

Introduction to Malware Reversing



a brief introduction to reversing malware

- *Brandon Lum* -



Brandon Lum



Cyber & Digital Security



Computer Science



Cryptography
@ University of
Wollongong

IT Audit and Pentest
@ KPMG



Contents

- ❖ About Reversing Malware
- ❖ Basic Static Analysis (& Anti-RE)
- ❖ Basic Dynamic Analysis

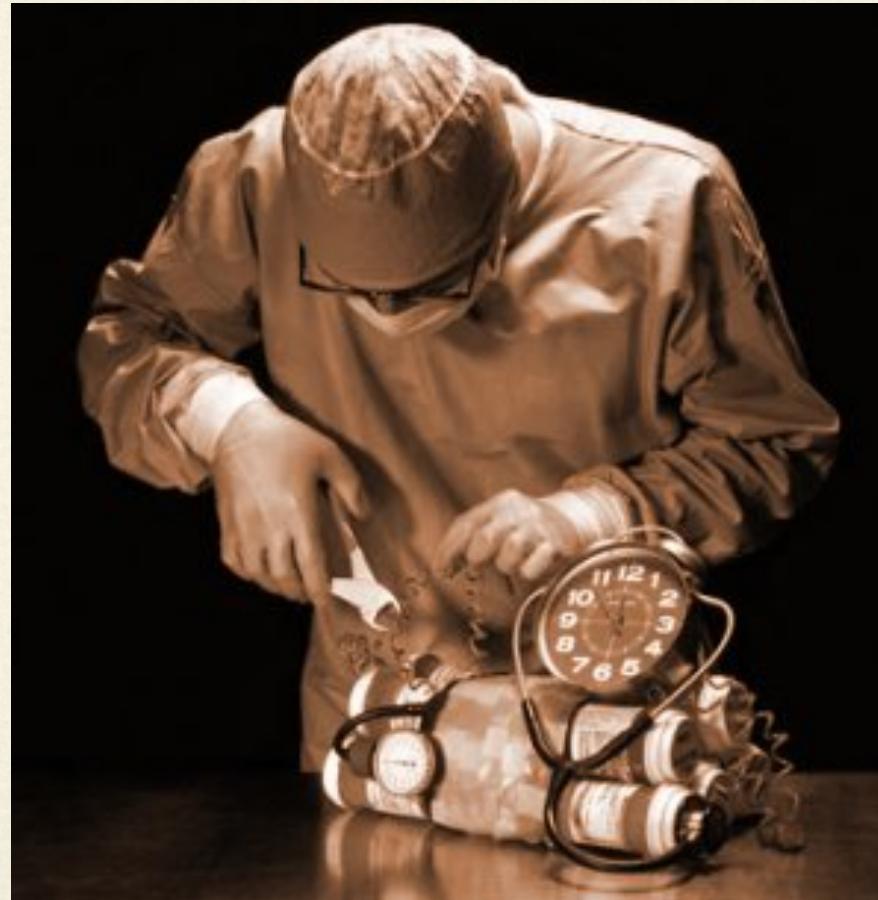
Why RE Malware?

- ❖ Prevention and Containment

- ❖ Research

- ❖ Fun

Reversing Malware



Reversing Malware

- ❖ Set up a Virtual Environment
- ❖ Get the necessary tools ready
- ❖ Snapshot is your best friend

Static Analysis



Basic Static Analysis

- ❖ Finding out information about the Malware by looking at the file itself (**without execution**).

Techniques

- ❖ File Signatures
- ❖ Strings within file
- ❖ Portable Executable (PE) Headers + Resources

File Signatures

File Signatures

- ❖ Leveraging on the analysis of others’
 - ❖ Anti-Viruses have their own analysis of Malware, based on:
 - ❖ Signature
 - ❖ Heuristics

File Signatures



VirusTotal is a free service that **analyzes suspicious files and URLs** and facilitates the quick detection of viruses, worms, trojans, and all kinds of malware.

No file selected

Choose File

Maximum file size: 32MB

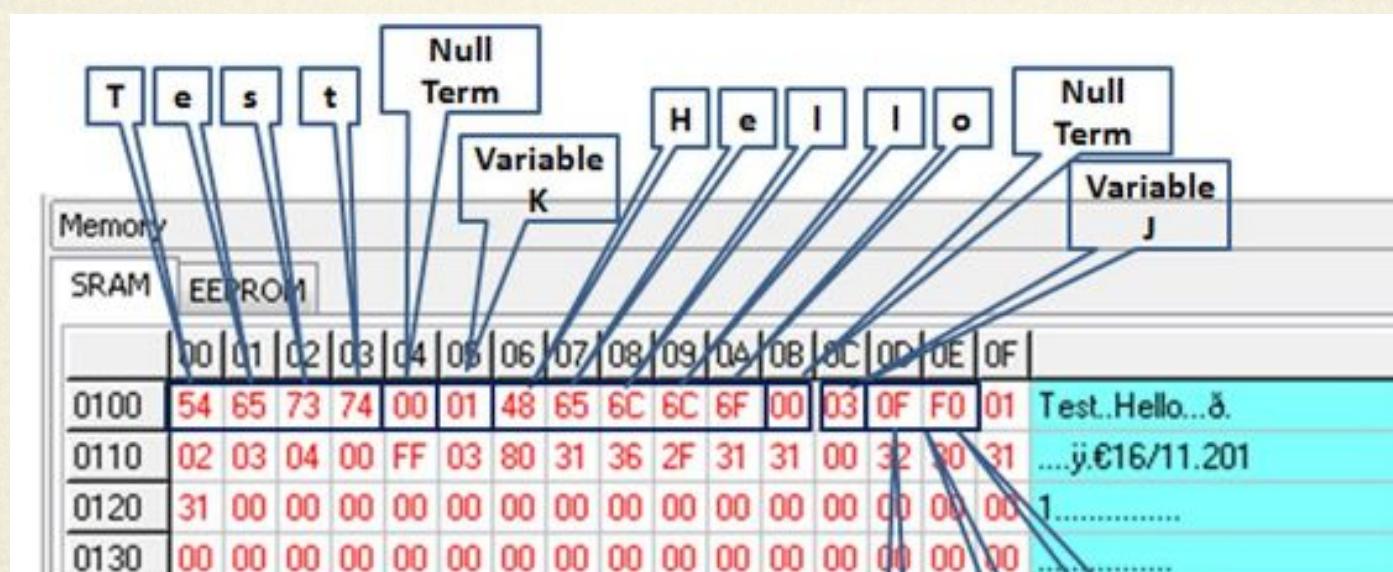
Scan it!

You may prefer to [scan a URL](#) or [search](#) through the VirusTotal dataset

Strings

Strings

- ❖ Strings are identified by a NULL terminating character



```
. _^]3
. hp @@
. SVW
. %8 @@
. %D @@
. %\ @@
. %` @@
. CloseHandle
. UnmapViewOfFile
. IsBadReadPtr
. MapViewOfFile
. CreateFileMappingA
. CreateFileA
. FindClose
. FindNextFileA
. FindFirstFileA
. CopyFileA
. KERNEL32.dll
. malloc
. exit
. MSVCRT.dll
. __p__fmode
. _stricmp
. kerne132.dll
. kernel32.dll
. .exe
. C:\*
. C:\windows\system32\kerne132.dll
. Kernel32.
. Lab01-01.dll
. C:\Windows\System32\Kernel32.dll
. WARNING_THIS_WILL_DESTROY_YOUR_MACHINE
. http://www.practicalmalwareanalysis.com/updater.exe
```

```
. _^]3
. hp @@
. SVW
. %8 @@
. %D @@
. %\ @@
. %` @@
. CloseHandle
. UnmapViewOfFile
. IsBadReadPtr
. MapViewOfFile
. CreateFileMappingA
. CreateFileA
. FindClose
. FindNextFileA
. FindFirstFileA
. CopyFileA
. KERNEL32.dll
. malloc
. exit
. MSVCRT.dll
. __p__fmode
. _stricmp
. kerne132.dll
. kernel32.dll
. .exe
. C:\*
. C:\windows\system32\kerne132.dll
. Kernel32.
. Lab01-01.dll
. C:\Windows\System32\Kernel32.dll
. WARNING_THIS_WILL_DESTROY_YOUR_MACHINE
. http://www.practicalmalwareanalysis.com/updater.exe
```

Strings

- ❖ IP Addresses (N)
- ❖ URLs (N)
- ❖ File Paths (H)
- ❖ Registry Keys (H)
- ❖ Error Messages (which may hint functionality)
- ❖ APIs, Functions, etc.

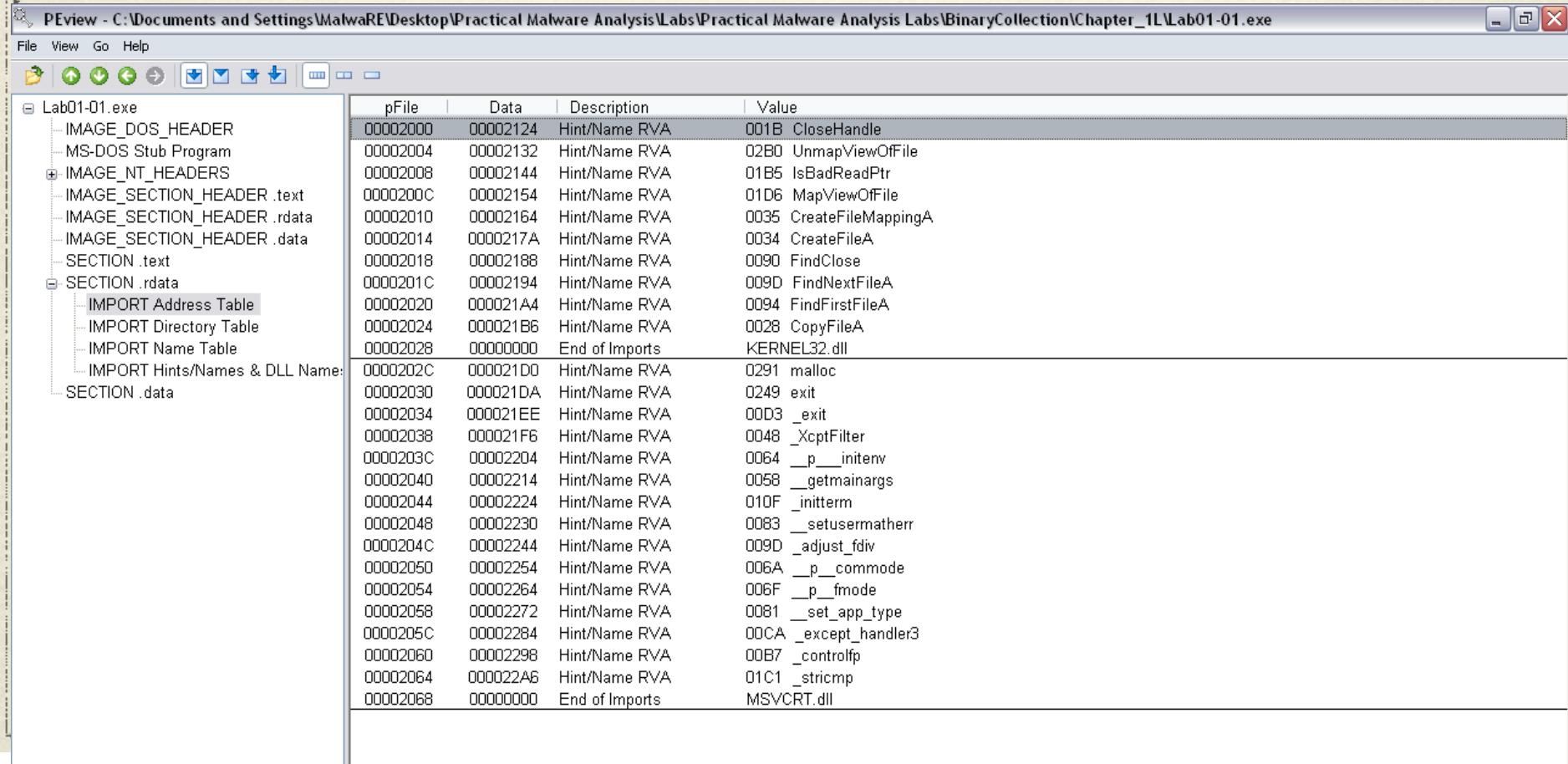
Portable Executables (PEs)

Portable Executables

- ❖ .text - Executable Code
- ❖ .data - Stores global data throughout program
- ❖ .rsrc - Stores resources needed by EXE
- ❖ Other sections:
.rdata, .idata, .edata, .pdata, .reloc

Portable Executable (PE) Headers

❖ Information about APIs and DLLs used



The screenshot shows the PEview interface with the file 'Lab01-01.exe' open. The left pane displays the file's structure, including sections like IMAGE_DOS_HEADER, IMAGE_NT_HEADERS, and various sections (.text, .rdata, .data). The right pane is a table of imports:

pFile	Data	Description	Value
00002000	00002124	Hint/Name RVA	001B CloseHandle
00002004	00002132	Hint/Name RVA	02B0 UnmapViewOfFile
00002008	00002144	Hint/Name RVA	01B5 IsBadReadPtr
0000200C	00002154	Hint/Name RVA	01D6MapViewOfFile
00002010	00002164	Hint/Name RVA	0035 CreateFileMappingA
00002014	0000217A	Hint/Name RVA	0034 CreateFileA
00002018	00002188	Hint/Name RVA	0090 FindClose
0000201C	00002194	Hint/Name RVA	009D FindNextFileA
00002020	000021A4	Hint/Name RVA	0094 FindFirstFileA
00002024	000021B6	Hint/Name RVA	0028 CopyFileA
00002028	00000000	End of Imports	KERNEL32.dll
0000202C	000021D0	Hint/Name RVA	0291 malloc
00002030	000021DA	Hint/Name RVA	0249 exit
00002034	000021EE	Hint/Name RVA	0003 _exit
00002038	000021F6	Hint/Name RVA	0048 _XcptFilter
0000203C	00002204	Hint/Name RVA	0064 __p__initenv
00002040	00002214	Hint/Name RVA	0058 __getmainargs
00002044	00002224	Hint/Name RVA	010F __initterm
00002048	00002230	Hint/Name RVA	0083 __setusermatherr
0000204C	00002244	Hint/Name RVA	009D __adjust_fdiv
00002050	00002254	Hint/Name RVA	006A __p__commode
00002054	00002264	Hint/Name RVA	006F __p__fmode
00002058	00002272	Hint/Name RVA	0081 __set_app_type
0000205C	00002284	Hint/Name RVA	00CA __except_handler3
00002060	00002298	Hint/Name RVA	00B7 __controlfp
00002064	000022A6	Hint/Name RVA	01C1 __strcmp
00002068	00000000	End of Imports	MSVCRT.dll

Other information in PE headers

- ❖ Compile Time
- ❖ Information of code and data sections within the PE

Demo 1:

Static Analysis
Lab 01-01 (Static Analysis)
Lab 01-04 (Resources)
Lab 01-02 (Packing)

Functions to Note

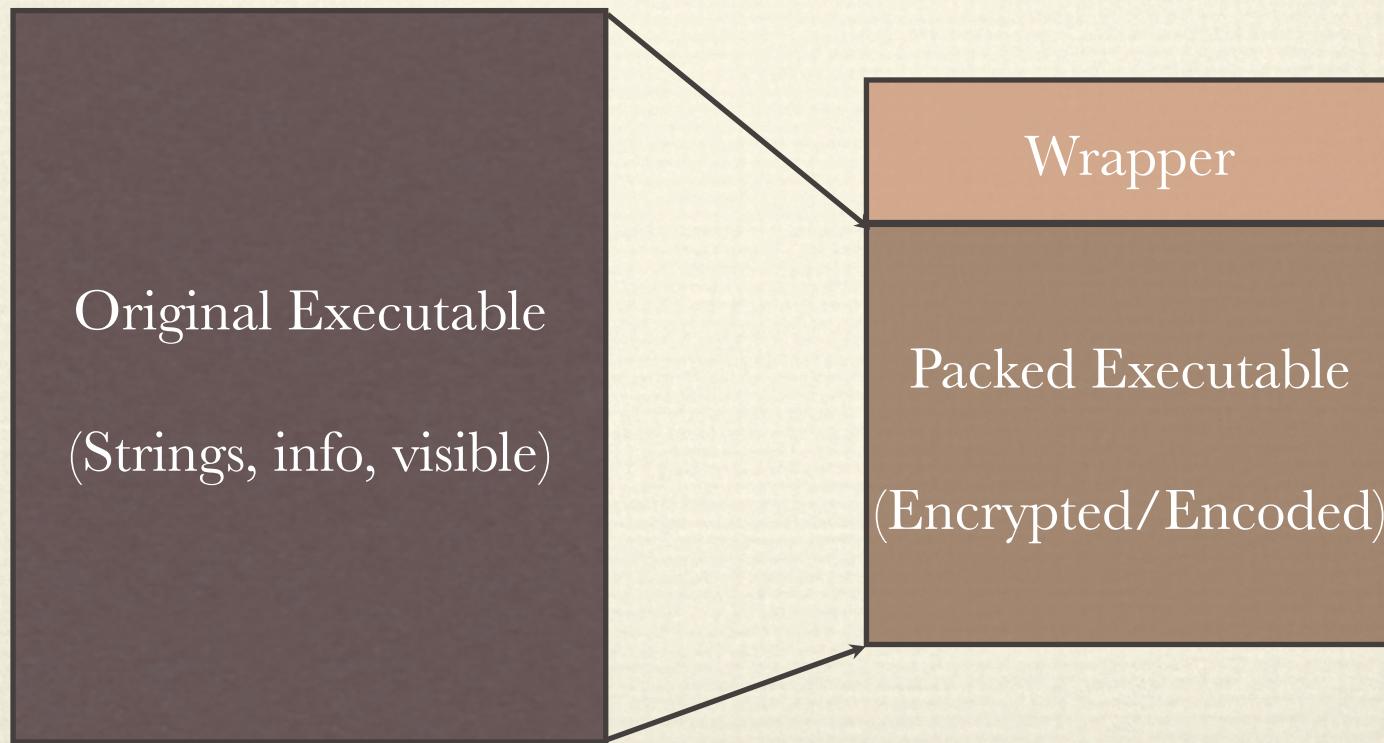
- ❖ Network Communications:
 - ❖ w32_32.dll: gethostbyaddr, connect, close socket, etc.
 - ❖ urlmon.dll: URLDownloadToFile
 - ❖ wininet.dll: InternetOpenURL, InternetConnect, etc.
- ❖ Process and Thread:
 - ❖ VirtualAllocEx, VirtualProtectEx,
ReadProcessMemory, WriteProcessMemory

Resources

- ❖ PEs contain resources
 - ❖ Images
 - ❖ Scripts
 - ❖ Other PEs

Anti-Analysis Techniques

❖ Packing of Executables



Anti-Analysis Techniques

- ❖ Tell-tale signs of Packing
 - ❖ Packers Naming Conventions
 - ❖ Virtual Size Allocation VS Raw Data
 - ❖ Less than normal Imported Functions

Dynamic Analysis



Basic Dynamic Analysis

- ❖ Finding out information about the Malware by through executing the malware

Basic Dynamic Analysis

- ❖ Things to look out for
- ❖ Different ways to capture and view

Malware Behavior

- ❖ Host
 - ❖ Registry
 - ❖ Files
 - ❖ Processes
 - ❖ etc.

Malware Behavior

- ❖ Network
 - ❖ DNS Requests
 - ❖ HTTP/FTP/SSH Requests
 - ❖ etc.

Capture and View

- ❖ Snapshots - Compare Before/After
- ❖ Capturing all events
- ❖ Watch real-time
- ❖ Set up Infrastructure

Demo 2:

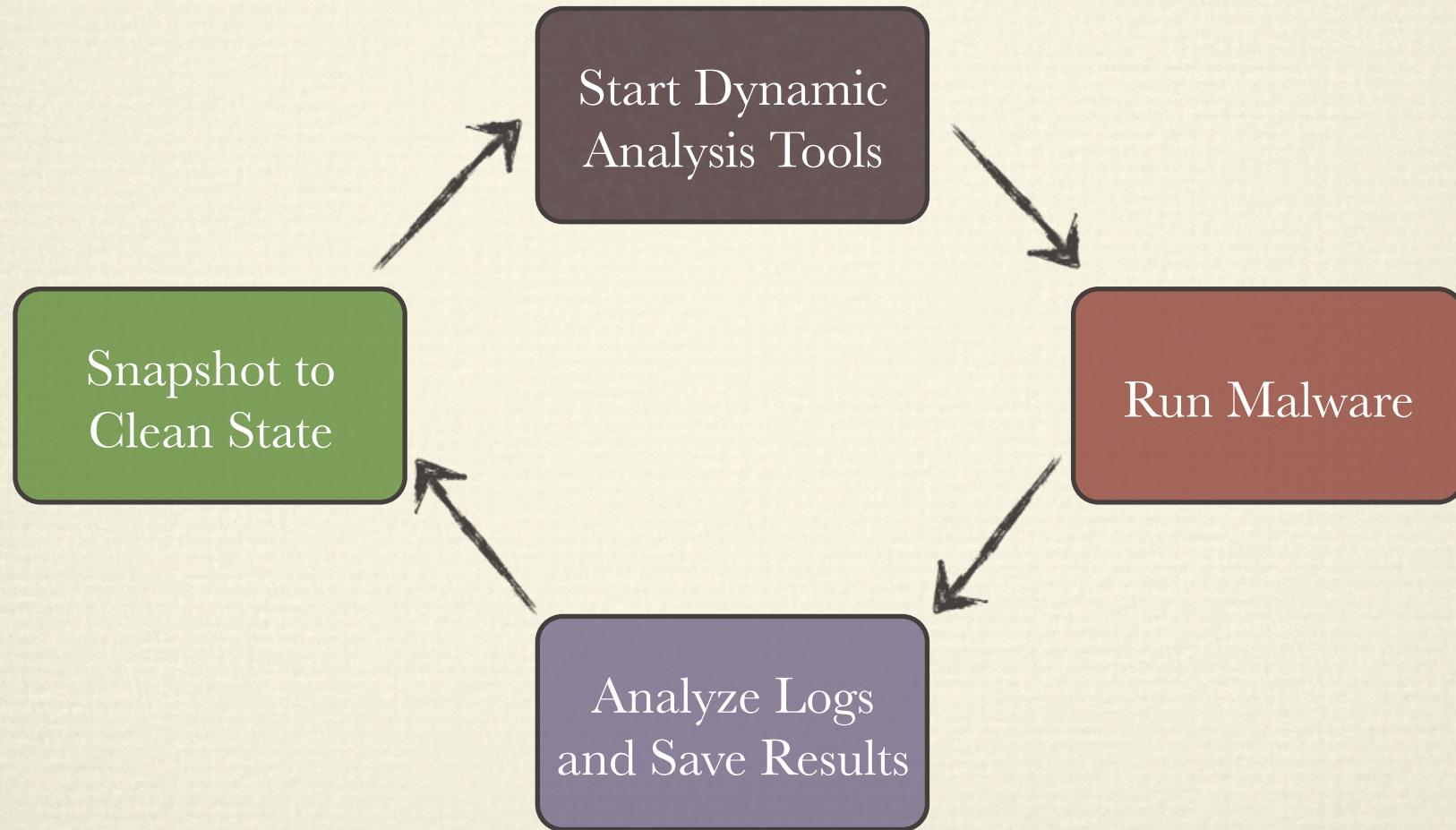
Snapshot & Capture

Demo 3:

Capture & Network

Lab 03-01

Methodology for Dynamic Analysis



Conclusion

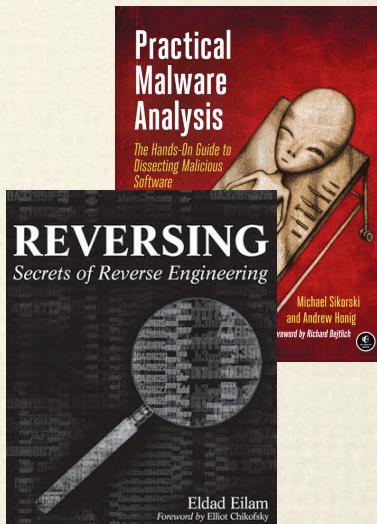
- ❖ Basic Static & Dynamic Analysis
 - ❖ Overview Malware Functionality
 - ❖ Clues to create signatures
- ❖ Next Step: Debug the code!

List of Tools

- ❖ Strings
- ❖ PEView
- ❖ Dependency Walker
- ❖ Resource Hacker
- ❖ Procmon
- ❖ Procexp
- ❖ Regshot
- ❖ Capture
- ❖ Wireshark
- ❖ Netcat/Fakenet
- ❖ FakeDNS/ApateDNS
- ❖ PEID
- ❖ UPX

References

- ❖ Malware RE Workshop by Chong Rong Hwa
- ❖ Practical Malware Analysis
- ❖ Reversing



Thank You



Brandon Lum

LUMJJB_AT_GMAIL_DOT_COM

<http://lumexsec.wordpress.com>

Lumex Security

:::Innovative Security Insights:::

