TRY HACK ME: Basic Dynamic Analysis Write-Up

Shefali Kumari · Following
19 min read · May 19, 2023

Task 1 Introduction-

Previously, we learned techniques to analyze malware without executing it in the Basic Static Analysis room. However, as we have learned, malware can use techniques to hide its features from a malware analyst. But no matter how good malware hides its features from static analysis, its primary purpose is to execute. And when malware executes, it leaves traces that a malware analyst can use to identify if it's malicious. We will use basic dynamic analysis techniques in this room to analyze the traces malware leaves when running.

Learning Objectives:

In this room, we will learn:

- Sandboxing and using a sandbox for malware analysis.
- The components of a sandbox and how to create one for yourself.

- Using ProcMon to monitor a process' activity.
- Using API Logger and API Monitor to identify API calls made by malware.
- Using ProcExp to identify if a process is modified maliciously.
- Using Regshot to track registry changes made by malware.

Pre-requisites:

Before starting this room, it is recommended that you complete the following rooms for a better understanding of the content in this room.

- Introduction to Windows API
- Windows Internals
- Intro to Malware Analysis
- Basic Static Analysis

Answer to the questions of this section-

No Answer Needed

Task 2 Sandboxing-

In all the malware analysis rooms, it has been emphasized that malware should only be analyzed in a controlled environment, ideally a virtual machine. However, this becomes increasingly important for the dynamic analysis of malware. The primary concern regarding performing static analysis on malware in a live environment is an accidental execution, but we intentionally execute malware in a dynamic analysis scenario. This makes it all the more important to ensure that malware is analyzed in a sandboxed environment.

So what is required to create a sandbox?

Broadly, the following setup will be required to create a sandbox:

- An isolated machine, ideally a virtual machine, that is not connected to live or production systems and is dedicated to malware analysis.
- The ability of the isolated or virtual machine to save its initial clean state and revert to that state once malware analysis is complete. This functionality is often

called creating and reverting a snapshot. We will need to revert to the original clean state before analyzing a new malware so that infection from the previous malware doesn't contaminate the analysis of the next one.

- Monitoring tools that help us analyze the malware while it's executing inside the Virtual Machine. These tools can be automated, as we see in automated sandboxes, or they can be manual, requiring the analyst to interact while performing analysis. We will learn about some of these tools later in the room.
- A file-sharing mechanism that can be used to introduce the malware into the Virtual Machine and sends the analysis data or reports out to us. Often, shared directories or network drives are used for this purpose. However, we must be careful that the shared directory is unmounted when executing the malware, as the malware might infect all the files. This is especially true of ransomware, which might encrypt all shared drives or directories.

In the Intro to Malware Analysis room, we learned about some automated sandboxes to help perform dynamic analysis. Below, we will learn about some tools to help create our sandbox, which gives us more analysis control. So let's start.

Virtualization:

A lot of commercial and free tools are available for virtualization. Some of the most famous ones include Oracle's VirtualBox and VMware's Player and Workstation. These three tools allow us to create Virtual Machines isolated from our local machine. However, VMWare Player can't create snapshots. For dynamic analysis of

malware, snapshot creation is a critical requirement, which makes VMWare Player unsuitable for malware analysis. VMWare Workstation and VirtualBox have the snapshot creation option and are, therefore, suitable for malware analysis. VirtualBox is free, but VMWare Workstation has a paid license.

Apart from these, server-based virtualization software like XenServer, QEmu, ESXi, etc., help with virtualization on a dedicated server. This type of setup is often used by enterprises for their virtualization needs. Security research organizations often use similar technologies to create a VM farm for large-scale virtualization.

For the scope of this room, we will be skipping the step of creating a VM and installing an OS in it. Please note that the VM's OS needs to be the same as the malware's target OS for dynamic analysis. In most scenarios, this will be the Windows OS. We will be covering tools related to Windows OS in this room.

Analysis Tools:

Once we have a VM with the OS installed, we need to have some analysis tools on the VM. Automated malware analysis systems have some built-in tools that analyze malware behaviour. For example, in Cuckoo's sandbox, cuckoomon is a tool that records malware activity in a Cuckoo sandbox setup. In the coming tasks, we will learn about some tools to perform manual dynamic analysis of malware. Once we have our required tools installed on the VM and before running any malware on the VM, we must take a snapshot. After analysis of every malware, we must revert the VM to this snapshot, which will hold the clean state of the VM. This will ensure that our analysis is not contaminated by different malware samples running simultaneously.

File-sharing:

Different platforms provide different options for sharing files between host and guest OS. In the most popular tools, i.e., Oracle VirtualBox or VMWare Workstation, the following options are common:

- Shared folder.
- Creating an iso in the host and mounting it to the VM.
- Clipboard copy and paste.

Apart from these, there are other, less common options, for example, running a web server on the guest where malware samples can be uploaded or mounting a

removable drive to the Virtual Machine. Please note that the more isolated the option to share files, the safer it will be for the host OS. Apart from sharing malware with the VM, the file-sharing option is also used to extract analysis reports from the VM.

Once we have created a VM, set up analysis tools, taken a snapshot, and placed the malware inside our sandbox, we can start analysing our malware. In the next task, we will learn about tools to help us.

Answer to the questions of this section-

Task 3 ProcMon-

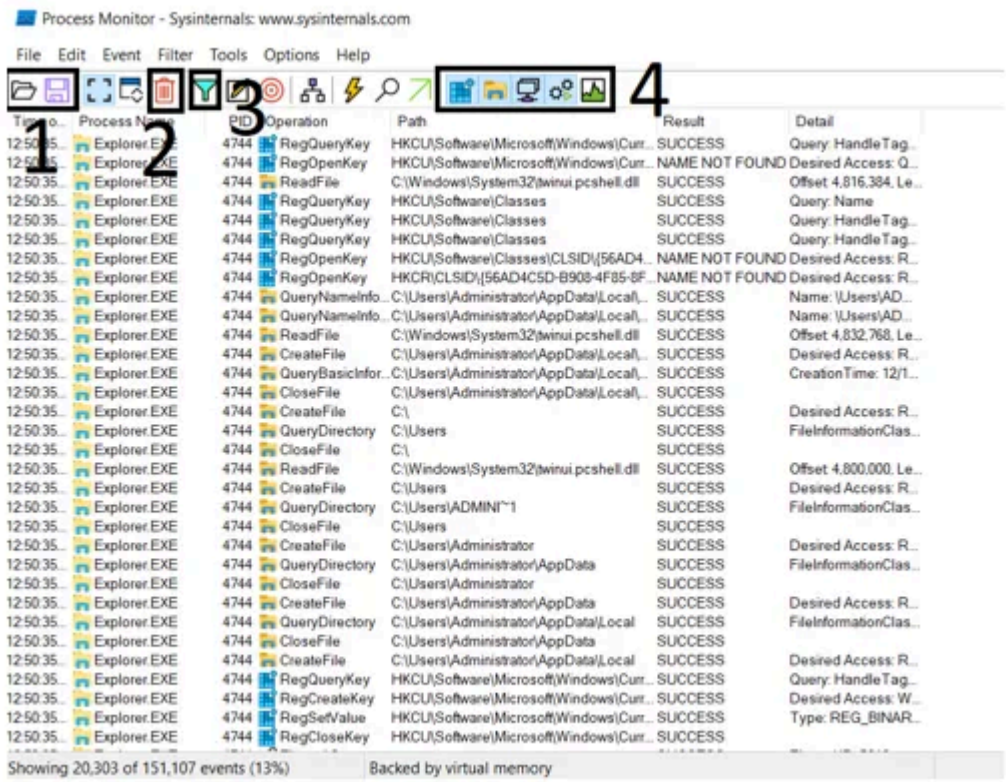
In this task, we will learn how to use Process Monitor, or ProcMon, to analyze malware's activities. ProcMon is part of the Sysinternals suite, a set of utilities created by a company named Winternals Software and purchased by Microsoft in 2006. Sysinternals consists of many handy utilities that provide advanced functionalities for Windows. Sysinternals utilities are widely used in Security research, and we will cover some of them in this room and from time to time in other rooms as well. So let's start with ProcMon.

Before moving forward, please start the attached VM. The VM will open in split view. Alternatively, you can use the following credentials to log into the machine:

Username: Administrator

Password: Passw0rd!

Once the machine has started, navigate to the following location to start ProcMon. Desktop > Tools > Utilities > procmon.exe. Once ProcMon is launched, the following window will appear.



The controls of ProcMon are self-explanatory, and a brief description is shown if we hover over one of the controls. The labels in the screenshot show some of the critical controls of the data visible below these controls.

1. Shows the Open and Save options. These options are for opening a file that contains ProcMon events or saving the events to a supported file.

2. Shows the Clear option. This option clears all the events currently being shown by ProcMon. It is good to clear the events once we execute a malware sample of interest to reduce noise.

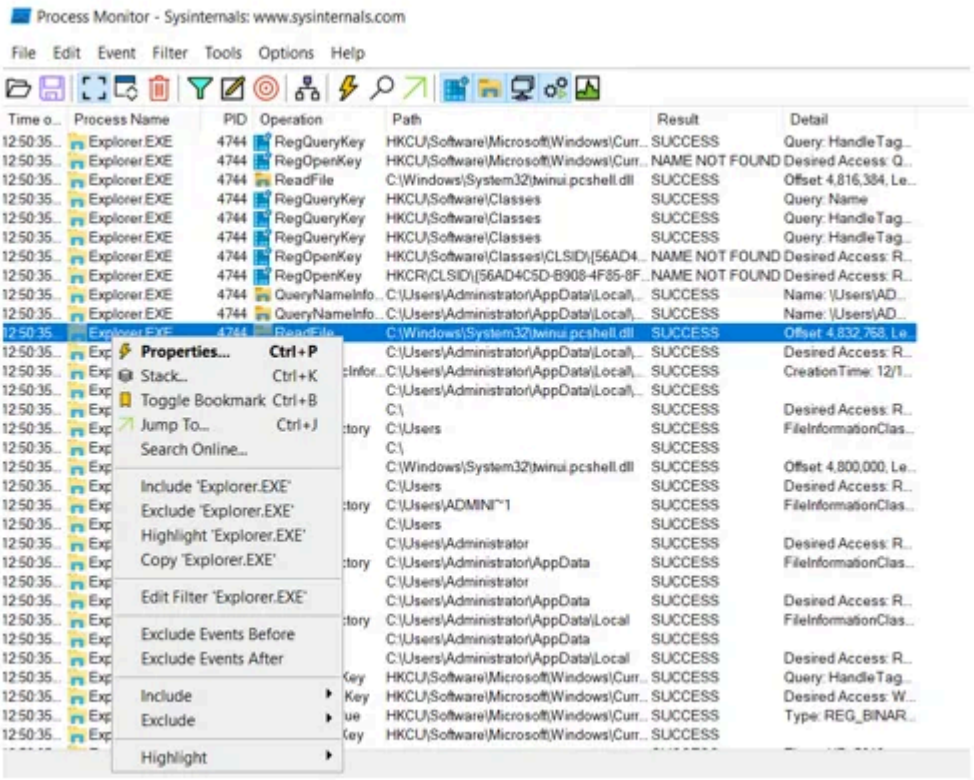
3. Shows the Filter option, which gives us further control over the events shown in the ProcMon window.

4. These are toggles to turn off or on Registry, FileSystem, Network, Process/Thread, and Profiling events.

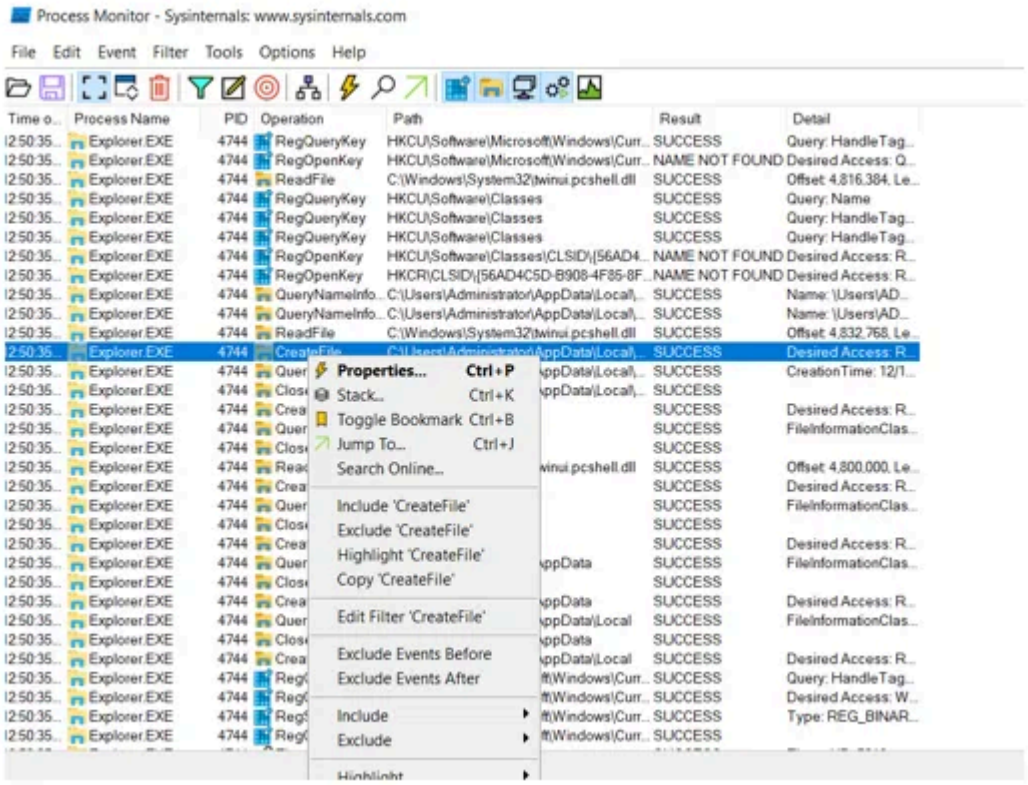
Below these controls, we can see from left to right the Time, Process, Process ID (PID), Event Name, Path, Result and Details of the activity. We can observe that events are shown in chronological order. Generally, ProcMon will show an overwhelming number of events occurring on the system. For ease of analysis, it is wise to filter the events to those of our interest.

Filtering Events:

ProcMon allows easy filtering of events from the events window itself. For example, check out the below screenshot.



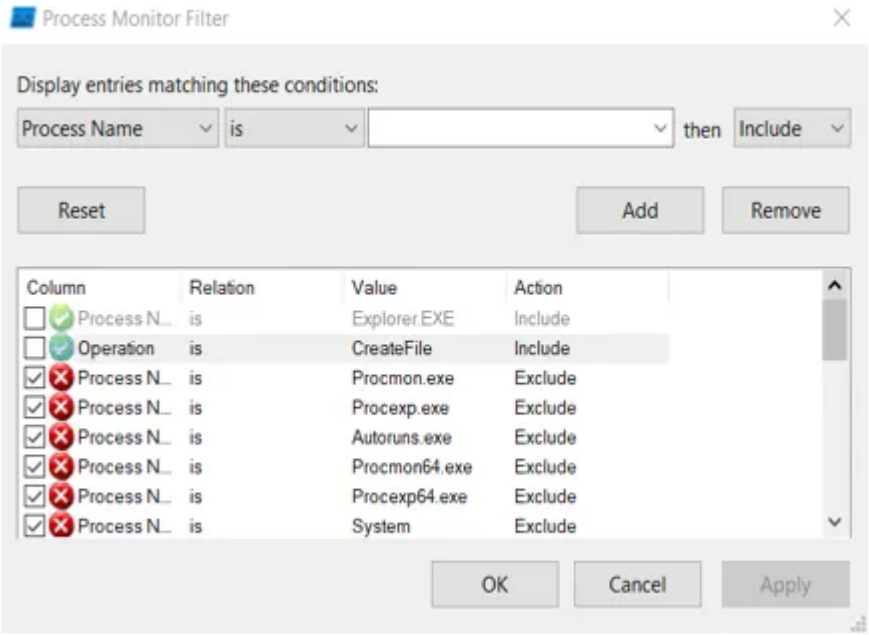
If we right-click on the process column on the process of our choice, a pop-up menu opens up. We can see different options in the pop-up menu. Some of these options are related to filtering. For example, if we choose the option `Include 'Explorer.EXE'`, ProcMon will only show events with Process Name Explorer.EXE. If we choose the option `Exclude 'Explorer.EXE'`, it will exclude Explorer.EXE from the results. Similarly, we can right-click on other columns of the events window to filter other options.



As seen in the screenshot above, when we right-click on an event, we can filter in/out an event. Similarly, we can add more filters to the results until we narrow down the results to the events of our interest. If we choose the `Include 'Explorer.EXE'` and `Include 'CreateFile'` events, ProcMon will only show us CreateFile events triggered by Explorer.EXE.

Advanced Filtering:

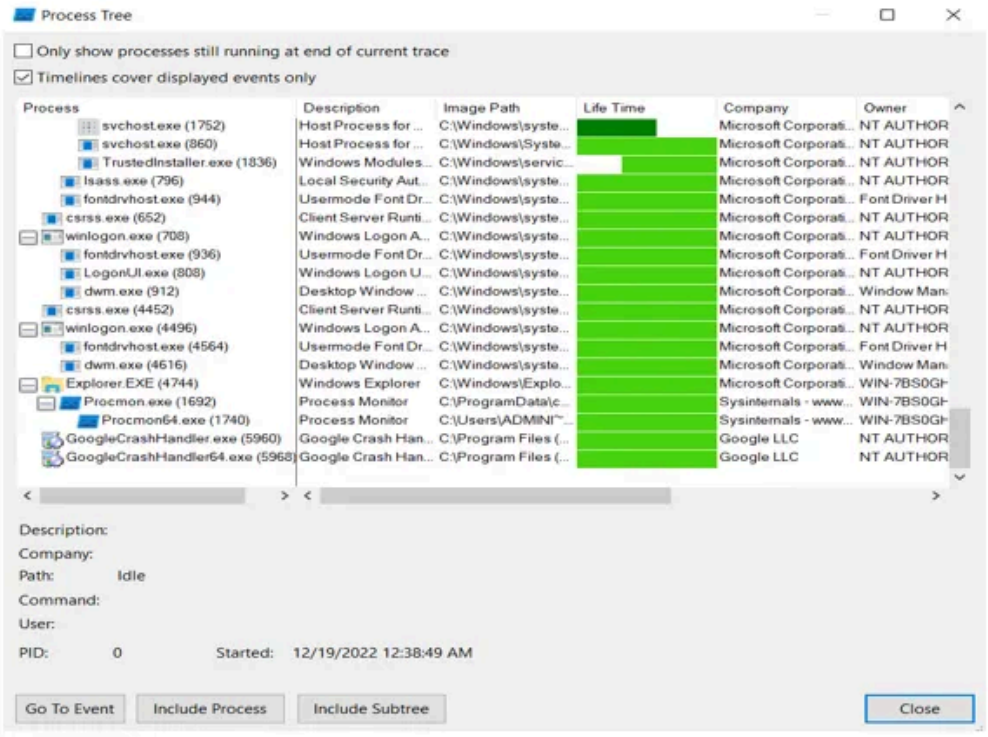
ProcMon also allows us to implement advanced filters. In the menu marked as number 3 in the first image in this task, we can see the option for filtering. When we click on this option, we see the following window pop up.



We can see some preset filters already applied in ProcMon, like the one for filtering out Procmon.exe. We can see that the filter process is quite simple to implement. We select filtering values, like Process Name, its relation, value, and action. If the checkbox is ticked, the filter is applied. Otherwise, the filter is ignored. We can see that the first two filters are not applied in this screenshot. The third filter states that if Process Name is Procmon.exe, then Exclude that event from reporting. Therefore, we don't see any events related to Procmon.exe. Here, it must be noted that an 'include' filter will show events related to only that entity. For example, if we include Explorer.EXE, only events with Process Name Explorer.EXE will be shown, and the rest will be filtered out.

Process Tree:

ProcMon also allows us to view all the existing processes in a parent-child relationship, forming a process tree. This can be done by clicking the  icon in the menu. This option helps identify the parents and children of different processes. As shown by ProcMon, an example process tree can be seen below.



Although a Process Tree is a good piece of information when analysing malware, we will look at it in detail when we explore ProcExp later in the room.

Answer to the questions of this section-

Monitor the sample `~Desktop\Samples\1.exe` using ProcMon. This sample makes a few network connections. What is the first URL on which a network connection is made?

94-73-155-12.cizgi.net.tr:2448 Correct Answer Hint

What network operation is performed on the above-mentioned URL?

TCP reconnect Correct Answer Hint

What is the name with the complete full path of the first process created by this sample?

C:\Users\Administrator\Desktop\samples\1.exe Correct Answer Hint

Before moving to the next task, terminate the VM instance and start it again so that we have our VM restarted from the snapshot. That way, it is not contaminated with the malware execution we already performed.

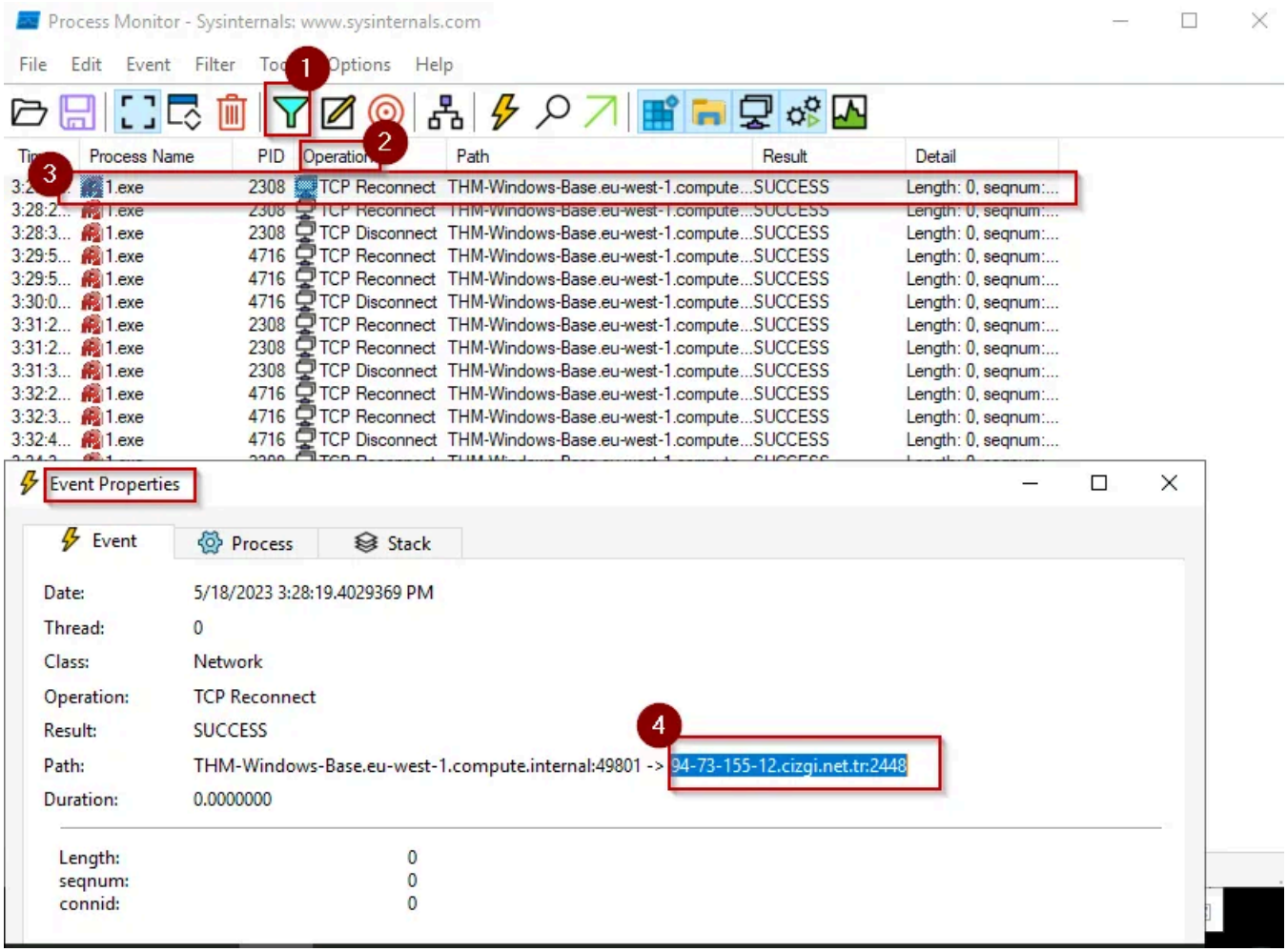
No answer needed Correct Answer

Answer:

1) Apply filter:

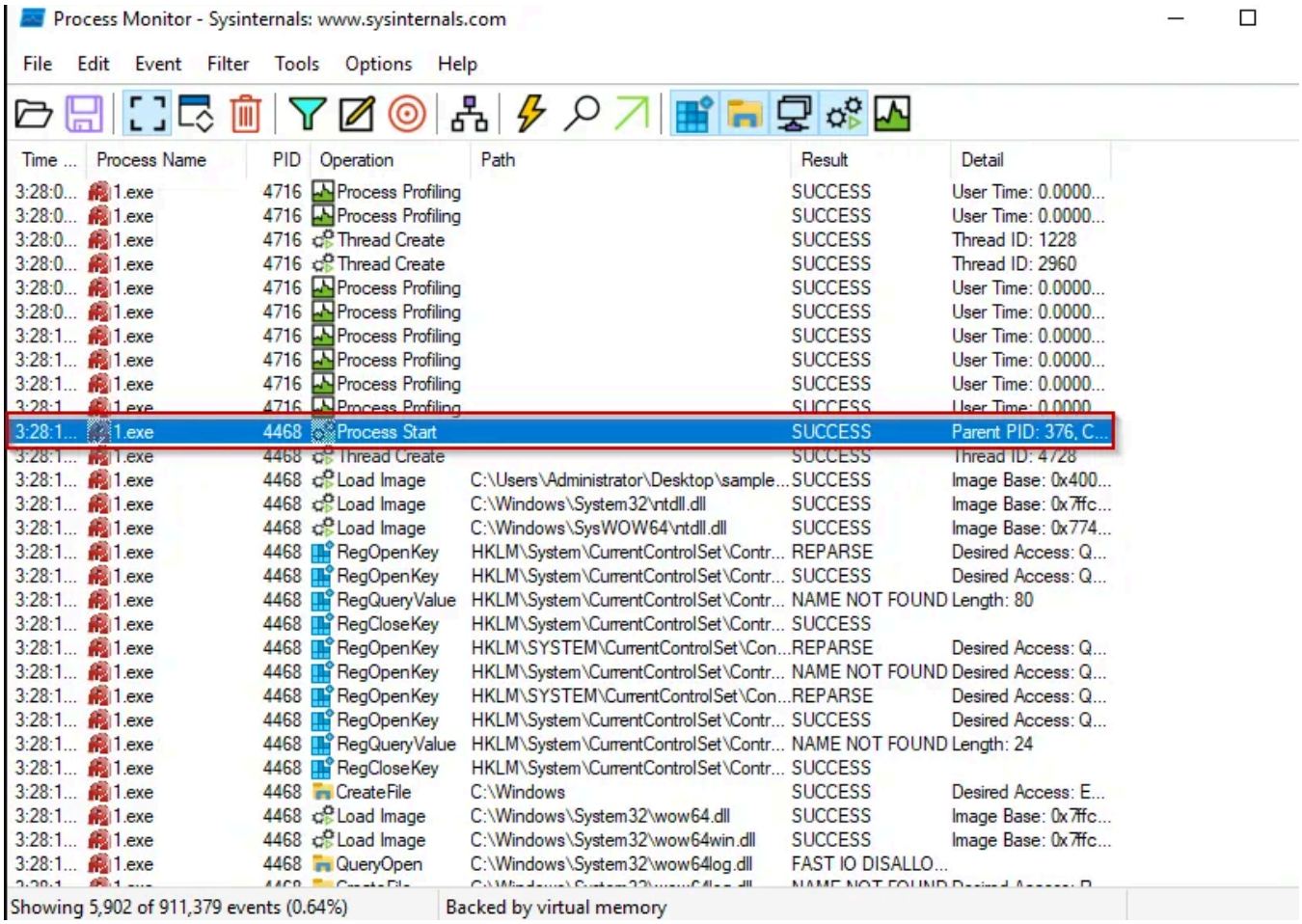
“Process name is 1.exe

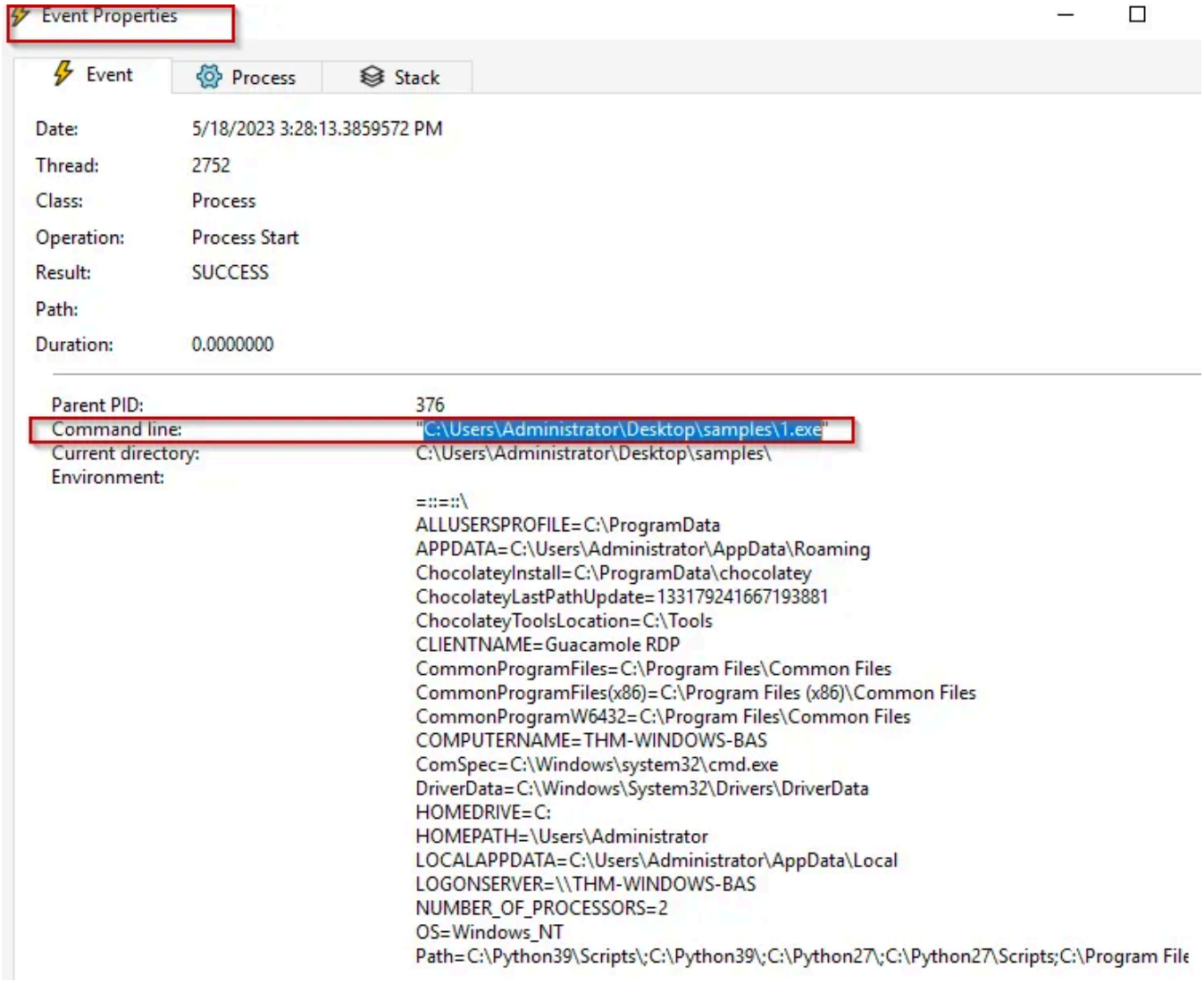
operation contains TCP”



2) Apply filter: “process name is 1.exe “

look for process start


Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help
Time ... Process Name PID Operation Path Result Detail
3:28:0... 1.exe 4716 Process Profiling SUCCESS User Time: 0.0000...
3:28:0... 1.exe 4716 Process Profiling SUCCESS User Time: 0.0000...
3:28:0... 1.exe 4716 Thread Create SUCCESS Thread ID: 1228
3:28:0... 1.exe 4716 Thread Create SUCCESS Thread ID: 2960
3:28:0... 1.exe 4716 Process Profiling SUCCESS User Time: 0.0000...
3:28:0... 1.exe 4716 Process Profiling SUCCESS User Time: 0.0000...
3:28:1... 1.exe 4716 Process Profiling SUCCESS User Time: 0.0000...
3:28:1... 1.exe 4716 Process Profiling SUCCESS User Time: 0.0000...
3:28:1... 1.exe 4716 Process Profiling SUCCESS User Time: 0.0000...
3:28:1... 1.exe 4468 Process Start SUCCESS Parent PID: 376, C...
3:28:1... 1.exe 4468 Thread Create SUCCESS Thread ID: 4728
3:28:1... 1.exe 4468 Load Image C:\Users\Administrator\Desktop\sample... SUCCESS Image Base: 0x400...
3:28:1... 1.exe 4468 Load Image C:\Windows\System32\ntdll.dll SUCCESS Image Base: 0x7ffc...
3:28:1... 1.exe 4468 Load Image C:\Windows\SysWOW64\ntdll.dll SUCCESS Image Base: 0x774...
3:28:1... 1.exe 4468 RegOpenKey HKLM\System\CurrentControlSet\Contr... REPARSE Desired Access: Q...
3:28:1... 1.exe 4468 RegOpenKey HKLM\System\CurrentControlSet\Contr... SUCCESS Desired Access: Q...
3:28:1... 1.exe 4468 RegQueryValue HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND Length: 80
3:28:1... 1.exe 4468 RegCloseKey HKLM\System\CurrentControlSet\Contr... SUCCESS
3:28:1... 1.exe 4468 RegOpenKey HKLM\SYSTEM\CurrentControlSet\Con... REPARSE Desired Access: Q...
3:28:1... 1.exe 4468 RegOpenKey HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND Desired Access: Q...
3:28:1... 1.exe 4468 RegOpenKey HKLM\SYSTEM\CurrentControlSet\Con... REPARSE Desired Access: Q...
3:28:1... 1.exe 4468 RegOpenKey HKLM\System\CurrentControlSet\Contr... SUCCESS Desired Access: Q...
3:28:1... 1.exe 4468 RegQueryValue HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND Length: 24
3:28:1... 1.exe 4468 RegCloseKey HKLM\System\CurrentControlSet\Contr... SUCCESS
3:28:1... 1.exe 4468 CreateFile C:\Windows SUCCESS Desired Access: E...
3:28:1... 1.exe 4468 Load Image C:\Windows\System32\wow64.dll SUCCESS Image Base: 0x7ffc...
3:28:1... 1.exe 4468 Load Image C:\Windows\System32\wow64win.dll SUCCESS Image Base: 0x7ffc...
3:28:1... 1.exe 4468 QueryOpen C:\Windows\System32\wow64log.dll FAST IO DISALLO...
Showing 5,902 of 911,379 events (0.64%)
Backed by virtual memory



Task 4 API logger and API monitor –

Before starting this task, restart the VM attached to the previous task.

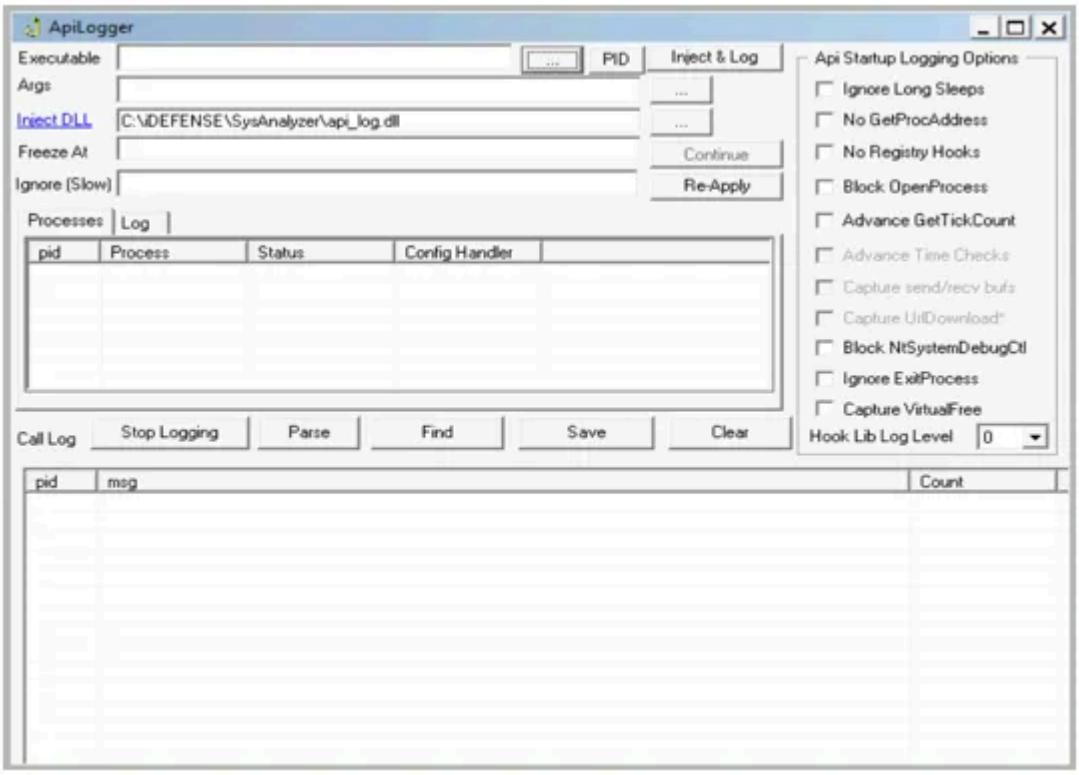
The Windows OS abstracts the hardware and provides an Application Programmable Interface (API) for performing all tasks. For example, there is an API for creating files, an API for creating processes, an API for creating and deleting registries and so on. Therefore, one way to identify malware behaviour is to monitor which APIs a malware calls. The names of the APIs are generally self-explanatory. However, Microsoft Documentation can be referred to for finding information about the APIs.

In this task, we will learn about API logger and API monitor tools which can help us identify what API calls malware is making.

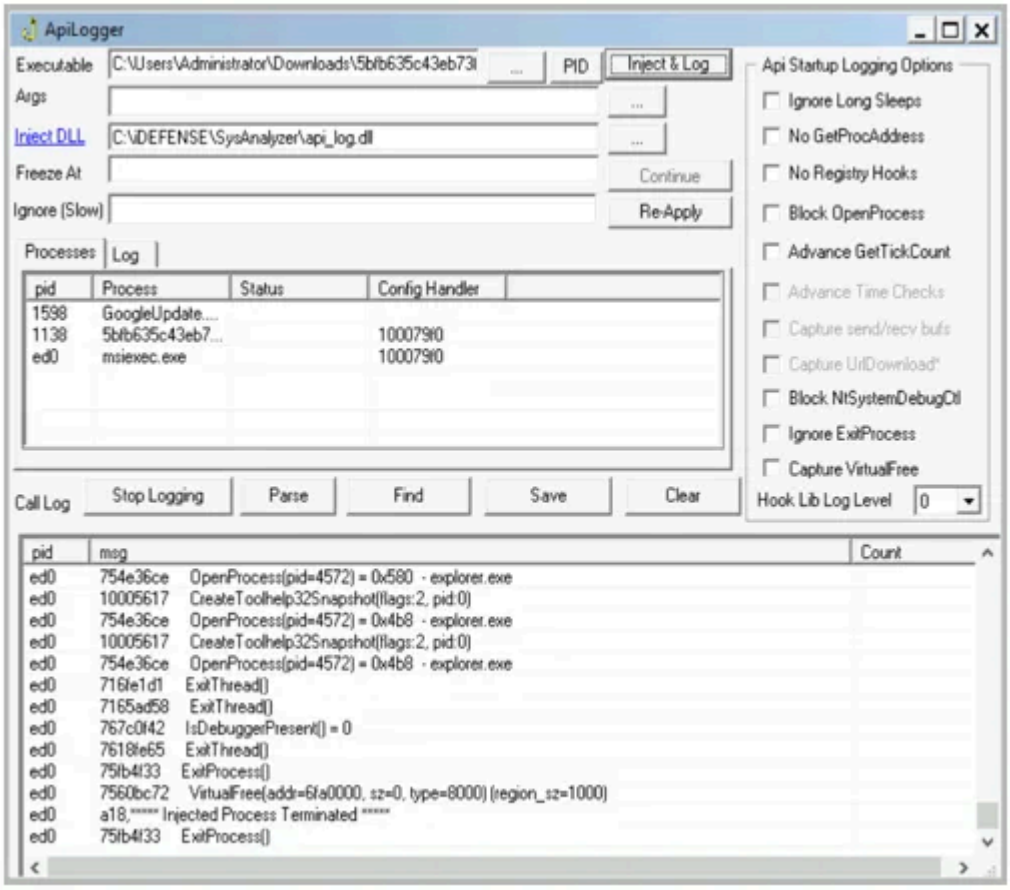
API Logger:

The API Logger is a simple tool that provides basic information about APIs called by a process. We can start API Logger in the attached VM by navigating to the path

`~Desktop\Tools\Utilities\ApiLogger.exe` . When we open the API logger tool, we see the following interface.

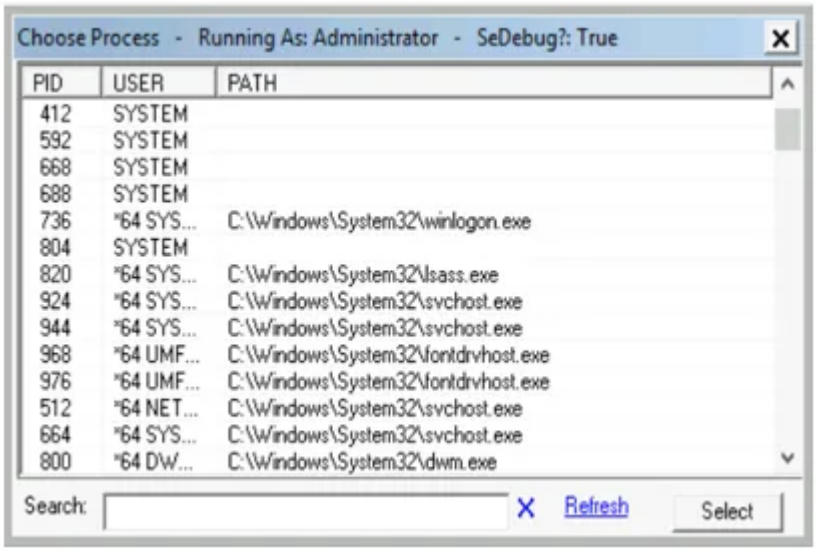


To open a new process, we can click the highlighted three-dot menu. When clicked, a file browser allows us to select the executable for which we want to monitor the API calls. Once we select the executable, we can click ‘Inject & Log’ to start the API logging process. We will see the log of API calls in the lower pane, as seen in the picture below. In the upper pane, we see the running processes and their PIDs.



We can see the PID of the process we monitor and the API called with basic information about the API in the ‘msg’ field.

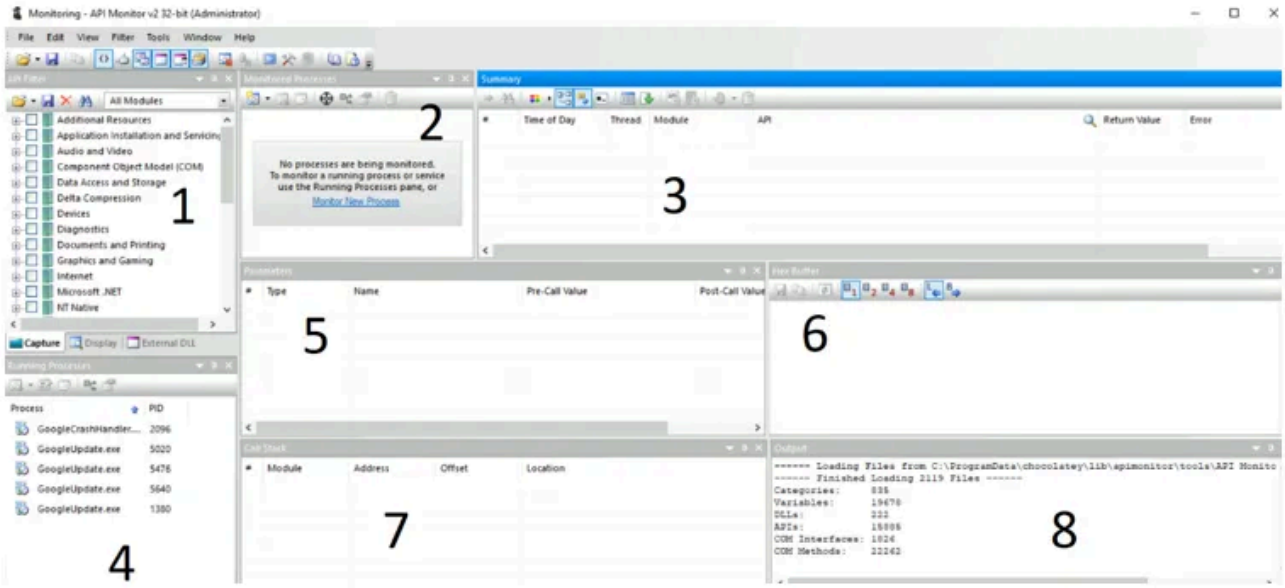
We can click the ‘PID’ menu for the API logger to log API calls of a running process. It will open the following window.



This Window shows processes with PIDs, the User that ran that process, and the image path of the process. The rest of the process is the same as the case with starting our process.

API Monitor:

The API Monitor provides more advanced information about a process's API calls. API Monitor has 32-bit and 64-bit versions for 32-bit and 64-bit processes, respectively. We can launch API Monitor by navigating to the path `~Desktop\Tools\Utilities\apimonitor-x64.exe` or `~Desktop\Tools\Utilities\apimonitor-x86.exe`. When we open API Monitor, we see the following Window.



As we can see, API Monitor has multiple tabs, as numbered in the image above.

1. This tab is a filter for the API group we want to monitor. For example, we have a group for 'Graphics and Gaming' related APIs, another for 'Internet' related APIs and so on. API Monitor will only show us APIs from the group we select from this menu.

2. This tab shows the processes being monitored for API calls. We can click the 'Monitor New Process' option to start monitoring a new process.

3. This tab shows the API call, the Module, the Thread, Time, Return Value, and any errors. We can monitor this tab for APIs called by a process.

4. This tab shows running processes that API Monitor can monitor.

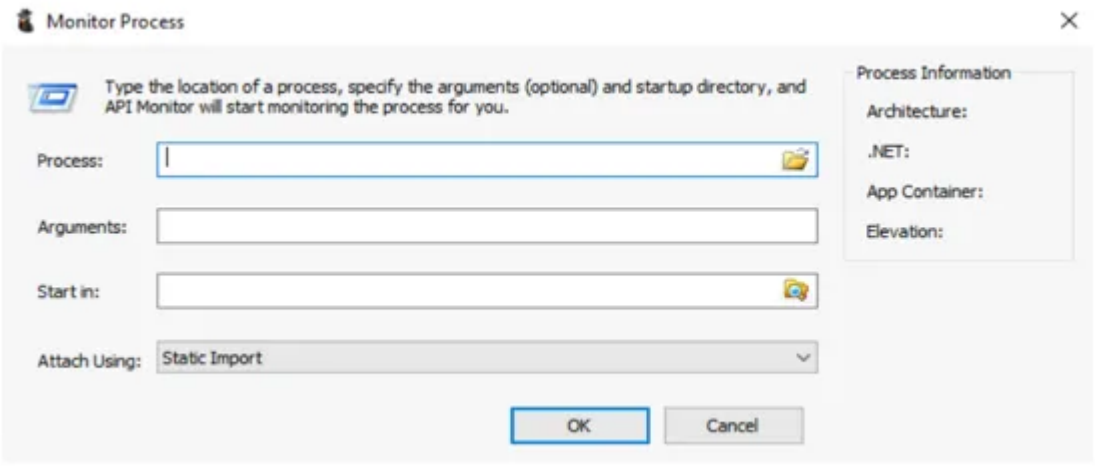
5. This tab shows the Parameters of the API call, including the values of those Parameters before and after the API calls.

6. This tab shows the Hex buffer of the selected value.

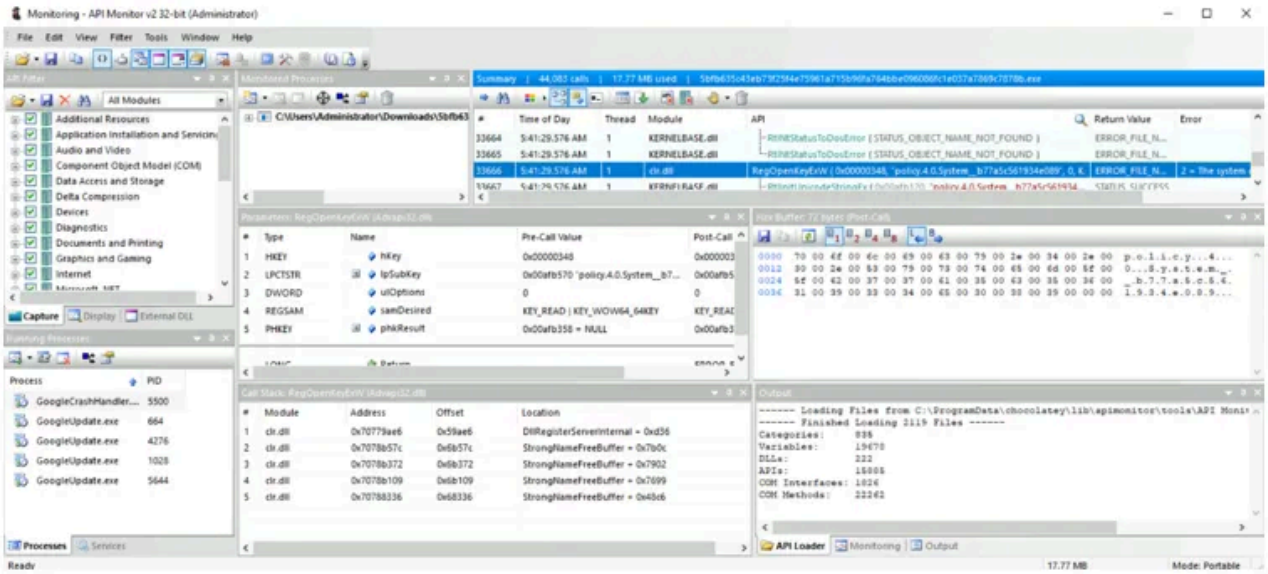
7. This tab shows the Call Stack of the process.

8. Finally, this tab shows the Output.

To understand it better, let's open a process in API Monitor. When we click the 'Monitor New Process' option in Tab 2, we see the following option.



In this menu, we can select the Process from a path, any arguments the process takes, the directory from where we want to start the process, and the method for attaching API Monitor. We can ignore the 'Arguments' and 'Start in' options if we don't have any arguments for the process and want to start it from the path where it is already located in. Once we open a process, we see the tabs populate as seen in the following image.

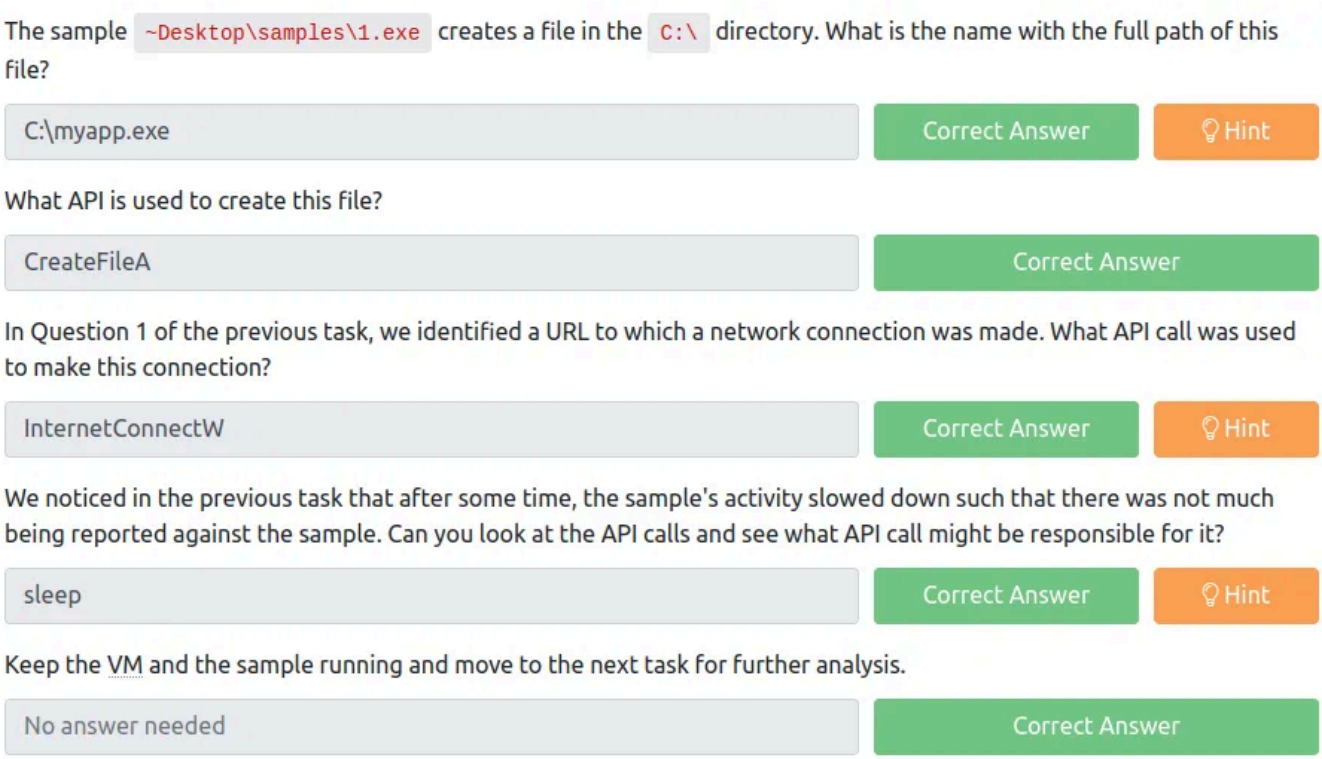


In the above image, we can see all the tabs being populated.

- In Tab 1, we see that we have selected all values so that we can monitor all the API calls.
- In Tab 2, we see the path of the process we are monitoring.
- In Tab 3, we see a summary of the API calls. The highlighted API call can be seen as RegOpenKeyExW. Hence we know that the process tried to open a registry key. We see that the API call returns an error, which we can see in the 'Return Value' field of this tab, and the error details can be found in this tab's 'Error' field.
- Tab 5 shows the parameters of the API call from before and after the API call was made.
- Tab 6 shows the selected value in Hex.
- Tab 7 shows the Call Stack of the process.

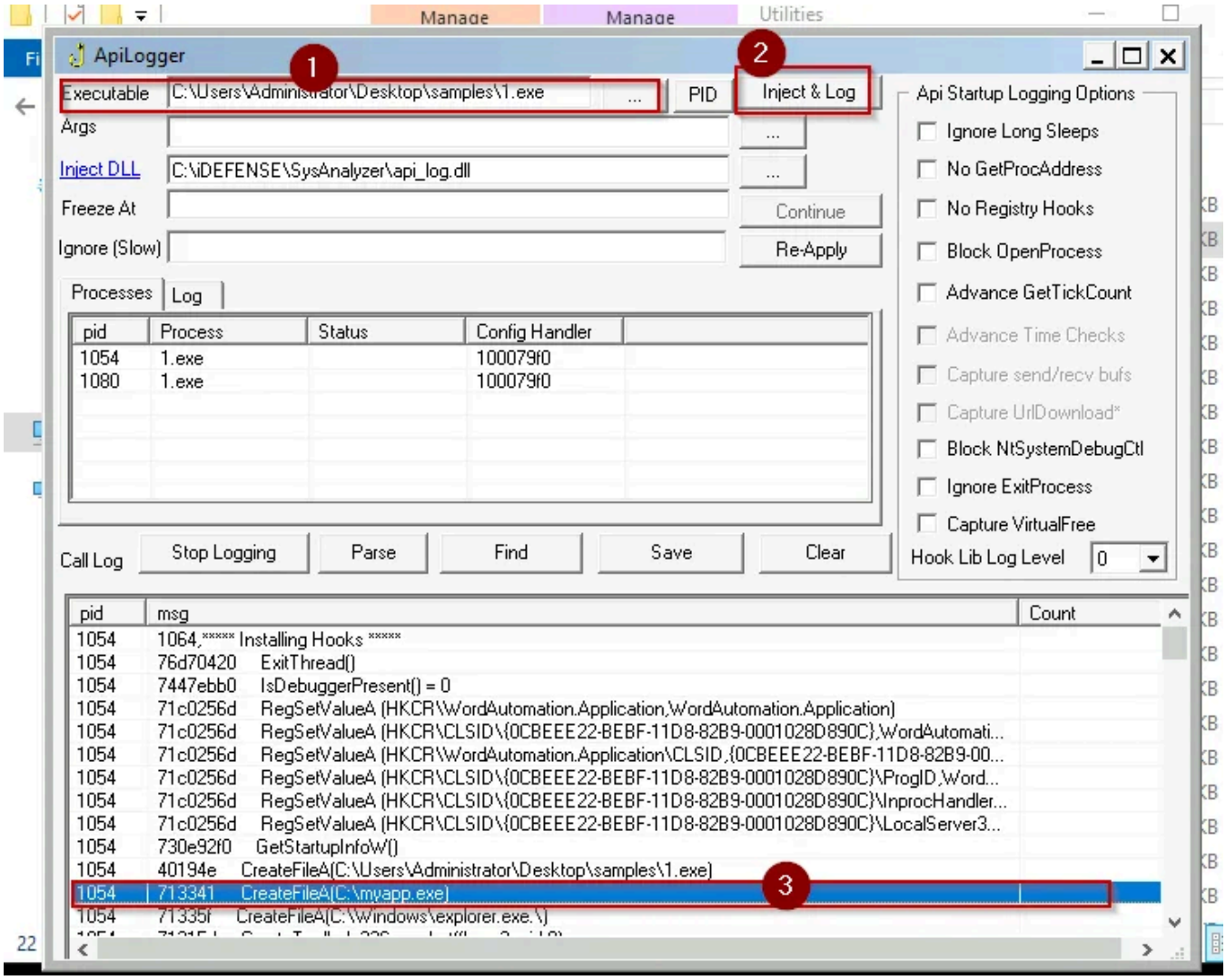
We see that API Monitor provides us with much more information about API calls by a process than API Logger. However, we must slow down the analysis process to digest all this information. When analyzing malware, we can decide whether to use API Logger or API Monitor based on our needs. Please head to the Introduction to Windows API room to learn more about API calls.

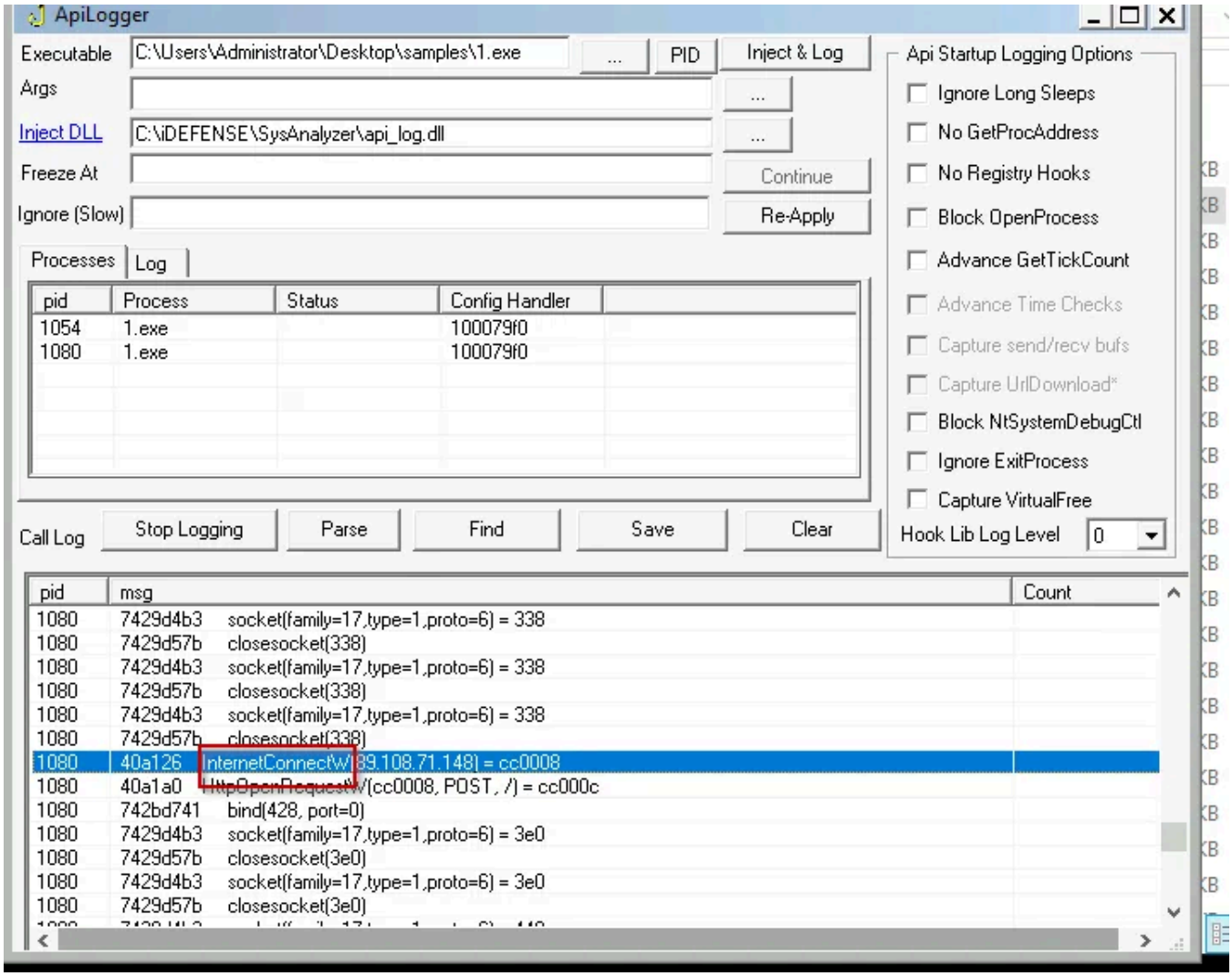
Answer to the questions of this section-

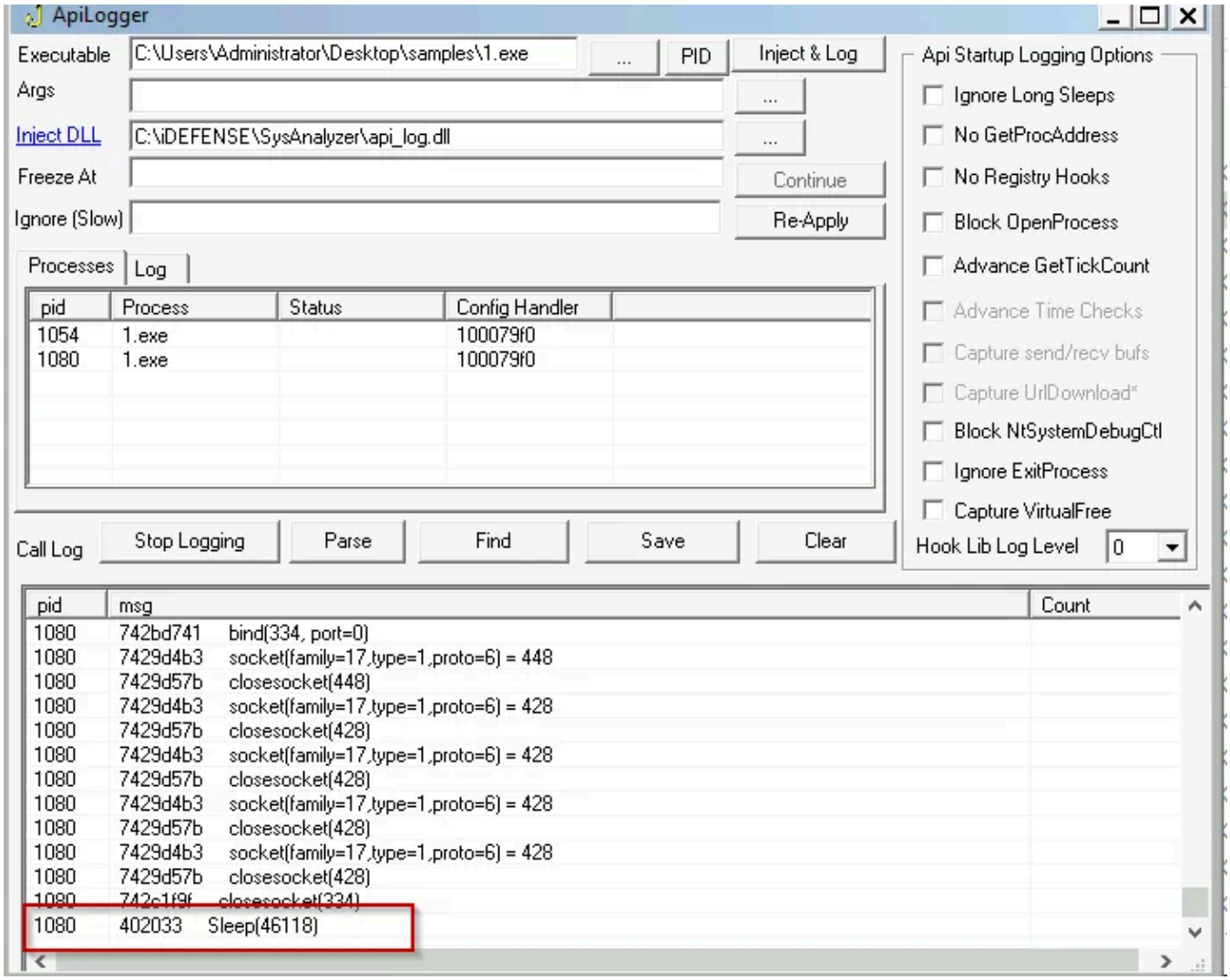


Answer:

1) Select executable “1.exe” and launch “Inject & Log”



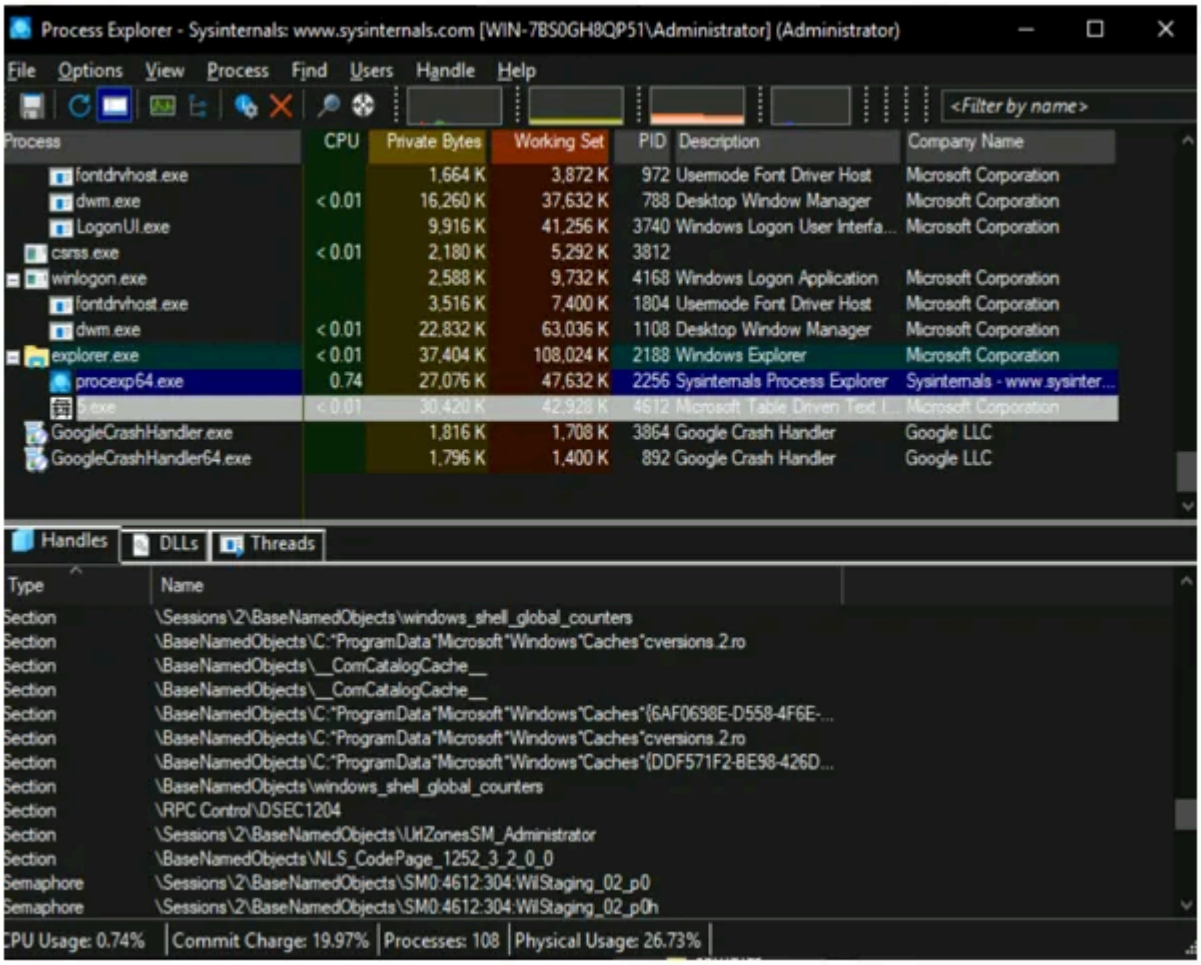

ApiLogger
Executable
C:\Users\Administrator\Desktop\samples\1.exe
PID
Inject & Log
Args
Inject DLL
C:\iDEFENSE\SysAnalyzer\api_log.dll
Freeze At
Continue
Ignore (Slow)
Re-Apply
Api Startup Logging Options
Ignore Long Sleeps
No GetProcAddress
No Registry Hooks
Block OpenProcess
Advance GetTickCount
Advance Time Checks
Capture send/recv bufs
Capture UrlDownload*
Block NtSystemDebugCtl
Ignore ExitProcess
Capture VirtualFree
Hook Lib Log Level
Processes
Log
pid Process Status Config Handler
1054 1.exe 100079f0
1080 1.exe 100079f0
Call Log
Stop Logging
Parse
Find
Save
Clear
pid msg Count
1080 7429d4b3 socket(family=17,type=1,proto=6) = 338
1080 7429d57b closesocket(338)
1080 7429d4b3 socket(family=17,type=1,proto=6) = 338
1080 7429d57b closesocket(338)
1080 7429d4b3 socket(family=17,type=1,proto=6) = 338
1080 7429d57b closesocket(338)
1080 40a126 InternetConnectW(
89.108.71.148) = cc0008
1080 40a1a0
(cc0008, POST, /) = cc000c
1080 742bd741 bind(428, port=0)
1080 7429d4b3 socket(family=17,type=1,proto=6) = 3e0
1080 7429d57b closesocket(3e0)
1080 7429d4b3 socket(family=17,type=1,proto=6) = 3e0
1080 7429d57b closesocket(3e0)



Task 5 Process Explorer –

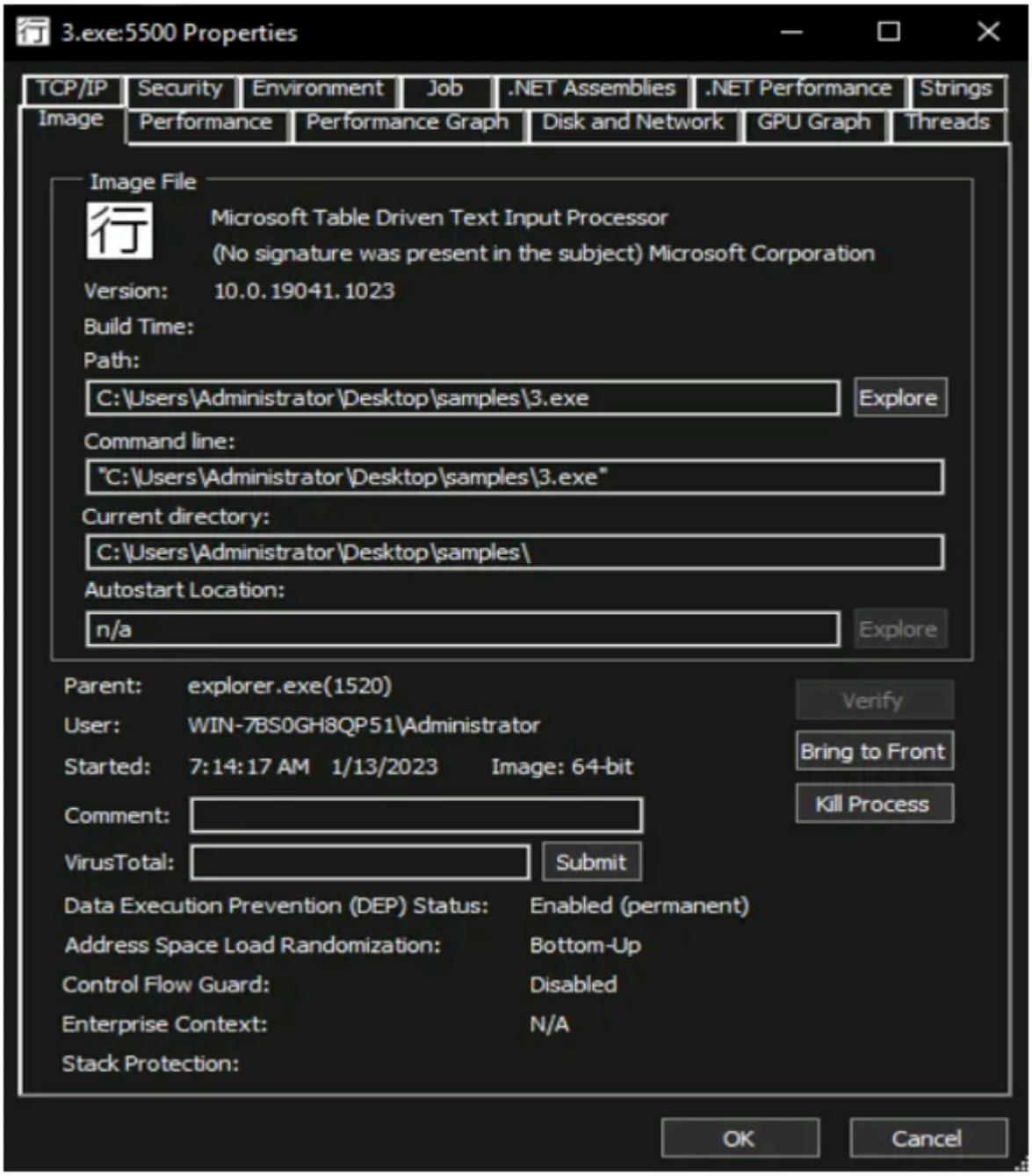
Process Explorer is another very useful tool from the Sysinternals Suite. It can be considered a more advanced form of the Windows Task Manager. Process Explorer is a very powerful tool that can help us identify process hollowing and masquerading techniques. We can open the Process Explorer tool by navigating to `~Desktop\Tools\Utilities\procexp.exe`. When we open Process Explorer, we see something like the below screenshot.

The above screenshot shows all the different processes running in the system in a tree format. We can also see their CPU utilization, memory usage, Process IDs (PIDs), Description, and Company name. We can enable the lower pane view from the ‘View’ menu to find more information about the processes. When enabled, we see the following screenshot.



When we select a process in the upper pane, we can see details about that process in the lower pane. Here, we see the Handles the process has opened for different Sections, Processes, Threads, Files, Mutexes, and Semaphores. Handles inform us about the resources being used in this process. If another process or a thread in another process is opened by a process, it can indicate code injection into that process. Similarly, we can see DLLs and Threads of the process in the other tabs of the lower pane.

For some more details about a selected process, we can look at the properties of the process. We can do that by right-clicking the process name in the process tree and selecting ‘Properties’. When we open the properties of a process, we see something like the below image.



Process Masquerading:

As seen in the above screenshot, the properties function shows us a lot of information about a process in its different tabs. Malware authors sometimes use process names similar to Windows processes or commonly used software to hide from an analyst's prying eyes. The 'Image' tab, as shown in the above screenshot, helps an analyst defeat this technique. By clicking the 'Verify' button on this tab, an analyst can identify if the executable for the running process is signed by the relevant organization, which will be Microsoft in the case of Windows binaries. In this particular screenshot, we can see that the Verify option has already been clicked. Furthermore, we can see the text '(No signature was present in the subject) Microsoft Corporation' at the top. This means that although the executable claims to

be from Microsoft, it is not digitally signed by Microsoft and is masquerading as a Microsoft process. This can be an indication of a malicious process.

We must note here that this verification process only applies to the Image of the process stored on the disk. If a signed process has been hollowed and its code has been replaced with malicious code in the memory, we might still get a verified signature for that process. To identify hollowed processes, we have to look somewhere else.

Process Hollowing:

Another technique used by malware to hide in plain sight is Process Hollowing. In this technique, the malware binary hollows an already running legitimate process by removing all its code from its memory and injecting malicious code in place of the legitimate code. This way, while an analyst sees a legitimate process, that process runs malicious code of the malware author. Process Explorer can help us identify this technique as well. When we open the ‘Strings’ tab in a process’s properties, we see something like the below screenshot.

At the bottom of the screenshot, we can see the options ‘Image’ and ‘Memory’. When we select ‘Image’, Process Explorer shows us strings present in the disk image of the process. When ‘Memory’ is selected, Process Explorer extracts strings from the process’s memory. In normal circumstances, the strings in the Image of a process will be similar to those in the Memory as the same process is loaded in the memory. However, if a process has been hollowed, we will see a significant difference between the strings in the Image and the process’s memory. Hence showing us that the process loaded in the memory is vastly different from the process stored on the disk.

Answer to the questions of this section-

What is the name of the first Mutex created by the sample ~Desktop\samples\1.exe? If there are numbers in the name of the Mutex, replace them with X.

\Sessions\2\BaseNamedObjects\SMX:XXXX:XXX:WilStaging_XX — Correct Answer — Hint

Is the file signed by a known organization? Answer with Y for Yes and N for No.

N — Correct Answer

Is the process in the memory the same as the process on disk? Answer with Y for Yes and N for No.

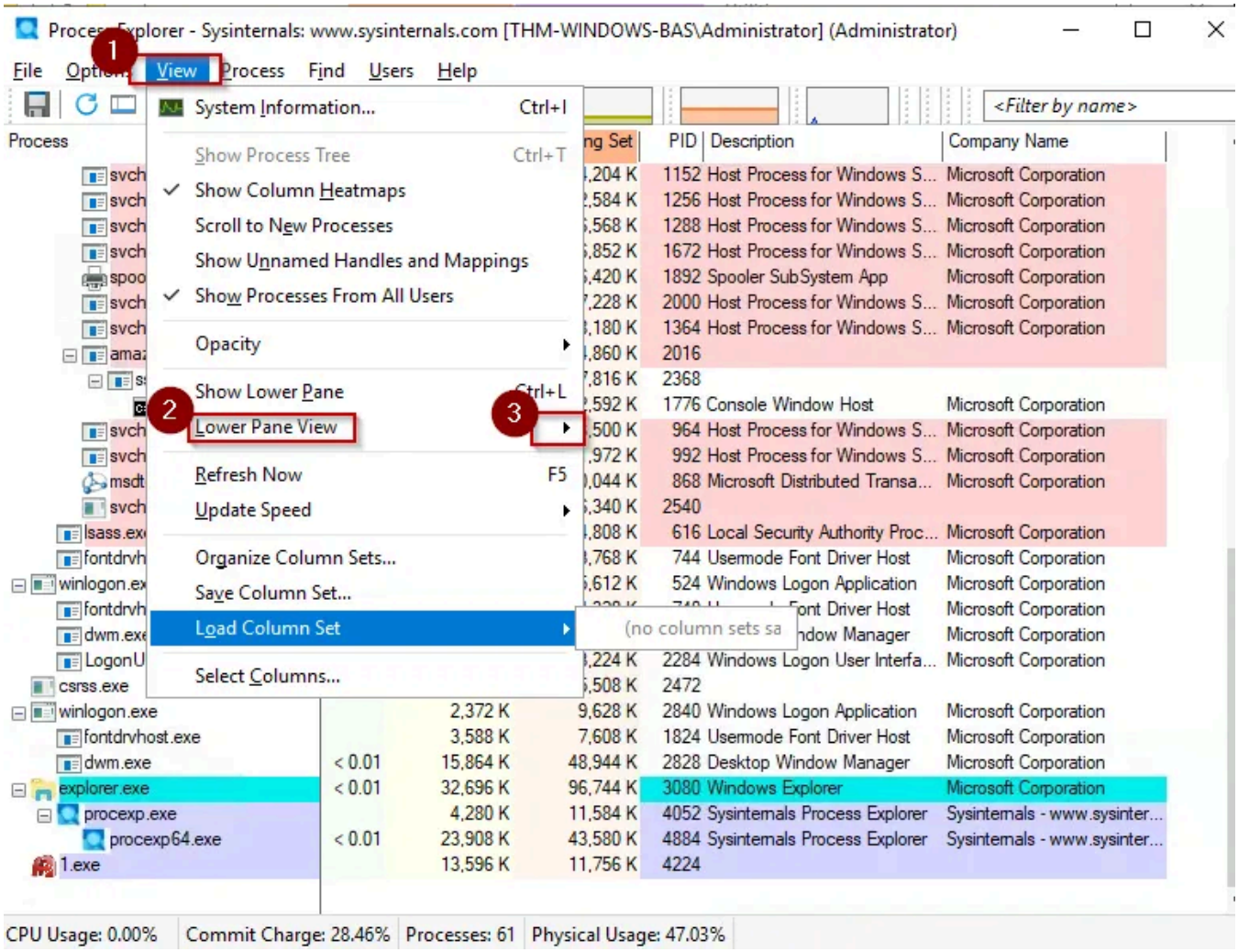
n — Correct Answer — Hint

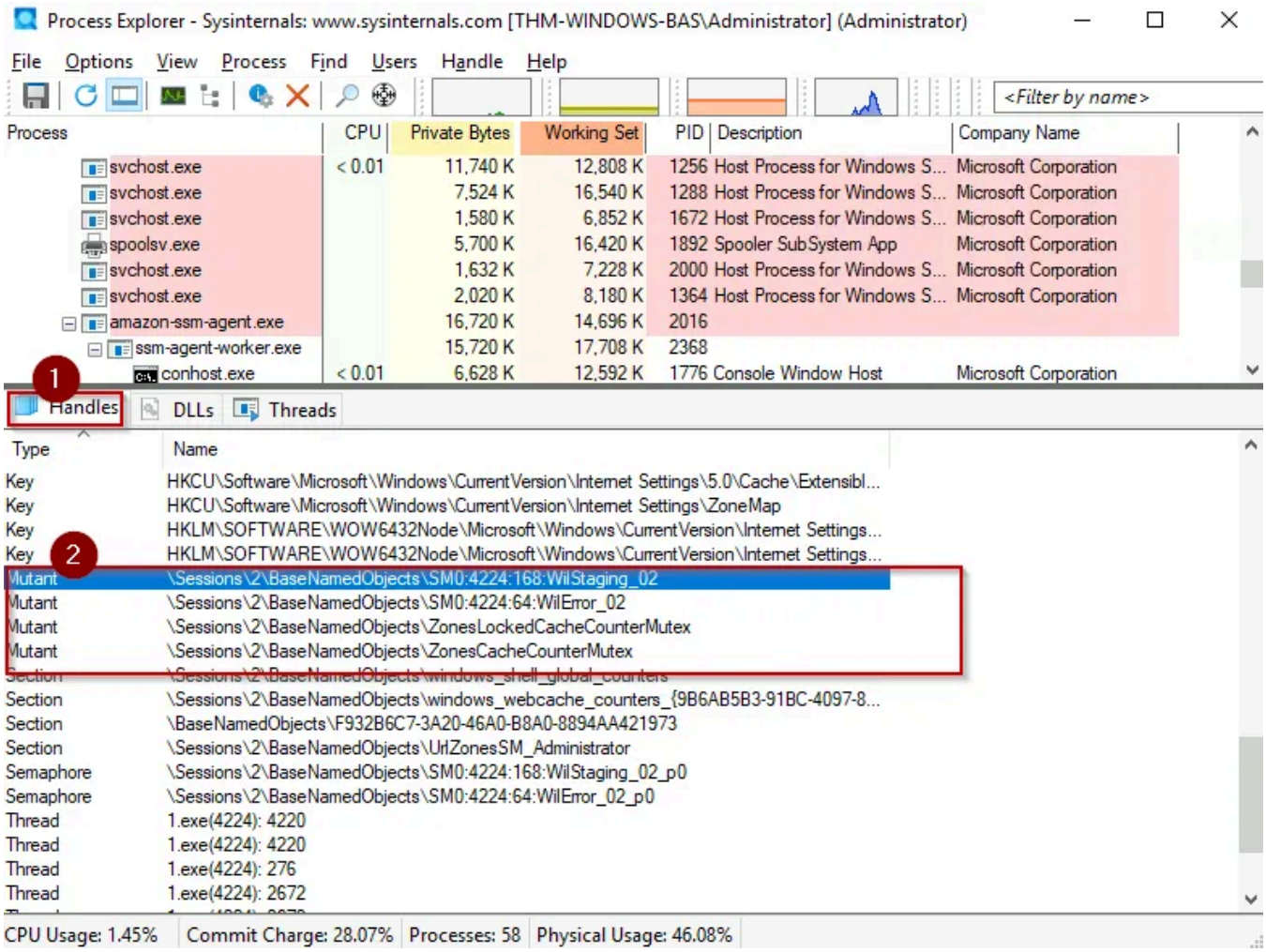
Before moving on to the next task, please terminate the VM and start it again to start fresh from the snapshot.

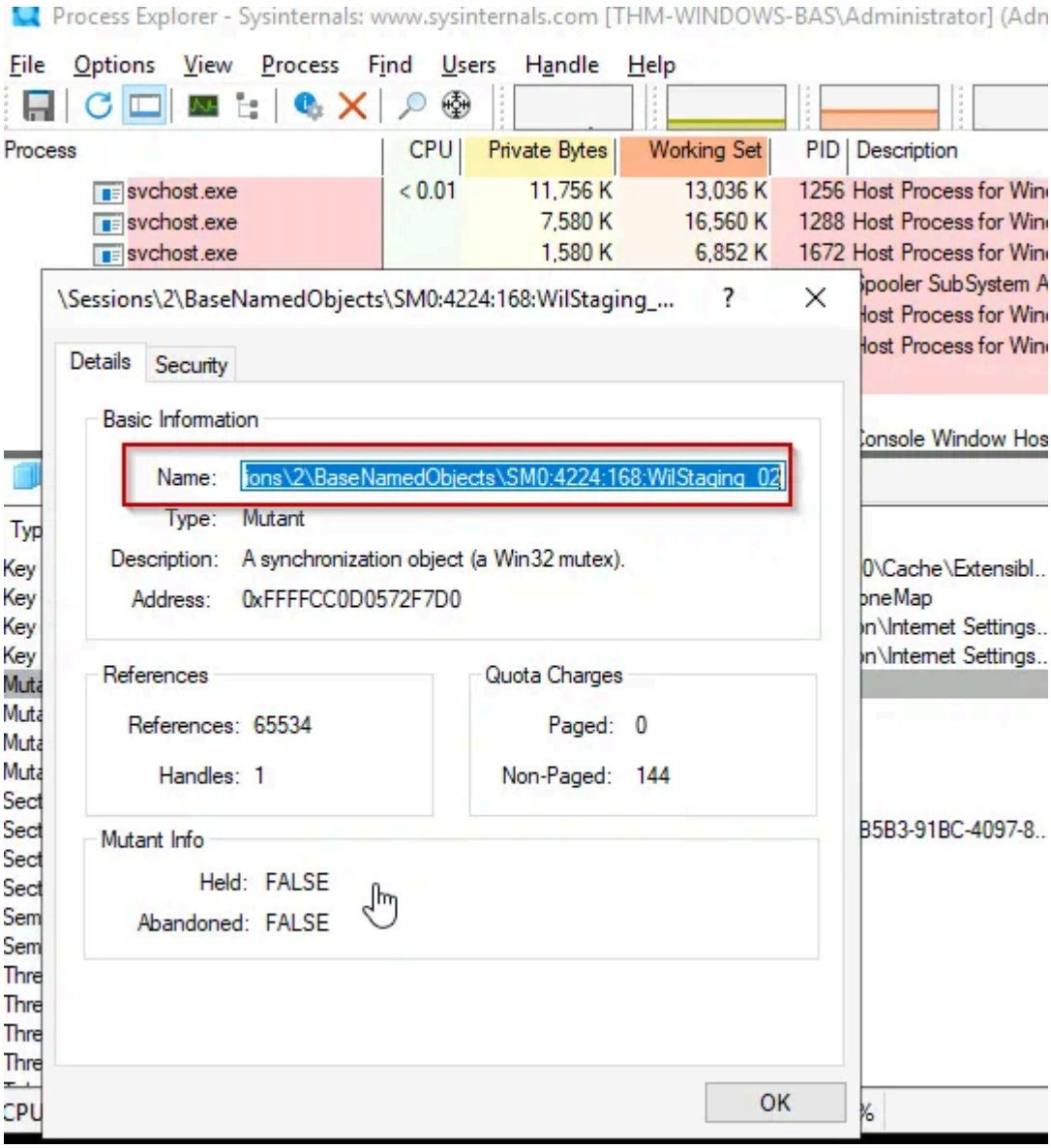
No answer needed — Correct Answer

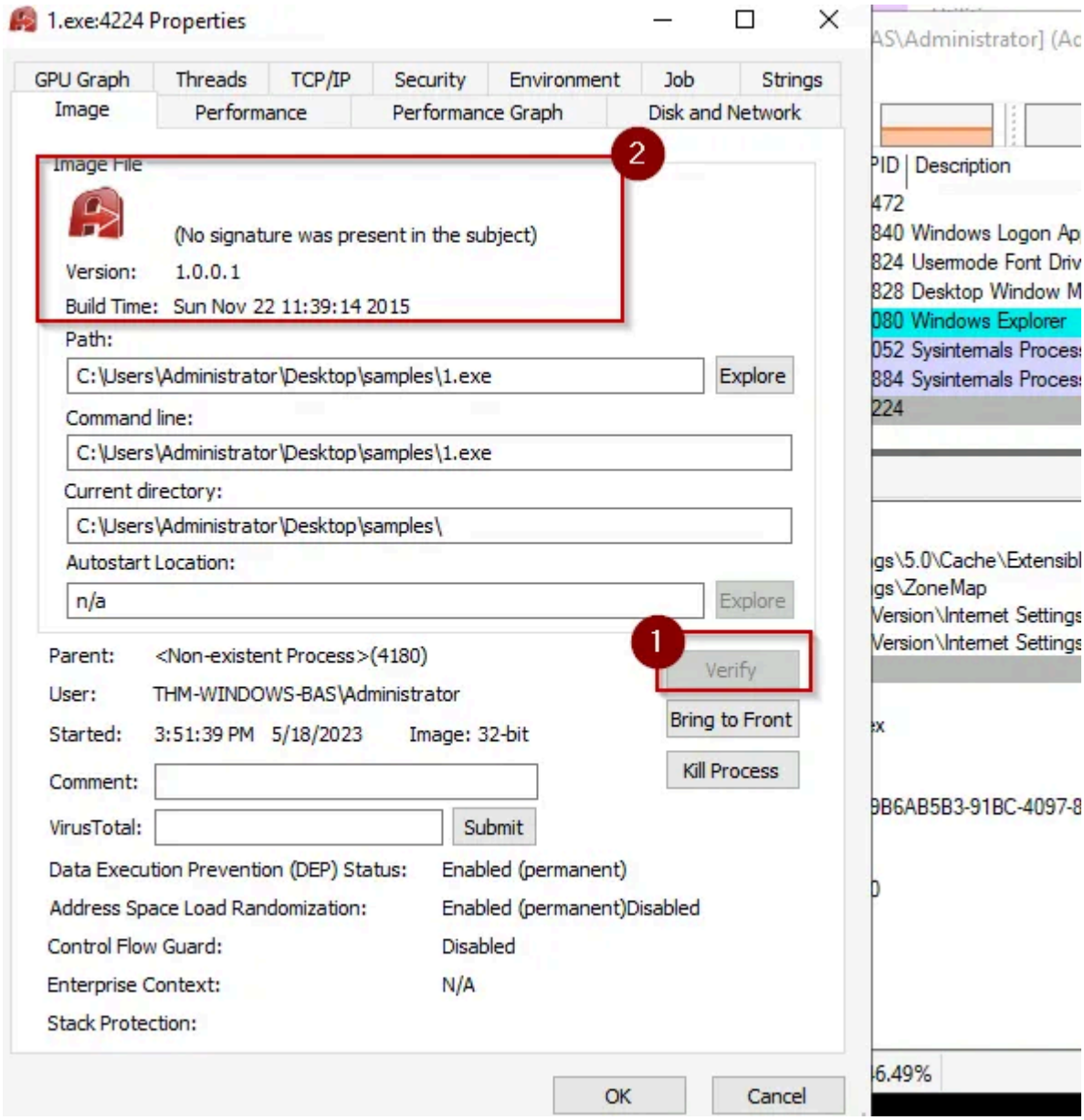
Answer:

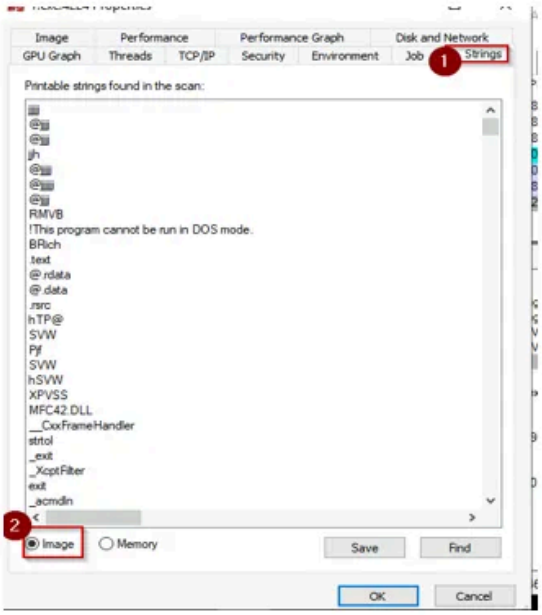
1. **Let executable “1.exe” keep running and launch “Process explorer”, see “Handle” to check mutex**

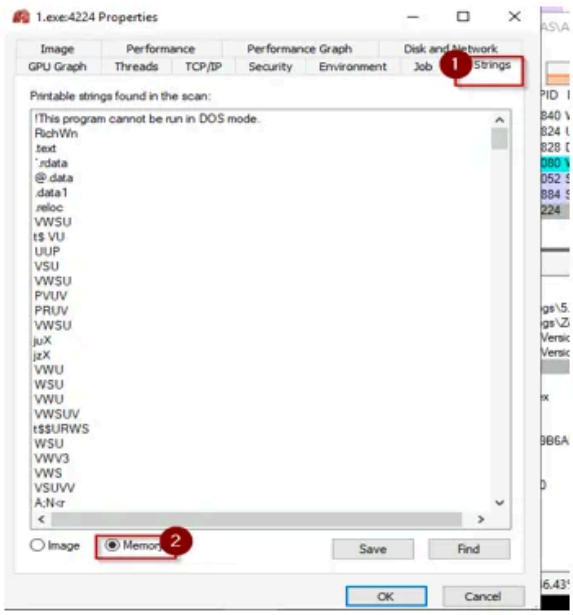



Process Explorer - Sysinternals: www.sysinternals.com [THM-WINDOWS-BAS\Administrator] (Administrator)
File Options View Process Find Users Handle Help
<Filter by name>
Process CPU Private Bytes Working Set PID Description Company Name
svchost.exe < 0.01 11,740 K 12,808 K 1256 Host Process for Windows S... Microsoft Corporation
svchost.exe 7,524 K 16,540 K 1288 Host Process for Windows S... Microsoft Corporation
svchost.exe 1,580 K 6,852 K 1672 Host Process for Windows S... Microsoft Corporation
spoolsv.exe 5,700 K 16,420 K 1892 Spooler SubSystem App Microsoft Corporation
svchost.exe 1,632 K 7,228 K 2000 Host Process for Windows S... Microsoft Corporation
svchost.exe 2,020 K 8,180 K 1364 Host Process for Windows S... Microsoft Corporation
amazon-ssm-agent.exe 16,720 K 14,696 K 2016
ssm-agent-worker.exe 15,720 K 17,708 K 2368
conhost.exe < 0.01 6,628 K 12,592 K 1776 Console Window Host Microsoft Corporation
1
Handles DLLs Threads
Type Name
Key HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensibl...
Key HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap
Key HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Internet Settings...
Key HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Internet Settings...
2
Mutant \Sessions\2\BaseNamedObjects\SM0:4224:168:WilStaging_02
Mutant \Sessions\2\BaseNamedObjects\SM0:4224:64:WilError_02
Mutant \Sessions\2\BaseNamedObjects\ZonesLockedCacheCounterMutex
Mutant \Sessions\2\BaseNamedObjects\ZonesCacheCounterMutex
Section \Sessions\2\BaseNamedObjects\windows_webcache_counters_{9B6AB5B3-91BC-4097-8...
Section \BaseNamedObjects\F932B6C7-3A20-46A0-B8A0-8894AA421973
Section \Sessions\2\BaseNamedObjects\UrlZonesSM_Administrator
Semaphore \Sessions\2\BaseNamedObjects\SM0:4224:168:WilStaging_02_p0
Semaphore \Sessions\2\BaseNamedObjects\SM0:4224:64:WilError_02_p0
Thread 1.exe(4224): 4220
Thread 1.exe(4224): 4220
Thread 1.exe(4224): 276
Thread 1.exe(4224): 2672
CPU Usage: 1.45% Commit Charge: 28.07% Processes: 58 Physical Usage: 46.08%


Process Explorer - Sysinternals: www.sysinternals.com [THM-WINDOWS-BAS\Administrator] (Adn
File Options View Process Find Users Handle Help
Process CPU Private Bytes Working Set PID Description
svchost.exe < 0.01 11,756 K 13,036 K 1256 Host Process for Win
svchost.exe 7,580 K 16,560 K 1288 Host Process for Win
svchost.exe 1,580 K 6,852 K 1672 Host Process for Win
\Sessions\2\BaseNamedObjects\SM0:4224:168:WilStaging_...
Details Security
Basic Information
Name: ions\2\BaseNamedObjects\SM0:4224:168:WilStaging_02
Type: Mutant
Description: A synchronization object (a Win32 mutex).
Address: 0xFFFFCC0D0572F7D0
References
References: 65534
Handles: 1
Quota Charges
Paged: 0
Non-Paged: 144
Mutant Info
Held: FALSE
Abandoned: FALSE
OK





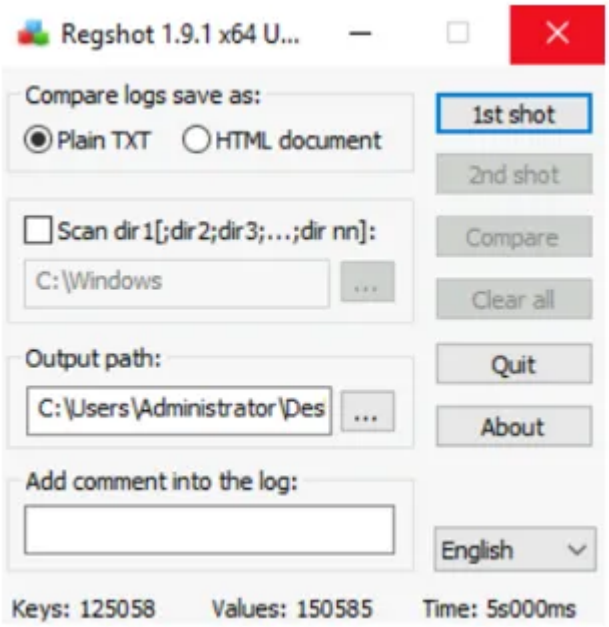


Task 6 Regshot –

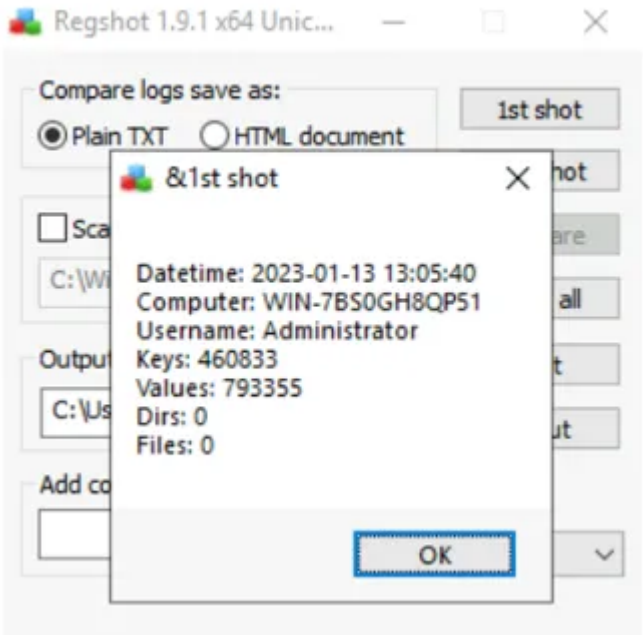
Before starting this task, please terminate the VM and restart it for a fresh start from the snapshot.

Regshot is a tool that identifies any changes to the registry (or the file system we select). It can be used to identify what registry keys were created, deleted, or modified during our dynamic analysis by malware. Regshot works by taking snapshots of the registry before and after the execution of malware and then comparing the two snapshots to identify the differences between the two. To execute Regshot in the attached VM, navigate to `~Desktop\Tools\Utilities\Regshot-x64-Unicode.exe`

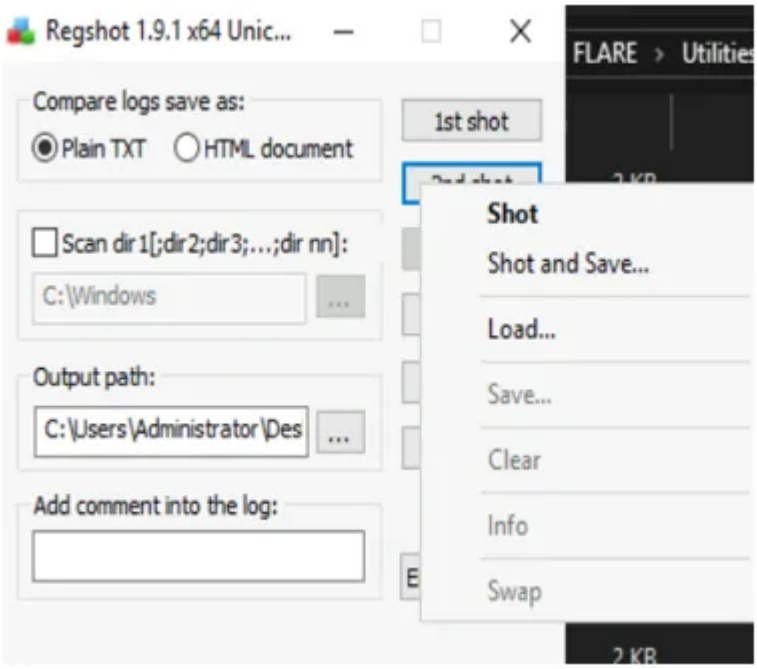
When we execute Regshot, we see the following interface. Please note that the Output path we see in the attached VM might differ.



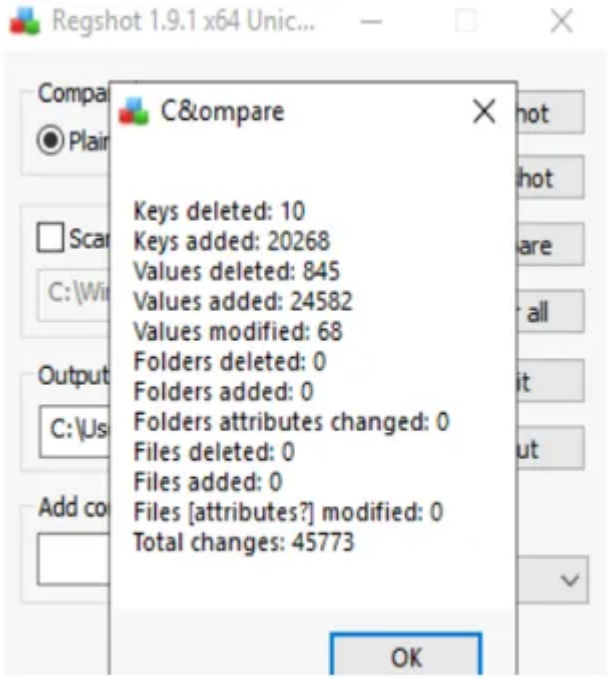
In this simple interface, if we select the Scan dir1 option, we can also scan for changes to the file system. However, for the sake of brevity, we will only cover registry changes in this room. To start, we can click on the '1st shot' option. It will ask us whether to take a shot or take a shot and save. Once the 1st shot is taken, we see something like the below screenshot.



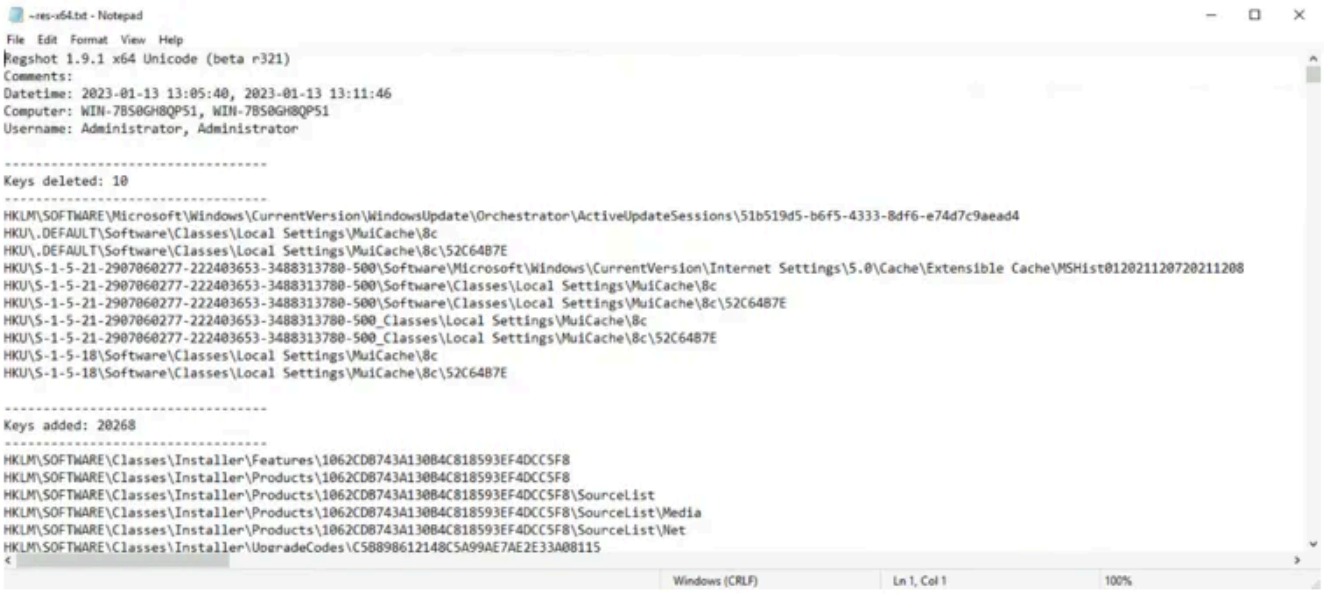
Now that we have saved a shot of the registry, we can execute the malware. Once we have executed the malware and are confident that it has performed its malicious activity, we take a 2nd shot. For this, we click the '2nd shot' option.



Now that we have both shots, we can compare them to identify the registry changes performed by the malware. We do that by clicking the 'Compare' option. We will see a summary that looks something like the below screenshot.



Notice that it shows Keys and Values that were added, deleted, and modified. It also shows changes to Files and Folders. We see zero changes to Folders and Files because we had disabled 'Scan dir1' while taking the shots. If we had enabled this option and provided directories to monitor, we would have seen details about filesystem changes made by the malware in our selected directories. For now, let's move on to the results of our execution. If we save the results by clicking on Compare > Output, Regshot provides us with the changes in the registry, as shown in the screenshot below.



Here we see the Date and time of the shots taken by Regshot, the computer name, the Username, and the version of Regshot. Below that, we can see a list of changes

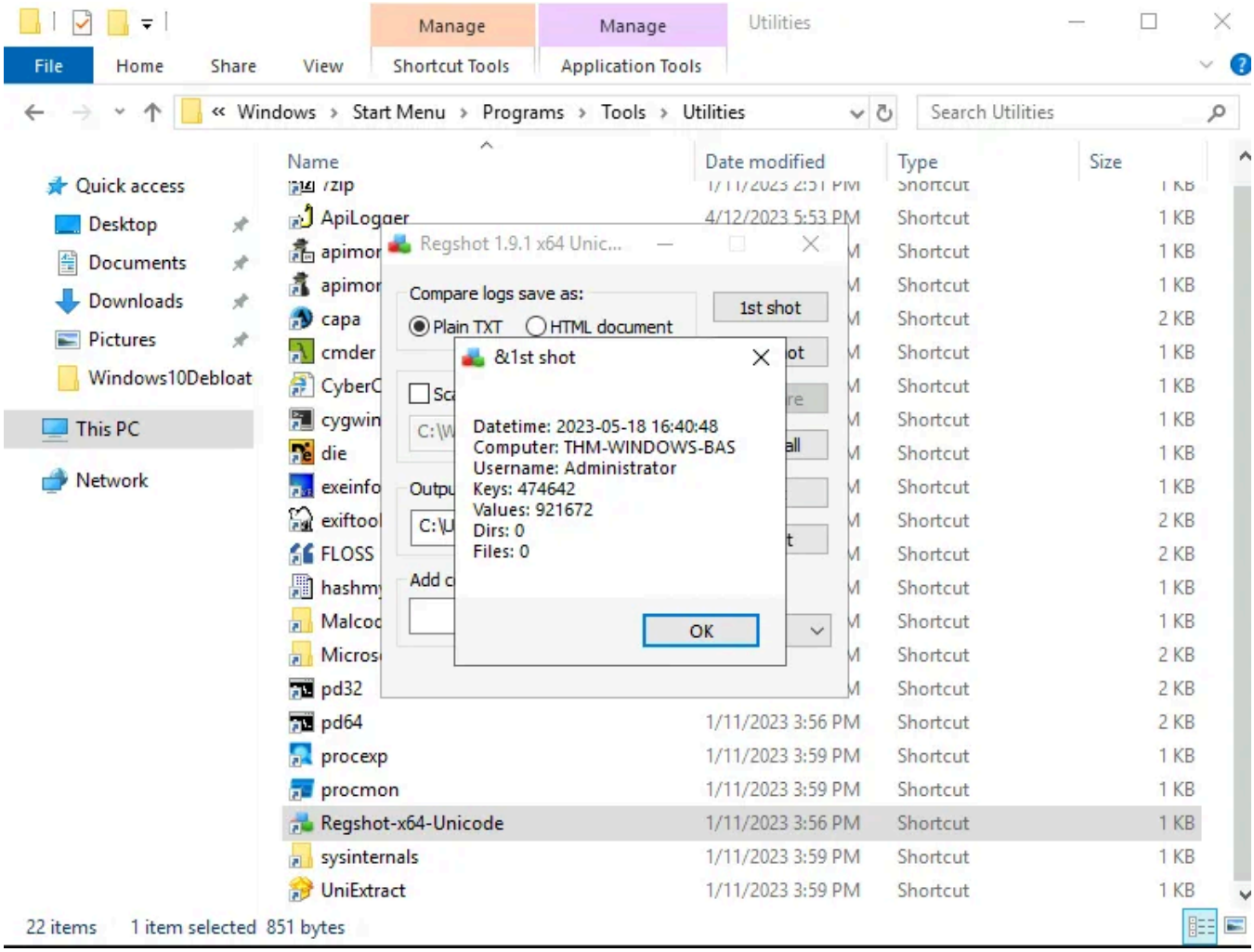
that were made to the registry, starting from Keys deleted.>

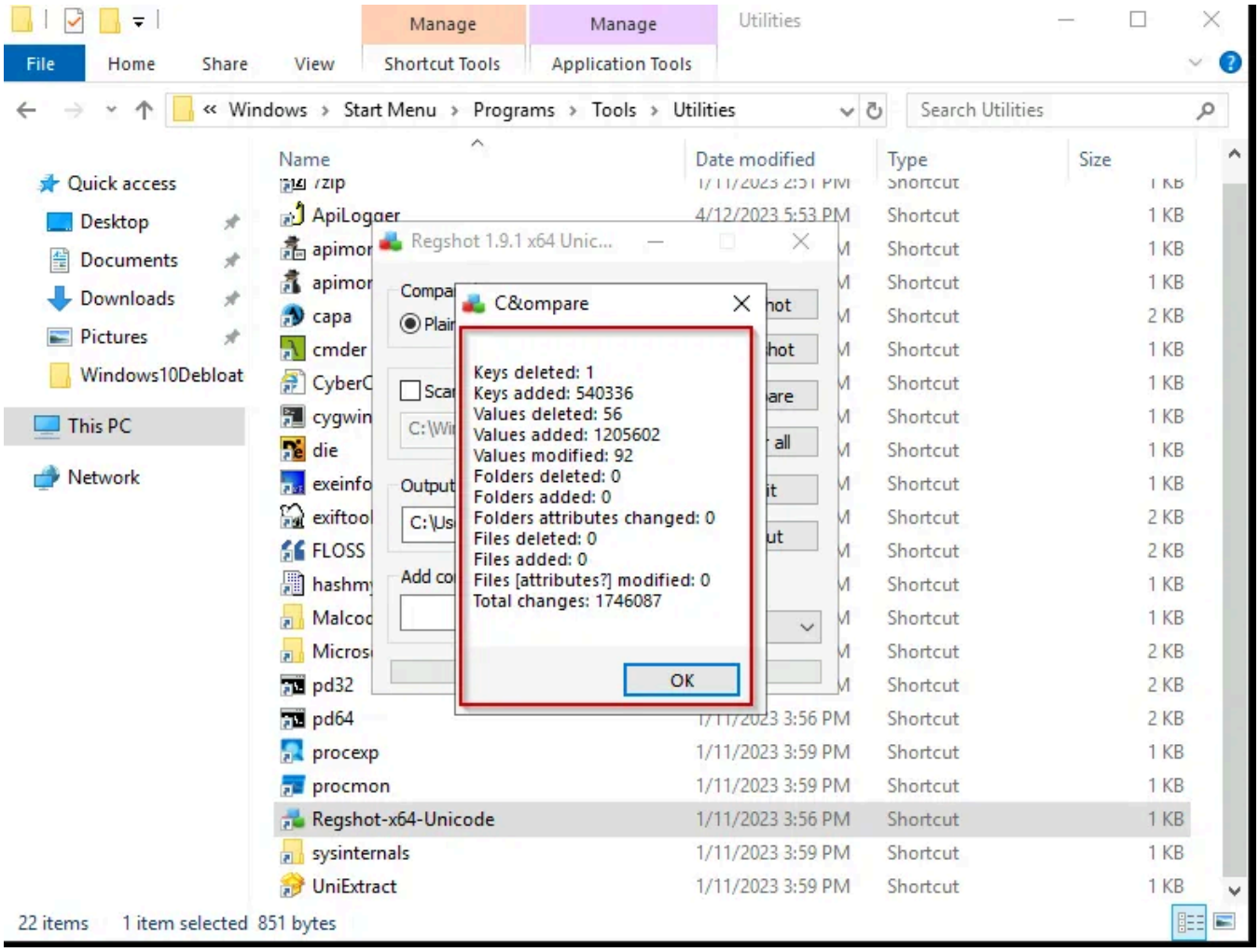
One advantage that Regshot enjoys over all the other tools discussed in this room is that it does not need to be running when we execute the malware. Some malware can check all the running processes and shut down if any analysis tool is running. When analyzing, we might often encounter malware samples that check for ProcExp, ProcMon, or API Monitor before performing any malicious activity and quitting if these processes are found. Therefore, these samples might thwart our analysis efforts. However, since Regshot takes a shot before and after the execution of the malware sample, it does not need to be running during malware execution, making it immune to this technique of detection evasion. On the flip side, we must ensure that no other process is running in the background while performing analysis with Regshot, as there is no filtering mechanism in Regshot, as we saw in the other tools. Hence, any noise created by background processes will also be recorded by Regshot, resulting in False Positives.

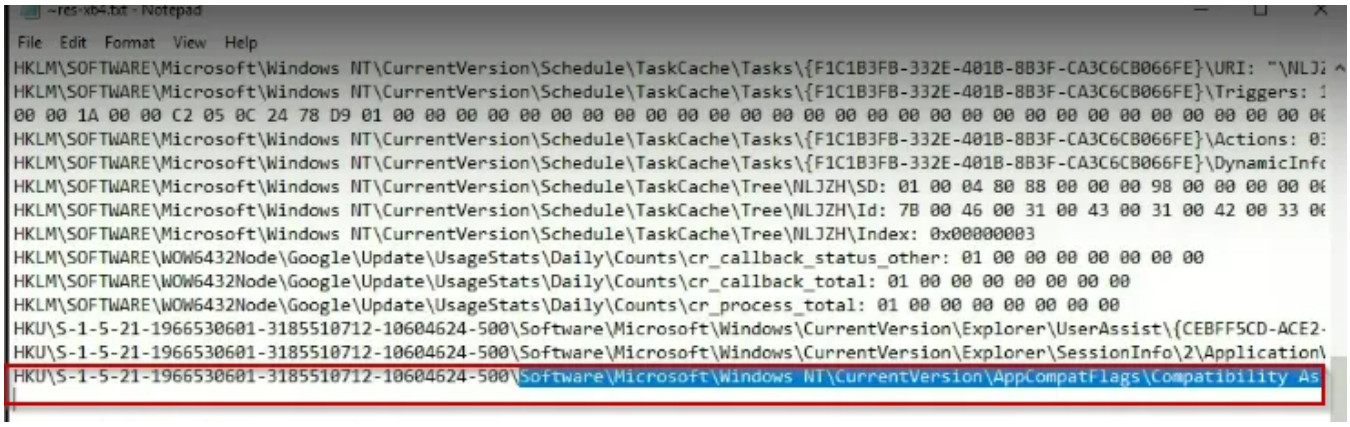
Answer to the questions of this section-

Answer:

1. **Take 1st Regshot and then Launch executable “3.exe” and Take 2nd Regshot and compare the output.**


Manage
Manage
Utilities
File
Home
Share
View
Shortcut Tools
Application Tools
« Windows › Start Menu › Programs › Tools › Utilities
Search Utilities
Quick access
Desktop
Documents
Downloads
Pictures
Windows10Debloat
This PC
Network
Name
Date modified
Type
Size
ApiLogger 4/12/2023 5:53 PM Shortcut 1 KB
Regshot 1.9.1 x64 Unic...
Compare logs save as:
Plain TXT
HTML document
1st shot
&1st shot
Datetime: 2023-05-18 16:40:48
Computer: THM-WINDOWS-BAS
Username: Administrator
Keys: 474642
Values: 921672
Dirs: 0
Files: 0
OK
pd64 1/11/2023 3:56 PM Shortcut 2 KB
procexp 1/11/2023 3:59 PM Shortcut 1 KB
procmon 1/11/2023 3:59 PM Shortcut 1 KB
Regshot-x64-Unicode 1/11/2023 3:56 PM Shortcut 1 KB
sysinternals 1/11/2023 3:59 PM Shortcut 1 KB
UniExtract 1/11/2023 3:59 PM Shortcut 1 KB
22 items
1 item selected 851 bytes





Have Fun and Enjoy Hacking! Do visit other rooms and modules on TryHackMe for more learning.

-by Shefali Kumai

For more cyber security learning follow me here-

https://www.youtube.com/channel/UCf-F-eATCUXYaUVk8Xl7OOQ

https://www.instagram.com/cybersecurity.cyber_seek/

Dynamic Analysis | Malware Analysis | Tryhackme Writeup | Sysinternals | Windows Internals

Following

Written by Shefali Kumari

380 Followers · 17 Following

Love Learning about Malware analysis, Threat hunting, Network Security and Incident Response Management professionally | https://youtube.com/channel/UCf-F-eATCU

Responses (1)

What are your thoughts?

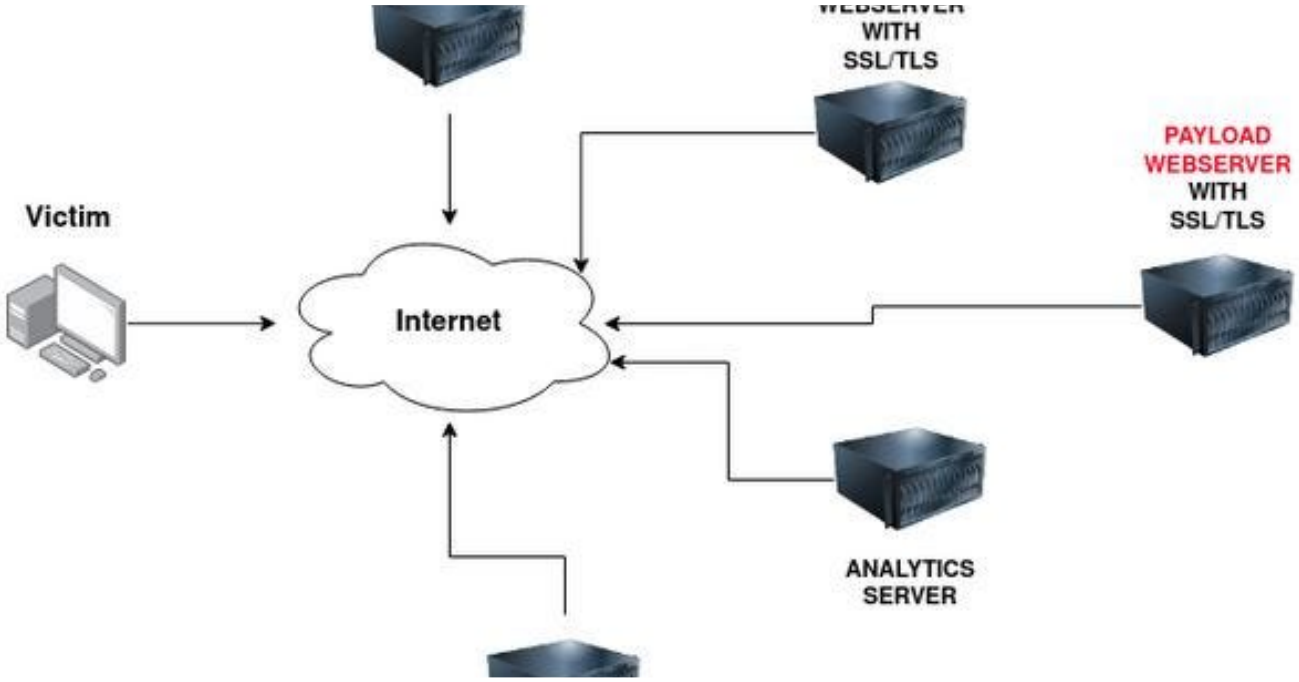
Respond

V spark
10 months ago

thanks you mam it was really a great help, i really really thank you mam

Reply

More from Shefali Kumari

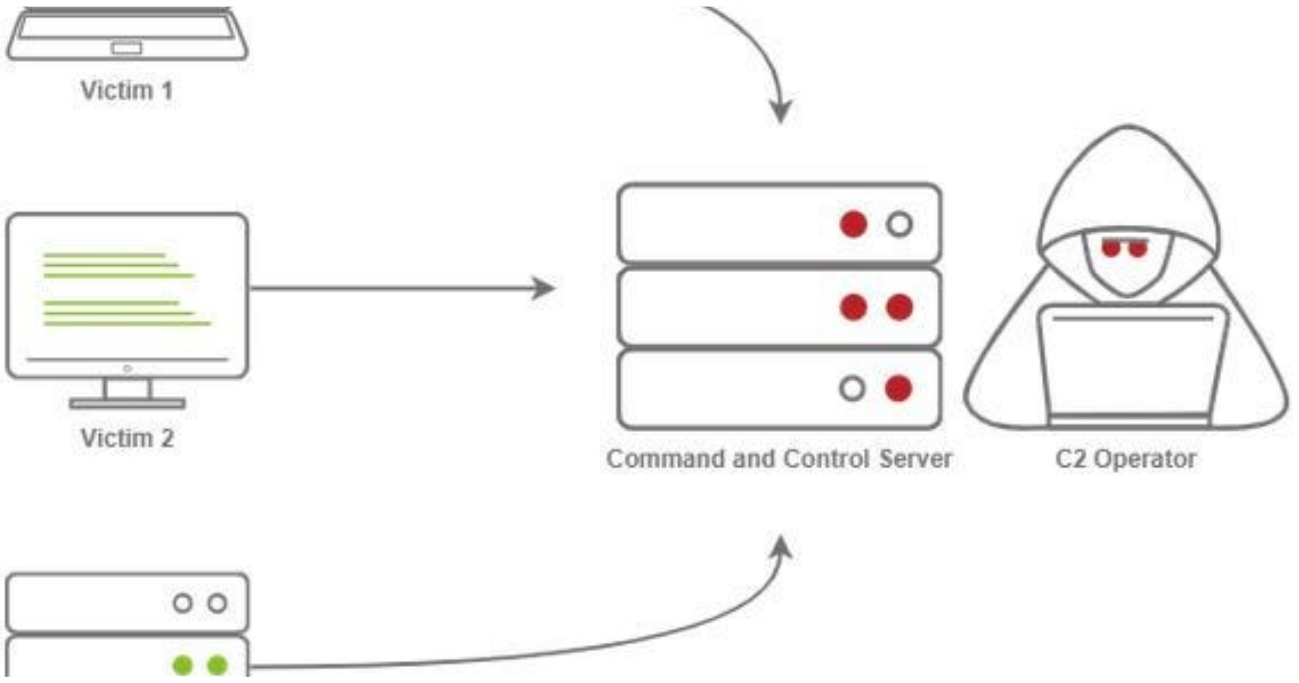


Shefali Kumari

TRY HACK ME: Write-Up Phishing

Task 2 Intro To Phishing Attacks -

Nov 12, 2021 12



Shefali Kumari

TRY HACK ME: Intro to C2 Write-Up

Task 1 Introduction -

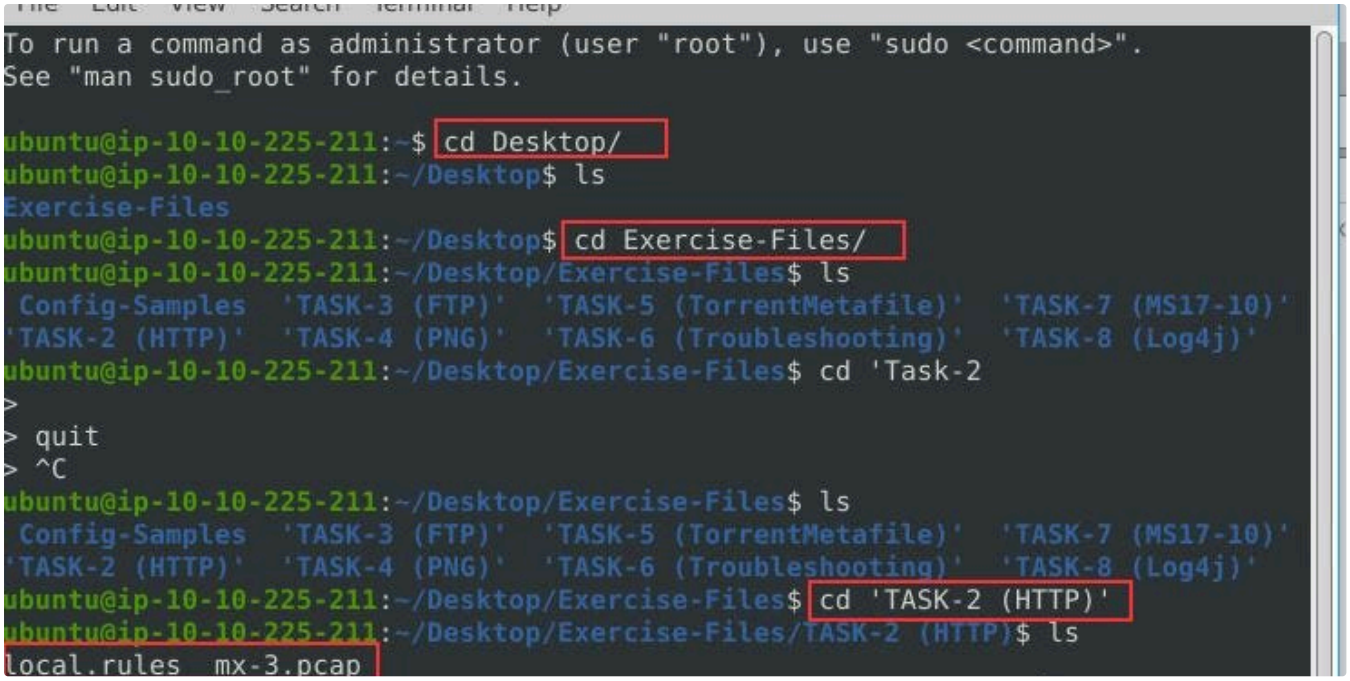
Mar 14, 2022 54

Shefali Kumari

TRY HACK ME: Write-Up Module-Vulnerability Research: Exploit Vulnerabilities

TASK 1: INTRODUCTION –

Oct 13, 2021 55 2



Shefali Kumari

TRY HACK ME: Snort Challenge-The Basics Write-Up

Task 1 Introduction-

Apr 24, 2022 5

See all from Shefali Kumari

Recommended from Medium

MAGESH

SigHunt-Tryhackme Writeup

You are tasked to create detection rules based on a new threat intel.

Oct 15, 2024

Trnty

TryHackMe | Introduction To Honeypots Walkthrough

A guided room covering the deployment of honeypots and analysis of botnet activities

Sep 7, 2024 10

Lists

Staff picks

793 stories · 1549 saves

Stories to Help You Level-Up at Work

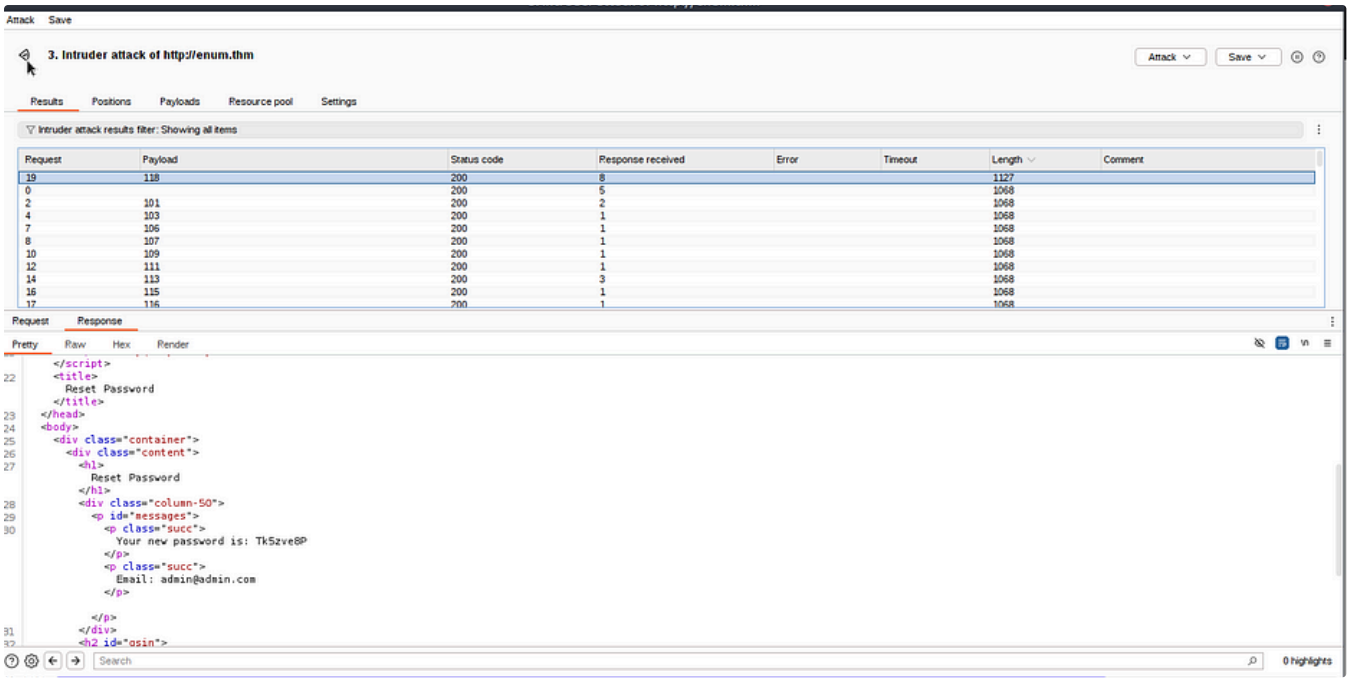
19 stories · 909 saves

Self-Improvement 101

20 stories · 3184 saves

Productivity 101

20 stories · 2697 saves



embossdotar

TryHackMe—Enumeration & Brute Force—Writeup

Key points: Enumeration | Brute Force | Exploring Authentication Mechanisms | Common Places to Enumerate | Verbose Errors | Password Reset...

Jul 31, 2024 26

Abhijeet Singh

Advent of Cyber 2024 [Day 3] Even if I wanted to go, their vulnerabilities wouldn't allow it.

So, Let's Start with the Questions. I hope you already read the story and all the given instructions —

Dec 4, 2024 2

In T3CH by Axoloth

TryHackMe | FlareVM: Arsenal of Tools| WriteUp

Learn the arsenal of investigative tools in FlareVM

Nov 28, 2024 50

Abhijeet Singh

Advent of Cyber 2024 [Day 4] I'm all atomic inside! | TryHackMe Walkthrough

Please go through the story, Cyber Attacks, the Kill Chain and MITRE ATT&CK related content for better understanding of this room.

Dec 5, 2024 1