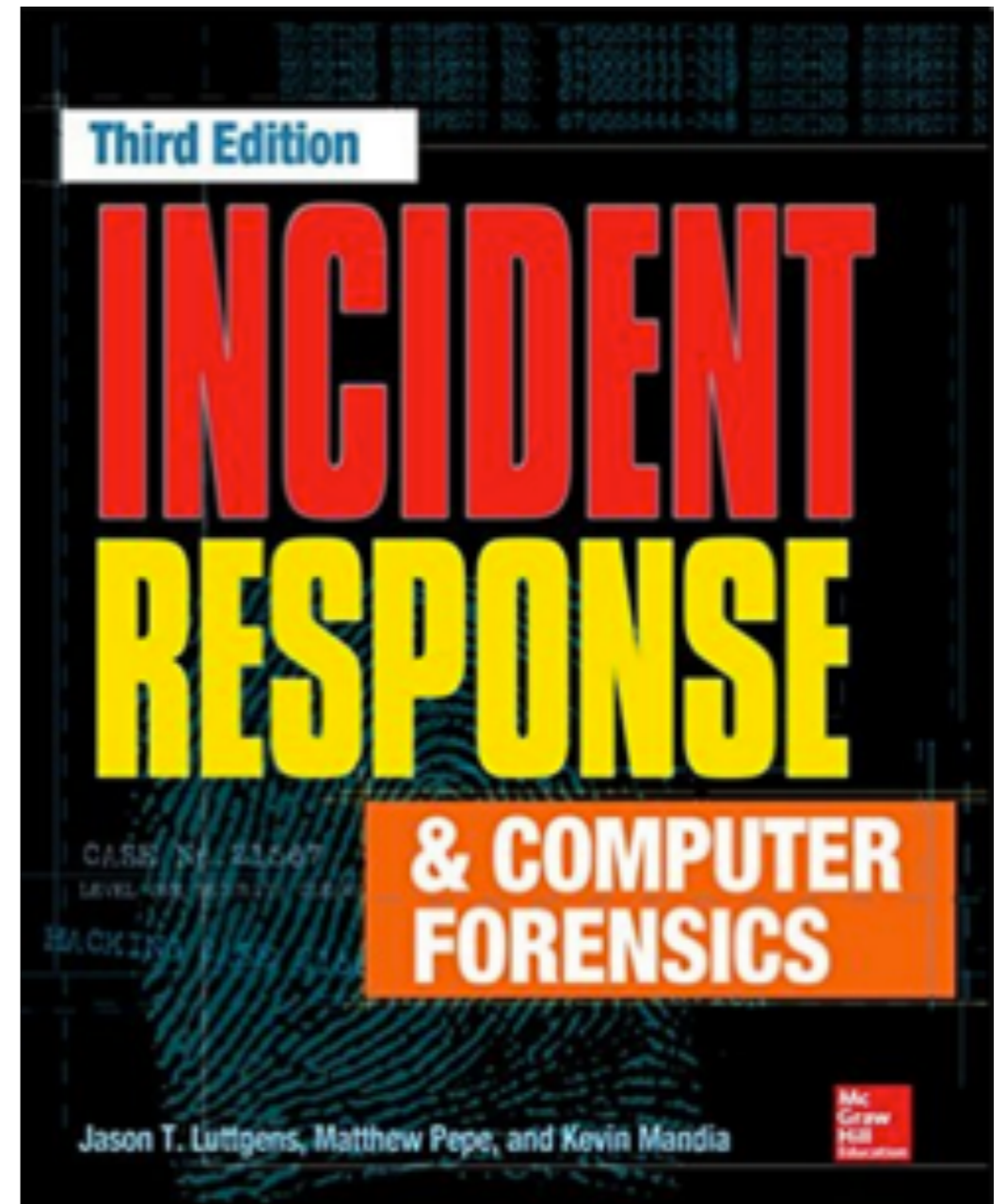


CNIT 152: Incident Response



12 Investigating Windows Systems (Part 3)

Updated 11-11-21

- Other artifacts of interactive sessions
- Memory forensics
- Alternative persistence mechanisms

Other Artifacts of Interactive Sessions

Interactive Sessions

- **For the purposes of this section, includes**
 - **Login with user at the console**
 - **Remote Desktop sessions**
 - **Screen sharing (via VNC or similar software)**

LNK Files

- **Shortcuts to files**
- **Serve as extensions to Windows Explorer**
- **Windows automatically creates LNKs for every opened file**
 - **To populate "Recent Files"**
- **Separate list in each user profile**

Where the LNK Files Are

- **C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Windows\Recent**
- **C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Office\Recent**

Evidence in LNK Files

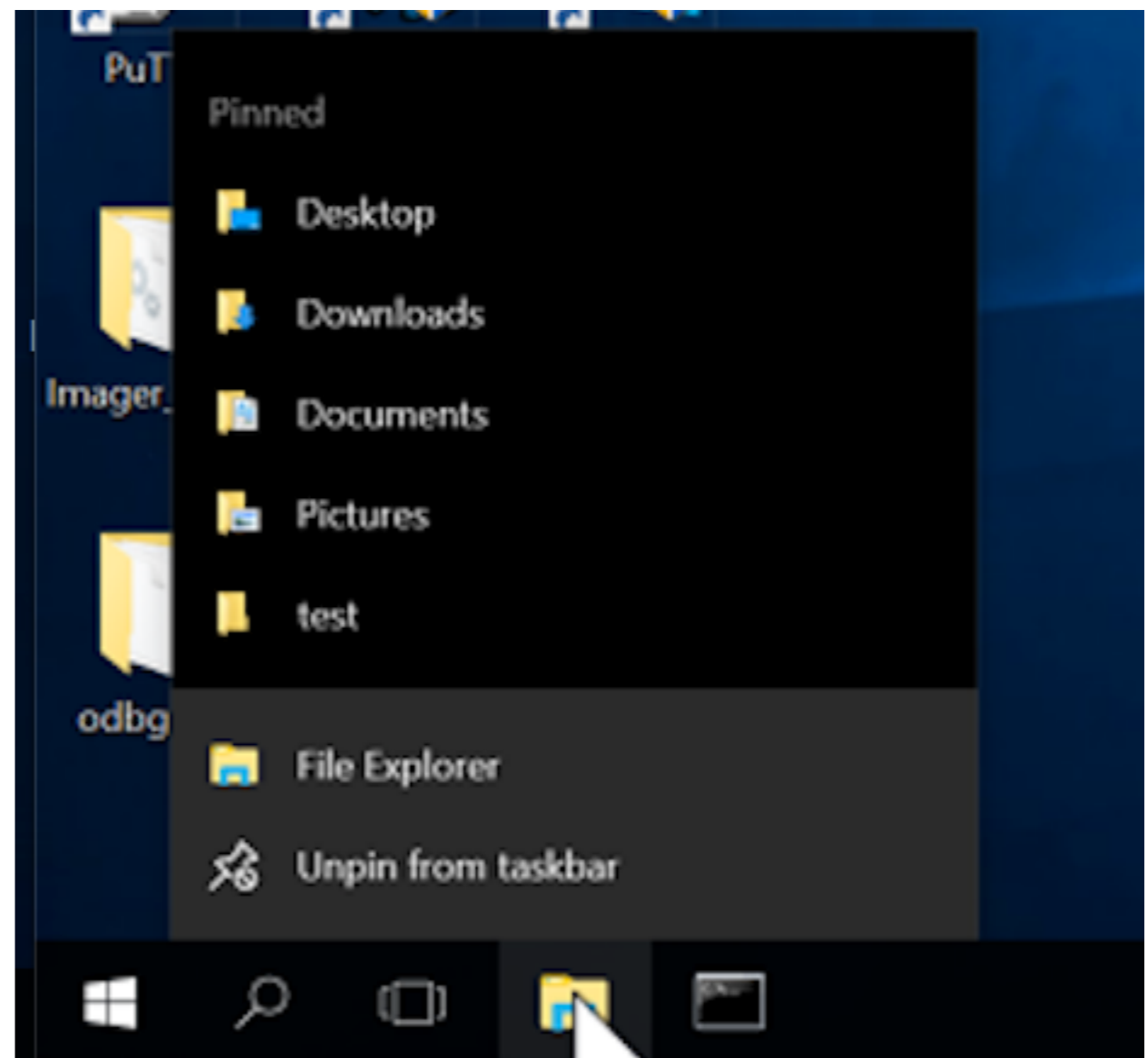
- Full file path (at the time the link was created)
- Network share name (if target file originated from such a source)
- Serial number for the source volume
- Attributes and logical size
- Standard Information Modified, Accessed, and Created timestamps for the referenced file at the time it was last opened
- A unique object identifier (ObjectID), also stored in the target file's MFT record and used by the Distributed Link Tracking service

Timeline

- **LNK files can show just what a user did**
- **Which files were accessed, and in what order**

Jump Lists

- **Right-click a taskbar icon to show recently used items**
- **Word shows recent Word files, etc.**



Where Jump Lists are Stored

C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations

C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations

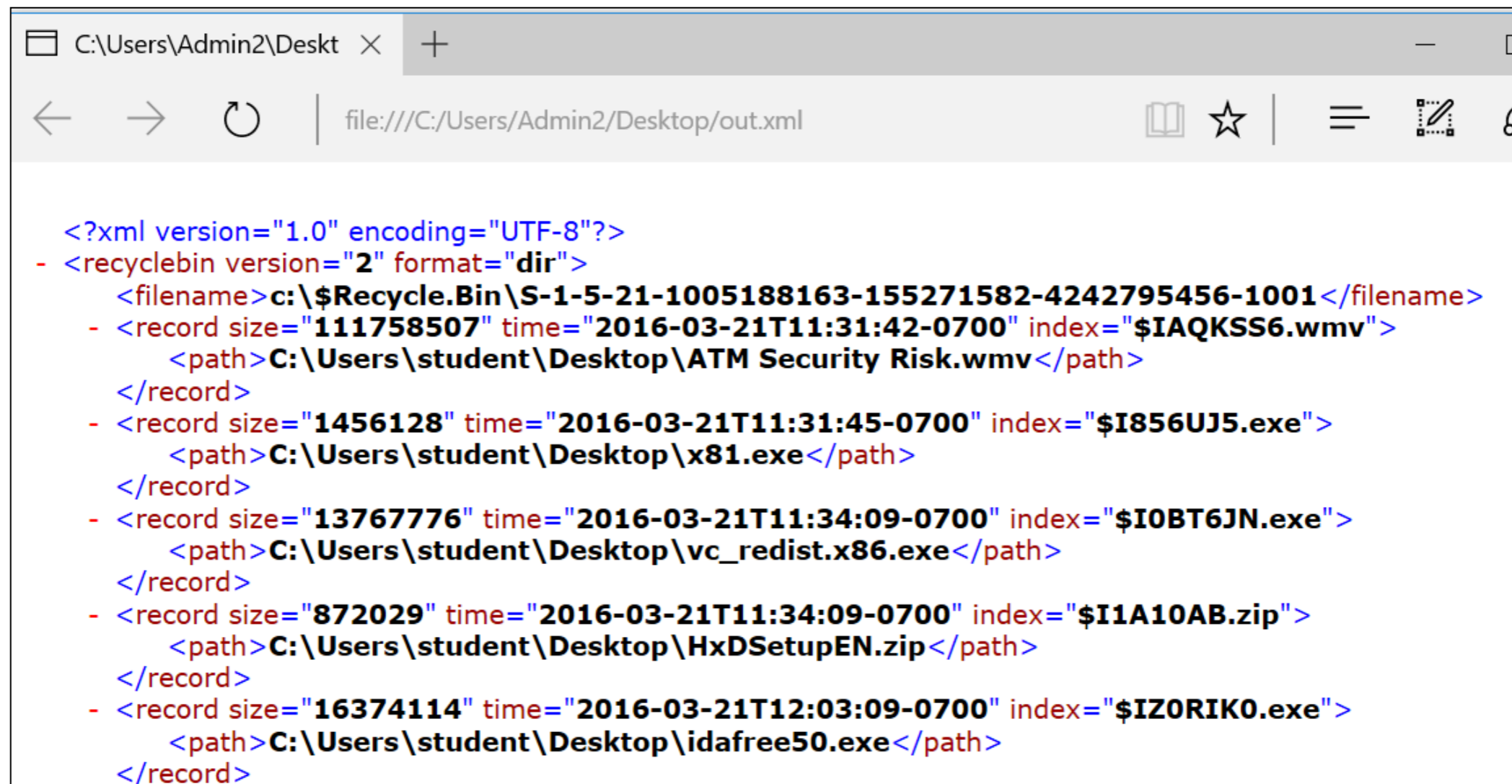
- **Not human-readable, you need tools**
 - **JumpLister for Windows 7-8**
 - **JLECmd for Windows 10 (link Ch 12t)**

The Recycle Bin

- **Located in `\$Recycle.Bin`**
- **Contains files deleted from the hard disk**
 - **But not if deleted from removable drives**
 - **Or from the Command Prompt**
 - **Or with Shift+Delete**

Rifiuti2 Tool

```
C:\Users\Admin2\Desktop\rifiuti2-0.6.1-win\x86>rifiuti-vista.exe --output=out --xml --localtime  
c:\$Recycle.Bin\S-1-5-21-1005188163-155271582-4242795456-1001
```



```
<?xml version="1.0" encoding="UTF-8"?>  
- <recyclebin version="2" format="dir">  
  <filename>c:\$Recycle.Bin\S-1-5-21-1005188163-155271582-4242795456-1001</filename>  
  - <record size="111758507" time="2016-03-21T11:31:42-0700" index="$IAQKSS6.wmv">  
    <path>C:\Users\student\Desktop\ATM Security Risk.wmv</path>  
  </record>  
  - <record size="1456128" time="2016-03-21T11:31:45-0700" index="$I856UJ5.exe">  
    <path>C:\Users\student\Desktop\x81.exe</path>  
  </record>  
  - <record size="13767776" time="2016-03-21T11:34:09-0700" index="$I0BT6JN.exe">  
    <path>C:\Users\student\Desktop\vc_redist.x86.exe</path>  
  </record>  
  - <record size="872029" time="2016-03-21T11:34:09-0700" index="$I1A10AB.zip">  
    <path>C:\Users\student\Desktop\HxDSetupEN.zip</path>  
  </record>  
  - <record size="16374114" time="2016-03-21T12:03:09-0700" index="$IZORIK0.exe">  
    <path>C:\Users\student\Desktop\idafree50.exe</path>  
  </record>
```

- **Link Ch 12u**

Velociraptor

Velociraptor Response and Mon +

172.16.123.38:8889/app/index.html#/collected/C.57ac19bf16a58b0a

all Show All Win10SUS connected admin

New Collection: Select Artifacts to collect

- Windows.Forensics.RecycleBin

Windows.Forensics.RecycleBin

Type: client
by Zach Stanford - @svch0st

This artefact will parse the **\$I** files found in the **\$Recycle.Bin** folder to obtain the time of deletion and the original path and file name.

Supports Recycle Bin format found in Vista onwards. This will not parse INFO2 files found in the "Recycler" folder from XP and below.

The layout of the Recycle Bin folder is in the in the form:

- 1 C:\\$Recycle.Bin\%SID%\

Each folder contains the following files:

- 1 \$R##### files; the original data
- 2 \$I##### files; the "Recycled" file's metadata

Select Artifacts Configure Parameters Specify Resources Review Launch

2021-11-11T23:16:16.354Z

Memory Forensics

Evidence in RAM

- Running processes and the system objects/resources with which they interact
- Active network connections
- Loaded drivers
- User credentials (which may be hashed, obfuscated, or even appear in clear text)
- Portions of nonvolatile sources of evidence such as the registry, event log, and Master File Table
- Remnants of previously executed console commands
- Remnants of clear-text data that is otherwise encrypted on disk
- Important data structures within the kernel that provide insight into process accounting, behavior, and execution

Types of Memory

- **Physical (RAM chips)**
- **Page file**
 - **Data moved out of RAM onto the hard disk**
 - **%SYSTEMDRIVE%\pagefile.sys**

Crash Dumps

- **Can be produced when Windows crashes with the "Blue-Screen of Death"**
- **Three levels**
 - **Kernel Memory Dump (default)**
 - **Small Memory Dump (Minidump)**
 - **Complete Memory Dump**

Crash Dump Storage

- **%LOCALAPPDATA%\Crashdumps**
- **Complete Memory Dump is most useful type**
- **But it's rarely turned on**

Hibernation Files

- **Saves the full contents of RAM on disk**
 - **%SYSTEMDRIVE%\Hiberfil.sys**
- **It's compressed and includes metadata**
 - **Link Ch 8t**
- **Volatility can parse it**

Running Processes

- **Volatility can recover**
 - **Process ID (PID)** A unique numeric identifier assigned upon process startup
 - **Parent PID** The ID of the process that was responsible for executing the current process
 - **Process name** The executable file's name
 - **Process path** The fully qualified path to the executable file
 - **Process command line** Any argument parameters supplied in the executable's command line
 - **Process start and exit times** If applicable
 - **Number of threads and handles**

```

root@kali:/usr/share/volatility# python vol.py pslist --profile=Win2008SP1x86 -f /root/Desktop/memdump.mem
Volatility Foundation Volatility Framework 2.3.1
*** Failed to import volatility.plugins.addrspaces.legacyintel (AttributeError: 'module' object has no attri
Offset(V)  Name                PID  PPID  Thds  Hnds  Sess  Wow64  Start
-----
0x82db0910 System                4    0    100   541  -----  0  2014-01-08 02:17:35 UTC+0000
0x8454c118 smss.exe              404   4     4     28  -----  0  2014-01-08 02:17:35 UTC+0000
0x84561968 csrss.exe             472  460    11   466    0  0  2014-01-08 02:17:35 UTC+0000
0x84450770 csrss.exe             516  508    10   305    1  0  2014-01-08 02:17:36 UTC+0000
0x84453770 wininit.exe           524  460     3     98    0  0  2014-01-08 02:17:36 UTC+0000
0x84465770 winlogon.exe          552  508     3    116    1  0  2014-01-08 02:17:36 UTC+0000
0x83632170 services.exe         604  524     6    250    0  0  2014-01-08 02:17:36 UTC+0000
0x844bf770 lsass.exe             616  524    13    610    0  0  2014-01-08 02:17:36 UTC+0000
0x844c2680 lsm.exe               624  524    10    208    0  0  2014-01-08 02:17:36 UTC+0000
0x84866d50 svchost.exe           788  604     6    298    0  0  2014-01-08 02:17:42 UTC+0000
0x845f37a8 svchost.exe           848  604     8    280    0  0  2014-01-08 02:17:42 UTC+0000
0x848fa118 svchost.exe           884  604    15    274    0  0  2014-01-08 02:17:42 UTC+0000
0x84914d90 svchost.exe           976  604     6    152    0  0  2014-01-08 02:17:42 UTC+0000
0x8491bd90 svchost.exe          1000  604    45   2072    0  0  2014-01-08 02:17:42 UTC+0000
0x8492a6d0 SLsvc.exe             1056  604     4     96    0  0  2014-01-08 02:17:42 UTC+0000
0x84937d90 svchost.exe          1088  604    17    567    0  0  2014-01-08 02:17:42 UTC+0000
0x84941d90 svchost.exe          1160  604    20    265    0  0  2014-01-08 02:17:42 UTC+0000
0x84945c30 svchost.exe          1188  604    22    596    0  0  2014-01-08 02:17:42 UTC+0000
0x8496e9f0 svchost.exe          1308  604    17    265    0  0  2014-01-08 02:17:43 UTC+0000
0x849c18a8 spoolsv.exe           1424  604    17    291    0  0  2014-01-08 02:17:49 UTC+0000
0x849d7610 armsvc.exe           1460  604     2     56    0  0  2014-01-08 02:17:49 UTC+0000
0x849dcd90 dns.exe               1480  604    10    164    0  0  2014-01-08 02:17:49 UTC+0000
0x849e1cc0 ftpbasicssvr.exe     1508  604     2     52    0  0  2014-01-08 02:17:49 UTC+0000
0x849f5888 svchost.exe           1576  604     5    124    0  0  2014-01-08 02:17:49 UTC+0000
0x849faad8 svchost.exe           1604  604     3     73    0  0  2014-01-08 02:17:49 UTC+0000

```

Kahoot!

12c-1

Handles

- **Used to access files, devices, and more from software**
- **Can help when analyzing malware**
- **Mutants or Mutexes are used for inter-process communication**
 - **To lock a resource so no other process changes it while it's in use**
 - **Used by malware to prevent re-infection**

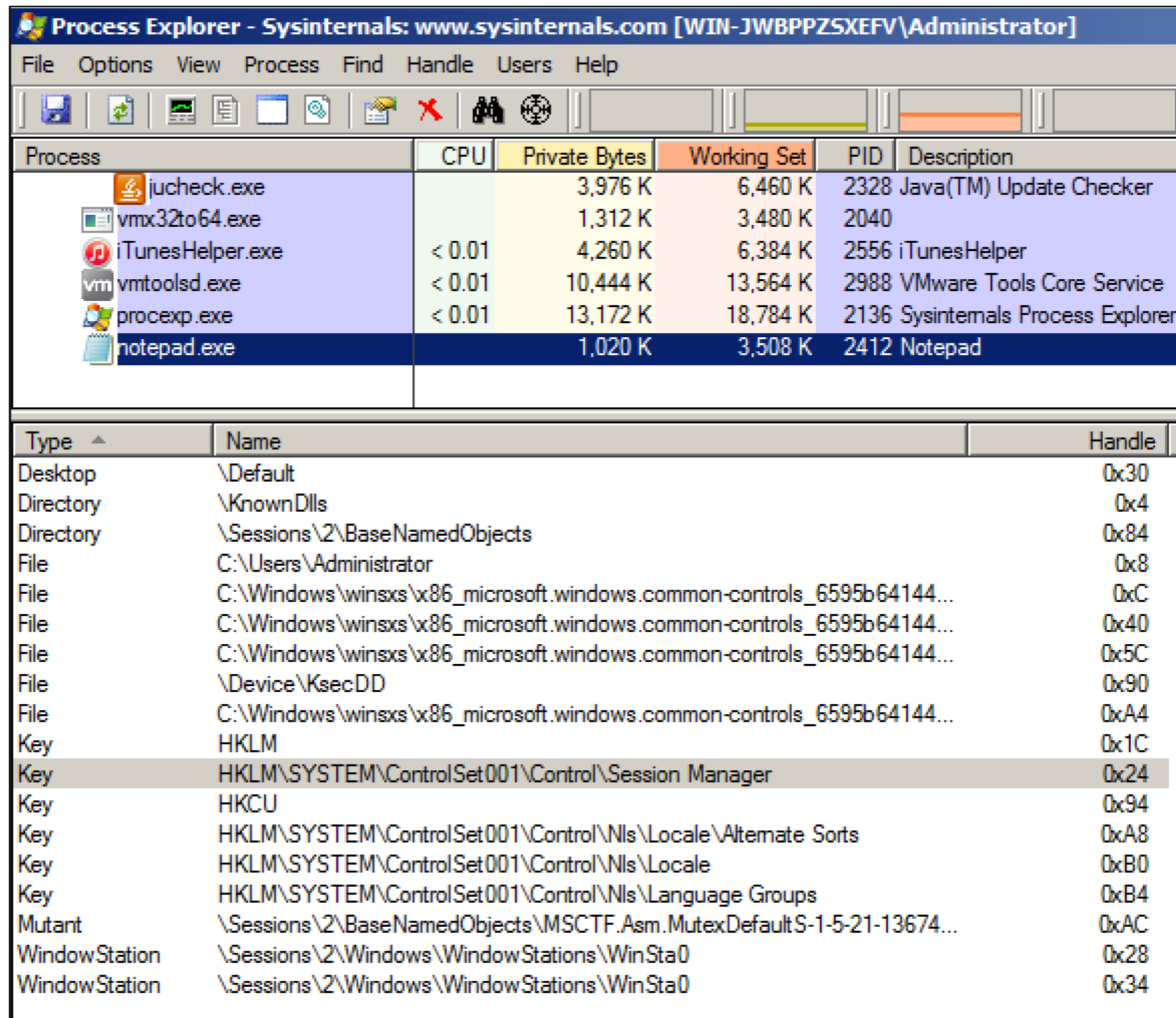
Handles for Zeus

```
$python vol.py -f ~/zeus.vmem handles -p 856 >> ~/zeusfiles/pid-856-  
handles.txt
```

```
0xff21b438 856 0x430 0x1f0003 Event  
0xff1398c0 856 0x434 0x12019f File \Device\NamedPipe\_AVIRA_2108  
0xff2a0e98 856 0x438 0x1f0003 Event  
0xff27b7e8 856 0x43c 0x1f0001 Mutant _AVIRA_2108  
0xff1b8638 856 0x440 0x1f03ff Thread TID 2004 PID 856  
0xff23d020 856 0x444 0x1f03ff Thread TID 2000 PID 856  
0x80f19200 856 0x450 0x1f0001 Mutant  
0xff1db540 856 0x454 0x100020 File \Device\HarddiskVolume1\WINDOWS  
indows.Common-Controls_6595b64144ccf1df_6.0.2600.2180_x-ww_a84f1ff9
```

- **Mutant AVIRA_2108 is a fingerprint of Zeus**
- **Link Ch 12u**

Handles for Notepad



The image shows a screenshot of Process Explorer from Sysinternals. The top window displays a list of running processes. The bottom window shows the handles for the selected process, notepad.exe.

Process	CPU	Private Bytes	Working Set	PID	Description
jucheck.exe		3,976 K	6,460 K	2328	Java(TM) Update Checker
vmx32to64.exe		1,312 K	3,480 K	2040	
iTunesHelper.exe	< 0.01	4,260 K	6,384 K	2556	iTunesHelper
vmtoolsd.exe	< 0.01	10,444 K	13,564 K	2988	VMware Tools Core Service
procexp.exe	< 0.01	13,172 K	18,784 K	2136	Sysinternals Process Explorer
notepad.exe		1,020 K	3,508 K	2412	Notepad

Type	Name	Handle
Desktop	\Default	0x30
Directory	\KnownDlls	0x4
Directory	\Sessions\2\BaseNamedObjects	0x84
File	C:\Users\Administrator	0x8
File	C:\Windows\winsxs\x86_microsoft.windows.common-controls_6595b64144...	0xC
File	C:\Windows\winsxs\x86_microsoft.windows.common-controls_6595b64144...	0x40
File	C:\Windows\winsxs\x86_microsoft.windows.common-controls_6595b64144...	0x5C
File	\Device\KsecDD	0x90
File	C:\Windows\winsxs\x86_microsoft.windows.common-controls_6595b64144...	0xA4
Key	HKLM	0x1C
Key	HKLM\SYSTEM\ControlSet001\Control\Session Manager	0x24
Key	HKCU	0x94
Key	HKLM\SYSTEM\ControlSet001\Control\Nls\Locale\Alternate Sorts	0xA8
Key	HKLM\SYSTEM\ControlSet001\Control\Nls\Locale	0xB0
Key	HKLM\SYSTEM\ControlSet001\Control\Nls\Language Groups	0xB4
Mutant	\Sessions\2\BaseNamedObjects\MSCTF.Asm.MutexDefaultS-1-5-21-13674...	0xAC
Window Station	\Sessions\2\Windows\WindowStations\WinSta0	0x28
Window Station	\Sessions\2\Windows\WindowStations\WinSta0	0x34

Velociraptor

Velociraptor Response and Mon +

172.16.123.38:8889/app/index.html#/collected/C.57ac19bf16a58b0a

mut

Show All

Win10SUS connected

admin

New Collection: Select Artifacts to collect

mut

Windows.Detection.Mutants

Windows.Detection.Mutants

Type: client

Enumerate the mutants from selected processes.

Mutants are often used by malware to prevent re-infection.

Parameters

Name	Type	Default	Description
processRegex		.	A regex applied to process names.
MutantNameRegex		.+	

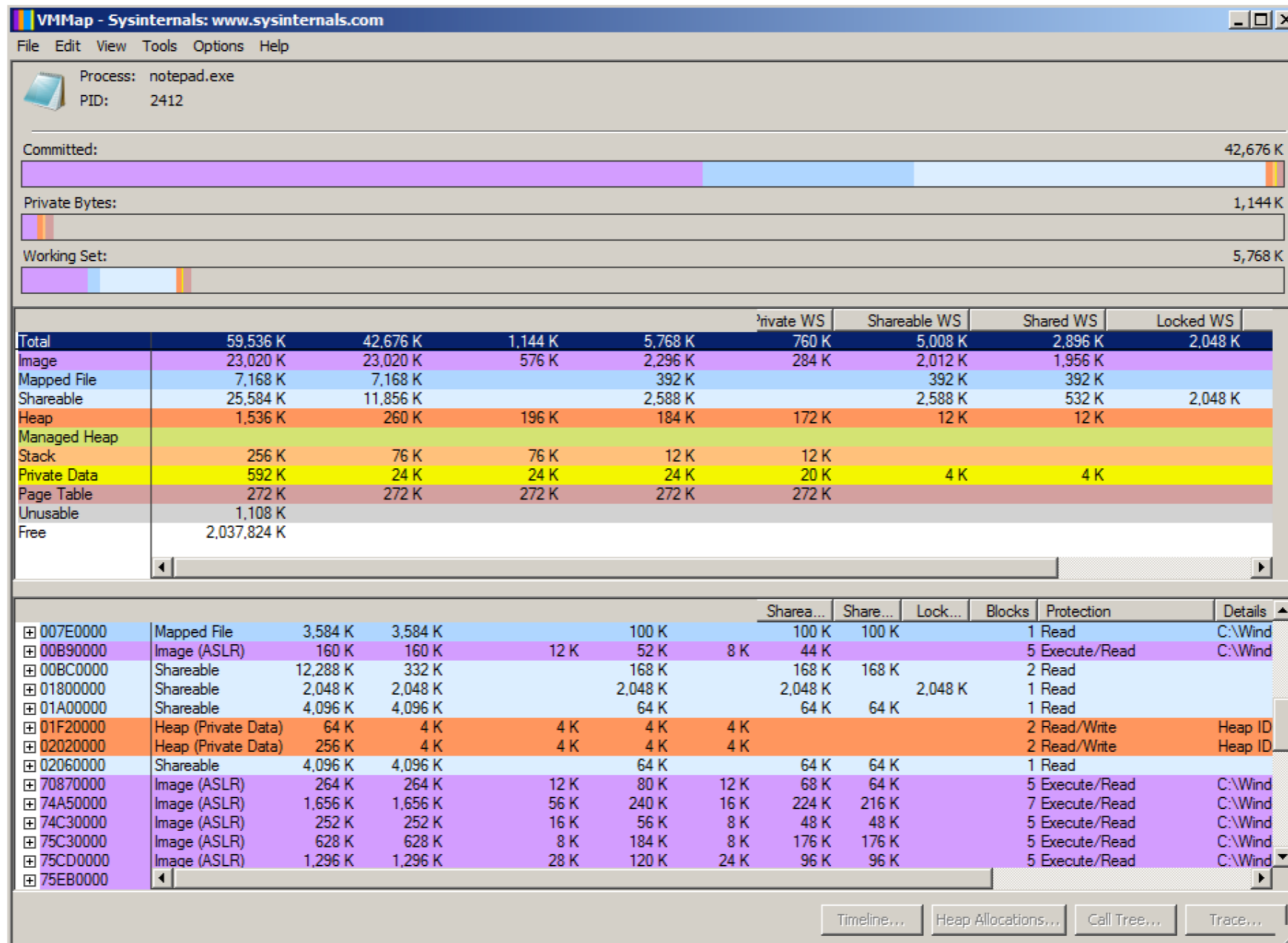
Select Artifacts Configure Parameters Specify Resources Review Launch

2021-11-11T23:21:30.310Z

Memory Sections

- **Each process has a virtual address space**
 - **Including RAM and some disk space in the pagefile**
- **The OS swaps data in and out of physical memory**
- **Virtual Address Descriptor (VAD) tree**
 - **A kernel data structure that shows how memory is used by each process (link Ch 12v)**

Memory Map for Notepad



Velociraptor

The screenshot shows the Velociraptor web interface in a browser window. The address bar displays the URL `172.16.123.38:8889/app/index.html#/collected/C.57ac19bf16a58b0a`. The interface includes a sidebar with navigation icons, a search bar, and a main content area. A modal dialog box titled "New Collection: Select Artifacts to collect" is open, showing a list of artifacts on the left and a detailed view of the selected artifact, "Windows.System.VAD", on the right. The artifact details include its type ("client"), a description ("Enumerate the memory regions of each running process."), and a table of parameters. The "processRegex" parameter is shown with a default value of "." and a description: "A regex applied to process names." At the bottom of the dialog, there are five buttons: "Select Artifacts", "Configure Parameters", "Specify Resources", "Review", and "Launch". The "Select Artifacts" button is highlighted in blue. The background interface shows a connection to "Win10SUS" and a user named "admin".

Velociraptor Response and Mon +

172.16.123.38:8889/app/index.html#/collected/C.57ac19bf16a58b0a

mut Show All Win10SUS connected admin

New Collection: Select Artifacts to collect

- Windows.Memory.Intezer
- Windows.Remediation.Quarantine
- Windows.Sys.PhysicalMemoryRanges
- Windows.System.VAD**
- Windows.Triage.ProcessMemory

Windows.System.VAD
Type: client
Enumerate the memory regions of each running process.

Parameters

Name	Type	Default	Description
processRegex		.	A regex applied to process names.

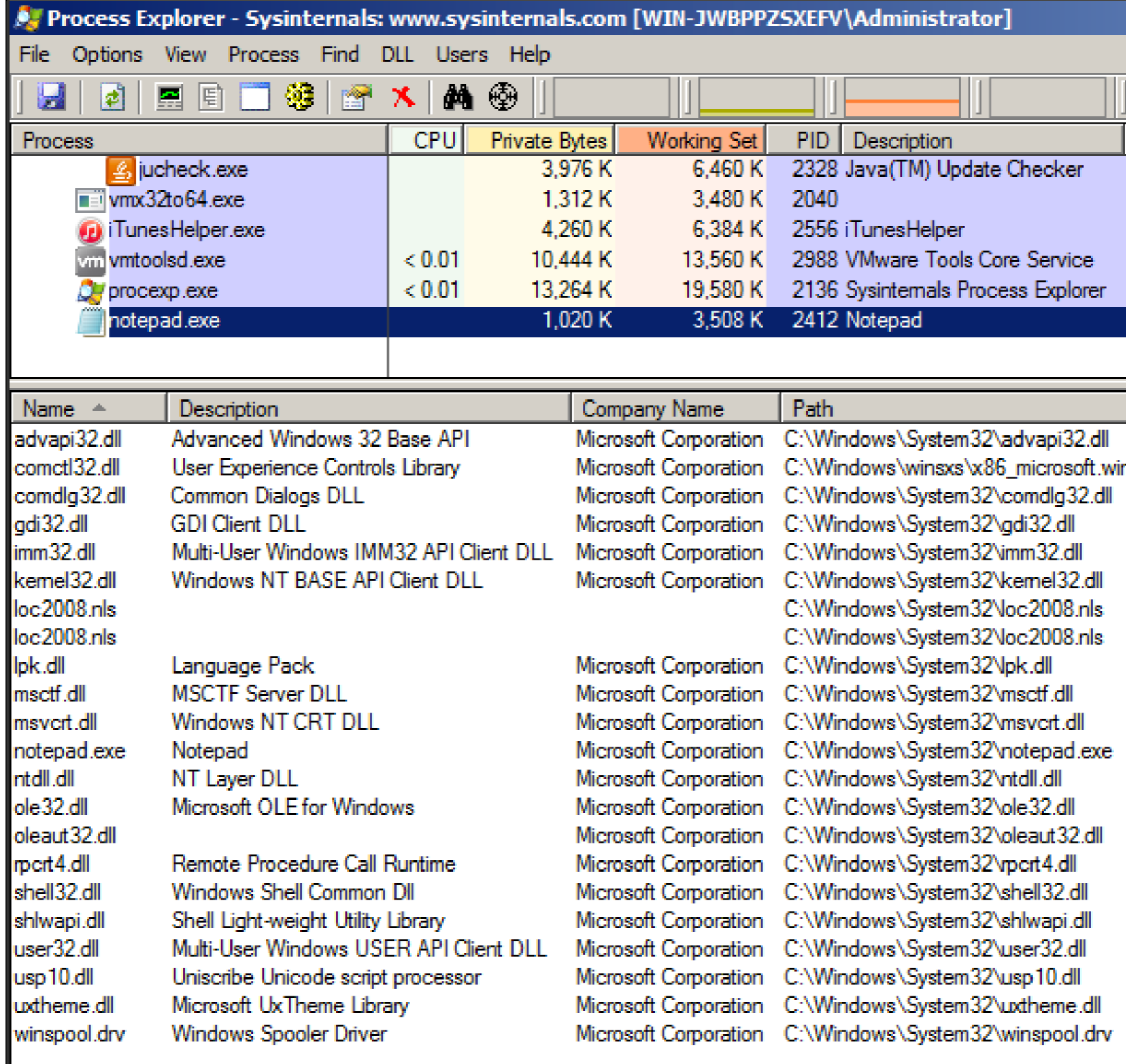
Select Artifacts Configure Parameters Specify Resources Review Launch

2021-11-11T23:25:31.311Z

Detecting Malicious DLLs

- **Check for valid digital signatures**
- **Known-good or -bad hash values**
- **Evidence of process-tampering attacks**
 - **Malware loading a DLL surreptitiously or running code in memory**

DLLs for Notepad



The screenshot shows Process Explorer with the following processes listed:

Process	CPU	Private Bytes	Working Set	PID	Description
jucheck.exe		3,976 K	6,460 K	2328	Java(TM) Update Checker
vmx32to64.exe		1,312 K	3,480 K	2040	
iTunesHelper.exe		4,260 K	6,384 K	2556	iTunesHelper
vmtoolsd.exe	< 0.01	10,444 K	13,560 K	2988	VMware Tools Core Service
procexp.exe	< 0.01	13,264 K	19,580 K	2136	Sysinternals Process Explorer
notepad.exe		1,020 K	3,508 K	2412	Notepad

The DLLs loaded by Notepad.exe are listed below:

Name	Description	Company Name	Path
advapi32.dll	Advanced Windows 32 Base API	Microsoft Corporation	C:\Windows\System32\advapi32.dll
comctl32.dll	User Experience Controls Library	Microsoft Corporation	C:\Windows\winsxs\x86_microsoft.win
comdlg32.dll	Common Dialogs DLL	Microsoft Corporation	C:\Windows\System32\comdlg32.dll
gdi32.dll	GDI Client DLL	Microsoft Corporation	C:\Windows\System32\gdi32.dll
imm32.dll	Multi-User Windows IMM32 API Client DLL	Microsoft Corporation	C:\Windows\System32\imm32.dll
kernel32.dll	Windows NT BASE API Client DLL	Microsoft Corporation	C:\Windows\System32\kernel32.dll
loc2008.nls			C:\Windows\System32\loc2008.nls
loc2008.nls			C:\Windows\System32\loc2008.nls
lpk.dll	Language Pack	Microsoft Corporation	C:\Windows\System32\lpk.dll
msctf.dll	MSCTF Server DLL	Microsoft Corporation	C:\Windows\System32\msctf.dll
msvcrt.dll	Windows NT CRT DLL	Microsoft Corporation	C:\Windows\System32\msvcrt.dll
notepad.exe	Notepad	Microsoft Corporation	C:\Windows\System32\notepad.exe
ntdll.dll	NT Layer DLL	Microsoft Corporation	C:\Windows\System32\ntdll.dll
ole32.dll	Microsoft OLE for Windows	Microsoft Corporation	C:\Windows\System32\ole32.dll
oleaut32.dll		Microsoft Corporation	C:\Windows\System32\oleaut32.dll
rpcrt4.dll	Remote Procedure Call Runtime	Microsoft Corporation	C:\Windows\System32\rpcrt4.dll
shell32.dll	Windows Shell Common Dll	Microsoft Corporation	C:\Windows\System32\shell32.dll
shlwapi.dll	Shell Light-weight Utility Library	Microsoft Corporation	C:\Windows\System32\shlwapi.dll
user32.dll	Multi-User Windows USER API Client DLL	Microsoft Corporation	C:\Windows\System32\user32.dll
usp10.dll	Uniscribe Unicode script processor	Microsoft Corporation	C:\Windows\System32\usp10.dll
uxtheme.dll	Microsoft UxTheme Library	Microsoft Corporation	C:\Windows\System32\uxtheme.dll
winspool.drv	Windows Spooler Driver	Microsoft Corporation	C:\Windows\System32\winspool.drv

Velociraptor

Velociraptor Response and Mon +

172.16.123.38:8889/app/index.html#/collected/C.57ac19bf16a58b0a

mut Show All Win10SUS connected admin

New Collection: Select Artifacts to collect

dll

- Windows.Detection.ForwardedImports
- Windows.Detection.Yara.NTFS
- Windows.Forensics.SolarwindsSunburst
- Windows.Persistence.Wow64cpu
- Windows.System.DLLs**

Windows.System.DLLs
Type: client
Enumerate the DLLs loaded by a running process. It includes hash value and certificate information.

Parameters

Name	Type	Default	Description
processRegex		.	A regex applied to process names.
dllRegex		.	A regex applied to the full dll

Select Artifacts Configure Parameters Specify Resources Review Launch

2021-11-11T23:26:40.313Z

Other Memory Artifacts

- **Network connections**
- **Loaded drivers**
 - **Runs in kernel, with elevated privileges**
- **Console command history**
- **Strings in memory**
- **Credentials**

Pagefile Analysis

- **Has no intrinsic structure**
- **Can search for strings**
- **Be careful: antivirus and host-based intrusion detection systems leave signatures in the pagefile**
 - **Suspicious IP addresses, domain names, and malware filenames**
- **Windows can clear the pagefile on shutdown, but this is not its default setting**

Analyzing Common In-Memory Attacks

- **Process injection**
- **Hooking**

Process Injection

- **A malicious *injecting* process causes a legitimate process (*injected*) to load and execute malicious code**
- **In-memory attack**
 - **Disk files do not change**
- ***Injected* process has no evidence indicating which process was responsible for the injection**

Methods of Process Injection

- **Use Windows APIs (requires Administrator or SYSTEM privileges)**
 - **Force target process to load a malicious DLL from disk**
 - **Directly write malicious code to target process's memory and invoke a remote thread to execute it**

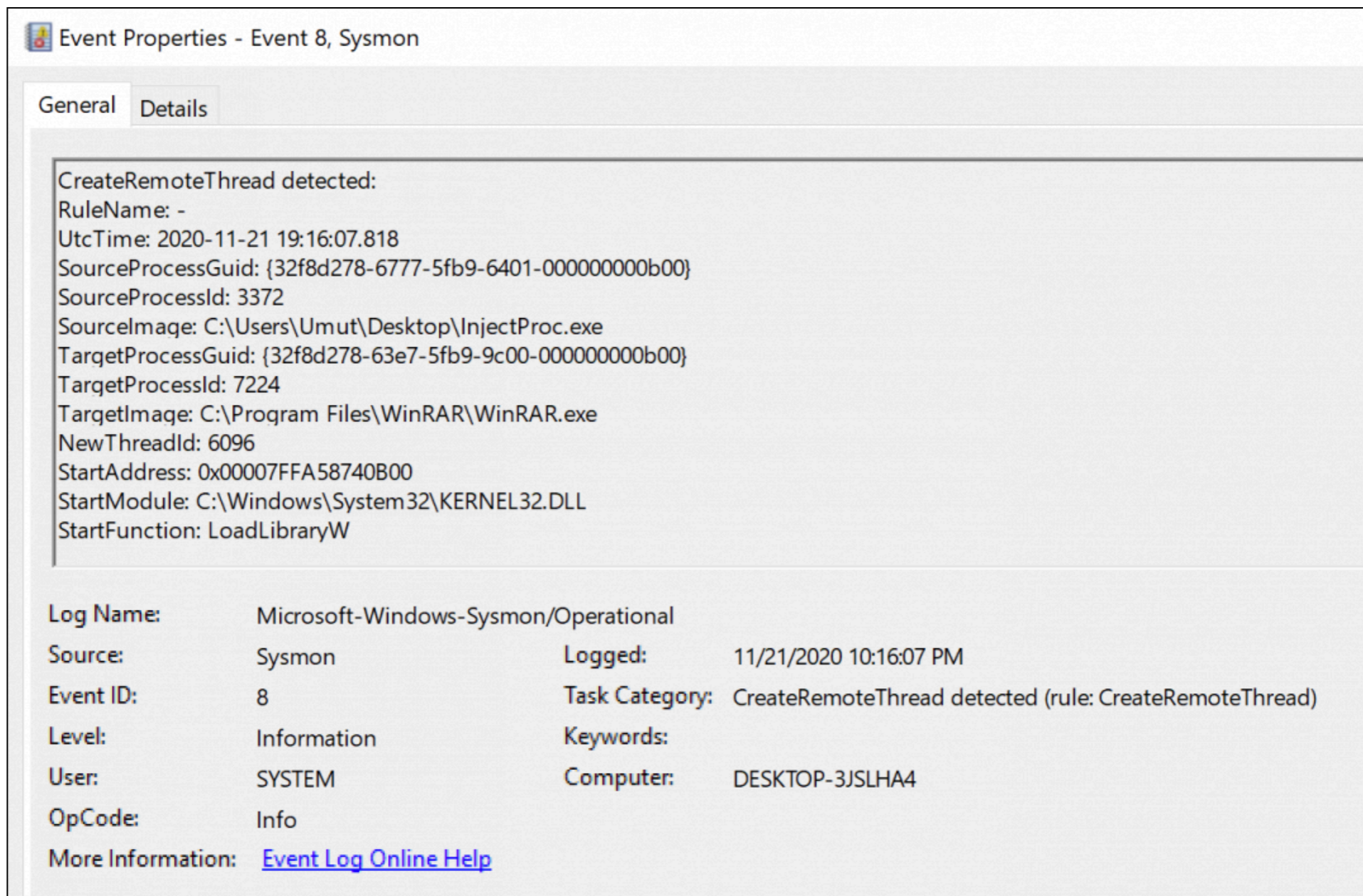
Process Replacement

- **Malware launches a legitimate executable in a suspended state**
- **Then overwrite process memory with malicious code**
- **Unsuspend it to execute**

Detecting Malicious Injection

- **Memory sections with Execute, Read and Write permissions**
- **Processes that don't match corresponding disk files**
 - **Links Ch 12w, 12x**

Sysmon Detecting Injection



The screenshot shows the 'Event Properties' window for 'Event 8, Sysmon'. The 'Details' tab is selected, displaying the following information:

CreateRemoteThread detected:
RuleName: -
UtcTime: 2020-11-21 19:16:07.818
SourceProcessGuid: {32f8d278-6777-5fb9-6401-000000000b00}
SourceProcessId: 3372
SourceImage: C:\Users\Umut\Desktop\InjectProc.exe
TargetProcessGuid: {32f8d278-63e7-5fb9-9c00-000000000b00}
TargetProcessId: 7224
TargetImage: C:\Program Files\WinRAR\WinRAR.exe
NewThreadId: 6096
StartAddress: 0x00007FFA58740B00
StartModule: C:\Windows\System32\KERNEL32.DLL
StartFunction: LoadLibraryW

Log Name: Microsoft-Windows-Sysmon/Operational
Source: Sysmon
Event ID: 8
Level: Information
User: SYSTEM
OpCode: Info
More Information: [Event Log Online Help](#)

Logged: 11/21/2020 10:16:07 PM
Task Category: CreateRemoteThread detected (rule: CreateRemoteThread)
Keywords:
Computer: DESKTOP-3JSLHA4

<https://letsdefend.io/blog/process-injection-detection-with-sysmon/>

Finding Persistence Mechanisms

- **The *injecting* process needs a persistence mechanism to survive reboots**
- **So it may be found in**
 - **Auto-run keys, DLL load-order hijacking, etc.**

Hooking

- **Allows code within running processes to intercept, modify, and view events such as function calls and data they return**
- **Windows provides many API mechanisms to do this**
- **Used by legitimate programs**
 - **Antivirus, host-based intrusion detection systems, application inventory software**

Malicious Hooking

- **Rootkits use hooking to hide files, processes, registry keys, or network connections**
- **Keyloggers may use SetWindowsHookEx to cause a malicious DLL function to be called whenever a keyboard event occurs**
- **Or use GetAsyncKeyState to constantly check the up/down state of keys**

Types of Hooks

- **Manipulate a process's Import Address Table**
 - **So it calls malicious functions instead of legitimate system functions**
- **Hook kernel structures such as the Interrupt Descriptor Table (IDT) and System Service Dispatch Table (SSDT)**
 - **Prevented on modern Windows systems by Kernel Patch Protection (KPP)**

Zeus Hook

- **The next slide shows the output of "apihooks" (a Volatility plugin)**
- **On a system infected with Zeus**
- **Shows an inline hook to the HttpSendRequestA function imported from Wininet.dll within the process space of Isass.exe**

Volatility Detecting Hooks

```
C:\Windows\system32\cmd.exe

*****
Hook mode: Usermode
Hook type: Inline/Trampoline
Process: 688 (lsass.exe)
Victim module: WININET.dll (0x771b0000 - 0x77256000)
Function: WININET.dll!HttpSendRequestA at 0x771c76b8
Hook address: 0xa27366
Hooking module: <unknown>

Disassembly(0):
0x771c76b8 e9a9fc8589      JMP 0xa27366
0x771c76bd 6a13           PUSH 0x13
0x771c76bf 6a00           PUSH 0x0
0x771c76c1 ff7518        PUSH DWORD [EBP+0x18]
0x771c76c4 ff7514        PUSH DWORD [EBP+0x14]
0x771c76c7 ff7510        PUSH DWORD [EBP+0x10]
0x771c76ca ff750c        PUSH DWORD [EBP+0xc]
0x771c76cd ff7508        PUSH DWORD [EBP+0x8]

Disassembly(1):
0xa27366 55           PUSH EBP
0xa27367 8bec        MOV EBP, ESP
0xa27369 e8cac7feff   CALL 0xa13b38
0xa2736e ff7510        PUSH DWORD [EBP+0x10]
0xa27371 8b4518        MOV EAX, [EBP+0x18]
```

Figure 12-36. Output of the Volatility Framework's apihooks plugin on a system infected with Zeus

Memory Analysis Tools

- **Acquisition tools**
 - **FTK Imager**
 - **Dumplt**
 - **Memoryze**
 - **Velociraptor**
- **Analysis tools**
 - **Memoryze**
 - **Volatility**

Kahoot!

12c-2

Alternative Persistence Mechanisms

Alternative Persistence Mechanisms

- **Startup folders**
- **Recurring tasks**
- **System binary modification**
- **The sticky keys attack**
- **DLL load-order hijacking**

Startup Folders

- **Any program or shortcut in this folder is launched**
 - **On startup or login**

On Windows Vista and Windows 7, the “all users” Startup folder is located at C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup, and the user-specific Startup folder is located at C:\Users\[username]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup.

Recurring Tasks

- **Use "at" or "schtasks" commands**
- **To make a task that recurs at regular times or days of the week**
- **Future and recurring scheduled tasks persist as .job files in %SYSTEMROOT%\Tasks**

System Binary Modification

- **Modify existing Windows binary**
 - **Typically one automatically loaded on bootup or login**
- **Add malicious code**
- **Time-stomp**
- **Will change MD5 hash and break signature, but not all legitimate binaries are signed**

Careful Modifications

- **Changes that cause Windows to crash or impair user experience will limit the attacker's ability to persist**
- **Attackers are more likely to replace noncritical executables or libraries**

Defenses

- **Windows File Protection in older versions of Windows (XP, 2000)**
 - **Easily bypassed by a local Administrator**
- **Replaced by Windows Resource Protection (WRP) in Windows Vista and later**
 - **Requires TrustedInstaller permissions to alter WFP-governed resources**
 - **More resistant to tampering**

The Sticky Keys Attack

- **Targets sethc.exe**
 - **A file that provides accessibility features**
 - **Replace sethc.exe with cmd.exe**
- **Press Shift key five times before logon**
 - **A command shell opens with SYSTEM privileges**
 - **Even works during a Remote Desktop Protocol session**

The Sticky Keys Attack

- **No longer works on Vista and later versions**
- **But there's another way to get the same result**

An attacker can simply set cmd.exe as the debugger for sethc.exe by adding the registry key
HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Image File Execution Options\sethc.exe,
creating a value of type REG_SZ named Debugger, and setting
the value data to C:\WINDOWS\system32\cmd.exe.

A screenshot of a web browser window. The address bar shows the URL <https://www.bleepingcomputer.com/news/security/windows-defender-ca>. The page title is **Windows Defender Can Detect Accessibility Tool Backdoors**. Below the title, it says **By Lawrence Abrams**. On the right side, there is a calendar icon for **November 27, 2018**, a clock icon for **12:31 PM**, and a comment icon with the number **4**.

Windows Defender will now detect when accessibility programs such as sethc.exe or utilman.exe have been hijacked by an Image File Execution Options debugger so that they can be used as a backdoor.

DLL Load-Order Hijacking

- **DLLs are loaded when a program launches**
- **But DLLs might be in many different folders**
- **"KnownDLLs" registry key lists known system DLLs and ensures that they are always loaded from %systemroot%\System32**

Registry Editor

File Edit View Favorites Help

ServiceProvider
 Session Manager
 AppCompatCache
 Configuration Mana
 DOS Devices
 Environment
 Executive
 FileRenameOperatic
 I/O System
 kernel
 KnownDLLs
 Memory Manageme
 Power
 Quota System
 SubSystems
 WPA
 SNMP
 StillImage
 Storage
 SystemInformation
 SystemResources
 Terminal Server
 TimeZoneInformation
 usbflags
 usbstor
 VAN
 Video
 VirtualDeviceDrivers
 Wdf
 WDI
 Windows
 Winlogon
 Winresume

Name	Type	Data
(Default)	REG_SZ	(value not set)
advapi32	REG_SZ	advapi32.dll
dbcatq	REG_SZ	dbcatq.dll
COMDLG32	REG_SZ	COMDLG32.dll
DllDirectory	REG_EXPAND_SZ	%SystemRoot%\system32
gdi32	REG_SZ	gdi32.dll
IERTUTIL	REG_SZ	IERTUTIL.dll
IMAGEHLP	REG_SZ	IMAGEHLP.dll
IMM32	REG_SZ	IMM32.dll
kernel32	REG_SZ	kernel32.dll
LPK	REG_SZ	LPK.dll
MSCTF	REG_SZ	MSCTF.dll
MSVCRT	REG_SZ	MSVCRT.dll
NORMALIZ	REG_SZ	NORMALIZ.dll
NSI	REG_SZ	NSI.dll
ole32	REG_SZ	ole32.dll
OLEAUT32	REG_SZ	OLEAUT32.dll
rpcrt4	REG_SZ	rpcrt4.dll
Setupapi	REG_SZ	Setupapi.dll
SHELL32	REG_SZ	SHELL32.dll
SHLWAPI	REG_SZ	SHLWAPI.dll
URLMON	REG_SZ	URLMON.dll
user32	REG_SZ	user32.dll
USP10	REG_SZ	USP10.dll
WININET	REG_SZ	WININET.dll
WLDAP32	REG_SZ	WLDAP32.dll
WS2_32	REG_SZ	WS2_32.dll

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\KnownDLLs

Unknown DLL Search Order

SafeDllSearchMode Disabled (0)	SafeDllSearchMode Enabled (1)
Directory where the application is loaded	Directory where the application is loaded
Current working directory	%systemroot%\system32
%systemroot%\system32	%systemroot%\system
%systemroot%\system	%systemroot%
%systemroot%	Current working directory
%PATH% environment variable entries	%PATH% environment variable entries

DLL Load-Order Hijacking

Works When:

- The legitimate DLL is not specified in KnownDLLs.
 - The legitimate DLL is not in the same directory as the application executable file.
 - The executable file does not use an absolute path to load the DLL.
-
- **ntshrui.dll is loaded by Windows Explorer and is vulnerable**
 - **A malicious ntshrui.dll in %systemroot% will launch when Explorer does**

Kahoot!

12c-3