Reverse Engineering: Towards Malware Analysis

Lecture – Windows Malware

Computer Security Practice

Outline

- Why Windows?
- The API
- File System
- Registry
- Internet

Why Focus on Windows?

- Most malware targets Windows platforms
- Solid understanding of Windows internals is needed valuable to analyze malware

Windows: Programming Conventions

Windows API

- Set of functionality that governs the way a program (including malware) interacts with OS libraries
- Very extensive, programmers have little need for 3rd party libraries
- Uses consistent terms, names and conventions

→ closer to english

Types and Hungarian Notation

Hungarian Notation

- Windows uses its own names for C types
- Hungarian Notation is used for API function identifiers
- Uses prefixes to make it easy to identify a variables type

Common Types

- WORD (`W`) – 16 bit unsigned value
- DWORD (`DW`) – double-WORD, 32 bit unsigned
- Handles (`H`) – reference to an object
- Long Pointer (`LP`) – pointer to another type
- Callback – function that will be called by the Windows API

65536

2

16-bit

8-bit

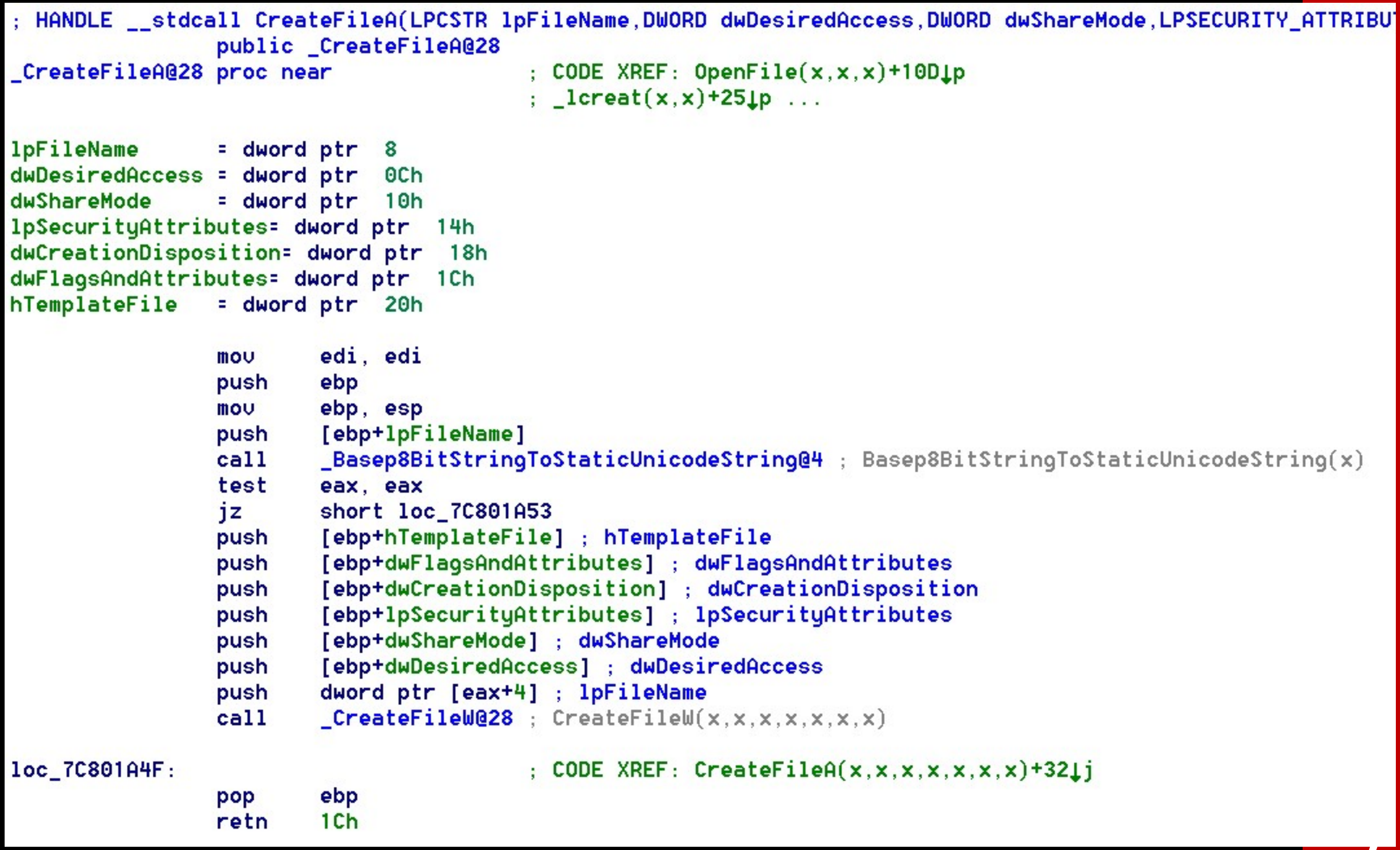
256

A-Z

a-z

Unicode and ASCII

- Windows uses Unicode under the hood
- `CreateFileA` vs. `CreateFileW`



```
; HANDLE __stdcall CreateFileA(LPCSTR lpFileName,DWORD dwDesiredAccess,DWORD dwShareMode,LPSECURITY_ATTRIBUT
                public _CreateFileA@28
_CreateFileA@28 proc near                 ; CODE XREF: OpenFile(x,x,x)+10D↓p
                                          ; _lcreat(x,x)+25↓p ...

lpFileName      = dword ptr  8
dwDesiredAccess = dword ptr  0Ch
dwShareMode     = dword ptr  10h
lpSecurityAttributes= dword ptr  14h
dwCreationDisposition= dword ptr  18h
dwFlagsAndAttributes= dword ptr  1Ch
hTemplateFile   = dword ptr  20h

                mov     edi, edi
                push    ebp
                mov     ebp, esp
                push    [ebp+lpFileName]
                call    _Basep8BitStringToStaticUnicodeString@4 ; Basep8BitStringToStaticUnicodeString(x)
                test    eax, eax
                jz      short loc_7C801A53
                push    [ebp+hTemplateFile] ; hTemplateFile
                push    [ebp+dwFlagsAndAttributes] ; dwFlagsAndAttributes
                push    [ebp+dwCreationDisposition] ; dwCreationDisposition
                push    [ebp+lpSecurityAttributes] ; lpSecurityAttributes
                push    [ebp+dwShareMode] ; dwShareMode
                push    [ebp+dwDesiredAccess] ; dwDesiredAccess
                push    dword ptr [eax+4] ; lpFileName
                call    _CreateFileW@28 ; CreateFileW(x,x,x,x,x,x,x)

loc_7C801A4F:                             ; CODE XREF: CreateFileA(x,x,x,x,x,x,x)+32↓j
                pop     ebp
                retn    1Ch
```

Handles

descriptors

file d

- Objects that have been opened or created in the OS
- Cannot be used in arithmetic operations, unlike pointers
- Stored and used in later functions to refer to the same object
- Whenever the object is referenced later, the handle is used

Malware

Windows API: Avoiding Detection by AntiVirus

Antivirus

Avoiding Detection by AntiVirus

Conventional

- Malwares exploit windows API in unconventional ways to avoid detection.

mal-code

- File Mapping – Loading a file into memory → Conv. → Unconv.
- Alternative Data Streams – hiding data within a file
- Registry Manipulation – Persistence and stealth

Interaction with the File System (FS)

- Common way for malware to interact with the FS is creating or modifying files, which can lead to good host-based indicators
- What the malware does with a file can indicate it's purpose

  Examples

  - Spyware may store browsing habits in a file
  - A file storing strings typed on the keyboard means there is keylogger functionality

Interacting with the filesystem

Loading File

- CreateFile – creates and opens files. Can also open existing files, pipes, streams and I/O devices. dwCreationDisposition controls whether it creates a new file or opens an existing file.
- ReadFile and WriteFile - Used to read and write files. Operate on files as a stream.

Conventional

- CreateFileMapping and MapViewOfFile - Commonly used by malware because it allows a file to be loaded into memory and execute it. Allows for easy jumping around a file. Does not use Windows Loader.

Unconventional

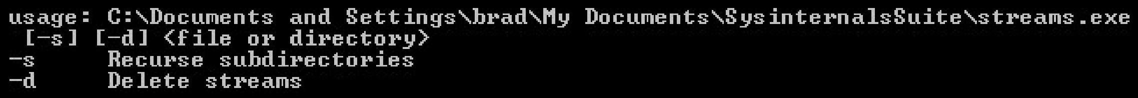
- AntiVirus usually hooks to (Create/Read/Write)File and not the advanced versions.

Alternate Data Streams (ADS)

Song

Song : artist.txt

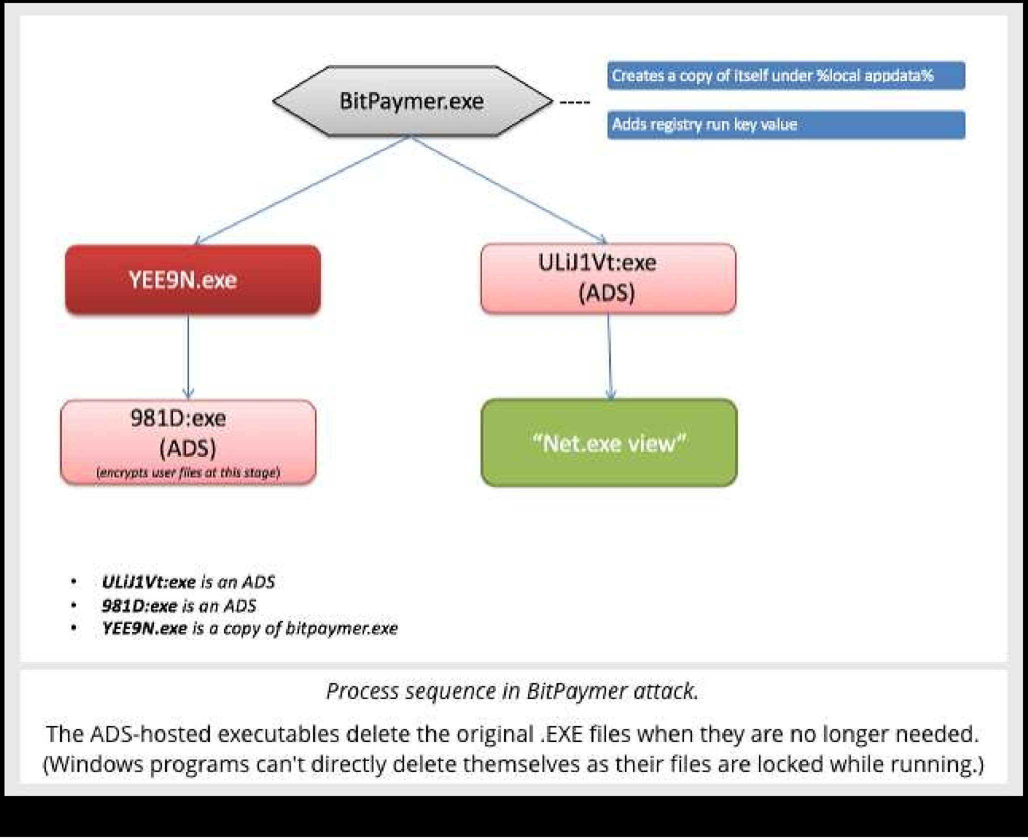
- Allows additional data to be appended to files, but not actually in the file. [what is the legitimate usecase?]
- NTFS only, does not work on FAT.
- Does not show up in a regular directory listing
  - Since Vista, `dir /R` will display ADS's
  - There are several GUI tools, as well as streams from SysInternals that will find and display ADS's
- Commonly used to hide data



```
usage: C:\Documents and Settings\brad\My Documents\SysinternalsSuite\streams.exe
 [-s] [-d] <file or directory>
-s     Recurse subdirectories
-d     Delete streams
```

Alternate Data Streams (ADS)

- Google ‘BitPaymer’ ransomware (2017)


BitPaymer.exe
Creates a copy of itself under %local appdata%
Adds registry run key value
YEE9N.exe
ULiJ1Vt:exe
(ADS)
981D:exe
(ADS)
(encrypts user files at this stage)
“Net.exe view”

- ***ULiJ1Vt:exe** is an ADS*
- ***981D:exe** is an ADS*
- ***YEE9N.exe** is a copy of bitpaymer.exe*

Process sequence in BitPaymer attack.

The ADS-hosted executables delete the original .EXE files when they are no longer needed. (Windows programs can't directly delete themselves as their files are locked while running.)

Windows Registry

- Used to store configuration information, such as settings and options
- Nearly all Windows configuration is in the registry
- Analogous: Linux conf files and rc files.

.bashrc
.zshrc
rc6

- Most used by malware for persistence or configuration data
- Can be a good source of host-based indicators and reveal clues of the malware’s purpose

Windows Registry

Definitions

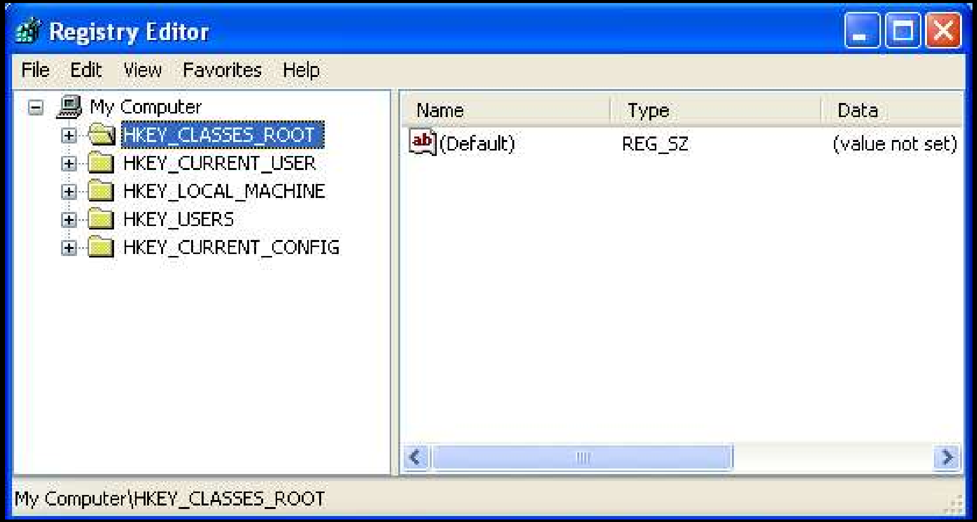
- **Root Key** – the registry is divided into 5 top-level sections called root keys. (Sometimes called `HKEY` or hive)
- **Subkey** – like a subfolder within a folder
- **Key** – Root keys and subkeys are both keys. A key is a folder in the registry that contains other keys or values
- **Value Entry** – name/value pair
- **Value or data** – data stored in a registry entry

Registry 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 config
- `HKEY_USERS` – Defines settings for the default user, new users and current users

Regedit

- Built-in utility to access and edit the registry.
- The window on the left shows open subkeys
- The right shows value entries in the subkey

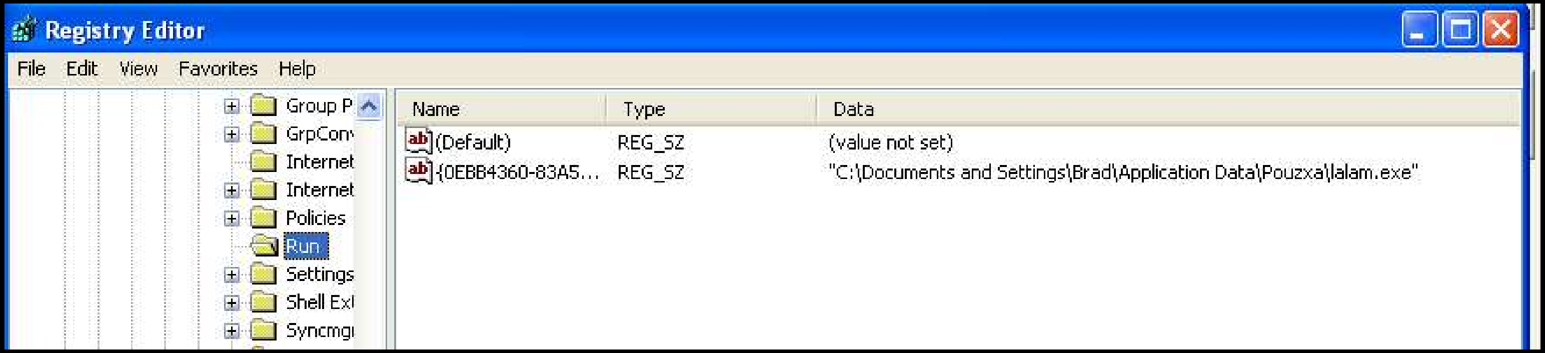


Name	Type	Data
(Default)	REG_SZ	(value not set)

My Computer\HKEY_CLASSES_ROOT

Run Subkey

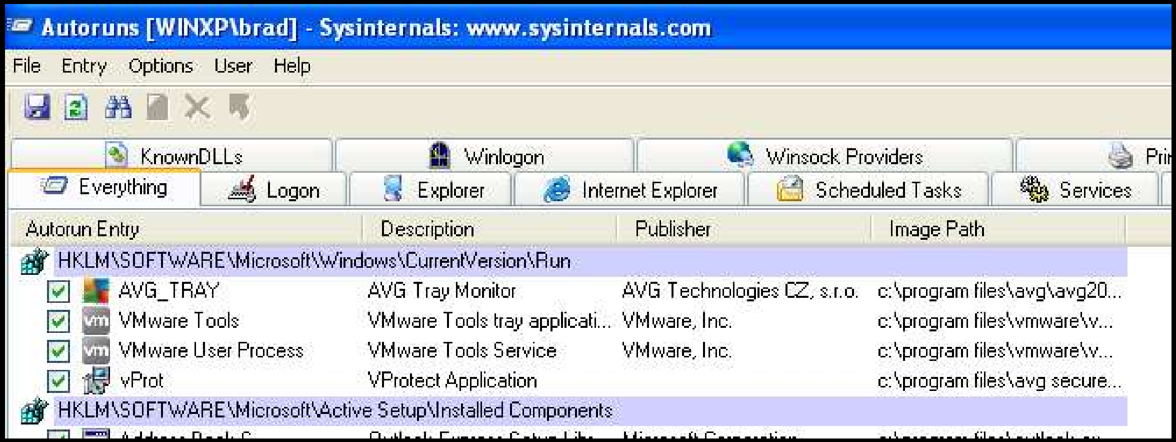
- `[HKLM|HKCU]\Software\Microsoft\Windows\CurrentVersion\Run`
- Popular place for malware to write to achieve persistence, but is not very stealthy
  - Example: Zeus trojan



Name	Type	Data
(Default)	REG_SZ	(value not set)
{0EBB4360-83A5...	REG_SZ	"C:\Documents and Settings\Brad\Application Data\Pouzxa\lalam.exe"

Autoruns

- Tool that is part of the SysInternals suite
- Checks many places in the registry that automatically run programs
- There is a GUI version and command-line



Autorun Entry	Description	Publisher	Image Path
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run			
☑ AVG_TRAY	AVG Tray Monitor	AVG Technologies CZ, s.r.o.	c:\program files\avg\avg20...
☑ VMware Tools	VMware Tools tray applicati...	VMware, Inc.	c:\program files\vmware\v...
☑ VMware User Process	VMware Tools Service	VMware, Inc.	c:\program files\vmware\v...
☑ vProt	VProtect Application		c:\program files\avg secure...
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components			

Common Malware Registry Functions

- Malware commonly uses functions that are part of the Windows API to manipulate the registry
  - `RegOpenKeyEx` – Opens a registry for editing and querying
  - `RegSetValueEx` – Adds a new value to the registry and sets it’s data
  - `RegGetValue` – Returns the value of a registry entry

Registry modification

```
• .text:00402869   lea     ecx, [esp+424h+hKey]
• .text:0040286B   push    ecx
• .text:0040286F   push    2                 ; samDesired
• .text:00402871   push    eax               ; ulOptions
• .text:00402872   push    offset SubKey     ;
"Software\\Microsoft\\Windows\\CurrentVersion\\Run"
• .text:00402877   push    HKEY_LOCAL_MACHINE ; hKey
• .text:0040287C   call    esi ; RegOpenKeyExW
• .text:0040287E   test    eax, eax
• .text:00402880   jnz     short loc_4028C5
• .text:00402882
• .text:00402882 loc_402882:
• .text:00402882   lea     ecx, [esp+424h+Data]
• .text:00402886   push    ecx               ; lpString
• .text:00402887   mov     bl, 1
• .text:00402889   call    ds:lstrlenW
• .text:0040288F   lea     edx, [eax+eax+2]
• .text:00402893   push    edx               ; cbData
• .text:00402894   mov     edx, [esp+428h+hKey]
• .text:00402898   lea     eax, [esp+428h+Data]
• .text:0040289C   push    eax               ; lpData
• .text:0040289D   push    1                 ; dwType
• .text:0040289F   push    0                 ; Reserved
• .text:004028A1   lea     ecx, [esp+434h+ValueName]
• .text:004028A8   push    ecx               ; lpValueName
• .text:004028A9   push    edx               ; hKey
• .text:004028AA   call    ds:RegSetValueExW
```

open reg

key

data

write reg

Windows API: Networking Programming

Networking API’s

- Lots of malware samples rely on networking
- There are many Windows API’s for network communication
- Malware commonly uses Berkeley compatible sockets for networking
- Berkeley compatible sockets functionality is almost identical on Windows and Unix systems

Berkeley Compatible Sockets

- socket – Creates a socket
- bind – Attaches a socket to a particular port 80
- listen – Tells a socket to listen for incoming connections
- accept – Opens/Accepts a connection with a remote socket
- connect – Opens a connection to a listening remote socket
- recv – Receives data from the remote socket
- send – Sends data to the remote socket
- ws2_32.dll

Server vs. Client

- Server side maintains an open socket waiting for an incoming connection

```
socket
bind
listen
accept
```

- Client connects to a waiting socket

```
socket
connect
```

- Malware can be either client or server

WinINet API

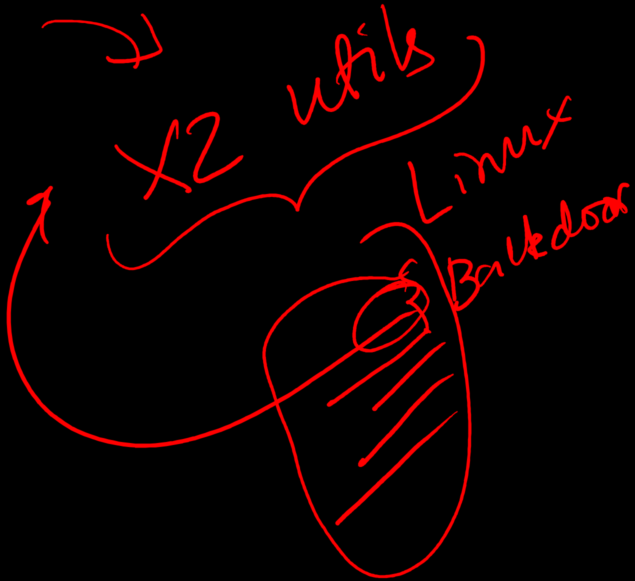
- High-level API
- Functions are stored in `Wininet.dll`
- Can tell if malware is using the WinINet API if it is importing functions from Wininet.dll
- Implements protocols such as HTTP and FTP
- Can tell more about malware based on the connections it opens

WinINet API

- `InternetOpen` – used to initialize a connection to the Internet
- `InternetOpenUrl` – connects to a URL (HTTP or FTP)
- `InternetReadFile` – allows the program to read the data from a file accessible on the internet
- So, who uses ws2_32.dll when Wininet.dll is better?

Malware authors

Legacy code



Windows: Tracking the Malware

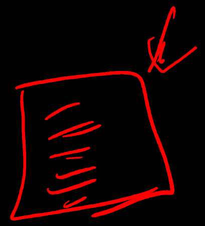
Following Malware

- There are many ways for a program to transfer execution other than jumps and calls
- It is important to analysis to realize how malware can introduce other code
- The most common way is through DLLs

DLLs

- Dynamic Link Libraries (DLL) – libraries that share code amongst many applications
- An executable file that does not run alone, but exports it’s functions for outside use
- DLL’s replace static libraries, which still exist but are not very common anymore
- Memory used by a DLL can be shared by more than one running process

How Malware Authors Use DLLs



- Store Malicious Code
  - Sometimes more convenient to store code than in an executable. Each process can only contain on executable, so malware uses DLLs to attach to other processes.
- Using Windows DLLs
  - Most malware uses the basic Windows DLLs. This is how they interact with the OS
- Using 3rd Party DLLs
  - Used to interact with other programs. If malware imports 3rd party DLL functions, it is mostly likely using that program for it’s purposes

Processes

- Malware can create a new process to execute code outside the current program
- `CreateProcess` is the function most commonly used by malware to create a new process
- Each process manages it's own resources
- A process contains one or more threads. Newer malware is increasingly multi-threaded
- There are typically many processes running on a Windows machine at any given time

CreateProcess

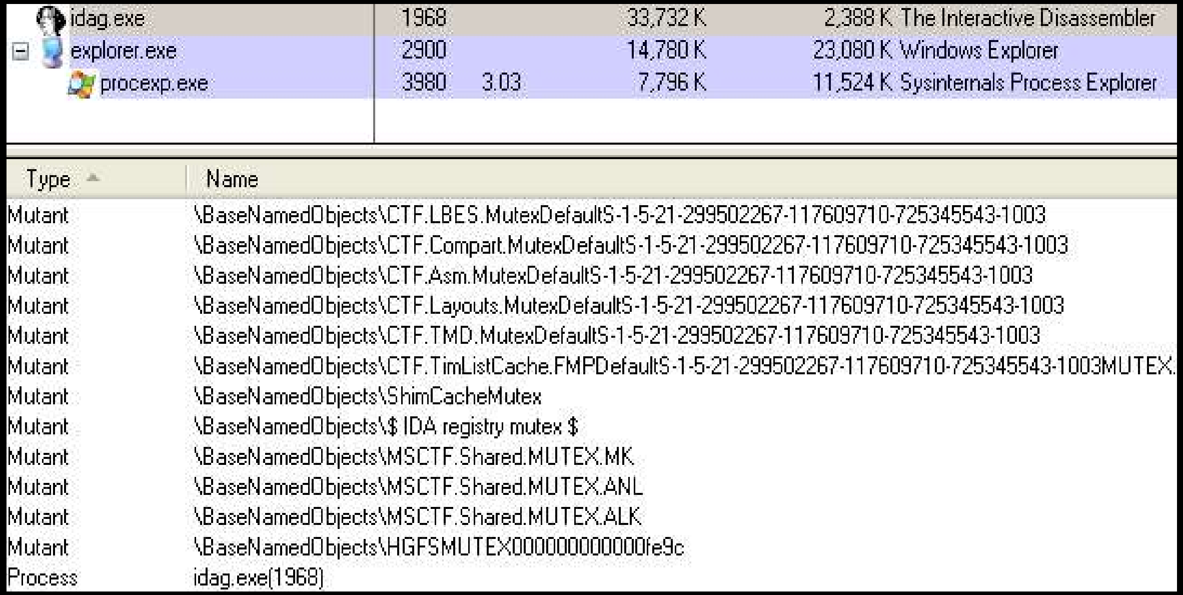
```
004010DA  mov     eax, dword ptr [esp+58h+SocketHandle]
004010DE  lea     edx, [esp+58h+StartupInfo]
004010E2  push    ecx               ; lpProcessInformation
004010E3  push    edx               ; lpStartupInfo
004010E4  mov     [esp+60h+StartupInfo.hStdError], eax
004010E8  mov     [esp+60h+StartupInfo.hStdOutput], eax
004010EC  mov     [esp+60h+StartupInfo.hStdInput], eax
004010F0  mov     eax, dword_403098
004010F5  push    0                 ; lpCurrentDirectory
004010F7  push    0                 ; lpEnvironment
004010F9  push    0                 ; dwCreationFlags
004010FB  mov     dword ptr [esp+6Ch+CommandLine], eax
004010FF  push    1                 ; bInheritHandles
00401101  push    0                 ; lpThreadAttributes
00401103  lea     eax, [esp+74h+CommandLine]
00401107  push    0                 ; lpProcessAttributes
00401109  push    eax               ; lpCommandLine
0040110A  push    0                 ; lpApplicationName
0040110C  mov     [esp+80h+StartupInfo.dwFlags], 101h
00401114  call    ds:CreateProcessA
```

Threads

- 1+ threads make up a process
- Threads are what the CPU actually executes
- Independent sequence of instructions
- Threads share memory space, but each has it's own registers and stack
- When one thread is running, it has complete control of the CPU. When the CPU switches to another thread, all register information is saved in a Thread Context, so changed values don't affect other threads
- `CreateThread` is most commonly used function to create a new thread

Mutexes

- Referred to as mutants when in the kernel
- Global objects, coordinate many processes and threads
- Mainly used to control access to shared resources
- Often used by malware



idag.exe	1968		33,732 K	2,388 K	The Interactive Disassembler
⊟ explorer.exe	2900		14,780 K	23,080 K	Windows Explorer
procexp.exe	3980	3.03	7,796 K	11,524 K	Sysinternals Process Explorer

Type	Name
Mutant	\BaseNamedObjects\CTF.LBES.MutexDefaultS-1-5-21-299502267-117609710-725345543-1003
Mutant	\BaseNamedObjects\CTF.Compart.MutexDefaultS-1-5-21-299502267-117609710-725345543-1003
Mutant	\BaseNamedObjects\CTF.Asm.MutexDefaultS-1-5-21-299502267-117609710-725345543-1003
Mutant	\BaseNamedObjects\CTF.Layouts.MutexDefaultS-1-5-21-299502267-117609710-725345543-1003
Mutant	\BaseNamedObjects\CTF.TMD.MutexDefaultS-1-5-21-299502267-117609710-725345543-1003
Mutant	\BaseNamedObjects\CTF.TimListCache.FMPDefaultS-1-5-21-299502267-117609710-725345543-1003MUTEX.
Mutant	\BaseNamedObjects\ShimCacheMutex
Mutant	\BaseNamedObjects\$ IDA registry mutex $
Mutant	\BaseNamedObjects\MSCTF.Shared.MUTEX.MK
Mutant	\BaseNamedObjects\MSCTF.Shared.MUTEX.ANL
Mutant	\BaseNamedObjects\MSCTF.Shared.MUTEX.ALK
Mutant	\BaseNamedObjects\HGFSMUTEX000000000000fe9c
Process	idag.exe(1968)

Services

- Run as background applications, without their own processes or threads
- Can be run in userspace or the kernel
- Code is scheduled and run without user input
- Normally run as `SYSTEM` or other privileged account
- Another form of persistence
- sc query/qc

Service Related Function

Persistent stealth

- `OpenSCManager` – Opens a handle to the service control (SC) manager. All code that will interact with services will use this function
- `CreateService` – Adds a new service to the SC, and allows the code to specify if the service will autorun
- `StartService` – Starts the designated service, only used if the service is set to run manually
- `StartServiceCtrlDispatcher` – Must be called by a service.

Component Object Model (COM)

- Standard that allows different software components to call each other's code without knowing the specifics of the called software
- Works with any programming language
- Used heavily in Windows, but 3rd party software only occasionally uses it
- Implemented using the client/server model
- `HKLM\SOFTWARE\Classes\CLSID\` and `HKCU\SOFTWARE\Classes\CLSID`

COM Continued

- Accessed via GUIDs, known as class identifiers (CLSIDs) and interface identifiers (IIDs)
  - `CoCreateInstance` is used to get access to COM functionality
  - `Navigate` is commonly used by malware to launch Internet Explorer and access an address (`IWebBrowser2` COM class interface)

Accessing COM object

```
• 00401024  lea   eax, [esp+18h+PointerToComObject]
• 00401028  push  eax             ; ppv
• 00401029  push  offset IID_IWebBrowser2 ; riid
• 0040102E  push  4               ; dwClsContext
• 00401030  push  0               ; pUnkOuter
• 00401032  push  offset stru_40211C ; rclsid
• 00401037  call  CoCreateInstance
```

Calling a COM object

```
0040105E  push    ecx
0040105F  push    ecx
00401060  push    ecx
00401061  mov     esi, eax
00401063  mov     eax, [esp+24h+PointerToComObject]
00401067  mov     edx, [eax]
00401069  mov     edx, [edx+2Ch]
0040106C  push    ecx
0040106D  push    esi
0040106E  push    eax
0040106F  call    edx
```

COM Server Malware

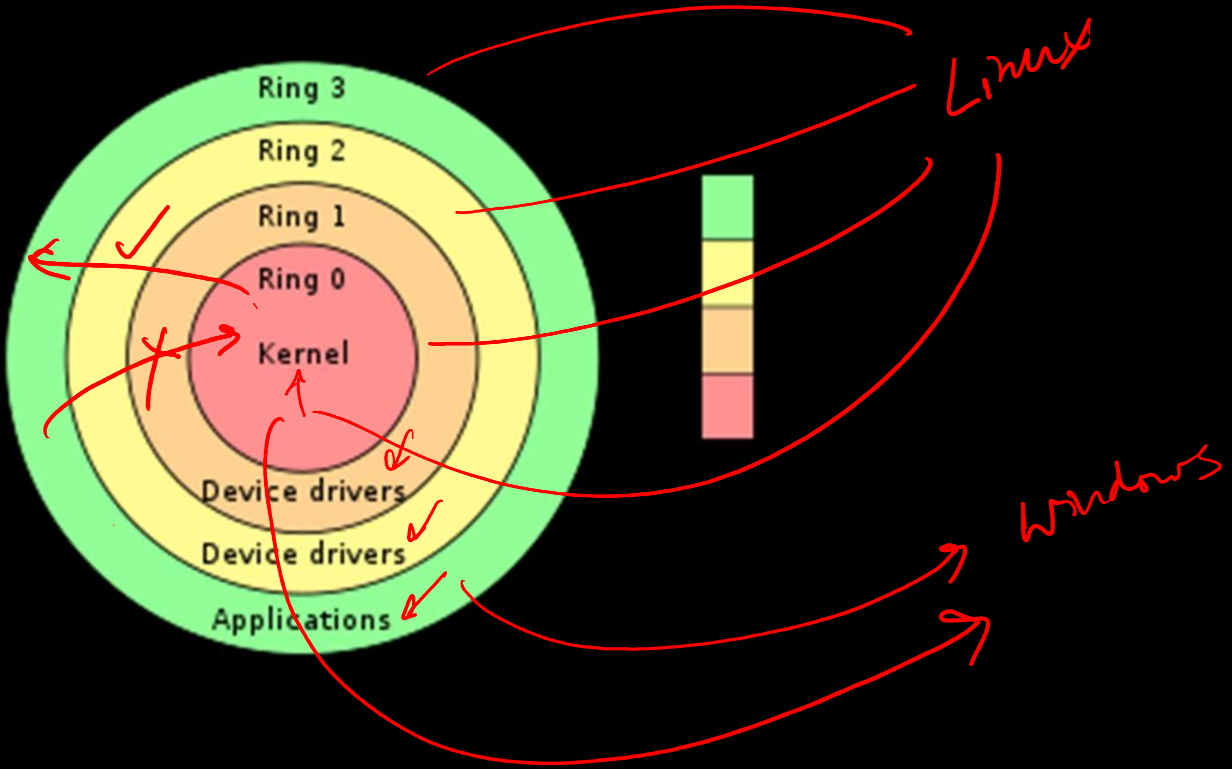
- Malware can implement a malicious COM server which is used by other programs
- Browser Helper Objects (BHOs) is a common place to implement COM server functionality
  - BHOs are 3rd party plugins for IE and can be used to run code inside the IE process
    - Example: To monitor and track internet usage

Windows: Kernel vs User Mode Malwares

Most

Common

Kernel Vs. User Mode



Kernel Vs. User Mode

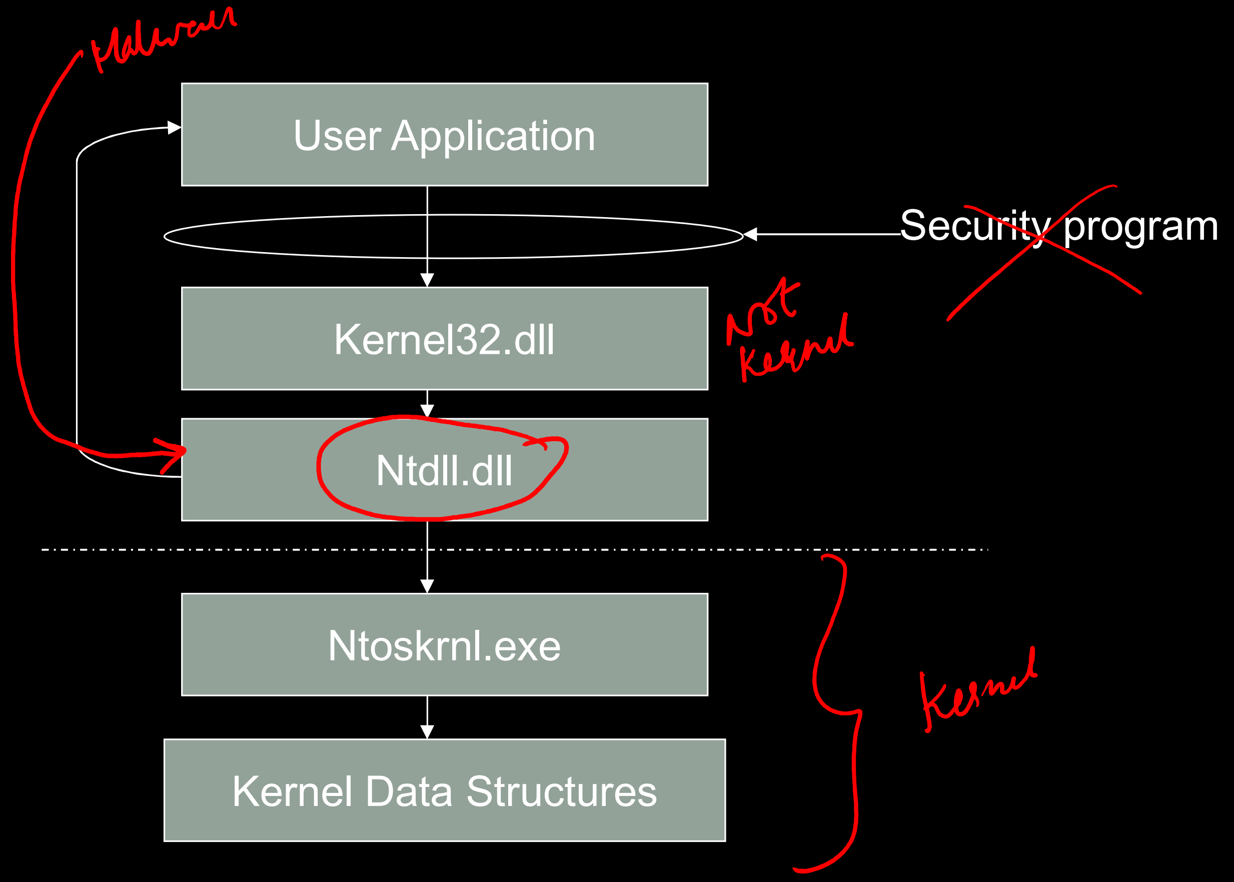
- 2 privilege levels of Windows
- All functions discussed previously are user mode
- Most code runs in user mode
  - Exceptions are the OS and hardware drivers
- Under normal circumstances, user mode cannot access hardware directly
- Kernel mode is important to malware because a lot more can be done than in user mode
- All processes running in the kernel share resources and memory
- Developing kernel mode code is much more difficult and can easily crash the system

BSOD

Native API

- Low-level API for interacting with Windows
- Rarely used by non-malicious programs → Malwares use this!
- Calling functions here bypasses the Windows API ← Win Def Anti virus
- Programs are not supposed to call the Native API, but the OS does not prevent it
- The Native API provides many functions not exposed in the Windows API
- NTDLL.DLL

Native API


Malware
User Application
~~Security program~~
Kernel32.dll
Not Kernel
Ntdll.dll
Ntoskrnl.exe
Kernel
Kernel Data Structures

?