the test environment isolated from our productive network. On my host machine (a laptop) I had a personal firewall running which should block all packets originating from the laboratory environment.
- **Flexibility:** After preparing the virtual machines for analysing the malware I took snapshots of the machines. After infecting „vm1“ I could revert the machine to its state before the infection. So I could execute the malware as often as I needed.
- **Portability:** Having installed vmware on my laptop I was able to analyse the malware at every place I wanted. It is also good for presenting the work to other people.

→ This is the main reason why I decided not to setup a laboratory with real, physical machines. I mainly worked on this analysis at two places, at my workplace and at home.

The virtual machines got their IP address configuration from the vmware DHCP service (see [figure 1](#)).

I installed 3 virtual machines:

vm1: Windows 2000 Professional SP4: This is the machine I infected with the malware. I installed the following analysis tools on this machine:

- BinText from Foundstone Inc.

This tool is used for extracting strings from binary files. I extracted strings with it from the malware executable msrll.exe.

- Registry Monitor from Sysinternals

This tool is used for monitoring the registry on windows machines. It shows all kind of accesses (read, add, modify, delete) of keys and values. I used it to monitor the registry while executing msrll.exe.

- TDImon from Sysinternals

TDImon is used for monitoring network activity (UDP and TCP) of the system it is installed on. It also shows the processes associated with the network connections. This tool I used for monitoring network activity after executing msrll.exe.

- File Monitor from Sysinternals

Filemon is used for monitoring the file system. It also shows the processes associated with the file system activities. This tool I used for monitoring file system activity after executing msrll.exe.

- Regshot from [regshot.yeah.net](http://regshot.yeah.net)

Regshot is used for comparing two snapshots of the windows registry. For example comparing the registry before and after executing a given program, which I did with msrll.exe.

- IDA Pro from Datarescue

IDA Pro is a disassembler which delivers as output all assembly instructions from a executable file. I used it for disassembling the uncompressed version of msrll.exe and the with LordPE dumped file.

- LordPE from [y0da.cjb.net](http://y0da.cjb.net)

With LordPE (PE = portable executable) you can edit and analyze executable files, for example dump them from memory, which I did with LordPE.

- OllyDbg from [home.t-online.de/home/Ollydbg](http://home.t-online.de/home/Ollydbg)

OllyDbg is a very feature rich and free debugger which is used to analyse a running program. OllyDbg can be attached to a running process in the RAM of a machine. I used it for analysing msrll.exe.

- md5sum

md5sum is a program to generate hash values from given files. It can be used to identify changes in files. I compared msrll.exe with the new file it places at c:\WINNT\system32\mfmm.

- Aspackdie

Aspackdie is a tool to revert the compression of executable files, which have been compressed with aspack. It is not an official aspack tool, aspack compressions are not intend to be reverted. I reverted the compression of msrll.exe with aspackdie.

vm2: Windows 2000 Professional SP4: This was one of my existing virtual machine's with the following tools:

- Etheral from [www.ethereal.com](http://www.ethereal.com)

Etheral is a free and powerful network protocol analyser.

- Windump from [windump.polito.it](http://windump.polito.it)

Windump is the Windows version of tcpdump, a network sniffer.

I used both tools for the first analysis steps (sniffing on the network).

- vm3: Red Hat Linux (kernel 2.4.18-10): This is the virtual machine I got in the REM course at SANS 2004 in Orlando. I used the following tools:
- IRC server and client
 

I used the IRC server and client for providing an IRC service and for communicating with the malware.
  - Snort from [www.snort.org](http://www.snort.org)

Snort is mainly used as intrusion detection system, but has also some options for sniffing on the network. I used it for analyzing network activity of the malware.
  - Netcat
 

With netcat you can easily open ports on a system. This may be helpful if you want to analyse a program that tries to connect to a special port and you don't have a application listening on this port (or if you don't know the application ...).
  - telnet
 

Terminal emulation program. I used it for communicating with the malware.
  - file command for analysing the malware behavior.
 

I used the file command to analyse the type (executable?) of the malware file.

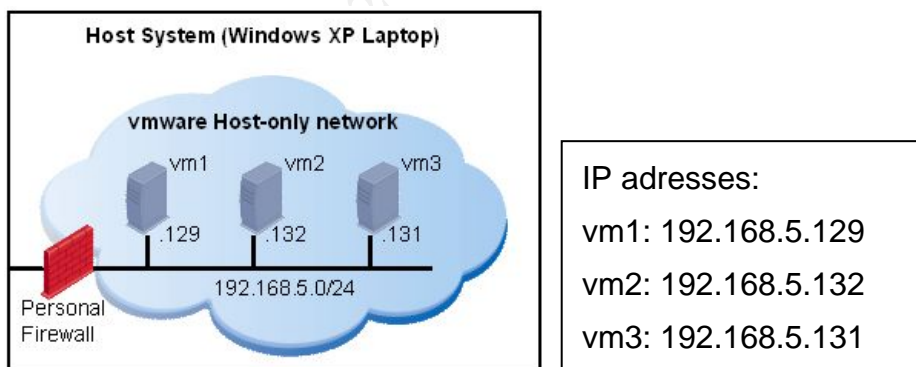


Figure 1: Laboratory Setup

### 3 Properties of the Malware Specimen

Within this chapter the properties of the malware are listed:

- **Type of file**

This is an „aspack“ compressed, executable file. The compression tool could be found with help of the embedded string “.aspack”, as shown in figure 2 (marked red). This means, that the executable extracts itself at runtime. Unfortunately aspack compression is not designed to be reverted. Perhaps that's a reason, why the malware author used this one. So it is a little bit harder to analyse the file. But there are several tools available, that can uncompress such files, like the tool aspackdie I used.

That the file is an windows executable could also be verified with the Linux “file” command on “vm3”:

```
[root@localhost software]# file msrll.exe
msrll.exe: MS Windows PE Intel 80386 GUI executable not relocatable
[root@localhost software]#
```

Figure 2: Linux file command

- **Size of the file**

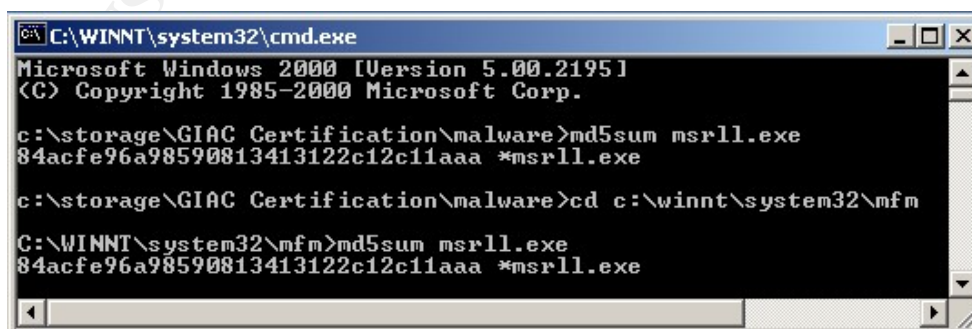
Size: 41,0 KB (41,984 bytes)

Size on disk: 44,0 KB (45,056 bytes)

For gathering this information I used the properties dialog of this file on “vm1” (right click on the file → properties)

- **MD5 hash of the file**

I used the tool md5sum on “vm1” for generating the fingerprints. I created fingerprints from the original file before execution and from the new generated file in c:\winnt\system32\mfm.



```
C:\WINNT\system32\cmd.exe
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

c:\storage\GIAC Certification\malware>md5sum msrll.exe
84acfe96a98590813413122c12c11aaa *msrll.exe

c:\storage\GIAC Certification\malware>cd c:\winnt\system32\mfm

C:\WINNT\system32\mfm>md5sum msrll.exe
84acfe96a98590813413122c12c11aaa *msrll.exe
```

Figure 3: MD5 of msrll.exe

The file in c:\winnt\system32\mfm is an exact copy of the original file, the MD5 hash values are identical. The original file was deleted from the folder it was executed.



- **Operating system(s) it runs on**

Windows 2000 Professional SP4 (→ “vm1”). I did not test any other operating system.

- **Strings embedded into the file msrll.exe**

These are the strings in the compressed executable, extracted with BinText on the machine „vm1“. I also uncompressed the file with „aspackdie“ (from <http://protools.anticrack.de/files/unpackers/aspackdie.zip>). The strings of this uncompressed file are too much to put them in this chapter of the document. You can find them in the appendix at the end of the document. I just listed the strings which looked like commands.

**Strings from the compressed msrll.exe:**

File pos	Mem pos	ID	Text
=====	=====	==	=====
0000004D	0040004D	0	!This program cannot be run in DOS mode.
00000178	00400178	0	.text
000001A0	004001A0	0	.data
000001F0	004001F0	0	.idata
00000218	00400218	0	.aspack
00000240	00400240	0	.adata
[snip]			
00009271	0051D071	0	VirtualAlloc
0000927E	0051D07E	0	VirtualFree
00009641	0051D441	0	kernel32.dll
0000964E	0051D44E	0	ExitProcess
0000965A	0051D45A	0	user32.dll
00009665	0051D465	0	MessageBoxA
00009671	0051D471	0	wsprintfA
0000967B	0051D47B	0	LOADER ERROR
00009688	0051D488	0	The procedure entry point %s could not be located in the dynamic link library %s
000096D9	0051D4D9	0	The ordinal %u could not be located in the dynamic link library %s

[snip]			
0000A16C	0051DF6C	0	kernel32.dll
0000A17B	0051DF7B	0	GetProcAddress
0000A18C	0051DF8C	0	GetModuleHandleA
0000A19F	0051DF9F	0	LoadLibraryA
0000A274	0051E074	0	advapi32.dll
0000A281	0051E081	0	msvcrt.dll
0000A28C	0051E08C	0	msvcrt.dll
0000A297	0051E097	0	shell32.dll
0000A2A3	0051E0A3	0	user32.dll
0000A2AE	0051E0AE	0	version.dll
0000A2BA	0051E0BA	0	wininet.dll
0000A2C6	0051E0C6	0	ws2_32.dll
0000A313	0051E113	0	AdjustTokenPrivileges
0000A32B	0051E12B	0	_itoa
0000A333	0051E133	0	__getmainargs
0000A343	0051E143	0	ShellExecuteA
0000A353	0051E153	0	DispatchMessageA
0000A366	0051E166	0	GetFileVersionInfoA
0000A37C	0051E17C	0	InternetCloseHandle
0000A392	0051E192	0	WSAGetLastError

Figure 4: embedded strings in the compressed msrll.exe

**Strings from the uncompressed msrll.exe, which look like commands:**

File pos	Mem pos	ID	Text
=====	=====	==	=====
00001326	00401326	0	?insmod
0000132E	0040132E	0	?rmmod
00001335	00401335	0	?lsmod
00002753	00402753	0	?ping
00002763	00402763	0	?smurf
0000276A	0040276A	0	?jolt
0000934E	0040934E	0	?clone
00009355	00409355	0	?clones
0000935D	0040935D	0	?login
00009364	00409364	0	?uptime
0000936C	0040936C	0	?reboot
00009374	00409374	0	?status
0000937C	0040937C	0	?jump
00009382	00409382	0	?nick
00009388	00409388	0	?echo
0000938E	0040938E	0	?hush
00009394	00409394	0	?wget
0000939A	0040939A	0	?join
000093A9	004093A9	0	?akick
000093B0	004093B0	0	?part
000093B6	004093B6	0	?dump
000093C6	004093C6	0	?md5p
000093CC	004093CC	0	?free
000093D7	004093D7	0	?update
000093DF	004093DF	0	?hostname
000093EE	004093EE	0	?!fif
000093FE	004093FE	0	?play
00009404	00409404	0	?copy
0000940A	0040940A	0	?move
00009415	00409415	0	?sums
00009423	00409423	0	?rmdir
0000942A	0040942A	0	?mkdir
00009436	00409436	0	?exec
00009440	00409440	0	?kill
00009446	00409446	0	?killall
0000944F	0040944F	0	?crash
0000946E	0040946E	0	?sklist
00009476	00409476	0	?unset
0000947D	0040947D	0	?uattr
00009484	00409484	0	?dccsk
00009490	00409490	0	?killsk

**Figure 5: embedded strings in the uncompressed msrll.exe, which look like commands**

© SANS Institute 2004, Author

## 4 Behavioral Analysis

After looking at the properties of the malware executable file, I started with the behavioral analysis.

Before executing the malware I prepared several tools to monitor the system and the network:

On „vm1“ I started regshot, TDImon, file monitor, registry monitor and took the first shot with regshot.

On „vm2“ I started ethereal to monitor the network.

Then I executed msrll.exe on „vm1“. After execution I waited a few seconds and took the second shot with regshot.

### 4.1 Monitoring registry and file system access

For monitoring registry and file system access the tools regshot, file monitor and registry monitor have been used. Inter alia the compare function of regshot provided the following information:

- **Keys added:**

The malware added the following keys:

*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*

- **Values added:**

The malware added the following values:

*HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\mfm\Security\Security: 01 00 14 80 A0 00 00 00 AC 00 00 00 14 00 00 00 30 00 00 00 02 00 1C 00 01 00 00 00 02 80 14 00 FF 01 0F 00 01 01 00 00 00 00 01 00 00 00 02 00 70 00 04 00 00 00 00 18 00 FD 01 02 00 01 01 00 00 00 00 05 12 00 00 00 76 00 63 00 00 00 1C 00 FF 01 0F 00 01 02 00 00 00 00 05 20 00 00 00 20 02 00 00 00 00 00 00 00 00 00 00 18 00 8D 01 02 00 01 01 00 00 00 00 05 0B 00 00 00 20 02 00 00 00 1C 00 FD 01 02 00 01 02 00 00 00 00 00 05 20 00 00 00 23 02 00 00 00 00 00 01 01 00 00 00 00 05 12 00 00 00 01 01 00 00 00 00 05 12 00 00 00*

*HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\mfm\Type: 0x00000120*

*HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\mfm\Start: 0x00000002*

*HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\mfm\ErrorControl: 0x00000002*

*HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\mfm\ImagePath: "C:\WINNT\system32\mfm\msrll.exe"*

*HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\mfm\DisplayName: "Rll enhanced drive"*

*HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\mfm\ObjectName: "LocalSystem"*

The same keys have also been added to:

*HKEY\_LOCAL\_MACHINE\SYSTEM\ControlSet001\..*

The malware added registry keys to provide automatic start at boot-time of the infected system. It registered the service „Rll enhanced drive“, also shown in the following screenshot:

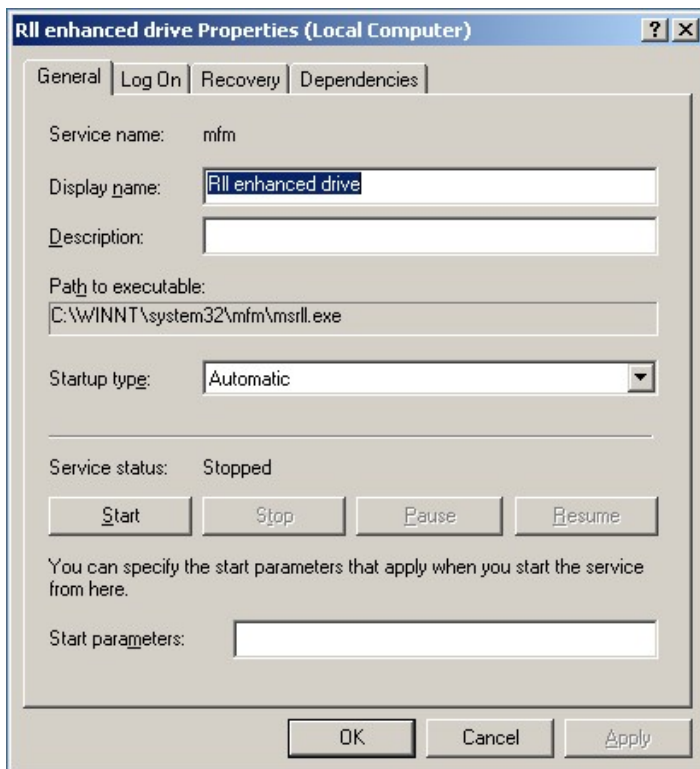


Figure 6: Service „RII enhanced drive“

- **Files added:**

*C:\WINNT\system32\mfm\jtram.conf*

*C:\WINNT\system32\mfm\msrll.exe*

This shows, that the malware added two files, msrll.exe and jtram.conf.

- **Files deleted:**

*C:\storage\GIAC Certification\malware\msrll.exe*

This is the folder where the malware was executed from – it removes its source executable file.

- **Folders added:**

*C:\WINNT\system32\mfm*

*C:\WINNT\system32\mfm\.*

*C:\WINNT\system32\mfm\.*

The malware also added the folder c:\winnt\system32\mfm\. This is the folder where it places the two files msrll.exe and jtram.conf

- **Files (or attributes) modified:**

The malware modified (or changed attributes) of the following files:

*C:\Documents and Settings\Administrator\Cookies\index.dat*

*C:\Documents and Settings\Administrator\Local Settings\History\History.IE5\index.dat*

*C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\index.dat*

*C:\Documents and Settings\Administrator\NTUSER.DAT*

*C:\Documents and Settings\Administrator\ntuser.dat.LOG*

*C:\WINNT\system32\config\SECURITY*

*C:\WINNT\system32\config\SECURITY.LOG*

*C:\WINNT\system32\config\software*

*C:\WINNT\system32\config\software.LOG*

*C:\WINNT\system32\config\system*

*C:\WINNT\system32\config\SYSTEM.ALT*

*C:\WINNT\wincmd.ini*

- **Values modified:**

The malware modified the following registry values:

*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Cryptography\ RNG\Seed: 74 DE 8E 70 5D 3D CE B9 FB F5 F2 37 30 73 84 C5 5C 3F 12 67 00 A6 03 C8 B7 39 0B 8C 53 E1 A0 7F 5E 25 87 59 2F D0 79 FE 53 EC 83 F3 6D DE 61 B8 B2 C4 4D 41 C0 5B D0 01 F8 4D 49 9C EE A4 D5 18 25 78 3E C5 2B 16 1C BA 17 2A FB 6B 5A 8A C7 1C*

*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Cryptography\ RNG\Seed: 48 FE 29 55 99 68 07 4A B8 08 A2 29 0C 38 DE B7 21 14 EF FF 57 3F D3 31 B1 11 A7 9F 52 0C 6E 39 98 E5 A5 03 39 E9 5C DF BF 40 89 37 7C 78 E0 41 EF 25 07 8F 32 E4 C2 26 DA D3 AC 5F 25 97 AA 5F 10 57 49 CC A8 ED B7 AC 4C 22 81 02 14 89 F8 FC*

*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Paths\Directory: "C:\Documents and Settings\Default User\Local Settings\Temporary Internet Files\Content.IE5"*

*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Paths\Directory: "C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5"*

*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Paths\path1\CachePath: "C:\Documents and Settings\Default User\Local Settings\Temporary Internet Files\Content.IE5\Cache1"*

*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Paths\path1\CachePath: "C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\Cache1"*

*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Paths\path2\CachePath: "C:\Documents and Settings\Default User\Local Settings\Temporary Internet Files\Content.IE5\Cache2"*

*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Paths\path2\CachePath: "C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\Cache2"*

*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Paths\path3\CachePath: "C:\Documents and Settings\Default User\Local Settings\Temporary Internet Files\Content.IE5\Cache3"*

*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Paths\path3\CachePath: "C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\Cache3"*

*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Paths\path4\CachePath: "C:\Documents and Settings\Default User\Local Settings\Temporary Internet Files\Content.IE5\Cache4"*

*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Paths\path4\CachePath: "C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\Cache4"*

## 4.2 Monitoring / redirecting network connections

For monitoring network connections / network access for the first time I used ethereal on "vm2". Of course I could have used the linux system "vm3", but at this state of the analysis I did not know enough about the malware. This was the only analysis I did with "vm2", after that I only used "vm1" and "vm3".

- **Send DNS request for collective7.zxy0.com**

After seeing this I edited the hosts file of „vm1“ and used the IP address of „vm3“ for collective7.zxy0.com.

- **Sends tcp syn to collective7.zxy0.com port 9999**

It is possible, that the malware is searching for an installed trojan. „The Prayer“ uses this port for example.

- **Sends tcp syn to collective7.zxy0.com port 8080**

Maybe the malware searches for a web proxy on collective7.zxy0.com

I tried to gather some more information about the connection attempts to the tcp ports 9999 and 8080 with netcat on „vm3“, but unfortunately there was nothing to see with „nc -p <port> -l -n“.

The only thing I could see for the requests to the ports 9999 and 8080 with ethereal was a tcp syn packet, the first packet of a tcp 3 way handshake, but nothing more → a connection could not be established.

- **Sends tcp syn to collective7.zxy0.com port 6667**

Here the malware tries to establish a connection to an IRC server.

The above mentioned connections can be seen in the screenshot of the ethereal capture from „vm2“:

7	31.089399	192.168.5.129	192.168.5.132	TCP	1053 > 8080 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
8	31.089492	192.168.5.132	192.168.5.129	TCP	[TCP ZeroWindow] 8080 > 1053 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
9	31.605961	192.168.5.129	192.168.5.132	TCP	1053 > 8080 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
10	31.606053	192.168.5.132	192.168.5.129	TCP	[TCP ZeroWindow] 8080 > 1053 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
11	32.154384	192.168.5.129	192.168.5.132	TCP	1053 > 8080 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
12	32.154485	192.168.5.132	192.168.5.129	TCP	[TCP ZeroWindow] 8080 > 1053 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
13	93.151959	192.168.5.129	192.168.5.132	TCP	1054 > 6667 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
14	93.152049	192.168.5.132	192.168.5.129	TCP	[TCP ZeroWindow] 6667 > 1054 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
15	93.652979	192.168.5.129	192.168.5.132	TCP	1054 > 6667 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
16	93.653081	192.168.5.132	192.168.5.129	TCP	[TCP ZeroWindow] 6667 > 1054 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
17	94.196912	192.168.5.129	192.168.5.132	TCP	1054 > 6667 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
18	94.196980	192.168.5.132	192.168.5.129	TCP	[TCP ZeroWindow] 6667 > 1054 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
19	124.166132	192.168.5.129	192.168.5.132	TCP	1055 > 9999 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
20	124.166209	192.168.5.132	192.168.5.129	TCP	[TCP ZeroWindow] 9999 > 1055 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
21	124.692671	192.168.5.129	192.168.5.132	TCP	1055 > 9999 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
22	124.692751	192.168.5.132	192.168.5.129	TCP	[TCP ZeroWindow] 9999 > 1055 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
23	125.213276	192.168.5.129	192.168.5.132	TCP	1055 > 9999 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460

Figure 7: Ethereal capture

- **Tries to connect to channel #mils on IRC server collective7.zxy0.com**

After the first analysis results I reverted my virtual machine „vm1“ to the state bevor executing the malware and prepared the linux system „vm3“. I started snort with the parameter „-vd“ (to use it as network sniffer) and started the IRC server. Then I executed the malware again. Now I could see that the malware tries to join the IRC channel „#mils“, as you can see in figure 5:

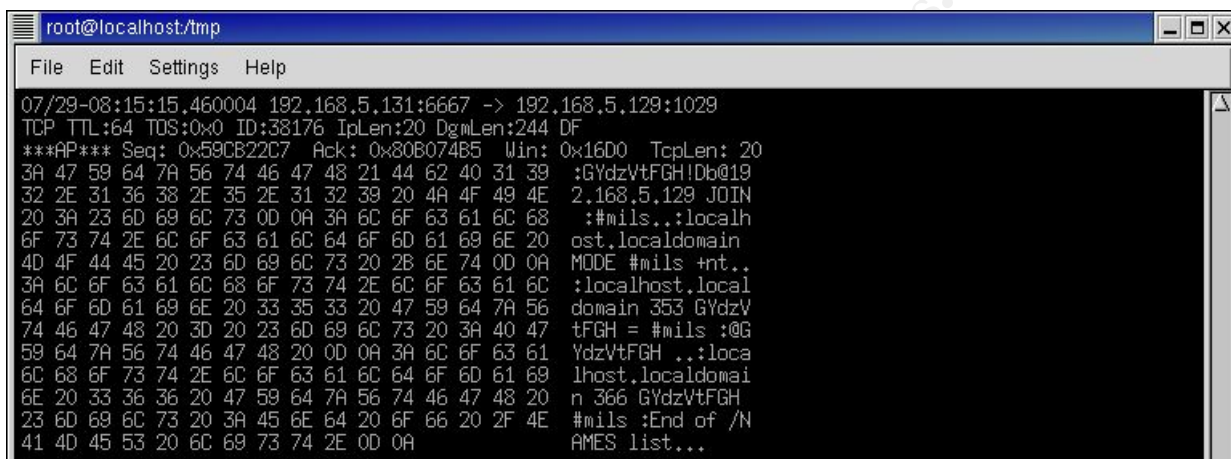


Figure 8: IRC join #mils

After this I started an IRC client on „vm3“ and joined #mils. With the command /who #mils I could see that the malware was online:

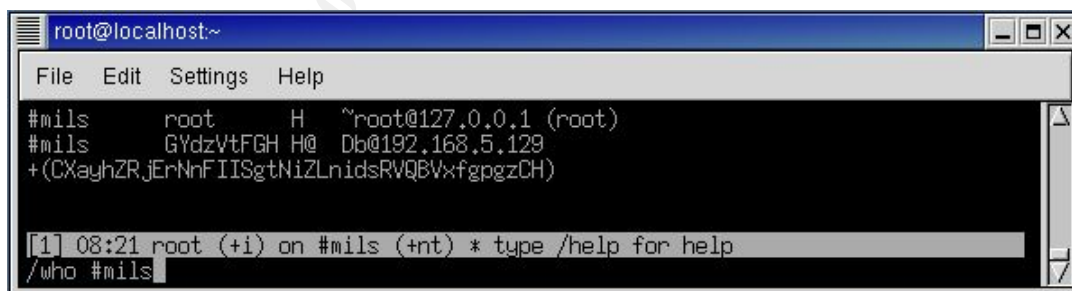


Figure 9: IRC /who #mils

Every time I reverted “vm1” and executed msrll.exe again, the malware joined IRC channel #mils with another nick, this shows that the nick is generated by random.

I tried to communicate with the malware using the most interesting strings I found with BinText in the uncompressed msrll.exe (the strings which looked like commands like ?login, ?uptime, ...), but unfortunately I had no success. You can find some more information about analysing IRC usage in chapter 5.



- **Listens on tcp ports 2200 and 113**

The malware also listens on tcp ports 2200 and 113 (ident) on the infected machine “vm1”. This could be seen with TDlmon on “vm1”, which also shows the corresponding processes to opened ports:

105	15.51...	msrll.exe:828	81521EC8	IRP_MJ_CREATE	TCP:0.0.0.0:2200	SUCCESS	Address Open
173	16.72...	msrll.exe:828	815CBA68	IRP_MJ_CREATE	TCP:0.0.0.0:113	SUCCESS	Address Open

**Figure 10: TDlmon: opened tcp ports 2200 and 113**

I also tried to communicate with the malware using this ports. Therefore I did a telnet on the ports 113 and 2200 from „vm3“ to „vm1“. It seems that tcp port 113 acts like the identification protocol (for more information see RFC 1413)

On port 2200 I got a prompt, but was not able to communicate with the malware.



**Figure 11: telnet on tcp ports 2200 and 113**

**Tcp port 113: (ident)**

IRC servers usually generate a ident request on tcp port 113 if some one tries to log on. The IRC servers try to verify the identity of the person who is logging in. Due to abuse by open proxies some ISPs (for example T-Online in Germany) require a correct ident reply. This may be the reason, that the malware provides the ident service.

This behaviour could be seen with snort on “vm3”. When the malware tries to connect and join an IRC channel, the IRC server makes an ident request to tcp port 113 on “vm1”. In [appendix 7.4](#) you can see the snort output of this communication.



### 4.3 Monitoring processes on the system

After execution of msrll.exe there was a new process (msrll.exe) on the infected machine. In the figure below you can see the taskmanager bevor execution (left picture) and after (right picture).

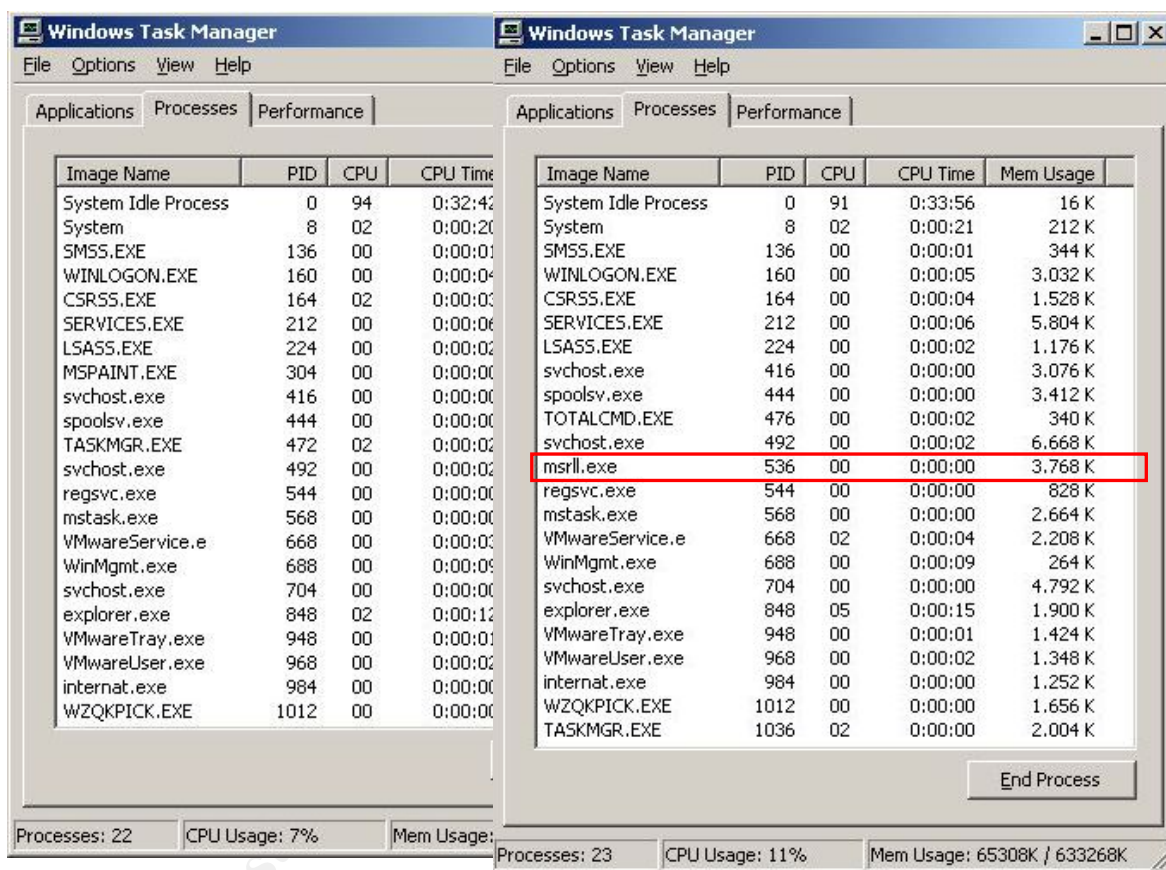


Figure 12: processes bevor and after execution of msrll.exe

## 5 Code Analysis

For the code analysis the following steps have been used:

1. Unpacking
2. Disassembling
3. Debugging

These steps are described in the next chapters.

### 5.1 LordPE / aspackdie

For disassembling the malware I tried to load the msrll.exe in IDA Pro. However the aspack compressions conceals contents of the malware, so an uncompressed / unpacked file was needed.

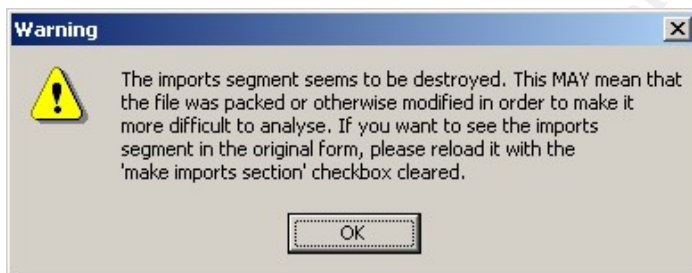


Figure 13: IDA Pro: warning while loading an encrypted file

Therefore I tried two different ways with the same results (I compared both files partially in IDA Pro):

#### 1. I uncompressed the msrll.exe with aspackdie

In the figure below you can see the result of using aspackdie, the file seems to be unpacked successfully.

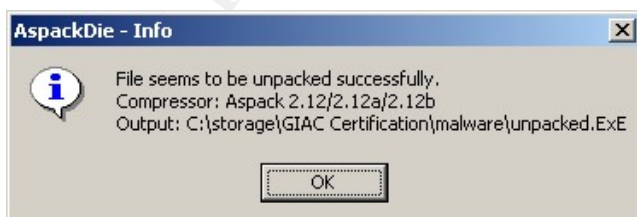


Figure 14: aspackdie

## 2. I dumped the contents with LordPE from memory

After starting LordPE you can see all running processes. To dump the process from memory you have to do a right click on the process msrll.exe and chose “dump full”. Then a “save as” window pops up, where you can save the dumped executable file.

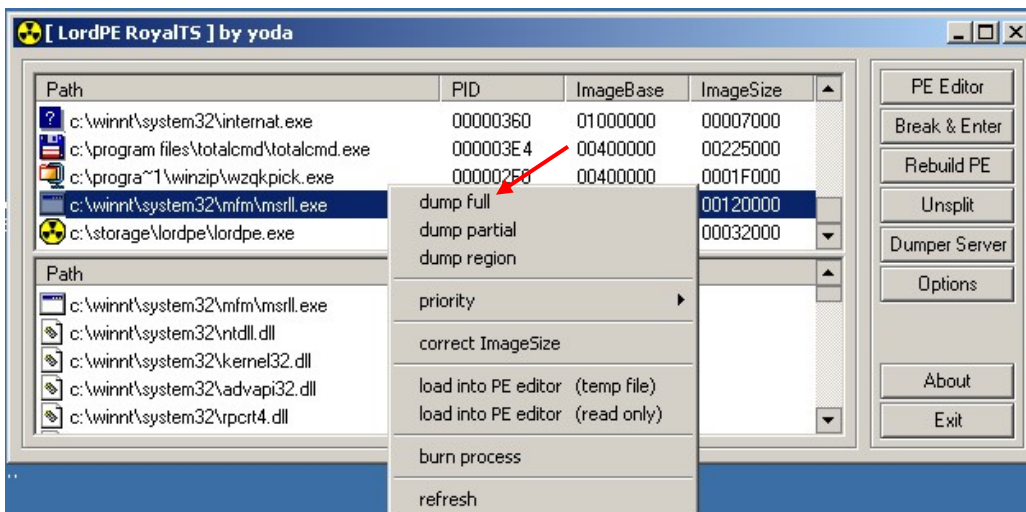


Figure 15: LordPE: “dump full”

Then I used the uncompressed output files for further analysis.

© SANS Institute 2004. Auth

## 5.2 IDA Pro

Programmers convert with compilers the source code of there programs into machine – readable code. With the help of a disassembler it is possible to extract all assembly instructions of such a file. I used IDA Pro on “vm1” for this action.

After loading the uncompressed file into IDA Pro I searched for interesting code segments. IDA Pro generates a list of the embedded strings in the “strings window”. With the help of this list I found a segment where I assume a password comparison. After several „cmp“ and „test“ commands the string „PASS“ is pushed to the stack. Maybe the malware has implemented an authentication system. This could be the reason why it was not possible to communicate with the malware with IRC or telnet on tcp port 2200.

```

IDA View-A | Hex View | Exports | Imports | Names | Functions | Strings | Structures | Enums
-----
.text:00405B2D ;
.text:00405B2D      push    ebp
.text:00405B2E      mov     ebp, esp
.text:00405B30      push    esi
.text:00405B31      push    ebx
.text:00405B32      mov     edx, [ebp+8]
.text:00405B35      mov     esi, [ebp+0Ch]
.text:00405B38      mov     ebx, [ebp+14h]
.text:00405B3B      mov     eax, [esi+205Ch]
.text:00405B41      test   eax, 2
.text:00405B46      jnz    short loc_405B9B
.text:00405B48      cmp    dword ptr [edx+4], 0
.text:00405B4C      jz     short loc_405B9B
.text:00405B4E      test   eax, 10h
.text:00405B53      jz     short loc_405B9B
.text:00405B55      sub    esp, 8
.text:00405B58      push  offset aPass ; "PASS"
.text:00405B5D      push  dword ptr [edx+4]
.text:00405B60      call  sub_405B72
.text:00405B65      add   esp, 10h
.text:00405B68      test  eax, eax
.text:00405B6A      jz    short loc_405B9B
.text:00405B6C      mov   eax, [ebx+0FCh]
.text:00405B72      test  eax, 10000h
.text:00405B77      jnz   short loc_405B9B
.text:00405B79      or    eax, 10000h
.text:00405B7E      mov  [ebx+0FCh], eax
.text:00405B84      sub  esp, 0Ch
.text:00405B87      push ebx
.text:00405B88      push offset aSLoggedIn ; "%s logged in"
.text:00405B8D      push esi
.text:00405B8E      push dword ptr [ebp+10h]
.text:00405B91      push  2
    
```

Figure 16: IDA Pro: Interesting code segment

### 5.3 OllyDbg

For further analysis OllyDbg was used. The goal was to find the password needed to authenticate at the malware.

After starting OllyDbg it was attached to the process msrll.exe. It came up with an „entry point alert“:

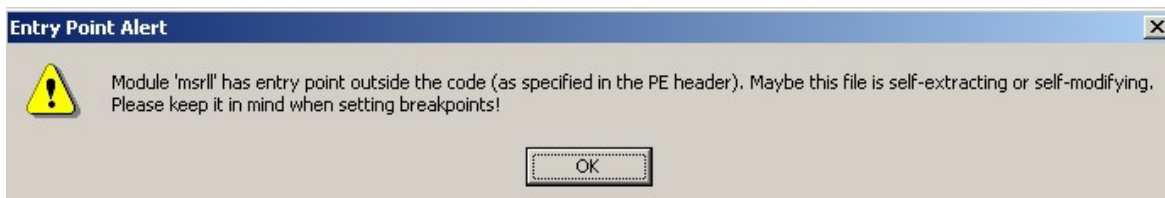


Figure 17: OllyDbg: Entry Point Alert

Then the memory map was opened with ALT+M. The address range, where the interesting code segment was found, was selected. With a right-click the option „Dump in CPU“ was chosen. After setting breakpoints (searching for the interesting address, right click on the line of code and choose breakpoint, type: memory, on access) several times at 00405B48 and 003F6E52 (this one I found more or less by accident) I found the authentication attempt I started on „vm3“ on the IRC channel „#mils“. In the figure below you can see my input:

*?login Mein\_Passwort*

The meaning of „Mein Passwort“ is “my password”. Unfortunately I could not find the part where a password comparison takes place. Nevertheless I was able to see that communication with the malware via IRC was working.

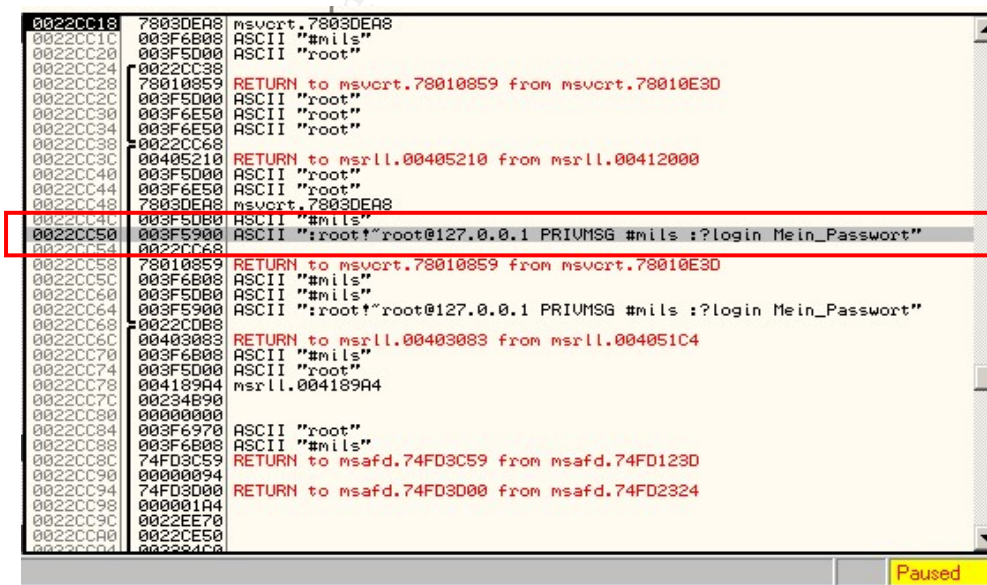


Figure 18: OllyDbg: processing the login attempt from IRC

Maybe the reason, why I could not find the password comparison is the entry point alert I got. I also tried to communicate over tcp port 2200, but I was not able to see this in OllyDbg.



## 6 Analysis Wrap-Up

The provided malware is a species of malicious code called „trojan horse“. After infecting the target machine one could communicate with the trojan via IRC. It seems that there are a lot of commands possible. I think ?jolt for example can be used for a DoS attack against Windows machines. It is possible, that the trojan can be used also for distributed DoS attacks.

There are also some commands which let the attacker execute, copy, move and so on programs and files on the infected machine. Other commands are for distributing and removing the trojan or getting information about the infected machine. So it is possible, that the trojan is used for getting sensitive data from an infected system.

The question “who would use the malware” is hard to answer. Maybe professional hackers for getting sensitive data from infected systems, it also could be that it is just used for denial of service attacks. In the past there were some trojans which provided a smtp relay for spammers. All those trojans show how creative and intelligent the programmers are today. But I could not find a smtp engine within this malware, so it seems not to be used for that.

### Several things can be inter alia learned by analysing such code:

- Always run up to date AV scanners
- Run personal firewalls
- Never open files you get by email from unknown senders
- Configure firewalls at least at the corporate edge very strict – inbound AND outbound
  - Normally your users don't need the ports 9999, 8080, 6667 opened at the corporate firewall. If they are closed, the malware can't communicate with anyone ... If an instant messaging protocol is needed, one should think about implementing an IM proxy, so that the communication can be controlled.
- Keep your systems up to date (patch management, ...)
- In addition it would make sense to run intrusion detection / prevention systems in the corporate network
  - A IDS signature which catches request on tcp port 9999 (which I think is a trojan port) could help to detect an infected system.
- ...

A system infected with the provided trojan has to be taken from the network until it is clean again.

### Unfortunately there are some „problems“ I could not solve:

- I could not identify a way to authenticate against the trojan
- I could not figure out why the trojan tries to connect to the tcp ports 8080 and 9999

- I could not figure out why the trojan opens the tcp port 2200

**To remove the trojan I did the following steps:**

1. I disabled the service „rlt enhanced drive“ as administrator in the services dialog

2. I removed the registry keys (and all subkeys and values)

*HKEY\_LOCAL\_MACHINE\SYSTEM\ControlSet001\Services\mfm* and

*HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\mfm*

After step 1 and after step 2 I rebooted the machine, deleted the folder *C:\WINNT\system32\mfm\* and had a clean system again.

© SANS Institute 2004, Author retains full rights.

## 7 Appendix

### 7.1 Table of Figures

Figure 1: Laboratory Setup .....	6
Figure 2: Linux file command .....	7
Figure 3: MD5 of msrll.exe.....	7
Figure 4: embedded strings in the compressed msrll.exe.....	8
Figure 5: embedded strings in the uncompressed msrll.exe, which look like commands .....	9
Figure 6: Service „Rll enhanced drive“ .....	11
Figure 7: Ethereal capture .....	13
Figure 8: IRC join #mils.....	14
Figure 9: IRC /who #mils.....	14
Figure 10: TDImon: opened tcp ports 2200 and 113.....	15
Figure 11: telnet on tcp ports 2200 and 113 .....	15
Figure 12: processes bevor and after execution of msrll.exe .....	16
Figure 13: IDA Pro: warning while loading an encrypted file .....	17
Figure 14: aspackdie .....	17
Figure 15: LordPE: “dump full” .....	18
Figure 16: IDA Pro: Interesting code segment .....	19
Figure 17: OllyDbg: Entry Point Alert.....	20
Figure 18: OllyDbg: processing the login attempt from IRC.....	20

### 7.2 References

- **“Malware: Fighting Malicious Code”**  
Authors: Ed Skoudis and Lenny Zeltser  
Published by Prentice Hall PTR (<http://www.phptr.com/title/0131014056>).  
ISBN: 0131014056; Published: Nov 7, 2003
- **RFC ident protocol**  
<http://www.faqs.org/rfcs/rfc1413.html>



### 7.3 Bintext output of the with aspackdie uncompressed executable

File pos	Mem pos	ID	Text
=====	=====	==	=====
0000004D	0040004D	0	!This program cannot be run in DOS mode.
00000088	00400088	0	[AspackDie!]
00000178	00400178	0	.text
000001A0	004001A0	0	.data
000001F0	004001F0	0	.idata
00000218	00400218	0	.aspack
00000240	00400240	0	.adata
00001326	00401326	0	?insmod
0000132E	0040132E	0	?rmod
00001335	00401335	0	?lsmode
00001399	00401399	0	%s: <mod name>
000013A8	004013A8	0	%s: mod list full
000013BA	004013BA	0	%s: err: %u
000013C6	004013C6	0	mod_init
000013CF	004013CF	0	mod_free
000013D8	004013D8	0	%s: cannot init %s
000013EB	004013EB	0	%s: %s loaded (%u)
000013FE	004013FE	0	%s: mod already loaded
00001416	00401416	0	%s:%s err %u
000015B5	004015B5	0	%s:%s not found
000015C5	004015C5	0	%s: unloading %s
000016AE	004016AE	0	[%u]: %s hinst:%x
00001712	00401712	0	unloading %s
000017A0	004017A0	0	%s: invalid_addr: %s
000017B5	004017B5	0	%s%s [port]
000018E8	004018E8	0	finished %s
00001A40	00401A40	0	%s <ip> <port> <t_time> <delay>
00001B32	00401B32	0	sockopt: %u
00001B3E	00401B3E	0	sendto err: %u
00001B4D	00401B4D	0	sockraw: %u
00001B59	00401B59	0	syn: done
00001FBC	00401FBC	0	%s <ip> <duration> <delay>
00002096	00402096	0	sendto: %u
000020A2	004020A2	0	jolt2: done
00002260	00402260	0	%s <ip> <p size> <duration> <delay>
00002356	00402356	0	Err: %u
0000235E	0040235E	0	smurf done
00002567	00402567	0	PhV#@
000025DE	004025DE	0	&err: %u
00002753	00402753	0	?ping
00002763	00402763	0	?smurf
0000276A	0040276A	0	?jolt
00002820	00402820	0	PONG :%s
0000283A	0040283A	0	0h (@
0000299D	0040299D	0	%s!%s@%s
00002B3D	00402B3D	0	%s!%s
00002BB6	00402BB6	0	SVh=+@
00002BD7	00402BD7	0	irc.nick
00002BE0	00402BE0	0	NICK %s
00002EEA	00402EEA	0	NETWORK=
00002FF8	00402FF8	0	irc.pre
000032CC	004032CC	0	__%s__
000032D2	004032D2	0	__%s__
000032D9	004032D9	0	__%s__

```

000032E1 004032E1 0 NICK %s
000032F0 004032F0 0 %s %s
000036B0 004036B0 0 irc.chan
00003775 00403775 0 %s %s
0000377B 0040377B 0 WHO %s
000037C8 004037C8 0 PPhV,@

```

```

File pos Mem pos ID Text
=====

```

```

00003A45 00403A45 0 USERHOST %s
00003A52 00403A52 0 logged into %s(%s) as %s
00003A97 00403A97 0 <$hE:@
00003ABB 00403ABB 0 PhR:@
00003B99 00403B99 0 nick.pre
00003BA2 00403BA2 0 %s-%04u
00003BAA 00403BAA 0 irc.user
00003BB3 00403BB3 0 irc.usereal
00003BBF 00403BBF 0 irc.real
00003BC8 00403BC8 0 irc.pass
00003BE0 00403BE0 0 tsend(): connection to %s:%u failed
00003C20 00403C20 0 USER %s localhost 0 :%s
00003C38 00403C38 0 NICK %s
00003DF5 00403DF5 0 Ph <@
000040BF 004040BF 0 PRIVMSG
00004100 00404100 0 trecv(): Disconnected from %s err:%u
0000446B 0040446B 0 NOTICE
00004472 00404472 0 %s %s :%s
00004615 00404615 0 Ph}D@
00004711 00404711 0 MODE %s -o+b %s *@%s
00004798 00404798 0 C'PSWh
000047B4 004047B4 0 Sh'G@
000047E7 004047E7 0 MODE %s -bo %s %s
0000487B 0040487B 0 Sh'G@
00004924 00404924 0 %s.key
00004A63 00404A63 0 Ph'G@
00004AA8 00404AA8 0 sk#%u %s is dead!
00004ABA 00404ABA 0 s_check: %s dead? pinging...
00004AD7 00404AD7 0 PING :ok
00004B00 00404B00 0 s_check: send error to %s disconnecting
00004B28 00404B28 0 expect the worst
00004B39 00404B39 0 s_check: killing socket %s
00004B54 00404B54 0 irc.knick
00004B5E 00404B5E 0 jtr.%u%s.iso
00004B6B 00404B6B 0 ison %s
00004B74 00404B74 0 servers
00004B7C 00404B7C 0 s_check: trying %s
00004DAA 00404DAA 0 Ph9K@
00004ED5 00404ED5 0 PhkK@
00004F41 00404F41 0 ShtK@
00004FD8 00404FD8 0 uYVh|K@
00005052 00405052 0 %s.mode
0000505A 0040505A 0 MODE %s %s
00005078 00405078 0 ShRP@
000050DA 004050DA 0 Sh$I@
000051A8 004051A8 0 PShZP@
000055A3 004055A3 0 mode %s +o %s
000055B2 004055B2 0 akick
000055B8 004055B8 0 mode %s +b %s %s
000055CA 004055CA 0 KICK %s %s
00005760 00405760 0 irc.pre
00005781 00405781 0 Set an irc sock to preform %s command on

```

```

000057AB 004057AB 0 Type
000057B3 004057B3 0 %csklist
000057BC 004057BC 0 to view current sockets, then
000057DC 004057DC 0 %cdccsk
000057E4 004057E4 0 <#>
000058B4 004058B4 0 %s: dll loaded
000058C3 004058C3 0 %s: %d
0000597B 0040597B 0 RhHY@
    
```

```

File pos Mem pos ID Text
=====
    
```

```

000059C6 004059C6 0 RhHY@
000059E1 004059E1 0 said %s to %s
000059EF 004059EF 0 usage: %s <target> "text"
00005A74 00405A74 0 %s not on %s
00005A81 00405A81 0 usage: %s <nick> <chan>
00005B20 00405B20 0 %s logged in
00005B87 00405B87 0 Sh [@
00005BA2 00405BA2 0 sys: %s bot: %s
00005BB2 00405BB2 0 performance counter not avail
00005C2B 00405C2B 0 usage: %s <cmd>
00005C3B 00405C3B 0 %s free'd
00005C45 00405C45 0 unable to free %s
00005C6F 00405C6F 0 Oh+\@
00005CAD 00405CAD 0 later!
00005CB4 00405CB4 0 unable to %s errno:%u
00005D40 00405D40 0 service:%c user:%s inet connection:%c contype:%s reboot privs:%c
00005E09 00405E09 0 Ph@]@
00005E23 00405E23 0 %-5u %s
00005F8F 00405F8F 0 %s: %s
00005F96 00405F96 0 %s: somefile
0000603F 0040603F 0 PhHY@
000060D4 004060D4 0 host: %s ip: %s
00006269 00406269 0 capGetDriverDescriptionA
00006292 00406292 0 cpus:%u
000062A0 004062A0 0 WIN%s (u:%s)%s mem:(%u/%u) %u%% %s %s
000065CB 004065CB 0 %s: %s (%u)
00006708 00406708 0 %s %s
00006754 00406754 0 %s bad args
000067BC 004067BC 0 3hTg@
000067DA 004067DA 0 akick
000067E8 004067E8 0 %s[%u] %s
000067F2 004067F2 0 %s removed
000067FD 004067FD 0 couldnt find %s
0000680D 0040680D 0 %s added
00006816 00406816 0 %s already in list
0000682A 0040682A 0 usage: %s +/- <host>
0000696F 0040696F 0 7h*h@
000069EB 004069EB 0 jtram.conf
000069F6 004069F6 0 %s /t %s
000069FF 004069FF 0 jtr.home
00006A08 00406A08 0 %s\%s
00006A0E 00406A0E 0 %s: possibly failed: code %u
00006A2B 00406A2B 0 %s: possibly failed
00006A3F 00406A3F 0 %s: exec of %s failed err: %u
00006A90 00406A90 0 u.exf
00006C2D 00406C2D 0 Ph+j@
00006C82 00406C82 0 Ph?j@
00006CBC 00406CBC 0 jtr.id
00006CC3 00406CC3 0 %s: <url> <id>
00006ED7 00406ED7 0 IREG
    
```

```

00006EDD 00406EDD 0 CLON
00006EE3 00406EE3 0 ICON
00006EF8 00406EF8 0 WCON
00006F40 00406F40 0 #%u [fd:%u] %s:%u [%s%s] last:%u
00006F63 00406F63 0 |=> [n:%s fh:%s] (%s)
00006F82 00406F82 0 |---[%s] (%u) %s
00006F96 00406F96 0 | |-%s%s] [%s]
00006FAD 00406FAD 0 |=> (%s) (%.8x)
0000716E 0040716E 0 B$PRhco@
00007360 00407360 0 %s <pass> <salt>
    
```

```

File pos Mem pos ID Text
=====
    
```

```

000073C8 004073C8 0 %s <nick> <chan>
0000748B 0040748B 0 PING %s
000074C9 004074C9 0 mIRC v6.12 Khaled Mardam-Bey
000074E7 004074E7 0 VERSION %s
0000751C 0040751C 0 dcc.pass
00007525 00407525 0 temp add %s
000075BD 004075BD 0 $h%u@
0000766A 0040766A 0 %s%u-%s
00007675 00407675 0 %s opened (%u)
000076A0 004076A0 0 %u bytes from %s in %u seconds saved to %s
000076CB 004076CB 0 (%s %s): incomplete! %u bytes
000076E9 004076E9 0 couldnt open %s err:%u
00007700 00407700 0 (%s) %s: %s
0000770C 0040770C 0 (%s) urlopen failed
00007720 00407720 0 (%s): inetopen failed
00007798 00407798 0 Whjv@
00007B9D 00407B9D 0 Ph w@
00007BE4 00407BE4 0 no file name in %s
00007DDB 00407DDB 0 %s created
00007E49 00407E49 0 %s %s to %s Ok
00007E8F 00407E8F 0 3hl~@
00007EE0 00407EE0 0 %0.2u/%0.2u/%0.2u %0.2u:%0.2u %15s %s
00007F09 00407F09 0 %s (err: %u)
0000806B 0040806B 0 ShHY@
00008085 00408085 0 err: %u
000080F8 004080F8 0 %s %s :ok
00008165 00408165 0 unable to %s %s (err: %u)
000081C3 004081C3 0 ShHY@
000081F5 004081F5 0 %-16s %s
00008200 00408200 0 %-16s (%u.%u.%u.%u)
00008489 00408489 0 [%s][%s] %s
00008595 00408595 0 closing %u [%s:%u]
000085A8 004085A8 0 unable to close socket %u
000087E2 004087E2 0 using sock #%u %s:%u (%s)
000087FD 004087FD 0 Invalid sock
0000880B 0040880B 0 usage %s <socks #>
000088D7 004088D7 0 leaves %s
000088E1 004088E1 0 :0 * * :%s
00008A96 00408A96 0 joins: %s
00008B82 00408B82 0 ACCEPT
00008B89 00408B89 0 resume
00008B90 00408B90 0 err: %u
00008B99 00408B99 0 DCC ACCEPT %s %s %s
00008BAE 00408BAE 0 dcc_resume: cant find port %s
00008BD1 00408BD1 0 dcc.dir
00008BD9 00408BD9 0 %s\s\s\s\s\s
00008BE5 00408BE5 0 unable to open (%s): %u
00008BFD 00408BFD 0 resuming dcc from %s to %s
    
```

```

00008C19 00408C19 0 DCC RESUME %s %s %u
0000934E 0040934E 0 ?clone
00009355 00409355 0 ?clones
0000935D 0040935D 0 ?login
00009364 00409364 0 ?uptime
0000936C 0040936C 0 ?reboot
00009374 00409374 0 ?status
0000937C 0040937C 0 ?jump
00009382 00409382 0 ?nick
00009388 00409388 0 ?echo
0000938E 0040938E 0 ?hush
00009394 00409394 0 ?wget

```

```

File pos  Mem pos  ID  Text
=====  =====  ==  =====

```

```

0000939A 0040939A 0 ?join
000093A9 004093A9 0 ?akick
000093B0 004093B0 0 ?part
000093B6 004093B6 0 ?dump
000093C6 004093C6 0 ?md5p
000093CC 004093CC 0 ?free
000093D7 004093D7 0 ?update
000093DF 004093DF 0 ?hostname
000093EE 004093EE 0 ?!fff
000093FE 004093FE 0 ?play
00009404 00409404 0 ?copy
0000940A 0040940A 0 ?move
00009415 00409415 0 ?sums
00009423 00409423 0 ?rmdir
0000942A 0040942A 0 ?mkdir
00009436 00409436 0 ?exec
00009440 00409440 0 ?kill
00009446 00409446 0 ?killall
0000944F 0040944F 0 ?crash
0000946E 0040946E 0 ?sklist
00009476 00409476 0 ?unset
0000947D 0040947D 0 ?uattr
00009484 00409484 0 ?dccsk
00009490 00409490 0 ?killsk
00009499 00409499 0 VERSION*
000094AE 004094AE 0 IDENT
000096BE 004096BE 0 %ud %02uh %02um %02us
000096D4 004096D4 0 %02uh %02um %02us
000096E6 004096E6 0 %um %02us
000099E0 004099E0 0 jtram.conf
000099EB 004099EB 0 jtr.*
000099F5 004099F5 0 DiCHFc2ioiVmb3cb4zZ7zWZH1oM=
00009A16 00409A16 0 conf_dump: wrote %u lines
0000A270 0040A270 0 get of %s incomplete at %u bytes
0000A2B0 0040A2B0 0 get of %s completed (%u bytes), %u seconds %u cps
0000A2F0 0040A2F0 0 error while writing to %s (%u)
0000A65C 0040A65C 0 chdir: %s -> %s (%u)
0000A750 0040A750 0 dcc_wait: get of %s from %s timed out
0000A790 0040A790 0 dcc_wait: closing [#%u] %s:%u (%s)
0000A9F0 0040A9F0 0 %4s #%.2u %s %ucps %u%% [sk#%u] %s
0000AA30 0040AA30 0 %u Send(s) %u Get(s) (%u transfer(s) total) UP:%ucps DOWN:%ucps Total:%ucps
0000AC95 0040AC95 0 PRQh0
0000ACD0 0040ACD0 0 send of %s incomplete at %u bytes
0000AD10 0040AD10 0 send of %s completed (%u bytes), %u seconds %u cps
0000AF50 0040AF50 0 cant open %s (err:%u) pwd:{%s}
0000AF70 0040AF70 0 DCC SEND %s %u %u %u

```

```

0000B751 0040B751 0 %s %s
0000B757 0040B757 0 %s exited with code %u
0000B76E 0040B76E 0 %s\%s
0000B774 0040B774 0 %s: %s
0000B77B 0040B77B 0 exec: Error:%u pwd:%s cmd:%s
0000BB40 0040BB40 0 dcc.pass
0000BB49 0040BB49 0 bot.port
0000BB52 0040BB52 0 %s bad pass from "%s" @ %s
0000BCC9 0040BCC9 0 %s: connect from %s
0000BD33 0040BD33 0 jtr.bin
0000BD3B 0040BD3B 0 msrll.exe
0000BD45 0040BD45 0 jtr.home
0000BD57 0040BD57 0 jtr.id
0000BD63 0040BD63 0 irc.quit
    
```

```

File pos Mem pos ID Text
=====
    
```

```

0000BD6E 0040BD6E 0 servers
0000BD80 0040BD80 0 collective7.zxy0.com,collective7.zxy0.com:9999!,collective7.zxy0.com:8080
0000BDCA 0040BDCA 0 irc.chan
0000BDD3 0040BDD3 0 #mils
0000BDE0 0040BDE0 0 $1$KZLPLKdF$W8kl8Jr1X8DOHZsmIp9qq0
0000BE20 0040BE20 0 $1$KZLPLKdF$55isA1ITvamR7bjAdBziX.
0000C02F 0040C02F 0 SSL_get_error
0000C03D 0040C03D 0 SSL_load_error_strings
0000C054 0040C054 0 SSL_library_init
0000C065 0040C065 0 SSLv3_client_method
0000C079 0040C079 0 SSL_set_connect_state
0000C08F 0040C08F 0 SSL_CTX_new
0000C09B 0040C09B 0 SSL_new
0000C0A3 0040C0A3 0 SSL_set_fd
0000C0AE 0040C0AE 0 SSL_connect
0000C0BA 0040C0BA 0 SSL_write
0000C0C4 0040C0C4 0 SSL_read
0000C0CD 0040C0CD 0 SSL_shutdown
0000C0DA 0040C0DA 0 SSL_free
0000C0E3 0040C0E3 0 SSL_CTX_free
0000C263 0040C263 0 kernel32.dll
0000C270 0040C270 0 QueryPerformanceCounter
0000C288 0040C288 0 QueryPerformanceFrequency
0000C2A2 0040C2A2 0 RegisterServiceProcess
0000C2B9 0040C2B9 0 jtram.conf
0000C5B1 0040C5B1 0 irc.user
0000C5BA 0040C5BA 0 %s : USERID : UNIX : %s
0000C6A4 0040C6A4 0 QUIT :FUCK %u
0000C742 0040C742 0 Killed!? Arrg! [%u]
0000C756 0040C756 0 QUIT :%s
0000C7E8 0040C7E8 0 SeShutdownPrivilege
0000C888 0040C888 0 %s\%s
0000C88E 0040C88E 0 %s\%s\%s
0000C897 0040C897 0 Rll enhanced drive
0000C8C0 0040C8C0 0 software\microsoft\windows\currentversion\run
0000C8EE 0040C8EE 0 /d "%s"
0000CE3D 0040CE3D 0 < u&
0000D010 0040D010 0 ./0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
0000EA60 0040EA60 0 usage %s: server[:port] amount
0000EB33 0040EB33 0 %s: %s
0000EB3E 0040EB3E 0 %s %s %s <PARAM>
0000EB80 0040EB80 0 %s: [NETWORK|all] %s <"parm"> ...
0000EE20 0040EE20 0 USER %s localhost 0 :%s
0000EE38 0040EE38 0 NICK %s
    
```

```

0000EEE4 0040EEE4 0 PSVh
0000F140 0040F140 0 md5.c
0000F146 0040F146 0 md != NULL
0000F8F1 0040F8F1 0 buf != NULL
0000F99F 0040F99F 0 hash != NULL
0000FAC5 0040FAC5 0 message digest
0000FAD4 0040FAD4 0 abcdefghijklmnopqrstuvwxyz
0000FB00 0040FB00 0 ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
0000FB40 0040FB40
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
0000FCE0 0040FCE0 0 sprng
0000FD11 0040FD11 0 sprng.c
0000FD19 0040FD19 0 buf != NULL
0000FDBC 0040FDBC 0 rc6.c
0000FDC2 0040FDC2 0 skey != NULL
0000FDCF 0040FDCF 0 key != NULL
0000FFD1 0040FFD1 0 ct != NULL

```

0

```

File pos Mem pos ID Text

```

```

=====

```

```

0000FFDC 0040FFDC 0 pt != NULL
0001023E 0041023E 0 #4EVgx
00010256 00410256 0 $5FWhy
00010282 00410282 0 #4EVgx
0001029A 0041029A 0 $5FWhy
000102C6 004102C6 0 #4EVgx
000102DE 004102DE 0 $5FWhy
000102F8 004102F8 0 gN]HU
000103C3 004103C3 0 desired_keysize != NULL
00010430 00410430 0 ctr.c
00010436 00410436 0 ctr != NULL
00010442 00410442 0 key != NULL
0001044E 0041044E 0 count != NULL
00010546 00410546 0 ct != NULL
00010551 00410551 0 pt != NULL
000106F0 004106F0 0 ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+
0001077F 0041077F 0 ?456789;<=
000107B7 004107B7 0 !"#%&'()*+,-./0123
00010850 00410850 0 base64.c
00010859 00410859 0 outlen != NULL
00010868 00410868 0 out != NULL
00010874 00410874 0 in != NULL
00010B30 00410B30 0 _ARGCHK '%s' failure on line %d of file %s
00010B8B 00410B8B 0 crypt.c
00010B93 00410B93 0 name != NULL
00010D79 00410D79 0 cipher != NULL
00010E70 00410E70 0 hash != NULL
00010F7A 00410F7A 0 prng != NULL
000110F0 004110F0 0 LibTomCrypt 0.83
00011102 00411102 0 Endianess: little (32-bit words)
00011123 00411123 0 Clean stack: disabled
00011139 00411139 0 Ciphers built-in:
0001114B 0041114B 0 Blowfish
00011157 00411157 0 RC2
0001115E 0041115E 0 RC5
00011165 00411165 0 RC6
0001116C 0041116C 0 Serpent
00011177 00411177 0 Safer+
00011181 00411181 0 Safer
0001118A 0041118A 0 Rijndael
00011196 00411196 0 XTEA

```

```

0001119E 0041119E 0 Twofish
000111AA 004111AA 0 CAST5
000111B3 004111B3 0 Noekeon
000111BF 004111BF 0 Hashes built-in:
000111D0 004111D0 0 SHA-512
000111DB 004111DB 0 SHA-384
000111E6 004111E6 0 SHA-256
000111F1 004111F1 0 TIGER
000111FA 004111FA 0 SHA1
00011202 00411202 0 MD5
00011209 00411209 0 MD4
00011210 00411210 0 MD2
00011218 00411218 0 Block Chaining Modes:
0001122E 0041122E 0 CFB
00011235 00411235 0 OFB
0001123C 0041123C 0 CTR
00011244 00411244 0 PRNG:
0001124A 0041124A 0 Yarrow
00011254 00411254 0 SPRNG
    
```

```

File pos Mem pos ID Text
=====
0001125D 0041125D 0 RC4
00011265 00411265 0 PK Algs:
0001126E 0041126E 0 RSA
00011275 00411275 0 DH
0001127B 0041127B 0 ECC
00011282 00411282 0 KR
00011289 00411289 0 Compiler:
00011293 00411293 0 WIN32 platform detected.
000112AF 004112AF 0 GCC compiler detected.
000112CA 004112CA 0 Various others: BASE64 MPI HMAC
00011313 00411313 0 /dev/random
00011430 00411430 0 Microsoft Base Cryptographic Provider v1.0
000114D2 004114D2 0 bits.c
000114D9 004114D9 0 buf != NULL
000114F6 004114F6 0 t9VWS
0001154A 0041154A 0 prng != NULL
00011832 00411832 0 <"tx< tf< t
00011846 00411846 0 < tV< t
00011852 00411852 0 < tJ< tF
00011A10 00411A10 0 -LIBGCCW32-EH-SJLJ-GTHR-MINGW32
000130B0 004130B0 0 <ip> <total secs> <p size> <delay>
00013350 00413350 0 modem
00013358 00413358 0 Lan
0001335E 0041335E 0 Proxy
0001336B 0041336B 0 none
00013390 00413390 0 m220 1.0 #2730 Mar 16 11:47:38 2004
000133D4 004133D4 0 unable to %s %s (err: %u)
00013420 00413420 0 unable to kill %s (%u)
00013437 00413437 0 %s killed (pid:%u)
00013470 00413470 0 AVICAP32.dll
0001347D 0041347D 0 unable to kill %u (%u)
00013494 00413494 0 pid %u killed
000134A2 004134A2 0 error!
000134A9 004134A9 0 ran ok
000134B0 004134B0 0 MODE %s +o %s
000134BF 004134BF 0 set %s %s
00013600 00413600 0 Mozilla/4.0
0001360C 0041360C 0 Accept: */*
0001361C 0041361C 0 <DIR>
    
```

Author retains full rights.



```

0001362B 0041362B 0 Could not copy %s to %s
00013643 00413643 0 %s copied to %s
00013653 00413653 0 0123456789abcdef
00013664 00413664 0 %s unset
0001366D 0041366D 0 unable to unset %s
00013AD4 00413AD4 0 (%s) %s
00013ADD 00413ADD 0 %s %s
00013BA0 00413BA0 0 libssl32.dll
00013BAD 00413BAD 0 libeay32.dll
00013BE0 00413BE0 0 <die|join|part|raw|msg>
0011B67A 0051B67A 0 AdjustTokenPrivileges
0011B692 0051B692 0 CloseServiceHandle
0011B6AA 0051B6AA 0 CreateServiceA
0011B6BE 0051B6BE 0 CryptAcquireContextA
0011B6D6 0051B6D6 0 CryptGenRandom
0011B6EA 0051B6EA 0 CryptReleaseContext
0011B702 0051B702 0 GetUserNameA
0011B712 0051B712 0 LookupPrivilegeValueA
0011B72A 0051B72A 0 OpenProcessToken
0011B73E 0051B73E 0 OpenSCManagerA
0011B752 0051B752 0 RegCloseKey
    
```

```

File pos  Mem pos  ID  Text
=====  =====  ==  ====
    
```

```

0011B762 0051B762 0 RegCreateKeyExA
0011B776 0051B776 0 RegSetValueExA
0011B78A 0051B78A 0 RegisterServiceCtrlHandlerA
0011B7AA 0051B7AA 0 SetServiceStatus
0011B7BE 0051B7BE 0 StartServiceCtrlDispatcherA
0011B7DE 0051B7DE 0 AddAtomA
0011B7EA 0051B7EA 0 CloseHandle
0011B7FA 0051B7FA 0 CopyFileA
0011B806 0051B806 0 CreateDirectoryA
0011B81A 0051B81A 0 CreateFileA
0011B82A 0051B82A 0 CreateMutexA
0011B83A 0051B83A 0 CreatePipe
0011B84A 0051B84A 0 CreateProcessA
0011B85E 0051B85E 0 CreateToolhelp32Snapshot
0011B87A 0051B87A 0 DeleteFileA
0011B88A 0051B88A 0 DuplicateHandle
0011B89E 0051B89E 0 EnterCriticalSection
0011B8B6 0051B8B6 0 ExitProcess
0011B8C6 0051B8C6 0 ExitThread
0011B8D6 0051B8D6 0 FileTimeToSystemTime
0011B8EE 0051B8EE 0 FindAtomA
0011B8FA 0051B8FA 0 FindClose
0011B906 0051B906 0 FindFirstFileA
0011B91A 0051B91A 0 FindNextFileA
0011B92A 0051B92A 0 FreeLibrary
0011B93A 0051B93A 0 GetAtomNameA
0011B94A 0051B94A 0 GetCommandLineA
0011B95E 0051B95E 0 GetCurrentDirectoryA
0011B976 0051B976 0 GetCurrentProcess
0011B98A 0051B98A 0 GetCurrentThreadId
0011B9A2 0051B9A2 0 GetExitCodeProcess
0011B9BA 0051B9BA 0 GetFileSize
0011B9CA 0051B9CA 0 GetFullPathNameA
0011B9DE 0051B9DE 0 GetLastError
0011B9EE 0051B9EE 0 GetModuleFileNameA
0011BA06 0051BA06 0 GetModuleHandleA
0011BA1A 0051BA1A 0 GetProcAddress
    
```

0011BA2E 0051BA2E 0 GetStartupInfoA  
0011BA42 0051BA42 0 GetSystemDirectoryA  
0011BA5A 0051BA5A 0 GetSystemInfo  
0011BA6A 0051BA6A 0 GetTempPathA  
0011BA7A 0051BA7A 0 GetTickCount  
0011BA8A 0051BA8A 0 GetVersionExA  
0011BA9A 0051BA9A 0 GlobalMemoryStatus  
0011BAB2 0051BAB2 0 InitializeCriticalSection  
0011BACE 0051BACE 0 IsBadReadPtr  
0011BADE 0051BADE 0 LeaveCriticalSection  
0011BAF6 0051BAF6 0 LoadLibraryA  
0011BB06 0051BB06 0 MoveFileA  
0011BB12 0051BB12 0 OpenProcess  
0011BB22 0051BB22 0 PeekNamedPipe  
0011BB32 0051BB32 0 Process32First  
0011BB46 0051BB46 0 Process32Next  
0011BB56 0051BB56 0 QueryPerformanceFrequency  
0011BB72 0051BB72 0 ReadFile  
0011BB7E 0051BB7E 0 ReleaseMutex  
0011BB8E 0051BB8E 0 RemoveDirectoryA  
0011BBA2 0051BBA2 0 SetConsoleCtrlHandler  
0011BBBA 0051BBBA 0 SetCurrentDirectoryA  
0011BBD2 0051BBD2 0 SetFilePointer

File pos Mem pos ID Text  
=====

0011BBE6 0051BBE6 0 SetUnhandledExceptionFilter  
0011BC06 0051BC06 0 Sleep  
0011BC0E 0051BC0E 0 TerminateProcess  
0011BC22 0051BC22 0 WaitForSingleObject  
0011BC3A 0051BC3A 0 WriteFile  
0011BC46 0051BC46 0 \_itoa  
0011BC4E 0051BC4E 0 \_stat  
0011BC56 0051BC56 0 \_strdup  
0011BC62 0051BC62 0 \_stricmp  
0011BC6E 0051BC6E 0 \_\_getmainargs  
0011BC7E 0051BC7E 0 \_\_p\_\_environ  
0011BC8E 0051BC8E 0 \_\_p\_\_fmode  
0011BC9E 0051BC9E 0 \_\_set\_app\_type  
0011BCB2 0051BCB2 0 \_beginthread  
0011BCC2 0051BCC2 0 \_cexit  
0011BCCE 0051BCCE 0 \_errno  
0011BCDA 0051BCDA 0 \_fileno  
0011BCEE 0051BCEE 0 \_onexit  
0011BCFA 0051BCFA 0 \_setmode  
0011BD06 0051BD06 0 \_vsprintf  
0011BD16 0051BD16 0 abort  
0011BD1E 0051BD1E 0 atexit  
0011BD32 0051BD32 0 clock  
0011BD3A 0051BD3A 0 fclose  
0011BD46 0051BD46 0 fflush  
0011BD52 0051BD52 0 fgets  
0011BD5A 0051BD5A 0 fopen  
0011BD62 0051BD62 0 fprintf  
0011BD6E 0051BD6E 0 fread  
0011BD7E 0051BD7E 0 fwrite  
0011BD8A 0051BD8A 0 malloc  
0011BD96 0051BD96 0 memcpy  
0011BDA2 0051BDA2 0 memset  
0011BDAE 0051BDAE 0 printf  
0011BDBA 0051BDBA 0 raise

```

0011BDCA 0051BDCA 0 realloc
0011BDD6 0051BDD6 0 setvbuf
0011BDE2 0051BDE2 0 signal
0011BDEE 0051BDEE 0 sprintf
0011BDFA 0051BDFA 0 srand
0011BE02 0051BE02 0 strcat
0011BE0E 0051BE0E 0 strchr
0011BE1A 0051BE1A 0 strcmp
0011BE26 0051BE26 0 strcpy
0011BE32 0051BE32 0 strerror
0011BE3E 0051BE3E 0 strncat
0011BE4A 0051BE4A 0 strncmp
0011BE56 0051BE56 0 strncpy
0011BE62 0051BE62 0 strstr
0011BE76 0051BE76 0 toupper
0011BE82 0051BE82 0 ShellExecuteA
0011BE92 0051BE92 0 DispatchMessageA
0011BEA6 0051BEA6 0 ExitWindowsEx
0011BEB6 0051BEB6 0 GetMessageA
0011BEC6 0051BEC6 0 PeekMessageA
0011BED6 0051BED6 0 GetFileVersionInfoA
0011BEEE 0051BEEE 0 VerQueryValueA
0011BF02 0051BF02 0 InternetCloseHandle
0011BF1A 0051BF1A 0 InternetGetConnectedState
0011BF36 0051BF36 0 InternetOpenA
    
```

```

File pos  Mem pos  ID  Text
=====  =====  ==  =====
    
```

```

0011BF46 0051BF46 0 InternetOpenUrlA
0011BF5A 0051BF5A 0 InternetReadFile
0011BF6E 0051BF6E 0 WSAGetLastError
0011BF82 0051BF82 0 WSASocketA
0011BF92 0051BF92 0 WSASStartup
0011BFA2 0051BFA2 0 __WSAFDIsSet
0011BFB2 0051BFB2 0 accept
0011BFC6 0051BFC6 0 closesocket
0011BFD6 0051BFD6 0 connect
0011BFE2 0051BFE2 0 gethostbyaddr
0011BFF2 0051BFF2 0 gethostbyname
0011C002 0051C002 0 gethostname
0011C012 0051C012 0 getsockname
0011C022 0051C022 0 htonl
0011C02A 0051C02A 0 htons
0011C032 0051C032 0 inet_addr
0011C03E 0051C03E 0 inet_ntoa
0011C04A 0051C04A 0 ioctlsocket
0011C05A 0051C05A 0 listen
0011C066 0051C066 0 ntohl
0011C076 0051C076 0 select
0011C08A 0051C08A 0 sendto
0011C096 0051C096 0 setsockopt
0011C0A6 0051C0A6 0 shutdown
0011C0B2 0051C0B2 0 socket
0011C0FC 0051C0FC 0 ADVAPI32.DLL
0011C1FC 0051C1FC 0 KERNEL32.dll
0011C21C 0051C21C 0 msvcrt.dll
0011C2E0 0051C2E0 0 msvcrt.dll
0011C2F0 0051C2F0 0 SHELL32.DLL
0011C30C 0051C30C 0 USER32.dll
0011C320 0051C320 0 VERSION.dll
0011C340 0051C340 0 WININET.DLL
    
```

```

0011C3B4 0051C3B4 0 WS2_32.DLL
0011D071 0051D071 0 VirtualAlloc
0011D07E 0051D07E 0 VirtualFree
0011D441 0051D441 0 kernel32.dll
0011D44E 0051D44E 0 ExitProcess
0011D45A 0051D45A 0 user32.dll
0011D465 0051D465 0 MessageBoxA
0011D471 0051D471 0 wsprintfA
0011D47B 0051D47B 0 LOADER ERROR
0011D488 0051D488 0 The procedure entry point %s could not be located in the dynamic link library %s
0011D4D9 0051D4D9 0 The ordinal %u could not be located in the dynamic link library %s
0011D6E6 0051D6E6 0 (08@P
0011D874 0051D874 0 D4||M
0011D9C0 0051D9C0 0 ;;F,s
0011D9CF 0051D9CF 0 ;;F0s
0011D9DB 0051D9DB 0 ;;F4s
0011DCB5 0051DCB5 0 D$$W3
0011DF6C 0051DF6C 0 kernel32.dll
0011DF7B 0051DF7B 0 GetProcAddress
0011DF8C 0051DF8C 0 GetModuleHandleA
0011DF9F 0051DF9F 0 LoadLibraryA
0011E074 0051E074 0 advapi32.dll
0011E081 0051E081 0 msvcrt.dll
0011E08C 0051E08C 0 msvcrt.dll
0011E097 0051E097 0 shell32.dll
0011E0A3 0051E0A3 0 user32.dll
0011E0AE 0051E0AE 0 version.dll

```

```

File pos Mem pos ID Text
=====
0011E0BA 0051E0BA 0 wininet.dll
0011E0C6 0051E0C6 0 ws2_32.dll
0011E113 0051E113 0 AdjustTokenPrivileges
0011E12B 0051E12B 0 _itoa
0011E133 0051E133 0 __getmainargs
0011E143 0051E143 0 ShellExecuteA
0011E153 0051E153 0 DispatchMessageA
0011E166 0051E166 0 GetFileVersionInfoA
0011E17C 0051E17C 0 InternetCloseHandle
0011E192 0051E192 0 WSAGetLastError

```

© SANS Institute 2004, Author retains full rights.

### 7.4 Snort capture of IRC Login and ident request

```

==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+
09/20-03:56:46.772149 192.168.5.131:6667 -> 192.168.5.129:1033
TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:48 DF
***A**S* Seq: 0x598189C8 Ack: 0x8E39ADE9 Win: 0x16D0 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+
09/20-03:56:46.792695 192.168.5.129:1033 -> 192.168.5.131:6667
TCP TTL:128 TOS:0x0 ID:217 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x8E39ADE9 Ack: 0x598189C9 Win: 0x4470 TcpLen: 20

==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+~+~+~+~+
09/20-03:56:46.793509 192.168.5.131:6667 -> 192.168.5.129:1033
TCP TTL:64 TOS:0x0 ID:37211 IpLen:20 DgmLen:86 DF
***AP*** Seq: 0x598189C9 Ack: 0x8E39ADE9 Win: 0x16D0 TcpLen: 20
4E 4F 54 49 43 45 20 41 55 54 48 20 3A 2A 2A 2A NOTICE AUTH :***
20 4C 6F 6F 6B 69 6E 67 20 75 70 20 79 6F 75 72 Looking up your
20 68 6F 73 74 6E 61 6D 65 2E 2E 2E 0D 0A hostname.....

==+==+~+~+~+~+~+~+~+~+~+~+~+~+~+~+~+~+~+~+~+~+~+~+
09/20-03:56:46.793706 192.168.5.131:1024 -> 192.168.5.1:53
UDP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:72 DF
Len: 52
31 20 01 00 00 01 00 00 00 00 00 03 31 32 39 1 .....129
01 35 03 31 36 38 03 31 39 32 07 69 6E 2D 61 64 .5.168.192.in-ad
64 72 04 61 72 70 61 00 00 0C 00 01 dr.arpa.....

==+==+==+==+==+==+==+==+==+==+==+==+~+~+~+~+~+~+~+~+~+~+~+~+
09/20-03:56:46.793963 192.168.5.131:1027 -> 192.168.5.129:113 → IDENT
TCP TTL:64 TOS:0x0 ID:62823 IpLen:20 DgmLen:60 DF
*****S* Seq: 0x58E4EDF5 Ack: 0x0 Win: 0x16D0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 137362 0 NOP WS: 0

==+==+==+==+==+==+==+~+~+~+~+~+~+~+~+~+~+~+~+~+~+~+~+~+~+~+~+~+
09/20-03:56:46.820793 192.168.5.129:113 -> 192.168.5.131:1027

```



# Upcoming Training

Click Here to  
**{Get CERTIFIED!}**



SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Community SANS Ottawa FOR610	Ottawa, ON	Sep 18, 2017 - Sep 23, 2017	Community SANS
Baltimore Fall 2017 - FOR610: Reverse-Engineering Malware: Malware Analysis Tools and Techniques	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Boston FOR610	Boston, MA	Oct 09, 2017 - Oct 14, 2017	Community SANS
SANS Tokyo Autumn 2017	Tokyo, Japan	Oct 16, 2017 - Oct 28, 2017	Live Event
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
Cyber Threat Intelligence Summit & Training 2018	Bethesda, MD	Jan 29, 2018 - Feb 05, 2018	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced