

# North Alabama ISSA Chapter Meeting

## Artifact-Driven Malware Analysis: A Digital Forensics Approach

21 April 2026



# Agenda

- Introduction
- History
- Digital Artifact Types
  - ✓ Processes
  - ✓ Registry
- Artifact Collection
- Demonstration
- Questions

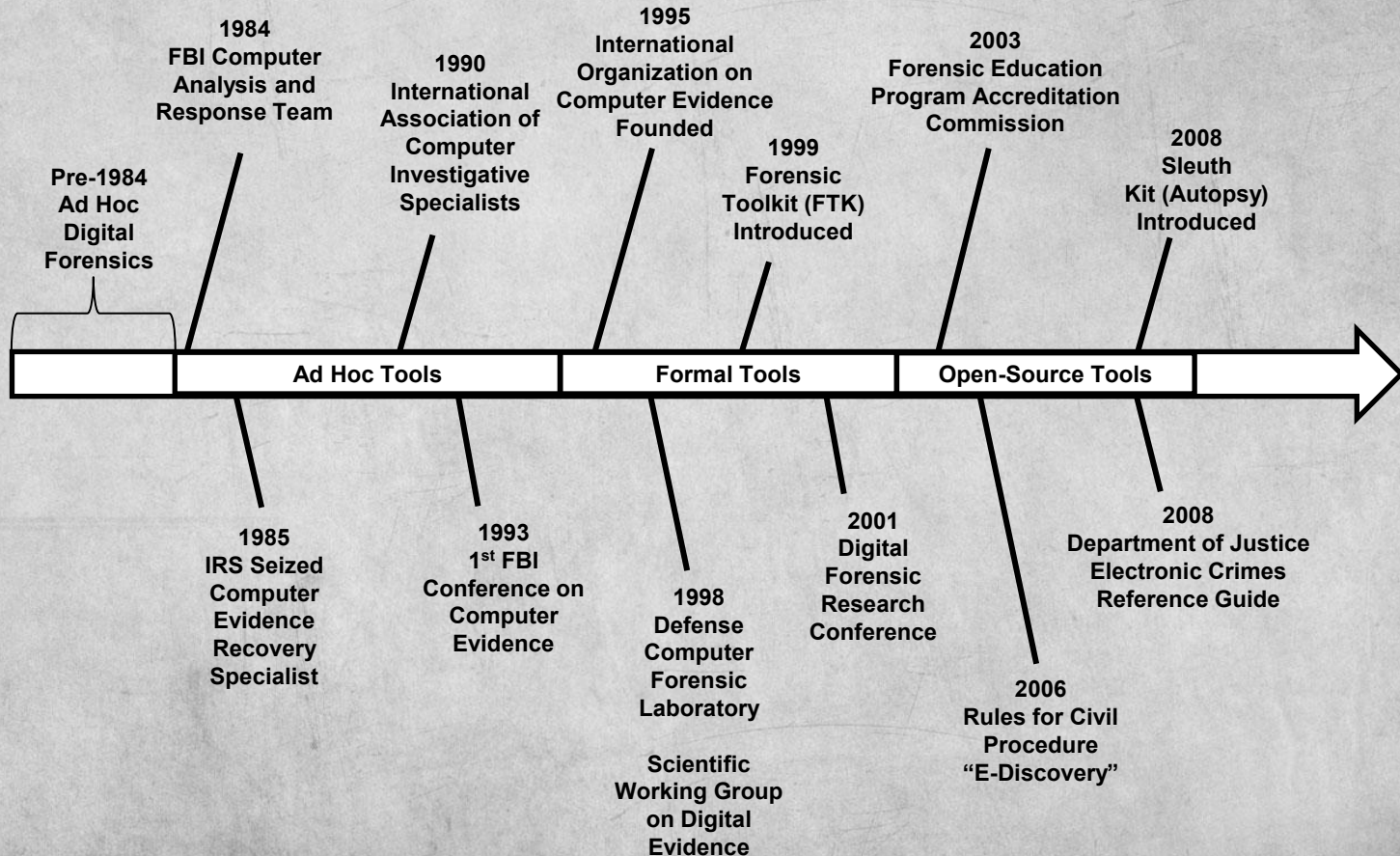
# Introduction

- Jason Cuneo
- West Point, 1998, B.S. Electrical Engineering
- U.S. Army, Infantry Officer, 1998 – 2004
- UAH, 2006, M.S. Electrical Engineering
- Auburn University, 2022, M.S. Computer Science and Software Engineering
- Experience
  - ✓ Defensive Cyber Operations
  - ✓ Digital Forensics
  - ✓ Risk Assessment
  - ✓ Cyber Training & Exercises
  - ✓ Advanced Networking
  - ✓ Vulnerability Mitigation
  - ✓ Blockchain



# **Digital Forensics History**

# Digital Forensics History



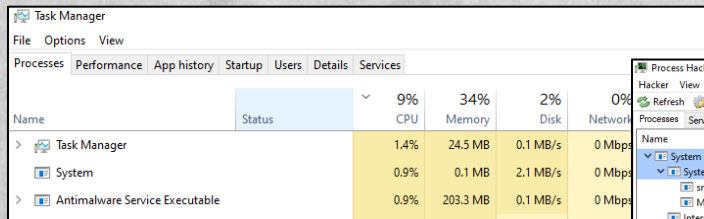
# Digital Forensics in Malware Analysis

- Digital forensics underpins malware reverse engineering, enabling analysts to identify, analyze, and interpret malicious activity in controlled environments
- Dynamic analysis executes malware in isolation to reveal behavior & functionality:
  - ✓ Controlled execution and behavioral observation
  - ✓ Process and system activity monitoring
  - ✓ OS-level interaction and code behavior analysis
- Comprehensive monitoring uncovers malware impact and intent, including:
  - ✓ Process activity
  - ✓ Network communications
  - ✓ Registry modifications and persistence mechanisms
  - ✓ File system changes and artifacts
- Understanding normal OS behavior is critical for identifying anomalies and reconstructing malicious activity

# **Process Definitions**

# Processes

- A process is an instance of a program currently being executed that contains all program code, data, and system resources needed to run
- Each process has its own memory space, file handles, and system resources and is managed by the operating system, which allocates resources, schedules execution, and provides inter-process communication
- Each process running in memory provide information on system activities and are critical to detecting malware infections
- When spawned, processes provide a wealth of information about their operation including process names and offsets, process IDs, parent process IDs, number of threads and handles, and process start and exit times

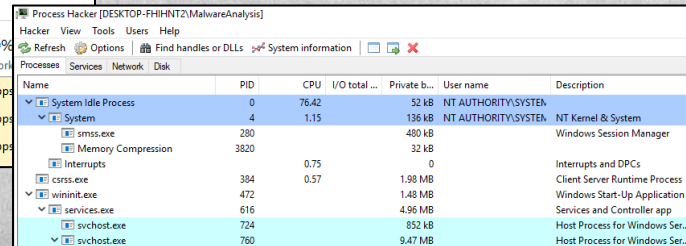


Task Manager

File Options View

Processes Performance App history Startup Users Details Services

Name	Status	CPU	Memory	Disk	Network
Task Manager		9%	34%	2%	0%
System		0.9%	0.1 MB	2.1 MB/s	0 Mbps
Antimalware Service Executable		0.9%	203.3 MB	0.1 MB/s	0 Mbps



Process Hacker [DESKTOP-FHH-HNT2\MalwareAnalysis]

Hacker View Tools Users Help

Refresh Options Find handles or DLLs System information

Processes Services Network Disk

Name	PID	CPU	I/O total ...	Private b...	User name	Description
System Idle Process	0	76.42		52 kB	NT AUTHORITY\SYSTEM	
System	4	1.15		136 kB	NT AUTHORITY\SYSTEM	NT Kernel & System
smss.exe	280			480 kB		Windows Session Manager
Memory Compression	3820			32 kB		
Interrupts		0.75		0		Interrupts and DPCs
csrss.exe	384	0.57		1.98 MB		Client Server Runtime Process
wininit.exe	472			1.48 MB		Windows Start-Up Application
services.exe	616			4.96 MB		Services and Controller app
svchost.exe	724			852 kB		Host Process for Windows Ser...
svchost.exe	760			9.47 MB		Host Process for Windows Ser...

# Process Definitions

- There are several process related definitions we need to understand before analyzing malware:
  - ✓ Process ID (PID)
    - Unique numerical identifier assigned to each running process by the OS
  - ✓ Parent Process
    - Process that spawns or creates another process
  - ✓ Child Process
    - Process created by a parent process
  - ✓ Process Tree
    - Hierarchical representation of processes showing parent-child relationships

# Windows Core Process Summary

Process Name	Parent Process	File Path	Singleton	Account	Start Time
SYSTEM	None	None	Yes	Local System	Boot
smss.exe	SYSTEM	C:\Windows\System32\smss.exe	No	Local System	Boot
wininit.exe	None	C:\Windows\System32\wininit.exe	Yes	Local System	Boot
taskhost.exe	services.exe	C:\Windows\System32\taskhost.exe	No	Many	Varies
lsass.exe	wininit.exe	C:\Windows\System32\lsass.exe	Yes	Local System	Boot
winlogon.exe	None	C:\Windows\System32\winlogon.exe	No	Local System	Varies
iexplore.exe	explorer.exe	C:\Program Files\Internet Explorer\iexplore.exe	No	Local Users	Varies
explorer.exe	userinit.exe	C:\Windows\explorer.exe	No	Local Users	Varies
lsm.exe	wininit.exe	C:\Windows\System32\lsm.exe	Yes	Local System	Boot
svchost.exe	services.exe	C:\Windows\System32\svchost.exe	No	Local System Network Service Local Service	Boot
services.exe	wininit.exe	C:\Windows\System32\services.exe	Yes	Local System	Boot
csrss.exe	None	C:\Windows\System32\csrss.exe	No	Local System	Boot

# Malware Process Attacks

# Process Attack Definitions

- Process Injection
  - ✓ Malware injects its code into legitimate processes to conceal its presence and bypass security measures
  - ✓ An example is where malware injects into running processes like explorer.exe or svchost.exe
- Process Hollowing
  - ✓ Malware creates a new process in a suspended state and replaces its legitimate code with its malicious payload
  - ✓ Upon process startup the malware runs while appearing as a legitimate process
- Process Privilege Escalation
  - ✓ Malware elevates privileges to manipulate processes and attempts to exploits vulnerabilities, weak configurations, or misconfigured access control

# Process Attack Definitions

- Process Termination
  - ✓ Malware terminates critical processes to disrupt system operation and disable security features
- Process Hooking
  - ✓ Malware hooks into system APIs or process functions to intercept and modify their behavior
  - ✓ Successful process hooking results in data manipulation, network traffic interception, and privilege escalation
- Anti-Analysis Techniques
  - ✓ Malware evades analysis and detection by active circumvention of analysis tools, virtual environments, and sandboxes

# **DLL Search Order**

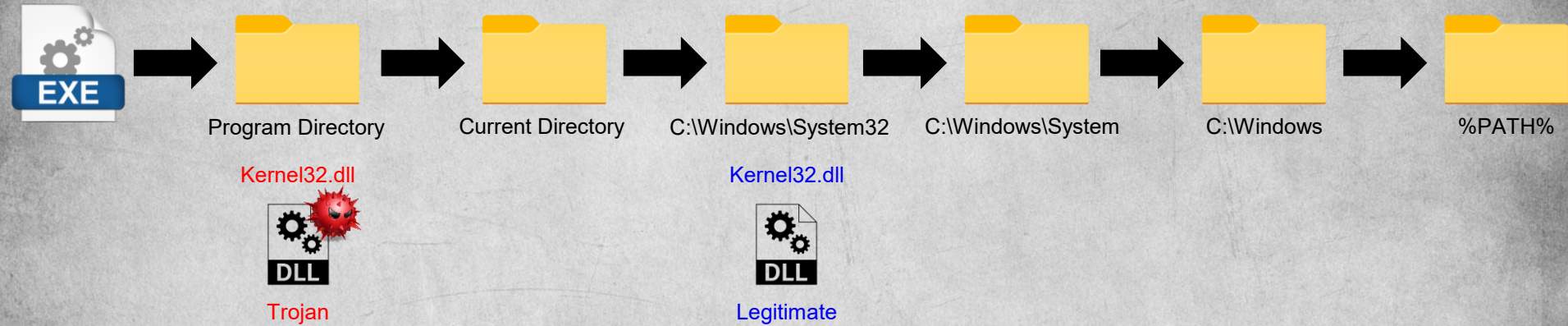
# DLL Search Order

- Windows executables will work only if supporting DLL's are available
- When an application requires a DLL to fulfill a function or utilize a specific feature, it follows a predefined search order to locate the required DLL file
- If the application finds a DLL with the same name in multiple directories during the search process, it uses the DLL that is found in the directory with the highest priority according to the search order

Kernel32.dll

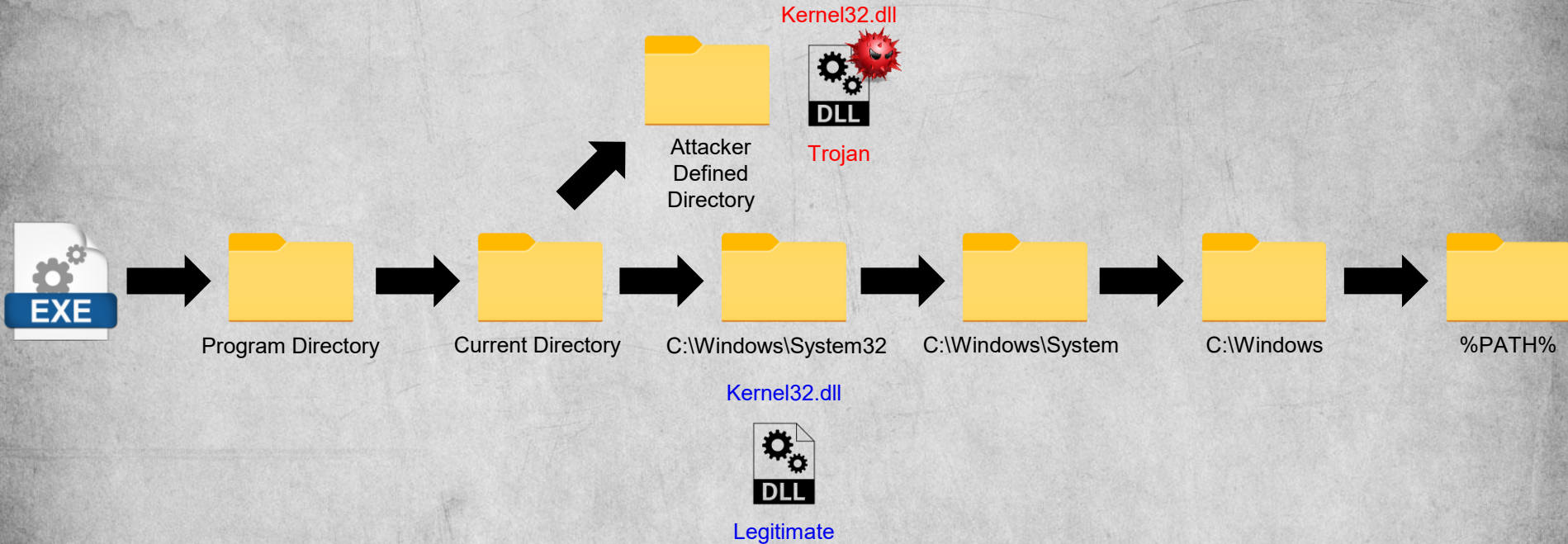


# Insecure DLL Loading



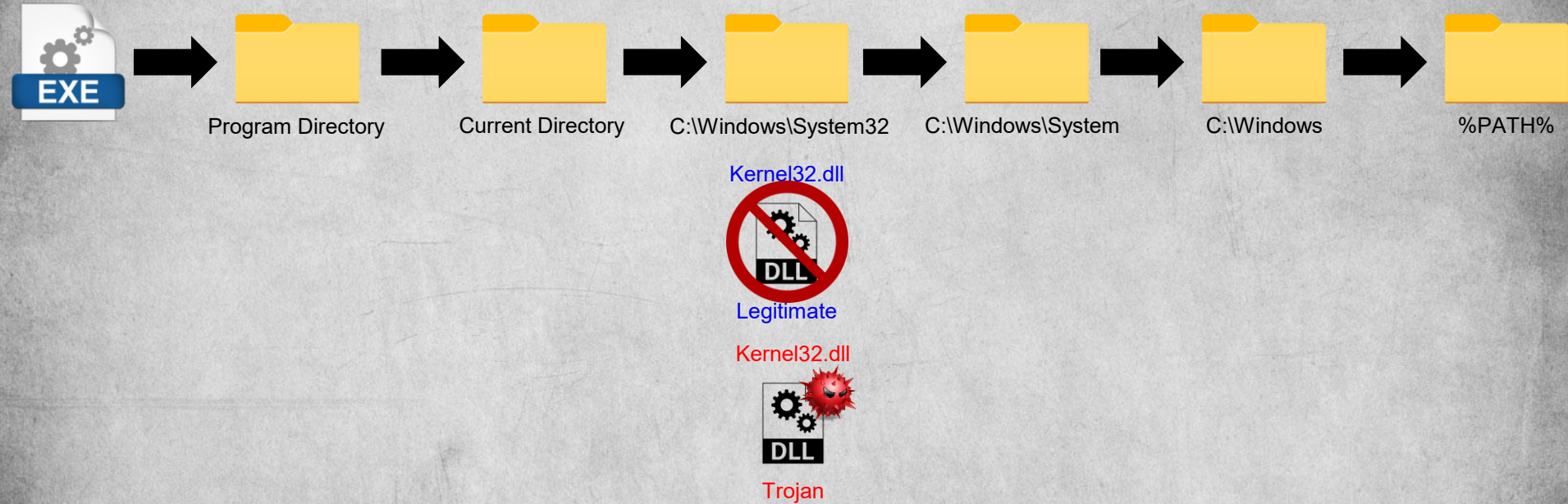
- Assume an executable needs Kernel32.dll to execute its functionality correctly
- In this scenario, the malware author plants a malicious DLL with the same name as a legitimate DLL and places it in a directory that the operating system searches before the legitimate DLL location

# DLL Search Path Manipulation



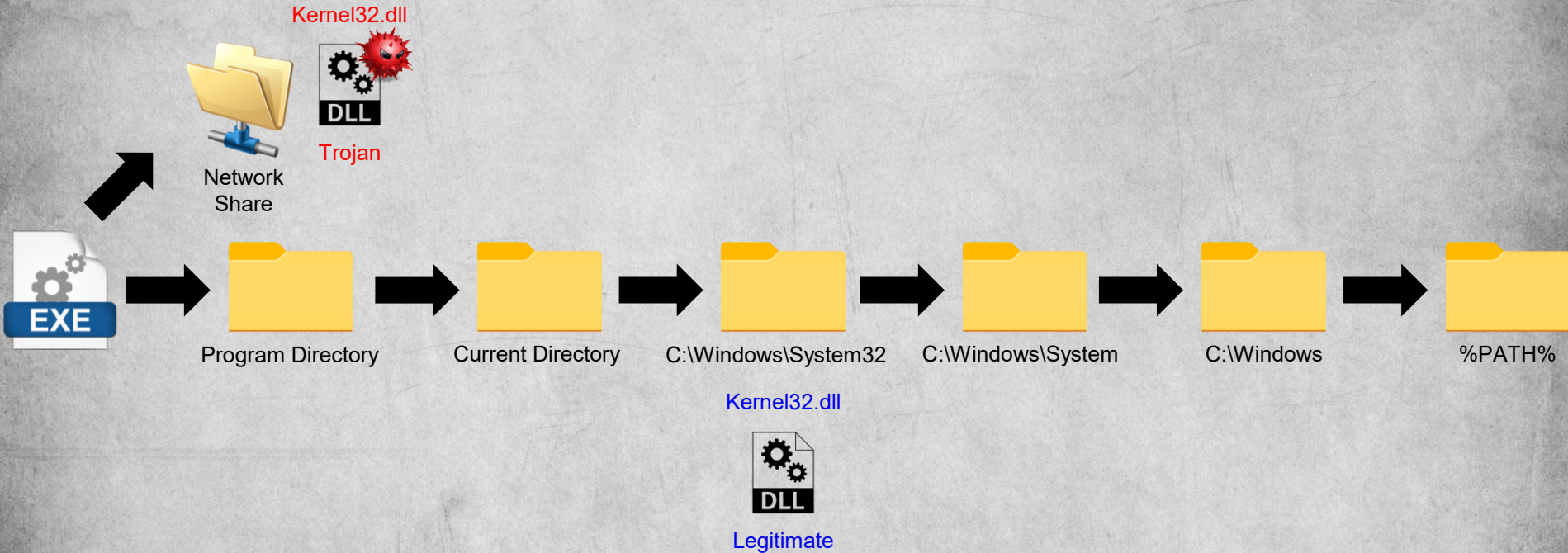
- In this scenario, the malware author manipulates the search path by altering environment variables which forces an application to load a malicious DLL from a directory under the attacker's control

# Weak File Permissions



- In this scenario, the malware author finds weak file permissions and replaces a legitimate DLL with a malicious DLL in a directory that the application searches during DLL loading

# DLL Side Loading



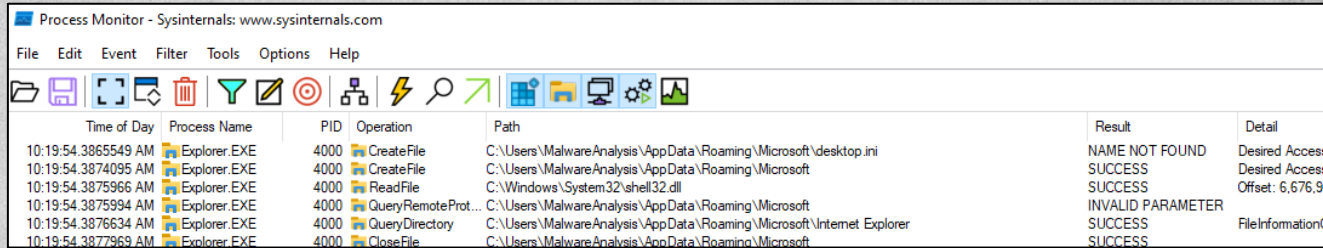
- In this scenario, the malware author uses external locations such as network shares or removable storage to make applications load external DLL's

# Process Artifacts

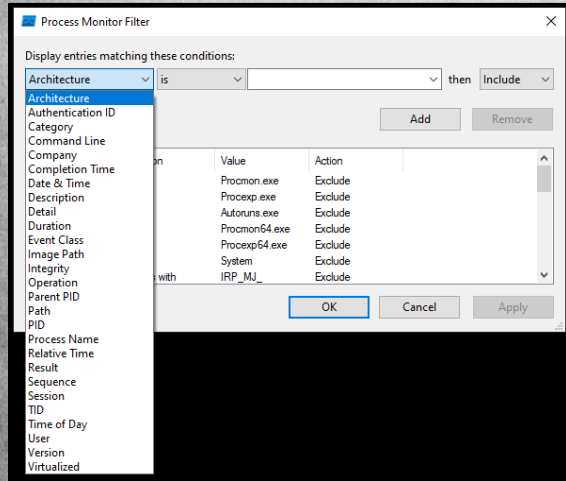
# Process Monitor

- Defined by Microsoft as “an advanced monitoring tool for Windows that shows real-time file system, registry, process, and thread activity”
- Combines functionality of two legacy Sysinternals utilities, Filemon and Regmon, and extends them by including:
  - ✓ Rich and non-destructive filtering
  - ✓ Comprehensive event properties including:
    - Session IDs
    - Usernames
    - Processes
    - Full thread stacks
    - Simultaneous logging to a file

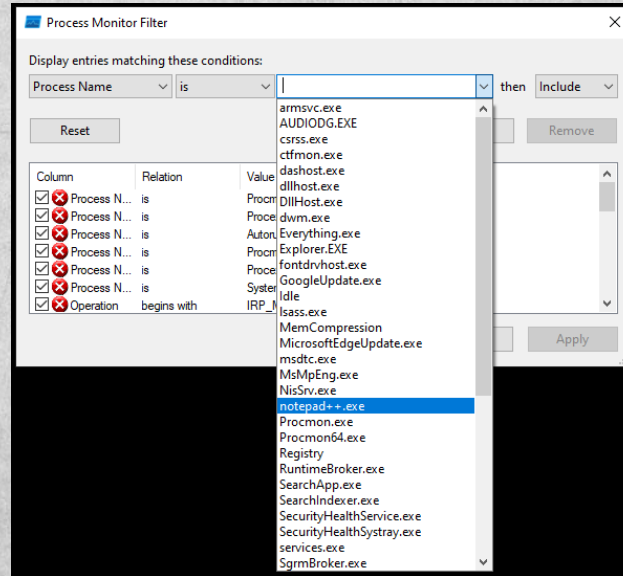
# Process Monitor Walkthrough



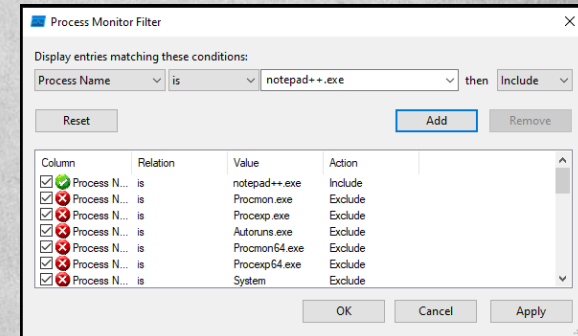
Time of Day	Process Name	PID	Operation	Path	Result	Detail
10:19:54.3865549 AM	Explorer.EXE	4000	CreateFile	C:\Users\MalwareAnalysis\AppData\Roaming\Microsoft\desktop.ini	NAME NOT FOUND	Desired Access:
10:19:54.3874095 AM	Explorer.EXE	4000	CreateFile	C:\Users\MalwareAnalysis\AppData\Roaming\Microsoft	SUCCESS	Desired Access:
10:19:54.3875966 AM	Explorer.EXE	4000	ReadFile	C:\Windows\System32\shell32.dll	SUCCESS	Offset: 6,676,99
10:19:54.3875994 AM	Explorer.EXE	4000	QueryRemoteProt...	C:\Users\MalwareAnalysis\AppData\Roaming\Microsoft	INVALID PARAMETER	
10:19:54.3876634 AM	Explorer.EXE	4000	QueryDirectory	C:\Users\MalwareAnalysis\AppData\Roaming\Microsoft\Internet Explorer	SUCCESS	FileInformationCl
10:19:54.3877969 AM	Explorer.EXE	4000	CloseFile	C:\Users\MalwareAnalysis\AppData\Roaming\Microsoft	SUCCESS	



Select a Filter Type



Select Filter Criteria



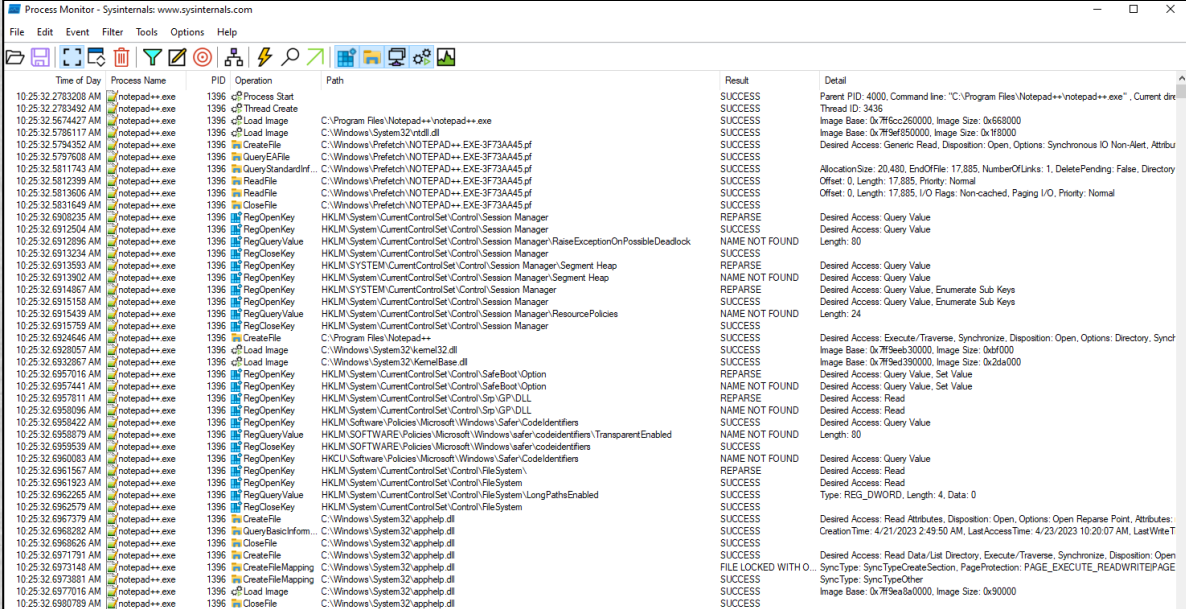
Add the Filter

# Process Monitor Filtering

- Once filtered, we will be able to see every activity that occurred during the startup and execution of the process

- Note the different operations

- ✓ Process Start
- ✓ Thread Create
- ✓ Load Image
- ✓ RegOpenKey
- ✓ RegQueryValue
- ✓ RegCloseKey
- ✓ File Creation
- ✓ QueryBasicInformation



Process Monitor - Sysinternals: www.sysinternals.com

Time of Day	Process Name	PID	Operation	Path	Result	Detail
10:25:32.283208 AM	notepad++ .exe	1396	Process Start		SUCCESS	Parent PID: 4000, Command Line: "C:\Program Files\notepad++\notepad++.exe", Current dir: Thread ID: 3436
10:25:32.2783492 AM	notepad++ .exe	1396	Thread Create		SUCCESS	
10:25:32.5674427 AM	notepad++ .exe	1396	Load Image	C:\Program Files\notepad++\notepad++.exe	SUCCESS	
10:25:32.578117 AM	notepad++ .exe	1396	Load Image	C:\Windows\System32\ntldr.dll	SUCCESS	
10:25:32.5794352 AM	notepad++ .exe	1396	CreateFile	C:\Windows\Prefetch\NOTEPAD++_EXE-3F73AA45.pf	SUCCESS	
10:25:32.5797608 AM	notepad++ .exe	1396	QueryEaFile	C:\Windows\Prefetch\NOTEPAD++_EXE-3F73AA45.pf	SUCCESS	
10:25:32.5811743 AM	notepad++ .exe	1396	QueryStandardInf...	C:\Windows\Prefetch\NOTEPAD++_EXE-3F73AA45.pf	SUCCESS	
10:25:32.5812399 AM	notepad++ .exe	1396	ReadFile	C:\Windows\Prefetch\NOTEPAD++_EXE-3F73AA45.pf	SUCCESS	
10:25:32.5913006 AM	notepad++ .exe	1396	ReadFile	C:\Windows\Prefetch\NOTEPAD++_EXE-3F73AA45.pf	SUCCESS	
10:25:32.5831649 AM	notepad++ .exe	1396	CloseFile	C:\Windows\Prefetch\NOTEPAD++_EXE-3F73AA45.pf	SUCCESS	
10:25:32.6908235 AM	notepad++ .exe	1396	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	REPARSE	Desired Access: Query Value
10:25:32.6912504 AM	notepad++ .exe	1396	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	Desired Access: Query Value
10:25:32.6912096 AM	notepad++ .exe	1396	RegQueryValue	HKLM\System\CurrentControlSet\Control\Session Manager\RaiseExceptionOnPossibleDeadlock	NAME NOT FOUND	Length: 80
10:25:32.6913234 AM	notepad++ .exe	1396	RegCloseKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	
10:25:32.6913593 AM	notepad++ .exe	1396	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Segment Heap	REPARSE	Desired Access: Query Value
10:25:32.6913902 AM	notepad++ .exe	1396	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager\Segment Heap	NAME NOT FOUND	Desired Access: Query Value
10:25:32.6914657 AM	notepad++ .exe	1396	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager	REPARSE	Desired Access: Query Value, Enumerate Sub Keys
10:25:32.6915159 AM	notepad++ .exe	1396	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	Desired Access: Query Value, Enumerate Sub Keys
10:25:32.6915439 AM	notepad++ .exe	1396	RegQueryValue	HKLM\System\CurrentControlSet\Control\Session Manager\ResourcePolicies	NAME NOT FOUND	Length: 24
10:25:32.6915759 AM	notepad++ .exe	1396	RegCloseKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	
10:25:32.6920446 AM	notepad++ .exe	1396	CreateFile	C:\Program Files\notepad++	SUCCESS	Desired Access: Execute/Traverse, Synchronize, Disposition: Open, Options: Directory, SyncI
10:25:32.6920957 AM	notepad++ .exe	1396	Load Image	C:\Windows\System32\kernel32.dll	SUCCESS	Image Base: 0x7f9e33000, Image Size: 0x2d000
10:25:32.6932867 AM	notepad++ .exe	1396	Load Image	C:\Windows\System32\KernelBase.dll	SUCCESS	Image Base: 0x7f9e33000, Image Size: 0x2da000
10:25:32.6957016 AM	notepad++ .exe	1396	RegOpenKey	HKLM\System\CurrentControlSet\Control\SafeBoot\Option	REPARSE	Desired Access: Query Value, Set Value
10:25:32.6957441 AM	notepad++ .exe	1396	RegOpenKey	HKLM\System\CurrentControlSet\Control\SafeBoot\Option	NAME NOT FOUND	Desired Access: Query Value, Set Value
10:25:32.6957911 AM	notepad++ .exe	1396	RegOpenKey	HKLM\System\CurrentControlSet\Control\GP\DLL	REPARSE	Desired Access: Read
10:25:32.6958096 AM	notepad++ .exe	1396	RegOpenKey	HKLM\System\CurrentControlSet\Control\Sec\GP\DLL	NAME NOT FOUND	Desired Access: Read
10:25:32.6958422 AM	notepad++ .exe	1396	RegOpenKey	HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers	SUCCESS	Desired Access: Query Value
10:25:32.695879 AM	notepad++ .exe	1396	RegQueryValue	HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers\TransparentEnabled	NAME NOT FOUND	Length: 80
10:25:32.6959039 AM	notepad++ .exe	1396	RegOpenKey	HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers	REPARSE	Desired Access: Read
10:25:32.6960083 AM	notepad++ .exe	1396	RegOpenKey	HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers	NAME NOT FOUND	Desired Access: Query Value
10:25:32.6961567 AM	notepad++ .exe	1396	RegOpenKey	HKLM\System\CurrentControlSet\Control\FileSystem\	REPARSE	Desired Access: Read
10:25:32.6961923 AM	notepad++ .exe	1396	RegOpenKey	HKLM\System\CurrentControlSet\Control\FileSystem	SUCCESS	Desired Access: Read
10:25:32.6962553 AM	notepad++ .exe	1396	RegQueryValue	HKLM\System\CurrentControlSet\Control\FileSystem\LongPathsEnabled	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
10:25:32.6962579 AM	notepad++ .exe	1396	RegCloseKey	HKLM\System\CurrentControlSet\Control\FileSystem	SUCCESS	
10:25:32.6967379 AM	notepad++ .exe	1396	CreateFile	C:\Windows\System32\apphelp.dll	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: CreationTime: 4/21/2023 2:49:50 AM, LastAccessTime: 4/23/2023 10:20:07 AM, LastWriteT
10:25:32.6968282 AM	notepad++ .exe	1396	QueryBasicInfor...	C:\Windows\System32\apphelp.dll	SUCCESS	
10:25:32.6968626 AM	notepad++ .exe	1396	CreateFile	C:\Windows\System32\apphelp.dll	SUCCESS	
10:25:32.6971791 AM	notepad++ .exe	1396	CreateFile	C:\Windows\System32\apphelp.dll	SUCCESS	Desired Access: Read Data/List Directory, Execute/Traverse, Synchronize, Disposition: Open
10:25:32.6973148 AM	notepad++ .exe	1396	CreateFileMapping	C:\Windows\System32\apphelp.dll	FILE LOCKED WITH O...	SyncType: SyncTypeCreateSection, PageProtection: PAGE_EXECUTE_READWRITEPAGE
10:25:32.6973881 AM	notepad++ .exe	1396	CreateFileMapping	C:\Windows\System32\apphelp.dll	SUCCESS	SyncType: SyncTypeOther
10:25:32.6977016 AM	notepad++ .exe	1396	Load Image	C:\Windows\System32\apphelp.dll	SUCCESS	Image Base: 0x7f9e3b000, Image Size: 0x90000
10:25:32.6980789 AM	notepad++ .exe	1396	CloseFile	C:\Windows\System32\apphelp.dll	SUCCESS	

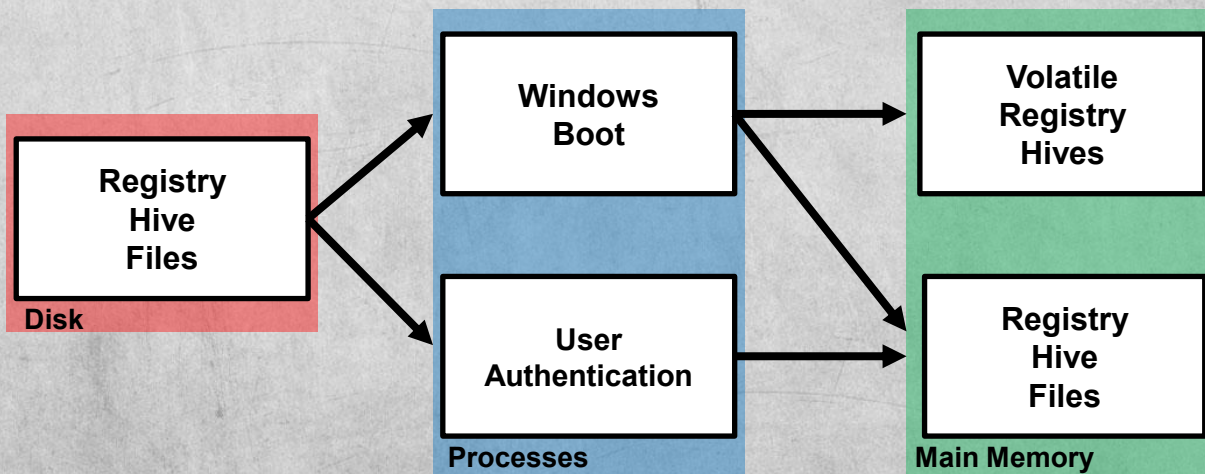
# Registry Artifacts

# Registry Monitoring

- The Windows operating system uses a centralized hierarchal database to store and manage configuration settings, preferences, and system-related information
- Windows does this for several reasons:
  - ✓ Efficient Access and Retrieval
  - ✓ Centralized Configuration
  - ✓ System Integration
  - ✓ User-specific Settings
  - ✓ Security and Access Control
  - ✓ Dynamic Configuration Changes
  - ✓ Compatibility and Legacy Support
- Malware abuses the Windows Registry and it is critical to understand how to collect and analyze artifacts produced by the Registry

# Windows Registry Operation

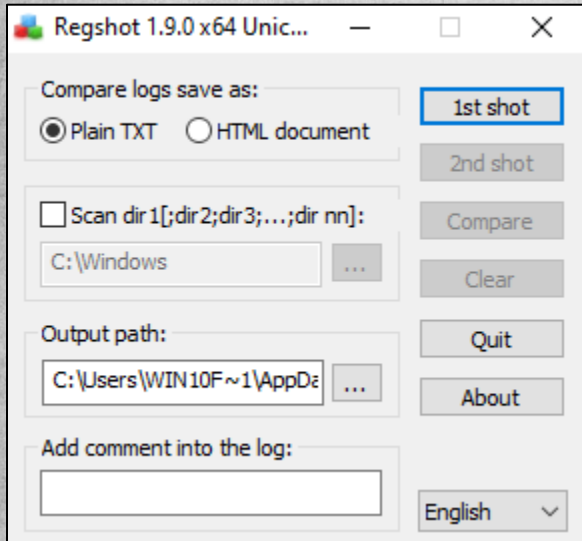
- A complete copy of the registry can only be found in main memory since it requires registry entries and active processes that create it
- From a forensics standpoint, this is significant because gathering a complete registry requires live memory capture



# RegShot

- Open-source utility for Windows that detects changes to Windows Registry before and after malware events
- An initial snapshot of the registry is compared against the registry after a malware event occurs and provides information about added, modified, or deleted registry keys, values, files, and folders
- RegShot provides a high-level final report for inclusion in malware analysis cases

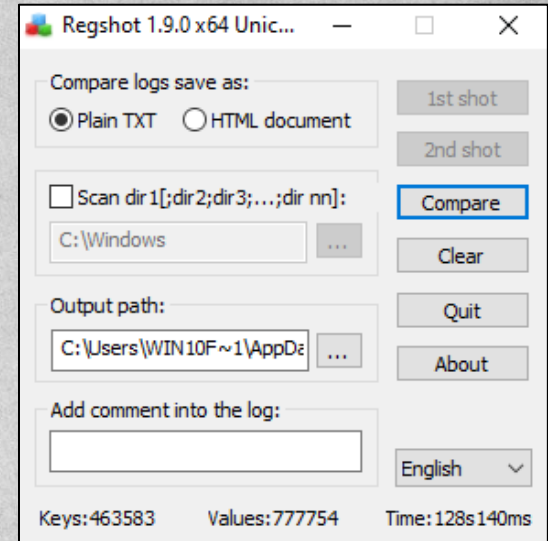
# Regshot



Run Before Malware Execution



After Execution Provide  
Sufficient Time Before  
Taking A Second Snapshot



Compare the Results

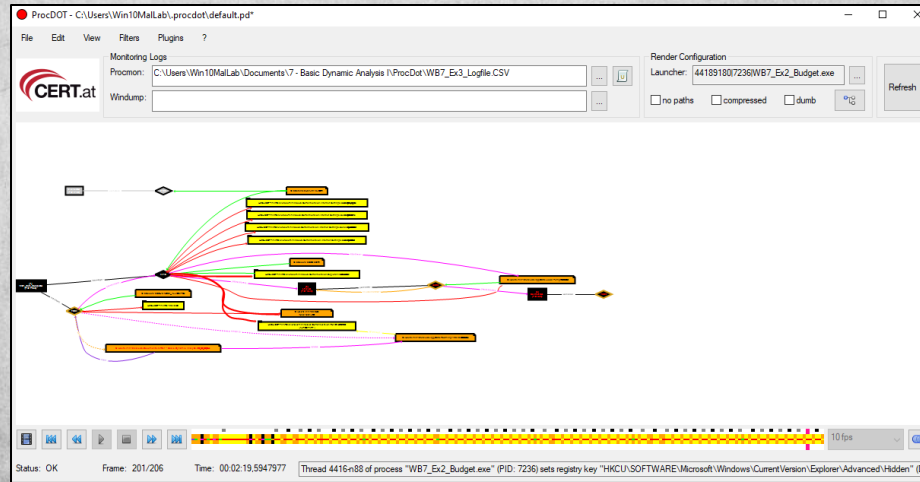
# **Timeline Analysis**

# ProcDot

- Created as an internal project at the Austrian CERT to support malware analysis workflows
- Designed and maintained by Christian Wojner, a malware analyst and reverse engineer at CERT.at.
- As the tool grew in complexity and features, it transitioned from a CERT.at project to a private project managed by Wojner, with its own dedicated site at [procdot.com](http://procdot.com)
- First public releases were made available through CERT.at's software section, later moved to [procdot.com](http://procdot.com) for downloads, documentation, and community support
- Still actively maintained, distributed as donationware, and widely used in malware research and education











# ProcDot Timelines

- Once ProcDot collects processes, network interfaces, and other artifacts, it then uses GraphViz to display events and the time in which they were generated



# ProcDot Events

- As events are added to the ProcDot graph, they are colorized to represent different activities
- This helps to show file reads and writes and identify process ownership

	<b>Read/Get/Receive.</b> Thicker lines indicate the number of log-records.		<b>Write/Set/Send.</b> Thicker lines indicate the number of log-records.
	<b>Create/Rename.</b> A process, a thread, or a file. Renames also come up with dashed initiator edges.		<b>Injection.</b> A thread is created in some other process.
	<b>Ownership.</b> Between process/thread. Created during monitoring.		<b>Ownership.</b> Between process/thread. Created before monitoring.
	<b>RegValue.</b> A file/URL that is set as Registry key value.		<b>Main Module.</b> For a specific process.
	<b>Delete/Kill.</b> A file/process.		<b>Additional Module.</b> For a specific thread.

# Event Step



- Once a timeline is created, ProcDot generates a playable video that shows each event and highlights new events
- This can be helpful when looking for initial infections and follow-on system impacts

# Demonstration

**Questions?**