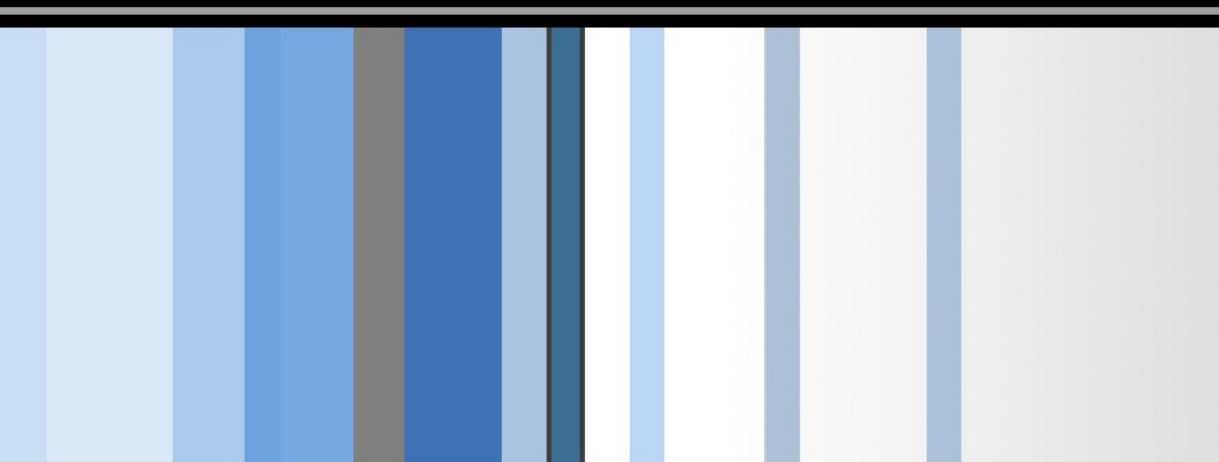


CSC 471 Advanced Topics in Computer Security

Modern Malware Analysis

Basic Analysis, DLL Injection

Si Chen (schen@wcupa.edu)



Course Outline

- Introduction
 - Virtual Machine
 - Static Analysis
- A “Hello World” Malware Example: DLL Injection (hack_dll.zip)
 - Behavior
 - Analysis
 - Source code
- DLL Injection

- What is a virtual machine?
 - Simply, a computer in your computer
 - Really, a segregated virtual environment that emulates real hardware
 - There are different types/methods



[VirtualBox](#)



[VMware](#)



[Parallels](#)

- Why are we using a virtual machine? (for this course)
 - Safety, reliability, consistency, it's easy
 - Keep the malware in a contained environment
 - Snapshots
 - Completely 100% revert the VM to an earlier state
 - If things go bad, no one cares

Static Analysis

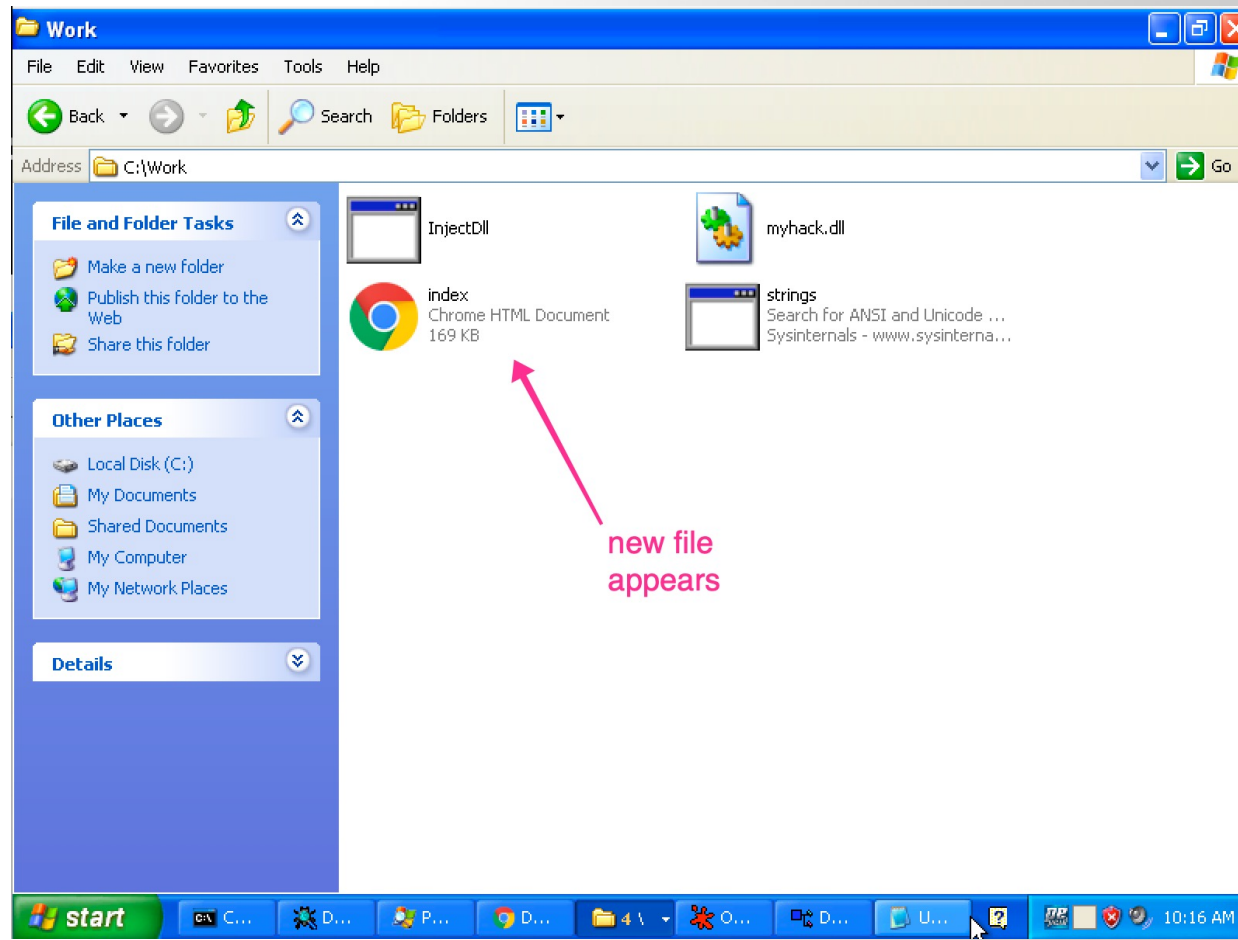
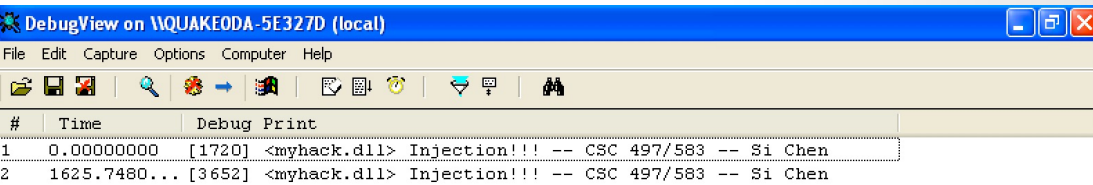
- Analyzing a sample without executing any code
- Safe
 - Infer functionality
- Provides a lot of useful information to guide dynamic and advanced analysis
- Lots of tools involved
- Can be an easy way to find signatures
 - URLs, filenames, registry keys

Let's try our first "Malware"

- Download and run XP VM image
- Open command line terminal and go to C:\Work
- Open a Notepad
- Open DebugView
- Open Process Explorer and find the PID of Notepad
- In command line, type
 - InjectDll.exe <PID OF NOTEPAD> myhack.dll

Screenshots

```
C:\Work>InjectDll.exe 3652 c:\Work\myhack.dll
InjectDll<"c:\Work\myhack.dll"> success!!!
```



Screenshots

Process Explorer - Sysinternals: www.sysinternals.com [QUAKE0DA-5E327D\quake0day] (Administrator)

File Options View Process Find DLL Users Help

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
Dbgview.exe		1,052 K	1,960 K	1892	DebugView	Sysinternals
procexp.exe		18,676 K	14,844 K	1648	Sysinternals Process Explorer	Sysinternals - www.sysinter...
chrome.exe	2.00	67,300 K	47,600 K	2452	Google Chrome	Google Inc.
chrome.exe		1,932 K	508 K	2464	Google Chrome	Google Inc.
chrome.exe	1.00	45,192 K	51,496 K	2848	Google Chrome	Google Inc.
chrome.exe		18,100 K	2,112 K	2940	Google Chrome	Google Inc.
OLLYDBG.EXE		9,020 K	2,364 K	3224		
loaddll.exe		616 K	416 K	3260		
PEID.exe		3,396 K	344 K	3276		
depends.exe		6,708 K	12,516 K	3476	Dependency Walker for Win...	Microsoft Corporation
notepad.exe		2,016 K	6,988 K	3652	Notepad	Microsoft Corporation

PID: 3652

Name	Description	Company Name	Path
kernel32.dll	Windows NT BASE API Client DLL	Microsoft Corporation	C:\WINDOWS\system32\kernel32.dll
locale.nls			C:\WINDOWS\system32\locale.nls
lpk.dll	Language Pack	Microsoft Corporation	C:\WINDOWS\system32\lpk.dll
msacm32.dll	Microsoft ACM Audio Filter	Microsoft Corporation	C:\WINDOWS\system32\msacm32.dll
msasn1.dll	ASN.1 Runtime APIs	Microsoft Corporation	C:\WINDOWS\system32\msasn1.dll
MSCTF.dll	MSCTF Server DLL	Microsoft Corporation	C:\WINDOWS\system32\MSCTF.dll
MSCTFIME.IME	Microsoft Text Frame Work Servic...	Microsoft Corporation	C:\WINDOWS\system32\MSCTFIME.IME
msv1_0.dll	Microsoft Authentication Package ...	Microsoft Corporation	C:\WINDOWS\system32\msv1_0.dll
msvcrt.dll	Windows NT CRT DLL	Microsoft Corporation	C:\WINDOWS\system32\msvcrt.dll
mswsock.dll	Microsoft Windows Sockets 2.0 S...	Microsoft Corporation	C:\WINDOWS\system32\mswsock.dll
myhack.dll			C:\Work\myhack.dll
netapi32.dll	Net Win32 API DLL	Microsoft Corporation	C:\WINDOWS\system32\netapi32.dll
normaliz.dll	Unicode Normalization DLL	Microsoft Corporation	C:\WINDOWS\system32\normaliz.dll
notepad.exe	Notepad	Microsoft Corporation	C:\WINDOWS\system32\notepad.exe
ntdll.dll	NT Layer DLL	Microsoft Corporation	C:\WINDOWS\system32\ntdll.dll

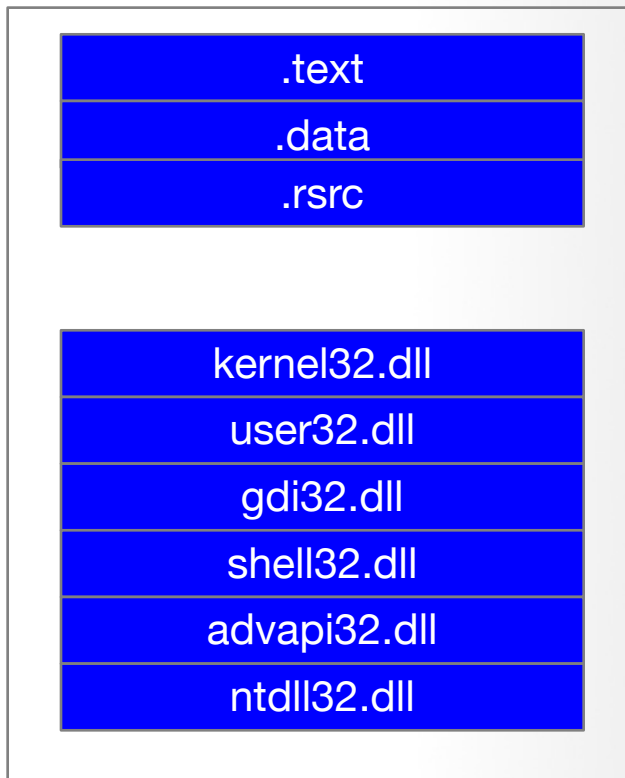
Injected DLL

CPU Usage: 3.00% Commit Charge: 62.84% Processes: 30 Physical Usage: 89.14%

Dynamic-link library (DLL)

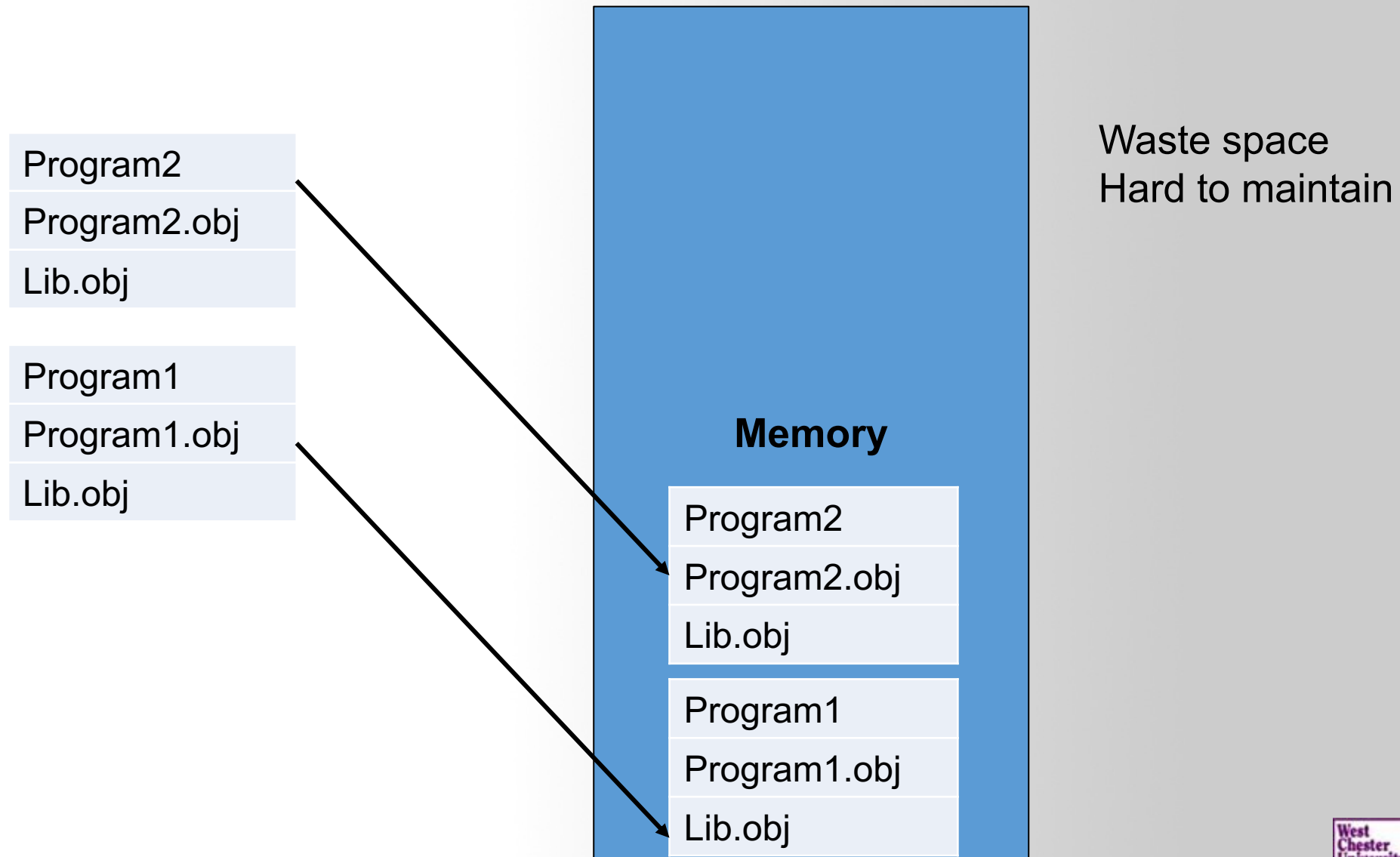
- **Dynamic-link library** (or **DLL**) is [Microsoft](#)'s implementation of the [shared library](#) concept in the [Microsoft Windows](#)

Notepad.exe Process

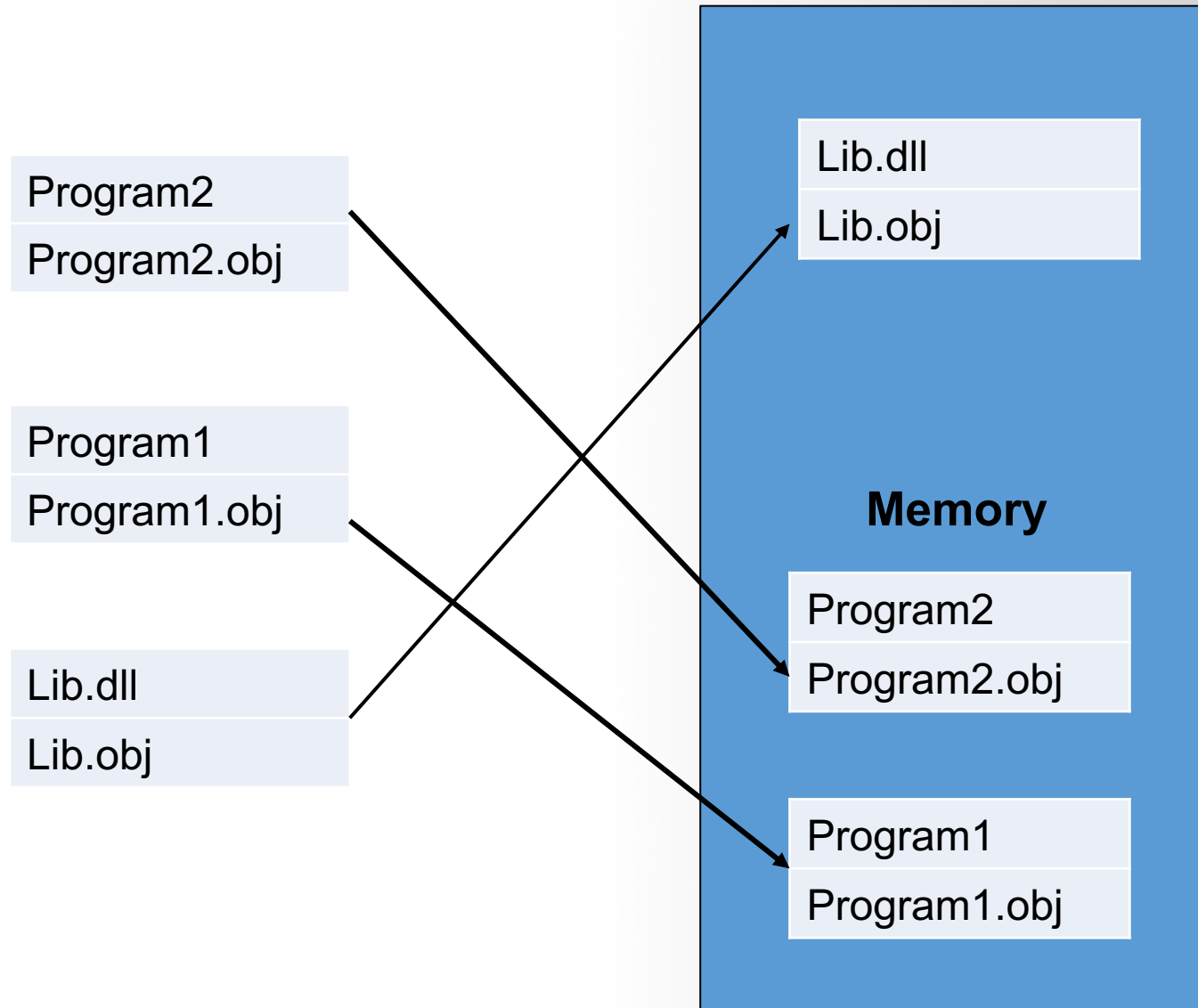


Dynamic Linking

Drawbacks of Static Linking



Dynamic Linking



Dynamic Linking in Linux and Windows

Linux	Windows
ELF file	.exe (PE)
.so (Shared object file)	.dll (Dynamic Linking Library)
.a	.lib (static linking library)
.o (intermediate file between compilation and linking, object file)	.obj

DLL Injection

- DLL injection is method of **injecting code** to some other processe's address space and **executing that piece of code on behalf of that process**.
- DLL injection provides a platform for **manipulating the execution of a running process**.
 - It's very commonly used for logging information while reverse engineering.
 - It has gained bad name for itself since it's mostly used by **malware** for stealth purposes:
 - Hiding malicious code into system process
 - Winlogon.exe, services.exe, svchost.exe explorer.exe
 - Open backdoor port
 - Connect remote server
 - Keylogging...
 - It's also frequently used within the game hacking world to code bots

DLL Injection

Memory Viewer

File Search View Debug Tools Kernel tools

Minesweeper

Game Help

3E21

Comment

112

exe+1390 [FFFFFFFF]

exe+400C

[XP.exe+108C] [7629D9F3]

A4D 23117

exe+3E5E

the stack

Size=1000 Physical Address=31DA2

B 0C 0D 0E 0F 0123456789ABCDEF

0 8E 00 00 00 (...

0 28 00 00 00 C...

0 10 00 00 00 X.Y.Z.Z.Y...

0 59 00 00 00 p...

0 E9 03 00 00 q..... o...

0 E8 03 00 00 /

0 EA 03 00 00

0 EB 03 00 00

0 EC 03 00 00

0 EC 03 00 00

0 EC 03 00 00

0 EC 03 00 00

010050B0 BE 02 00 00 EC 03 00 00 C0 02 00 00 EC 03 00 00

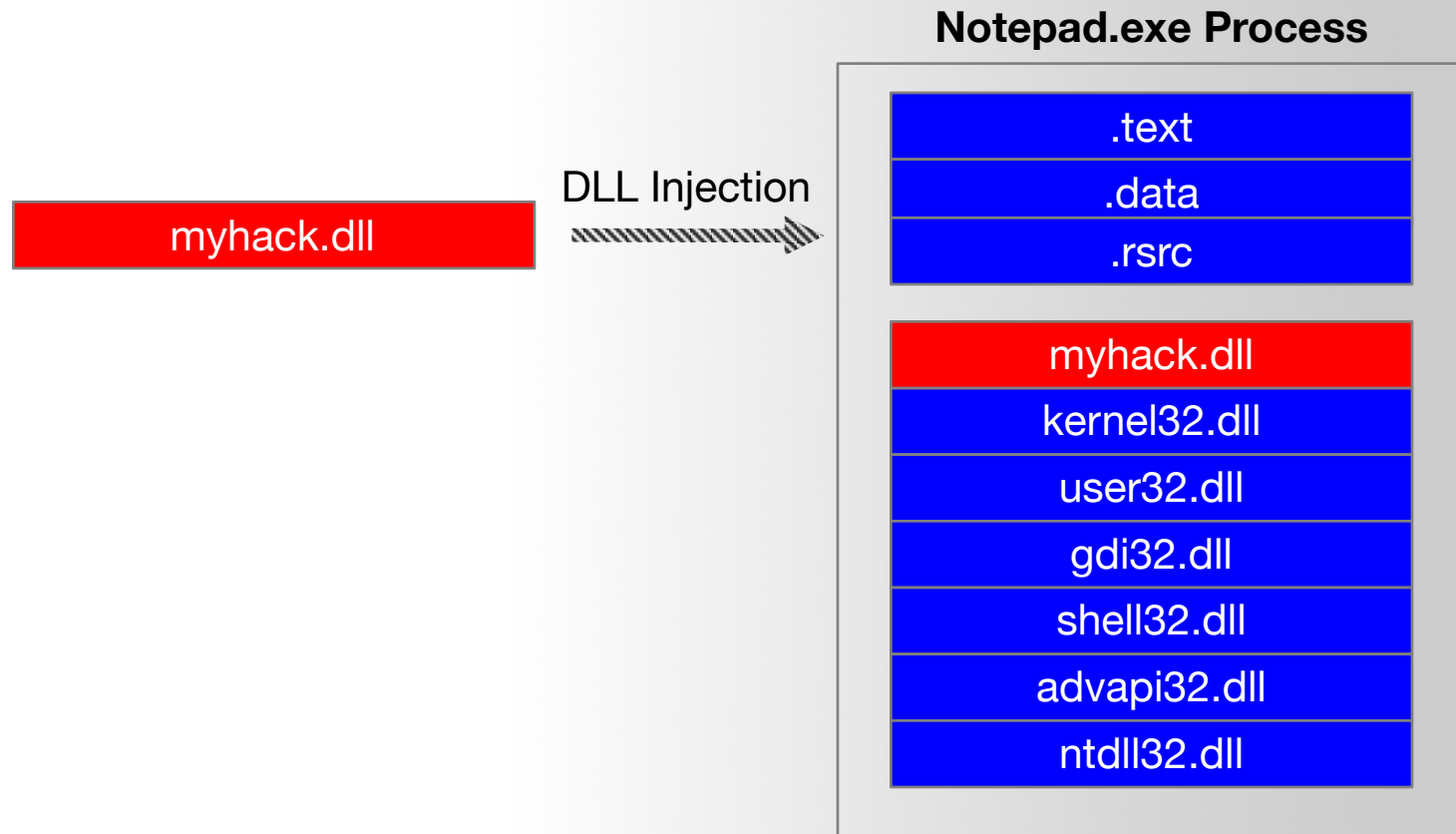
010050C0 C2 02 00 00 EC 03 00 00 00 00 00 00 00 00 00 00

010050D0 28 12 00 01 1C 12 00 01 0C 12 00 01 00 12 00 01

DLL Injection



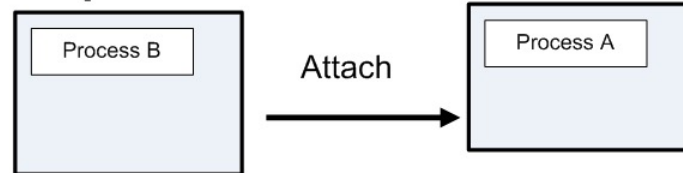
DLL Injection



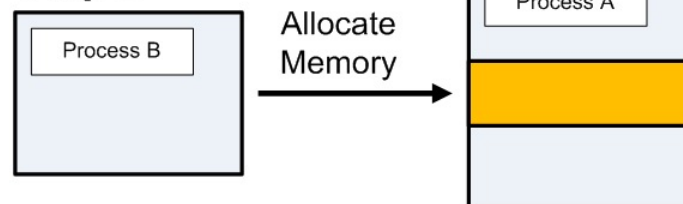
DLL Injection

DLL Injection

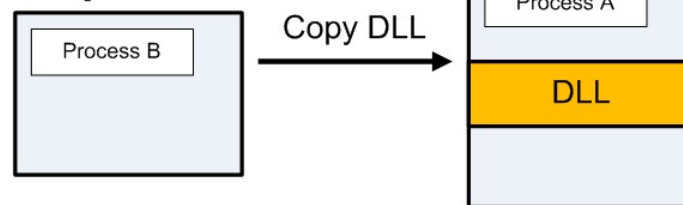
Step 1



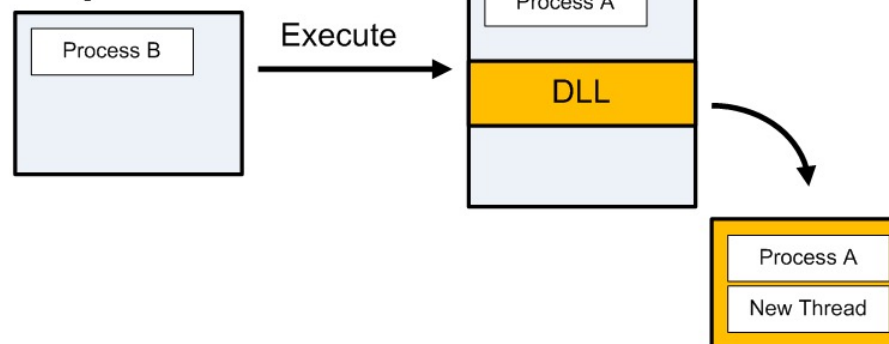
Step 2



Step 3



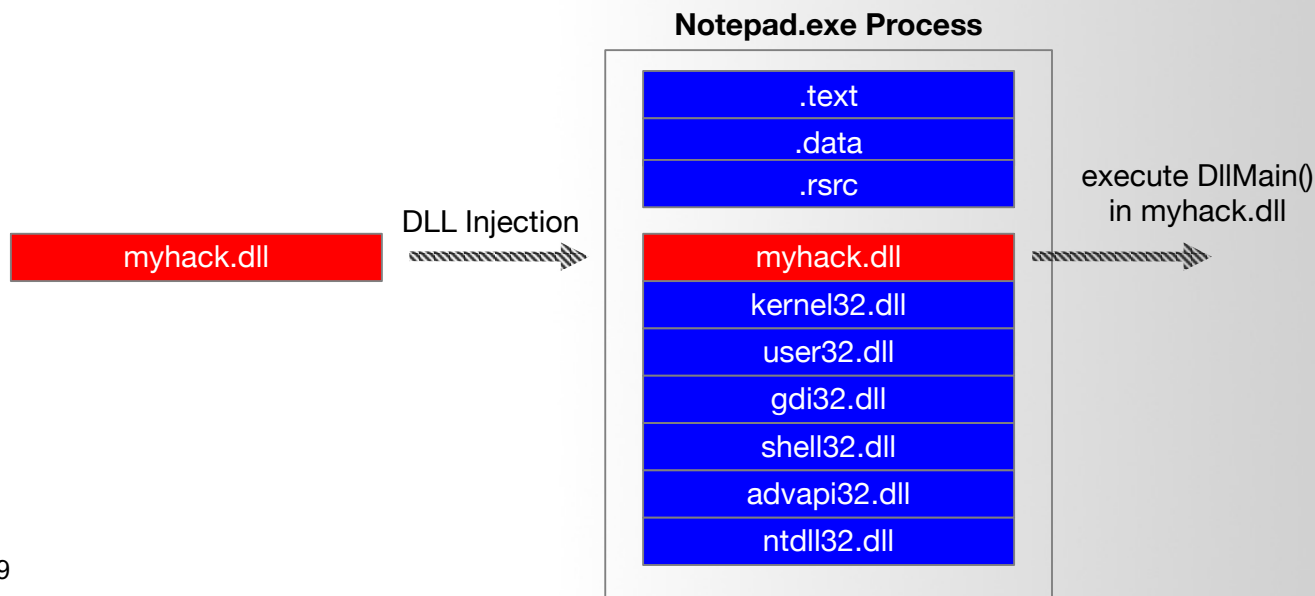
Step 4



DllMain entry point

05/30/2018 • 7 minutes to read

An optional entry point into a dynamic-link library (DLL). When the system starts or terminates a process or thread, it calls the entry-point function for each loaded DLL using the first thread of the process. The system also calls the entry-point function for a DLL when it is loaded or unloaded using the [LoadLibrary](#) and [FreeLibrary](#) functions.



Source Code of myhack.dll

```
myhack.cpp > No Selection

1 #include "windows.h"
2 #include "tchar.h"
3
4 #pragma comment(lib, "urlmon.lib")
5
6 #define DEF_URL      (L"http://www.naver.com/index.html")
7 #define DEF_FILE_NAME (L"index.html")
8
9 HMODULE g_hMod = NULL;
10
11 DWORD WINAPI ThreadProc(LPVOID lParam)
12 {
13     TCHAR szPath[_MAX_PATH] = {0,};
14
15     if( !GetModuleFileName( g_hMod, szPath, MAX_PATH ) )
16         return FALSE;
17
18     TCHAR *p = _tcsrchr( szPath, '\\ ' );
19     if( !p )
20         return FALSE;
21
22     _tcscpy_s(p+1, _MAX_PATH, DEF_FILE_NAME);
23
24     URLDownloadToFile(NULL, DEF_URL, szPath, 0, NULL);
25
26     return 0;
27 }
28
29 BOOL WINAPI DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
30 {
31     HANDLE hThread = NULL;
32
33     g_hMod = (HMODULE)hinstDLL;
34
35     switch( fdwReason )
36     {
37     case DLL_PROCESS_ATTACH :
38         OutputDebugString(L"<myhack.dll> Injection!!! -- CSC 497/583 -- Dr. Chen");
39         hThread = CreateThread(NULL, 0, ThreadProc, NULL, 0, NULL);
40         CloseHandle(hThread);
41         break;
42     }
43
44     return TRUE;
45 }
```

Source Code of myhack.dll

fdwReason [in]

The reason code that indicates why the DLL entry-point function is being called. This parameter can be one of the following values.

Value	Meaning
DLL_PROCESS_ATTACH 1	<p>The DLL is being loaded into the virtual address space of the current process as a result of the process starting up or as a result of a call to LoadLibrary. DLLs can use this opportunity to initialize any instance data or to use the TlsAlloc function to allocate a thread local storage (TLS) index.</p> <p>The <i>lpReserved</i> parameter indicates whether the DLL is being loaded statically or dynamically.</p>

```
28
29 BOOL WINAPI DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
30 {
31     HANDLE hThread = NULL;
32
33     g_hMod = (HMODULE)hinstDLL;
34
35     switch( fdwReason )
36     {
37     case DLL_PROCESS_ATTACH :
38         OutputDebugString(L"<myhack.dll> Injection!!! -- CSC 497/583 -- Dr. Chen");
39         hThread = CreateThread(NULL, 0, ThreadProc, NULL, 0, NULL);
40         CloseHandle(hThread);
41         break;
42     }
43
44     return TRUE;
45 }
```

Source Code of myhack.dll

```
myhack.cpp > No Selection
1 #include "windows.h"
2 #include "tchar.h"
3
4 #pragma comment(lib, "urlmon.lib")
5
6 #define DEF_URL      (L"http://www.naver.com/index.html")
7 #define DEF_FILE_NAME (L"index.html")
8
9 HMODULE g_hMod = NULL;
10
11 DWORD WINAPI ThreadProc(LPVOID lParam)
12 {
13     TCHAR szPath[_MAX_PATH] = {0,};
14
15     if( !GetModuleFileName( g_hMod, szPath, MAX_PATH ) )
16         return FALSE;
17
18     TCHAR *p = _tcsrchr( szPath, '\\ ' );
19     if( !p )
20         return FALSE;
21
22     _tcscpy_s(p+1, _MAX_PATH, DEF_FILE_NAME);
23
24     URLDownloadToFile(NULL, DEF_URL, szPath, 0, NULL);
25
26     return 0;
27 }
28
29 BOOL WINAPI DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
30 {
31     HANDLE hThread = NULL;
32
33     g_hMod = (HMODULE)hinstDLL;
34
35     switch( fdwReason )
36     {
37     case DLL_PROCESS_ATTACH :
38         OutputDebugString(L"<myhack.dll> Injection!!! -- CSC 497/583 -- Dr. Chen");
39         hThread = CreateThread(NULL, 0, ThreadProc, NULL, 0, NULL);
40         CloseHandle(hThread);
41         break;
42     }
43
44     return TRUE;
45 }
```

■ Finding Strings ^[1]

- A string in a program is a sequence of characters such as “the.”
- A program contains strings if it prints a message, connects to a URL, or copies a file to a specific location.
- Searching through the strings can be **a simple way to get hints about the functionality of a program.**
 - For example, if the program accesses a URL, then you will see the URL accessed stored as a string in the program.
- You can use the **Strings** program to search an executable for strings, which are typically stored in either ASCII or Unicode format.

Static analysis (myhack.dll)

```
C:\Work>strings.exe myhack.dll_
```

```
modf
ldexp
_cabs
_hypot
fmod
frexp
_y0
_y1
_yn
_logb
_nextafter
index.html
http://www.naver.com/index.html
<myhack.dll> Injection!!! -- CSC 497/583 -- Si Chen
QI\
QI\
QI\
QI\
```

```
BOOL WINAPI DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
    HANDLE hThread = NULL;

    g_hMod = (HMODULE)hinstDLL;

    switch( fdwReason )
    {
        case DLL_PROCESS_ATTACH :
            OutputDebugString(L"<myhack.dll> Injection!!! -- CSC 497/583 -- Dr. Chen");
            hThread = CreateThread(NULL, 0, ThreadProc, NULL, 0, NULL);
            CloseHandle(hThread);
            break;
    }

    return TRUE;
}
```

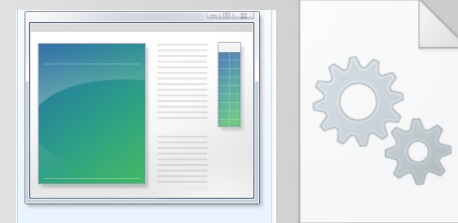
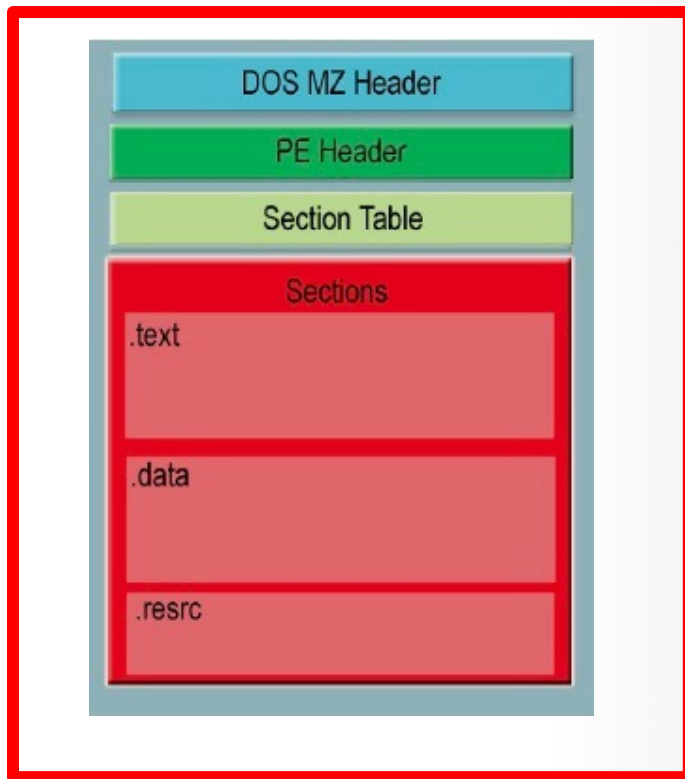

Static analysis (myhack.dll)

```
4%5
7.787K7R7^7v7<7
7g8n8
9&9R9
9%.:. :g:r:e<
>&>+>6>A>U>
?l?
0G0^0i0q0!0
1A1^1
2"2+363
5<535c5
6"6j6
7<7
7.8
9:9I9I9s9
:#:A:h:}:
;>;;H;a;r;i;
<"<></<J<Q<
=U>
1o2M3t3
6k6
7^7>7
8>83808J8Z8
;3;s;
<!<'<+<1<5<?<R<L<v<
```

Sometimes the strings detected by the Strings program are not actual strings.

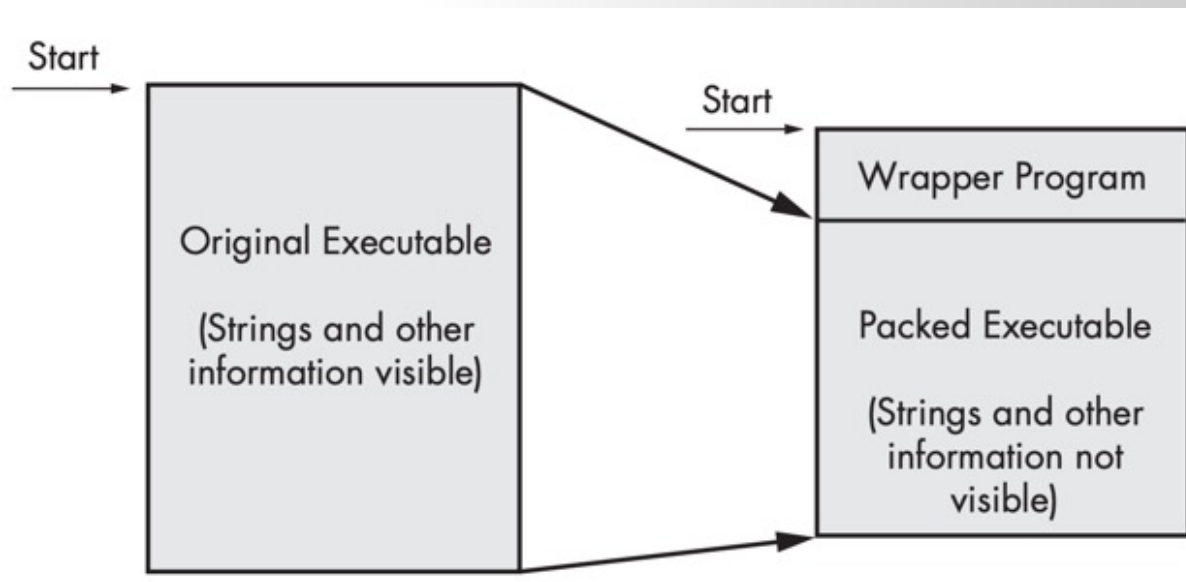
Portable Executable (PE) file

- A Portable Executable (**PE**) **file** is the standard binary **file** format for an **Executable (.exe) or DLL** under Windows NT, Windows 95, and Win32.

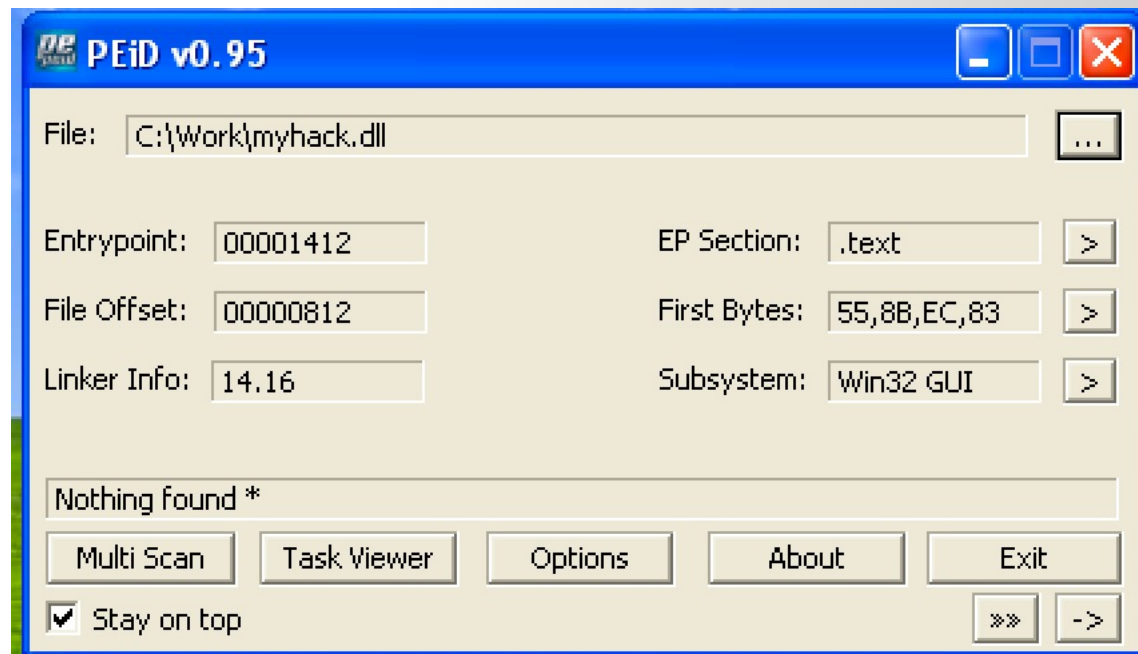


Packed and Obfuscated Malware

- Malware writers often use **packing or obfuscation** to make their files more difficult to detect or analyze.
- **Obfuscated** programs are ones whose execution the malware author has attempted to hide.
- **Packed** programs are a subset of obfuscated programs in which the malicious program is compressed and cannot be analyzed.
- Both techniques will severely limit your attempts to statically analyze the malware.



Packed and Obfuscated Malware



Exploring Dynamically Linked Functions with Dependency Walker

Dependency Walker - [myhack.dll]

File Edit View Options Profile Window Help

MYHACK.DLL

- KERNEL32.DLL
- URLMON.DLL
 - MSVCRT.DLL
 - NTDLL.DLL
 - OLE32.DLL
 - OLEAUT32.DLL
 - RPCRT4.DLL
 - SHLWAPI.DLL
- ADVAPI32.DLL
- GDI32.DLL
- KERNEL32.DLL
- MSVCRT.DLL
- USER32.DLL

PI	Ordinal ^	Hint	Function	Entry Point
C	N/A	116 (0x0074)	URLDownloadToFileW	Not Bound

E	Ordinal ^	Hint	Function	Entry Point
0#	100 (0x0064)	N/A	N/A	0x0009A681
0#	101 (0x0065)	N/A	N/A	0x00029102
0#	102 (0x0066)	N/A	N/A	0x0006EF11
0#	103 (0x0067)	N/A	N/A	0x00040695
0#	104 (0x0068)	N/A	N/A	0x00040CDC
0#	105 (0x0069)	N/A	N/A	0x00040B45

Module	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum	CPU	Subsystem	Sym
IESHIMS.DLL	Error opening file. The system cannot find the file specified (2).								
WER.DLL	Error opening file. The system cannot find the file specified (2).								
MPR.DLL	04/14/2008 3:00a	04/13/2008 4:10p	59,904	A	0x00013C87	0x00013C87	x86	Console	CV
ADVAPI32.DLL	04/22/2013 12:37a	04/22/2013 1:37a	618,496	A	0x000A48F8	0x000A48F8	x86	Console	CV
GDI32.DLL	10/09/2013 4:12a	10/09/2013 5:12a	287,744	A	0x00052131	0x00052131	x86	Console	CV
IERTUTIL.DLL	10/12/2013 10:24p	10/12/2013 11:24p	2,006,016	A	0x001EBA24	0x001EBA24	x86	GUI	CV
KERNEL32.DLL	03/12/2014 2:48a	03/12/2014 2:48a	993,880	A	0x000F63D5	0x000F63D5	x86	Console	CV
MSVCRT.DLL	10/29/2008 12:48a	10/29/2008 1:48a	343,040	A	0x000605CC	0x000605CC	x86	GUI	CV
MYHACK.DLL	01/23/2019 9:49p	01/23/2019 9:48p	75,264	A	0x00000000	0x0001AF14	x86	GUI	CV,I
NTDLL.DLL	12/09/2010 6:15a	12/09/2010 7:15a	718,336	A	0x000B9EE6	0x000B9EE6	x86	Console	CV
OLE32.DLL	08/05/2013 4:30a	08/05/2013 5:30a	1,289,728	A	0x00148858	0x00148858	x86	Console	CV

Warning: At least one delay-load dependency module was not found.

For Help, press F1

Notepad.exe Process

.text

.data

.rsrc

kernel32.dll

user32.dll

gdi32.dll

shell32.dll

advapi32.dll

ntdll32.dll

Common DLLs

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

Notepad.exe Process

.text

.data

.rsrc

kernel32.dll

user32.dll

gdi32.dll

shell32.dll

advapi32.dll

ntdll32.dll

Exploring Dynamically Linked Functions with Dependency Walker

Dependency Walker - [myhack.dll]

File Edit View Options Profile Window Help

MYHACK.DLL

- KERNEL32.DLL
- URLMON.DLL
- MSVCRT.DLL
- NTDLL.DLL
- OLE32.DLL
- OLEAUT32.DLL
- RPCRT4.DLL
- SHLWAPI.DLL
- ADVAPI32.DLL
- GDI32.DLL
- KERNEL32.DLL
- MSVCRT.DLL
- USER32.DLL

Module	File Time Stamp	Link Time Stamp	File Size
IESHIMS.DLL	Error opening file. The system cannot find the file specified		
WER.DLL	Error opening file. The system cannot find the file specified		
MPR.DLL	04/14/2008 3:00a	04/13/2008 4:10p	59,904
ADVAPI32.DLL	04/22/2013 12:37a	04/22/2013 1:37a	618,496
GDI32.DLL	10/09/2013 4:12a	10/09/2013 5:12a	287,744
IERTUTIL.DLL	10/12/2013 10:24p	10/12/2013 11:24p	2,006,016
KERNEL32.DLL	03/12/2014 2:48a	03/12/2014 2:48a	993,880
MSVCRT.DLL	10/29/2008 12:48a	10/29/2008 1:48a	343,040
MYHACK.DLL	01/23/2019 9:49p	01/23/2019 9:48p	75,264
NTDLL.DLL	12/09/2010 6:15a	12/09/2010 7:15a	718,336
OLE32.DLL	08/05/2013 4:30a	08/05/2013 5:30a	1,289,728

Warning: At least one delay-load dependency module was not found.
For Help, press F1

myhack.cpp > No Selection

```
1 #include "windows.h"
2 #include "tchar.h"
3
4 #pragma comment(lib, "urlmon.lib")
5
6 #define DEF_URL (L"http://www.naver.com/index.html")
```

`#pragma comment` is a compiler directive which indicates Visual C++ to leave a comment in the generated object file. The comment can then be read by the linker when it processes object files.

`#pragma comment(lib, libname)` tells the linker to add the 'libname' library to the list of library dependencies, as if you had added it in the project properties at **Linker->Input->Additional dependencies**

See [#pragma comment](#) on MSDN

```
19 if( !p )
20     return FALSE;
21
22 _tcsncpy_s(p+1, _MAX_PATH, DEF_FILE_NAME);
23
24 URLDownloadToFile(NULL, DEF_URL, szPath, 0, NULL);
25
26 return 0;
27 }
28
29 BOOL WINAPI DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
30 {
31     HANDLE hThread = NULL;
32
33     g_hMod = (HMODULE)hinstDLL;
34
35     switch( fdwReason )
36     {
37     case DLL_PROCESS_ATTACH :
38         OutputDebugString(L"<myhack.dll> Injection!!! -- CSC 497/583 -- Dr. Chen");
39         hThread = CreateThread(NULL, 0, ThreadProc, NULL, 0, NULL);
40         CloseHandle(hThread);
41         break;
42     }
43
44     return TRUE;
45 }
```

Q & A

