

# Reversing The Malware

A Manual and Automated Detection Approach

Author: Shakeel Ali

Organization: Cipher Storm Ltd (UK)

Date Published: 07 September 2009

Publisher: *Prima Infosarana Media PT (InfoKomputer)*



## Introduction

A 'malware' is a piece of code written by malicious adversary to infiltrate the computer systems without user's permission. With the vertical growth in adoption of IT systems in corporate networks, data centers, financial institutions and educational sector has revealed the key electronic threats, out of which a malware is considered as a powerful driver. To let you understand the scope of malware threat, the following table will summarize the malware categories, targeted operating systems, their infection vectors and generic symptoms to identify them. Rest of the article will focus through various reverse engineering and detection intelligent techniques to fight against these malwares in manual and automated fashion.

<b>Malware Types</b>	Viruses, Worms, Trojans, Rookits, Spyware, Backdoors, Keylogger
<b>Targeted OS(s)</b>	Windows, Linux, Unix, Mac OSX, Symbian, Mobile PC
<b>Infection Vector</b>	System files, boot sector, macros, emails, P2P, IM messaging, bluetooth, web applications and many others
<b>Generic Symptoms</b>	Unknown network traffic, infected emails, unwanted executables in the system, abnormal system behavior, disabled security software, degrade in system performance, loss of financial assets, etc.

**Table 1: Malware Types, Targeted OS(s), Infection Vector, Generic Symptoms**

## Malware Reverse Engineering

The recent increase in sophistication of malware attacks has put forward exceptional challenges for the reverse engineers to examine the code in-depth and extract the fully functional source to track back its creators. Several known factors affecting this investigation process involve, use of encryption, code obfuscation methods, advanced packers and protectors. These sort of tools and techniques may allow an attacker to easily bypass corporate firewalls, anti-viruses, intrusion detection and prevention systems (IDS/IPS). However, to remediate against such threats, use of industry's best debuggers, decompilers, unpackers, disassemblers, sandbox environment, data extraction and deobfuscation tools comes into practice to purely identify the roots of these malwares and prevent any data being stolen. Before stepping into reversing methodology, it is wise to understand the security toolset and environment in which the analysis can be performed.

### 1. Tools of the Trade

#### *Debuggers*

**Usage:** Tracing the program execution to find and remove the known bugs.

**Tools:** OllyDbg, Malzilla, EDB, GDB, Radar, RosAsm, WinAppDbg, WinDbg, W32Dasm, Firebug, Immunity Debugger, PaiMei, SoftICE, Wintruder, VMKD, Hackman Suite



### ***Decompilers***

**Usage:** Perform reverse translation of low-level code to the higher-level language code.

**Tools:** Reflector .Net, Coddec, DE Decompiler, Hex-Rays, RaceVB6, RecStudio, SWF Decompiler, DJ Java Decompiler

### ***Deobfuscation Tools***

**Usage:** Obfuscation techniques make the code harder to read when decompiled, however, de-obfuscation methods convert them into readable code which makes it easier to analyze.

**Tools:** .NET DeObfuscator, Malzilla, dotNetTools Win32, pynary

### ***Disassemblers***

**Usage:** The purpose of disassembler is to translate the binary code into assembly code. It differs from decompiler which usually targets high-level language code instead of assembly code.

**Tools:** IDA Pro, P32DASM, Radare, PVDasm, BeaEngine, Hiew, dedexer, Flasm, Nemo440, PE Explorer, PEBrowse

### ***Unpackers***

**Usage:** Unpackers usually assist in packer identification and decompression of the original binary data which has been packed using standard packers.

**Tools:** ArmaGeddon, SysAnalyzer, Wildtangent Unwrapper, Explorer Suite, Memoryze, ImpREC, .NET Generic Unpacker, ActiveMARK Decrypter, ap0x Unpack Engine SDK, dotNet Sniffer Win32, GUnPacker, MSIL Dumper, Quick Unpack, Smartassassin, swfdecrypt, PEiD, Universal Import Fixer, xTracer

### ***Hex Editors***

**Usage:** The main purpose of hex editor is to assist the binary code or byte code analysis. They are particularly helpful in computer forensics, low-level data processing and data recovery.

**Tools:** Explorer Suite, FileInsight, WinHex, Hiew, Hex Workshop

### ***Code Profilers***

**Usage:** Code profilers are helpful to measure the performance of the executable code. This profiling technique may also help the investigator to record the program's behavior on execution.

**Tools:** CodeAnalyst Performance Analyzer, AQtime, DevPartner Studio, Hotch, Profile Coverage Tool, Process Stalker

### ***Reverse Engineering Frameworks***

**Usage:** These frameworks are provided with a complete set of tools necessary to conduct forensic operations on known or unknown binaries.

**Tools:** Fenris, Radare, Damn Vulnerable Linux, ERESI Framework, Malcode Analysis Pack, PaiMei

### *Network Monitoring Tools*

**Usage:** Monitoring the network activity is helpful while examining the execution of a malware which may lead to an important clue about unauthorized network connections.

**Tools:** Wireshark, SysAnalyzer, Sysinternals Tools, Tcpdump, Fport, Malcode Analysis Pack, oSpy

### *Automated Sandbox Environments*

**Usage:** To save the time and cost for analyzing the malware more efficiently and quickly, these threat analysis systems can be used to perform fast dynamic behavioral breakdowns and advanced heuristics on malware activity.

**Tools:** Anubis, ThreatExpert, Bochs Emulator, CWSandbox, Joebox, NoVirusThanks, Norman SandBox, Sunbelt Sandbox, Sandboxie, VirSCAN, VirusTotal

### *Virtualization*

**Usage:** In order to facilitate the efficient and reliable malware analysis process, it is highly recommended to use the controlled and isolated environment that is flexible and unobtrusive.

**Tools:** VMWare, Virtual PC, Virtual Box, XenExpress

## **2. Methodology**

### **i. Manual Detection Approach**

Examining the malware manually is one of the standard approaches used by many organizations to conduct post forensics on data theft incidents. It also worth noticing that the manual detection is a time and cost incentive. However, the investigator has more flexibility to understand and record each set of activity done by the malware. Some of the key steps followed under this approach are given below:

1. Set up the independent lab in a controlled and isolated environment.
2. Conduct the behavioral analysis using different set of tools to detect the malware activities. Such as, monitor the local and network interactions.
3. To understand the inner workings of a malware perform static code analysis.
4. Unpack the malware, if needed, using various tools and techniques. For instance, use PEiD tool for packer identification, use Ollydbg or IDA Pro for the extraction of infected executable. If unsuccessful, perform dynamic code analysis on packed executable.
5. Carry out repetitive tests to check the integrity and accuracy of your results.
6. Document your findings according to industry's acceptable standards.

## ii. Automated Detection Approach

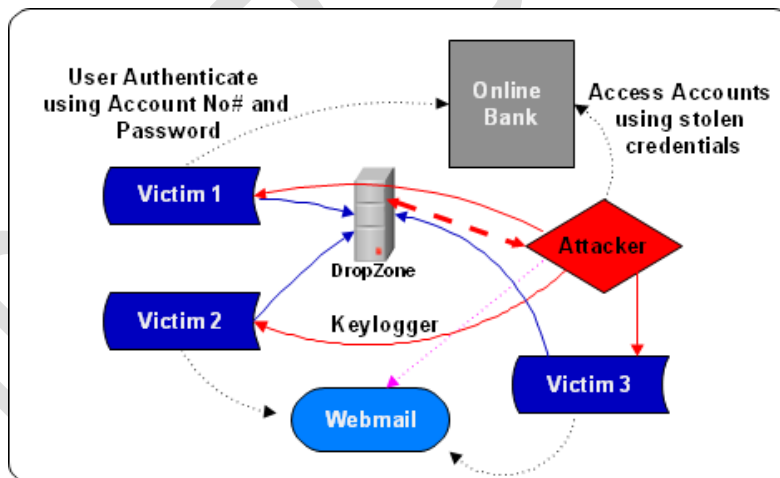
Performing the automated analysis on suspicious malware can give an equivalent amount of information that can be obtained through manual process. However, due to the fast processing and interpretation of these automated engines, most individual organizations prefer to use third-party malware detection solutions. These solutions carry out the dynamic behavioral and advanced heuristic analysis in a controlled virtualized environment. For example, ThreatExpert, Anubis, CWSandbox and Norman Sandbox. Typical steps involved in this approach are:

1. Identify the suspected malware executable file.
2. Upload and submit the file to any of these services mentioned previously.
3. Wait for the results until the analysis is completed.
4. Document your findings on the basis of results provided.

It should be noted that these services are freely available but some of them also provide commercial solutions for high-end corporate users to perform parallel analysis independently.

## 3. Case Study of Banking Malwares

By studying and analyzing the two well-known banking trojans reported during 2008-2009 can help understand the detection approaches described previously.



**Figure 1: Banking Trojan Architecture**

\*DropZone - An exchange point for keyloggers to store victim credentials

### A. Limbo or Nethell

#### 1. Manual Analysis

- ✓ A malware sample when executed results into creation of BHO (Browser Helper Object). This was verified under the following registry key: HKLM\SOFTWARE\Windows\CurrentVersion\Explorer\Browser Helper Objects



- ✓ On final inspection, it has also revealed that the stolen credentials are stored on the server.

```
https://onlineservices.wachovia.com/auth/AuthService
Referer:
Keys: www.wachovia.com julXXX mckinXXX minimXXX july2XXX
Data:
action=uidLogin
msie windows nt 5.1%3B .net clr 1.0.3705%3B [...]
requestTimestamp=1213758461593
userid=julXXXmckinXXX
password=july2XXX
```

### 2. Automated Analysis

The automated behavioral analysis has been published at Anubis online malware detection service. This can be accessed through the following URL:

<http://tinyurl.com/zeusreport>

## 4. Challenges for Anti-Malware Solutions

Increase in the productivity of malwares has also increased the sophistication of analyzing the malware code. Today, many hackers use advanced evasion techniques to bypass the pattern matching technology. These techniques may involve polymorphic and metamorphic nature of the code. In general terms, polymorphic malware is one which changes its appearance every time it executes or propagates itself. However, due to certain limitations in decrypted code which remains same on every execution, it is possible to detect using memory based signature. Similarly, metamorphic malware re-code itself every time it executes or propagates which makes it more challenging as compared to those of polymorphic malwares. The basic techniques applied under metamorphic code involve addition of NOP instructions, permuting use registers, function reordering, data structure and program flow modification. Detection of such malware is challenging and can be analyzed under strict semantic identification and behavioral analysis.

## 5. Digital Underground Economy

Several security researchers have recently studied and published their reports to highlight the key drivers of the underground economy. This includes the major portion of data theft incidents via malwares. According to CSI 2008 report, there is a vertical increment in data theft incidents occurred during 2007-2008. The main entities causing these financial losses include viruses, worms, bots, trojans, insider abuse, laptop theft and insecure applications. Similarly another source of information, Arbor Infrastructure Security Report 2008, also published the information on severity of botnets and their increasing vector of adoption among many underground hackers to perform multi-activities. These activities typically involve denial of service attacks, spam and phishing threats. It also worth noticing, that such services and information is always being shared among cyber criminals in forums, newsgroups or IRC channels. Selling stolen bank credentials,

providing DDoS attack service, trading credit card details or other financial records and providing proxy services are among the most common traded at an underground market.



Services	Prices
Bank Accounts	\$10-\$1000
Credit Cards	\$1-\$20
Paypal Account	\$50-\$1000
Email Access	\$4-\$30
Proxies	\$5-\$40

**Table 2: Overview of Underground Marketplace**

## Corporate Protection Scheme

To respond the active and the emerging threats, there is always a need for proactive security measures to provide edge protection within commercial and non-commercial organizations. This can be accomplished by following the steps below:

- ✓ Deploy a multi-layered protection on your network using firewalls, anti-viruses, and intrusion detection and prevention systems. Make sure that the anti-virus and IDS/IPS signatures are up to date and those of firewall rules should comply with the organization security policy.
- ✓ Conform to PCI-DSS or other similar industry compliance to ensure the confidentiality, integrity and availability of information on your network.
- ✓ Employ the strict network monitoring policy and prohibit any illegal inbound or outbound access to the organization resource. For instance, a simple snort rule can be used to detect social security number (SSN) on-fly over the network. This will ensure that no such information will be transferred in clear text or even rejected in certain cases.
- ✓ If your organization is acting as a financial entity, such as bank, it is vital to devise online security using advanced solutions such as Mobile TAN (mTAN) or at least two-factor authentication.
- ✓ Increase the security awareness among member staff and customers by providing security awareness training periodically.