





Lecture 8

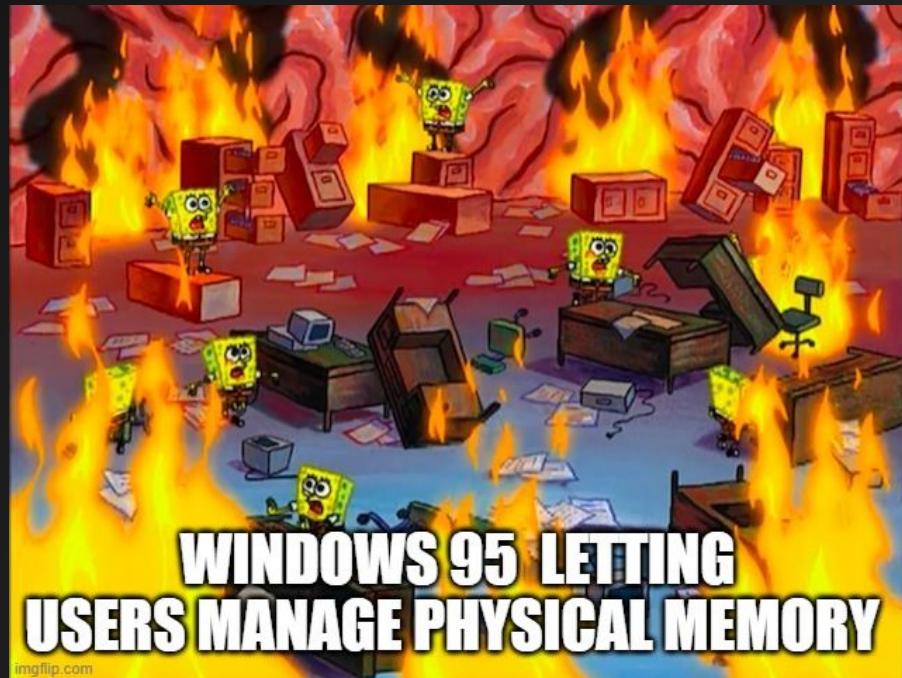
Files & I/O

Review: Kernel Objects

- Objects are created and managed by the *Kernel Object Manager*
- The Object Manager lives in kernel space and is beyond the scope of this lecture. That said...
- The object manager maintains a count of the number of references to an object.
- When an object is created, the object manager sets the object's reference count to one.
- Once that counter falls to zero, the object is freed.
- When a process ends (or crashes), all of its handles are destroyed
- Creating an object usually involves a win32 api call
`Create[ObjectName]`

Why do we interact with Objects via Handles?

- If you mess something up in a processes, it might hoard resources/crash. The resources used by the process are cleaned up when it terminates.
- If you mess something up in kernel space, resources might be locked until a reboot, or you could get an instant BSOD.
- Access control is also a plus



More on Kernel Objects

The Object Manager is responsible for (among other things)

- Creation and destruction of objects
- Controlling access
- Maintaining a list of which processes is using what object
- Managing object lifetimes: i.e. when is it safe to auto-destroy an object?

Kernel Object Examples

- Files
- Threads
- Processes
- Devices
- Mutexes
- Symbolic links
- Access Tokens

Viewing Handle Usage

Sysinternals Handle64

```
PS C:\tools\sysinternals> .\handle.exe /help
```

```
Nthandle v4.22 - Handle viewer  
Copyright (C) 1997-2019 Mark Russinovich  
Sysinternals - www.sysinternals.com
```

```
usage: handle [[-a [-l]] [-u] | [-c <handle> [-y]] | [-s]] [-p <process>|<pid>] [name] [-nobanner]  
-a      Dump all handle information.  
-l      Just show pagefile-backed section handles.  
-c      Closes the specified handle (interpreted as a hexadecimal number).  
        You must specify the process by its PID.  
        WARNING: Closing handles can cause application or system instability.  
-y      Don't prompt for close handle confirmation.  
-s      Print count of each type of handle open.  
-u      Show the owning user name when searching for handles.  
-p      Dump handles belonging to process (partial name accepted).  
name    Search for handles to objects with <name> (fragment accepted).  
-nobanner Do not display the startup banner and copyright message.
```

```
No arguments will dump all file references.
```

```
PS C:\tools\sysinternals> .\handle.exe -p brave.exe
```

```
Nthandle v4.22 - Handle viewer  
Copyright (C) 1997-2019 Mark Russinovich  
Sysinternals - www.sysinternals.com
```

Example:

```
.\handle64.exe -p brave.exe
```

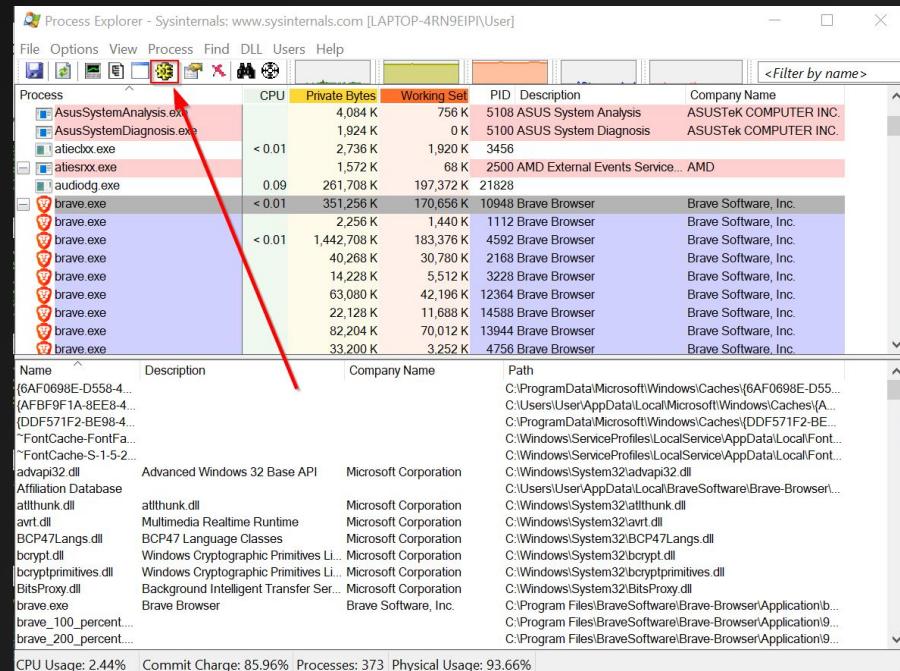
```
PS C:\tools\sysinternals> .\handle64.exe -p brave.exe

Nthandle v4.22 - Handle viewer
Copyright (C) 1997-2019 Mark Russinovich
Sysinternals - www.sysinternals.com

brave.exe pid: 10948 LAPTOP-4RN9E1PI\User
 70: File      C:\Program Files\BraveSoftware\Brave-Browser\Application\98.1.35.101
 200: Section   \Sessions\1\BaseNamedObjects\windows_shell_global_counters
 2F8: File      C:\Program Files\BraveSoftware\Brave-Browser\Application\98.1.35.101\icudtl.dat
 3A8: File      C:\Users\User\AppData\Local\BraveSoftware\Brave-Browser\User Data\BrowserMetrics\BrowserMetrics-620A6E26-2AC4.pma
 3B0: File      C:\Program Files\BraveSoftware\Brave-Browser\Application\98.1.35.101\chrome_100_percent.pak
 3B8: File      C:\Program Files\BraveSoftware\Brave-Browser\Application\98.1.35.101\chrome_200_percent.pak
 3C0: File      C:\Program Files\BraveSoftware\Brave-Browser\Application\98.1.35.101\Locales\en-US.pak
 3C8: File      C:\Program Files\BraveSoftware\Brave-Browser\Application\98.1.35.101\resources.pak
 3D0: File      C:\Program Files\BraveSoftware\Brave-Browser\Application\98.1.35.101\brave_resources.pak
 3D8: File      C:\Program Files\BraveSoftware\Brave-Browser\Application\98.1.35.101\brave_100_percent.pak
 3E0: File      C:\Program Files\BraveSoftware\Brave-Browser\Application\98.1.35.101\brave_200_percent.pak
 420: File      C:\Users\User\AppData\Local\BraveSoftware\Brave-Browser\User Data\Default\Local Extension Settings\mnojpmjdmbbfmejpflffi
fhffcmidifd\LOCK
 4A8: File      C:\Program Files\BraveSoftware\Brave-Browser\Application\98.1.35.101
 524: File      C:\Windows\WinSxS\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.19041.1110_none_60b5254171f9507e
 5B0: Section   \Windows\Theme1197514762
 5B4: Section   \Sessions\1\Windows\Theme4022028298
 5C0: Section   \BaseNamedObjects\__ComCatalogCache__
 5CC: Section   \BaseNamedObjects\__ComCatalogCache__
 5D0: File      C:\Windows\Registration\R000000000007.clb
 654: Section   \Sessions\1\BaseNamedObjects\2ac4HWNDFInterface:80072
```

Process Explorer

- Task manager on steroids
- Provides information about processes, parent/child relationships, Handles, DLLs, in memory strings... and more!
- To View handle we click



The screenshot shows the Process Explorer application window. The title bar reads "Process Explorer - Sysinternals: www.sysinternals.com [LAPTOP-4RN9E\PN\User]". The main interface displays a table of processes with columns: Process, CPU, Private Bytes, Working Set, PID, Description, and Company Name. A red arrow points from the text "To View handle we click" to the handle icon in the toolbar above the table. The table lists several processes, including "brave.exe" which is highlighted. Below the table, there is a detailed list of handles with columns: Name, Description, Company Name, and Path. The bottom status bar shows "CPU Usage: 2.44%" and "Commit Charge: 85.96% Processes: 373 Physical Usage: 93.66%".

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
AsusSystemAnalysis.exe	4,084 K	756 K	5108 ASUS System Analysis	ASUSTek COMPUTER INC.		
AsusSystemDiagnosis.exe	1,924 K	0 K	5100 ASUS System Diagnosis	ASUSTek COMPUTER INC.		
atiexec.exe	< 0.01	2,736 K	1,920 K	3456		
atiexec.exe		1,572 K	68 K	2500 AMD External Events Service...	AMD	
audiogd.exe	0.09	261,708 K	197,372 K	21282		
brave.exe	< 0.01	351,256 K	170,656 K	10948	Brave Browser	Brave Software, Inc.
brave.exe		2,256 K	1,440 K	1112	Brave Browser	Brave Software, Inc.
brave.exe		1,442,708 K	183,376 K	4592	Brave Browser	Brave Software, Inc.
brave.exe		40,268 K	30,780 K	2168	Brave Browser	Brave Software, Inc.
brave.exe		14,228 K	5,512 K	3228	Brave Browser	Brave Software, Inc.
brave.exe		63,080 K	42,196 K	12364	Brave Browser	Brave Software, Inc.
brave.exe		22,128 K	11,688 K	14588	Brave Browser	Brave Software, Inc.
brave.exe		82,204 K	70,012 K	13944	Brave Browser	Brave Software, Inc.
brave.exe		33,200 K	3,252 K	4756	Brave Browser	Brave Software, Inc.

Name	Description	Company Name	Path
(6AF0698E-D558-4...			C:\ProgramData\Microsoft\Windows\Caches\{6AF0698E-D55...
(AEBF9F1A-8EE8-4...			C:\Users\User\AppData\Local\Microsoft\Windows\Caches\{A...
{DDF571F2-BE98-4...			C:\ProgramData\Microsoft\Windows\Caches\{DDF571F2-BE...
FontCache-FontFa...			C:\Windows\ServiceProfiles\LocalService\AppData\Local\Font...
FontCache-S-1-5-2...			C:\Windows\ServiceProfiles\LocalService\AppData\Local\Font...
adapi32.dll	Advanced Windows 32 Base API	Microsoft Corporation	C:\Windows\System32\adapi32.dll
Affiliation Database			C:\Users\User\AppData\Local\BraveSoftware\Brave-Browser\...
alithunk.dll	alithunk.dll	Microsoft Corporation	C:\Windows\System32\alithunk.dll
avrt.dll	Multimedia Realtime Runtime	Microsoft Corporation	C:\Windows\System32\avrt.dll
BCP47Langs.dll	BCP47 Language Classes	Microsoft Corporation	C:\Windows\System32\BCP47Langs.dll
bcrypt.dll	Windows Cryptographic Primitives Li...	Microsoft Corporation	C:\Windows\System32\bcrypt.dll
bcryptprimitives.dll	Windows Cryptographic Primitives Li...	Microsoft Corporation	C:\Windows\System32\bcryptprimitives.dll
BitsProxy.dll	Background Intelligent Transfer Ser...	Microsoft Corporation	C:\Windows\System32\BitsProxy.dll
brave.exe	Brave Browser	Brave Software, Inc.	C:\Program Files\BraveSoftware\Brave-Browser\Application\9...
brave_100_percent...			C:\Program Files\BraveSoftware\Brave-Browser\Application\9...
brave_200_percent...			C:\Program Files\BraveSoftware\Brave-Browser\Application\9...

File Objects

- File Objects are a type of Kernel Objects
- “File objects function as the logical interface between kernel and user-mode processes and the file data that resides on the physical disk.”
- File Object != Disk file, but all disk files are File Objects
- Files on the filesystem (most of the time) are actually symbolic links to some file saved on a physical disk

The Windows I/O System

- Any (useful) operating system needs to be able to interact with external entities called “devices”
- By interact, I mean receive input and send output to external devices.
- For example, our PCs wouldn’t be very useful to us without the ability to send keystrokes via a keyboard, and have the computer display output to the screen
- A *driver* provides a *software interface* for the exchange of data between the device and OS
- The Windows I/O manager is a kernel feature that “manages the communication between applications and the interfaces provided by device drivers”
- In other words, the IO system abstracts away complicated, and dangerous parts of performing I/O on physical and logical devices

Windows I/O system: Userland perspective

- Userland processes call into the I/O system using the Windows (or Native) API
- Any File I/O operation is initiated by a kernel mode entity within the I/O system called the *I/O Manager*
- In order to initiate this processes, the Native API invokes a *syscall*
- Recall that a syscall is the lowest level, userland function available to us: this allows userland processes to call into Kernel space via an API

The Windows File system

- From the perspective of an application/user, files are organized in a Tree structure
- There are many different types of files, but the basic ones are directories, and Regular files
- Directories can have child nodes
- Regular files cannot have child nodes



File Paths

- Unix Paths start at the root directory “/”
- A path is either a *relative* or *absolute* path to a file
- Example: “`../folder_a/file_a.txt`” is a relative path
- “`..`” is the previous directory one level out
- “`.`” is the current directory
- Example: “`/home/remnux/Desktop/inetsim.conf`” is an absolute path to a file
- On Windows, we use “`\`” as opposed to “`/`” for a path separator

Tree Structure

`tree /f .` creates an ASCII art tree representation of the tree structure with root node equal to the current directory

It then “Walks” the tree and displays all files and directories

```
C:\dev\CS-501-malware-course\Assignments>tree /f .
Folder PATH listing for volume OS
Volume serial number is [REDACTED]
C:\DEV\CS-501-MALWARE-COURSE\ASSIGNMENTS
    HW0
        hw0.exe

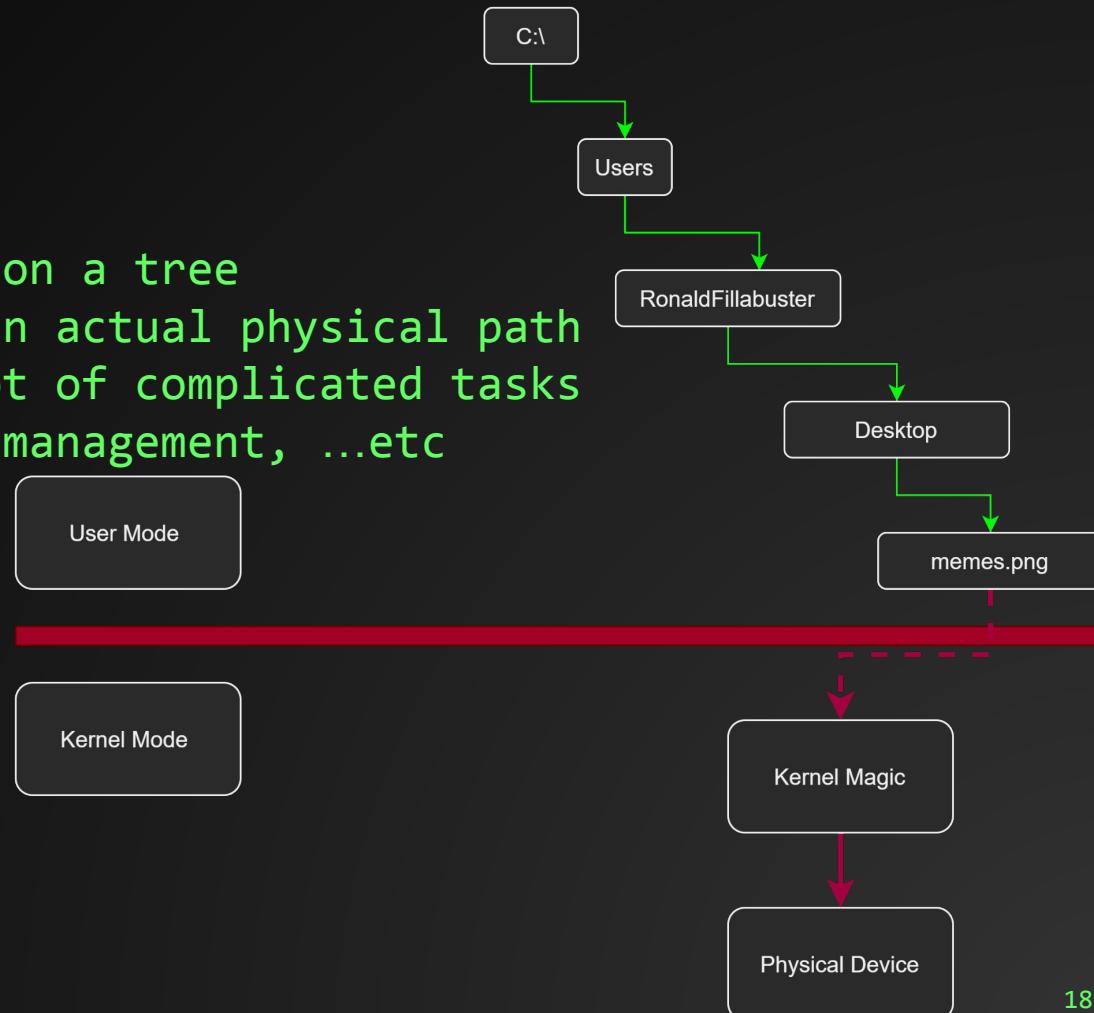
    HW1
        activate.c
        fibonacci.c
        highground.c
        Makefile
        myls.c
        mystrings.c
        reverse.c

    HW2
        CrackMe
            crackme0.exe
            crackme1.exe
            crackme2.exe
            crackme3.exe
            crackme4.exe
            crackme5.exe
            crackme6.exe
            crackme7.exe
            crackme8.exe

    Suspicious
```

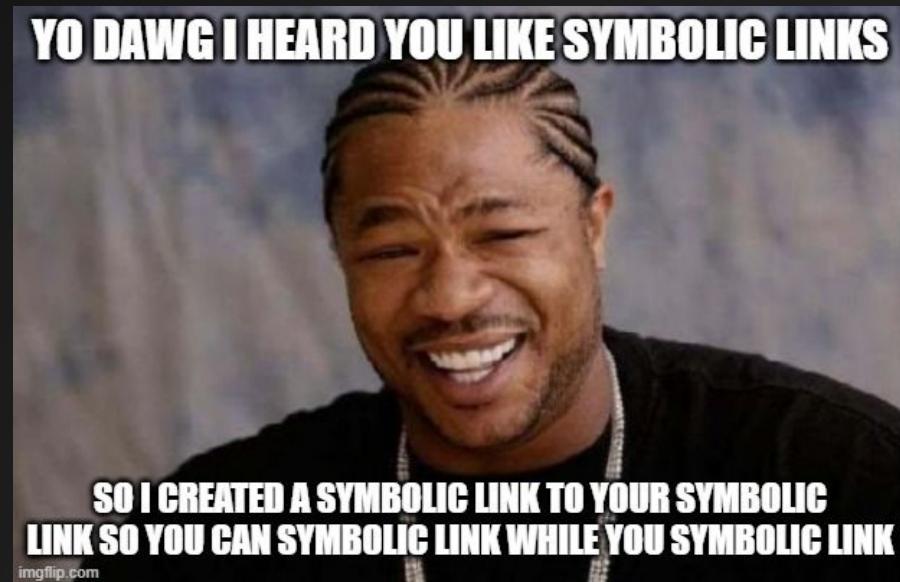
Directory Tree

- Paths are always paths on a tree
- The path is mapped to an actual physical path
- The kernel handles a lot of complicated tasks such as Disk IO, power management, ...etc



Symbolic Links

- A symbolic link is an alias for a symbolic link target
- On Windows, we usually pass symbolic links as arguments to functions
- Logical drives like C:\ are actually symbolic links to (usually) volumes on a disk
- WinObj64 is a sysinternals tool that allows us to view Kernel objects including Symbolic links



C:\ is a symbolic link

WinObj - Sysinternals: www.sysinternals.com

File Edit Find View Options Help

Quick Find: C:

Name	Type	Symbolic Link Target
∞ C:	SymbolicLink	\Device\HarddiskVolume6

\\ \ ArcName BaseNamedObjects Callback Device Driver DriverStores FileSystem GLOBAL?? KernelObjects KnownDlls KnownDlls32 NLS ObjectTypes RPC Control Security Sessions UMDFCommunicationPorts Windows

The screenshot shows the WinObj application interface. On the left is a tree view of system objects under the root. In the center is a table with three columns: Name, Type, and Symbolic Link Target. The entry for 'C:' is highlighted with a blue selection bar. Two black arrows point from the text 'C:\ is a symbolic link' at the top of the slide to the 'Name' and 'Symbolic Link Target' columns respectively in the table.

Terminology: Files

In this class, we will use the following Convention for talking about File Objects:

- File: What you normally think of when someone says a file.
I.e., something with a symbolic link on disk.
- File Object: A kernel object

If I say File, I mean a file on disk

If I say File Object, I am referring to a Kernel Object

Getting a Handle to a File Object

- `CreateFile(A|W)`: Win32 call to create a handle to a File Object. Supports all versions of Windows.
- `CreateFile2`: Similar to `CreateFileW`, but can be called from a special type of Application called a Universal Windows Platform (UWP). Is only supported on Windows 8 and above
- `NtCreateFile`: Native API call to create a File Object. All of the functions above call into it!

CreateFileA

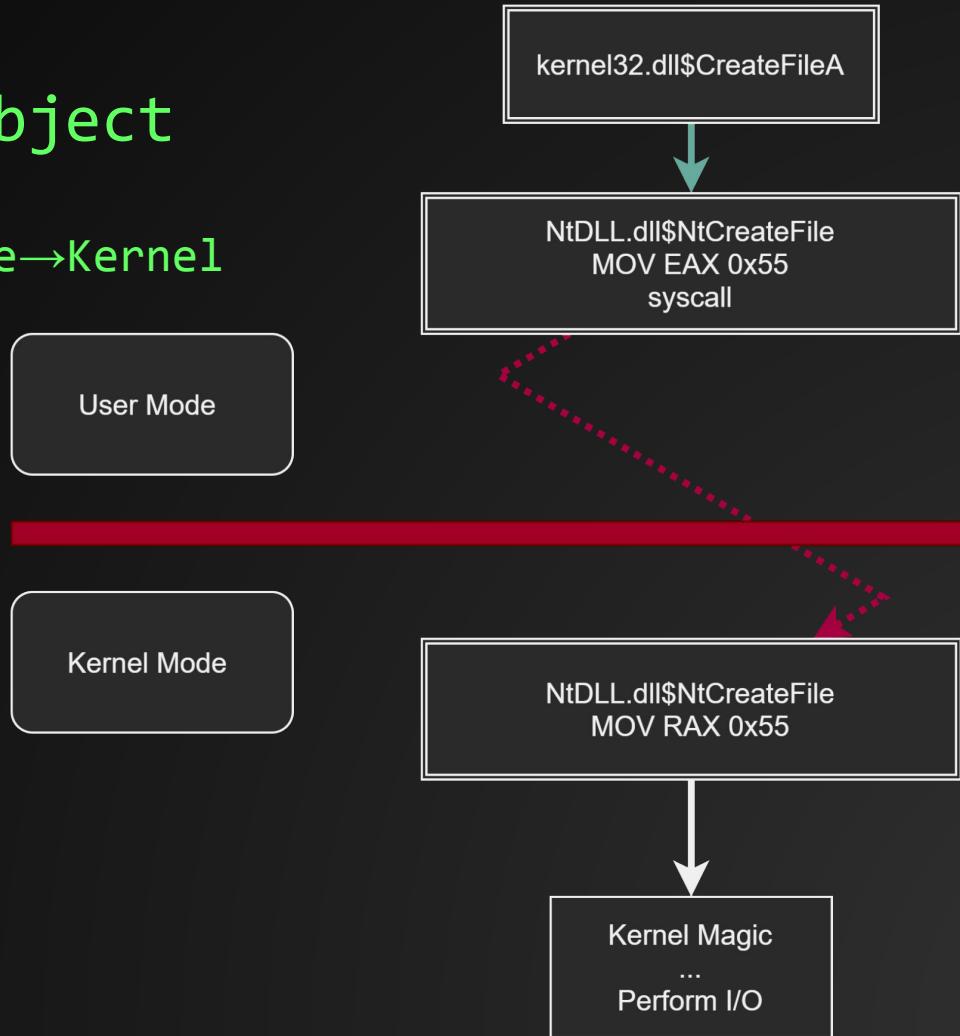
```
HANDLE CreateFileA(  
    [in]          LPCSTR          lpFileName,  
    [in]          DWORD           dwDesiredAccess,  
    [in]          DWORD           dwShareMode,  
    [in, optional] LPSECURITY_ATTRIBUTES lpSecurityAttributes,  
    [in]          DWORD           dwCreationDisposition,  
    [in]          DWORD           dwFlagsAndAttributes,  
    [in, optional] HANDLE          hTemplateFile  
) ;
```

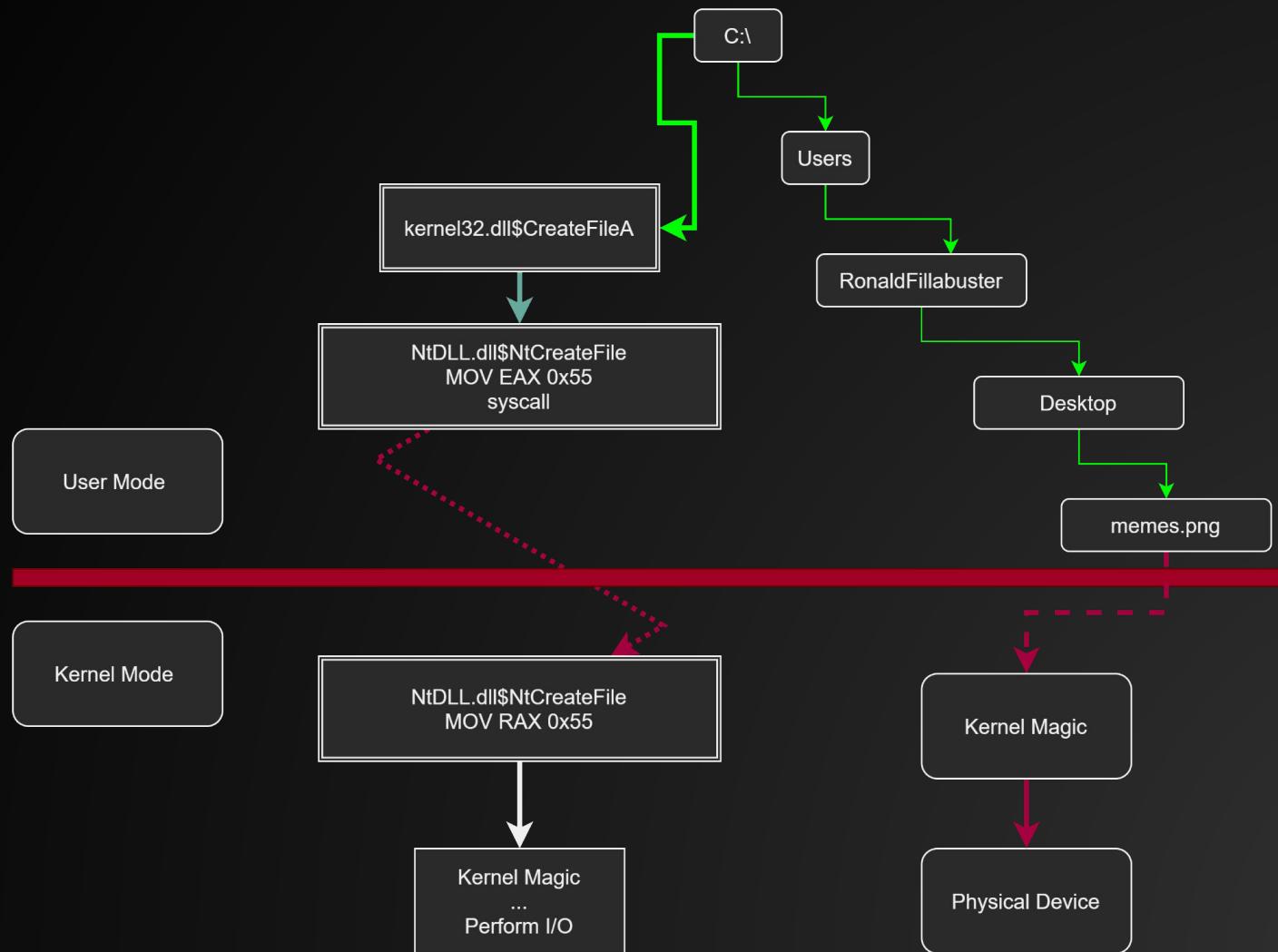
- Demo: reading the Docs
- <https://docs.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-createfilea>

Creating a File Object

CreateFileA → NtCreateFile → Kernel
Magic

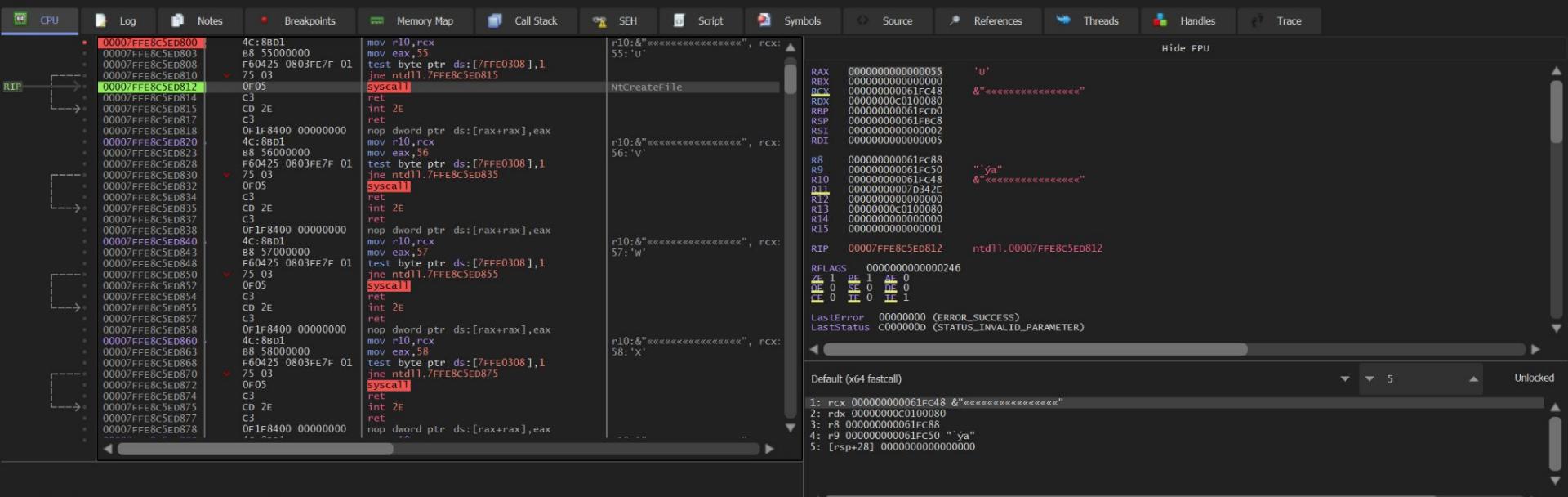
I/O System abstraction





NtCreateFile

00007FFE8C5ED812 | 0F05 | syscall | NtCreateFile

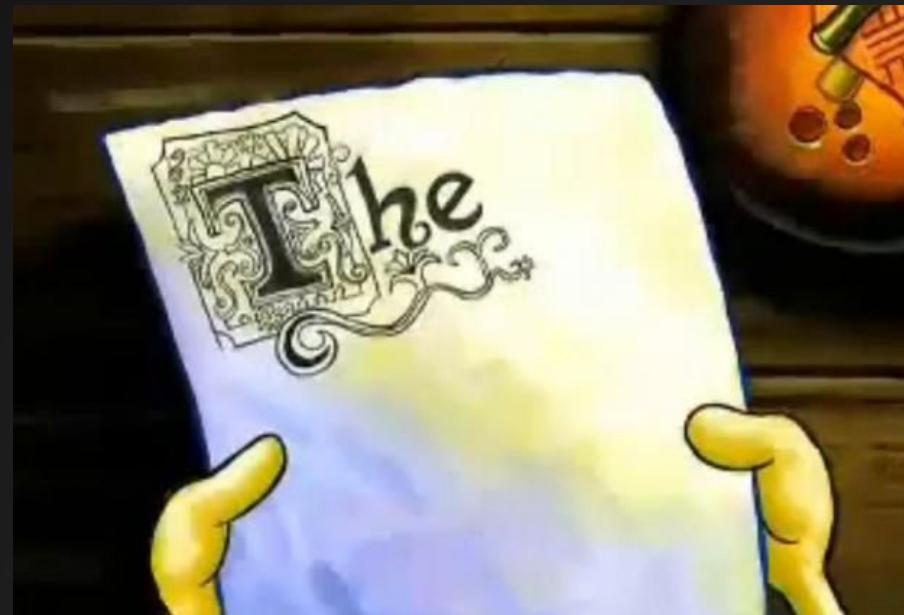


Mapping Symbolic Links to Target paths

```
DWORD QueryDosDeviceA(  
    [in, optional] LPCSTR lpDeviceName,  
    [out]          LPSTR  lpTargetPath,  
    [in]           DWORD   ucchMax  
)
```

Querying File Information

- Calculating the File Size
- Reading/modifying file “times”
- Reading/modifying File Attributes
- How? You will figure it out on your homework :D



File Attributes

- Metadata associated with file objects committed to disk
- When creating Files, you can control these attributes with File Attribute masks
- Examples:
 - FILE_ATTRIBUTE_NORMAL The mask you use for normal files
 - FILE_ATTRIBUTE_DIRECTORY The file is a directory
 - FILE_ATTRIBUTE_HIDDEN The file is hidden
 - FILE_ATTRIBUTE_NOT_CONTENT_INDEXED The file won't be indexed
 - FILE_ATTRIBUTE_READONLY The file is read only.
 - FILE_ATTRIBUTE_TEMPORARY File attempts to stay in memory.
- Full:<https://docs.microsoft.com/en-us/windows/win32/fileio/file-attribute-constants>

Common Footguns

- DWORD is an unsigned 32 bit integer
- When Windows NT was first rolled out, the thought of having lots of files larger than 2^{32} bytes was a bit of an insane/edge case prospect
- 2^{32} bytes is about 4.3 Gigabytes
- On the other hand, 2^{64} bytes is about $1.8 * 10^{10}$ gigabytes
 - Or 18,446,744.1 terabytes. That is pretty large. For now, 64bit unsigned integers should be sufficient to house file sizes
- GetFileSize returns a 32bit unsigned integer
- If it is larger than 4gbs, then you need to use lpFileSizeHigh
- If you set lpFileSizeHigh to null, and fs>4gb...you will have a bad time
- Please use GetFileSizeEx

Parameters

[in] hFile

A handle to the file.

[out, optional] lpFileSizeHigh

A pointer to the variable where the high-order doubleword of the file size is returned. This parameter can be NULL if the application does not require the high-order doubleword.

Return value

If the function succeeds, the return value is the low-order doubleword of the file size, and, if lpFileSizeHigh is non-NULL, the function puts the high-order doubleword of the file size into the variable pointed to by that parameter.

If the function fails and lpFileSizeHigh is NULL, the return value is INVALID_FILE_SIZE. To get extended error information, call GetLastError. When lpFileSizeHigh is NULL, the results returned for large files are ambiguous, and you will not be able to determine the actual size of the file. It is recommended that you use GetFileSizeEx instead.

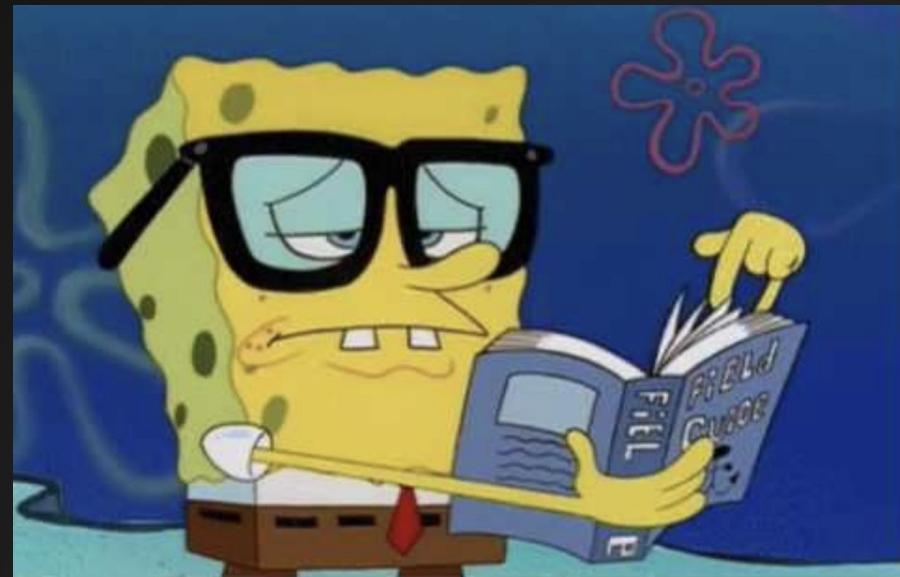
If the function fails and lpFileSizeHigh is non-NULL, the return value is INVALID_FILE_SIZE and GetLastError will return a value other than NO_ERROR.

Synchronous File I/O

- Windows performs I/O asynchronously under the hood
- That said, async I/O is considerably more complicated than Synchronous I/O
- Here synchronous means the thread performing I/O is blocked (in a waiting state) until the I/O is completed
- Async allows the the thread to go do other work while it waits for the I/O to finish
- For now, we will only look at Synchronous I/O

File I/O: Reading

- Like everything involving I/O, we first call `CreateFile`
- You should then determine the size of the file
- Allocate a buffer large enough to fit all of the data.
 - If the data is to be interpreted as a string, make sure to have enough space for an added null byte
 - If the file is Unicode encoded, make sure to have enough space for a wide char null terminator!



Remarks on Memory Allocation

- We will dedicate more time to this later on in the class, but you should currently be familiar with C-runtime style memory management. I.e., `malloc` `free`
- C++: `new/delete`
 - Whereas `malloc` only lets you allocate in terms of bytes, `new` lets you allocate space according to an objects type!
 - I.e., `malloc(sizeof(int) * 100)` vs `new int[100];`
- Modern C++: Smart pointers
 - `<memory>` → `std::make_unique` ← smart pointer that is automatically freed once the object goes out of scope!

Question: Reading Files

If we want to read a file into memory and print its contents, do we use stack or heap memory?

Answer:

Either. Heap is more flexible, and can handle arbitrary files.
Stack can read a file in chunks, or have some maximum size file.
....but you should still use Heap memory :-)

Protecting Memory

What happens when we free a block of sensitive memory?

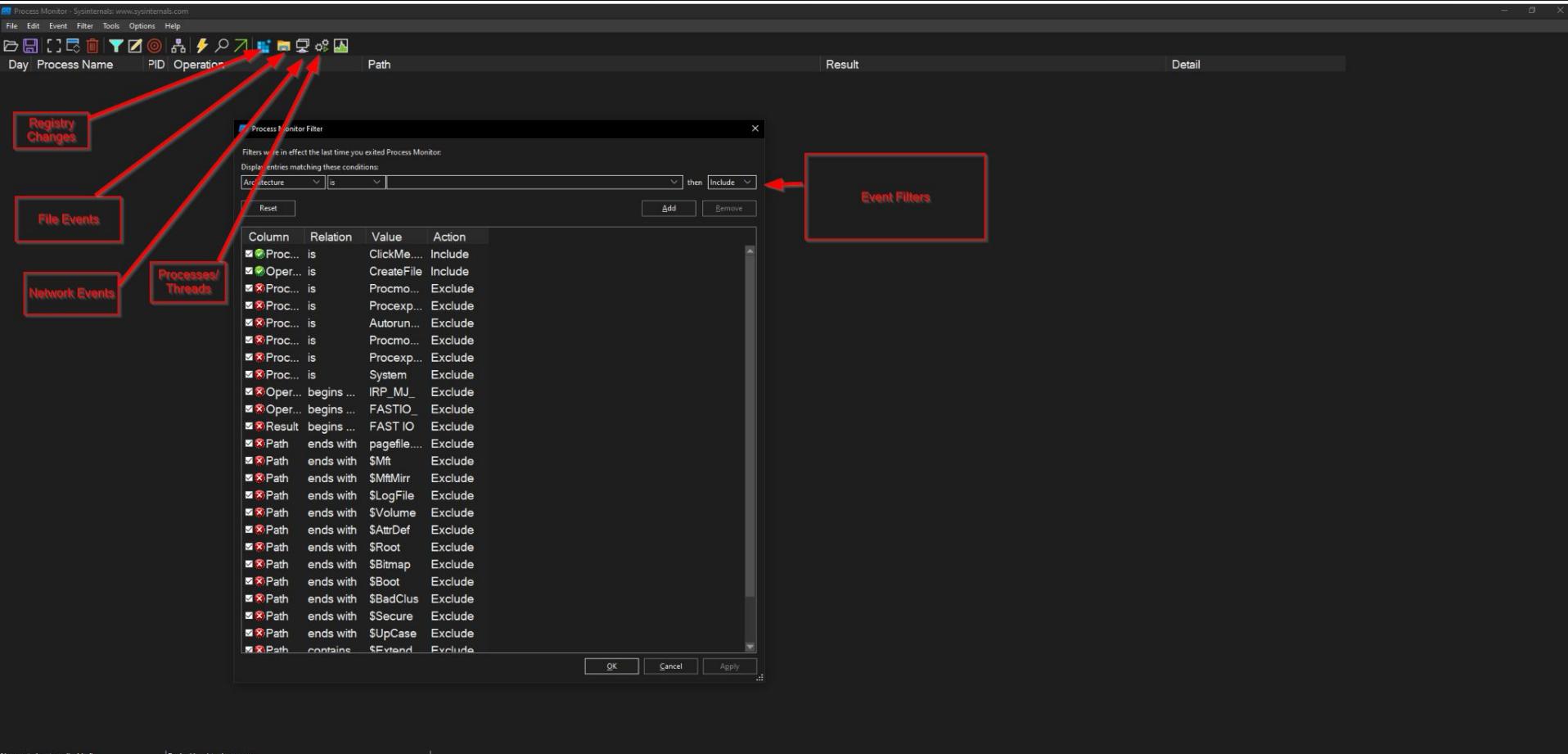
Demo:
Reading a file
C, C++, Modern c++

File IO/ Writing

Let's figure it out together!

Procmon: Looking at File operations

- Monitoring tool for Windows
- Displays shows real-time file system, Registry, network and process/thread activity.
- Also provides Image load information, meaning if a binary uses Explicit Dynamic Linking, we can see it load the DLL using only Procmon.
- Procmon is an incredibly powerful tool- so much so- that a lot of malware will look for the presence of Procmon running as an anti-debugging tactic

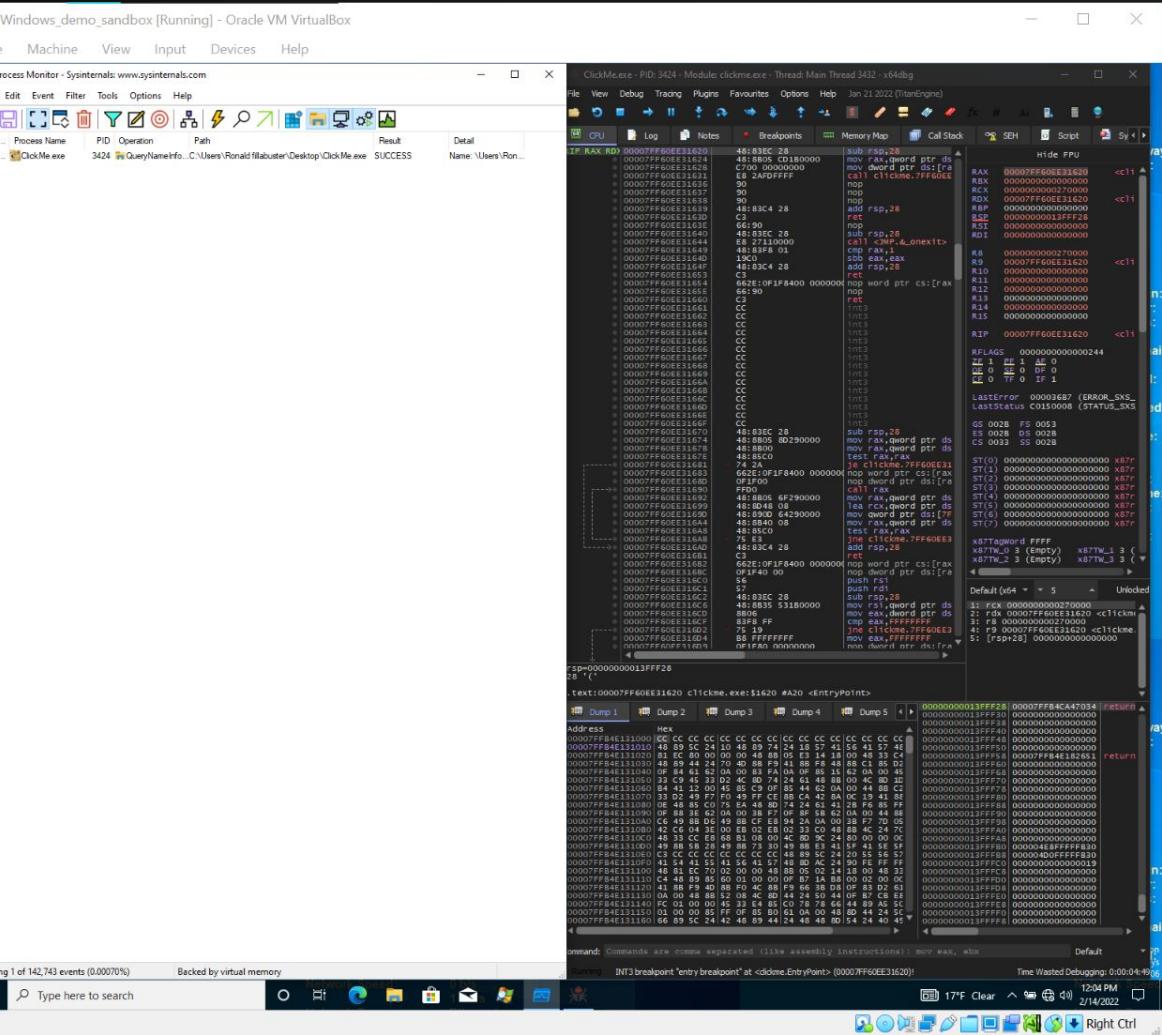
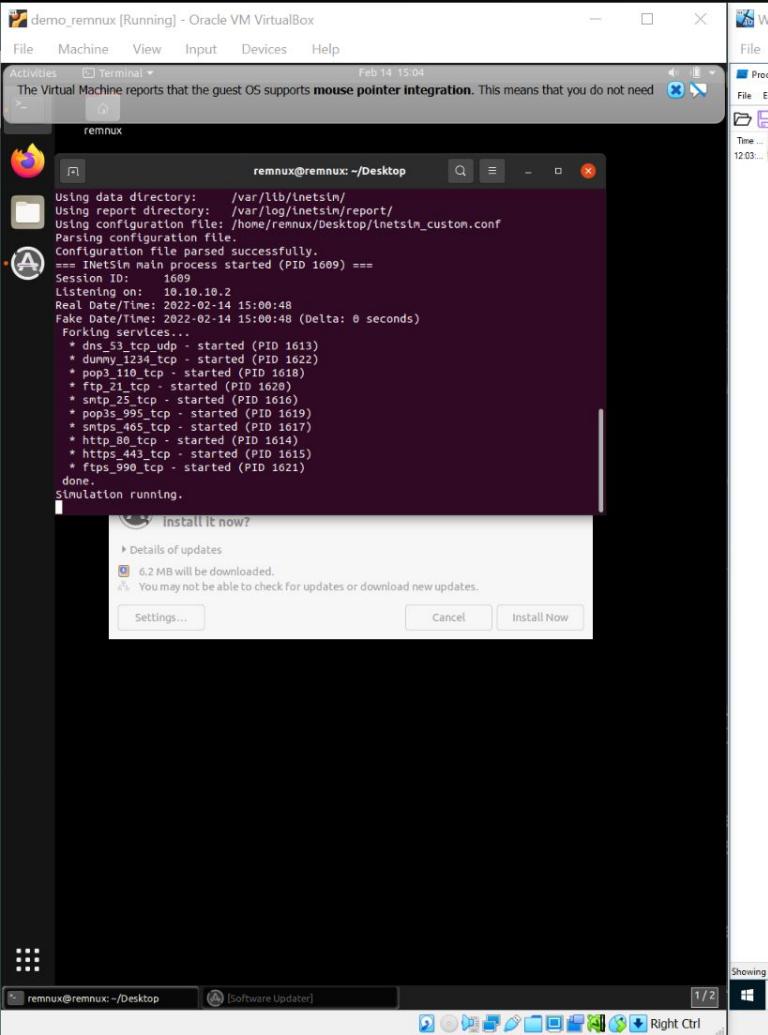


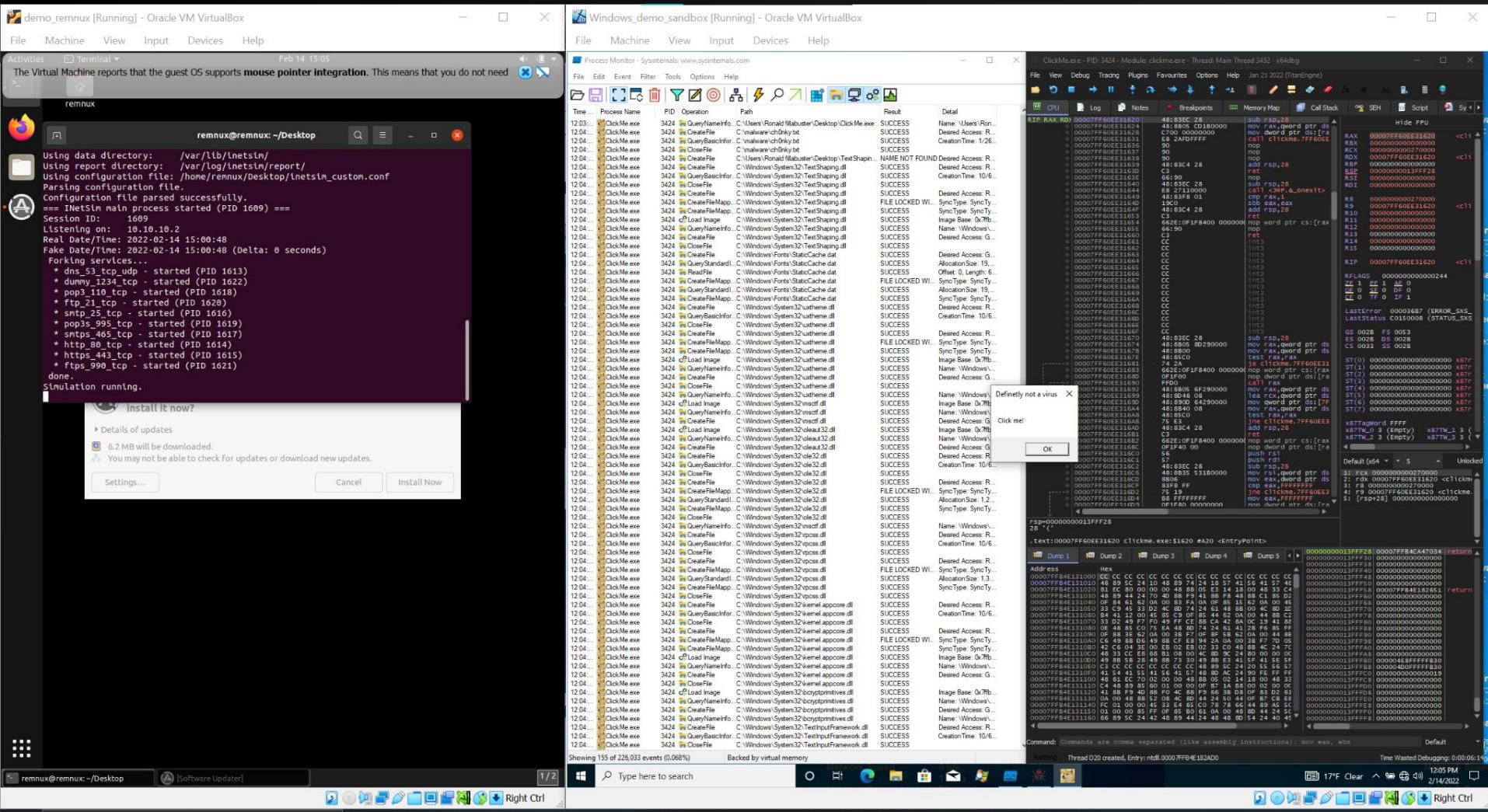
Demo

Using Procmon on ClickMe.exe

Filters for File IO

- Start Procmon and turn off live capture
- Start ClickMe.exe via x64dbg
- Set a filter for Proc name is ClickMe.exe
- Disable Registry, network and processes information
- Make sure File events are enabled
- BP sleep, set RCX=0
- Enable capture
- Run the binary
- Observe the calls to `CreateFile`
- Alternatively, `bp CreateFileA` in x64dbg





demo_renux [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Feb 14 15:06

The Virtual Machine reports that the guest OS supports **mouse pointer integration**. This means that you do not need to enable mouse support in the host's BIOS.

remnux

remnux@remnux: ~/Desktop

Using data directory: /var/lbt/lnetslm/
Using report directory: /var/log/lnetslm/report/
Using configuration file: /home/remnux/Desktop/lnetslm_custom.conf
Parsing configuration file.
Configuration file parsed successfully.
== INetStm main process started (PID 1609) ==
Session ID: 1609
Listening on: 10.10.10.2
Real Date/Time: 2022-02-14 15:00:48
Fake Date/Time: 2022-02-14 15:00:48 (Delta: 0 seconds)
Forking services...
* dns_53_tcp_udp - started (PID 1613)
* dummy_1234_tcp - started (PID 1622)
* pop3_110_tcp - started (PID 1618)
* ftp_21_tcp - started (PID 1620)
* smtp_25_tcp - started (PID 1616)
* pop3s_995_tcp - started (PID 1619)
* smtps_465_tcp - started (PID 1617)
* http_80_tcp - started (PID 1614)
* https_443_tcp - started (PID 1615)
* ftpts_990_tcp - started (PID 1621)
done.
Simulation running.

Install it now?

Details of updates

6.2 MB will be downloaded.

You may not be able to check for updates or download new updates.

Settings... Cancel Install Now

Windows_demo_sandbox [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

x64dbg

File New Debug Tracing Plugins Favours Options Help Jan 21, 2022 (TiteEngine)

Time ... Process Name PID Operation Path Result Detail

12:06... ClickMe.exe 3424 QueryDirectory C:\Users\Ronald filabuster\AppData\ SUCCESS FileInformationClass
12:06... ClickMe.exe 3424 CloseFile C:\Users\Ronald filabuster\AppData\ SUCCESS
12:06... ClickMe.exe 3424 CreateFile C:\Users\Ronald filabuster\AppData\Local\ SUCCESS Desired Access: R- FileInformationClass
12:06... ClickMe.exe 3424 QueryDirectory C:\Users\Ronald filabuster\AppData\Local\ SUCCESS
12:06... ClickMe.exe 3424 CloseFile C:\Users\Ronald filabuster\AppData\Local\ SUCCESS Desired Access: R- FileInformationClass
12:06... ClickMe.exe 3424 CreateFile C:\Users\Ronald filabuster\AppData\Local\Temp\TrustMeNotMalware.exe SUCCESS Desired Access: R- FileInformationClass
12:06... ClickMe.exe 3424 QueryDirectory C:\Users\RONALD-1\AppData\Local\Temp\ SUCCESS
12:06... ClickMe.exe 3424 CloseFile C:\Users\RONALD-1\AppData\Local\Temp\ SUCCESS
12:06... ClickMe.exe 3424 CreateFile C:\Users\RONALD-1\AppData\Local\Temp\ SUCCESS Desired Access: R- FileInformationClass
12:06... ClickMe.exe 3424 QueryDirectory C:\Windows\System32\apphelp.dll SUCCESS
12:06... ClickMe.exe 3424 CloseFile C:\Windows\System32\apphelp.dll SUCCESS
12:06... ClickMe.exe 3424 CreateFileMap C:\Windows\System32\apphelp.dll SUCCESS Desired Access: R- SyncType: SyncTy...
12:06... ClickMe.exe 3424 CreateFileMap C:\Windows\System32\apphelp.dll FILE LOCKED WI... SyncType: SyncTy...
12:06... ClickMe.exe 3424 Load Image C:\Windows\System32\apphelp.dll SUCCESS SyncType: SyncTy...
12:06... ClickMe.exe 3424 QueryNameInfo C:\Windows\System32\apphelp.dll SUCCESS Name: Windows... Desired Access: G-
12:06... ClickMe.exe 3424 CloseFile C:\Windows\System32\apphelp.dll SUCCESS
12:06... ClickMe.exe 3424 CreateFileMap C:\Windows\System32\apphelp.dll SUCCESS Desired Access: R- SyncType: SyncTy...
12:06... ClickMe.exe 3424 CreateFileMap C:\Windows\System32\apphelp.dll FILE LOCKED WI... SyncType: SyncTy...
12:06... ClickMe.exe 3424 QueryStandard C:\Windows\apppatch\yeyman.adb SUCCESS AllocationSize: 4.0...
12:06... ClickMe.exe 3424 CreateFileMap C:\Windows\apppatch\yeyman.adb FILE LOCKED WI... SyncType: SyncTy...
12:06... ClickMe.exe 3424 QueryStandard C:\Windows\apppatch\yeyman.adb SUCCESS AllocationSize: 4.0...
12:06... ClickMe.exe 3424 CreateFileMap C:\Windows\apppatch\yeyman.adb SUCCESS SyncType: SyncTy...
12:06... ClickMe.exe 3424 CloseFile C:\Windows\apppatch\yeyman.adb SUCCESS
12:06... ClickMe.exe 3424 CreateFileMap C:\Users\Ronald filabuster\AppData\Local\Temp\ SUCCESS Desired Access: G- SyncType: SyncTy...
12:06... ClickMe.exe 3424 CloseThread Exit Thread ID: 326... SUCCESS
12:06... ClickMe.exe 3424 CreateFileMap C:\Windows\apppatch\yeyman.adb SUCCESS Desired Access: G- SyncType: SyncTy...
12:06... ClickMe.exe 3424 CreateFileMap C:\Windows\apppatch\yeyman.adb SUCCESS AllocationSize: 4.0...
12:06... ClickMe.exe 3424 CreateFileMap C:\Windows\apppatch\yeyman.adb FILE LOCKED WI... SyncType: SyncTy...
12:06... ClickMe.exe 3424 CloseThread Exit Thread ID: 326... SUCCESS
12:06... ClickMe.exe 3424 CreateFileMap C:\Windows\apppatch\yeyman.adb SUCCESS Desired Access: G- SyncType: SyncTy...
12:06... ClickMe.exe 3424 CreateFileMap C:\Windows\apppatch\yeyman.adb SUCCESS AllocationSize: 4.0...
12:06... ClickMe.exe 3424 CloseThread Exit Thread ID: 7464... SUCCESS
12:06... ClickMe.exe 3424 CreateFileMap C:\Windows\apppatch\yeyman.adb SUCCESS Desired Access: G- SyncType: SyncTy...
12:06... ClickMe.exe 3424 CloseThread Exit Thread ID: 3432... SUCCESS
12:06... ClickMe.exe 3424 Process Exit Exit Status: 0, User: Default

Showing 642 of 304,652 events (0.21%) Backed by virtual memory

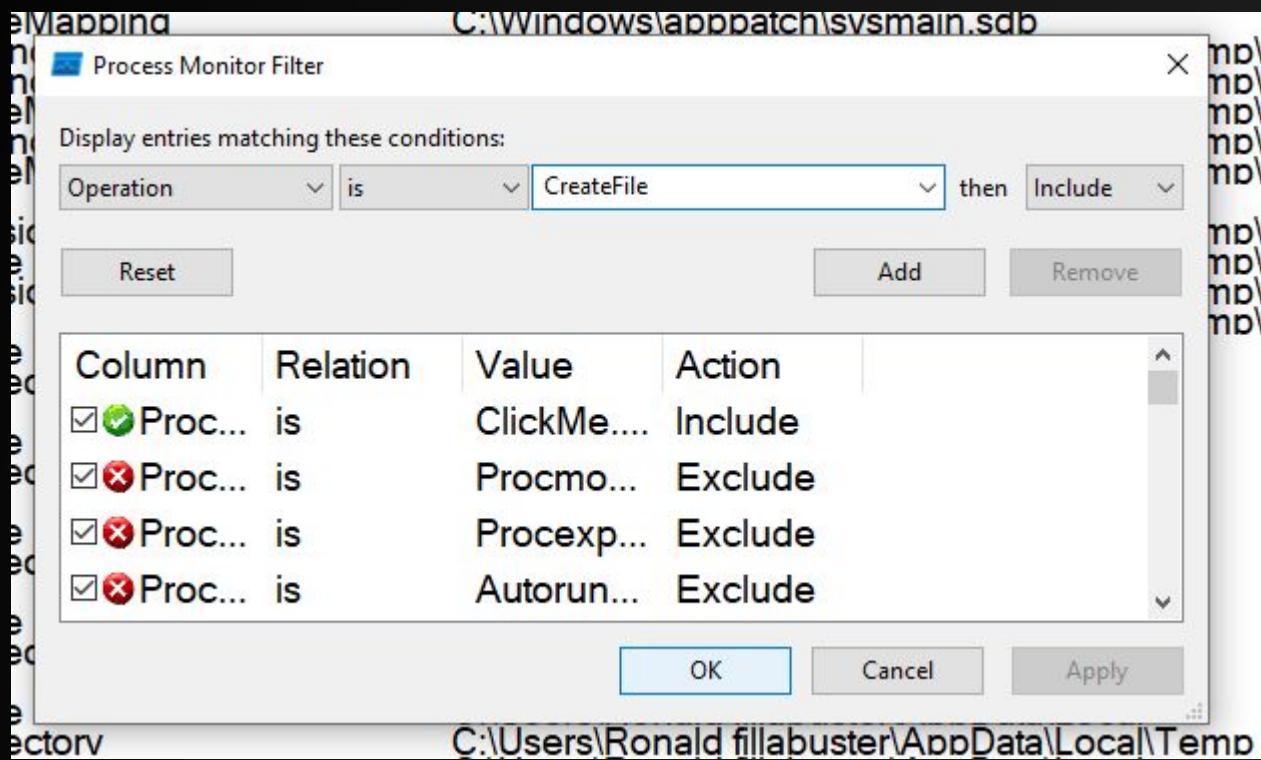
Type here to search

remnux@remnux: ~/Desktop

Software Updater

17% Clear 12:06 PM 2/14/2022 Right Ctrl

Filtering by Operation





Procmon File tips

- Procmon displays a LOT of information when no filters are applied
- Many organizations will install Sysmon- a system wide implementation of Procmon that will send all of the events to an analysis environment
- For us, it is best to run a processes via x64dbg, turn off capture, and set filters for the target processes, enable capture then run
- Once the processes is done/we are happy with the execution, we can search through the mountain of data created
- I usually start with all `CreateFile` calls, as they can give information about loaded DLLs, files opened/created