Reverse Engineering

7. Malware

Ing. Martin Jirkal
Ing. Tomáš Zahradnický, EUR ING, Ph.D.

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra počítačových systémů

Creation of this material was supported from the development project RPP 2015
Aktualizace a tvorba bezpečnostních předmětů a Shelltest.

Version 1.0

Friday 1st January, 2016

Table of Contents

Motivation

- Reverse engineering is vastly used by anti-virus companies for malware analysis.
- Computer forensic investigators often analyze malware to estimate damage inflicted.
- Lately banks have demanded security experts that are able to analyze malware.
- Malware a is good source of knowledge and newest trends.
- It is good to know what happened when anti-virus detects malware and if more steps are needed.

Introduction to Malware

Malware

Malware (Malicious Software) is any software intentionally created to harm the computer user in any way.

- Malware creators use most modern and most obscure methods to protect their code from analysis.
- Most advanced malware creators work similarly to professional software companies with subscription/support plans for their users, modular design of their product, and frequent updates to malware core.
- Historically first malware samples were created as experiments and jokes. Today it evolved into rich industry full of variability.

Malware Classification

- Malware categorize into 3 basic groups. Viruses, Worms, and Trojans.

Virus

A virus infects other computer binaries and converts them into itself without compromising their original functionality.

Worm

A worm copies itself in such a way that it will get transferred and likely executed in different than the originating system.

Trojan

A trojan is malware that is neither Virus nor Worm.

Malware Sub-Classication I

- Malware can be sub-categorized into many variants according on the functionality.

Backdoor

A backdoor malware reacts on commands received from the attacker.

Adware

Adware manipulates existing ads or adds new advertising to the computer/browser/etc. without user approval.

Spy

A spy is malware that steals user/computer sensitive information. Most commonly passwords and bank account information.

Malware Sub-Classication II

Downloader

A downloader is a generic type of malware that is smuggled into a computer in order to download other malware from the Internet.

Exploit

An exploit is a type of malware binary abusing vulnerability of a system/application to elevate privileges, execute code, or deploy malware, usually a downloader.

Rootkit

A rootkit is a driver with purpose is to hide existence of malware.

Bootkit

A bootkit infects the Master Boot Record or Volume Boot Record. Usually in order to load a rootkit.

Malware Sub-Classication III

Coinminer

A coinminer is coinmining software that is operating without user knowledge or approval.

Ransomware

A ransomware is malware that restricts access to computer functionality and demands a ransom for enabling this functionality back.

BadJoke

A special kind of malware that can be meant as a joke but is unremovable without a computer specialist.

- There are many more classes of malware. These are the most common and agreed upon by all anti-virus vendors.

Potentially Unwanted Application

Potentially Unwanted Application

A Potentially Unwanted Application (PUA) or Potentially Unwanted Program (PUP) is an application that is legitimate but can be distributed without user approval or with shady methods. Alternatively it can be software that is often abused by malware.

- Every anti-virus vendor has slightly different rules for PUA classification. As PUA can be legitimate, it should not be handled as normal malware.
- Bundled software that is installed without user approval is PUA.
- Command line coinminers are often marked as PUA as they are distributed with malware.

Virus

- Key Features
  - Hide itself in a working binary → hard to detect (?)
  - Opens and writes into a lot of different files → easy to detect by actions.
- The most typical infecting algorithm
  1. Find an executable binary
  2. Open the last section of the binary and expand it or find enough free space in between sections.
  3. Insert virus code into that space
  4. Change the entry point or change the program so it will eventually jump into virus code.

Worm I

- A typical worm infection:
    - Copy itself onto removable media
    - Copy itself into a shared folder of a P2P sharing application
    - Send itself through applications (Mail, Facebook, Skype, . . .)
    - Using a security flaw
    - Copy itself into a DVD burning queue

```
CSIDL_CDBURN_AREA = C:\Documents and
Settings\Administrator\Local Settings\Application
Data\Microsoft\CD Burning
```

Worm II

- The most typical infecting algorithm through USB:
  1. Search for all removable media.
  2. Copy and hide itself on the media (`SetFileAttributes`).
  3. Create `autorun.inf` in the root directory of the inserted media to be executed when the USB media is inserted into the computer.

autorun.inf

```
[AutoRun]
open=hiddenMalware.exe
shellexecute=hiddenMalware.exe
shell\Auto\command=hiddenMalware.exe
```

Encrypted Malware

When malware is detected by an anti-virus there its author has 2 options:

1. **Create a new malware version that is different enough**
   - + It can take longer before first detection.
   - – Creating new malware is a hard and long process.
   - – New malware must be programmed uniquely to prevent heuristic detection. There is only a finite number of ways how to create desired functionality. Even a single reused module can ease malware detection.
2. **Create custom run-time decryption**
   - + A fast development process.
   - + Can be effectively modified and multiple undetected versions can be released per day.
   - + Can be made to evade heuristic detection.
   - – A good anti-virus can catch encrypted files by emulation or in memory during run-time.

Run-Time Decryption

1. Allocate a memory block for an encrypted file (`VirtualAlloc`).
2. Give execute privileges to the allocated block (`VirtualProtect`).
3. Decrypt the file into that block.
4. Process relocations, if needed.
5. Fill the IAT by loading libraries (`LoadLibrary`, `GetProcAddress`).
6. Jump/Call the entry point.

Injection

- Injecting malware has taken malware one step further.
- Instead of decrypting malware binary into allocated memory, it is copied into a common windows process like `svchost.exe` or `explorer.exe`.
- Such injected file is well hidden, even from professionals.
- More about process injection in Lecture 6.

Monetization Techniques

- In the past malware was created to make fun of someone or to destroy someone's work. This is good for nothing (unless we count for an industrial sabotage). Today, the first and the most important priority of every malware is to make money for his creator.
- Steal computer power
  - Botnet — DDoS, URL clickers, spambots
  - Coinminers
- Steal information
  - Passwords — Bank accounts, services, emails, ...
  - Personal information — name, date of birth, email, phone number, ...
- Demand a ransom
  - Restore access if money is paid.
  - Your data will be published/destroyed unless ...
- Advertising
  - Advertising in a computer is added or changed to benefit the attacker.
  - Ads on a computer are clicked without user's knowledge.
  - A fake tool demanding upgrade to a PRO/FULL version.

Persistence I

- The second priority of almost every malware is to persist in the computer after restart or even after a disk wipe.

Start up Registers

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run]
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce]
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices]
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce]
[HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\Winlogon\Userinit]

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run]
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce]
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices]
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce]
[HKEY_CURRENT_USER\Software\Microsoft\WindowsNT\CurrentVersion\Windows]
```

Persistence II

Start up Directory

```
C:\Documents and Settings\Martin.Jirkal\StartMenu\Programs\Startup --- Windows XP
C:\Users\Martin.Jirkal\AppData\Roaming\Microsoft\Windows\StartMenu\Programs\Startup --- Windows 7
```

Schedule Manager

```
schtasks / Create /tn NotSuspiciousTask /sc ONLOGON /tr C:\temp\malware.exe
```

Service Manager

```
sc create NotSuspiciousService binPath="C:\temp\malware.exe"start=auto
```

Extension Association

```
[HKEY_CLASSES_ROOT\.exe] = MalwareExtensionOpener
[HKEY_CLASSES_ROOT\MalwareExtensionOpener\shell\open\command] = C:\temp\malware.exe
```

- Infected Master Boot Record to be loaded during the computer booting process.

Persistence III

Library search order hijacking. Put dynamic library malware in such a place that it is loaded instead of the target library.

Order with SafeDllSearchMode on (Windows Vista and later default)

1. The directory from which the application loaded.
2. The system directory (`C:\Windows\System32`)
3. The 16-bit system directory. (`C:\Windows\System`)
4. The Windows directory.
5. The current directory.
6. The directories that are listed in the PATH environment variable.

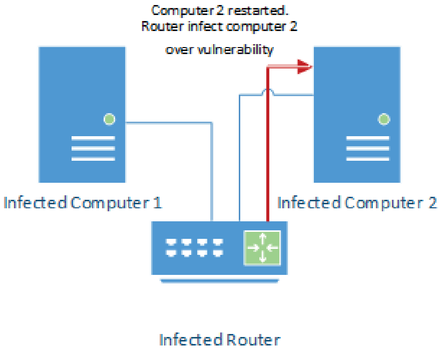
Persistence IV

Order with SafeDllSearchMode off (Windows XP default)

1. The directory from which the application loaded.
2. The current directory.
3. The system directory (`C:\Windows\System32`)
4. The 16-bit system directory. (`C:\Windows\System`)
5. The Windows directory.
6. The directories that are listed in the PATH environment variable.

Persistence — Survive disk format

- Infect firmware of connected devices or BIOS
- Reinfection over LAN (WAN)



Malware Hiding I

- Process injection is the most popular hiding method. `Iexplore.exe` communicating over the Internet is not suspicious and is hard to detect.
- A list of commonly injected processes:
  - svchost.exe
  - explorer.exe
  - csrss.exe
- Name malware as a known file and/or store it into a trusted location (`C:\Windows\system32`).

Malware Hiding II

- The rootkit driver can hide presence at the kernel level. Usual functions:
  - Hide process
  - Hide net communication
  - Hide files
- To hide a file some rootkits save file to unused sectors on the disk.
- Bootkit malware infects the Master Boot Record to launch malware but also to disable advanced computer protection like the anti-virus software.

Malware Hiding III

Rootkit hide process — FU Rootkit

```
case IOCTL_ROOTKIT_HIDEME:
  if (( InputBufferLength < sizeof(DWORD)) || ( InputBuffer == NULL ))
  {
    IoStatus->Status = STATUS_INVALID_BUFFER_SIZE ;
    break;
  }

  find_PID = *(( DWORD *) InputBuffer );
  if( find_PID == 0x00000000 )
  {
    IoStatus->Status = STATUS_INVALID_PARAMETER;
    break;
  }

  eproc = FindProcessEPROC(find_PID);
  if( eproc == 0x00000000 )
  {
    IoStatus->Status = STATUS_INVALID_PARAMETER;
    break;
  }

  plist_active_procs = (LIST_ENTRY*) (eproc + FLINKOFFSET);
  *(( DWORD *) plist_active_procs -> Blink ) = ( DWORD ) plist_active_procs->Flink;
  *(( DWORD *) plist_active_procs -> Flink + 1) = ( DWORD ) plist_active_procs->Blink ;

  break;
```