MONTANA STATE UNIVERSITY

11/20/2025

Malicious Code Analysis

Fangtian Zhong
CSCI 591

Gianforte School of Computing
Norm Asbjornson College of Engineering
E-mail: fangtian.zhong@montana.edu

Part One

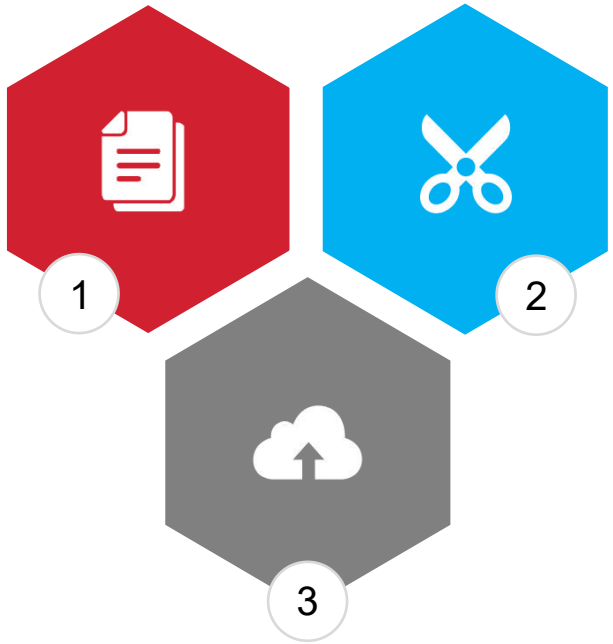
01

Dynamic Malware Analysis

2025-10-19

What is Dynamic Analysis?

- Allow the malware to run.



- Complement static analysis.
- Help us to understand the program behavior.

Dynamic Analysis

Basic dynamic analysis

- Executing target binary and observing its behavior.

Advanced dynamic analysis

- Executing target binary and using a debugger to analyze internal states.

Basic Dynamic Analysis

- Monitered execution of a program in order to perform analysis.
- Often performed after static analysis is done.

Advantages

- Efficient way to determine program functionality.
- Able to check file activity, process creation, network activity, etc.

Disadvantages

- Non-functional paths may be explored.

Executing the Malware

- In most cases, you can just double-click the “exe” file.
  - You may want to run it from the command-line as well.
- What if the extension is not “exe”?
  - You can change it. Verify if it is a PE file using a PE parser.
- What if it is a DLL?
  - You can run a DLL using the rundll32 program.
  - Format: C:\> rundll32.exe <name>.dll

Dynamic Analysis Systems

- Dynamic analysis is run in a safe environment on dedicated physical or virtual machines (in order not to expose the users' system to unnecessary risks)
  - ***Physical machines*** are set up on isolated networks, disconnected from the Internet or any other network, to prevent malware from spreading
  - ***Virtual machines*** emulate the functionality of a physical computer, where the OS running on the virtual machine is isolated from the host OS
    - *One limitation is that some malware can detect when they are running in a virtual machine, and they will execute differently than when in a physical machine*
  - A related term is ***sandbox***, referring to a physical or virtual environment for running malware, which isolates executables from other system resources and applications.
    - *Although they share characteristics with physical and virtual machines, sandboxes can be more limited (e.g., they can run in the browser), while physical and virtual machines always act as a complete system*
    - *For example, online sandboxes are websites where one can submit a sample file and receive a report about its behavior*

Dynamic Features for Malware Classification

Dynamic features are extracted from the execution of malware at runtime

- **Memory and registers usage -** values stored in the memory and different registers during the execution can distinguish benign from malicious programs
  - Ghiasi et al. (2015) monitored the memory content and register values before and after each invoked API call
  - They used similarity scores between the benign and malicious files in a training set to train an ML model for malware detection
- **Instruction traces -** sequence of processor instructions called during the execution of a program
  - Dynamic instruction traces are more robust indicators of the program's behavior than static traces, since compression and encryption can obfuscate code instructions from static analysis
  - Carlin et al. (2017) analyzed traces of opcodes to detect malware by Random Forest and Hidden Markov Model classifiers

Dynamic Features for Malware Classification

- **Network traffic -** monitoring the traffic entering and exiting the network can provide helpful information to detect malicious behavior
  - E.g., when malware infects a host machine, it may establish communication with an external server to download updates, other malware, or leak private and sensitive information from the host machine
  - Bekerman et al. (2015) extracted 972 features from the network traffic, and used them for developing Decision Tree and Random Forest malware classifiers
- **API call traces -** traces for accessing file systems, devices, processes, threads and error handling, and also to access functions such as the Windows registry, manage user accounts, etc.
  - Uppal et al. (2014) proposed traditional ML-based classifiers using n-grams of features extracted from traces of invoked API calls.

Sandboxing

- Sandboxing is a security technique used to isolate running applications from the rest of the system. It works by creating a virtual environment, or sandbox, in which an application or process can run without affecting other parts of the system.
- The sandboxed environment provides a controlled and secure environment for testing or running potentially risky applications. If the application performs any suspicious behavior, it will be confined within the sandbox and prevented from affecting other parts of the system.

Sandboxing

- Sandboxing can be implemented in various ways, such as virtual machines, containerization, or operating system-level sandboxes. For example, a virtual machine can be used to create a separate operating system environment within the host operating system, while containerization can be used to create isolated environments for individual applications or processes.

Sandboxes

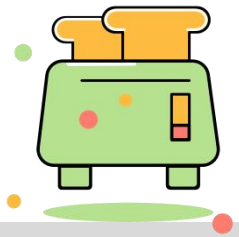
- All-in-one software solutions to analyze the execution of a program.
- Provides security mechanisms for running untrusted programs in a safe environment.
- Lets you moniter behavior/changes to the system.
- The “real” system remains isolated-so, it does not get infected.

Sandboxes

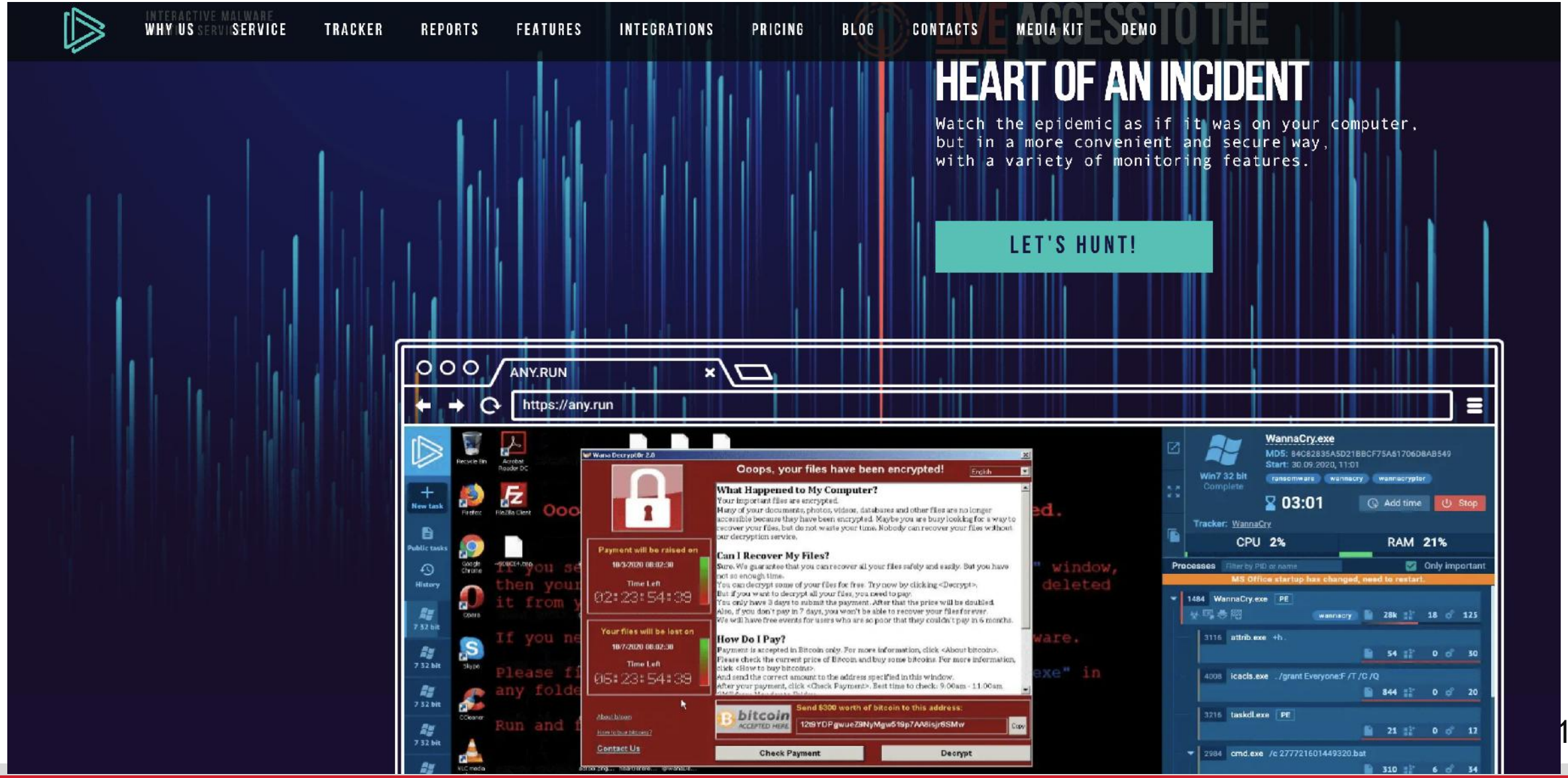
- Usually use virtual components
  - Simulates network services to allow program to execute as it “normally” would
- Sandboxes for malware analysis
  - There are many free and commercial versions available
  - Lets you analyze a variety of file types: EXE, PDF, Office documents, URLs, etc.

Sandboxes-drawbacks

- May run the EXE w/o command line arguments.
- Execution may wait for response from C2.
- Malware may find out that it is running in a sandbox .
  - an anti-analysis technique
  - in that case, the malware may change its behavior
- Often the environment is not properly setup.



Sandbox-Example:https://any.run/

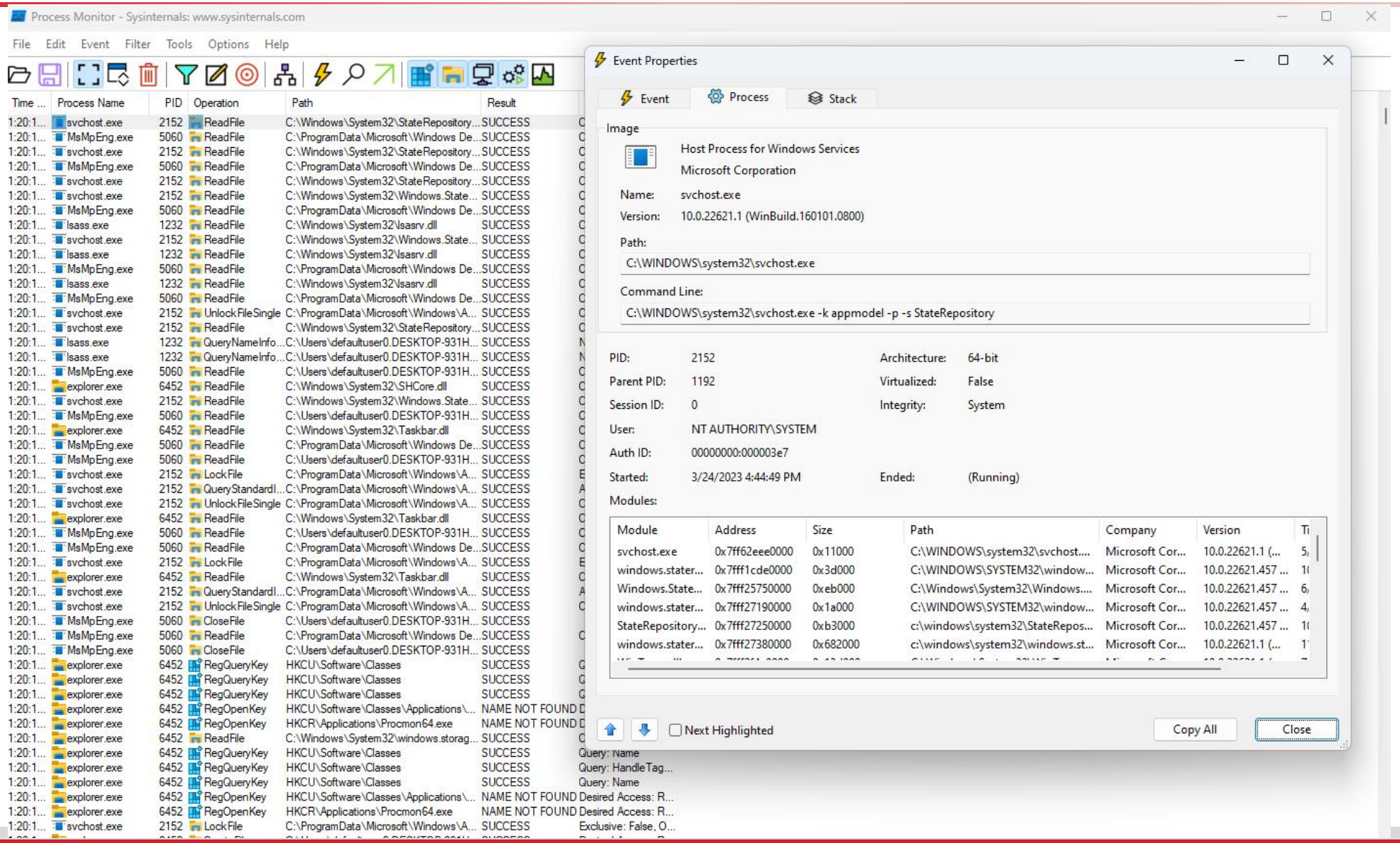


Moniter System Activity

Process Moniter

- Allow monitering of registry, file system, network, process, and thread activities.
- Moniter all system calls
- Captures a lot of data (>50,000 events in a minute)
- Use RAM to capture events
  - can easily crash a VM - so, run for a limited amount of time
- Not a reliable tool for network activities
  - so other tools needs to be used

Running Malware-Process Moniter



Running Malware-Process Moniter

- To narrow the result, use filtering
- You may want to filter on:

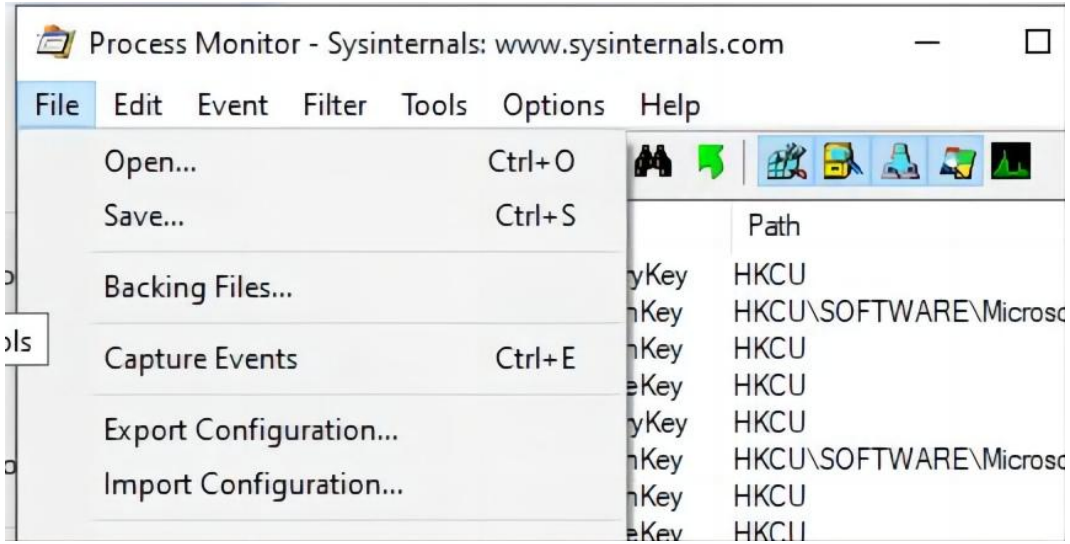
  Executables running on the system
  System call (such as RegSetValue, Create File, WriteFile, etc.)

Note: Filtering does not prevent from consuming too much memory though.

Example1-Track File and Registry Changes

- Let's say, you need to track access to the registry key HKEY_CURRENT_USER\Software\test and file c:\ps\procmon_example.txt.
- When Process Monitor starts, it begins capturing all events according to the default filters.

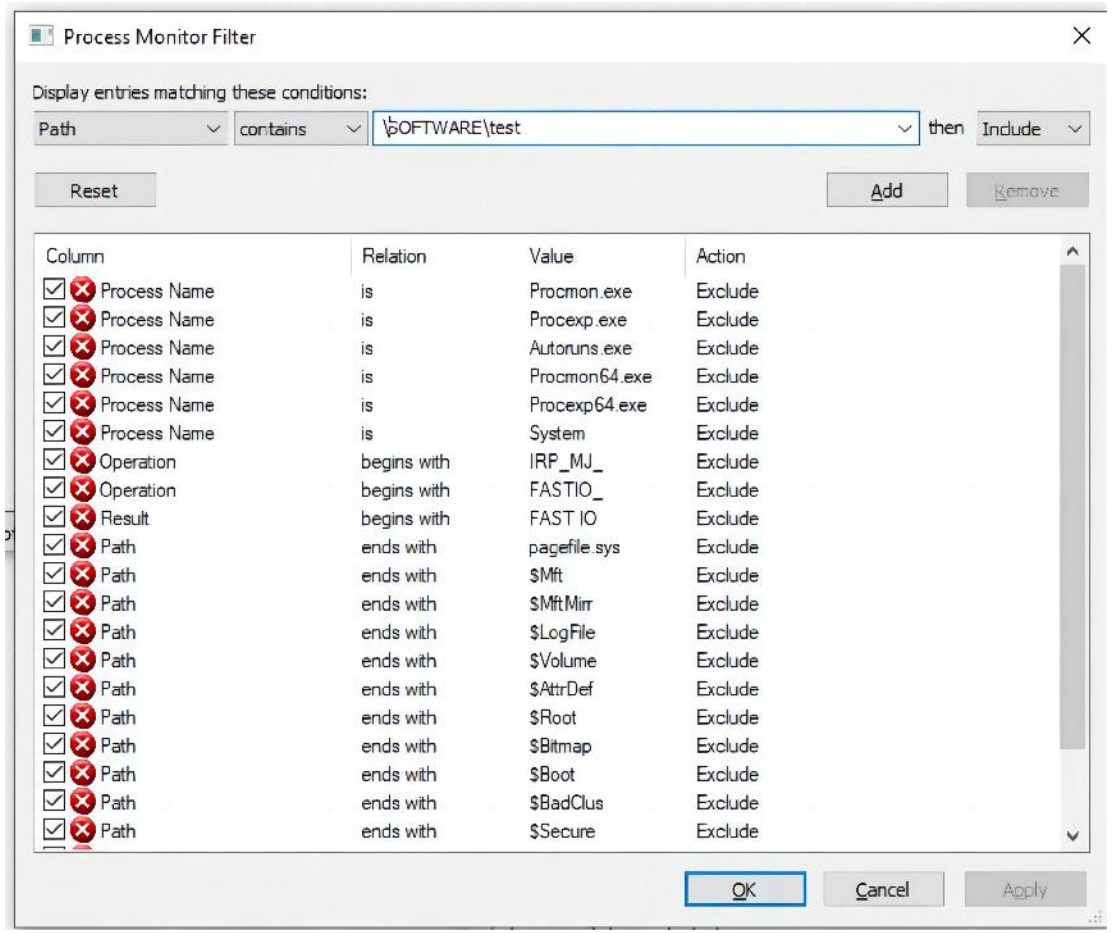
- Step 1. Stop capturing events by unchecking the option File > Capture Events (Ctrl+E).
- Step 2. Clear the current ProcMon log (Edit > Clear Display).



Example1-Track File and Registry Changes

Let's say, you need to track access to the registry key HKEY_CURRENT_USER\Software\test and file c:\ps\procmon_example.txt.

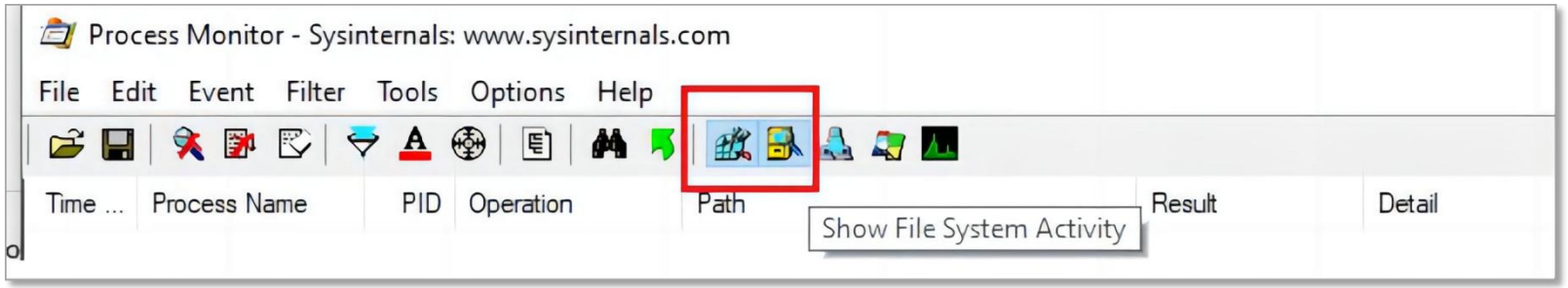
- Step 3. Now you need to configure the Process Monitor filters (Filter > Filter).
- Step 4. Create a filter for monitoring access to the registry key: Path > contains > \SOFTWARE\test > Include.
- Step 5. Click Add to add a new filter to the list



Example1-Track File and Registry Changes

Let's say, you need to track access to the registry key HKEY_CURRENT_USER\Software\test and file c:\ps\procmon_example.txt.

- Step 6. Now add a file access event filter: Path > is > c:\ps\procmon_example.txt > Include.
- Step 7. Make sure the following options are enabled in the toolbar: Show Registry Activity, Show File System Activity.
  - The Show Network Activity and Show Process, and Threads Activity options can be disabled.



Example1-Track File and Registry Changes

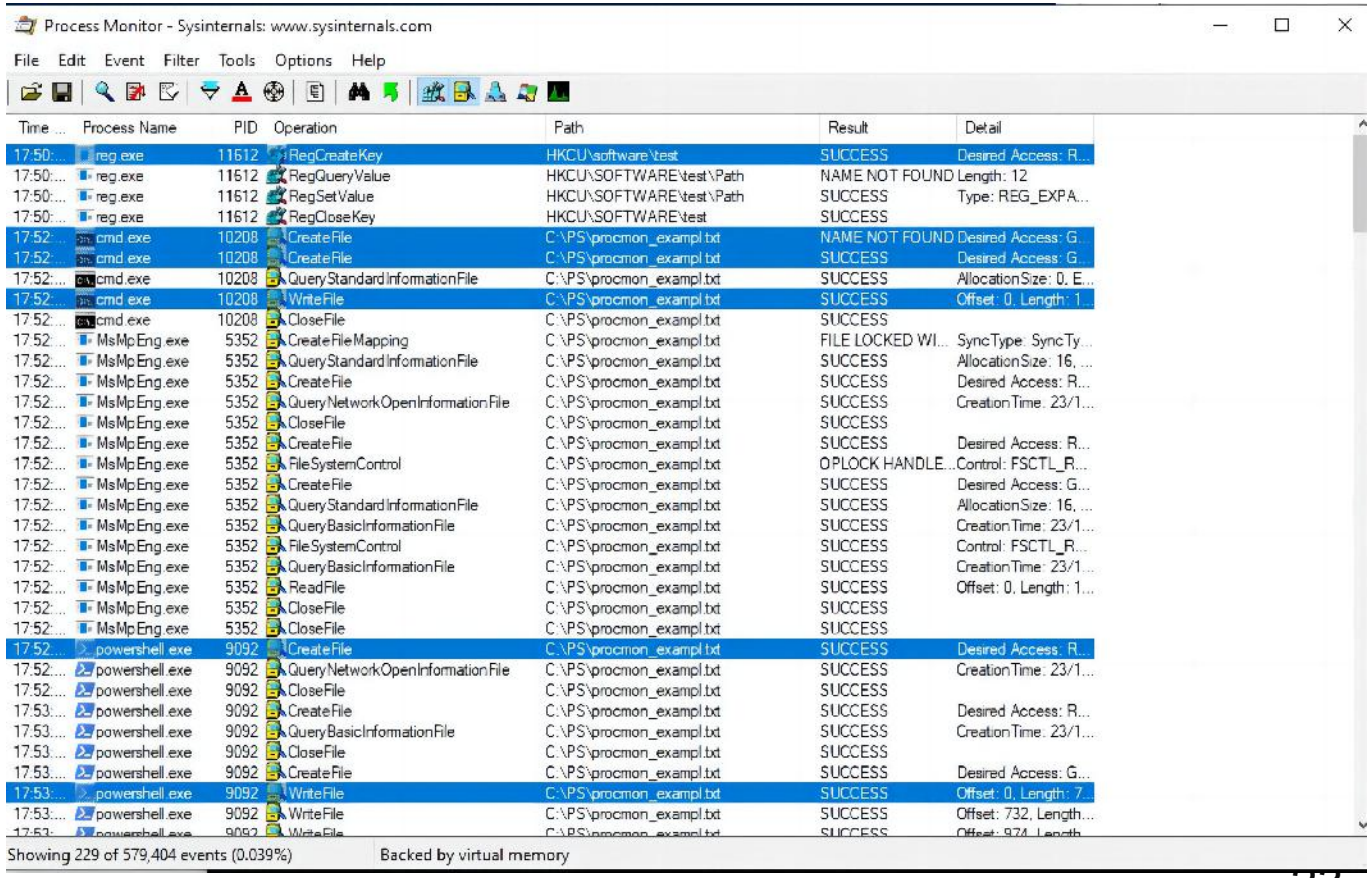
Let's say, you need to track access to the registry key HKEY_CURRENT_USER\Software\test and file c:\ps\procmon_example.txt.

- Step 8. Start event monitoring File > Capture Event.
- Step 9. Let's create a reg parameter key in the specified registry key using the command prompt:
  - reg add hkcu\software\test /v Path /t REG_EXPAND_SZ /d ^%systemroot^%
- Step 10. Let's write some data into the procmon_example.txt file using the command prompt:
  - echo %date%>>c:\ps\procmon_example.txt
- Step 11. And using PowerShell:
  - Get-Process|out-file C:\ps\procmon_example.txt

Example1-Track File and Registry Changes

Let's say, you need to track access to the registry key HKEY_CURRENT_USER\Software\test and file c:\ps\procmon_example.txt.

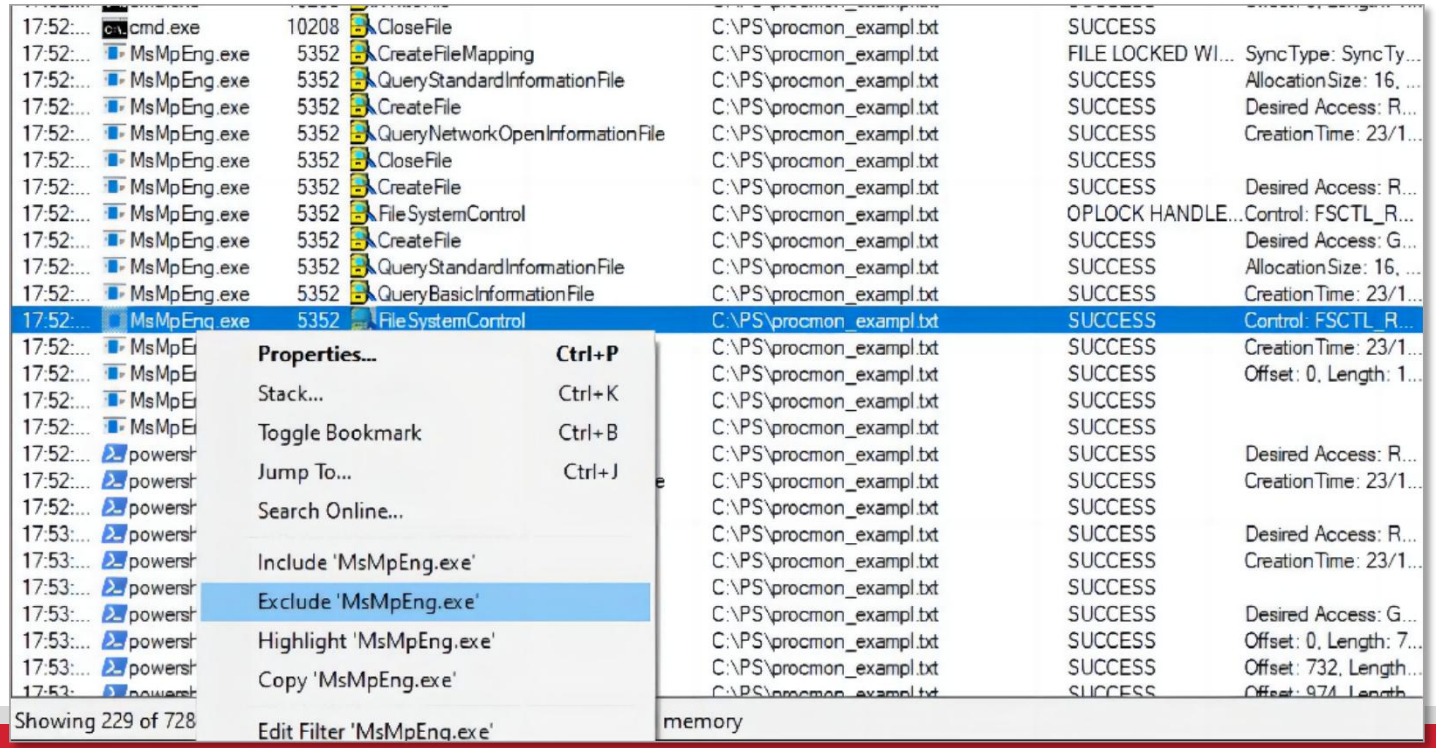
- It contains events for creating a registry key by the reg.exe process (Operation > RegCreateKey).
- It also contains events of creation (Create File) and writing to a file (WriteFile) by the processes cmd.exe and powershell.exe.



Example2-Exclude System Process

- Let's say, exclude msmpeng.exe (Antimalware Service Executable). This is the core process of the antimalware detection engine in Windows Defender.

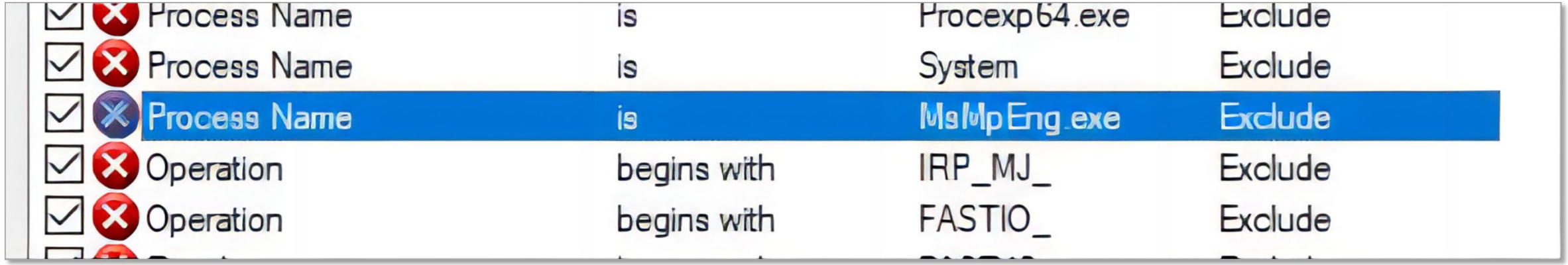
- To exclude the events of this process from the ProcMon log, right-click on the process name msmpeng.exe and select Exclude.



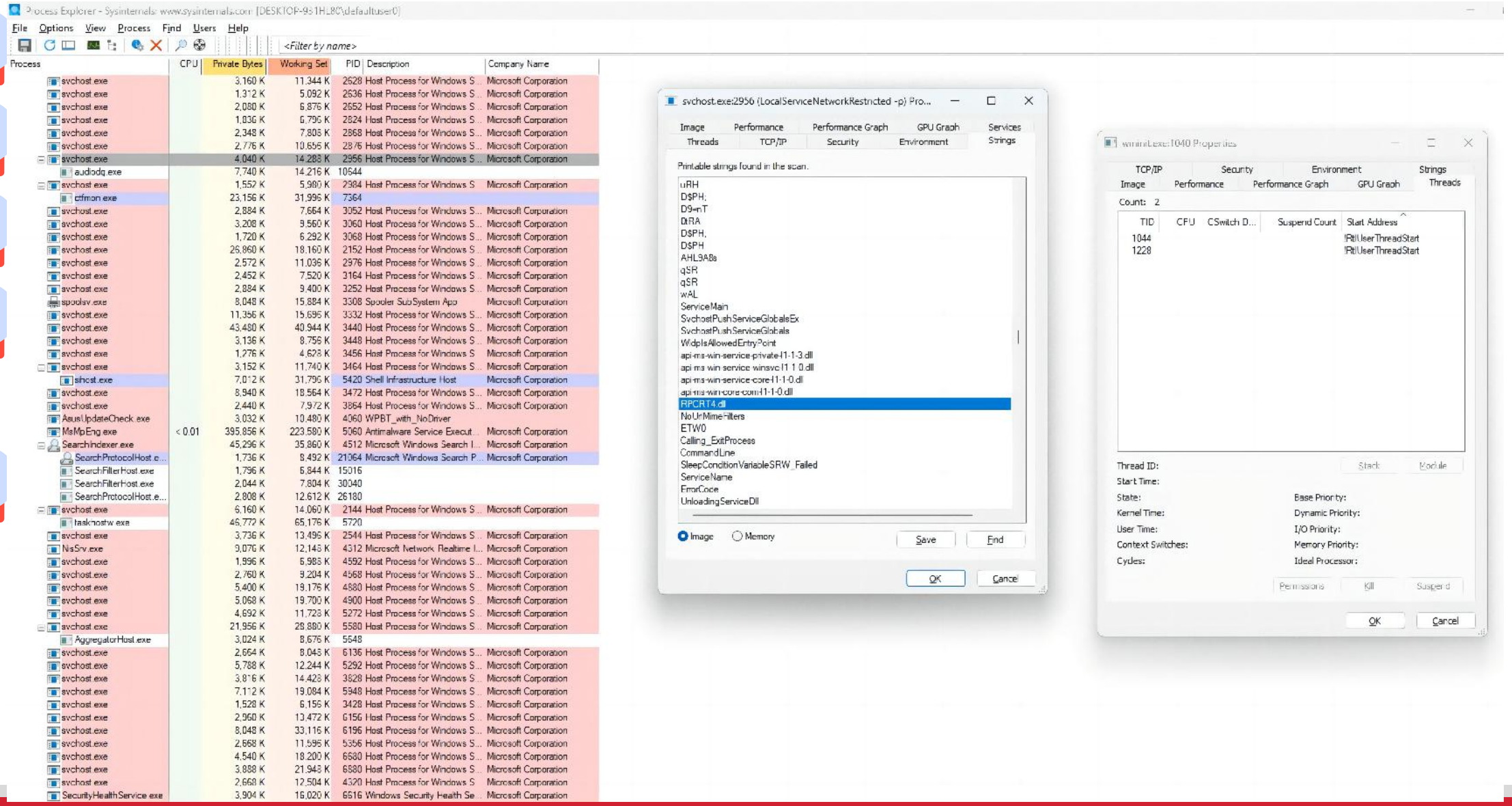
Example2-Exclude System Process

- Let's say, exclude msmpeng.exe (Antimalware Service Executable). This is the core process of the antimalware detection engine in Windows Defender.

- This process will be added to the ProcMon filter with the Exclude value. It means that the ProcMon log won't display any activity from this process.



Process Explorer



Process Explorer Vs. Process Moniter

Process Explorer

- Shows current state of each process
- Shows files, registry keys and thread loaded by each running process

Process Moniter

- In addition to monitoring, it logs process information- all events
- Logs show the file, registry, network, etc. the process attempted to use - successful or not
- “Access Denied” events also appear

Monitoring Network Activities

WHY?

- Most malware will need to communicate with external services/entities.
  - Download additional malware, files
  - Exchange/obtain keys for encryption
  - C2-Command and Control: Receive instructions and check-in
  - Extract data
  - Infect other machines
- **Question:** Do we allow them access to network?

Faking a Network

- It is too risky to allow a malware to access the network.
- Faking a network allows us to find out how/what is communicated.
- **Important:** Faking requires that the malware does not realize it is executing on a virtualized environment.

Faking a Network-FakeNet

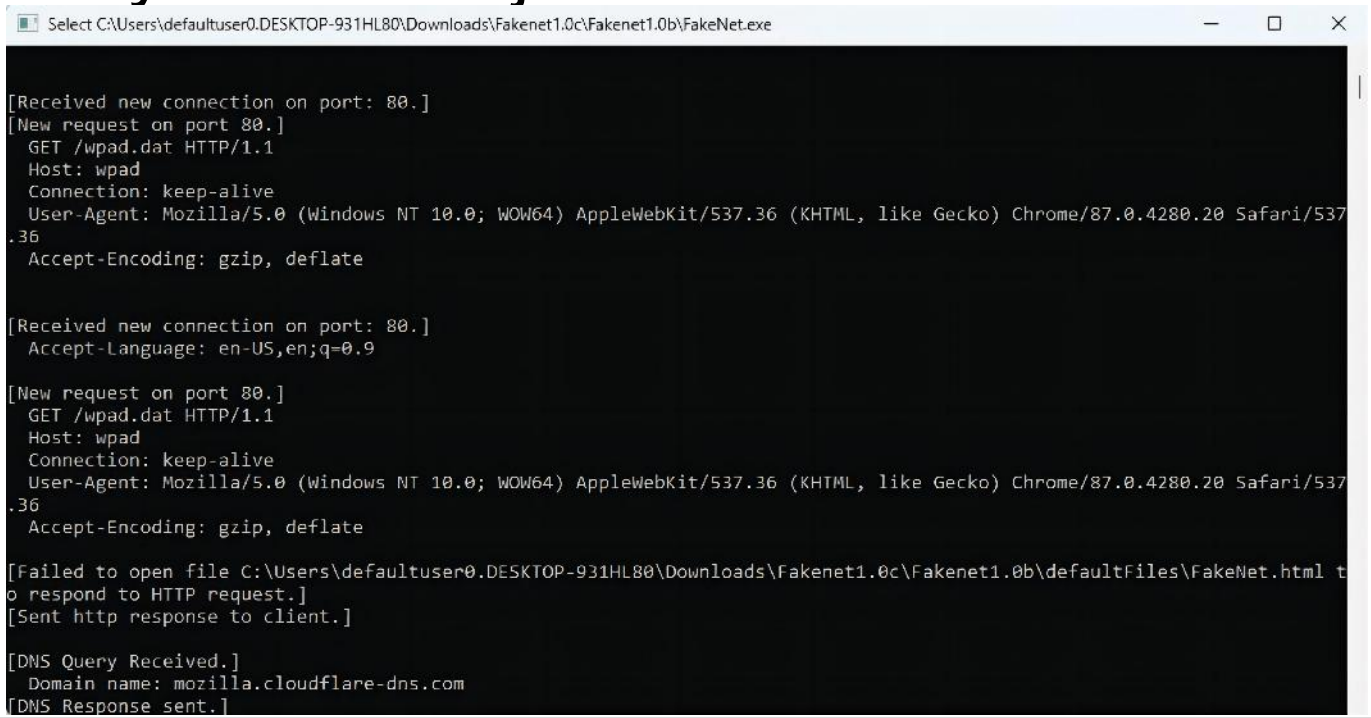
- An open source tool.
- Allow users to intercept and redirect all or specific network traffic.
- You can identify malware functionality and capture network signatures.

Faking a Network-FakeNet

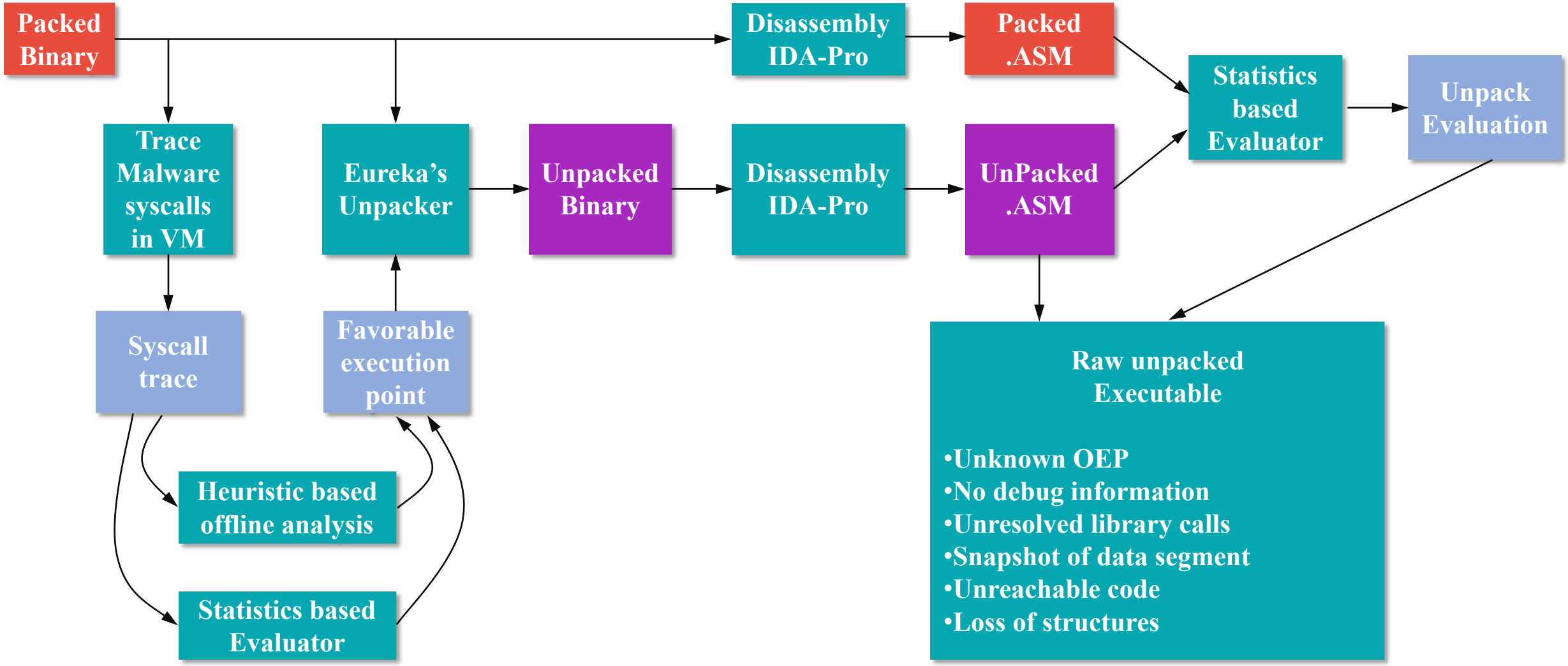
- 1. Fakenet takes over DNS on port 53.
- 2. It listens to the TCP ports 80, 443 and 25.
- 3. It supports DNS, HTTP and SSL protocols.

FakeNet-Use

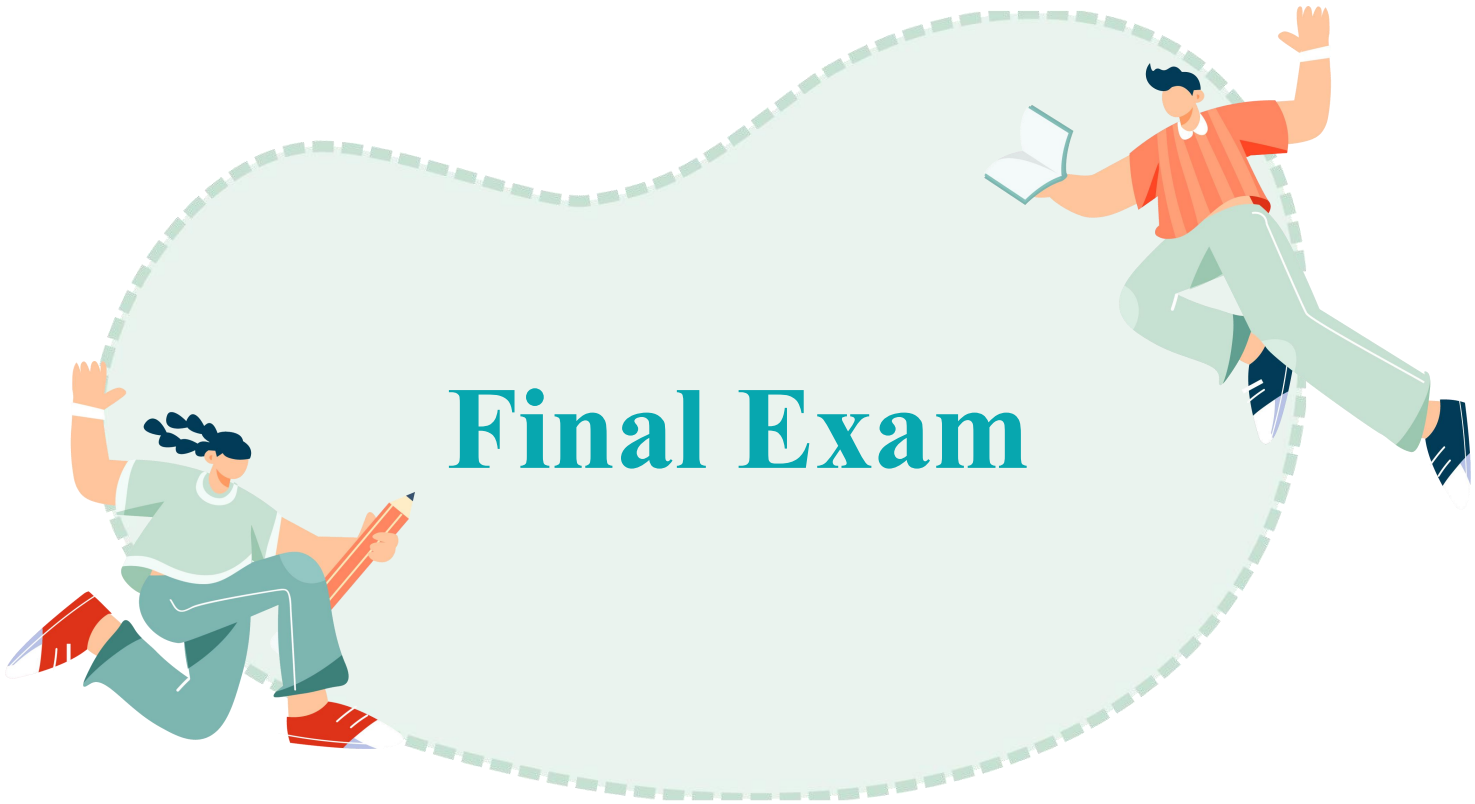
- 1. Stop most programs that connect to the Internet prior to running Fakenet.
- 2. Just run the program you want to analyze.
- 3. Still get some noise from Windows itself and maybe background processes that you cannot just terminate.



The Eureka Workflow



Questions



THE END

Fangtian Zhong

CSCI 591

Gianforte School of Computing
Norm Asbjornson College of Engineering
E-mail: fangtian.zhong@montana.edu

11/20/2025