



南开大学

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月异

# 恶意代码分析与防治技术

## 第7章 Windows恶意代码

王志

zwang@nankai.edu.cn

updated on 2022-10-29

南开大学 网络安全安全学院

2022/2023



允公允能 日新月异

# Outline

- Windows API
- Windows Registry
- Networking APIs
- Following Running Malware
- Kernel Mode vs. User Mode
- Native API





南開大學

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月異



# Windows API



允公允能 日新月异

# What is the API?

- A broad set of functionality
  - File operation
  - Network operation
  - .....
- API governs how programs interact with Windows OS





允公允能 日新月异

# Windows API

- Concepts
  - Data Types
  - Hungarian Notation
  - Handles
  - File System Functions
  - Special Files





允公允能 日新月异

# Data Types

- Windows API has its own names to represent data types
  - **DWORD** for 32-bit unsigned integers
  - **WORD** for 16-bit unsigned integers





允公允能 日新月异

# Data Types

- WinDef.H

```
typedef int BOOL;
```

```
typedef unsigned char BYTE;
```

```
typedef unsigned short WORD;
```

```
typedef unsigned long DWORD;
```

- WinNT.H

```
typedef BYTE BOOLEAN;
```

```
typedef PVOID HANDLE;
```

```
#define CALLBACK __stdcall
```





允公允能 日新月异

# Hungarian Notation

## ● Hungarian Notation

- Clarity and consistency
- Variable prefix notations
  - data type
- Comment Block
- Class Declaration Header
- .....







允公允能 日新月异

# Common API Data Types

Prefix	API Type	C Type
w	WORD	16-bit unsigned value
dw	DWORD	32-bit unsigned value
H	HANDLE	A reference to an object
LP	Long Pointer	A pointer to another type
Callback	Callback	A function called by API





允公允能 日新月异

# Handles

- Items opened or created in the OS, like
  - window, process, menu, file, ...
- Handles are like pointers to those objects
  - They are not pointers, however
- The only thing you can do with a handle is **store** it and use it in a later function call to **refer** to the same object





允公允能 日新月异

# Handle Example

- The CreateWindowEx function returns an **HWND**, a handle to the window
- To do anything to that window (such as DestroyWindow) , use the handle
- HWND can not be used as a pointer or arithmetic value.





允公允能 日新月异

# File System Functions

- CreateFile, ReadFile, WriteFile
  - Normal file input/output
- CreateFileMapping, MapViewOfFile
  - Used by malware, loads file into RAM
  - Can be used to execute a file without using the Windows loader





# File Namespace

- Prefix \\?\
- Disables string parsing
- Allows longer filenames
  - exceed the **MAX\_PATH** limits that are enforced by the Windows APIs.
- **Network Shared files** like \\server\share





允公允能 日新月异

# Device Namespace

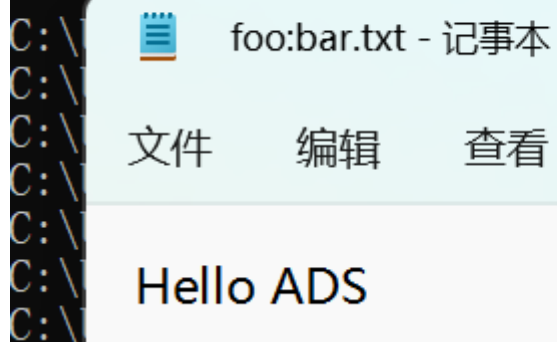
- Prefix \\.
- Access devices like physical disks and volumes
  - without going through the file system
  - `\\.\PhysicalDriveX`, `\\.\CdRomX`, `\\.\COM56`
  - `CreateFile` opens both files and devices

```
PS C:\Users\dell> wmic diskdrive list brief
Caption                DeviceID              Model                Partitions  Size
-----
INTEL SSDPEKNW010T8    \\. \PHYSICALDRIVE0  INTEL SSDPEKNW010T8  3           1024203640320
```

# Alternate Data Streams (ADS)

- Add one file to another file
- Not list in the directory
- Malware authors like ADS

```
C:\Users\dell\nkamg>notepad foo:bar.txt
```



foo:bar.txt - 记事本

文件 编辑 查看

Hello ADS

```
C:\Users\dell\nkamg>echo Hello World! > foo
```

```
C:\Users\dell\nkamg>dir
驱动器 C 中的卷没有标签。
卷的序列号是 0000-FE6C
```

C:\Users\dell\nkamg 的目录

```
2022/10/29  22:52    <DIR>          .
2022/10/29  22:51    <DIR>          ..
2022/10/29  22:52             15 foo
                        1 个文件          15 字节
                        2 个目录 865,467,006,976 可用字节
```

```
C:\Users\dell\nkamg>echo Hello ADS > foo:bar.txt
```

```
C:\Users\dell\nkamg>dir
驱动器 C 中的卷没有标签。
卷的序列号是 0000-FE6C
```

C:\Users\dell\nkamg 的目录

```
2022/10/29  22:53    <DIR>          .
2022/10/29  22:51    <DIR>          ..
2022/10/29  22:53             15 foo
                        1 个文件          15 字节
                        2 个目录 865,466,241,024 可用字节
```





南開大學

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月異

# Windows Registry





允公允能 日新月异

# Registry Purpose

- OS and program configuration settings
  - Desktop background, mouse preferences, etc.
- Malware uses the registry for persistence
  - Making malware re-start when the system reboots





允公允能 日新月异

# Registry

- Hierarchical Database
- Two basic elements:
  - **Key**: Container Object similar to folder
  - **Value**: Non-Container Object similar to file
- Keys may contain values or further subkeys





允公允能 日新月异

# Registry

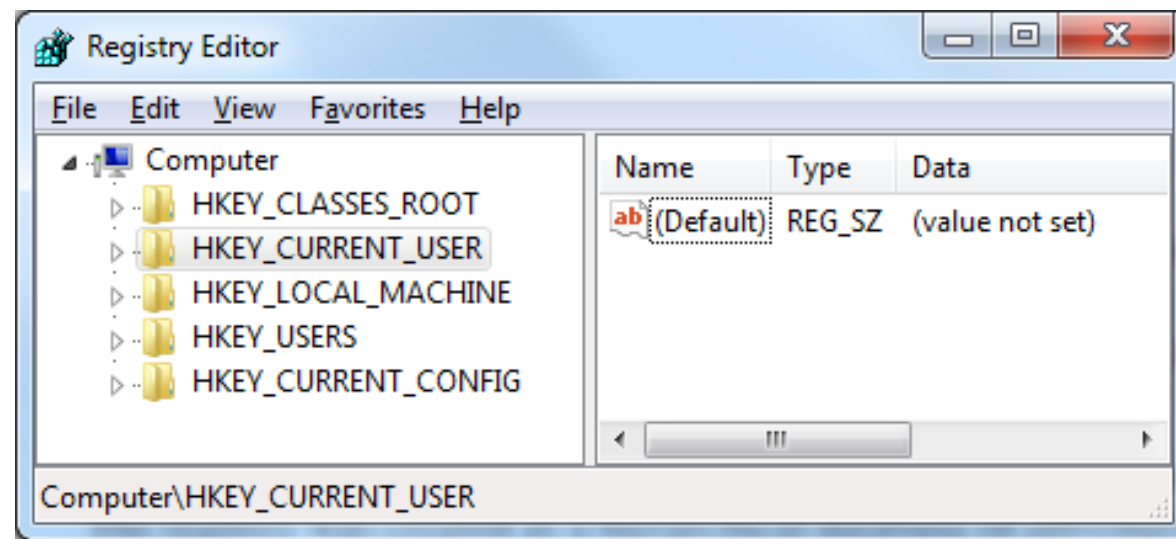
- Structure:

- Similar to Windows' path names
- backslashes indicate levels of **hierarchy**
- Registry keys can only be accessed from a **root** key.



# Windows Registry

- **Root Keys** There are 5 root keys
- **Subkey** A folder within a folder
- **Key** A folder; can contain folders or values
- **Value Entry** Two parts: name and data
- **Value or Data** The data stored in a registry entry
- REGEDIT Tool to view/edit the Registry





允公允能 日新月异

# Registry Example

- HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows
  - refers to the subkey "Windows" of the subkey "Microsoft" of the subkey "Software" of the HKEY\_LOCAL\_MACHINE root key.





允公允能 日新月异

# Root Keys

## Registry Root Keys

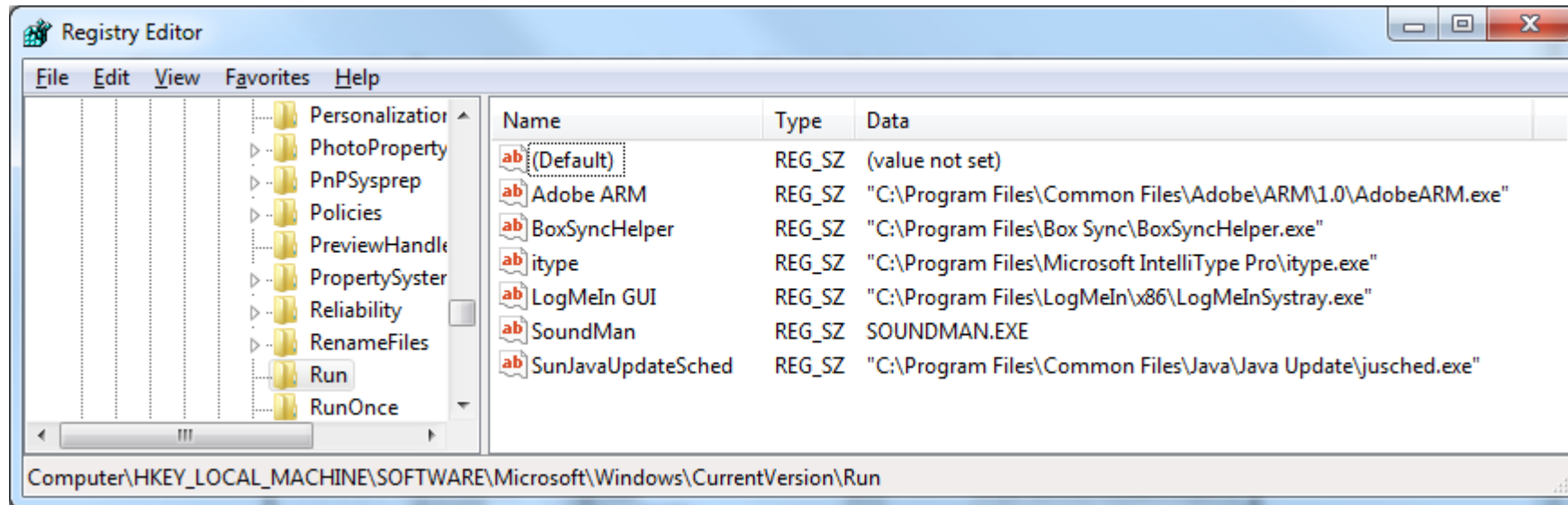
The registry is split into the following five root keys:

- **HKEY\_LOCAL\_MACHINE (HKLM)**. Stores settings that are global to the local machine
- **HKEY\_CURRENT\_USER (HKCU)**. Stores settings specific to the current user
- **HKEY\_CLASSES\_ROOT**. Stores information defining types
- **HKEY\_CURRENT\_CONFIG**. Stores settings about the current hardware configuration, specifically differences between the current and the standard configuration
- **HKEY\_USERS**. Defines settings for the default user, new users, and current users



# Run Key

- **HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run**
  - Executables that start when a user logs on





允公允能 日新月异

# Autoruns

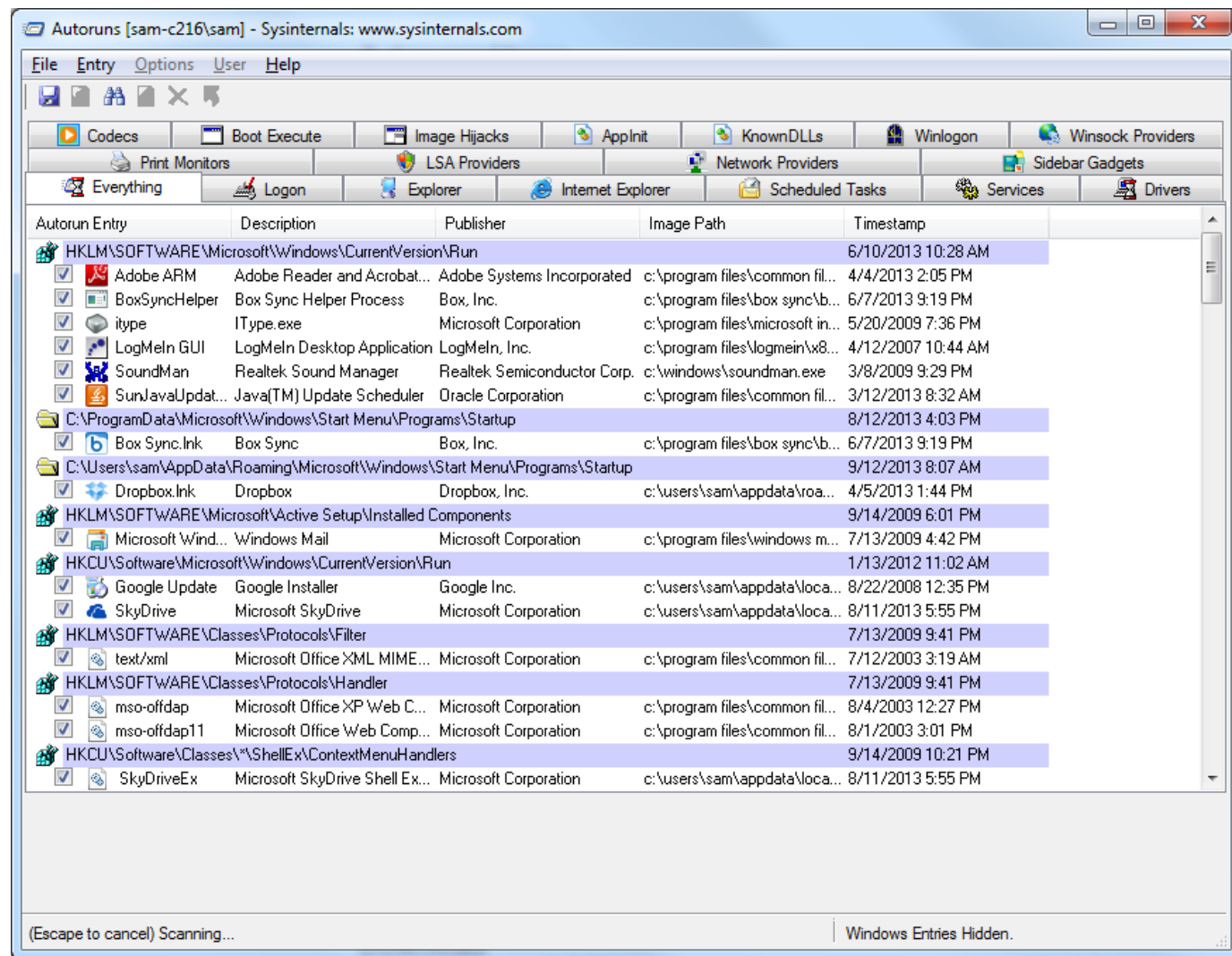
- Sysinternals tool
- Lists code that will run automatically when system starts
  - Executables
  - DLLs loaded into IE and other programs
  - Drivers loaded into Kernel
  - It checks more than 25 registry locations





允公允能 日新月异

# Autoruns



南开大学  
Nankai University

# Common Registry Functions

- RegOpenKeyEx
  - Opens a registry key for editing and querying
- RegSetValueEx
  - Adds a new value to the registry & sets its data
- RegGetValue
  - Returns the data for a value entry in the Registry
- Note: Documentation will omit the trailing **W** (wide) or **A** (ASCII) character in a call like RegOpenKeyExW



允公允能 日新月异

# Ex, A, and W Suffixes

## FUNCTION NAMING CONVENTIONS

When evaluating unfamiliar Windows functions, a few naming conventions are worth noting because they come up often and might confuse you if you don't recognize them. For example, you will often encounter function names with an Ex suffix, such as `CreateWindowEx`. When Microsoft updates a function and the new function is incompatible with the old one, Microsoft continues to support the old function. The new function is given the same name as the old function, with an added Ex suffix. Functions that have been significantly updated twice have two Ex suffixes in their names.

Many functions that take strings as parameters include an A or a W at the end of their names, such as `CreateDirectoryW`. This letter does *not* appear in the documentation for the function; it simply indicates that the function accepts a string parameter and that there are two different versions of the function: one for ASCII strings and one for wide character strings. Remember to drop the trailing A or W when searching for the function in the Microsoft documentation.





允公允能 日新月异

# Registry Code

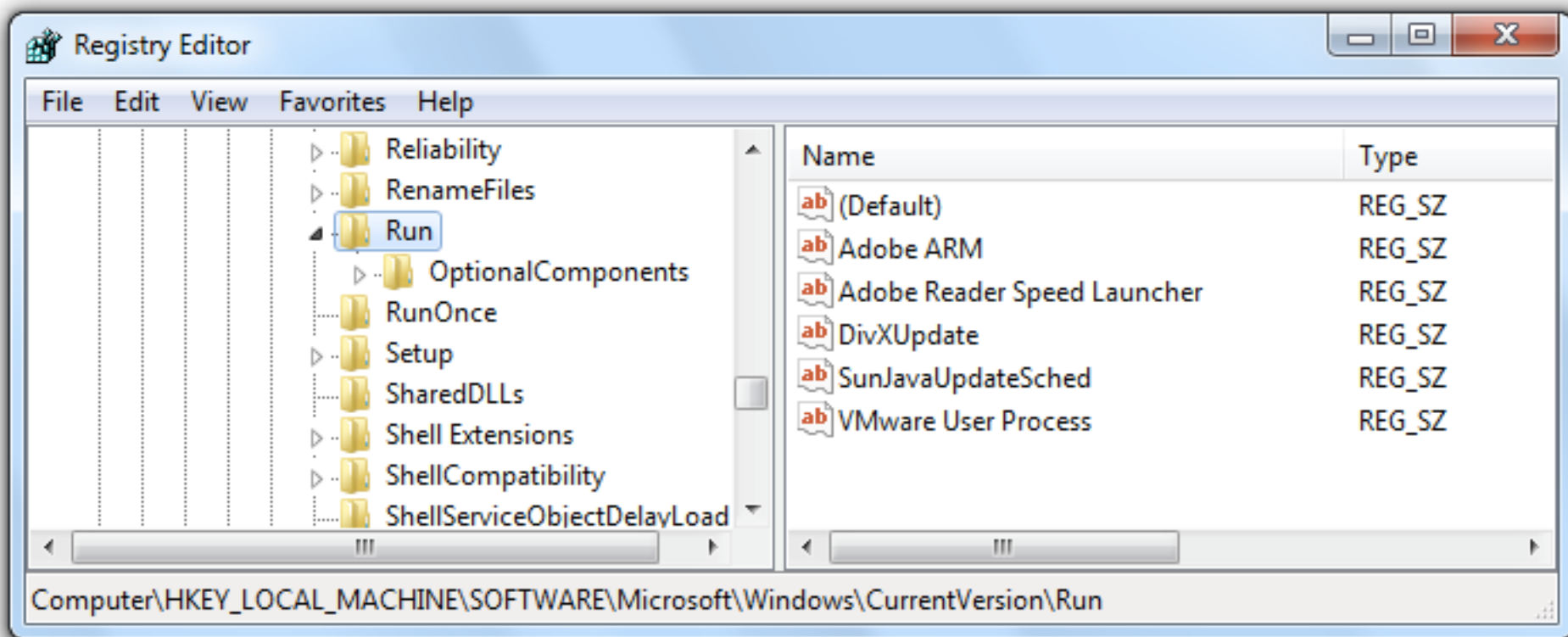
*Example 8-1. Code that modifies registry settings*

```
0040286F    push    2                ; samDesired
00402871    push    eax              ; ulOptions
00402872    push    offset SubKey    ;
"Software\\Microsoft\\Windows\\CurrentVersion\\Run"
00402877    push    HKEY_LOCAL_MACHINE ; hKey
```



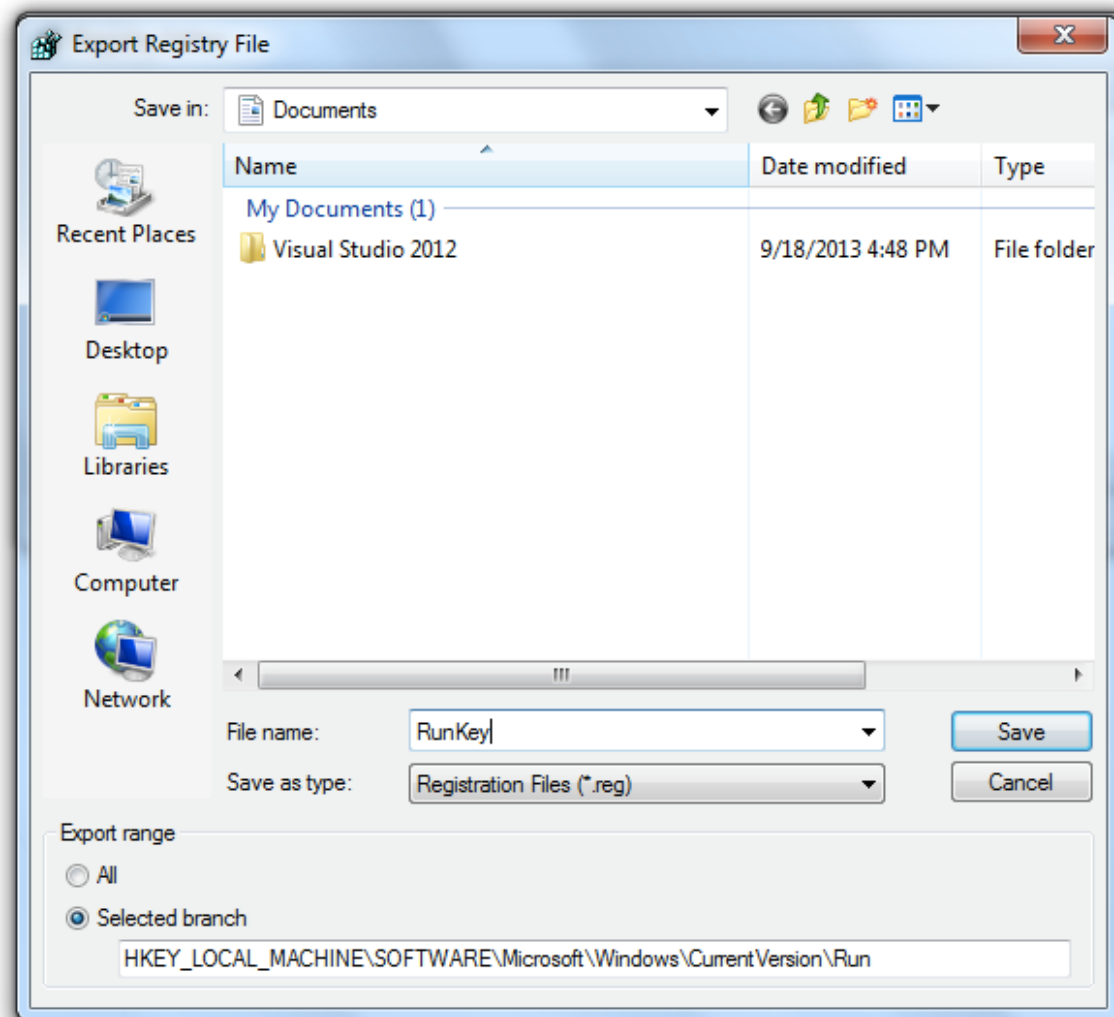
允公允能 日新月异

# .REG Files



允公允能 日新月异

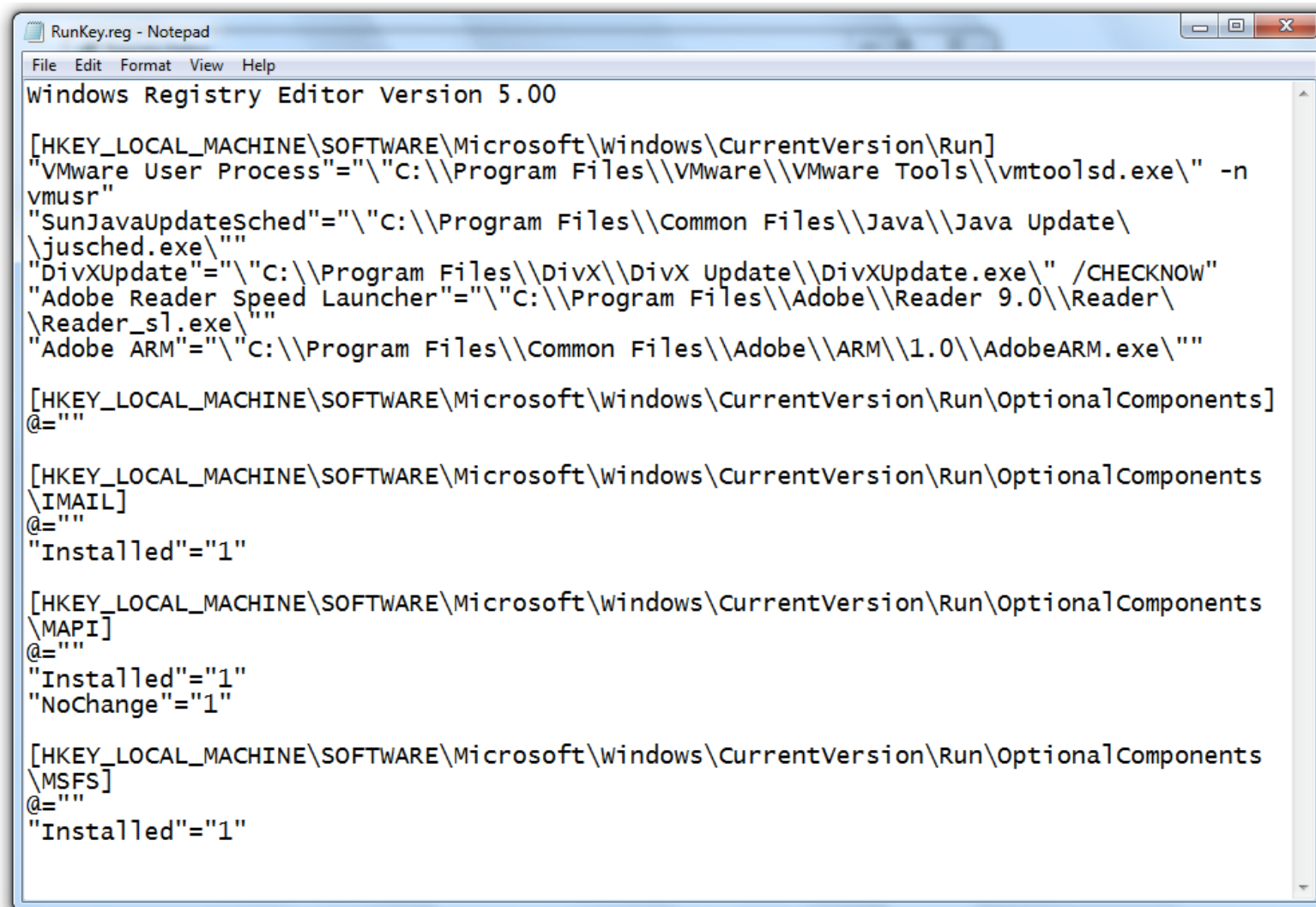
# .REG Files





允公允能 日新月异

# .REG Files



```
RunKey.reg - Notepad
File Edit Format View Help
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
"VMware User Process"="\"C:\\Program Files\\VMware\\VMware Tools\\vmtoolsd.exe\" -n
vmusr"
"SunJavaUpdateSched"="\"C:\\Program Files\\Common Files\\Java\\Java Update\\
\\jusched.exe\"""
"DivXUpdate"="\"C:\\Program Files\\DivX\\DivX Update\\DivXUpdate.exe\" /CHECKNOW"
"Adobe Reader Speed Launcher"="\"C:\\Program Files\\Adobe\\Reader 9.0\\Reader\\
\\Reader_sl.exe\"""
"Adobe ARM"="\"C:\\Program Files\\Common Files\\Adobe\\ARM\\1.0\\AdobeARM.exe\"""

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\OptionalComponents]
@=""

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\OptionalComponents
\IMAIL]
@=""
"Installed"="1"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\OptionalComponents
\MAPI]
@=""
"Installed"="1"
"NoChange"="1"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\OptionalComponents
\MSFS]
@=""
"Installed"="1"
```



南開大學

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月異



# Networking APIs





允公允能 日新月异

# Berkeley Compatible Sockets

- Winsock libraries, primarily in *ws2\_32.dll*
  - Almost identical in Windows and Unix





Function	Description
----------	-------------

socket	Creates a socket
--------	------------------

bind	Attaches a socket to a particular port, prior to the accept call
------	--

listen	Indicates that a socket will be listening for incoming connections
--------	--

accept	Opens a connection to a remote socket and accepts the connection
--------	--

connect	Opens a connection to a remote socket; the remote socket must be waiting for the connection
---------	---

recv	Receives data from the remote socket
------	--------------------------------------

send	Sends data to the remote socket
------	---------------------------------

### NOTE

*The `WSAStartup` function must be called before any other networking functions in order to allocate resources for the networking libraries. When looking for the start of network connections while debugging code, it is useful to set a breakpoint on `WSAStartup`, because the start of networking should follow shortly.*





允公允能 日新月异

# Server and Client Sides

- Server side
  - Maintains an open socket waiting for connections
  - Calls, in order, **socket**, **bind**, **listen**, **accept**
  - Then **send** and **recv** as necessary
- Client side
  - Connects to a waiting socket
  - Calls, in order, **socket**, **connect**
  - Then **send** and **recv** as necessary



# Simplified Server Program

Realistic code would call  
**WSAGetLastError** many  
times

```
00401041 push    ecx                ; lpWSAData
00401042 push    202h             ; wVersionRequested
00401047 mov     word ptr [esp+250h+name.sa_data], ax
0040104C call    ds:WSAStartup
00401052 push    0                ; protocol
00401054 push    1                ; type
00401056 push    2                ; af
00401058 call    ds:socket
0040105E push    10h              ; namelen
00401060 lea     edx, [esp+24Ch+name]
00401064 mov     ebx, eax
00401066 push    edx              ; name
00401067 push    ebx              ; s
00401068 call    ds:bind
0040106E mov     esi, ds:listen
00401074 push    5            ; backlog
00401076 push    ebx              ; s
00401077 call    esi ; listen
00401079 lea     eax, [esp+248h+addrlen]
0040107D push    eax              ; addrlen
0040107E lea     ecx, [esp+24Ch+hostshort]
00401082 push    ecx              ; addr
00401083 push    ebx              ; s
00401084 call    ds:accept
```





允公允能 日新月异

# The WinINet API

- Higher-level API than Winsock
- Library name is "*Wininet.dll*"
- Implements Application-layer protocols like HTTP and FTP
  - **InternetOpen** – connects to Internet
  - **InternetOpenURL** – connects to a URL
  - **InternetReadFile** – reads data from a downloaded file





南開大學

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月異



Following Running Malware



允公允能 日新月异

# Transferring Execution

- **jmp** and **call** transfer execution to another part of code, but there are other ways
  - DLLs
  - Processes
  - Threads
  - Mutexes
  - Services
  - Component Object Model (COM)
  - Exceptions







允公允能 日新月异

# DLLs (Dynamic Link Libraries)

- **Share code** among multiple applications
- DLLs export code that can be used by other applications
- **Static libraries** were used before DLLs
  - They still exist, but are much less common
  - They cannot share memory among running processes
  - Static libraries use more RAM than DLLs







允公允能 日新月异

# DLL Advantages

- Using DLLs already included in Windows makes code **smaller**
- Software companies can also make **custom DLLs**
  - Distribute DLLs along with EXEs





允公允能 日新月异

# How Malware Authors Use DLLs

- Store **malicious code** in DLL
  - Sometimes load malicious DLL into another process
- Using **Windows** DLLs
  - Nearly all malware uses basic Windows DLLs
- Using **third-party** DLLs
  - Use Firefox DLL to connect to a server, instead of Windows API



南开大学  
Nankai University



允公允能 日新月异

# Basic DLL Structure

- DLLs are very similar to EXEs
- Same **PE** file format
- A single flag indicates that it's a DLL instead of an EXE
- DLLs have more exports & fewer imports
- **DllMain** is the main function, **not exported**, but specified as the **entry point** in the PE Header
  - Called when a function loads or unloads the library





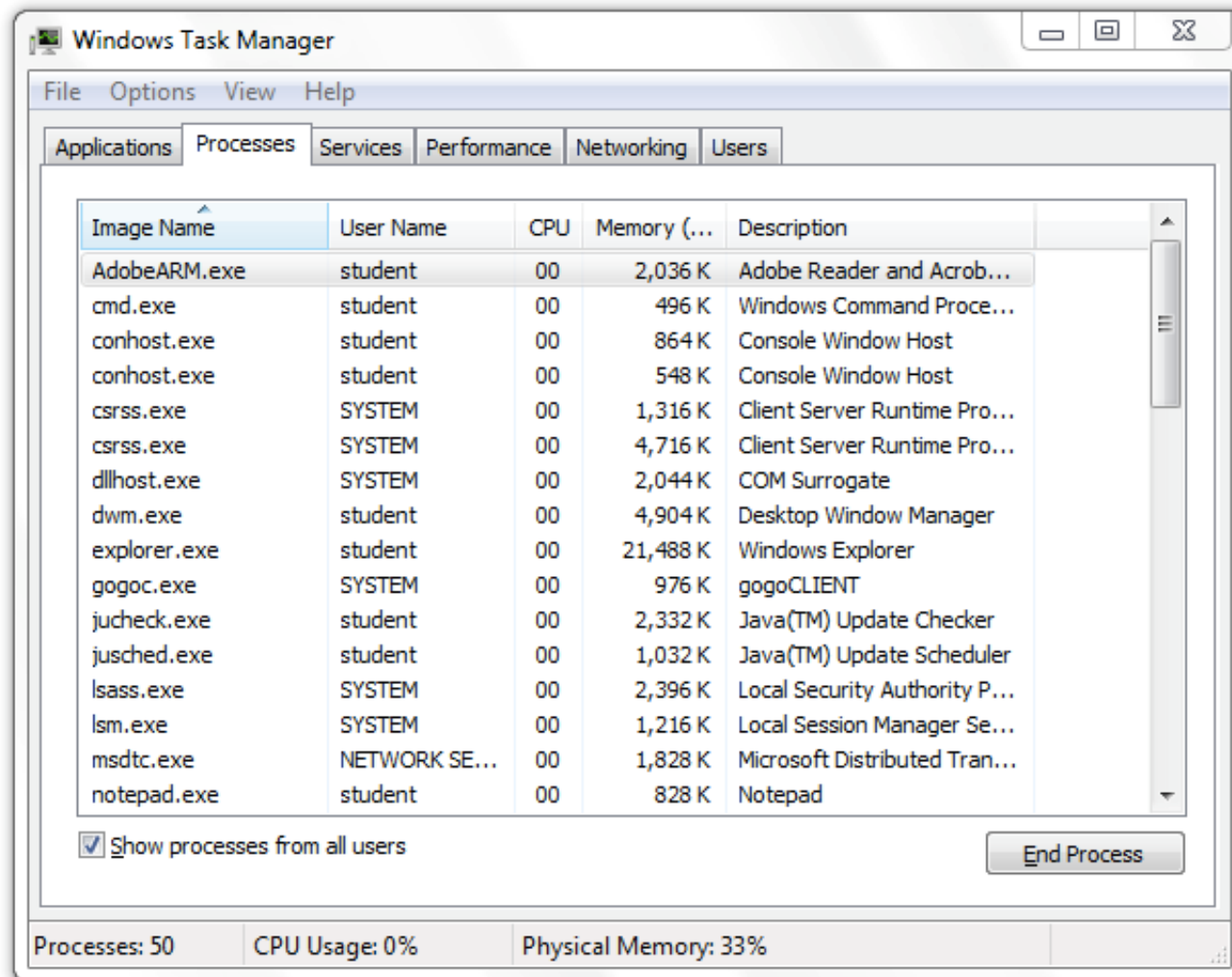
允公允能 日新月异

# Processes

- Every program being executed by Windows is a **process**
- Each process has its own **resources**
  - Handles, memory
- Each process has one or more **threads**
- Older malware run as an **independent** process
- Newer malware executes its code **as part of** another process



# Many Processes Run at Once





允公允能 日新月异

# Memory Management

- Each process uses **resources**, like CPU, file system, and memory
- OS allocates memory to each process
- Two processes accessing the same memory address actually access different locations in **RAM**
  - **Virtual address space**





允公允能 日新月异

# Creating a New Process

- **CreateProcess**

- Can create a simple remote shell with one function call
- **STARTUPINFO** parameter contains handles for standard **input**, standard **output**, and standard **error** streams
  - Can be set to a socket, creating a remote shell







允公允能 日新月异

# Windows MSDN

```
BOOL WINAPI CreateProcess(  
    _In_opt_      LPCTSTR                lpApplicationName,  
    _Inout_opt_  LPTSTR                 lpCommandLine,  
    _In_opt_      LPSECURITY_ATTRIBUTES lpProcessAttributes,  
    _In_opt_      LPSECURITY_ATTRIBUTES lpThreadAttributes,  
    _In_          BOOL                  bInheritHandles,  
    _In_          DWORD                  dwCreationFlags,  
    _In_opt_      LPVOID                 lpEnvironment,  
    _In_opt_      LPCTSTR                 lpCurrentDirectory,  
    _In_          LPSTARTUPINFO          lpStartupInfo,  
    _Out_         LPPROCESS_INFORMATION lpProcessInformation  
);
```





允公允能 日新月异

# STARTUPINFO

```
typedef struct _STARTUPINFO {  
    DWORD    cb;  
    LPTSTR   lpReserved;  
    LPTSTR   lpDesktop;  
    LPTSTR   lpTitle;  
    DWORD    dwX;  
    DWORD    dwY;  
    DWORD    dwXSize;  
    DWORD    dwYSize;  
    DWORD    dwXCountChars;  
    DWORD    dwYCountChars;  
    DWORD    dwFillAttribute;  
    DWORD    dwFlags;  
    WORD     wShowWindow;  
    WORD     cbReserved2;  
    LPBYTE   lpReserved2;  
    HANDLE   hStdInput;  
    HANDLE   hStdOutput;  
    HANDLE   hStdError;  
} STARTUPINFO, *LPSTARTUPINFO;
```





允公允能 日新月异

# Process Information

```
typedef struct _PROCESS_INFORMATION {  
    HANDLE hProcess;  
    HANDLE hThread;  
    DWORD   dwProcessId;  
    DWORD   dwThreadId;  
} PROCESS_INFORMATION, *LPPROCESS_INFORMATION;
```

Contains information about a newly created process and its primary thread.



# Code to Create a Shell

*Example 8-4. Sample code using the CreateProcess call*

```
004010DA  mov     eax, dword ptr [esp+58h+SocketHandle]
004010DE  lea     edx, [esp+58h+StartupInfo]
004010E2  push    ecx                ; lpProcessInformation
004010E3  push    edx                ; lpStartupInfo
004010E4  1mov    [esp+60h+StartupInfo.hStdError], eax
004010E8  2mov    [esp+60h+StartupInfo.hStdOutput], eax
004010EC  3mov    [esp+60h+StartupInfo.hStdInput], eax
004010F0  4mov    eax, dword_403098
004010F5  push    0                  ; lpCurrentDirectory
004010F7  push    0                  ; lpEnvironment
004010F9  push    0                  ; dwCreationFlags
004010FB  mov     dword ptr [esp+6Ch+CommandLine], eax
```

- Loads socket handles, StdError, StdOutput and StdInput into lpProcessInformation





```
004010FF  push    1                ; bInheritHandles
00401101  push    0                ; lpThreadAttributes
00401103  lea     eax, [esp+74h+CommandLine]
00401107  push    0                ; lpProcessAttributes
00401109  5push   eax              ; lpCommandLine
0040110A  push    0                ; lpApplicationName
0040110C  mov     [esp+80h+StartupInfo.dwFlags], 101h
00401114  6call   ds:CreateProcessA
```

- CreateProcess has 10 parameters
  - lpCommandLine
  - lpProcessInformation
  - lpStartupInfo





# Process and Thread

- Application
  - Consists of one or more **processes**
- Process
  - An executing program
  - One or more **threads** are running in the context of the process
- Thread
  - Basic **unit** to which the OS allocates CPU time





允公允能 日新月异

# Threads

- Processes are **containers**
  - Each process contains one or more threads
- Threads are what Windows actually executes
- Threads
  - Independent sequences of instructions
  - Executed by CPU without waiting for other threads
  - Threads within a process **share** the same memory space
  - Each thread has its **own** registers and stack







允公允能 日新月异

# Thread Context

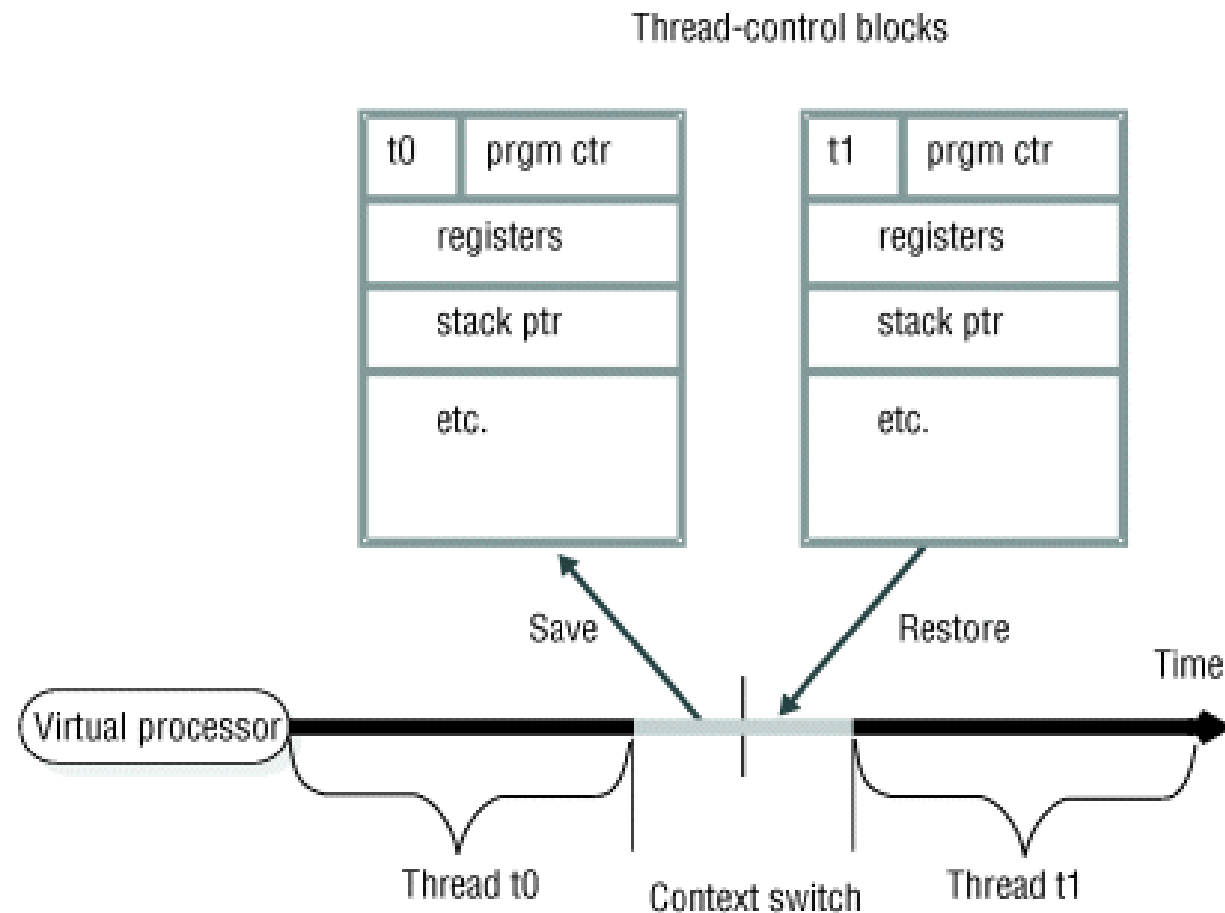
- When a thread is running, it has complete **control** of the CPU
- Other threads cannot affect the **state of the CPU**
- When a thread changes a **register**, it does not affect any other threads
- When the OS switches to another thread, it saves all CPU values in a **structure** called the **thread context**





允公允能 日新月异

# Thread Context Switch





允公允能 日新月异

# Creating a Thread

- **CreateThread**

- Caller specified a **start** address, also called a **start** function

```
HANDLE WINAPI CreateThread(  
    _In_opt_ LPSECURITY_ATTRIBUTES lpThreadAttributes,  
    _In_     SIZE_T                 dwStackSize,  
    _In_     LPTHREAD_START_ROUTINE lpStartAddress,  
    _In_opt_ LPVOID                 lpParameter,  
    _In_     DWORD                   dwCreationFlags,  
    _Out_opt_ LPDWORD                lpThreadId  
);
```





允公允能 日新月异

# How Malware Uses Threads

- Use **CreateThread** to load a malicious DLL into a process
  - Virtual Protect
  - VirtualAlloc
  - CreateThread

```
ter((gproc kernel32.dll VirtualProtect), (gdele  
nPointer((gproc kernel32.dll VirtualAlloc), (gc  
rFunctionPointer((gproc msvcrt.dll memset), (gc
```

```
nPointer((gproc kernel32.dll CreateThread), (gc
```

```
ter((gproc kernel32.dll CreateThread), (gdeleg
```





允公允能 日新月异

# How Malware Uses Threads

- Create two threads, for input and output
  - Used to communicate with a running application
  - **input**: listen on a socket or pipe of a process
  - **output**: read from socket or pipe of a process





允公允能 日新月异

# Interprocess Coordination with Mutexes

- **Mutexes** are global objects that coordinate multiple processes and threads
- Mutexes often use **hard-coded** names which can be used to identify malware

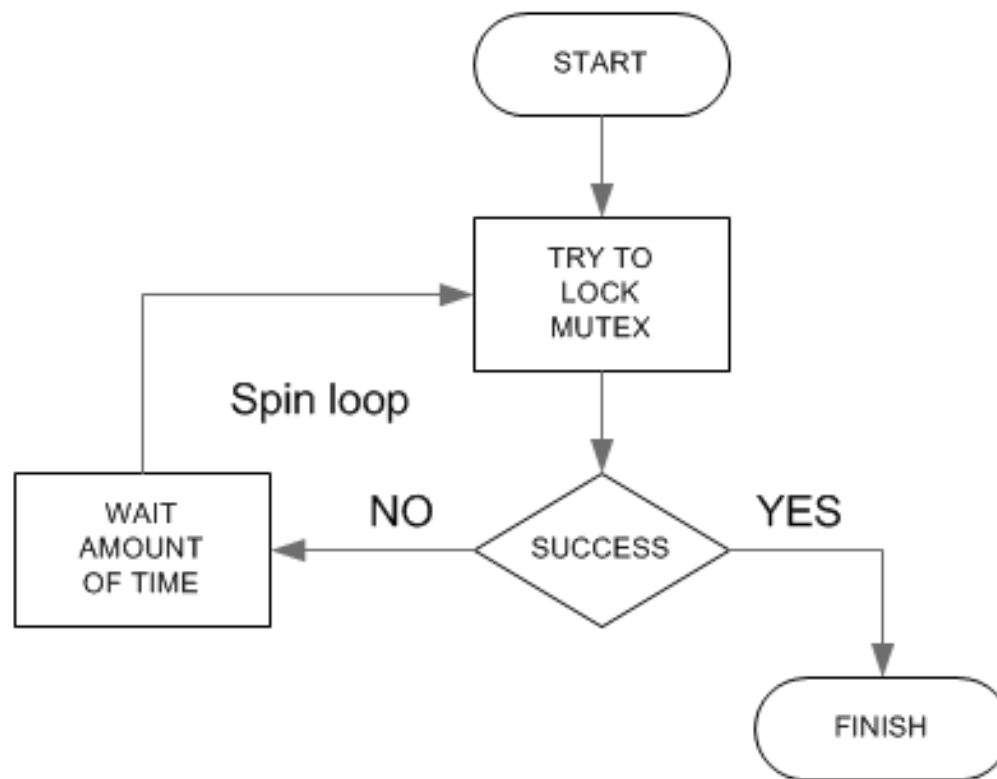


南开大学  
Nankai University



允公允能 日新月异

# Mutex



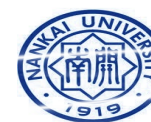




允公允能 日新月异

# Functions for Mutexes

- **WaitForSingleObject**
  - Gives a thread access to the mutex
  - Any subsequent threads attempting to gain access to it must wait
- **ReleaseMutex**
  - Called when a thread is done using the mutex
- **CreateMutex**
- **OpenMutex**
  - Gets a handle to another process's mutex



# Making Sure Only One Copy of Malware is Running

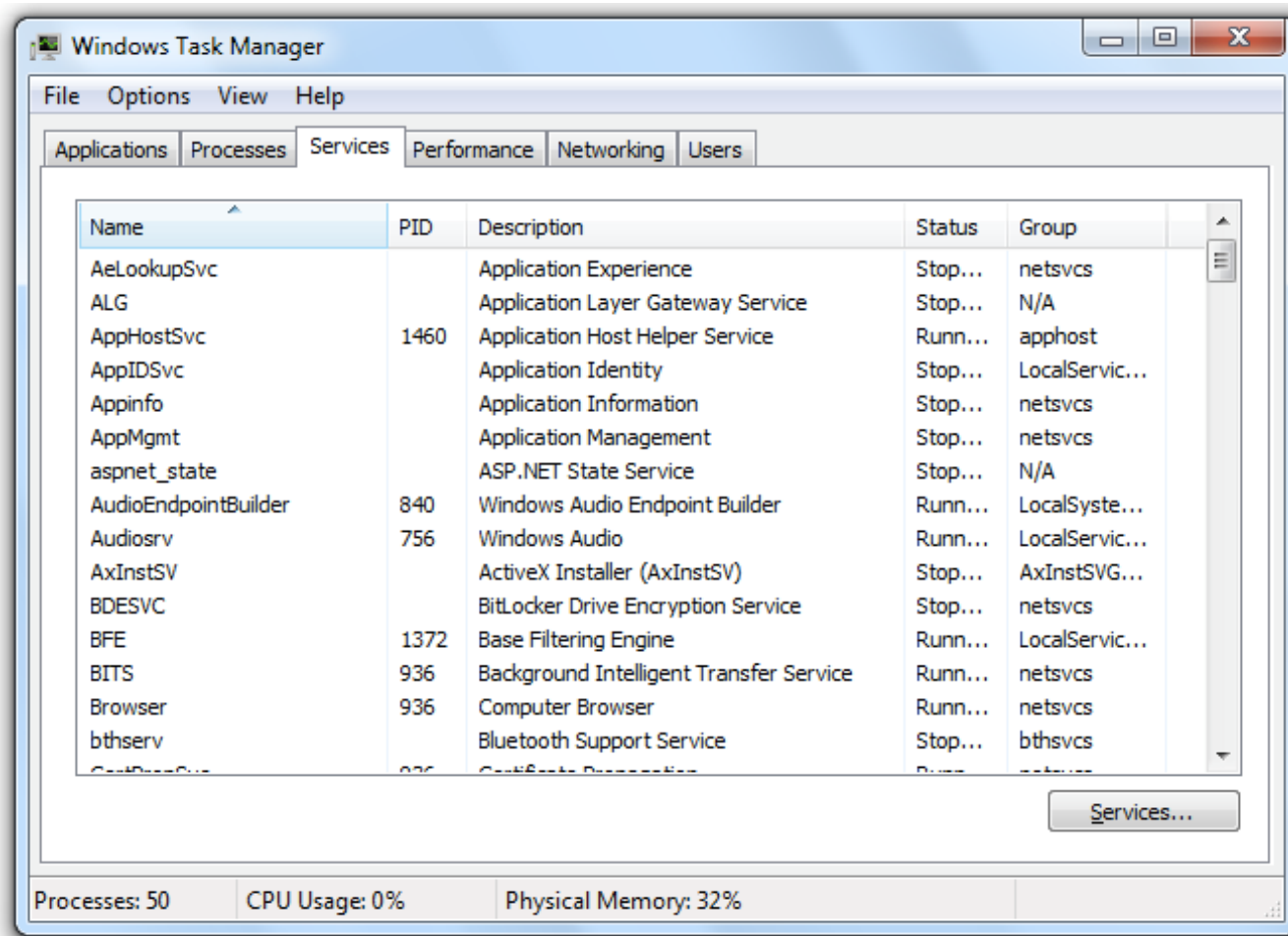
- **OpenMutex** checks if HGL345 exists
- If not, it is created with **CreateMutex**
- **test eax, eax**  
sets Z flag if eax is zero

```
01000  push  offset Name      ; "HGL345"
01005  push  0                 ; bInheritHandle
01007  push  1F0001h           ; dwDesiredAccess
0100C  ❶call  ds:__imp__OpenMutexW@12 ; OpenMutexW(x,x
01012  ❷test  eax, eax
01014  ❸jz    short loc_40101E
01016  push  0                 ; int
01018  ❹call  ds:__imp__exit
0101E  push  offset Name      ; "HGL345"
01023  push  0                 ; bInitialOwner
01025  push  0                 ; lpMutexAttributes
01027  ❺call  ds:__imp__CreateMutexW@12 ; CreateMutexW
```



# Services

- Services run in the background **without user input**





允公允能 日新月异

# SYSTEM Account

- Services often run as **SYSTEM** which is even more powerful than the Administrator
- Services can run **automatically** when Windows starts
  - An easy way for malware to maintain **persistence**
  - Persistent malware survives a restart





允公允能 日新月异

# Service API Functions

- OpenSCManager
  - Returns a handle to the **Service Control Manager**
- **CreateService**
  - Adds a new service to the Service Control Manager
  - Can specify whether the service will start automatically at boot time
- **StartService**
  - Only used if the service is set to start manually





允公允能 日新月异

# Svchost.exe

- WIN32\_ **SHARE** \_PROCESS

- Most common type of service used by malware
- Stores code for service in a DLL
- Combines several services into a single shared process named **svchost.exe**



允公允能 日新月异

# Svchost.exe in Process Explorer

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

File Options View Process Find DLL Users Help

Process	PID	CPU	Private Bytes	Working Set	Description
System Idle Process	0	97.61	0 K	24 K	
System	4	0.15	44 K	672 K	
Interrupts	n/a	0.42	0 K	0 K	Hardware Inter
smss.exe	260		224 K	792 K	Windows Sess
csrss.exe	352		2,472 K	4,160 K	Client Server R
wininit.exe	404		892 K	3,360 K	Windows Start
services.exe	508		4,312 K	6,512 K	Services and C
svchost.exe	640		2,904 K	7,208 K	Host Process f
WmiPrvSE.exe	3736		1,768 K	4,752 K	WMI Provider I
svchost.exe	708		3,196 K	6,716 K	Host Process f
svchost.exe	756		14,268 K	14,420 K	Host Process f
audiodg.exe	1680		15,016 K	14,024 K	Windows Audio
svchost.exe	840	< 0.01	44,436 K	50,672 K	Host Process f
dwm.exe	2848	0.20	88,212 K	34,328 K	Desktop Windo
svchost.exe					
svchost.exe					
svchost.exe					
spoolsv.exe					
svchost.exe					
svchost.exe					
gogoc.exe					
sqlwriter.exe					

Command Line:  
C:\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted

Path:  
C:\Windows\System32\svchost.exe (LocalSystemNetworkRestricted)

Services:  
Desktop Window Manager Session Manager [UxSms]  
Distributed Link Tracking Client [TrkWks]  
Network Connections [Netman]  
Offline Files [CscService]  
Program Compatibility Assistant Service [PcaSvc]  
Remote Desktop Services UserMode Port Redirector [UmRdpService]  
Superfetch [SysMain]  
Windows Audio Endpoint Builder [AudioEndpointBuilder]  
Windows Driver Foundation - User-mode Driver Framework [wudfsvc]

Name Description



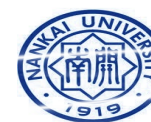




允公允能 日新月异

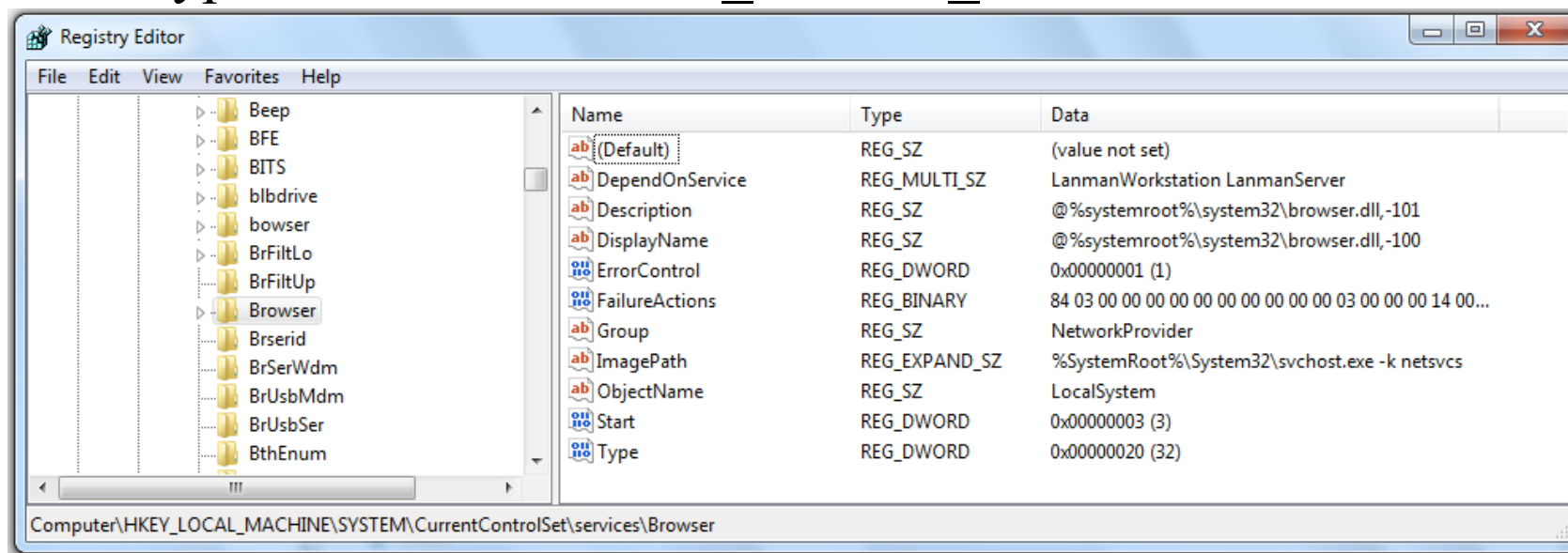
# Other Common Service Types

- WIN32\_**OWN**\_PROCESS
  - Runs as an EXE in an independent process
- KERNEL\_**DRIVER**
  - Used to load code into the Kernel



# Service Information in the Registry

- HKLM\System\CurrentControlSet\Services
- Start value = 0x03 for "Load on Demand"
- Type = 0x20 for WIN32\_SHARE\_PROCESS





允公允能 日新月异

# SC Command

- Included in Windows
- Gives information about Services

```
C:\Windows\System32>sc qc Browser
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: Browser
        TYPE               : 20  WIN32_SHARE_PROCESS
        START_TYPE           : 3    DEMAND_START
        ERROR_CONTROL         : 1    NORMAL
        BINARY_PATH_NAME      : C:\Windows\System32\svchost.exe -k netsvcs
        LOAD_ORDER_GROUP      : NetworkProvider
        TAG                   : 0
        DISPLAY_NAME          : Computer Browser
        DEPENDENCIES           : LanmanWorkstation
                             : LanmanServer
        SERVICE_START_NAME    : LocalSystem

C:\Windows\System32>
```





允公允能 日新月异

# Component Object Model (COM)

- Microsoft COM allows different software **components** to share code
  - **reuse** software component
- Client/server framework
  - Client, programs using COM object
  - Server, reusable software component



南开大学  
Nankai University



允公允能 日新月异

# Microsoft COM

- Microsoft provides a large number of COMs
  - Internet Explorer
  - Office Word
- Every thread that uses COM must call **OleInitialize** or **CoInitializeEx** before calling other COM libraries



南开大学  
Nankai University



允公允能 日新月异

# GUIDs, CLSIDs, IIDs

- COM objects are **accessed** via Globally Unique Identifiers (**GUIDs**)
- There are several types of GUIDs, including
  - Class Identifiers (**CLSIDs**)
    - in Registry at HKEY\_CLASSES\_ROOT\CLSID
  - Interface Identifiers (**IIDs**)
    - in Registry at HKEY\_CLASSES\_ROOT\Interface





允公允能 日新月异

# COM Server Malware

- Browser Helper Objects(**BHOs**)
  - third-party **plug-ins** for Internet Explorer
  - monitor Internet traffic
  - track browser usage
  - without running malware own process







允公允能 日新月异

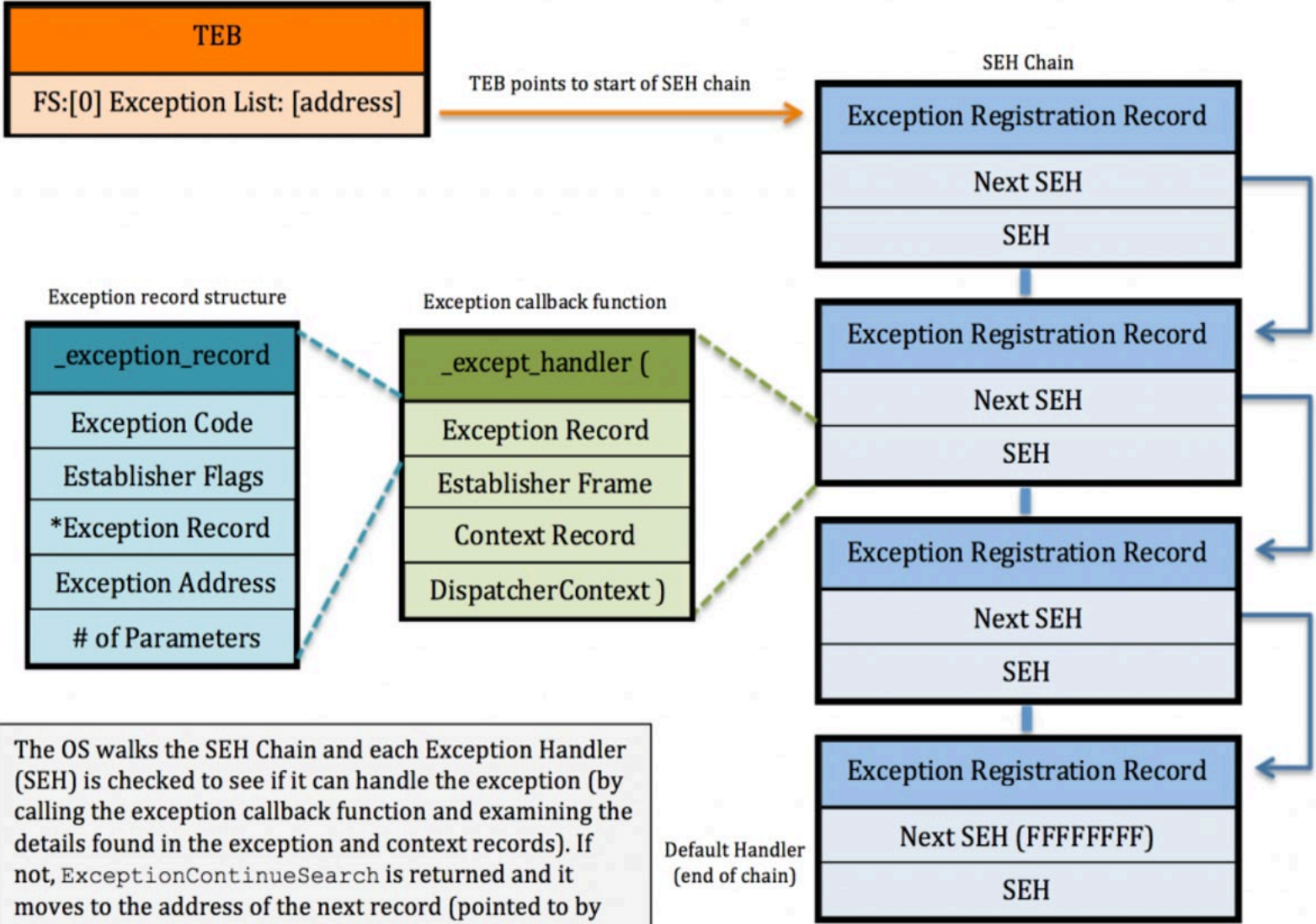
# Exceptions

- Exceptions are caused by errors, such as division by zero or invalid memory access
- When an exception occurs, execution transfers to the **Structured Exception Handler**





# Windows SEH Chain (simplified)



The OS walks the SEH Chain and each Exception Handler (SEH) is checked to see if it can handle the exception (by calling the exception callback function and examining the details found in the exception and context records). If not, ExceptionContinueSearch is returned and it moves to the address of the next record (pointed to by Next SEH) and continues down the chain until it finds a suitable exception handler or hits the last, default handler (FFFFFFFF)

# fs:0 Stores Exception Location

*Example 8-13. Storing exception-handling information in fs:0*

```
01006170  push  1offset loc_10061C0
01006175  mov    eax, large fs:0
0100617B  push  2eax
0100617C  mov    large fs:0, esp
```

- FS is one of six Segment Registers
- Structured Exception Handling(SEH)
  - MS mechanism for handling exceptions



南開大學

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月異



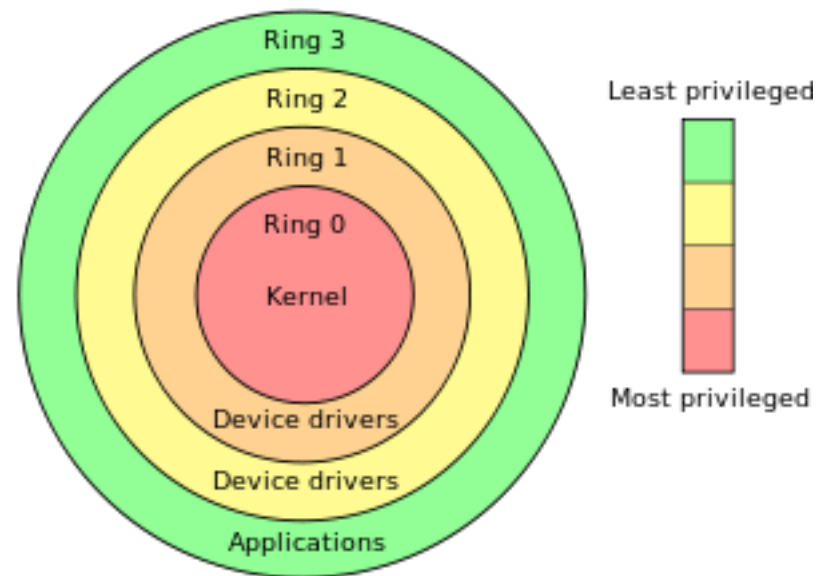
# Kernel Mode vs. User Mode



允公允能 日新月异

# Two Privilege Levels

- Ring 0: Kernel Mode
- Ring 3: User mode
- Rings 1 and 2 are not used by Windows





允公允能 日新月异

# User Mode

- Nearly all code runs in user mode
  - Except **OS** and hardware **drivers**, which run in kernel mode
- User mode cannot access **hardware** directly
- Restricted to a subset of CPU instructions
- Can only manipulate hardware through the Windows API







允公允能 日新月异

# User Mode Processes

- Each process has its own memory, security permissions, and resources
- If a user-mode program executes an invalid instruction and **crashes**,  
Windows can reclaim the resources and terminate the program







允公允能 日新月异

# Calling the Kernel

- It's not possible to jump directly from user mode to the kernel
- SYSENTER, SYSCALL, or INT 0x2E instructions use lookup **tables** to locate predefined functions





允公允能 日新月异

# Kernel Processes

- All kernel processes **share** resources and memory addresses
- Fewer security **checks**
- If kernel code executes an invalid instruction, the **OS crashes** with the Blue Screen of Death
- Antivirus software and firewalls run in Kernel mode





允公允能 日新月异

# Malware in Kernel Mode

- More powerful than user-mode malware
- **Auditing** doesn't apply to kernel
- Almost all **rootkits** use kernel code
- Most malware does not use kernel mode





南開大學

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月異



# The Native API



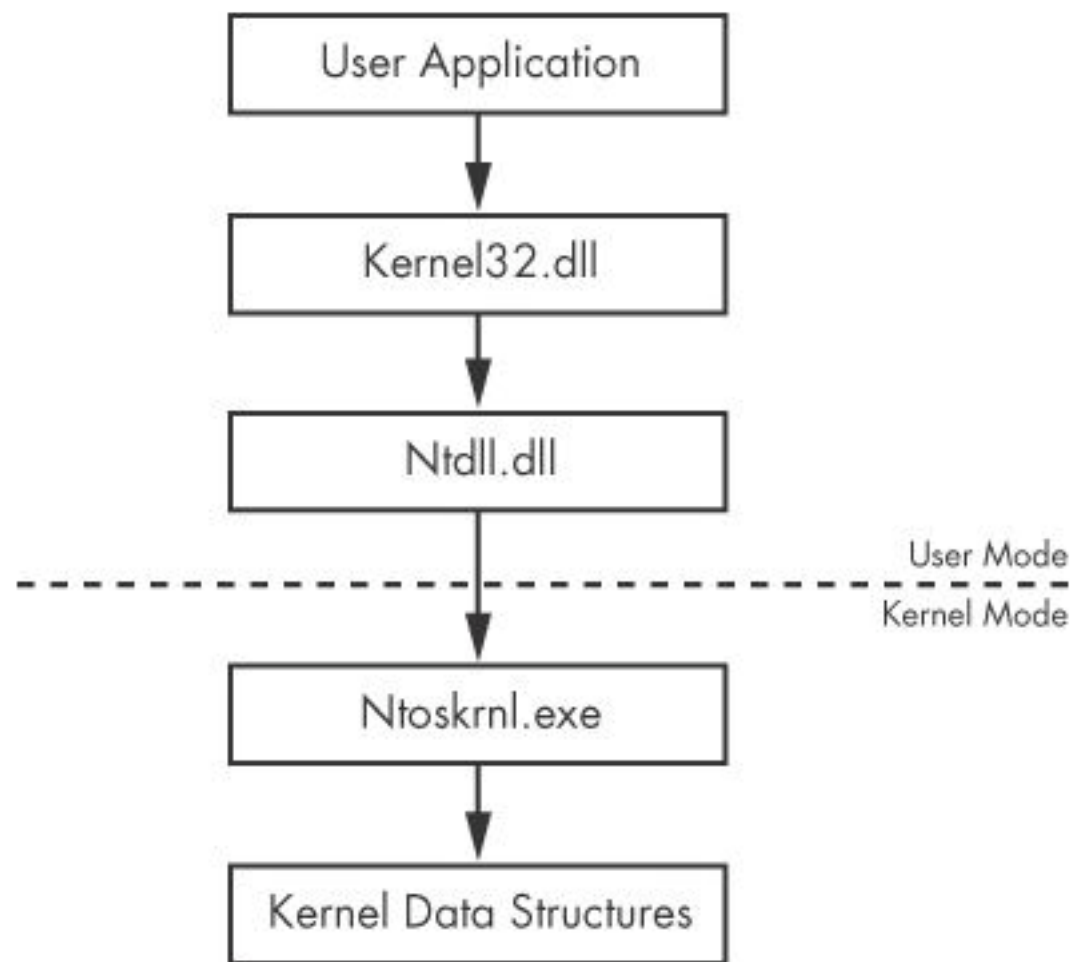
允公允能 日新月异

# The Native API

- Lower-level interface for interacting with Windows
- **Rarely** used by nonmalicious programs
- **Popular** among malware writers



- Ntdll.dll manages interactions between user space and the kernel
- Ntdll functions make up the Native API



*Figure 8-3. User mode and kernel mode*



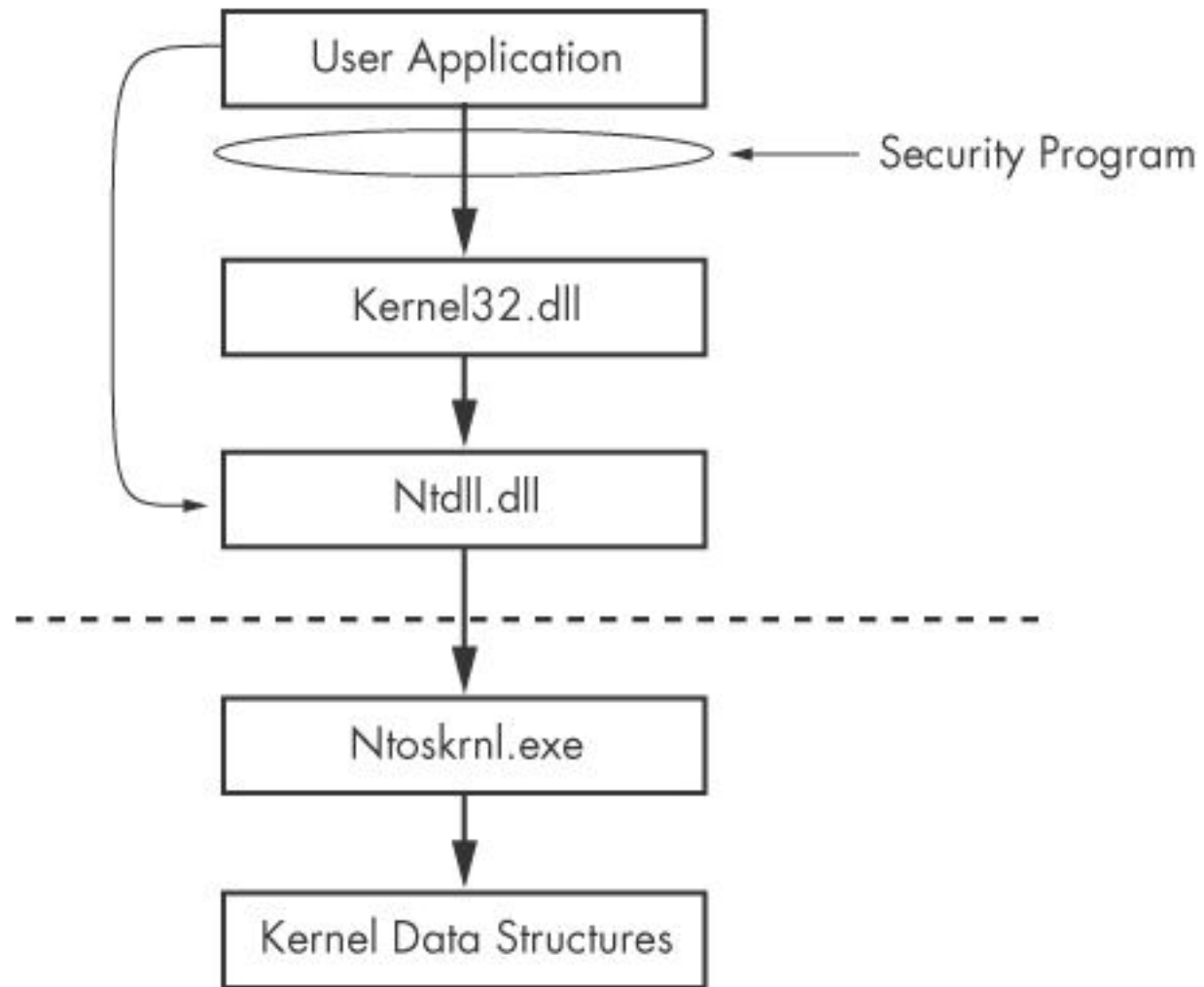
允公允能 日新月异

# The Native API

- **Undocumented**
- Intended for internal Windows use
- Can be used by programs
- Native API calls can be more **powerful** and **stealthier** than Windows API calls







*Figure 8-4. Using the Native API to avoid detection*



允公允能 日新月异

# Popular Native API Calls in Malware

- NtQuerySystemInformation
- NtQueryInformationProcess
- NtQueryInformationThread
- NtQueryInformationFile
- NtQueryInformationKey
  - Provide much more information than any available Win32 calls





允公允能 日新月异

# Popular Native API Calls in Malware

- NtContinue
  - Returns from an exception
  - Can be used to transfer execution in complicated ways
  - Used to **confuse** analysts and make a program more difficult to debug





允公允能 日新月异

# Outline

- Windows API
- Windows Registry
- Networking APIs
- Following Running Malware
- Kernel Mode vs. User Mode
- Native API





南开大学

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月异

# 恶意代码分析与防治技术

## 第7章 Windows恶意代码

王志

zwang@nankai.edu.cn

updated on 2022-10-29

南开大学 网络安全安全学院

2022/2023