CHAPTER 5: MALICIOUS SOFTWARES

Information Security
Nguyễn Đăng Quang
Fall 2021

GOALS

- Identify types of malicious software
- Identify viruses vs worms
- Malware Detection and Prevention approaches
- Understand the features of bots & CC mechanism.
- Understand the features of APTs
- Practice basic static & dynamic malware analysis

MALICIOUS SOFTWARE

Programs exploiting system vulnerabilities

Known as malicious software or malware

- Program fragments that need a host program
  - e.g. viruses, and backdoors, browser plug-ins, extensions, scripts
- Independent self-contained programs
  - e.g. worms, bots, APTs

Sophisticated threat to computer systems

MALWARE TERMINOLOGY

Virus: *attaches itself to a program*

Worm: *propagates copies of itself to other computers*

Logic bomb: *"explodes" when a condition occurs*

Trojan horse: *fakes/contains additional functionality*

Backdoor (trapdoor): *allows unauthorized access to functionality*

Mobile code: *moves unchanged to heterogeneous platforms*

Auto-rooter Kit (virus generator): *malicious code (virus) generators*

Spammer and flooder programs: *large volume of unwanted "pkts"*

Keyloggers: *capture keystrokes*

Rootkit: *sophisticated hacker tools to gain root-level access*

Zombie: *software on infected computers that launch attack on others (aka bot)*

VIRUSES

Piece of software that infects programs

- Modifying them to include a copy of the virus to make it executes secretly when host program is run.

Specific to operating system and hardware

- taking advantage of their details and weaknesses

A typical virus goes through phases of:

- dormant: *idle*
- propagation: *copies itself to other programs*
- triggering: *activated to perform functions*
- execution: *the function is performed*

VIRUS STRUCTURE

Components:

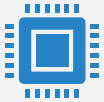
infection mechanism enables replication
trigger: event that makes payload activate
payload: what it does, malicious or benign

Prepended/postpended/embedded

When infected program invoked, executes virus code then original program code

Can block initial infection (difficult) or propagation (with access controls)

CATEGORIES



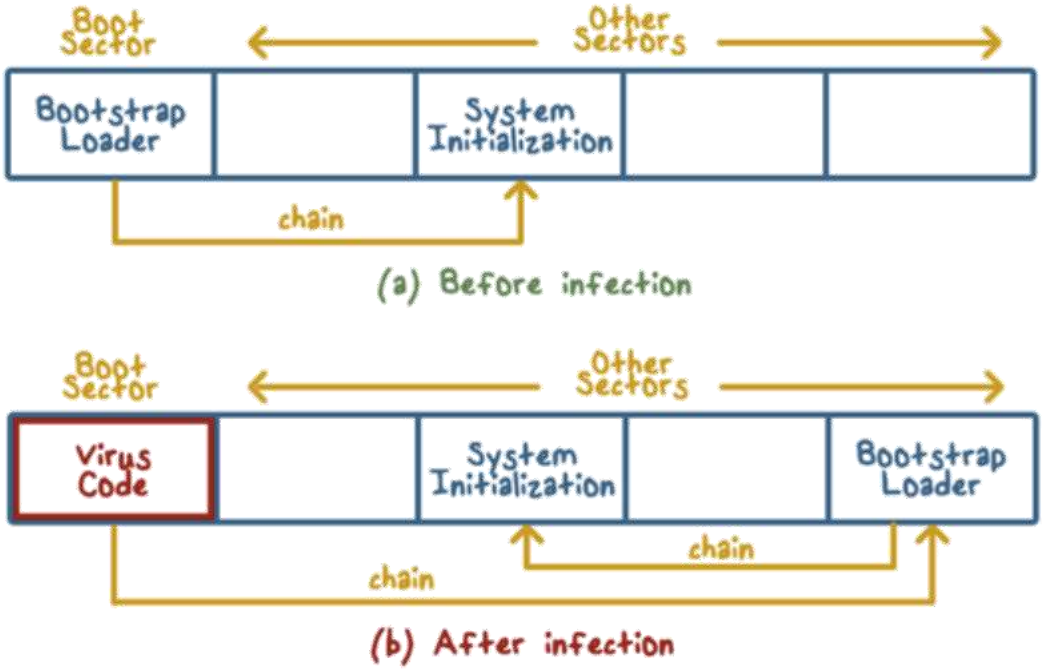
Memory-resident: infect running program

Macro virus: embedded in documents, run/spread when opened

Boot sector virus: run/spread when booted

Polymorphic virus: encrypted part of the virus using a randomly generated key.

BOOT SECTOR VIRUS



MACRO AND SCRIPTING VIRUSES

Became very common in mid-1990s since

- platform independent
- infect documents
- easily spread

Exploit macro capability of Office apps

- executable program embedded in office doc
- often a form of Basic

More recent releases include protection

Recognized by many anti-virus programs

COMMON TECHNIQUE FOR SPREADING

First, a macro that contains virus is created then attached to a Word Document.

When the infected document is opened, the macro (virus) executes.

The virus performs malicious activities. It can copy itself to the global macro file so that whenever the user opens/creates a new document, the global macro will be copied into the document, and that's another way that the macro virus can spread.

ROOTKIT

Memory resident virus.

It typically modifies some of the code and even data structures of the OS in order to perform malicious activities.

Ex: Hide the malware file when listing directory with ls command or modify output of ps command to hide the running of malware.

WORMS

Independent malicious program that does not require host programs.

Use network connections to spread from one system to the other.

A major advance malware since 1990s that develops together with the internet.

Later evolved into botnets in around 2005.

MORRIS WORM

One of best known worms

Released by Robert Morris in 1988

- Affected 6,000 computers; cost $10-$100 M

Various attacks on UNIX systems

- cracking password file to use login/password to logon to other systems
- exploiting a bug in the finger protocol
- exploiting a bug in sendmail

If succeed have remote shell access

- sent bootstrap program to copy worm over

MORE RECENT WORM ATTACKS

Code Red

- July 2001 exploiting MS IIS bug
- probes random IP address, does DDoS attack
- consumes significant net capacity when active
- 360,000 servers in 14 hours

Code Red II variant includes backdoor: hacker controls the worm

SQL Slammer *(exploited buffer-overflow vulnerability)*

- early 2003, attacks MS SQL Server
- compact and very rapid spread

Mydoom *(100 M infected messages in 36 hours)*

- mass-mailing e-mail worm that appeared in 2004
- installed remote access backdoor in infected systems

STATE OF WORM TECHNOLOGY

Multiplatform: not limited to Windows

Multi-exploit: Web servers, emails, file sharing …

Ultrafast spreading: do a scan to find vulnerable hosts

Polymorphic: each copy has a new code

Metamorphic: change appearance/behavior

Transport vehicles (e.g., for DDoS)

Zero-day exploit of unknown vulnerability (to achieve max surprise/distribution)

MALWARE PREVENTION & DETECTION APPROACHES

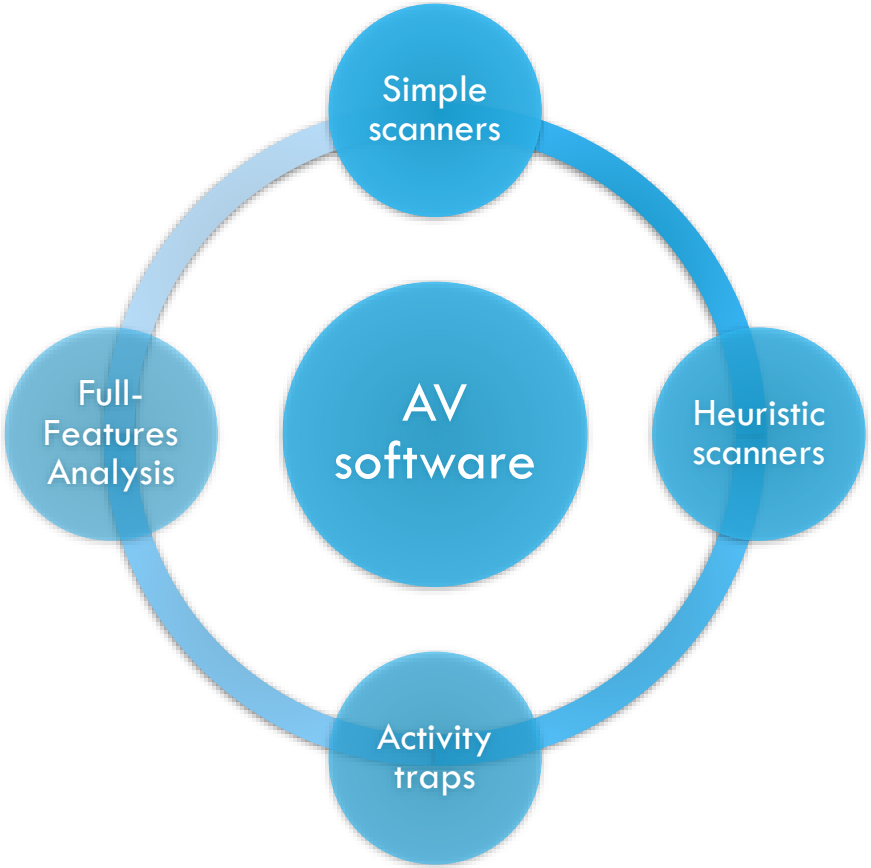
APPROACHES

Prevention: Limit contact to the untrusted outside world.

Detection: use a monitor to watch out signs of malware infection → main countermeasure .

Removal: once malware infection is detected, it must be removed and apply patches to the system.

GENERATIONS OF ANTIVIRUS SOFTWARE



MODERN MALWARE

MODERN MALWARE

Modern malware have advanced capabilities:

- robust command control infrastructures,
- the abilities to evade analysis and detection.

This section covers:

- Botnets & APTs,
- Basic malware analysis and detection techniques

MALWARE IN THE PAST

The majority (early 2000) were designed for experimenting or demonstrating some capabilities:

- Defacing web pages, large-scale DoS,
- How fast/large a worm/malware can spread → mostly for fun & fame.

MODERN MALWARE

Take control of resources for profits and political gains.

Tend to be technically sophisticated:

- Utilized popular peer-to-peer protocols to set up communications,
- Utilized cloud computing to support malicious activities,
- Latest crypto algorithms to perform authentication & encryption to protect their own communications from analysis.

BOTNET

The most prevalent modern-day malware.

Bot is a zombie.

Botnet is a network of bots controlled by an attacker to perform coordinated malicious activities.

With a network of bots, the aggregated combination of power can be very large.

Are now the key platform for most of the internet-based attacks and frauds.

IDENTIFY BOTS & DEFINITIONS

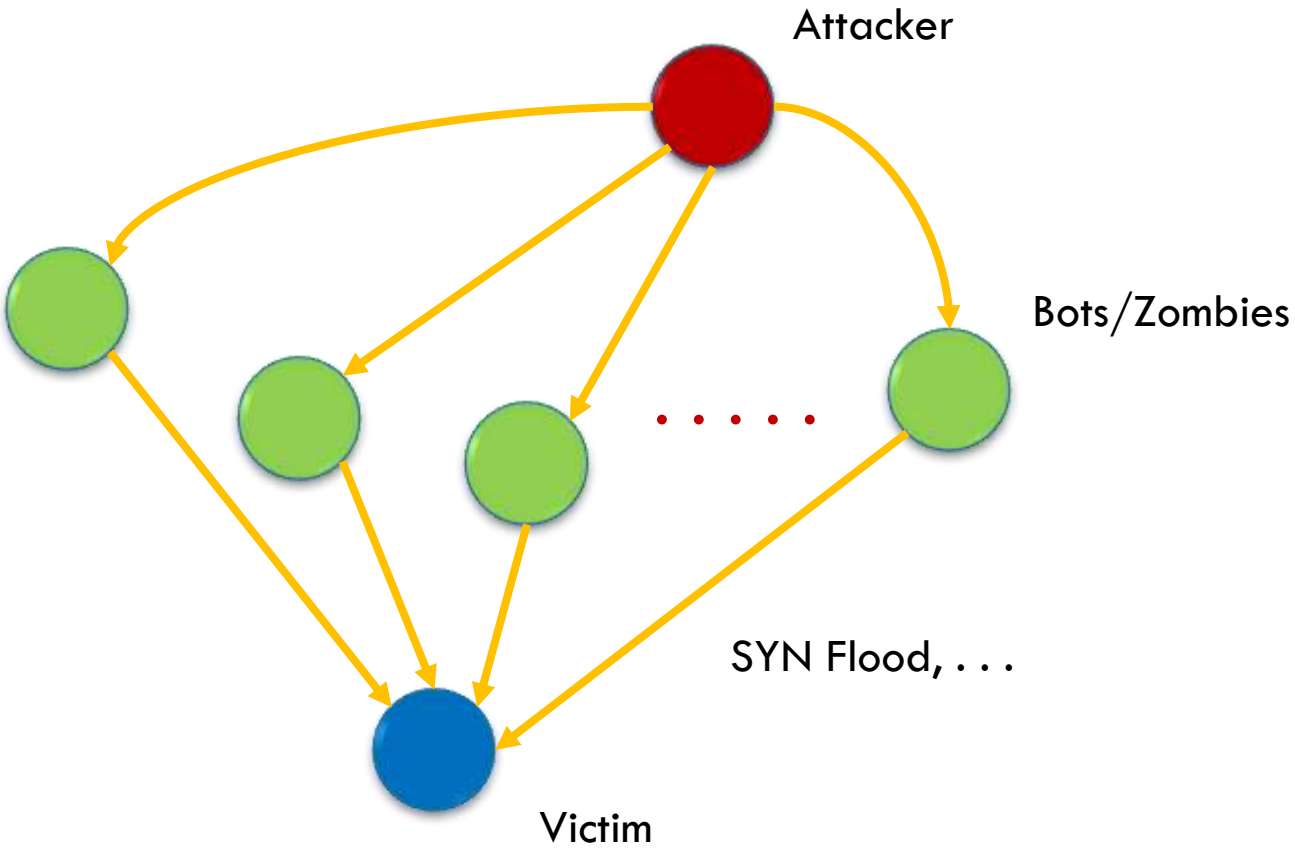
a) Spamming

b) Click fraud

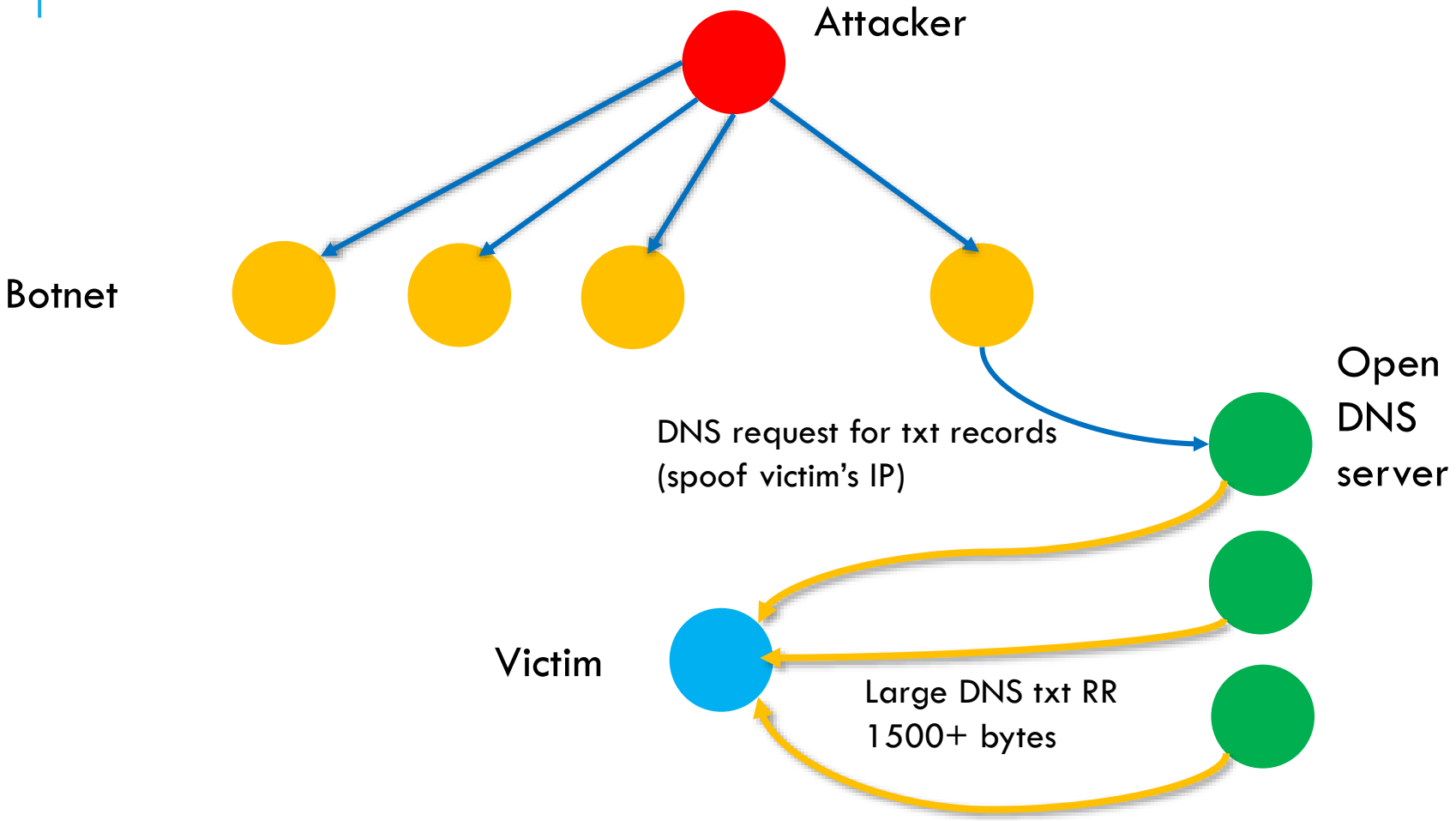
c) Phishing

1. Used by botmasters to fraudulently increase revenue from advertisers.
2. Used to gather valuable financial information.
3. Infected machines send out emails

DDOS USING BOTNETS



AMPLIFIED DISTRIBUTED REFLECTIVE ATTACKS



BOTNET COMMAND AND CONTROL

Botnet is a network of compromised computers that the Botmaster uses for malicious purposes.

The Botmaster needs to communicate with the bots to command and control → C&C for short.

Without C&C, a botnet is not a network, but just a collection of isolated infected machines

BOTNET C&C PROBLEM

Creating, Spreading is easy.

How can the botmaster know which computers have been affected → how to contact and use them?

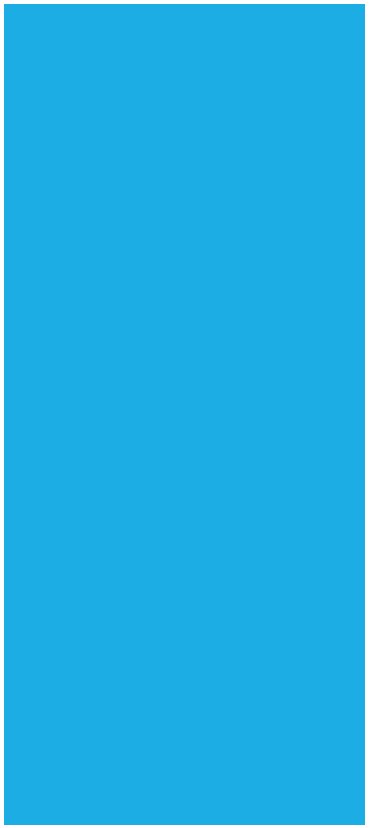
- Hard-coded email?
- Hard-coded IP?

BOTNET C&C – NAIVE APPROACH

Have the victims **contact** the botmaster…

Problems:

- Botmaster's address must be hardcoded in botcode → not stealthy
- The address will be easily recognized and be banned by the network administrator



BOTNET C&C DESIGN CONSIDERATIONS

Efficient and reliable:
Able to reach to a set of bots within a time limit

Stealthy:
Hard to detect (blended with regular traffic)

Resilient:
Hard to disable or block.

DNS BASED BOTNET C&C

Many botnets use DNS for C&C.

DNS is always allowed in a network.

The Botmaster releases malware where the domain name of the C&C servers are coded.

Dynamic DNS is preferred by botmaster because they allow the frequent changes of the mapping between domain name and IP address.

HOW TO DETECT & STOP BOTNET CC

QUIZ

Which of the following C&C schemes provide:

- Efficient/reliable communications
- Stealth communication (hard to detect)
- Resilient communication (hard to disrupt)

❑A Gmail account is used for C&C, email address is hardcoded in botcode.

❑P2P protocol is used for C&C, query string is hardcoded in botcode.

❑A news website has been set up for C&C, commands can be parsed from news articles. The website and the parsing logic have to be hardcoded in the botcode.

ADVANCED PERSISTENT THREAT (APT)

APT tends to target specific organizations, while botnets tend to have bots all over the internet.

Advanced: special malware (a special version of a common malware),

Persistent: Long-term presence, multi-step, “low and slow”,

Threat: Target at high-value organization and information

APT LIFECYCLE

MALWARE ANALYSIS

GOAL OF MALWARE ANALYSIS

Produces information about a malware → can be used for detection and response to malware.

Steps:

1. Identify suspect binary files that might contains malicious code for full analysis,
2. Analyze suspect files to develop host-based signatures and monitor network traffic to develop network signatures,
3. Figure out exactly how the malware works

MALWARE ANALYSIS TECHNIQUES

Basic static:
Examining the executable file without viewing the actual instructions to confirm malicious/not malicious

Basic dynamic:
Running the malware and observing its behavior on the system in order to remove the infection

Advanced static:
Loading the executable into a disassembler and looking at the program instructions in order to discover what the program does

Advanced dynamic :
debugger to examine the internal state of a running malicious executable

BASIC STATIC TECHNIQUES

1

Using antivirus tools to confirm maliciousness.

2

Using hashes to identify malware.

3

Gleaning information from a file's string, functions, and headers.

ANTIVIRUS SCANNING

Antivirus tools rely mainly on a database of identifiable pieces of known suspicious code (*file signatures*), as well as behavioral and pattern-matching analysis (*heuristics*) to identify suspect files.

Rare malware often goes undetected by antivirus software because it's simply not in the database.

Because the various antivirus programs use different signatures and heuristics, it's useful to run several different antivirus programs against the same piece of suspected malware.

VirusTotal (*http://www.virustotal.com/*) generates a report that provides the total number of engines that marked the file as malicious, the malware name, and, if available, additional information about the malware.

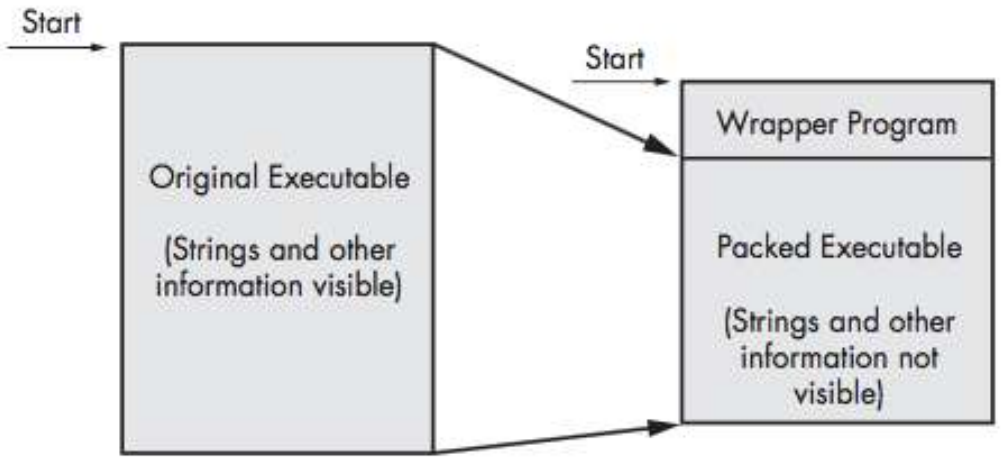
HASING: A FINGERPRINT FOR MALWARE

Hashing is a common method used to uniquely identify malware.

The malicious software is run through a hashing program that produces a unique *hash* that identifies that malware (a sort of fingerprint).

The MD5 hash function is the one most commonly used for malware analysis, though the Secure Hash Algorithm 1 (SHA-1) is also popular

Packing refers to the process of compressing and encrypting part of an executable program. ➔ that part becomes data, instead of an instruction set.
https://bit.ly/2iFbcTH



PACKED AND OBFUSCATED MALWARE

TOOLS

http://www.virustotal.com

PEViewer

Upx Unpacker

Dependency Walker

PE Explorer

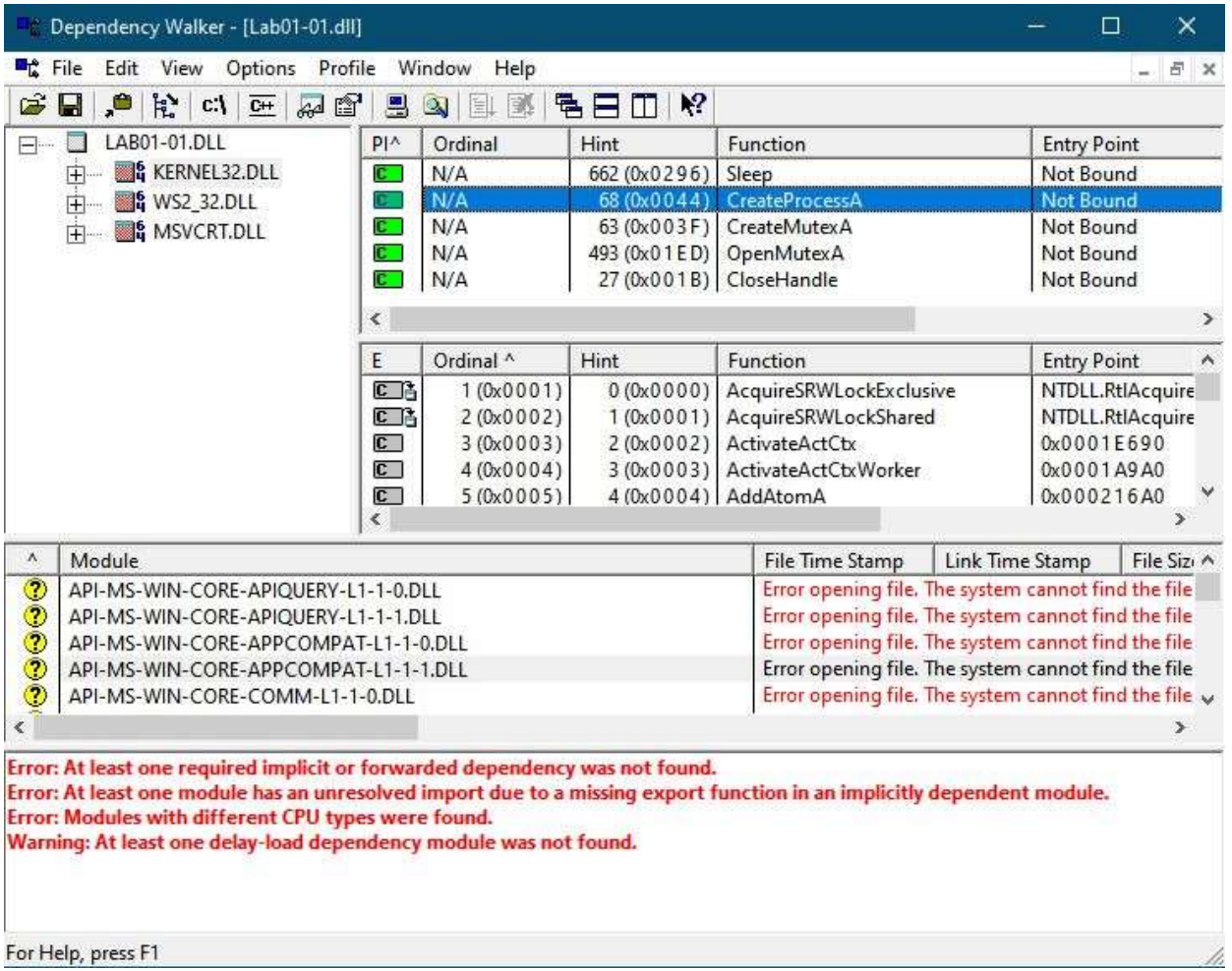
PORTABLE EXECUTABLE (PE) FILE FORMAT

The PE file format is a data structure that contains the information necessary for the Windows OS loader to manage the wrapped executable code.

Nearly every file with executable code that is loaded by Windows is in the PE file format.

PE files begin with a header that includes information about the code, the type of application, required library functions, and space requirements. The information in the PE header is of great value to the malware analyst

DEPENDENCY WALKER



LINKED LIBRARIES AND FUNCTIONS

Imports are functions used by one program that are actually stored in a different program, such as library codes that contain functionality common to many programs.

Libraries code can be connected to the main executable by static or dynamic *linking*.

Knowing how the library code is linked is critical to understanding of the malware because the information we can find in the PE file header depends on how the library code has been linked.

Dependency walker (http://dependencywalker.com) can be used to list dynamically linked functions in an executable

PROGRAM'S DLLS

DLL	Description
Kernel32.dll	This is a very common DLL that contains core functionality, such as access and manipulation of memory, files, and hardware.
Advapi32.dll	This DLL provides access to advanced core Windows components such as the Service Manager and Registry.
User32.dll	This DLL contains all the user-interface components, such as buttons, scroll bars, and components for controlling and responding to user actions.
Gdi32.dll	This DLL contains functions for displaying and manipulating graphics.
Ntdll.dll	This DLL is the interface to the Windows kernel. Executables generally do not import this file directly, although it is always imported indirectly by *Kernel32.dll*. If an executable imports this file, it means that the author intended to use functionality not normally available to Windows programs. Some tasks, such as hiding functionality or manipulating processes, will use this interface.
WSock32.dll and *Ws2_32.dll*	These are networking DLLs. A program that accesses either of these most likely connects to a network or performs network-related tasks.
Wininet.dll	This DLL contains higher-level networking functions that implement protocols such as FTP, HTTP, and NTP.

PE FILE HEADERS AND SECTIONS

sections	description
.text	code section
.rdata	contains the import and export information, which is the same information available from both Dependency Walker & PEViewer
.data	Program's global data
.rsrc	the resources used by the executable that are not considered part of the executable, such as icons, images, menus, and strings.

BASIC STATIC TECHNIQUE DEMO

BASIC DYNAMIC ANALYSIS

Dynamic analysis is any examination performed after executing malware.

Dynamic analysis techniques are the second step in the malware analysis process.

Dynamic analysis is typically performed after basic static analysis has reached a dead end.

It can involve monitoring malware as it runs or examining the system after the malware has executed.

BASIC DYNAMIC ANALYSIS TECHNIQUES

Running malware in VMware.

Monitoring with process monitor (procmon)

Faking a network

- ApateDNS: ApateDNS spoofs DNS responses to a user-specified IP address by listening on UDP port 53 on the local machine.
- Netcat: Network monitoring
- Wireshark: Network sniffing
- INetSim: Linux-based software suite for simulating common Internet services