

"Cyber Analytics" Malware and Static Analysis Lecture 4

John Cavazos

Dept of Computer & Information Sciences

University of Delaware



- Malware=Malicious software
- Software with malicious intent
 - Different from software with bugs



- Programs that attach themselves to another program to gain access to your machine
 - Insertion phase is inserting itself into file
 - Execution phase is performing some (possibly null) action
- May do nothing on your machine or may destroy all your files
- Seek to use your machine as a launching point to infect other machines



- Like a virus but they are self-contained programs (they don't need a host)
- Copy themselves from from machine-tomachine
- Scan for other vulnerable machines



- Program with an overt purpose (known to the user) and a covert purpose (not known to user)
 - Often called a Trojan



- Some programs are "free" but they support their costs by sending ads to your machines
 - E.g., Kazaa



- You download a music player
- The music player includes another program that is installed and running continuously
- This program records the websites you visit and send them to a database



- Internet Security Emerging Threats List
 - Hackers use Instant Messaging to Spread Viruses And Worms
 - Phishing fraud is prevalent and sophisticated
 - Viruses attack any mobile device
 - Hackers target online banking accounts
 - Internet crimes continue to go unreported



- Taking malware apart to study it
 - Static analysis
 - Looking at malware internals
 - NOT running
 - Dynamic analysis
 - Looking at malware behavior
 - Running it and capture behavior

Program Analysis

- Given an executable, how do we find out what it does?
 - Try to find the program online
 - Analyze source code to find clues
 - Search for the name of the program
 - Perform source code review
 - Execute the program in a sandbox
 - Some programs can break out of a sandbox / jail

Program Compilation

- Compiler
 - Translates HLL code to Assembly / ILL
- Assembler
 - Translates Assembly code to machine language
- Linker
 - Creates object code out of several modules
 - A program usually makes library calls (stdio)

Program Compilation

- Statically Linked: All library code is part of the object code
- Dynamically Linked: Program calls library functions. (DLL)
- Stripping: Removes all human-readable symbols from object code.
 - Combats reverse engineering.
- Packing with UPX, etc.
 - upx.sourceforge.net
 - Compresses source code (achieves ratios of 20% 40%)

Program Analysis

- Static Analysis:
 - Determine the type of executable
 - ELF file in Unix
 - Exe-type in Windows
 - Symbol Extraction:
 - Use a program like "strings" to find symbols
 left in object code
 - Names give hints on program
 - Will not work for stripped files



strings /bin/mkdir

@(#) Copyright (c) 1983, 1992, 1993

The Regents of the University of California. All rights reserved.

\$FreeBSD: src/bin/mkdir/mkdir.c,v 1.26 2002/06/30 05:13:54 obrien Exp \$

@(#)PROGRAM:mkdir PROJECT:file_cmds-251

m:pv

invalid file mode: %s

```
mkdir: created directory '%s'
```

```
usage: mkdir [-pv] [-m mode] directory ...
```

- Investigate source code
- Use Reversing Tools:
 - Disassembler:
 - Decodes binary machine code into a readable assembly language text
 - IDA-Pro
 - ILDasm (Microsoft .Net IL disassembler)
 - RADARE2

Decompilers

- Attempt to produce a high-level language source-code-like representation from a binary
- Never completely possible because
 - The compiler removes some information
 - The compiler optimizes the code

- Artifacts to look for:
 - Names of functions
 - Especially API functions
 - Data strings
 - Names of constant strings
 - Names of directories
 - Identification of compiler

- Compilers generate different types of code for the same high-level language features
 - Function Calls:
 - Order in which parameters are pushed on stack
 - Use of certain registers to pass variables
 - Use of stack / registers to return a value
 - Division of labor between callee and caller

- This allows us to recognize the compiler with which an executable was created
- Programmers using assembly will not follow the same standards throughout the code
 - Hence, we can recognize assembly writers as well



Break

<pre></pre>	Name Name Mailt Get_keys Mailt Get_keys GatsynckeyState(x) GatsynckeyS	
Image: Normer Provided Structure Image: Normer Provided Structure Image: Normer Provided Structure Image: Norme	Name F Maillt F get_keys F main C Sleep(x) C AllocConsole() C Get&syncKcyState(x) C ShowWindow(x,x) C FindWindow(x,x) F closesocket(x) F send(x,x,x) F recv(x,x,x) F recv(x,x,x) F socket(x,x,x) F socket(x,x,x) F blone(v) Line 3 of 541 **** Strings window	
Bit W ··· · N × St. H ··· · S N K /·· · / : ; # # U · h A A * A A W ··· · N × St. H ··· · S N K /·· · / : ; # # U · h A A * A A W ··· · N × St. H ··· · S N K /·· · / : ; # # U · h A A * A A W ··· · N × St. H ··· · S N K /·· · / : ; # # U · h A A * A A W ··· · N × St. H ··· · S N K /·· · / : ; # # U · h A A * A A W ··· · N × St. H ··· · · · · · · · · · · · · · · · ·	Name F Maillt F get_keys F main C Sleep(x) C AllocConsole() C Get&syncKcyState(x) C ShowWindow(x,x) C FindWindow(x,x) F closesocket(x) F send(x,x,x) F recv(x,x,x) F recv(x,x,x) F socket(x,x,x) F socket(x,x,x) F blone(v) Line 3 of 541 **** Strings window	
<pre> Hex YewA In Exports In monts Names Tructions *** Strings A Structures En Enume Ext: 00402468 ext: 00402468</pre>	Name F Maillt F get_keys F main C Sleep(x) C AllocConsole() C Get&syncKcyState(x) C ShowWindow(x,x) C FindWindow(x,x) F closesocket(x) F send(x,x,x) F recv(x,x,x) F recv(x,x,x) F socket(x,x,x) F socket(x,x,x) F brane(v) Line 3 of 541 **** Strings window	
<pre>Hex ViewA D Exports D Exports N Names P Functions ** Strings A Structures En Enums rext:00402/468 ext:00402/468 ext:00402/468 icet:00402/468 icet:0040</pre>	Name F Maillt F get_keys F main C Sleep(x) C AllocConsole() C Get&syncKcyState(x) C ShowWindow(x,x) C FindWindow(x,x) F closesocket(x) F send(x,x,x) F recv(x,x,x) F recv(x,x,x) F socket(x,x,x) F socket(x,x,x) F brane(v) Line 3 of 541 **** Strings window	
<pre>ext:00402468 ext:00402466 locret_402468: ; CODE XREF: check_managed_app+3A[†]j ext:00402466 locret_402468 ; tetn ext:00402466 check_managed_app endp ext:00402466 ; illillillillillillillillillillillillill</pre>	Name F Maillt F get_keys F main C Sleep(x) C AllocConsole() C Get&syncKcyState(x) C ShowWindow(x,x) C FindWindow(x,x) F closesocket(x) F send(x,x,x) F recv(x,x,x) F recv(x,x,x) F socket(x,x,x) F socket(x,x,x) F brane(v) Line 3 of 541 **** Strings window	
<pre>rext:00402468 rext:00402468 rext:00402468 rext:00402468 retn rext:00402468 retn rext:00402468 retn rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402477 rext:00402478 rext:00402477 rext:00402477 rext:00402477 rext:00402477 rext:00402477 rext:00402477 rext:00402477 rext:00402486 rext:00402487 rext:00402480 rext:00402480 rext:00402480 rext:00402480 rext:00402480 rext:00402480 rext:00402480 rext:00402480 rext:00402480 rext:00402480 rext:00402480 rext:00402480 rext:00402480 rext:00402480 rext:00402495 rext:00402495 rext:00402496 rext:00402</pre>	Name F Maillt F get_keys F main C Sleep(x) C AllocConsole() C Get&syncKcyState(x) C ShowWindow(x,x) C FindWindow(x,x) F closesocket(x) F send(x,x,x) F recv(x,x,x) F recv(x,x,x) F socket(x,x,x) F socket(x,x,x) F brane(v) Line 3 of 541 **** Strings window	
<pre>rext:00402468 rext:00402468 rext:00402468 rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402466 rext:00402477 rext:00402478 rext:00402478 rext:00402477 rext:00402477 rext:00402477 rext:00402478 rext:00402478 rext:00402477 rext:00402478 rext:00402478 rext:00402478 rext:00402478 rext:00402478 rext:00402478 rext:00402478 rext:00402478 rext:00402478 rext:00402486 rext:00402487 rext:00402488 rext:00402488 rext:00402488 rext:00402488 rext:00402488 rext:00402488 rext:00402488 rext:00402488 rext:00402488 rext:00402488 rext:00402488 rext:00402488 rext:00402488 rext:00402488 rext:00402480 rext:004024</pre>	Name F Maillt F get_keys F main C Sleep(x) C AllocConsole() C Get&syncKcyState(x) C ShowWindow(x,x) C FindWindow(x,x) F closesocket(x) F send(x,x,x) F recv(x,x,x) F recv(x,x,x) F socket(x,x,x) F socket(x,x,x) F brane(v) Line 3 of 541 **** Strings window	
<pre>:ext:80402468 locret_402468: ; CODE XREF: check_managed_app+3A1j :ext:80402468 ; check_managed_app endp :ext:80402468 check_managed_app endp :ext:80402468 check_managed_app endp :ext:80402466 [! :ext:80402466] ! :ext:80402466 ; !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</pre>	F Mailt F get_keys F _main C Sleep(x) C AllocConsole() C Get&syncKeyState(x) C ShowWindow(x,x) C FindWindow(x,x) F closesocket(x) F send(x,x,x) F closesocket(x) F send(x,x,x) F connect(x,x,x) F socket(x,x,x) F btons(v) Line 3 of 541	
<pre>:ext:0040246B retn :ext:0040246B reck_managed_app endp :ext:0040246C [:ext:0040246C ; !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</pre>	F get_keys Fmain S leep(x) C AllocConsole() C GetAsyncKeyState(x) C ShowWindow(x,x) C FindWindow(x,x) F closesocket(x) F send(x,x,x) F recv(x,x,x) F connect(x,x,x) F socket(x,x,x) F bione(v) Line 3 of 541 **** Strings window	
<pre>:ext:80402468 check_managed_app endp :ext:80402466 iext:80402466 ; !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</pre>	F main Sleep(x) AllocConsole() GetAsyncKeyState(x) ShowWindow(x,x) F closesocket(x) F send(x,x,x) F cerv(x,x,x) F cerv(x,x,x) F socket(x,x,x) F socket(x,x,x) F bitnef(x) Line 3 of 541 **** Strings window	
<pre>:ext:00402466 rext:0040246C; ; !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</pre>	C AllocConsole() C GetAsyncKeyState(x) C ShowWindow(x,x) C Findw/indowA(x,x) F closesocket(x) F send(x,x,x) F recv(x,x,x) F connect(x,x,x) F socket(x,x,x) F s	
<pre>:ext:0040246C ; !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</pre>	C GetAsyncKeyState(x) C ShowWindow(x,x) C FindWindow(x,x) F closesocket(x) F send(x,x,x) F connect(x,x,x) F connect(x,x,x) F socket(x,x,x) F bitnef(x) Line 3 of 541	
<pre>:ext:0040246C :ext:0040246C :ext:0040246C :ext:0040246C :ext:0040246C push 18h ; ExceptionInfo :ext:0040246C push offset stru_400330 ; int :ext:00402473 callSEH_prolog :ext:00402477 callSEH_prolog :ext:00402477 callSEH_prolog :ext:00402477 callSEH_prolog :ext:00402477 callSEH_prolog :ext:00402477 callSEH_prolog :ext:00402484 movEbp-18h], esp :ext:00402484 push esi , sp :ext:00402488 push esi ; lpVersionInformation :ext:00402488 push esi ; lpVersionInformation :ext:00402488 call ds:_imp_GetVersionExA@4 ; Get extended information about the :ext:00402480 push esi ; version of the operating system :ext:00402490 movosplatform, ecx :ext:00402491 movwinmajor, eax :ext:00402492 movwinminor, edx :ext:00402496 movwinminor, edx :ext:00402496 movwinminor, edx :ext:00402496 movsi, [esi+0Ch] :ext:00402406 movosver, esi</pre>	C ShowWindow(x,x) FindWindowA(x,x) F closesocket(x) F send(x,x,x) F recv(x,x,x) F connect(x,x,x) F socket(x,x,x) F btons(v) Line 3 of 541	
<pre>:ext:0040246C ; Attributes: library function :ext:0040246C mainCRTStartup proc near :ext:0040246C push 18h ; ExceptionInfo :ext:0040246E push offset stru_408338 ; int :ext:00402476 callSEH_prolog :ext:004024778 mov edi, 94h :ext:004024778 mov eax, edi :ext:00402477 callChkstk :ext:00402487 mov esi, esp :ext:00402480 mov [esi], edi :ext:00402488 push esi ; lpVersionInformation :ext:00402480 call ds:imp_GetVersionExA@4 ; Get extended information about the :ext:00402480 mov eax, [esi+10h] :ext:00402480 mov eax, [esi+4] :ext:00402495 mov eax, [esi+4] :ext:00402496 mov ex, [esi+8] :ext:00402496 mov ex, [esi+8] :ext:00402496 mov ex, [esi+8] :ext:00402496 mov ex, [esi+8] :ext:00402496 mov ex, [esi+9] :ext:00402496 mov ex, [esi+10h] :ext:00402496 mov ex, [esi+10h] :ext:00402496 mov ex, [esi+8] :ext:00402496 mov ex, [esi+8] :ext:00402496 mov ex, [esi+8] :ext:00402496 mov ex, [esi+10h] :ext:00402496 mov ex, [esi+10h]</pre>	C FindWindowA(x,x) F closesocket(x) F send(x,x,x,x) F recv(x,x,x,x) F connect(x,x,x) F socket(x,x,x) F blons(v) Line 3 of 541 Une 3 of 541	
<pre>:ext:0040246C :ext:0040246C mainCRTStartup</pre>	F closesocket(x) F send(x,x,x,x) F recv(x,x,x) F connect(x,x,x) F socket(x,x,x) F bhane(v) Line 3 of 541 "" Strings window	
:ext:0040246C push 18h ; ExceptionInfo :ext:0040246C push offset stru_408338 ; int :ext:0040246C push offset stru_408338 ; int :ext:00402473 call _SEH_prolog :ext:00402478 mov edi, 94h :ext:00402470 mov eax, edi :ext:00402477 call _chkstk :ext:00402484 mov [ebp-18h], esp :ext:00402487 mov esi, esp :ext:00402488 push esi :ext:00402480 call ds:_imp_GetVersionExA044 ; Get extended information about the :ext:00402480 call ds:_imp_GetVersionExA044 ; Get extended information about the :ext:00402480 call ds:_imp_GetVersionExA044 ; Get extended information about the :ext:00402480 call ds:_imp_GetVersionExA044 ; Get extended information about the :ext:00402480 mov ecx, [esi+10h] :ext:00402480 mov _osplatform, ecx :ext:00402495 mov _winmajor, eax :ext:00402495 mov ext:041 :ext:00402406 mov winminor, edx <td>F send(x.x.x) F recv(x.x.x) F connect(x.x.x) F socket(x.x.x) F socket(x.x.x) Line 3 of 541</td> <td></td>	F send(x.x.x) F recv(x.x.x) F connect(x.x.x) F socket(x.x.x) F socket(x.x.x) Line 3 of 541	
:ext:0040246E push offset stru_400330 ; int :ext:00402473 callSEH_prolog :ext:00402473 mov edi, 94h :ext:00402470 mov eax, edi :ext:0040247F call _chkstk :ext:00402484 mov [ebp-18h], esp :ext:00402486 mov [esi], edi :ext:00402486 call ds:_imp_GetVersionInformation :ext:00402480 call ds:_imp_GetVersionExA04 ; Get extended information about the :ext:00402480 mov _osplatform, ecx :ext:00402495 mov _osplatform, ecx :ext:00402496 mov _edx, [esi+10h] :ext:00402498 mov _edx, [esi+2] :ext:00402498 mov _edx, [esi+3] :ext:00402498 mov _edx, [esi+10h] :ext:00402406 mov _winmior, edx :ext:00402406 mov _winmior, edx :ext:00402406 mov _esi, [FFFh <td:ext:00402406< td=""> mov _winmior, ed</td:ext:00402406<>	F recv(x,x,x) F connect(x,x,x) F socket(x,x,x) F ktrons(v) Line 3 of 541	
:ext:00402473 callSEH_prolog :ext:00402478 mov edi, 94h :ext:00402477 mov eax, edi :ext:00402477 callChkstk :ext:00402487 mov [ebp-18h], esp :ext:00402488 mov [esi], edi :ext:00402480 mov [esi], edi :ext:00402480 call ds:_imp_GetVersionExA@4; Get extended information :ext:00402480 call ds:_iversion of the operating system :ext:00402480 movosplatform, ecx :ext:00402492 moveax, [esi+10h] :ext:00402490 moveax, [esi+4] :ext:00402498 movwinmajor, eax :ext:00402406 movwinminor, edx :ext:00402406 movwinminor, esi	socket(x,x,x) bitone(v) Line 3 of 541 """ Strings window	
:ext:0040247D mov eax, edi :ext:0040247F call _chkstk :ext:0040248F mov [ebp-18h], esp :ext:00402487 mov [esi], edi :ext:00402488 push esi ; lpVersionInformation :ext:00402480 call ds:impGetVersionExA04 ; Get extended information about the :ext:00402480 call ds:impGetVersionExA04 ; Get extended information about the :ext:0040248C ; version of the operating system :ext:00402492 mov _csi = inpimpimpimpimpimpimpimp	Line 3 of 541	
:ext:0040247F call _chkstk :ext:00402484 mov [ebp-18h], esp :ext:00402487 mov esi, esp :ext:00402489 mov [esi], edi :ext:00402480 call ds:imp_GetVersionExA04 ; Get extended information about the :ext:00402480 call ds:imp_GetVersionExA04 ; Get extended information about the :ext:00402480 call ds:imp_GetVersionExA04 ; Get extended information about the :ext:00402480 call ds:imp_GetVersionExA04 ; Get extended information about the :ext:00402480 call ds:imp_GetVersionExA04 ; Get extended information about the :ext:00402480 mov ecx, [esi+10h] :ext:00402492 mov _osplatform, ecx :ext:00402498 mov _eax, [esi+4] :ext:00402498 mov _edx, [esi+8] :ext:00402403 mov edx, [esi+8] :ext:00402404 mov _winmior, edx :ext:00402404 mov _esi, [FFh :ext:00402404 mov _esi, [esi+8Ch] :ext:00402405 mov _esi, [esi+0Ch] :ext:00402405 mov _esi, [esi+8Ch]	Line 3 of 541	
:ext:00402484 mov [ebp-18h], esp :ext:00402487 mov esi, esp :ext:00402488 push esi ; lpVersionInformation :ext:00402486 call ds:imp_GetVersionExA044 ; Get extended information about the :ext:00402486 call ds:imp_GetVersionExA044 ; Get extended information about the :ext:00402486 ; version of the operating system ; version of the operating system :ext:00402492 mov _osplatform, ecx :ext:00402498 mov _eax, [esi+10h] :ext:00402498 mov _eax, [esi+4] :ext:00402408 mov _eax, [esi+8] :ext:00402408 mov _winmajor, eax :ext:00402408 mov _winminor, edx :ext:00402406 mov _winminor, edx :ext:00402406 mov esi, 7FFFh :ext:0040240F and esi, 7FFFh :ext:00402485 mov _osuer, esi	Strings window	
:ext:00402487 mov esi, esp :ext:00402489 mov [esi], edi :ext:00402480 call ds:imp_GetVersionExA@4 ; Get extended information about the ; version of the operating system :ext:00402480 mov ecx, [esi+10h] :ext:00402490 mov ecx, [esi+10h] :ext:00402495 mov _osplatform, ecx :ext:00402495 mov eax, [esi+4] :ext:00402408 mov eax, [esi+8] :ext:00402408 mov _winmior, eax :ext:00402406 mov _winminor, edx :ext:00402406 mov esi, [Fei+8Ch] :ext:00402406 mov esi, 7FFFh :ext:00402405 mov esi, 7FFFh :ext:00402485 mov _osuer, esi	Strings window	
<pre>:ext:00402488</pre>		
<pre>:ext:0040248C call ds:impGetVersionExAQ4 ; Get extended information about the</pre>	Address Length T S	
:ext:0040248C ; version of the operating system :ext:00402492 mov ecx, [esi+10h] :ext:00402495 mov _osplatform, ecx :ext:0040249B mov eax, [esi+4] :ext:0040249E mov _winmajor, eax :ext:004024A3 mov edx, [esi+8] :ext:004024A6 mov _winminor, edx :ext:004024A6 mov esi, [FFh :ext:004024A6F and esi, 7FFFh :ext:004024B5 mov _osver, esi	Address Eorigan 1 o	ring 🔼
:ext:00402495 nov _osplatform, ecx :ext:00402498 nov eax, [esi+4] :ext:00402498 nov _winmajor, eax :ext:004024A3 nov edx, [esi+8] :ext:004024A6 nov _winminor, edx :ext:004024A6 nov esi, [esi+0Ch] :ext:004024AF and esi, 7FFh :ext:004024B5 nov _osver, esi		nnect
:ext:00402498 mov ēax, [esi+4] :ext:0040249E mov _winmajor, eax :ext:004024A3 mov edx, [esi+8] :ext:004024A6 mov _winminor, edx :ext:004024AC mov esi, [esi+8Ch] :ext:004024AF and esi, 7FFFh :ext:044024B5 mov _osver, esi	"" .rdata:0 00000010 C Co	nnecting\n
:ext:0040249E mov _winmajor, eax :ext:004024A3 mov edx, [esi+8] :ext:004024A6 mov _winminor, edx :ext:004024A6 mov _esi, [esi+0Ch] :ext:004024AF and esi, 7FFFh :ext:004024B5 mov _osver, esi		thostbyname
:ext:004024A6mov _winminor, edx:ext:004024ACmov esi, [esi+0Ch]:ext:004024AFand esi, 7FFh:ext:004024B5mov _osver, esi		SAStartup faile
:ext:004024AC mov esi, [esi+0Ch] :ext:004024AF and esi, 7FFFh :ext:004024BS mov _osver, esi		1TP.log
:ext:004024AF and esi,7FFh :ext:004024B5 mov _osver,esi		tp.scu.edu ng33k@gmail.
		e Log Dude
:ext:0040248B cmp ecx, 2		n[CAPS LOCk
:ext:004024BE jz short loc 4024CC		n['\'']\r\n
.cxt:00402402 0 02 esi, 8000h		n[]}]\r\n
:ext:004024C6 mov _osver, esi		n[\\]]\r\n
:ext:004024CC :ext:004024CC loc 4024CC: ; CODE XREF: mainCRTStartup+521j		n[[{]\r\n n[~]\r\n
:ext:004024CC loc_4024CC: ; CODE XREF: mainCRTStartup+52†j :ext:004024CC shl eax, 8		ni Thrìn
	>	
e 'C:\Program Files\IDA\idc\onload.idc' ction 'OnLoad'		
ing the input file to explore the input file right now.		
ype' at 00408D10 is deleted		
ignature: Microsoft VisualC 2-7/net runtime ype information		
gate_stkargs: function is already typed ment information is propagated		



- Malware writers can use anti-reversing techniques
 - Eliminate symbolic information
 - Encrypt code
 - Code obfuscation
 - Make high level constructs difficult to understand



- Anti-debugger Methods:
 - Use the IsDebuggerPresent API to protect against user-level debuggers
 - Use the NTQuerySystemInformation API to determine if a kernel debugger is attached to the system
 - Set a trap flag and check whether it is still there
 - A debugger would "swallow" it
 - Put in bogus bytes over which the code jumps
 - Does not work for all disassemblers



- Set a trap flag and check whether it is still there
 - A debugger would "swallow" it

Note: A good explanation of what this means is found here (<u>http://leons.im/posts/anti-debug-with-trap-flag-register/</u>). Briefly, this means is that an application can set up a "trap" (also known as an "exception") that must be handled during the execution of the program. "An exception is an event that occurs during the execution of a program, and requires execution of code outside the normal flow of control."[1]

During normal execution, the exception can be set up to be "handled" in a way that execution of the code continues along one path of the code, allowing the program to proceed. However during debug mode, "exceptions" or "traps" will not be "handled" and therefore execution will go down a different path of the code. A malware writer can write their program so that it exits when execution goes down this different path. Thus, executing the program during normal versus debug mode can take two different paths and the malware writer can cause the program to exit when execution goes down the different debug path.

[1] https://msdn.microsoft.com/en-us/library/windows/desktop/ms680657(v=vs.85).aspx



Packer Malware Infected host

Packer functionalities

- Compress
- Encrypt (custom)
- Randomize (polymorphism)
- Anti-debug technique (fake jmps)
- Add junk code
- Anti-VM (next week!)

