

Exercise Task 2

Part 1: Normal programs

- ▶ Choose 5 different normal programs and use Process Monitor and Process Explorer to get an overview of the events that occur during a normal startup procedure.
- ▶ Use Paint and Notepad to create two new files and save them to disk. Find the corresponding events in Process Monitor. Do these actions trigger loading of additional DLLs than those loaded at startup?



Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help



Time o...	Process Name	PID	Operation	Path	Result	Detail
12:16:2...	Calculator.exe	5656	Process Start		SUCCESS	Parent PID: 812, C...
12:16:3...	PaintStudio.View.exe	13328	Process Start		SUCCESS	Parent PID: 4232, ...
12:16:3...	iexplore.exe	2600	Process Start		SUCCESS	Parent PID: 5104, ...
12:16:3...	IEXPLORE.EXE	11104	Process Start		SUCCESS	Parent PID: 2600, ...
12:19:2...	mspaint.exe	10004	Process Start		SUCCESS	Parent PID: 5104, ...



Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time ...	Process Name	PID	Operation	Path	Result	Detail
12:16:2...	Calculator.exe	5656	Process Start		SUCCESS	Parent PID: 812, C...
12:16:2...	Calculator.exe	5656	Thread Create		SUCCESS	Thread ID: 14260
12:16:2...	Calculator.exe	5656	Load Image	C:\Program File...	SUCCESS	Image Base: 0x7ff...
12:16:2...	Calculator.exe	5656	Load Image	C:\Windows\Sys...	SUCCESS	Image Base: 0x7ff...
12:16:2...	Calculator.exe	5656	CreateFile	C:\Windows\Pre...	SUCCESS	Desired Access: G...
12:16:2...	Calculator.exe	5656	QueryStandardI...	C:\Windows\Pre...	SUCCESS	AllocationSize: 28...
12:16:2...	Calculator.exe	5656	ReadFile	C:\Windows\Pre...	SUCCESS	Offset: 0, Length: ...
12:16:2...	Calculator.exe	5656	ReadFile	C:\Windows\Pre...	SUCCESS	Offset: 0, Length: ...
12:16:2...	Calculator.exe	5656	CloseFile	C:\Windows\Pre...	SUCCESS	
12:16:2...	Calculator.exe	5656	RegOpenKey	HKEY\SYSTEM\...	REPARSE	Desired Access: Q...
12:16:2...	Calculator.exe	5656	RegOpenKey	HKEY\SYSTEM\...	NAME NOT FOUND	Desired Access: Q...
12:16:2...	Calculator.exe	5656	RegOpenKey	HKEY\SYSTEM\...	REPARSE	Desired Access: Q...
12:16:2...	Calculator.exe	5656	RegOpenKey	HKEY\System\...	SUCCESS	Desired Access: Q...
12:16:2...	Calculator.exe	5656	RegQueryValue	HKEY\System\...	NAME NOT FOUND	Length: 24
12:16:2...	Calculator.exe	5656	RegCloseKey	HKEY\System\...	SUCCESS	
12:16:2...	Calculator.exe	5656	RegOpenKey	HKEY\SYSTEM\...	REPARSE	Desired Access: Q...
12:16:2...	Calculator.exe	5656	RegOpenKey	HKEY\System\...	SUCCESS	Desired Access: Q...
12:16:2...	Calculator.exe	5656	RegQueryValue	HKEY\System\...	NAME NOT FOUND	Length: 24
12:16:2...	Calculator.exe	5656	RegCloseKey	HKEY\System\...	SUCCESS	
12:16:2...	Calculator.exe	5656	CreateFile	C:\Program File...	SUCCESS	Desired Access: E...
12:16:2...	Calculator.exe	5656	Load Image	C:\Windows\Sys...	SUCCESS	Image Base: 0x7ff...
12:16:2...	Calculator.exe	5656	Load Image	C:\Windows\Sys...	SUCCESS	Image Base: 0x7ff...
12:16:2...	Calculator.exe	5656	RegQueryValue	HKEY\System\...	NAME NOT FOUND	Length: 524
12:16:2...	Calculator.exe	5656	QueryNameInfo	C:\Windows\Sys...	SUCCESS	Name: Windows\S...
12:16:2...	Calculator.exe	5656	RegQueryValue	HKEY\System\...	NAME NOT FOUND	Length: 524
12:16:2...	Calculator.exe	5656	QueryNameInfo	C:\Windows\Sys...	SUCCESS	Name: Windows\S...
12:16:2...	Calculator.exe	5656	RegOpenKey	HKEY\System\...	REPARSE	Desired Access: Q...
12:16:2...	Calculator.exe	5656	RegOpenKey	HKEY\System\...	NAME NOT FOUND	Desired Access: Q...
12:16:2...	Calculator.exe	5656	RegOpenKey	HKEY\System\...	REPARSE	Desired Access: R
12:16:2...	Calculator.exe	5656	RegOpenKey	HKEY\System\...	NAME NOT FOUND	Desired Access: R
12:16:2...	Calculator.exe	5656	RegOpenKey	HKEY\Software\...	SUCCESS	Desired Access: Q...
12:16:2...	Calculator.exe	5656	RegQueryValue	HKEY\SOFTWA...	NAME NOT FOUND	Length: 80
12:16:2...	Calculator.exe	5656	RegCloseKey	HKEY\SOFTWA...	SUCCESS	
12:16:2...	Calculator.exe	5656	RegOpenKey	HKEY\Software\...	NAME NOT FOUND	Desired Access: Q...
12:16:2...	Calculator.exe	5656	RegOpenKey	HKEY\System\...	REPARSE	Desired Access: R
12:16:2...	Calculator.exe	5656	RegOpenKey	HKEY\System\...	SUCCESS	Desired Access: R
12:16:2...	Calculator.exe	5656	RegQueryValue	HKEY\System\...	SUCCESS	Type: REG_DWO...
12:16:2...	Calculator.exe	5656	RegCloseKey	HKEY\System\...	SUCCESS	
12:16:2...	Calculator.exe	5656	Load Image	C:\Windows\Sys...	SUCCESS	Image Base: 0x7ff...

Showing 6,289 of 863,843 events (0.72%)

Backed by virtual memory

Process Monitor Filter

Display entries matching these conditions:

Process Name	is	Calculator.exe	then
--------------	----	----------------	------

Reset Add

Column	Relation	Value	Action
<input checked="" type="checkbox"/> Process Na	is	Calculator.exe	Include
<input type="checkbox"/> Operation	is	Process Start	Include
<input checked="" type="checkbox"/> Process Na	is	Procmon.exe	Exclude
<input checked="" type="checkbox"/> Process Na	is	Procexp.exe	Exclude
<input checked="" type="checkbox"/> Process Na	is	Autoruns.exe	Exclude
<input checked="" type="checkbox"/> Process Na	is	Procmon64.exe	Exclude
<input checked="" type="checkbox"/> Process Na	is	Procexp64.exe	Exclude

OK Cancel

Time	Process Name	PID	Operation	Path
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\Prefetch\MSPAINT.EXE-76E10B24.pf
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\win32k\microsoft.windows.common-controls_6595b64144cc1df_6_0_7601_18837_none_41e85514
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\rtld.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\kernel32.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\KernelBase.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\ole32.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\oleacc.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\advapi32.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\escort.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\sechost.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\crypt.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\gdi32.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\user32.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\ole32.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\oleaut32.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\oleui32.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\olepk32.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\objc32.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\comdlg32.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\ehkraspi.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\win32k\microsoft.windows.common-controls_6595b64144cc1df_6_0_7601_18837_none_41e85514
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\shell32.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\oleprxy.dll
1:19:42	mapedit.exe	2676	CreateFile	C:\Windows\System32\oleprxy.dll

Time	Process Name	PID	Operation	Path
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\Prefetch\SNIPPING TOOL.EXE-EFFFDAFE pf
12:00	Snipping Tool.exe	3658	CreateFile	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessories
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\seachost.dll
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\seachost.dll
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\Snipping Tool.exe Local
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\win32\microsoft.windows.gdiplus_559964144ccfd1f_5_0.7601.23807_none_e02fa5e05e011fb02
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\win32\microsoft.windows.common-controls_559964144ccfd1f_5_0.7601.23807_none_e02fa5e05e011fb02
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\win32\microsoft.windows.gdiplus_559964144ccfd1f_1_1_7601.23807_none_e02fa5e05e011fb02
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\win32\microsoft.windows.common-controls_559964144ccfd1f_1_1_7601.23807_none_e02fa5e05e011fb02
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\Snipping Tool.exe Local
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\win32\microsoft.windows.common-controls_559964144ccfd1f_5_0.7601.18837_none_41e55142
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\win32\w8l\microsoft.windows.common-controls_559964144ccfd1f_5_0.7601.18837_none_41e55142
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\win32\w8l\microsoft.windows.common-controls_559964144ccfd1f_5_0.7601.18837_none_41e55142
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\atheme.dll
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\leacc.dll
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\leacc.dll
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\lecc.dll
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\lecc.dll
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\radm.dll
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\crystal.dll
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\crystal.dll
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\crystal.dll
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\imm32.dll
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\imm32.dll
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\imm32.dll
12:00	Snipping Tool.exe	3658	CreateFile	C:\Windows\System32\imm32.dll

Part 1: What to learn?

- ▶ Be able to recognize common behavior at startup.
- ▶ Use filters to see events from only one process in Process Monitor.
- ▶ Find the startup of a process in Process Monitor.
- ▶ Find the dll files loaded by a process in Process Explorer.

Part 2: Malicious programs

- ▶ Turn off Windows Defender in your VM (and keep it off).
- ▶ Download the labs from the book's website:
<https://practicalmalwareanalysis.com/labs/>
- ▶ Do the labs 3-1 and 3-3. Note that these two pieces of malware are written specifically for XP, so they will not work exactly as in the solutions described in the book.

Turn off Windows Defender

To turn it off in GUI only works for a while. Alternatives:

- ▶ Open regedit as admin
- ▶ Go to HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\WindowsDefender
- ▶ If there is a key named DisableAntiSpyware, set it to 1 and reboot. You're finished.
- ▶ If not, right click the Windows Defender folder in the folder hierarchy.
- ▶ New → DWORD (32-bit), set the name to DisableAntiSpyware and value to 1. Reboot

With group policy:

- ▶ Open gpedit.msc
- ▶ Computer configuration → Administrative templates → windows components → Windows Defender Antivirus
- ▶ Find the item “Turn off Windows Defender Antivirus” and set it to Enabled.

Or go to *Virus and threat protection settings* and add the Labs folder to Exclusions.

Lab 3-1

Examine the executable Lab03-01.exe using basic dynamic analysis.
Use Process Monitor, Process Explorer (and Regshot).



Process Explorer - Sysinternals: www.sysinternals.com [MSEdgeWIN10\IEUser]



File Options View Process Find DLL Users Help

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
010Editor.exe		47,436 K	37,492 K	11024	010 Editor - Professional Tex...	SweetScape Software
Procmon.exe		3,116 K	20,828 K	7568	Process Monitor	Sysinternals - www.sysinte...
Procmon64.exe	27.71	19,444 K	67,840 K	14408	Process Monitor	Sysinternals - www.sysinte...
iexplore.exe	0.01	9,416 K	36,900 K	2600	Internet Explorer	Microsoft Corporation
iexplore.exe	< 0.01	13,356 K	32,436 K	11104	Internet Explorer	Microsoft Corporation
PracticalMalwareAnalysis-Lab...	0.28	9,172 K	30,272 K	4116		
procexp64.exe	5.97	27,228 K	51,888 K	1496	Sysinternals Process Explorer	Sysinternals - www.sysinte...
cmd.exe		4,672 K	3,444 K	356	Windows Command Processor	Microsoft Corporation
conhost.exe	0.67	6,344 K	22,928 K	11888	Console Window Host	Microsoft Corporation
Lab03-01.exe	< 0.01	1,280 K	5,796 K	11976		
Lab03-01.exe	Susp...	384 K	148 K	14144		
WerFault.exe	19.14	3,612 K	11,124 K	12368	Windows Problem Reporting	Microsoft Corporation
jusched.exe		1,528 K	6,380 K	7900	Java Update Scheduler	Oracle Corporation



Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time ...	Process Name	PID	Operation	Path	Result	Detail
12:33:3...	Lab03-01.exe	11976	Process Start		SUCCESS	Parent PID: 356 C...
12:33:3...	Lab03-01.exe	11976	Thread Create		SUCCESS	Thread ID: 14236
12:33:3...	Lab03-01.exe	11976	Load Image	C:\Users\IEUser\... SUCCESS		Image Base: 0x40...
12:33:3...	Lab03-01.exe	11976	Load Image	C:\Windows\Sys... SUCCESS		Image Base: 0x7f...
12:33:3...	Lab03-01.exe	11976	Load Image	C:\Windows\Sys... SUCCESS		Image Base: 0x77...
12:33:3...	Lab03-01.exe	11976	RegOpenKey	HKLMSYSTEM... REPARSE		Desired Access: Q...
12:33:3...	Lab03-01.exe	11976	RegOpenKey	HKLMSYSTEM... NAME NOT FOUND	Desired Access: Q...	
12:33:3...	Lab03-01.exe	11976	RegOpenKey	HKLMSYSTEM... REPARSE		Desired Access: Q...
12:33:3...	Lab03-01.exe	11976	RegOpenKey	HKLMSYSTEM... SUCCESS		Desired Access: Q...
12:33:3...	Lab03-01.exe	11976	RegQueryValue	HKLMSYSTEM... NAME NOT FOUND	Length: 24	
12:33:3...	Lab03-01.exe	11976	RegCloseKey	HKLMSYSTEM... SUCCESS		
12:33:3...	Lab03-01.exe	11976	CreateFile	C:\Windows	SUCCESS	Desired Access: E...
12:33:3...	Lab03-01.exe	11976	Load Image	C:\Windows\Sys...	SUCCESS	Image Base: 0x77...
12:33:3...	Lab03-01.exe	11976	Load Image	C:\Windows\Sys...	SUCCESS	Image Base: 0x77...
12:33:3...	Lab03-01.exe	11976	CreateFile	C:\Windows	SUCCESS	Desired Access: R...
12:33:3...	Lab03-01.exe	11976	QueryNameInfo	C:\Windows	SUCCESS	Name: \Windows
12:33:3...	Lab03-01.exe	11976	CloseFile	C:\Windows	SUCCESS	
12:33:3...	Lab03-01.exe	11976	RegOpenKey	HKLMSOFTWA...	SUCCESS	Desired Access: R
12:33:3...	Lab03-01.exe	11976	RegQueryValue	HKLMSOFTWA... NAME NOT FOUND	Length: 520	
12:33:3...	Lab03-01.exe	11976	RegCloseKey	HKLMSOFTWA...	SUCCESS	Type: REG_SZ, Le...
12:33:3...	Lab03-01.exe	11976	Load Image	C:\Windows\Sys...	SUCCESS	Image Base: 0x77...
12:33:3...	Lab03-01.exe	11976	RegOpenKey	HKLMSYSTEM...	REPARSE	Desired Access: Q...
12:33:3...	Lab03-01.exe	11976	RegOpenKey	HKLMSYSTEM... REPARSE		Desired Access: Q...
12:33:3...	Lab03-01.exe	11976	RegOpenKey	HKLMSYSTEM...	SUCCESS	Desired Access: Q...
12:33:3...	Lab03-01.exe	11976	RegSetInfoKey	HKLMSYSTEM...	SUCCESS	KeySetInformation...
12:33:3...	Lab03-01.exe	11976	RegQueryValue	HKLMSYSTEM... NAME NOT FOUND	Length: 24	
12:33:3...	Lab03-01.exe	11976	RegCloseKey	HKLMSYSTEM...	SUCCESS	
12:33:3...	Lab03-01.exe	11976	CreateFile	C:\Users\IEUser\...	SUCCESS	Desired Access: E...
12:33:3...	Lab03-01.exe	11976	Load Image	C:\Windows\Sys...	SUCCESS	Image Base: 0x74...
12:33:3...	Lab03-01.exe	11976	Load Image	C:\Windows\Sys...	SUCCESS	Image Base: 0x75...
12:33:3...	Lab03-01.exe	11976	RegQueryValue	HKLMSYSTEM... NAME NOT FOUND	Length: 524	
12:33:3...	Lab03-01.exe	11976	QueryNameInfo	C:\Windows\Sys...	SUCCESS	Name: \Windows\S...
12:33:3...	Lab03-01.exe	11976	RegQueryValue	HKLMSYSTEM... NAME NOT FOUND	Length: 524	

Process Monitor Filter

Display entries matching these conditions:

Operation	is	Process Create	then
-----------	----	----------------	------

Column	Relation	Value	Action
<input checked="" type="checkbox"/> Process Na	is	calculator.exe	Include
<input checked="" type="checkbox"/> Process Na	is	iexplore.exe	Include
<input checked="" type="checkbox"/> PID	is	11976	Include
<input type="checkbox"/> Operation	is	Process Start	Include
<input type="checkbox"/> Operation	is	Thread Create	Include
<input type="checkbox"/> Operation	is	Process Create	Include
<input checked="" type="checkbox"/> Process Na	is	Procmon.exe	Exclude

Showing 476 of 536,238 events (0.088%)

Backed by virtual memory



Lab 3-1: What to learn?

- ▶ Identify the processes started by an executable.
- ▶ Use filters in Process Monitor to analyze all relevant processes.
- ▶ WerFault is started because a problem is detected and may be reported.
- ▶ Identify files that are written to and changes in the registry using Regshot or filters in Process Monitor.

Lab 3-3

Examine the executable Lab03-03.exe using basic dynamic analysis.
Use Process Monitor, Process Explorer (and Regshot).

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time o...	Process Name	PID	Operation	Path	Result	Detail
1:03:55...	Lab03-03.exe	10008	Process Start		SUCCESS	
1:03:55...	Lab03-03.exe	10008	Thread Create		SUCCESS	
1:03:55...	Lab03-03.exe	10008	Load Image	C:\Users\lEUser\Documents\lits\Pract...	SUCCESS	
1:03:55...	Lab03-03.exe	10008	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS	
1:03:55...	Lab03-03.exe	10008	Load Image	C:\Windows\SysWOW64\ntdll.dll	SUCCESS	
1:03:55...	Lab03-03.exe	10008	CreateFile	C:\Windows\Prefetch\LAB03-03.EXE...	NAME NOT FOUND	
1:03:55...	Lab03-03.exe	10008	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Co...	REPARSE	
1:03:55...	Lab03-03.exe	10008	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Co...	REPARSE	
1:03:55...	Lab03-03.exe	10008	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Co...	SUCCESS	
1:03:55...	Lab03-03.exe	10008	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Co...	NAME NOT FOUND	
1:03:55...	Lab03-03.exe	10008	RegCloseKey	HKEY\SYSTEM\CurrentControlSet\Co...	SUCCESS	
1:03:55...	Lab03-03.exe	10008	CreateFile	C:\Windows	SUCCESS	
1:03:55...	Lab03-03.exe	10008	Load Image	C:\Windows\System32\wow64.dll	SUCCESS	
1:03:55...	Lab03-03.exe	10008	Load Image	C:\Windows\System32\wow64win.dll	SUCCESS	
1:03:55...	Lab03-03.exe	10008	CreateFile	C:\Windows\System32\wow64log.dll	NAME NOT FOUND	
1:03:55...	Lab03-03.exe	10008	Load Image	C:\Windows\System32\kernel32.dll	SUCCESS	
1:03:55...	Lab03-03.exe	10008	Load Image	C:\Windows\SysWOW64\kernel32.dll	SUCCESS	
1:03:55...	Lab03-03.exe	10008	Load Image	C:\Windows\System32\user32.dll	SUCCESS	
1:03:55...	Lab03-03.exe	10008	CreateFile	C:\Windows	SUCCESS	
1:03:55...	Lab03-03.exe	10008	QueryNameInfo...	C:\Windows	SUCCESS	
1:03:55...	Lab03-03.exe	10008	CloseFile	C:\Windows	SUCCESS	
1:03:55...	Lab03-03.exe	10008	RegOpenKey	HKEY\Software\Microsoft\Wow64\x86...	SUCCESS	
1:03:55...	Lab03-03.exe	10008	RegQueryValue	HKEY\SOFTWARE\Microsoft\Wow64... NAME NOT FOUND		
1:03:55...	Lab03-03.exe	10008	RegQueryValue	HKEY\SOFTWARE\Microsoft\Wow64... SUCCESS		
1:03:55...	Lab03-03.exe	10008	RegCloseKey	HKEY\SOFTWARE\Microsoft\Wow64... SUCCESS		
1:03:55...	Lab03-03.exe	10008	Load Image	C:\Windows\System32\wow64cpu.dll	SUCCESS	
1:03:55...	Lab03-03.exe	10008	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Co...	REPARSE	
1:03:55...	Lab03-03.exe	10008	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Co...	NAME NOT FOUND	
1:03:55...	Lab03-03.exe	10008	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Co...	REPARSE	
1:03:55...	Lab03-03.exe	10008	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Co...	SUCCESS	
1:03:55...	Lab03-03.exe	10008	RegSetInfoKey	HKEY\SYSTEM\CurrentControlSet\Co...	SUCCESS	
1:03:55...	Lab03-03.exe	10008	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Co...	NAME NOT FOUND	Length: 24
1:03:55...	Lab03-03.exe	10008	RegCloseKey	HKEY\SYSTEM\CurrentControlSet\Co...	SUCCESS	
1:03:55...	Lab03-03.exe	10008	CreateFile	C:\Users\lEUser\Documents\lits\Pract...	SUCCESS	Desired Access: E...
1:03:55...	Lab03-03.exe	10008	Load Image	C:\Windows\SysWOW64\kernel32.dll	SUCCESS	Image Base: 0x74...
1:03:55...	Lab03-03.exe	10008	Load Image	C:\Windows\SysWOW64\KernelBas...	SUCCESS	Image Base: 0x75...
1:03:55...	Lab03-03.exe	10008	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Co...	NAME NOT FOUND	Length: 524
1:03:55...	Lab03-03.exe	10008	QueryNameInfo...	C:\Windows\SysWOW64\KernelBas...	SUCCESS	Name: \Windows\IS

Showing 346 of 424,854 events (0.081%)

Backed by virtual memory

Process Monitor Filter

Display entries matching these conditions:

Process Name is Lab03-03.exe then Include

Reset

Add

Remove

Column	Relation	Value	Action
<input checked="" type="checkbox"/> Process Na	is	Lab03-03.exe	Include
<input checked="" type="checkbox"/> PID	is	11976	Include
<input type="checkbox"/> Operation	is	Process Start	Include
<input type="checkbox"/> Operation	is	Thread Create	Include
<input type="checkbox"/> Operation	is	Process Create	Include
<input type="checkbox"/> Operation	is	RegSetValue	Include
<input checked="" type="checkbox"/> Process Na	is	Procmon.exe	Exclude

OK

Cancel

Apply

Image Base: 0x77...

Desired Access: Q...

Desired Access: Q...

Desired Access: Q...

Desired Access: Q...

KeySetInformation...

Length: 24



Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time o... Process Name
1:03:55... Lab03-03.exe

PID Operation Path Result Detail
10008 Process Create C:\Windows\SysWOW64\svchost.exe SUCCESS
PID: 12508, Comm...

Process Monitor Filter

Display entries matching these conditions:

Process Name is Lab03-03.exe then Include

Reset Add Remove

Column	Relation	Value	Action
<input checked="" type="checkbox"/> Process Na	is	Lab03-03.exe	Include
<input type="checkbox"/> PID	is	11976	Include
<input type="checkbox"/> Operation	is	Process Start	Include
<input type="checkbox"/> Operation	is	Thread Create	Include
<input checked="" type="checkbox"/> Operation	is	Process Create	Include
<input type="checkbox"/> Operation	is	RegSetValue	Include
<input checked="" type="checkbox"/> Process Na	is	Procmon.exe	Exclude

OK Cancel Apply

Process Explorer - Sysinternals: www.sysinternals.com [MSEdgeWIN10\IEUser]

File Options View Process Find DLL Users Help

Process CPU Private Bytes Working Set PID Description Company Name

cmd.exe	2,216 K	1,912 K	6932	Windows Command Processor	Microsoft Corporation
conhost.exe	6,144 K	13,124 K	1052	Console Window Host	Microsoft Corporation
010Editor.exe	47,496 K	26,412 K	11024	010 Editor - Professional Text...	SweetScape Software
Procmon.exe	2,968 K	15,928 K	7568	Process Monitor	Sysinternals - www.sysinte...
Procmemory.exe	4.31	18,820 K	14408	Process Monitor	Sysinternals - www.sysinte...
procexp64.exe	17.55	27,184 K	1496	Sysinternals Process Explorer	Sysinternals - www.sysinte...
cmd.exe	3,672 K	3,240 K	356	Windows Command Processor	Microsoft Corporation
conhost.exe	0.09	6,268 K	11888	Console Window Host	Microsoft Corporation
Lab03-03.exe		612 K	3,060 K	10008	
svchost.exe	Susp...	560 K	424 K	12508 Host Process for Windows S...	Microsoft Corporation
svchost.exe		248 K	52 K	12212 Host Process for Windows S...	Microsoft Corporation
WerFault.exe	< 0.01	3,652 K	9,672 K	7988 Windows Problem Reporting	Microsoft Corporation
jusched.exe		1,528 K	5,324 K	7900 Java Update Scheduler	Oracle Corporation

Lab 3-3: What to learn?

- ▶ A familiar process name can be used to hide.
- ▶ When the parent process dies, the child process is orphaned, making it more difficult to see where it came from.
- ▶ Normal svchost processes are children of services.exe, not orphaned.
- ▶ Do not trust a process just because it has a familiar name. If its location in the process hierarchy, its filepath or anything else is unusual, check it out.

Part 3: Reversing with IDA

Do parts of Lab 5-1: Tasks 1-16

Part 3: What to learn

- ▶ Learn to navigate in IDA.
- ▶ Learn to find the types of information concerned in these tasks.
- ▶ Understand basic assembly. Example: How was the value put in eax? What are the parameters of the function?
- ▶ Be able to interpret IDA screenshots similar to the screenshots and relevant functions in this exercise.

1. What is the address of DllMain?

DllEntryPoint0x1001516D in the .text section

IDA - Lab05-01.dll C:\Users\jUser\Documents\its\PracticalMalwareAnalysis-Labs\Practical Malware Analysis Labs\BinaryCollection\Chapter_5L\Lab05-01.dll

File Edit Jump Search View Options Windows Help

Functions window

IDA View-A Data Hex View-1 Structures Enums Imports Exports

Function name

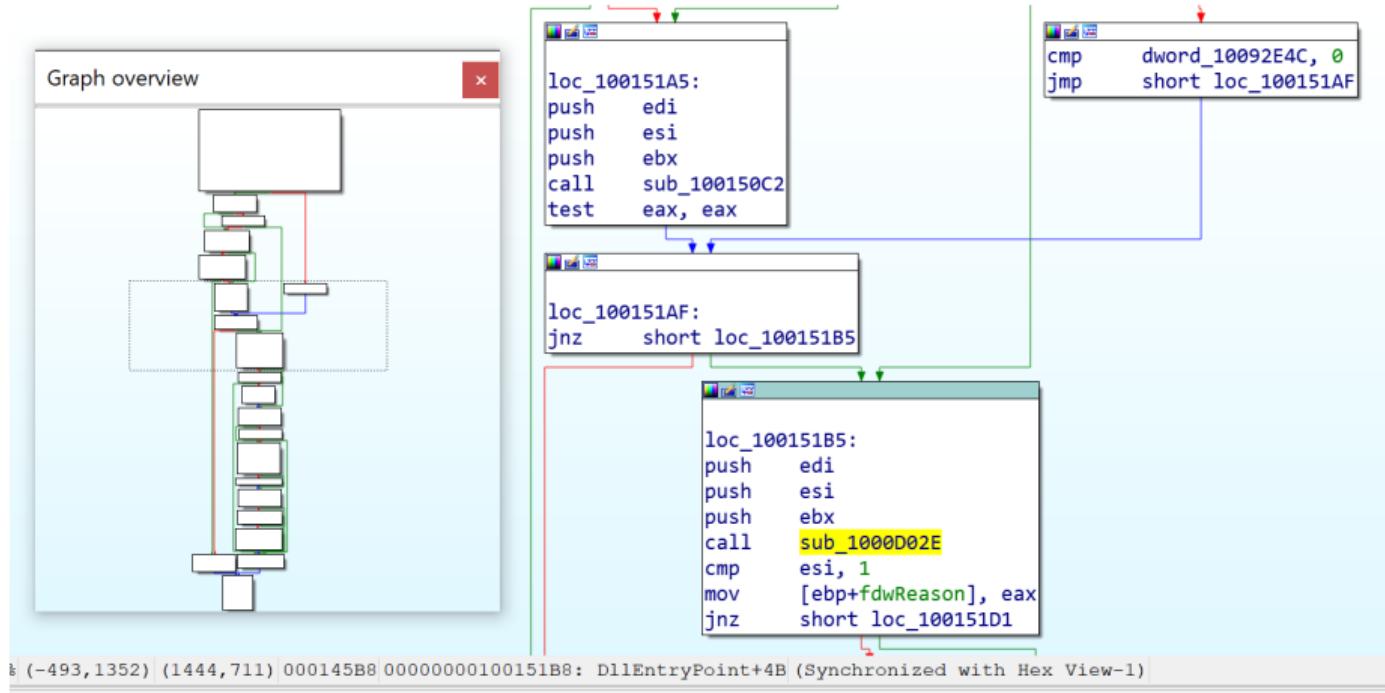
- sub_10014E90
- sub_10014F10
- memcpy
- strcpy
- strlen
- memset
- memcmp
- printf
- _alloca_probe
- strcat
- sub_10014F80
- strcmp
- operator delete(void *)
- operator new(uint)
- sub_10015030
- ftol
- sub_1001505C
- sub_10015088
- abs
- sub_100150A0
- _CxxThrowException
- sub_100150C2
- DllEntryPoint
- wcslen
- _Clacos
- _Clipow
- _dlloexit
- type_info::~type_info(void)
- __tmain

.text:1001516D public DllEntryPoint
.text:1001516D proc near
.text:1001516D hinstDLL = dword ptr 8
.text:1001516D fdwReason = dword ptr 0Ch
.text:1001516D lpReserved = dword ptr 10h
.text:1001516D push ebp
.text:1001516E mov ebp, esp
.text:10015170 push ebx
.text:10015171 mov ebx, [ebp+hinstDLL]
.text:10015174 push esi
.text:10015175 mov esi, [ebp+fdwReason]
.text:10015178 push edi
.text:10015179 mov edi, [ebp+lpReserved]
.text:1001517C test esi, esi
.text:1001517E jnz short loc_10015189
.text:10015180 cmp dword_10092E4C, 0
.text:10015187 jmp short loc_100151AF
.text:10015189 ;
.text:10015189 loc_10015189: ; CODE XREF: DllEntryPoint+11fj
.text:10015189 cmp esi, 1
.text:1001518C jz short loc_10015193
.text:1001518E cmp esi, 2
.text:10015191 jnz short loc_10015185
.text:10015193 loc_10015193: ; CODE XREF: DllEntryPoint+1F1fj

Line 332 of 348

0001456D 000000001001516D: DllEntryPoint (Synchronized with Hex View-1)

1. What is the address of DllMain?



1. What is the address of DllMain?

0x1000D02E in the .text section

IDA - Lab05-01.dll C:\Users\IEUser\Documents\its\PracticalMalwareAnalysis-Labs\Practical Malware Analysis Labs\BinaryCollection\Chapter_5L\Lab05-01.dll

File Edit Jump Search View Options Windows Help

Functions window

Function name

- sub_10014F90
- sub_10014F10
- memcpy
- strcpy
- strlen
- memset
- memcmp
- printf
- _alloca_probe
- strcat
- sub_10014FB0
- strcmp
- operator delete(void *)
- operator new(uint)
- sub_10015030
- fopen
- sub_1001505C
- sub_10015088
- abs
- sub_100150A0
- _CxxThrowException
- sub_100150C2
- DllEntryPoint
- wcslen
- _Clacos
- _Cipow
- _dlopenexit
- type_info::~type_info(void)
- _initterm

Libary function Regular function Instruction External symbol

IDA View-A Hex View-1 Structures Enums Imports Exports

```
.text:1000D02E
.text:1000D02E
.text:1000D02E ; int __cdecl sub_1000D02E(int argc, const char **argv, const char **envp)
.text:1000D02E sub_1000D02E proc near ; CODE XREF: DllEntryPoint+4B↑
.text:1000D02E
.text:1000D02E
.text:1000D02E argc = dword ptr 4
.text:1000D02E argv = dword ptr 8
.text:1000D02E envp = dword ptr 0Ch
.text:1000D02E
.text:1000D02E mov eax, [esp+argv]
.text:1000D032 dec eax
.text:1000D033 jnz loc_10000107
.text:1000D039 mov eax, [esp+argc]
.push ebx
.text:1000D03E mov ds:hModule, eax
.text:1000D044 mov eax, off_10019044
.push esi
.text:1000D048 push eax, 0Dh
.text:1000D049 add eax, edi
.push edi
.text:1000D04C push eax
.call strlen ; Str
.text:1000D04E mov ebx, ds>CreateThread
.text:1000D053 mov esi, ds:_strnicmp
.text:1000D059 mov edi, esi
.xor edi, edi
.pop ecx
.test eax, eax
.jz short loc_1000D089
```

Line 332 of 348

0000C42E 000000001000D02E: sub_1000D02E (Synchronized with Hex View-1)

2. Use the Imports window to browse to gethostbyname. Where is the import located?

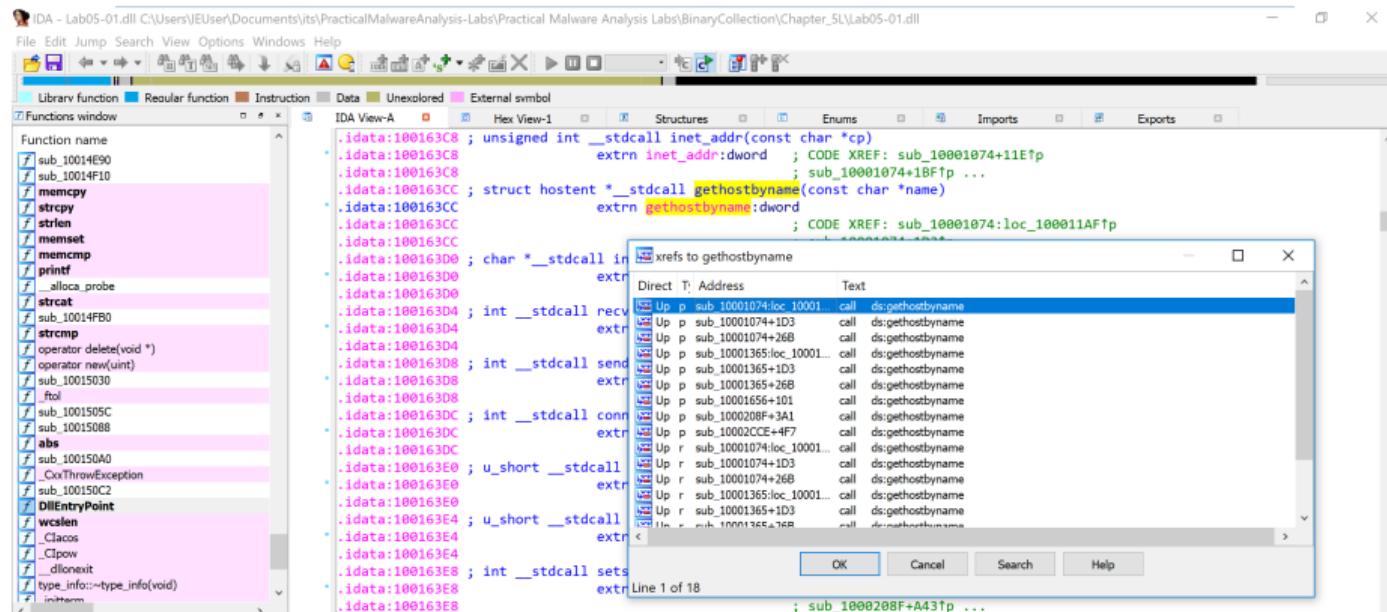
Start writing *gethostbyname* while in the Imports window.

0x100163CC in the .idata section

3. How many functions call gethostbyname?

Double click in the Imports window. Press X while at the name.

It is called nine times, by five different functions.



4. Focusing on the call to gethostbyname at 0x10001757, which DNS request will be made?

pics.practicalmalwareanalysis.com

```
.text:1000173A          test    eax, eax
.text:1000173C          jz      loc_100017ED
.text:10001742          cmp     dword_1008E5CC, ebx
.text:10001748          jnz     loc_100017ED
.text:1000174E          mov     eax, off_10019040
.text:10001753          add     eax, 0Dh
.text:10001756          push    eax      ; name
.text:10001757          call    ds:gethostbyname
.text:1000175D          mov     esi, eax
.text:1000175F          cmp     esi, ebx
.text:10001761          jz      short loc_100017C0
.text:10001763          movsx   eax, word ptr [esi+0Ah]
.text:10001767          push    eax      ; Size
.data:10019040 off_10019040  dd offset aThisIsRdoPicsP
.data:10019040           ; DATA XREF: sub_10001656:loc_10001722r
.data:10019040           ; sub_10001656+F8r ...
.data:10019040           ; "[This is RDO]pics.practicalmalwareanalys"...
```

IDA View-A

• .data:1001918F	db	0
• .data:10019190	db	0
• .data:10019191	db	20h
• .data:10019192	db	0
• .data:10019193	db	0
• .data:10019194 alhisIsRdoPics_	db	[This is RDO]pics.practicalmalwareanalysis.com',0
• .data:10019194	db	; DATA XREF: .data:off_10019040t0
• .data:100191C2	db	0
• .data:100191C3	db	0
• .data:100191C4	db	0
• .data:100191C5	db	0
• .data:100191C6	db	0
• .data:100191C7	db	0

Consider the subroutine at 0x10001656

Press G and enter the address.

5. How many local variables has IDA recognized?

23, those with negative offsets. (Exact number may vary between versions.)

6. How many parameters has IDA recognized?

1, lpThreadParameter, which has positive offset

```
.text:10001656 ; DWORD stdcall sub_10001656(LPVOID lpThreadParameter)
.text:10001656 sub_10001656 proc near ; DATA XREF: sub_1000D02E+C8↓o
.text:10001656
.text:10001656 var_675 = byte ptr -675h
.text:10001656 var_674 = dword ptr -674h
.text:10001656 hModule = dword ptr -670h
.text:10001656 timeout = timeval ptr -66Ch
.text:10001656 name = sockaddr ptr -664h
.text:10001656 var_654 = word ptr -654h
.text:10001656 Dst = dword ptr -650h
.text:10001656 Str1 = byte ptr -644h
.text:10001656 var_640 = byte ptr -640h
.text:10001656 CommandLine = byte ptr -63Fh
.text:10001656 Str = byte ptr -63Dh
.text:10001656 var_638 = byte ptr -638h
.text:10001656 var_637 = byte ptr -637h
.text:10001656 var_544 = byte ptr -544h
.text:10001656 var_50C = dword ptr -50Ch
.text:10001656 var_500 = byte ptr -500h
.text:10001656 Buf2 = byte ptr -4FCh
.text:10001656 readfds = fd_set ptr -48Ch
.text:10001656 buf = byte ptr -388h
.text:10001656 var_3B0 = dword ptr -3B0h
.text:10001656 var_1A4 = dword ptr -1A4h
.text:10001656 var_194 = dword ptr -194h
.text:10001656 WSAData = WSADATA ptr -190h
.text:10001656 lpThreadParameter= dword ptr 4
.text:10001656
```

00000A56 000000000010001656: sub_10001656 (Synchronized with Mex View-1)

7. Use the Strings window to locate the string \cmd.exe /c in the disassembly. Where is it located?

View - Open subviews - Strings (Shift+F12)

Start writing the string to locate it. Note that the string is actually starting with \\.

Alternatively: Click the String column to sort alphabetically.

The string is located in the xdoors_d section.

Address	Length	Type	String
xdoors_d:0t.....	/TAAAAA	C	Now Run UninstallSA %s
xdoors_d:10... 00000016	C	Get ServiceName-> "%s"	
xdoors_d:10... 00000016	C	Get ProcessName-> "%s"	
xdoors_d:10... 00000015	C	Get ModuleName-> "%s"	
xdoors_d:10... 00000015	C	Get ModulePath-> "%s"	
xdoors_d:10... 00000021	C	\V\NGet Install Way->InstallIRTy\r\n\r\n	
xdoors_d:10... 00000021	C	\V\NGet Install Way->InstallPE\r\n\r\n	
xdoors_d:10... 0000002F	C	\V\NGet Install Way->InstallSB Or InstallRSB\r\n\r\n	
xdoors_d:10... 00000021	C	\V\NGet Install Way->InstallSA\r\n\r\n	
xdoors_d:10... 00000018	C	\V\NGet ServiceName-> "%s"	
xdoors_d:10... 00000018	C	\V\NGet ProcessName-> "%s"	
xdoors_d:10... 00000017	C	\V\NGet ModuleName-> "%s"	
xdoors_d:10... 00000017	C	\V\NGet ModulePath-> "%s"	
xdoors_d:10... 00000029	C	CreateProcess() GetLastError reports %d\r\n	
xdoors_d:10... 00000007	C	inject	
xdoors_d:10... 00000009	C	minstall	
xdoors_d:10... 00000008	C	mmodule	
xdoors_d:10... 00000006	C	mhost	
xdoors_d:10... 00000006	C	mbase	
xdoors_d:10... 0000000A	C	robotwork	
xdoors_d:10... 00000009	C	language	
xdoors_d:10... 00000007	C	uptime	
xdoors_d:10... 00000005	C	idle	
xdoors_d:10... 0000000F	C	\V\N\r\r\n0x%02x\r\n\r\n	
xdoors_d:10... 00000008	C	enmagic	
xdoors_d:10... 00000005	C	exit	
xdoors_d:10... 00000005	C	quit	
xdoors_d:10... 00000011	C	\Command.exe /c	
xdoors_d:10... 0000000D	C	\cmd.exe /c	
xdoors_d:10... 00000118	C	Hi,Master [%d/%d/%d %d:%d:%d]\r\nWelCome Back...Are You Enjoying Today?\r\n\r...	

8. What is happening in the area of code that references \cmd.exe /c?

Double click the string, press X to find cross references.

It appears to set up a remote shell connection. We see calls like *CreatePipe* and *recv*.

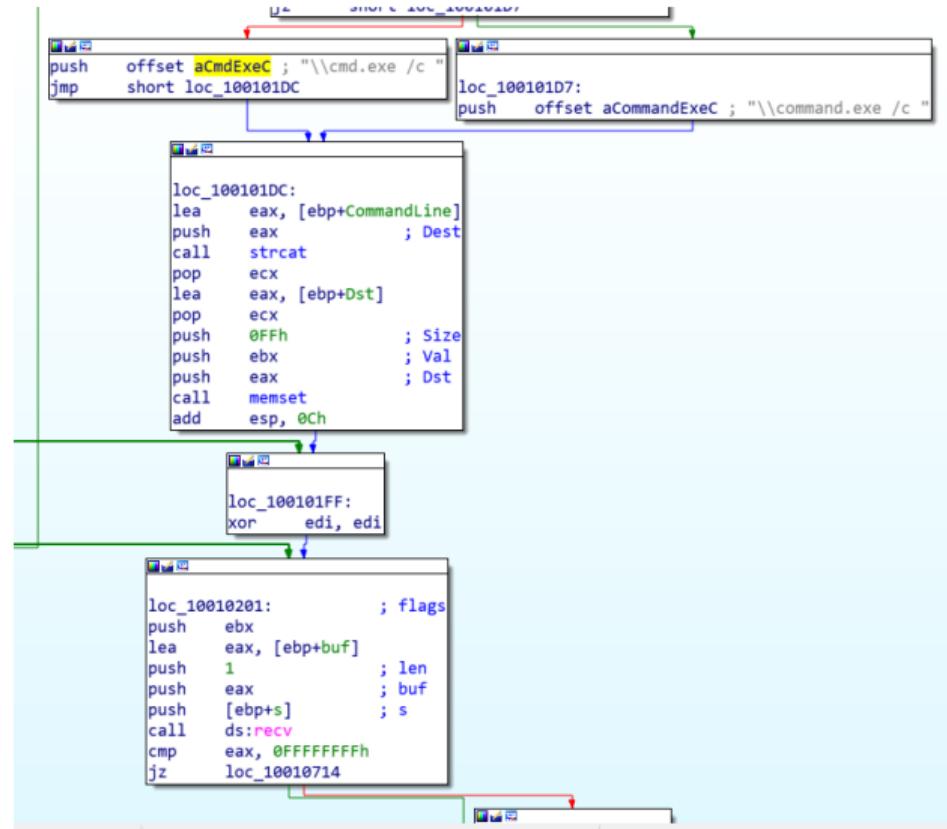
```
.text:1001016A    push    eax          ; hWritePipe
.text:1001016B    lea     eax, [ebp+hReadPipe]
.text:1001016E    push    eax          ; hReadPipe
.text:1001016F    mov     [ebp+NumberOfBytesRead], ebx
.text:10010172    mov     [ebp+PipeAttributes.nLength], 0Ch
.text:10010179    mov     [ebp+PipeAttributes.lpSecurityDescriptor], ebx
.text:1001017C    mov     [ebp+PipeAttributes.bInheritHandle], 1
.text:10010183    call    ds:CreatePipe
.text:10010189    test   eax, eax
.text:10010188    jz     loc_10010714
.text:10010191    lea     eax, [ebp+StartupInfo]
.text:10010194    mov     [ebp+StartupInfo.cb], 44h
.text:10010198    push   eax          ; lpStartupInfo
.text:1001019C    call    ds:GetStartupInfoA
.text:100101A2    mov     eax, [ebp+hWritePipe]
.text:100101A5    push   400h          ; uSize
.text:100101AA    mov     [ebp+StartupInfo.hStdError], eax
.text:100101AD    mov     [ebp+StartupInfo.hStdOutput], eax
.text:100101B0    lea     eax, [ebp+CommandLine]
.text:100101B6    mov     [ebp+StartupInfo.wShowWindow], bx
.text:100101BA    push   eax          ; lpBuffer
.text:100101BB    mov     [ebp+StartupInfo.dwFlags], 101h
.text:100101C2    call    ds:GetSystemDirectoryA
.text:100101C8    cmp    dword_1008ESC4, ebx
.text:100101CE    jz     short loc_100101D7
.text:100101D0    push   offset aCmdExeC ; "\\\cmd.exe / c "
.text:100101D5    jmp    short loc_100101DC
.text:100101D7 ;
```

0000F572 0000000010010172: sub_1000FF58+21A (Synchronized with Hex View-1)

8. What is happening in the area of code that references \cmd.exe /c?

Double click the string, press X to find cross references.

It appears to set up a remote shell connection. We see calls like *CreatePipe* and *recv*.



8. What is happening in the area of code that references \cmd.exe /c?

At the beginning of the same function, we see a string being used.

The screenshot shows a debugger interface with two main panes. The left pane displays assembly code for a function, likely a shellcode generator. The right pane shows a call graph with several nodes and red edges indicating control flow. A specific node in the graph is highlighted with a green border, and its corresponding assembly code is shown in a callout box at the bottom right.

```
div    ecx
push   eax
movzx  eax, [ebp+SystemTime.wSecond]
push   eax
movzx  eax, [ebp+SystemTime.wMinute]
push   eax
movzx  eax, [ebp+SystemTime.wHour]
push   eax
movzx  eax, [ebp+SystemTime.wDay]
push   eax
movzx  eax, [ebp+SystemTime.wMonth]
push   eax
movzx  eax, [ebp+SystemTime.wYear]
push   eax
lea    eax, [ebp+Dest]
push   offset aHiMasterDDDDDD ; "Hi,Master [%d/%d/%d %d:%d:%d]\r\nWelCom...
push   eax ; Dest
call  ds:sprintf
add   esp, 44h
xor   ebx, ebx
lea    eax, [ebp+Dest]
push   ebx
push   eax ; Str
call  strlen
pop   ecx
push   eax ; len
lea    eax, [ebp+Dest]
push   eax ; int
push   [ebp+s] ; s
call  sub_100038EE
add   esp, 10h
cmp   eax, 0xFFFFFFFFh
jz    loc_10010714
```

```
lea    eax, [ebp+Buffer]
push   ebx
push   eax ; Str
call  strlen
```

```
xdoors_d:10095B44 ; char aHiMasterDDDDDD[]
xdoors_d:10095B44 aHiMasterDDDDDD db 'Hi,Master [%d/%d/%d %d:%d:%d]',0Dh,0Ah
xdoors_d:10095B44                                     ; DATA XREF: sub_1000FF58+145t0
xdoors_d:10095B44 db 'WelCome Back...Are You Enjoying Today?',0Dh,0Ah
xdoors_d:10095B44 db 0Dh,0Ah
xdoors_d:10095B44 db 'Machine Uptime [%-.2d Days %.2d Hours %-.2d Minutes %-.2d Secon'
xdoors_d:10095B44 db 'ds]',0Dh,0Ah
xdoors_d:10095B44 db 'Machine IdleTime [%-.2d Days %.2d Hours %-.2d Minutes %-.2d Seco'
xdoors_d:10095B44 db 'nds]',0Dh,0Ah
xdoors_d:10095B44 db 0Dh,0Ah
xdoors_d:10095B44 db 'Encrypt Magic Number For This Remote Shell Session [0x%02x]',0Dh,0Ah
xdoors_d:10095B44 db 0Dh,0Ah,0
```

9. What is dword_1008E5C4?

Follow the cross reference for write, and the function call setting eax:

xrefs to dword_1008E5C4			
Direct	T	Address	Text
	w	sub_10001656+22	mov dword_1008E5C4, eax
	D...	r sub_10007312+E	cmp dword_1008E5C4, edi
	D...	r sub_1000FF58+270	cmp dword_1008E5C4, ebx

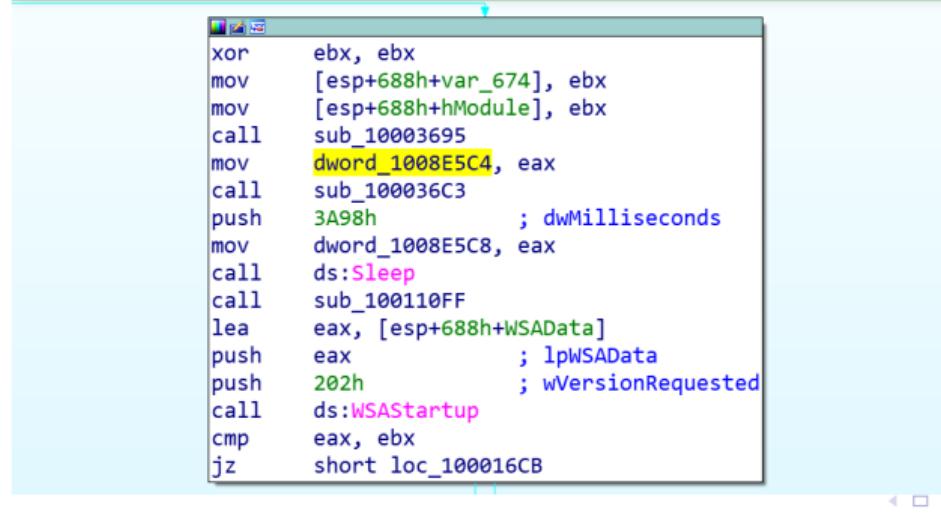
OK

Cancel

Search

Help

Line 1 of 3



```
xor    ebx, ebx
mov    [esp+688h+var_674], ebx
mov    [esp+688h+hModule], ebx
call   sub_10003695
mov    dword_1008E5C4, eax
call   sub_100036C3
push   3A98h           ; dwMilliseconds
mov    dword_1008E5C8, eax
call   ds:Sleep
call   sub_100110FF
lea    eax, [esp+688h+WSAData]
push   eax             ; lpWSAData
push   202h            ; wVersionRequested
call   ds:WSAStartup
cmp    eax, ebx
jz    short loc_100016CB
```

9. What is dword_1008E5C4?

Follow the cross reference for write, and the function call setting eax to find the following function:

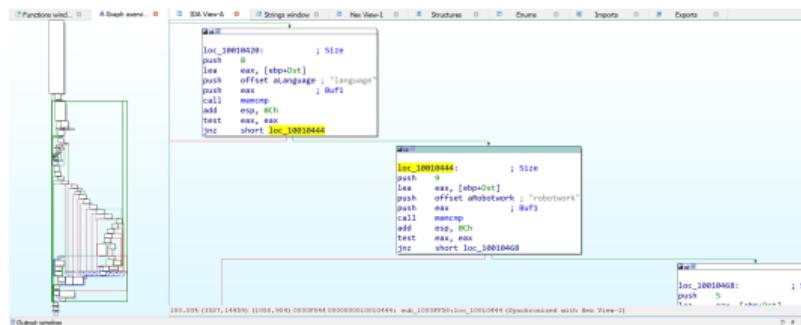
```
.text:10003695 ; sub_10003B75+74p ...
.text:10003695
.text:10003695 VersionInformation= _OSVERSIONINFOA ptr -94h
.text:10003695
.text:10003695      push    ebp    |
.text:10003696      mov     ebp, esp
.text:10003698      sub     esp, 94h
.text:1000369E      lea     eax, [ebp+VersionInformation]
.text:100036A4      mov     [ebp+VersionInformation.dwOSVersionInfoSize], 94h
.text:100036AE      push    eax      ; lpVersionInformation
.text:100036AF      call    ds:GetVersionExA
.text:100036B5      xor     eax, eax
.text:100036B7      cmp     [ebp+VersionInformation.dwPlatformId], 2
.text:100036BE      setz    al
.text:100036C1      leave
.text:100036C2      retn
.text:100036C2 sub_10003695 endp
.text:100036C2
```

It is the OS version.

10. In the subroutine at 0x1000FF58, what happens if the string comparison to robotwork is successful (when memcmp returns 0)?

The registry values at

`HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WorkTime` and `WorkTimes` are queried (and sent over the remote shell connection).



```
push    9          ; CODE XREF: su
        ; Size
lea     eax, [ebp+Dst]
push    offset aRobotwork ; "robotwork"
push    eax          ; Buf1
call    memcmp
add    esp, 0Ch
test   eax, eax
jnz    short loc_10010468
push    [ebp+s]      ; s
call    sub_100052A2
jmp    short loc_100103F6
```

```
.text:100052D9        stosw
.text:100052DB        stosb
.text:100052DC        lea    eax, [ebp+phkResult]
.text:100052DF        push   eax          ; phkResult
.text:100052E0        push   0F003Fh      ; samDesired
.text:100052E5        push   0             ; ulOptions
.text:100052E7        push   offset aSoftwareMicros ; "SOFTWARE\\Microsoft\\Windows\\CurrentVe".
.text:100052EC        push   80000002h      ; hKey
.text:100052F1        call   ds:RegOpenKeyExA
.text:100052F7        test   eax, eax
.text:100052F9        jz    short loc_10005309
.text:100052FB        push   [ebp+phkResult] ; hKey
.text:100052FE        call   ds:RegCloseKey
.text:10005304        jmp    loc_100053F6
.text:10005309 ; -----
.text:10005309 loc_10005309:           ; CODE XREF: sub_100052A2+571j
.push   ebx
.text:10005309        lea    eax, [ebp+cbData]
.text:1000530D        push   esi
.text:1000530E        push   eax          ; lpcbData
.text:1000530F        lea    eax, [ebp+Data]
.text:10005315        mov    ebx, ds:RegQueryValueExA
.text:10005318        push   eax          ; lpData
.text:1000531C        lea    eax, [ebp+Type]
.text:1000531F        push   eax          ; lpType
.text:10005320        push   0             ; lpReserved
.text:10005322        push   offset aWorktime ; "WorkTime"
```

300046E0 00000000100052E0: sub_100052A2+3E (Synchronized with Hex View-1)

11. What does the export PSLIST do?

Find PSLIST in the Exports window. Double-click it to go to the function. It can take two code paths. Follow the calls in both paths. They use CreateToolhelp32Snapshot to help them grab a process listing and sending the result over the socket.

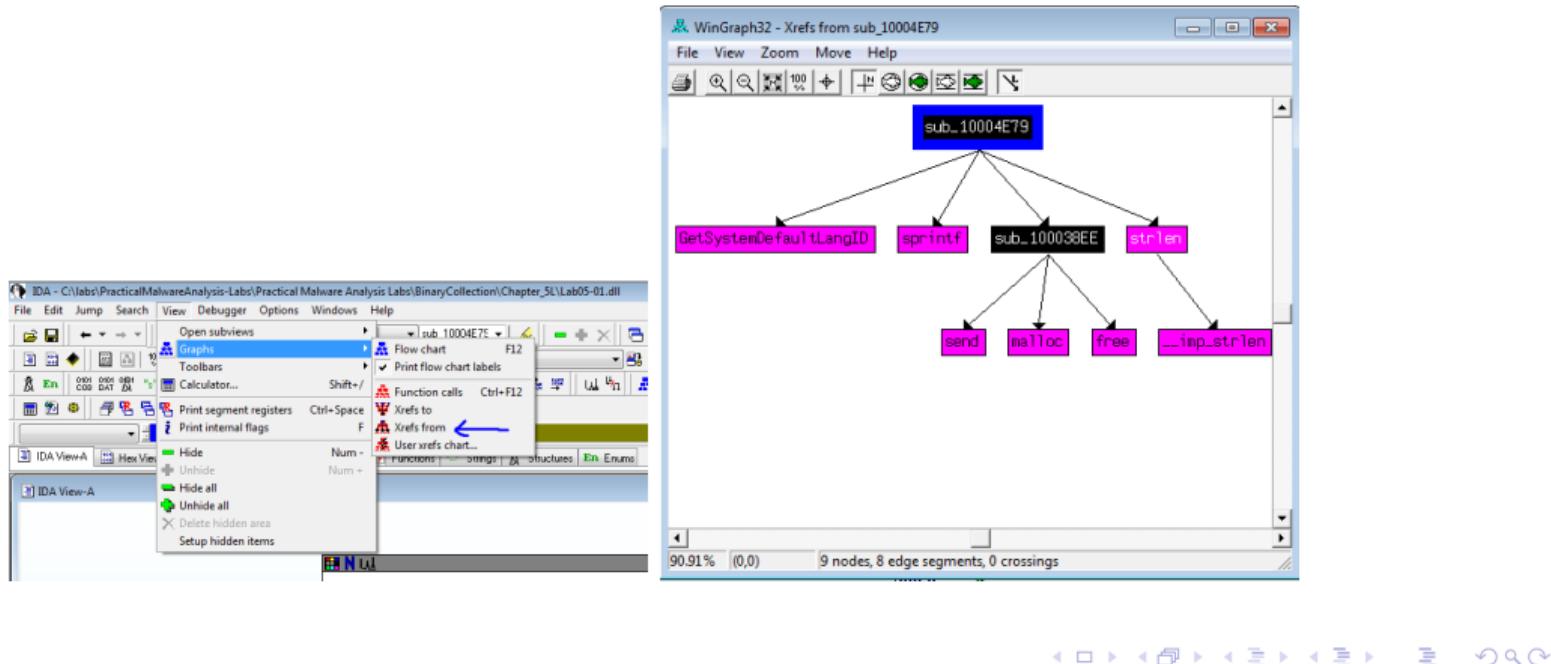
The screenshot shows the OllyDbg debugger interface with three main windows:

- Assembly Window:** Displays the assembly code for the `PSLIST` function. It includes several `int _stdcall PSLIST(int, int, char *Str, int)` declarations and a main loop that calls `loc_1000704E`.
- Exports Window:** Shows the export table with the following entries:

Name	Address	Ordinal
InstallRT	0000000010000B47	1
InstallSA	0000000010000EC1	2
InstallSB	0000000010000E92	3
PSLIST	0000000010007025	4
ServiceMain	000000001000C3F0	5
StartXS	0000000010007ECB	6
UninstallRT	000000001000F405	7
UninstallSA	000000001000A005	8
UninstallSB	000000001000F138	9
DllEntryPoint	00000000100015160	[main entry]
- CreateToolhelp32Snapshot Function:** A disassembly view of the `CreateToolhelp32Snapshot` function, which is called from the `PSLIST` function. It uses `rep stosd` to copy data to the stack and then calls `CreateToolhelp32Snapshot` with specific parameters.

12. Graph the cross references from sub_10004E79. Which API functions could be called by entering this function? What could you rename this function?

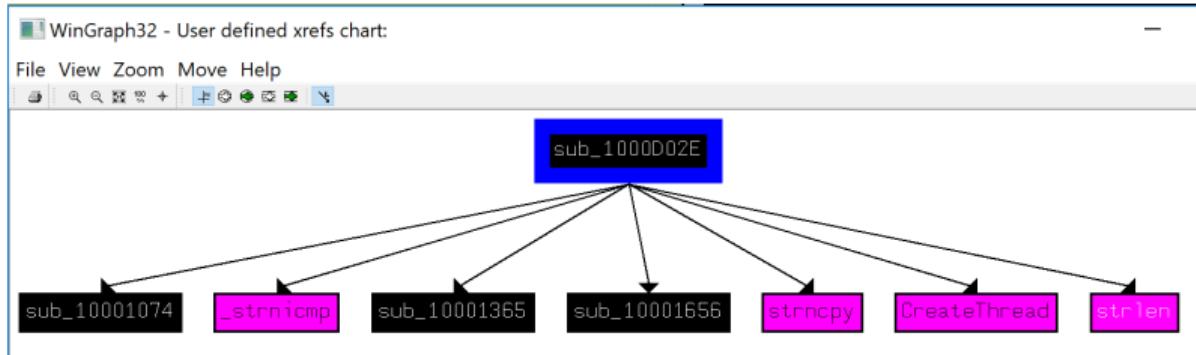
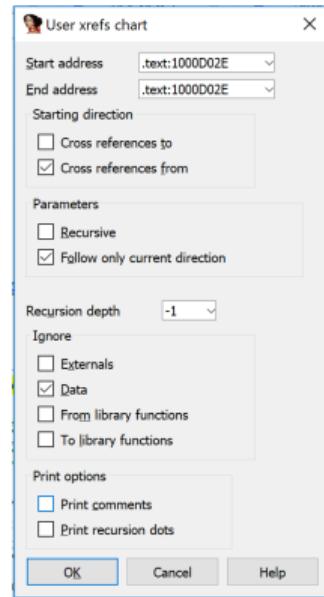
Go to the function, and open the graph view while having the cursor at the function name. GetSystemDefaultLangID, send, and sprintf are called. It could be called GetSystemLanguage.



13. How many Windows API functions does DllMain call directly? How many at depth 2?

View→Graphs→User xrefs chart

Direct calls: strcpy, strnicmp, CreateThread, and strlen. Several at depth 2.



14. At 0x10001385 there is a call to Sleep. How long will the program sleep?

- ▶ off_10019020 leads us to the string [This is CTI]30
- ▶ eax is set to 30 ms using the value found from off_10019020+D.
- ▶ eax is multiplied by 1000, giving 30 seconds.

To change the number 3E8h into decimal 1000, rightclick and choose decimal format.

```
.text:1000133B
.text:1000133B loc_1000133B:                 ; CODE XREF: sub_10001074+71↑j
.text:1000133B             mov     dword_1008E5CC, ebp
.text:10001341
.text:10001341 loc_10001341:                 ; CODE XREF: sub_10001074+10F↑j
.text:10001341             ; sub_10001074+1B0↑j ...
.text:10001341             mov     eax, off_10019020
.text:10001346             add     eax, 0Dh
.text:10001349             push    eax      ; Str
.text:1000134A             call    ds:atoi
.text:10001350             imul    eax, 3E8h
.text:10001356             pop     ecx
.text:10001357             push    eax      ; dwMilliseconds
.text:10001358             call    ds:Sleep
.text:1000135E             xor     ebp, ebp
.text:10001360             jmp    loc_100010B4
.text:10001360 sub_10001074:                endp
+text:10001360
```

15. At 0x10001701 is a call to the function `socket`. What are the three parameters?

6, 1 and 2.

The screenshot shows the assembly view in IDA Pro. The assembly code is as follows:

```
.text:100016FB loc_100016FB:           ; CODE XREF: sub_10001656+374!j
.text:100016FB                         ; sub_10001656+A09!j
    push    6                           ; protocol
    push    1                           ; type
    push    2                           ; af
    call    ds:socket
    mov     edi, eax
    cmp     edi, 0FFFFFFFFFFh
    jnz    short loc_10001722
    call    ds:WSAGetLastError
    push    eax
    push    offset aSocketGetlaste ; "socket() GetLastError reports %d\n"
    call    ds:__imp_vprintf
    pop     ecx
    pop     ecx
    pop     ecx
.loc_10001722:                         ; CODE XREF: sub_10001656+B6f!j
    mov     eax, off_100019040
    push    10h                          ; Size
    add    eax, 00h
    push    offset asc_10093540 ; "
    push    eax                          ; Buf1
    call    memcmp
    add    esp, 0Ch
    test   eax, eax
    jz     loc_100017ED
    cmp    dword_1000E5CC, ebx
00000B01 0000000010001701: sub_10001656+AB {Synchronized with Hex View-1}
```

16. Use the MSDN page for socket and the named symbolic constants functionality in IDA to make the parameters more meaningful.

Find the list of relevant named symbolic constants at: <https://docs.microsoft.com/en-us/windows/win32/api/winsock2/nf-winsock2-socket>

The screenshot shows the assembly dump in IDA Pro. The code is as follows:

```
push offset ProcName ; "Plug_KeyLog_Restart"
push eax ; hModule
call ds:GetProcAddress
call eax

loc_100016F5:
mov ebp, ds:closesocket

loc_100016FB:      ; protocol
push 6
push 1             ; type
push 2             ; af
call ds:socket
mov edi, eax
cmp edi, 0FFFFFFFh
jnz short loc_10001722

call ds:WSAGetLastError
push eax
push offset aSocketGetlaste ; "socket() GetLastError reports %d\n"

00000000100016FF: sub_10001656+A9 (Synchronized with Hex View-1)
```

The screenshot shows the assembly dump in IDA Pro. The code is as follows:

```
push offset ProcName ; "Plug_KeyLog_Restart"
push eax ; hModule
call ds:GetProcAddress
call eax

loc_100016F5:
mov ebp, ds:closesocket

loc_100016FB:      ; protocol
push IPPROTO_TCP
push SOCK_STREAM ; type
push AF_INET      ; af
call ds:socket
mov edi, eax
cmp edi, 0FFFFFFFh
jnz short loc_10001722

call ds:WSAGetLastError
push eax
push offset aSocketGetlaste ; "socket() GetLastError reports %d\n"

FF 00000000100016FF: sub_10001656+A9 (Synchronized with Hex View-1)
```

Part 4: Executable with known source

Use the source code to see what the executable does.

Find out the same using Process Monitor, Process Explorer, and IDA.

Part 4: What to learn

- ▶ Learn to navigate in IDA and understand what you see.
- ▶ Identify the command line parameters. Learn to recognize command line parameters in IDA.
- ▶ In dynamic analysis, what you see happening is not necessarily everything the executable can do.
- ▶ Run the executable with all variants of input to see all types of behavior. Find the related events in Process Monitor.
- ▶ When you want to learn how to reverse a specific behavior, you can write a related program, compile it and reverse it.

Exercise Task 3

Part 1: Integers

If your computer program uses unsigned 8-bit integers, what will the result of the following be?

- ▶ $255 + 2 = 1$
- ▶ $0 - 1 = 255$

If your computer program used signed 8-bit integers, what would the result of the following be:

- ▶ $0 - 1 = -1$

Exercise Task 3

Part 2: Stack cookies

- ▶ Read the section 30.3.1 in *Low level Software Security by Example* (linked from Canvas).
- ▶ Go to the Files folder in Canvas and download and extract lecture03.zip .
- ▶ Make a text file containing at least 28 A's, and place it in the folder vuln-cookie.
- ▶ Open vuln-cookie.exe in Immunity Debugger, with the text file as an argument.
- ▶ Place a breakpoint (F2) at the address 0x01401066, and run (F9) until you hit the breakpoint.
- ▶ Step over (F8) the call to strcpy. Notice that strcpy overflows the return address of the current function.
- ▶ Continue running (F9), and notice that the program does not attempt to return into 41414141. Why is that?

Exercise Task 3

Part 2: Stack cookies

Viewing the disassembly of our function, we see that a cookie gets loaded into EAX (01401046), then XOR'ed with the frame pointer (0140104B) and placed on the stack as a canary (0140104D).

Before restoring the frame pointer and returning, the canary is fetched from the stack (0140106E), XOR'ed with the frame pointer (01401071) and then checked with a call to vuln-coo.014011FA (01401073)

01401040	\$ 55	PUSH EBP
01401041	. 8BEC	MOU EBP,ESP
01401043	. 83EC 14	SUB ESP,14
01401046	. A1 40D04101	MOU EAX,DWORD PTR DS:[141D040]
0140104D	. 33C5	XOR EAX,EBP
0140104D	. 8945 FC	MOU DWORD PTR SS:[EBP-4],EAX
01401050	. 33C0	XOR EAX,EAX
01401052	. 8945 EC	MOU DWORD PTR SS:[EBP-14],EAX
01401055	. 8945 F0	MOU DWORD PTR SS:[EBP-10],EAX
01401058	. 8945 F4	MOU DWORD PTR SS:[EBP-C],EAX
0140105B	. 8945 F8	MOU DWORD PTR SS:[EBP-8],EAX
0140105E	. 8B4D 08	MOU ECX,DWORD PTR SS:[EBP+8]
01401061	. 51	PUSH ECX
01401062	. 8D55 EC	LEA EDX,DWORD PTR SS:[EBP-14]
01401065	. 52	PUSH EDX
01401066	E8 95FFFFFF	CALL vuln-coo.strcpy
0140106B	. 83C4 08	ADD ESP,8
0140106E	. 8B4D FC	MOU ECX,DWORD PTR SS:[EBP-4]
01401071	. 33CD	XOR ECX,EBP
01401073	. E8 82010000	CALL vuln-coo.014011FA
01401078	. 8BE5	MOU ESP,EBP
0140107A	. 5D	POP EBP
0140107B	. C3	RETN

Exercise Task 3

Part 2: Stack cookies

When we continue running after the call to `strcpy`, an interrupt (INT29) is issued by `vuln-coo.014011FA`. This is a Windows mechanism called *fast fail*, and is a way for a potentially corrupted process to request immediate process termination. The constant 2 which is loaded into ECX before issuing the interrupt, is the fast fail failure code for a stack cookie check failure.

The screenshot shows a debugger interface with two main panes. The left pane displays assembly code with memory addresses and opcodes. The right pane shows the processor registers.

Assembly Code:

Address	OpCode	Comment
01401482	. 5D	POP EBP
01401483	L: C3	RETN
01401484	> 55	PUSH EBP
01401485	. 8BEC	MOU EBP,ESP
01401487	. 81EC 24030000	SUB ESP,324
0140148D	. 6A 17	PUSH 17
0140148F	. E8 AA080000	CALL <JMP.&KERNEL32.IsProcessorFeatureP
01401494	. 85C0	TEST EAX,EAX
01401496	. 74 05	JE SHORT vuln-coo.0140149D
01401498	. 6A 02	PUSH 2
0140149A	. 59	POP ECX
0140149B	. CD 29	INT 29

Registers (FPU):

Name	Value
EAX	00000001
ECX	00000002
EDX	000001E0
EBP	00349000
ESP	0019F9E4
EBP	0019FD08
ESI	0141DE84 vuln-coo.0141DE84
EDI	005D5AA0
EIP	0140149B vuln-coo.0140149B
C	0 ES 002B 32bit 0(FFFFFFF)

Exercise Task 3

Part 3: Size checking

Give a fix to the flaw in the code below for concatenating two strings:

```
char buf [128];
combine(char *s1, size_t len1,
        char *s2, size_t len2)
{
    if (len1 + len2 + 1 <= sizeof(buf)) {
        strncpy(buf, s1, len1);
        strncat(buf, s2, len2);
    }
}
```

Exercise Task 3

Part 3: Size checking

```
char buf[128];
combine(char *s1, size_t len1,
        char *s2, size_t len2)
{
    if (len1 + len2 + 1 <= sizeof(buf) && len1 < len1 + len2) {
        strncpy(buf, s1, len1);
        strncat(buf, s2, len2);
    }
}
```

For an n bit unsigned integer, the maximal value for len2 is $2^n - 1$, and $\text{len1} + 2^n - 1 \bmod 2^n = \text{len1} - 1$; if there is an integer overflow, the result will be smaller than len1 . (No claim that this is the optimal solution.)

Exercise Task 4

Part 1: Stack canaries

- ▶ Explain the principle of stack canaries, and how it protects specific pointers stored in a stack frame.

A stack canary is a specific value placed inside the call stack frame, at the memory address just above (i.e. with lower address than) the saved EBP and the saved EIP to protect those values. The program verifies that the stack canary has not been modified before the program pops (fetches) the saved EIP and EBP to exit the call. If the stack canary has not been modified, then it is unlikely that the saved EIP and EBP have been modified by a buffer overflow, so the program continues normally. If the stack canary has been modified, then it is likely that the saved EIP and EBP have been modified, which is a sign of a buffer overflow attack, so the program is terminated.

Exercise Task 4

Part 1: Stack canaries

- ▶ What are the disadvantages of using stack canaries?

Disadvantages are:

- ▶ Slows down the computer, because it creates overhead code to be executed.
- ▶ Only protects EIP and EBP values, not the rest of the stack frame.

Exercise Task 4

Part 1: Stack canaries

- ▶ Explain very roughly a scenario for attacking stack canaries so that buffer overflow attacks will not be detected.

If the attacker can recompute/learn the valid stack canary value, then the attacker can keep the canary value unchanged when overflowing EBP and EIP.

Also, if the attacker's code executes before returning from the function, the stack cookie will not stop the attack.

Exercise Task 4

Part 2: Data Execution Prevention (DEP)/Non-executable stack (NX)

- ▶ Go to the Files folder in Canvas and download and extract lecture04.zip .
- ▶ Open vuln-sec.exe in Immunity Debugger, with the file input.bin as an argument.
- ▶ Open Process Explorer (procexp.exe) from Sysinternals, and check if vuln-sec.exe has ASLR and DEP enabled.
- ▶ Place a breakpoint (F2) at the address 0x00401027, and run (F9) until you hit the breakpoint.
- ▶ Step over (F8) the call to strcpy. Notice that strcpy overflows the return address of the current function.
- ▶ Single step (F7) until the function return, and we jump into the stack. What happens when we try to execute the NOPs?

Exercise Task 4

Part 2: Data Execution Prevention (DEP)/Non-executable stack (NX)

- ▶ Open Process Explorer (procexp.exe) from Sysinternals, and check if vuln-sec.exe has ASLR and DEP enabled.

We see that DEP has been enabled, but not ASLR.

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	DEP	Integrity	ASLR
winhex.exe		11,276 K	23,508 K	3848	WinHex	X-Ways Software Technolo...	Enabled (permanent)	Medium	ASLR
explorer.exe	0.01	11,420 K	30,560 K	3912	Internet Explorer	Microsoft Corporation	Enabled (permanent)	Medium	ASLR
explorer.exe	< 0.01	37,896 K	65,992 K	2352	Internet Explorer	Microsoft Corporation	Enabled (permanent)	Low	ASLR
calc.exe		12,840 K	16,472 K	2784	Windows Calculator	Microsoft Corporation	Enabled (permanent)	Medium	ASLR
ImmunityDebugger.exe	0.08	20,792 K	23,506 K	2252	Immunity Debugger, 32-bit an...		Disabled	Medium	
vuln-sec.exe		156 K	1,880 K	3148			Enabled (permanent)	Medium	
procexp.exe	0.66	10,980 K	19,488 K	2016	Sysinternals Process Explorer	Sysinternals - www.sysinter...	Enabled (permanent)	Medium	ASLR
SearchIndexer.exe		17,000 K	10,936 K	2576	Microsoft Windows Search In...	Microsoft Corporation	n/a		ASLR
System Idle Process	97.71	0 K	24 K	0			n/a	n/a	
System	0.03	52 K	236 K	4			n/a	n/a	
Interrupts		1.27	0 K	0 K	n/a Hardware Interrupts and DPCs		n/a	n/a	
smss.exe			260 K	828 K	228		n/a	n/a	
csrss.exe	< 0.01		1,408 K	3,508 K	308		n/a	n/a	
csrss.exe	0.04		2,284 K	6,564 K	356		n/a	n/a	
conhost.exe			996 K	4,404 K	1192 Console Window Host	Microsoft Corporation	Enabled (permanent)	Medium	ASLR
wmifnt.exe			896 K	3,316 K	364		n/a	n/a	
services.exe		4,088 K	7,108 K	452			n/a	n/a	
svchost.exe		3,016 K	7,128 K	564	Host Process for Windows S...	Microsoft Corporation	n/a		ASLR
VBoxService.exe		1,712 K	4,400 K	628	VirtualBox Guest Additions S...	Oracle Corporation	n/a		ASLR
sppsvc.exe		2,072 K	7,192 K	3972	Microsoft Software Protection...	Microsoft Corporation	n/a		ASLR
lsass.exe		3,300 K	9,068 K	460	Local Security Authority Proc...	Microsoft Corporation	n/a		ASLR
lsm.exe		1,816 K	4,516 K	468			n/a	n/a	
winlogon.exe		2,444 K	5,960 K	408			n/a	n/a	

CPU Usage: 2.29% Commit Charge: 14.28% Processes: 42 Physical Usage: 28.85%

Exercise Task 4

Part 2: Data Execution Prevention (DEP)/Non-executable stack (NX)

- ▶ Single step (F7) until the function return, and we jump into the stack. What happens when we try to execute the NOPs?

We get an Access Violation when trying to execute the instruction on the stack, as it is marked non-executable.

The screenshot shows a debugger interface with several windows:

- Registers (FPU) window:** Shows CPU registers:
 - EAX 0012FB20
 - ECX 0012FD54
 - EDX 0043C515 vuln-sec.0043C515
 - EBX 7FFDF000
 - ESP 0012FB38
 - EBP 00909090
 - ESI 0044A778 OFFSET vuln-sec._argc
 - EDI 001750F8
 - EIP 0012FB20
- Memory Dump window:** Shows memory starting at address 00449000.

Address	Hex dump	ASCII
00449000	55 73 61 67 65 3A 20 25	Usage: %
00449008	73 20 46 49 4C 45 0A 00	s FILE..
00449010	72 00 00 00 5B 45 52 52	r...[ERR]
00449018	4F 52 5D 20 63 6F 75 6C	OR] coul
00449020	64 20 6E 6F 74 20 6F 70	d not op
00449028	65 6F 20 25 73 0A 00 00	en %s...
- Registers (CPU) window:** Shows CPU registers:
 - C 0 ES 0023 32bit 0(FFFFFFFF)
 - P 0 CS 0018 32bit 0(FFFFFFFF)
 - A 1 SS 0023 32bit 0(FFFFFFFF)
- Status Bar:** Shows the error message: "Access violation when executing [0012FB20] - use SHIFT+F7/F8/F9 to pass exception to program".

Exercise Task 4

Part 3: Address space layout randomization (ASLR)

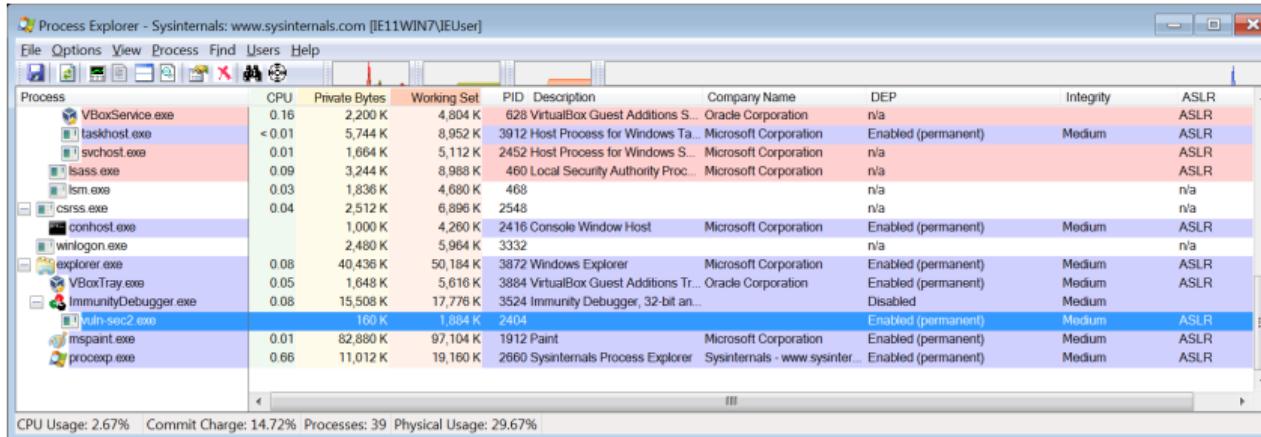
- ▶ Open vuln-sec2.exe in Immunity Debugger, with the file input.bin as an argument.
- ▶ Open Process Explorer (procexp.exe) from Sysinternals, and check if vuln-sec2.exe has ASLR and DEP enabled.
- ▶ Try to find the address of the call to strcpy (hint: offset 0x27 from the beginning of the file).
- ▶ Place a breakpoint (F2) at the address you found, and run (F9) until you hit the breakpoint.
- ▶ Step over (F8) the call to strcpy. Notice that strcpy overflows the return address of the current function.
- ▶ What happens when the function returns? In what way does it differ from what happened in Part 2?

Exercise Task 4

Part 3: Address space layout randomization (ASLR)

- ▶ Open Process Explorer (procexp.exe) from Sysinternals, and check if vuln-sec2.exe has ASLR and DEP enabled.

We see that both DEP and ASLR have been enabled.



The screenshot shows the Process Explorer interface with the following data:

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	DEP	Integrity	ASLR
VBoxService.exe	0.16	2,200 K	4,804 K	628	VirtualBox Guest Additions S...	Oracle Corporation	n/a	ASLR	ASLR
taskhost.exe	< 0.01	5,744 K	8,952 K	3912	Host Process for Windows Ta...	Microsoft Corporation	Enabled (permanent)	Medium	ASLR
svchost.exe	0.01	1,664 K	5,112 K	2452	Host Process for Windows S...	Microsoft Corporation	n/a	ASLR	ASLR
lsass.exe	0.09	3,244 K	8,988 K	460	Local Security Authority Proc...	Microsoft Corporation	n/a	ASLR	ASLR
lsm.exe	0.03	1,836 K	4,680 K	468			n/a	n/a	n/a
csrss.exe	0.04	2,512 K	6,896 K	2548			n/a	n/a	n/a
conhost.exe		1,000 K	4,260 K	2416	Console Window Host	Microsoft Corporation	Enabled (permanent)	Medium	ASLR
winlogon.exe		2,480 K	5,964 K	3332			n/a	n/a	n/a
explorer.exe	0.08	40,436 K	50,184 K	3872	Windows Explorer	Microsoft Corporation	Enabled (permanent)	Medium	ASLR
VBoxTray.exe	0.05	1,648 K	5,616 K	3884	VirtualBox Guest Additions Tr...	Oracle Corporation	Enabled (permanent)	Medium	ASLR
ImmunityDebugger.exe	0.08	15,508 K	17,776 K	3524	Immunity Debugger, 32-bit an...		Disabled	Medium	ASLR
vuln-sec2.exe		160 K	1,884 K	2404			Enabled (permanent)	Medium	ASLR
mspaint.exe	0.01	82,880 K	97,104 K	1912	Paint	Microsoft Corporation	Enabled (permanent)	Medium	ASLR
procexp.exe	0.66	11,012 K	19,160 K	2660	Sysinternals Process Explorer	Sysinternals - www.sysint...	Enabled (permanent)	Medium	ASLR

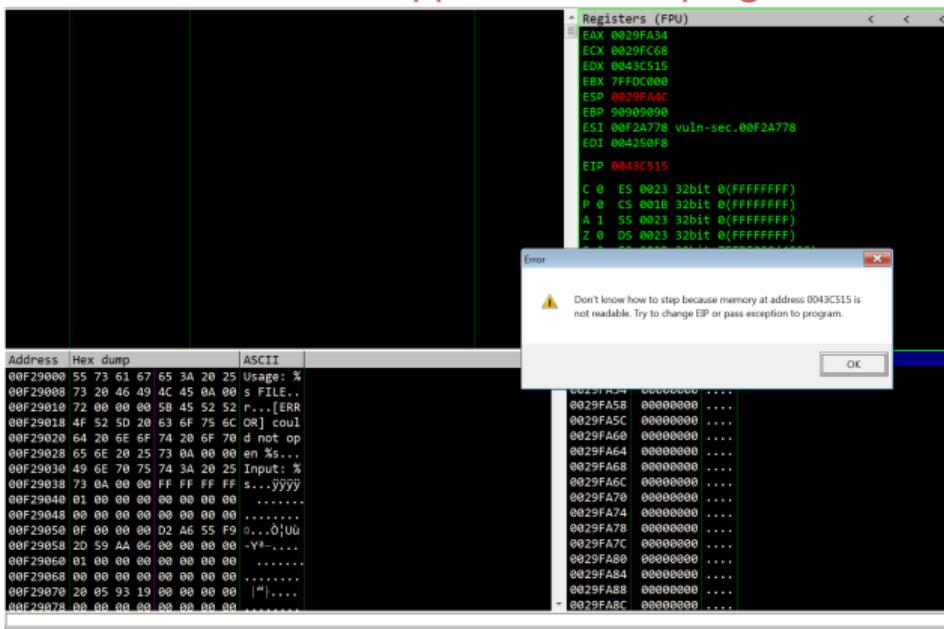
CPU Usage: 2.67% Commit Charge: 14.72% Processes: 39 Physical Usage: 29.67%

Exercise Task 4

Part 3: Address space layout randomization (ASLR)

- ▶ What happens when the function returns? In what way does it differ from what happened in Part 2?

The address we return into is not mapped, and the program crashes.



Exercise Task 4

Part 4: DEP/NX and ASLR questions

- ▶ Explain the principle of DEP/NX.

DEP/NX is a mechanism that allows ranges of memory locations to be marked as non-executable. The stack does not normally contain executable code, so the stack memory can be marked as non-executable. Even if an attacker is able to inject malicious code into the stack, the malicious code will not be executed even if the instruction pointer is set to point to it, instead the program or computer will simply crash.

Exercise Task 4

Part 4: DEP/NX and ASLR questions

- ▶ Describe a type of buffer overflow attack that cannot be prevented by DEP/NX.

DEP/NX does not protect against attacks where the attacker redirects the control flow to existing, executable code in the program's memory.

This class of attacks are typically referred to as *return-to-libc*, where the attacker modifies the saved EIP so that the program execution returns to specific program library functions that by definition is executable.

ROP (return-oriented programming) is a variant, where the attacker chain together numerous small pieces of code (called *gadgets*), all ending in the return instruction, each doing a small amount of useful work for the attacker. We will talk more about ROP in the last lecture of the course.

Exercise Task 4

Part 4: DEP/NX and ASLR questions

- ▶ Explain the principle of ASLR.

ASLR hinders buffer overflow attacks by making it more difficult for an attacker to predict target memory addresses.

For example, attackers trying to execute code using *return-to-libc* must first locate the code to be executed, and if that is difficult then the attack will most likely fail.

With ASLR the memory addresses have to be guessed or leaked, and a mistaken guess will usually lead to the application crashing.

Exercise Task 4

Part 4: DEP/NX and ASLR questions

- ▶ Is enabling ASLR for just the main executable sufficient, or should it be enabled for all the dynamically linked libraries as well?

For ASLR to be efficient, all the dynamically linked libraries must have ASLR enabled. Almost all libraries will have enough reusable code fragments the attacker can use to eventually gain arbitrary code execution.

Exercise Task 5

Part 1: Password length

Assume that a password can only contain the 26 characters from the alphabet.

- ▶ How many different passwords are possible if a password is at most n , $n = 4, 6, 8$, characters long and there is no distinction between upper case and lower case characters?

$$4 \text{ characters: } 26^4 + 26^3 + 26^2 + 26 = 4.75 * 10^5$$

$$6 \text{ characters: } 26^6 + 26^5 + 26^4 + 26^3 + 26^2 + 26 = 3.21 * 10^8$$

$$8 \text{ characters: } 26^8 + 26^7 + 26^6 + 26^5 + 26^4 + 26^3 + 26^2 + 26 = 2.17 * 10^{11}$$

Exercise Task 5

Part 1: Password length

Assume that a password can only contain the 26 characters from the alphabet.

- ▶ How many different passwords are possible if a password is at most n, n = 4,6,8, characters long and passwords are case sensitive?

$$\text{4 characters: } 52^4 + 52^3 + 52^2 + 52 = 1.52 * 10^7$$

$$\text{6 characters: } 52^6 + 52^5 + 52^4 + 52^3 + 52^2 + 52 = 4.11 * 10^{10}$$

$$\text{8 characters: } 52^8 + 52^7 + 52^6 + 52^5 + 52^4 + 52^3 + 52^2 + 52 = 1.11 * 10^{14}$$

Exercise Task 5

Part 2: Brute force

Assume that passwords have length six and all alphanumerical characters, upper and lower case, can be used in their construction. How long will a brute force attack take on average if:

- ▶ it takes one tenth of a second to check a password?
- ▶ it takes a microsecond to check a password?

There are 62 symbols, so overall there are 62^6 possible passwords. In the first case, the search takes about 91 years, in the second case about 8 hours. The purpose of the exercise is to demonstrate that speed-ups in password checking are not relevant to individual end users but help an attacker.

Exercise Task 5

Part 3: Hashes and salts

1. What is the advantage of storing password databases as hash values instead of in plaintext?

Should an attacker obtain a copy of the passwords database, then the attacker does not get passwords directly if they are stored as hash values. The one-way property of hash functions makes it computationally infeasible to find a password from the hash value.

Exercise Task 5

Part 3: Hashes and salts

1. What is the advantage of storing passwords as salted hash values instead of just as hash values?

Should an attacker obtain a copy of a database of hashed passwords, then the attacker could use a precomputed hash table of all possible or likely passwords up to a certain length. The computing and storage capacity in modern devices makes it possible to prepare such hash tables. Then the attacker could just look up each password hash value in the hash table to find the plaintext password. In order to prevent this attack, hashing passwords with a salt (random value) would force the attacker to compute a separate hash table for every salt value, which isn't feasible, making precomputed hash/rainbow tables useless. Hence, salting makes it much more difficult to crack hashed passwords.

In addition, salting the password before hashing ensures that equal passwords get different hash values. In case one password has been cracked, this prevents attack against multiple users with the same password.

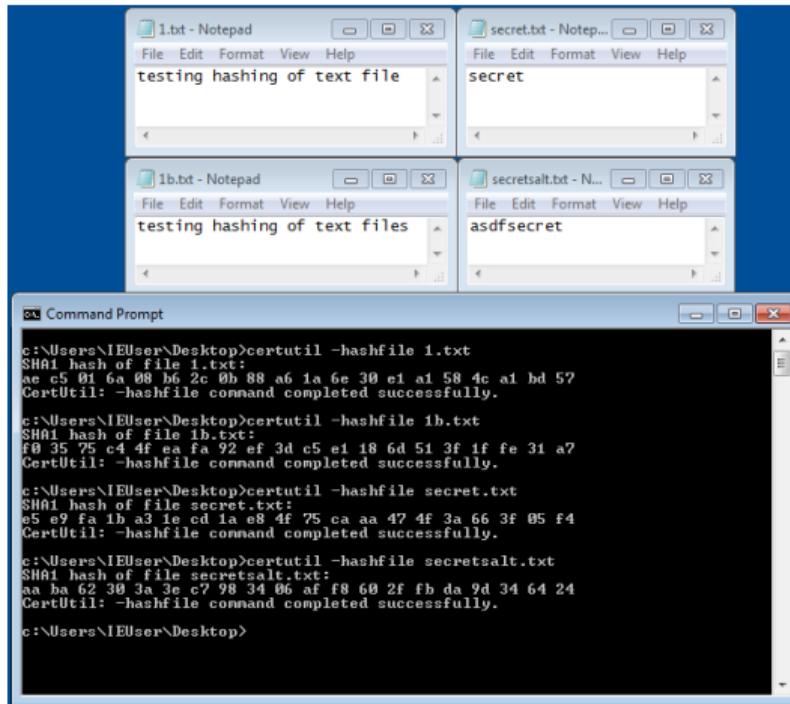
Exercise Task 5

Part 4: More hashes and salts

- ▶ Create a text file (`c:\pw.txt`) on your Windows 10 VM, and write a random string (without any linebreaks).
- ▶ Open a command prompt and use the command `certutil` to hash the file:
`certutil -hashfile c:\pw.txt SHA1.`
- ▶ Make note of the hash value.
- ▶ Add a single character to the text file.
- ▶ Hash the file again, and note the hash value. Is it similar to the previous hash?
- ▶ Substitute the string with the word *secret*, and rehash the file. Do you get the same result as in the lecture slides?
- ▶ Try rehashing after appending the salt used in the slides.

Exercise Task 5

Part 4: More hashes and salts



The screenshot shows a Windows desktop environment with four open windows:

- 1.txt - Notepad**: Contains the text "testing hashing of text file".
- secret.txt - Notepad**: Contains the text "secret".
- 1b.txt - Notepad**: Contains the text "testing hashing of text files".
- secretsalt.txt - Notepad**: Contains the text "asdfsecret".

Below these windows is a Command Prompt window titled "Command Prompt". The output of the command "certutil -hashfile" is displayed:

```
c:\>certutil -hashfile 1.txt
SHA1 hash of file 1.txt:
ae c5 01 6a 08 b6 2c 0b 88 a6 1a 6e 30 e1 a1 58 4c a1 bd 52
CertUtil: -hashfile command completed successfully.

c:\>certutil -hashfile 1b.txt
SHA1 hash of file 1b.txt:
f0 35 75 c4 4f ea fa 92 ef 3d c5 e1 18 6d 51 3f 1f fe 31 a7
CertUtil: -hashfile command completed successfully.

c:\>certutil -hashfile secret.txt
SHA1 hash of file secret.txt:
e5 e9 fa 1b a3 fe cd 1a e8 4f 75 ca aa 47 4f 3a 66 3f 05 f4
CertUtil: -hashfile command completed successfully.

c:\>certutil -hashfile secretsalt.txt
SHA1 hash of file secretsalt.txt:
aa ba 62 30 3a 3e c7 98 34 06 af f8 60 2f fb da 9d 34 64 24
CertUtil: -hashfile command completed successfully.

c:\>
```

Exercise Task 5

Part 5: Decryption

You have found the following data in binary format: $d=000011110000111100000001$. Reversing of the malware told you that you first need to xor the data with the string key, and thereafter reverse a Caesar cipher with a shift of 5. Decrypt it and write it as an ASCII string:

- 1 Write key in binary format using an ASCII table (e.g. a = 01100001) and xor the result with the data d .
- 2 Convert the result from binary format to ASCII.
- 3 Replace each letter with the letter five positions earlier in the alphabet (A-Z). Wrap around if necessary.

Perform the steps by hand to master the techniques.

Resource: <http://sticksandstones.kstrom.com/appen.html>

Exercise Task 5

Part 5: Decryption

1. Write key in binary format using an ASCII table (e.g. a = 01100001):

key = 011010110110010101111001

Xor the result with the data $d=000011110000111100000001$:

011010110110010101111001

000011110000111100000001

011001000110101001111000

2. Convert the result from binary format to ASCII:

011001000110101001111000 = djx

3. Replace each letter with the letter five positions earlier in the alphabet (A-Z). Wrap around if necessary.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

V W X Y Z A B C D E F G H I J K L M N O P Q R S T U

d = y, j = e, x = s

The data d decodes to yes.

Resource: <http://sticksandstones.kstrom.com/appen.html>

Exercise Task 6

Part 1: Questions

1. In Linux, access rights are defined for users and groups. To facilitate better security management, users are placed into groups. How does Linux decide on an access request when an individual user has fewer privileges than its group?

In Linux there are only positive access rights, and permission bits are examined in the order of priority: owner > group > others. When the user in question is the owner, and the owner does not have a required right that has been given to the group, access will be denied. When the user is not the owner, access will be granted.

Exercise Task 6

Part 1: Questions

2. How can access rights given to a group be withheld from individual members of that group?

In standard Linux there is no mechanism to withhold rights from individual group members. A workaround might be to create a new group without that particular member, and let this new group own the resource.

Exercise Task 6

Part 1: Questions

3. In Linux systems users have a UID and username. Explain the relationship between usernames and UIDs. Which identifier is used for authentication, and which identifier is used to make access control decisions? Where are passwords stored in Linux systems, and who has access to them?

In Linux systems each user has a username which is mapped to a UID (User Identity). Multiple users can be mapped to the same UID, but a single username can only have a single UID. The username is the key to find the hashed password which is used for user authentication. The UID is used when the system approves access to resources.

Exercise Task 6

Part 1: Questions

4. How are passwords protected in Linux systems?

Passwords are normally stored in a salted and hashed form to make brute force cracking as difficult as possible. Systems administrators can reduce the likelihood of brute force attacks by making the list of hashed passwords unreadable by unprivileged users. A simple way to do this would be to make the `passwd` file itself readable only by the root user. However, this would restrict access to other data in that file such as username-to-UID mappings, which would break many existing utilities and provisions. The typical solution is to let a "shadow" password file hold the password hashes separate from the other data in the world-readable `passwd` file. For local files, this is usually `/etc/shadow` on Linux and Unix systems, which readable only by root. In networked environments the password hashes are typically located in an LDAP directory and accessible by root processes through the LDAP protocol.

Exercise Task 6

Part 2: Users and Groups

This practical exercise will introduce user and group management in Linux.

Part 2a: Download Xubuntu

- ▶ Download the Linux virtual machine (Xubuntu) from
<http://unik4270.project.ifi.uio.no/xubuntu.zip>.
- ▶ Extract the zip-file and open the VM in VirtualBox.
- ▶ Log in with the username **student** and the password **tek5510**.

Exercise Task 6

Part 2b: User creation

First we need to create some users at our system. Do this by opening the Xubuntu menu in the top left corner, and start writing [Settings Manager](#) in the search field. Open it when it shows in the result pane. Start writing [Users and Groups](#) in the search field of the Settings window. Open it when it shows in the result pane.

You have now opened the User Settings window. Click the [Add](#) button. Provide the [Student](#) user password when asked to authenticate.

1. Create the user Ole and give him a password.
2. Create the user Dole, and give him the same password as Ole.
3. Create the user Doffen with a different password than the others.

Exercise Task 6

Part 2c: The /etc/shadow file

Take a look at the /etc/shadow file that stores the passwords. Detailed information about this file can be found at [1] or by issuing the command `man shadow` in a terminal.

Note that you need to have root privileges to access the file (`sudo cat /etc/shadow`).

```
student:$6$kQy5lyss$9eZnAMZXAz7Tyt45yPzHizh6Z/rnisHtQ5Bn2Nr0WQ/dB2horbL0ge8eARfg022de1Mol9Uxk9nrTrqDC7A3X0:16674:0:99999:7:::  
vboxadd:!:16674::::::  
ole:$6$9Roat28Y$qNKVHOPDuumeE9bkKdz.QUFaT56JvCuPOCHs9S13alryM/FgxLL6QGh5DL7JclgnaqKAU1dv7mB1mnhk4zJSy1:17448:0:99999:7:::  
jole:$6$nSsUX5.Y$BTbH8HlIIImaPZAЕ5MP0klWVs.oNXa3KFY76ePuXlQpSFJK8Jwlv4Aue7UNSdktASRvfxMi6gay4fqhuNFrEfie0:17448:0:99999:7:::  
doffen:$6$ZFQ0FFV7$o3ZmK1D8cHV/AvmS84xJ.580rKR4WY50JGLyni/GI8FtPZnPxgk39MvmrkfDqWrC3lmRdDBkZAqXTxaYD0XaN/:17448:0:99999:7:::  
student@student-vm:~$
```

[1] <http://www.cyberciti.biz/faq/understanding-etcshadow-file/>

Exercise Task 6

Part 2c: The /etc/shadow file

- ▶ What could happen if any users could read this file?

They would get the password hashes, and could start cracking the passwords.

- ▶ What could happen if they could write?

They could change the password hash (and salt) of any user to what they want, in effect setting the password.

Exercise Task 6

Part 2c: The /etc/shadow file

At the end of this file you should find your new users, and after the colon there is a prefix for the password between \$ that indicates which hashing algorithm being used, and the hashed password is written from this and until the next colon.

- ▶ Are the hashed passwords for Ole and Dole the same?

No, they are not.

- ▶ What does this mean? (Hint: `man crypt` in a terminal, and read the NOTES section.)

From `man crypt`:

So `5salt$encrypted` is an SHA-256 encoded password
and `6salt$encrypted` is an SHA-512 encoded one.

As Ole and Dole have different salts, their password hashes will be different.

Exercise Task 6

Part 2c: The /etc/shadow file

We have created (changed) the passwords today, so the “last password change field” should show (calculate to) today. Check if this is correct.

Tip: the command `date +%s` gives you seconds since 00:00:00, Jan 1, 1970 (a GNU extension).

```
ole:$6$9Roat28Y$qNKVHOPDuumeE9bkKdz.QUFaT56JvCuP0CHs9S13alryM/FgxLL6QGh5DL7JclgnaqKAU1dv7mB1mnhk4zJSy1:17448:0:99999:7:::  
dole:$6$nsSsUX5.Y$8Th8HlIImaPZAE5MP0kwVs.oNXa3KFY76ePuXlQpSFJK8Jwlv4Auue7UNSdktASRvfxMi6gay4fqhuNFrEfie0:17448:0:99999:7:::  
doffen:$6$ZFQ0FFV7$o3ZmK1D8cHV/AvmS84xJ.580rKR4WY50JGLyni/GI8FtPZnPxgk39MvmrfKDqWrC3lmRdDBkZAqXTxaYD0XaN/:17448:0:99999:7:::  
student@student-vm:~$ date +%s  
1507567261  
student@student-vm:~$ echo "1507567261 / (60*60*24)" | bc  
17448  
student@student-vm:~$
```

Exercise Task 6

Part 2d: The /etc/passwd file

Try to open the /etc/passwd file. Detailed information about this file can be found at [1] or by issuing the command `man 5 passwd` in a terminal (note the 5).

- ▶ What access do you have in user mode?

Read access.

- ▶ What could go wrong if users could write to this file?

They could alter the UID and GID for their username, in effect giving themselves elevated privileges.

Hint: What would happen if you change your group access field into someone else's number for group access?

[1] <http://www.cyberciti.biz/faq/understanding-etcpassword-file-format/>

Exercise Task 6

Part 2d: The /etc/passwd file

Open /etc/passwd with root privileges (`pkexec mousepad /etc/passwd`) and change the line for Doffen such that it contains Ole's number for identity and group. Change the path to home directory to /home/Ole. Your change should look something like this (the number may vary):

Ole:x:1001:1001::/home/Ole:/bin/bash

Doffen:x:1001:1001::/home/Ole:/bin/bash

Log in as Doffen (you can do a logout/login from the menu, or you can use the command `su -l doffen`).

- ▶ Do you get access to Oles home directory? **Yes.**
- ▶ Do a `whoami`. What was the result? **The result was ole.**
- ▶ Can you think of a way to change the etc/passwd file to prevent another user from getting access to his files? **Change his UID/GUID to another user.**

Undo the changes made to /etc/passwd before proceeding to the next section.

Exercise Task 6

Part 2e: Groups

Go back to the User Settings window and press the Manage Groups button. Add two groups called woodpeckers and scrimshanker. Put Ole and Dole in the group woodpeckers. Put Dole and Doffen in the group scrimshanker.

Login as Ole using `su -l ole`. Make a file called tent.make using the touch command:

```
touch tent.make
```

Look at the permissions of tent.make by using ls:

```
ls -l tent.make    -rw-rw-r-- 1 ole ole 0 okt. 9 18:58 tent.make
```

Login as Dole and try to access the file.

- ▶ Did it work? Yes.
- ▶ Can you both read from and write to the file? No, only read.
- ▶ Why/why not? dole is not the owner, nor part of the ole group, so the permission for others apply.

Exercise Task 6

Part 2e: Groups

We need to change the group ownership of the file. Log on as Ole again and type:

```
chown :woodpeckers tent.make
```

Check that Dole has full access to the file now. Yes, Dole has full access.

Check that group access is working by switching to Doffen and check that he has not write access to the file. No, Doffen does not have write access.

Exercise Task 6

Part 2f: Multi-group membership

Log in as Doffen and create the file dirty.trick in the home directory. Now, change the ownership and permissions so that anyone in the scrimshanker group have full access to this file. Now, switch to Dole and check that you can write to the file. Check Doles group identity with the command id.

- ▶ How are memberships in several groups handled in this system?

A user has one primary group, but can be member of several other groups. Any resource, however, belongs to one group only.

Exercise Task 7

Exercise Task 7

Part 1: Security related registry hives

1. Start Regedit as System: Psexec.exe -s -i regedit.exe
2. Use Process Explorer to confirm that it runs as System
3. Locate the SAM hive
4. Locate the System hive

Exercise Task 7

Part 1: Security related registry hives

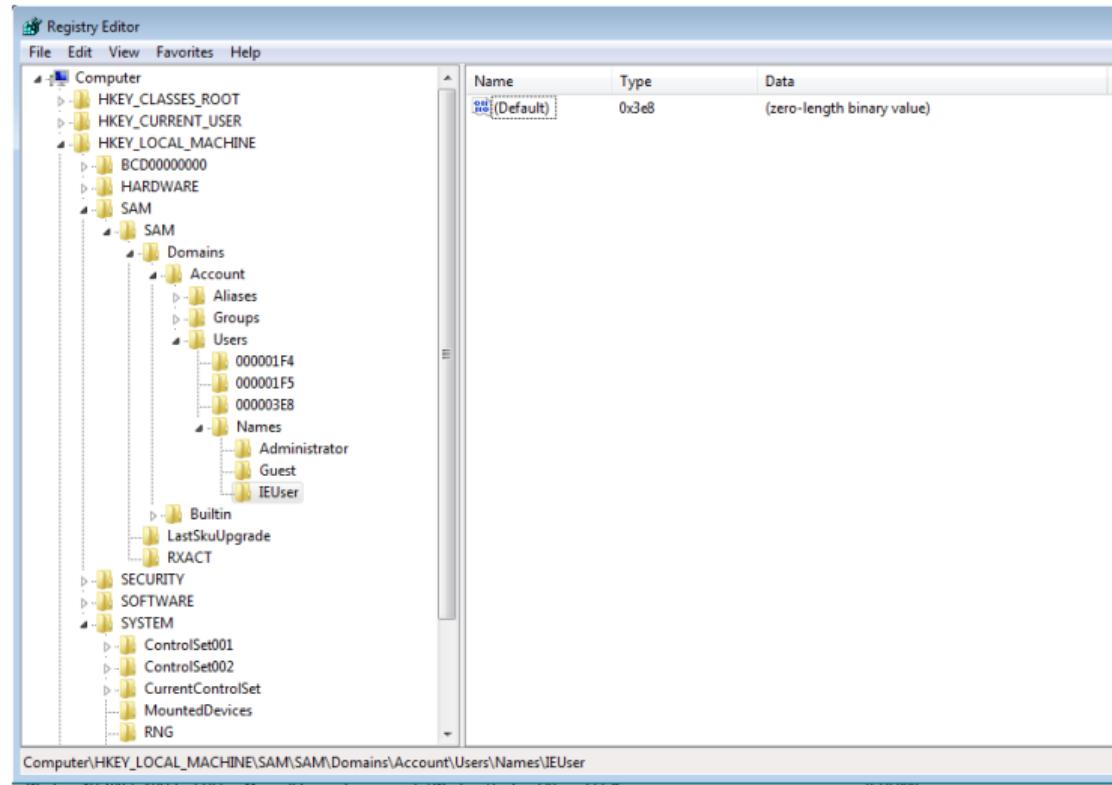
- 1 Start Regedit as System: Psexec.exe -s -i regedit.exe
- 2 Use Process Explorer to confirm that it runs as System

Process	CPU	Private Bytes	Working Set	PID	Description	Session	User Name	Path	Integrity	ASLR	Company Name
svchost.exe	0.02	28,768 K	17,284 K	1132	Host Process for Windows Services	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe	System	ASLR	Microsoft Corporation
spooler.exe	4,632 K	6,356 K	1244	Spooler SubSystem App	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\spooler.exe	System	ASLR	Microsoft Corporation	
svchost.exe	8,632 K	6,712 K	1284	Host Process for Windows Services	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe	System	ASLR	Microsoft Corporation	
vmicsvc.exe	0.02	2,180 K	4,344 K	1356	Virtual Machine Integration Component Service	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\vmicsvc.exe	System	ASLR	Microsoft Corporation
vmicsvc.exe	0.01	1,688 K	4,728 K	1376	Virtual Machine Integration Component Service	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\vmicsvc.exe	System	ASLR	Microsoft Corporation
vmicsvc.exe	0.01	1,156 K	3,332 K	1416	Virtual Machine Integration Component Service	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\vmicsvc.exe	System	ASLR	Microsoft Corporation
vmicsvc.exe	0.02	1,180 K	3,424 K	1444	Virtual Machine Integration Component Service	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\vmicsvc.exe	System	ASLR	Microsoft Corporation
vmicsvc.exe	0.01	1,200 K	3,480 K	1468	Virtual Machine Integration Component Service	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\vmicsvc.exe	System	ASLR	Microsoft Corporation
svchost.exe	3,736 K	7,308 K	1504	Host Process for Windows Services	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe	System	ASLR	Microsoft Corporation	
taskhost.exe	7,336 K	7,232 K	344	Host Process for Windows Tasks	1	IE11WIN7\IEUser	C:\Windows\System32\taskhost.exe	Medium	ASLR	Microsoft Corporation	
explorer.exe	0.19	28,384 K	37,344 K	1032	Windows Explorer	1	IE11WIN7\IEUser	C:\Windows\explorer.exe	Medium	ASLR	Microsoft Corporation
VBoxTray.exe	0.02	1,384 K	4,360 K	1104	VirtualBox Guest Additions Tray Application	1	IE11WIN7\IEUser	C:\Windows\System32\VBoxTray.exe	Medium	ASLR	Oracle Corporation
cmd.exe	1,796 K	2,272 K	3000	Windows Command Processor	1	IE11WIN7\IEUser	C:\Windows\System32\cmd.exe	High	ASLR	Microsoft Corporation	
PaExec.exe	1,656 K	4,356 K	3188	Execute processes remotely	1	IE11WIN7\IEUser	c:\sysint\PaExec.exe	High	ASLR	Sysinternals - www.aysinter...	
procexp.exe	4,172 K	13,252 K	19,096 K	2224	Syinternals Process Explorer	1	IE11WIN7\IEUser	C:\sysint\procexp.exe	High	ASLR	Sysinternals - www.aysinter...
SearchIndexer.exe	16,892 K	8,816 K	24,776 K	276	Microsoft Windows Search Indexer	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\SearchIndexer.exe	System	ASLR	Microsoft Corporation
conhost.exe	964 K	4,456 K	3008	Console Window Host	1	IE11WIN7\IEUser	C:\Windows\System32\conhost.exe	High	ASLR	Microsoft Corporation	
PSEXESVC.exe	892 K	2,780 K	3200	PsExec Service	0	NT AUTHORITY\SYSTEM	C:\Windows\PSEXESVC.exe	System	ASLR	Sysinternals	
regedit.exe	4,172 K	7,256 K	3268	Registry Editor	1	NT AUTHORITY\SYSTEM	C:\Windows\regedit.exe	System	ASLR	Microsoft Corporation	
svchost.exe	1,092 K	3,816 K	3852	Host Process for Windows Services	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe	System	ASLR	Microsoft Corporation	
WmiPrvSE.exe	0.01	1,752 K	4,604 K	2844	WMI Provider Host	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\wmiPrvSE.exe	System	ASLR	Microsoft Corporation

Exercise Task 7

Part 1: Security related registry hives

3 Locate the SAM hive



Exercise Task 7

Part 1: Security related registry hives

4 Locate the System hive

The screenshot shows the Windows Registry Editor window. The left pane displays a tree view of registry keys under the 'Computer' root key. The 'SYSTEM' key is expanded, showing subkeys like 'ControlSet001', 'ControlSet002', 'CurrentControlSet', 'MountedDevices', 'RNG', 'Select', and 'Setup'. The 'Setup' key is highlighted with a blue selection bar. The right pane contains a table listing registry values for the selected 'Setup' key.

Name	Type	Data
(Default)	REG_SZ	(value not set)
CloneTag	REG_MULTI_SZ	Mon Jul 13 21:55:53 2009
CmdLine	REG_SZ	
OOBEInProgress	REG_DWORD	0x00000000 (0)
OsLoaderPath	REG_SZ	\
RestartSetup	REG_DWORD	0x00000000 (0)
SetupPhase	REG_DWORD	0x00000000 (0)
SetupType	REG_DWORD	0x00000000 (0)
SystemPartition	REG_SZ	\Device\HarddiskVolume1
SystemSetupInP...	REG_DWORD	0x00000000 (0)
WorkingDirectory	REG_SZ	C:\Windows\Panther

Exercise Task 7

Part 1: Security related registry hives

5 Find your stored user password

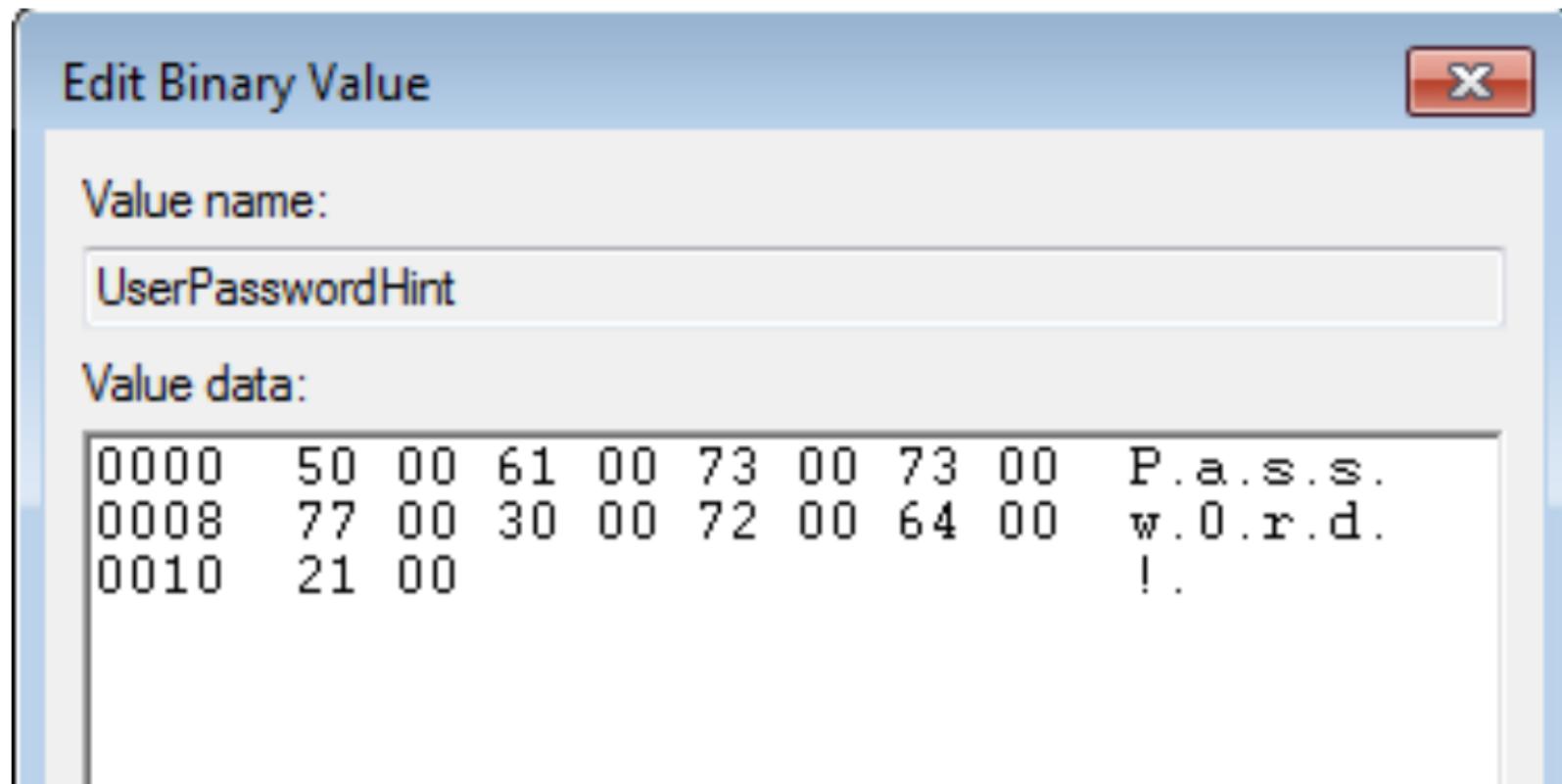
The screenshot shows the Windows Registry Editor interface. The left pane displays a tree view of registry keys under HKEY_LOCAL_MACHINE\SAM\Domains\Account\Users. Key nodes include Aliases, Groups, and several user accounts (000001F4, 000001F5, 000003E8). The 000003E8 key is expanded to show sub-nodes Names, Administrator, Guest, and IEUser. The right pane contains a table with three rows of data, corresponding to the user accounts in the tree:

Name	Type	Data
ab (Default)	REG_SZ	(value not set)
F	REG_BINARY	02 00 01 00 00 00 00 00 30 ab c6 cd b9 46 d3 01 00 0...
UserPasswordHint	REG_BINARY	50 00 61 00 73 00 73 00 77 00 30 00 72 00 64 00 21 00
V	REG_BINARY	00 00 00 00 bc 00 00 00 02 00 01 00 bc 00 00 00 0c 0...

Exercise Task 7

Part 1: Security related registry hives

- 5 Find your stored user password



Exercise Task 7

Part 2: Tokens

1. Start cmd.exe as normal user, Administrator and as System. (cmd as System: Psexec.exe -s -i cmd.exe)
2. Use Process Explorer to confirm that they run with Low, Medium and High integrity, respectively.
3. Use Process Hacker to investigate the tokens of the three processes.
4. Compare the tokens. What do you see?

Process Hacker: <https://wj32.org/processhacker/nightly.php>

Exercise Task 7

Part 2: Tokens

- 1 Start cmd.exe as normal user, Administrator and as System. (cmd as System: Psexec.exe -s -i cmd.exe)
- 2 Use Process Explorer to confirm that they run with Low, Medium and High integrity, respectively.

Process	CPU	Private Bytes	Working Set	PID	Description	Session	User Name	Path	Integrity	ASLR	Company Name	
svchost.exe		2,208 K	4,812 K	684	Host Process for Windows Services	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe	System	ASLR	Microsoft Corporation	
svchost.exe		13,752 K	11,056 K	736	Host Process for Windows Services	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe	System	ASLR	Microsoft Corporation	
audiodg.exe		15,000 K	13,804 K	2604	Windows Audio Device Graph Isolation	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\audiodg.exe	System	ASLR	Microsoft Corporation	
svchost.exe		36,632 K	40,476 K	856	Host Process for Windows Services	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe	System	ASLR	Microsoft Corporation	
dwm.exe	< 0.01	1,012 K	3,304 K	252	Desktop Window Manager	1 IE11WIN7\IEUser		C:\Windows\System32\dwm.exe	Medium	ASLR	Microsoft Corporation	
svchost.exe		4,616 K	9,196 K	896	Host Process for Windows Services	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe	System	ASLR	Microsoft Corporation	
svchost.exe	0.01	17,512 K	26,688 K	940	Host Process for Windows Services	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe	System	ASLR	Microsoft Corporation	
svchost.exe	0.04	11,212 K	9,888 K	1152	Host Process for Windows Services	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe	System	ASLR	Microsoft Corporation	
spooler.exe		4,640 K	6,388 K	1252	Spooler SubSystem App	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\spooler.exe	System	ASLR	Microsoft Corporation	
svchost.exe		8,780 K	6,784 K	1288	Host Process for Windows Services	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe	System	ASLR	Microsoft Corporation	
vmicsvc.exe	0.01	2,244 K	4,432 K	1360	Virtual Machine Integration Component Service	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\vmicsvc.exe	System	ASLR	Microsoft Corporation	
vmicsvc.exe	0.01	1,756 K	4,820 K	1380	Virtual Machine Integration Component Service	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\vmicsvc.exe	System	ASLR	Microsoft Corporation	
vmicsvc.exe	0.01	1,220 K	3,428 K	1420	Virtual Machine Integration Component Service	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\vmicsvc.exe	System	ASLR	Microsoft Corporation	
vmicsvc.exe	0.02	1,248 K	3,520 K	1448	Virtual Machine Integration Component Service	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\vmicsvc.exe	System	ASLR	Microsoft Corporation	
vmicsvc.exe	0.01	1,260 K	3,572 K	1472	Virtual Machine Integration Component Service	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\vmicsvc.exe	System	ASLR	Microsoft Corporation	
svchost.exe		3,652 K	7,384 K	1508	Host Process for Windows Services	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe	System	ASLR	Microsoft Corporation	
taskhost.exe	0.02	15,600 K	15,236 K	1864	Host Process for Windows Tasks	1 IE11WIN7\IEUser		C:\Windows\System32\taskhost.exe	Medium	ASLR	Microsoft Corporation	
explorer.exe	3.76	28,980 K	35,932 K	344	Windows Explorer	1 IE11WIN7\IEUser		C:\Windows\explorer.exe	Medium	ASLR	Microsoft Corporation	
VBoxTray.exe		0.02	1,368 K	4,424 K	512	VirtualBox Guest Additions Tray Application	1 IE11WIN7\IEUser		C:\Windows\System32\VBoxTray.exe	Medium	ASLR	Oracle Corporation
cmd.exe		1,820 K	2,260 K	4036	Windows Command Processor	1 IE11WIN7\IEUser		C:\Windows\System32\cmd.exe	Medium	ASLR	Microsoft Corporation	
cmd.exe		1,864 K	2,352 K	4060	Windows Command Processor	1 IE11WIN7\IEUser		C:\Windows\System32\cmd.exe	High	ASLR	Microsoft Corporation	
PsExec.exe		1,716 K	4,440 K	2104	Execute processes remotely	1 IE11WIN7\IEUser		c:\sysnt\PsExec.exe	High	ASLR	Sysinternals - www.sysinternals...	
procexp.exe	8.58	13,392 K	19,400 K	2316	Sytematic Process Explorer	1 IE11WIN7\IEUser		c:\sysnt\procexp.exe	High	ASLR	Sysinternals - www.sysinternals...	
explorer.exe	0.03	7,600 K	19,044 K	3804	Internet Explorer	1 IE11WIN7\IEUser		C:\Program Files\Internet Explorer\explorer.exe	Medium	ASLR	Microsoft Corporation	
explore.exe	0.01	19,264 K	34,244 K	2872	Internet Explorer	1 IE11WIN7\IEUser		C:\Program Files\Internet Explorer\explore.exe	Low	ASLR	Microsoft Corporation	
SearchIndexer.exe		22,820 K	14,172 K	2528	Microsoft Windows Search Indexer	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\SearchIndexer.exe	System	ASLR	Microsoft Corporation	
conhost.exe		1,512 K	6,944 K	1700	Console Window Host	1 IE11WIN7\IEUser		C:\Windows\System32\conhost.exe	Medium	ASLR	Microsoft Corporation	
conhost.exe		968 K	4,524 K	968	Console Window Host	1 IE11WIN7\IEUser		C:\Windows\System32\conhost.exe	High	ASLR	Microsoft Corporation	
PSEXECV.exe		920 K	2,772 K	2948	PsExec Service	0	NT AUTHORITY\SYSTEM	C:\Windows\PSEXECV.exe	System	ASLR	Sysinternals	
cmd.exe		1,708 K	2,076 K	3648	Windows Command Processor	1 NT AUTHORITY\SYSTEM		C:\Windows\System32\cmd.exe	System	ASLR	Microsoft Corporation	
conhost.exe		904 K	4,248 K	3500	Console Window Host	1 NT AUTHORITY\SYSTEM		C:\Windows\System32\conhost.exe	System	ASLR	Microsoft Corporation	
WmiPrvSE.exe		1,760 K	4,596 K	2280	WMI Provider Host	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\wmiPrvSE.exe	System	ASLR	Microsoft Corporation	

Exercise Task 7

Part 2: Tokens

- 3 Use Process Hacker to investigate the tokens of the three processes.
- 4 Compare the tokens. What do you see?

The image displays three side-by-side windows from the Process Hacker application, each showing the "Token" tab of a process's properties dialog. The processes are cmd.exe with PIDs 4036, 4060, and 3648 respectively.

cmd.exe (4036) Properties

Name	Flags
BUILTIN\Administrators	Use for deny only (disabled)
BUILTIN\Users	Mandatory (default enabled)
CONSOLE LOGON	Mandatory (default enabled)
Everyone	Mandatory (default enabled)
IE11WIN7\None	Mandatory (default enabled)
LOCAL	Mandatory (default enabled)
Mandatory Label\Medium Ma...	Integrity
NT AUTHORITY\Authenticat...	Mandatory (default enabled)
NT AUTHORITY\INTERACTIVE	Mandatory (default enabled)

cmd.exe (4060) Properties

Name	Flags
BUILTIN\Administrators	Mandatory (default enabled)
BUILTIN\Users	Mandatory (default enabled)
CONSOLE LOGON	Mandatory (default enabled)
Everyone	Mandatory (default enabled)
IE11WIN7\None	Mandatory (default enabled)
LOCAL	Mandatory (default enabled)
Mandatory Label\High Mand...	Integrity
NT AUTHORITY\Authenticat...	Mandatory (default enabled)
NT AUTHORITY\INTERACTIVE	Mandatory (default enabled)

cmd.exe (3648) Properties

Name	Flags
BUILTIN\Administrators	Owner (default enabled)
Everyone	Mandatory (default enabled)
Mandatory Label\System Ma...	Integrity
NT AUTHORITY\Authenticat...	Mandatory (default enabled)