Malware Analysis
Analysis Prevention Techniques

- Packing
- Obfuscation
- Anti-Patching
- Debugger Detection
- VM Detection
- Polymorphic Malware
Obfuscation

Confuse Disassembly Algorithms

Figure 15-2: A jz instruction followed by a jnz instruction

| 74 03 | jz     | short near ptr loc_4011C4+1 |
| 75 01 | jnz    | short near ptr loc_4011C4+1 |
|       | loc_4011C4: |                           |
|       | E8 58 C3 90 90 | call near ptr 90D0D521h   |
Obfuscation

Figure 15-3: False conditional of xor followed by a jz instruction

Figure 15-5: Multilevel inward-jumping sequence
Obfuscation

Other methods include:
- Using function pointers
- Nonstandard `ret` usage
- Structured Exception Handler misuse
Packers

- Program used to compress and/or encrypt executable.
- Makes static analysis harder.
- Antivirus programs have signatures for common packers.
- UPX, ASPack, Themida, Petite, PELock, WinUPack
Packers

```
047000 dw \x0047000
047000 dd \x02A1FF3h, \x02002500h, \x02004E00h, \x02006CB00h, \x0E00756D8h
047000 dd \x021650E10h, \x0D966C96Ch, \x017B0374h, \x24F98F22h, \x0E59A6CB9h
047000 dd \x029F672603h, \x2873273Eh, \x72D83FB3h, \x43472912h, \x00005h
047000 dd \x0250B2CD7h, \x0DE89F93Bh, \x2D9B0F07h, \x919107A2h, \x0F00FDC7h
047000 dd \x0256D4425h, \x91917064h, \x28389191h, \x91911484h, \x03C9C919h
047000 dd \x0291918898h, \x546C9191h, \x91911168h, \x0A8049191h, \x9191BC00h
047000 dd \x02407C9191h, \x9191B85Ch, \x78B89191h, \x91914C50h, \x0C40C9191h
047000 dd \x029191B434h, \x90189191h, \x91919420h, \x28490191h, \x9191C010h
047000 dd \x022A49191h, \x91910874h, \x0BAC9191h, \x9B76030h, \x58489191h
047000 dd \x02149868A0h, \x1F01E804h, \x0E13B82h, \x303B7202h, \x3D74903h
047000 dd \x020FB3B1FC0h, \x23FBC3FFh, \x4A93E4DE6h, \x2D6FC902h, \x1157FB4h
047000 dd \x025AB94005h, \x0BF09D01h, \x6F7250FFh, \x7463656Ah, \x41E431h
047000 dd \x027B415618h, \x00FFFFFCCH, \x0C4020EFh, \x0C8E524BBh, \x0BD45025Ch
047000 dd \x02393D273Dh, \x3339D2A9h, \x5F775990h, \xD0A0475323h, \x80A97F6h
047000 dd \x0290FFFF1h, \x03A6E9C37h, \x9933A40Fh, \x0B711FC66h, \x0A0000Ch
047000 dd \x025493D360h, \x10D19FB7h, \x53E7719h, \x6D726F46h, \x61010D65h
047000 dd \x02961BDBDh, \x2300421Ah, \x122400FFh, \x0EEBB1D6Fh, \x59293C35h
047000 dd \x02830B7C32h, \x3460304h, \x0F68DFFFFh, \x135E01DDh, \x77000202h
```
Handling Packers

PEiD - A tool for identifying unpackers

PE Explorer

IDA Pro Universal Unpacker Plugin

OllyDbg + OllyDump
Anti-Debugging

PEB.\texttt{BeingDebugged} flag set by debugger

API Calls:
- \texttt{IsDebuggerPresent}
- \texttt{OutputDebugString}
- \texttt{NTQueryInformationProcess}

\begin{tabular}{|l|l|}
\hline
\textbf{mov method} & \textbf{push/pop method} \\
\hline
\texttt{mov eax, dword ptr fs:[30h]} & \texttt{push dword ptr fs:[30h]} \\
\texttt{mov ebx, byte ptr [eax+2]} & \texttt{pop edx} \\
\texttt{test ebx, ebx} & \texttt{cmp byte ptr [edx+2], 1} \\
\texttt{jz NoDebuggerDetected} & \texttt{je DebuggerDetected} \\
\hline
\end{tabular}
Anti-Debugging

Check ProcessHeap and NTGlobal flags

Find evidence of debugger on system:
  Registry entries
  FindWindow API call

Monitor timing of execution
Check for breakpoints
Anti-Virtual Machine

Malware can try to detect whether it is in a VM
Anti-Virtual Machine

VM Fingerprints

- Descriptor Table addresses (IDT, LDT, etc.)
- Running Processes (eg. VMWare Tools)
- Registry entries that include "VMWare"
- Default virtual machine hardware
- Common VM MAC addresses
- VMWare specific I/O port (see next slide)
- Basically, any difference between a VM and a real computer.
By default, VMWare has a special I/O port that it uses for things like copy/paste and file sharing functions.

```asm
// perform fingerprint
mov eax, 'VMXh'   // VMware magic value (0x564D5868)
mov ecx, 0Ah      // special version cmd (0x0a)
mov dx, 'VX'      // special VMWare I/O port (0x5658)

in eax, dx        // special I/O cmd

mov a, ebx         // data
mov b, ecx         // data (eax gets also modified but w
Anti-Virtual Machine

ScoopyNG - The VMware Detection Tool

Windows version v1.0

Test 1: IDT
IDT base: 0x80b95400
Result: Native OS

Test 2: LDT
LDT base: 0xdead0000
Result: Native OS

Test 3: GDT
GDT base: 0x80b95000
Result: Native OS

Test 4: STR
STR base: 0x28000000
Result: Native OS

Test 5: VMware "get version" command
Result: VMware detected
Version: Workstation

Test 6: VMware "get memory size" command
Result: VMware detected

Test 7: VMware emulation mode
Result: Native OS or VMware without emulation mode (enabled acceleration)

tk, 2008
[ www.trapkit.de ]
Anti-Virtual Machine

Adding these lines to .vmx file will prevent some methods:

```python
isolation.tools.getPtrLocation.disable = "TRUE"
isolation.tools.setPtrLocation.disable = "TRUE"
isolation.tools.setVersion.disable = "TRUE"
isolation.tools.getVersion.disable = "TRUE"
monitor_control.disable_directexec = "TRUE"
monitor_control.disable_chksimd = "TRUE"
monitor_control.disable_ntreloc = "TRUE"
monitor_control.disable_selfmod = "TRUE"
monitor_control.disable_reloc = "TRUE"
monitor_control.disable_btinout = "TRUE"
monitor_control.disable_btmemspace = "TRUE"
monitor_control.disable_btpriv = "TRUE"
monitor_control.disable_btseg = "TRUE"
```
Anti-Virtual Machine

If all else fails, try to find the part of the code that checks for a VM and nop it out or bypass it.
Resources

Practical Malware Analysis by Michael Sikorski and Andrew Honig

The Art of Unpacking Mark Vincent Yason, Malcode Analyst, IBM Internet Security Systems

TrapKit - The maker of the ScoopyNG VM detection tool http://www.trapkit.de/research/vmm/scoopyng/index.html