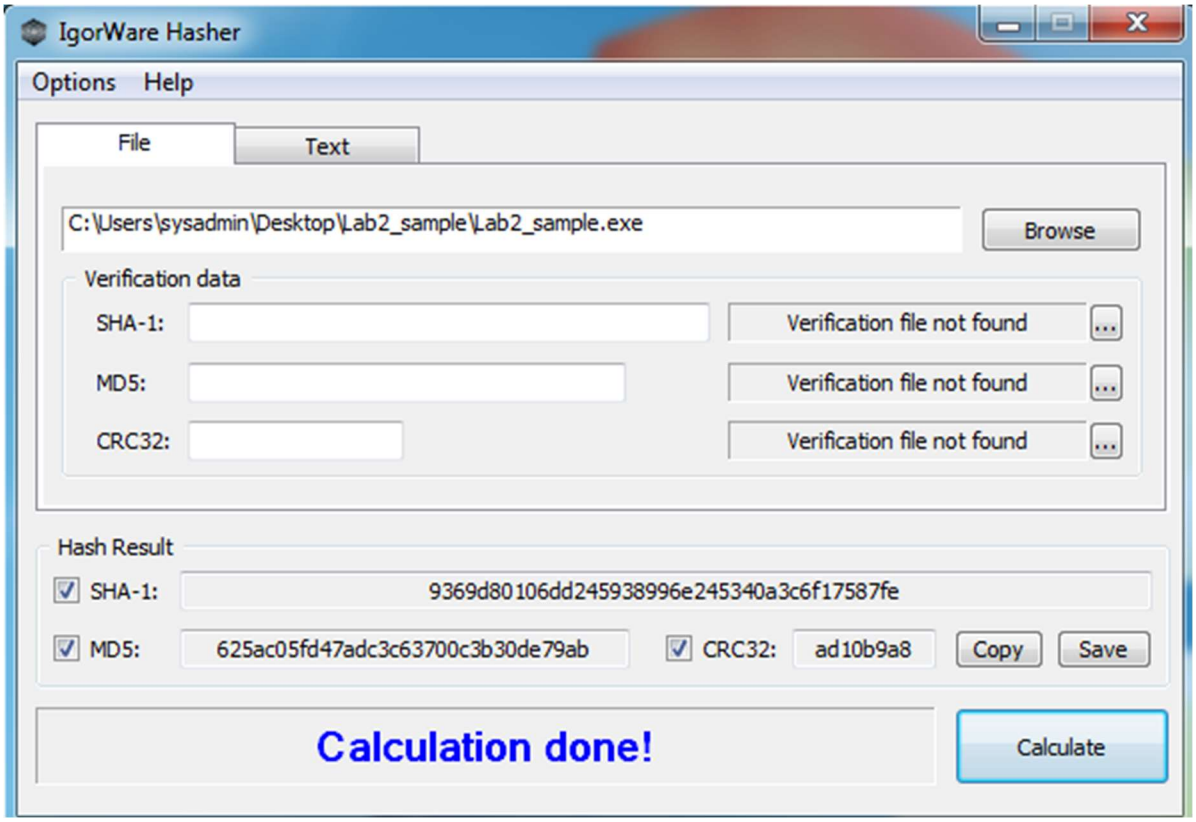
Lab 2 – Dynamic Analysis
Gunnar Yonker

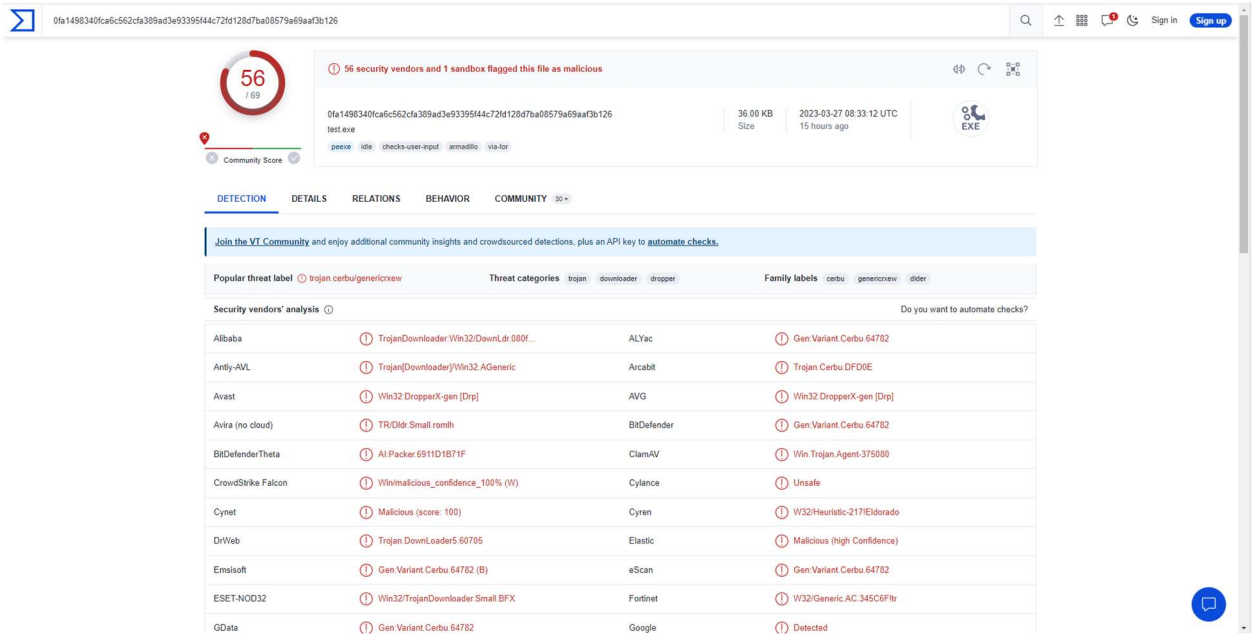
Static Analysis – Preliminary Information

IgorWare Hasher



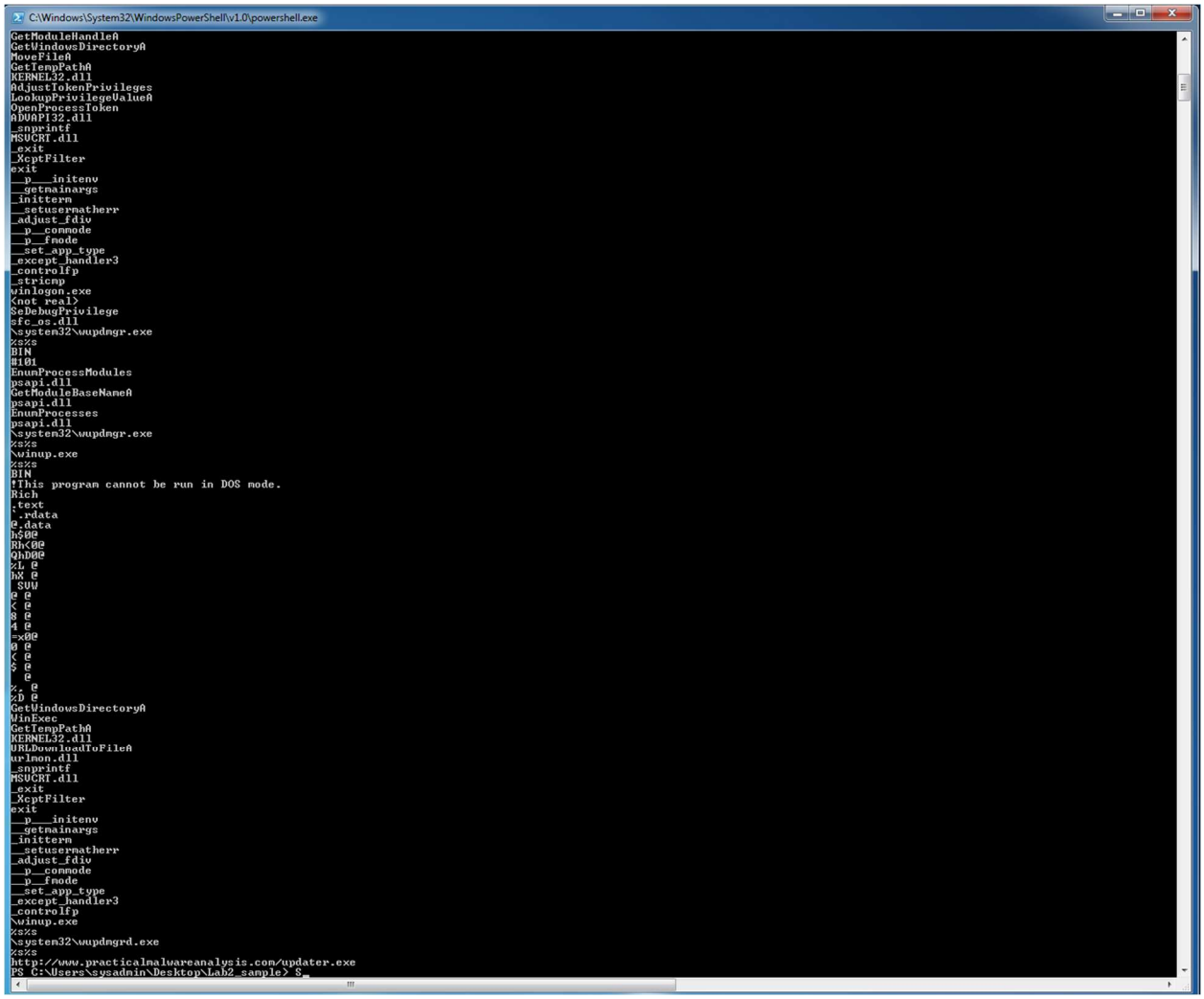
SHA-1: 9369d80106dd245938996e245340a3c6f17587fe

VirusTotal



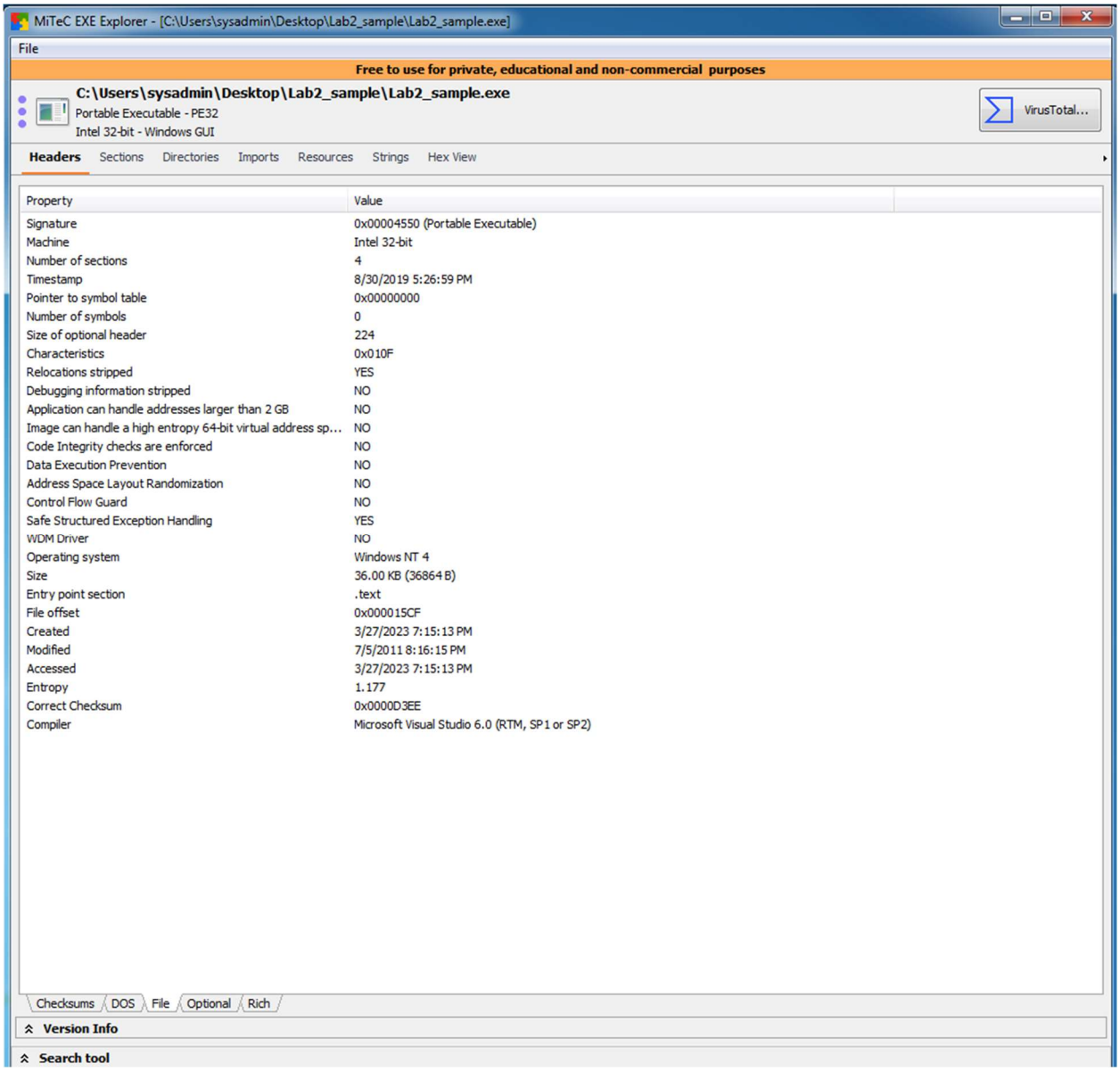
Lab 2 – Dynamic Analysis
Gunnar Yonker

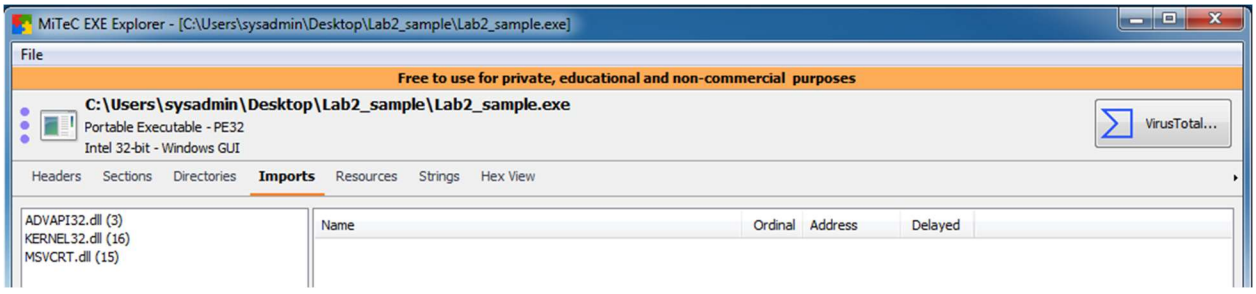
strings Utility



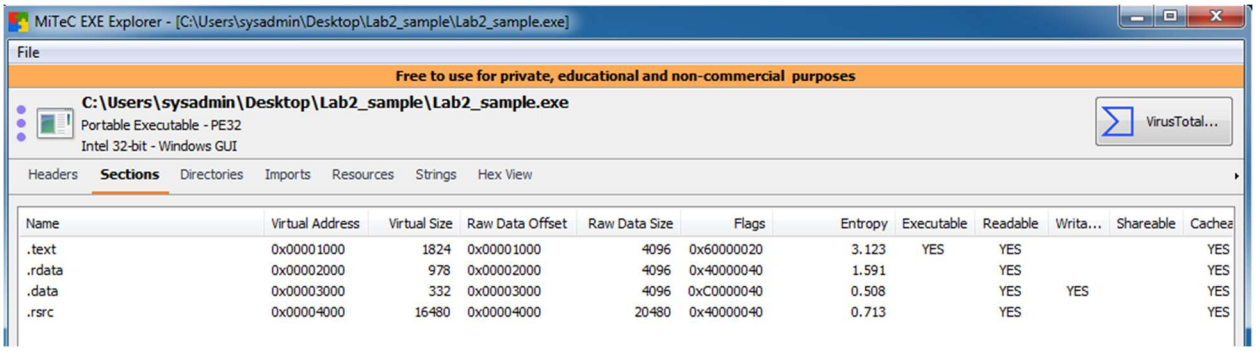
Lab 2 – Dynamic Analysis
Gunnar Yonker

MiTec EXE Explorer



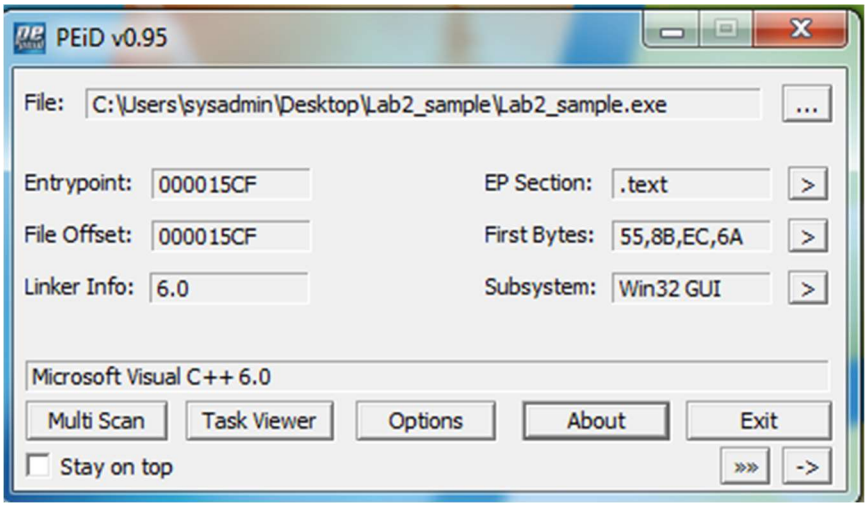


Lab 2 – Dynamic Analysis
Gunnar Yonker



Name	Virtual Address	Virtual Size	Raw Data Offset	Raw Data Size	Flags	Entropy	Executable	Readable	Writa...	Shareable	Cachea
.text	0x00001000	1824	0x00001000	4096	0x60000020	3.123	YES	YES			YES
.rdata	0x00002000	978	0x00002000	4096	0x40000040	1.591		YES			YES
.data	0x00003000	332	0x00003000	4096	0xC0000040	0.508		YES	YES		YES
.rsrc	0x00004000	16480	0x00004000	20480	0x40000040	0.713		YES			YES

PEiD



Lab 2 – Dynamic Analysis
Gunnar Yonker

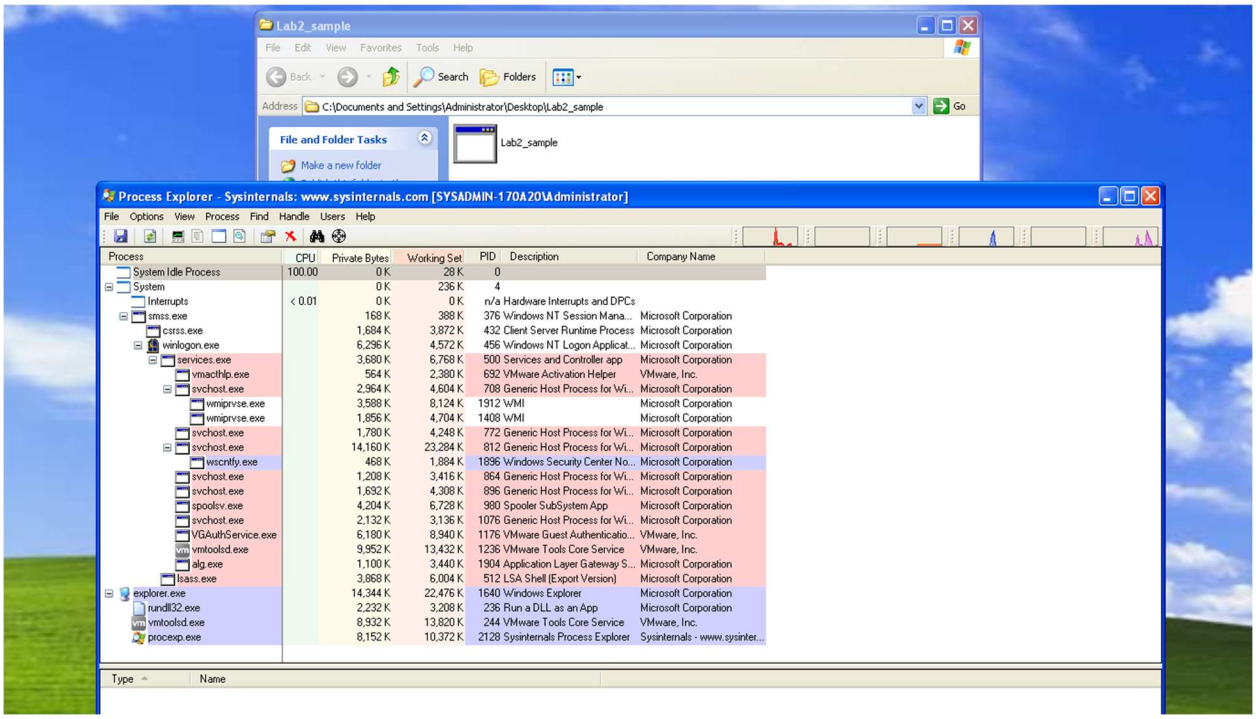
Lab 2 Dynamic Analysis Tools:

Tool: Process Explorer

Purpose of Tool: The Process Explorer tool allows the users to view detailed information about running processes, including the ability to see which processes have loaded specific DLLs. It also gives the user information on which processes have opened certain files or registry keys and also gives the user the ability to terminate processes. It is a more powerful version of the Windows task manager giving the user information such as DLLs loaded, child processes spawned, TCP/IP ports used by the process, resource consumption and signature verification.

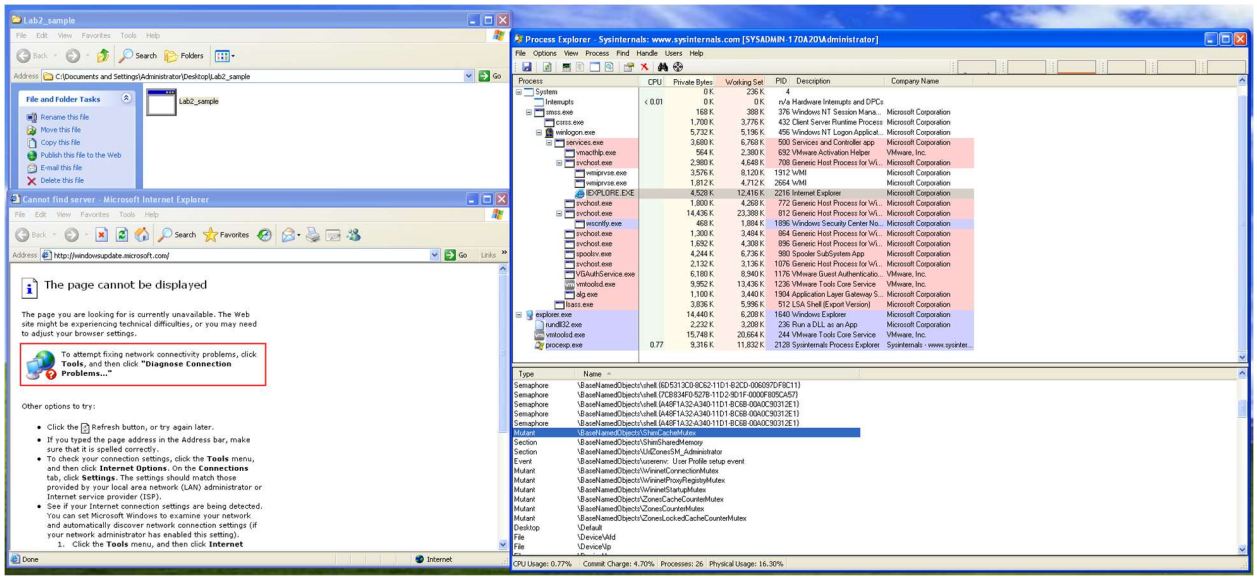
Relevant information from tool:

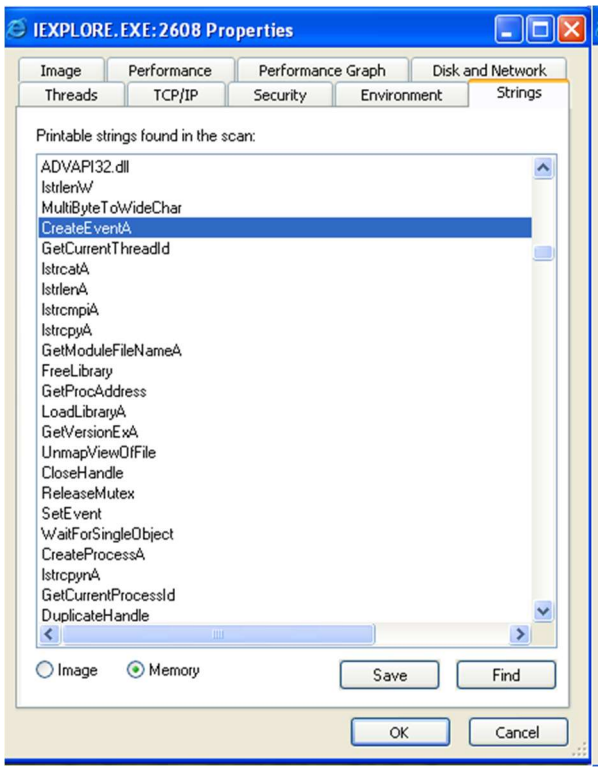
Before malware:

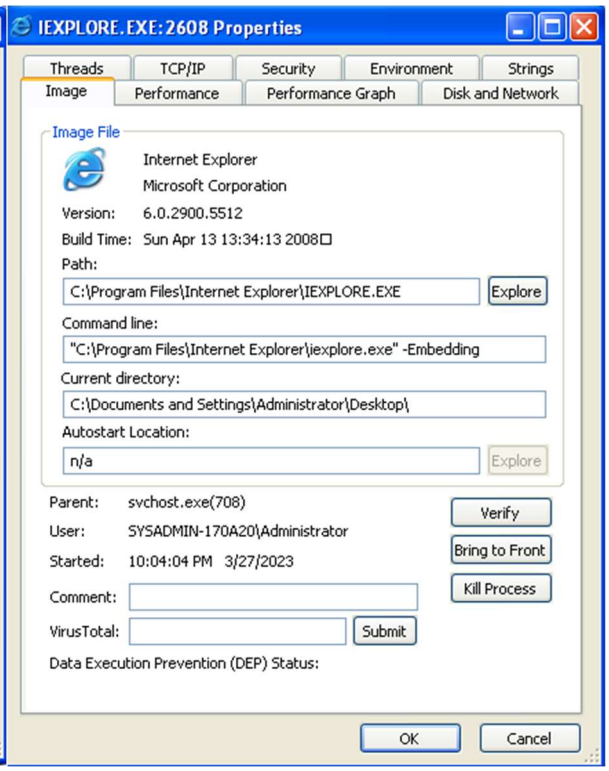


Lab 2 – Dynamic Analysis
Gunnar Yonker

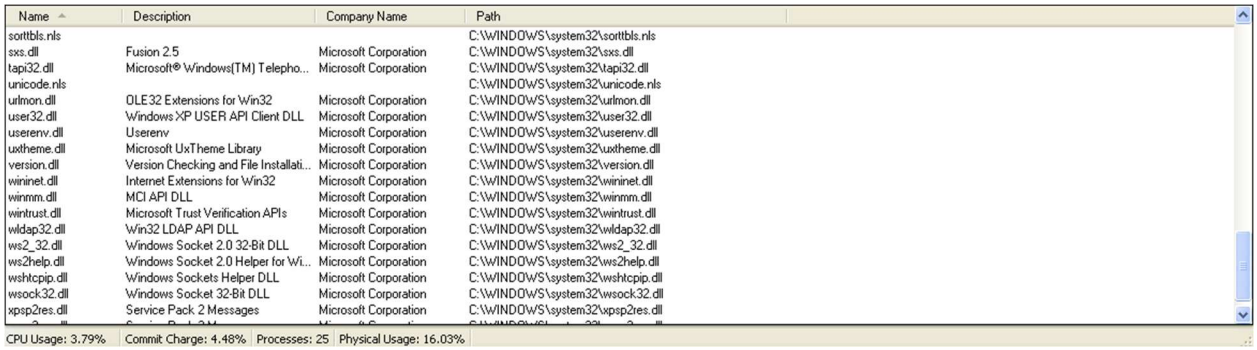
After malware:







Lab 2 – Dynamic Analysis
Gunnar Yonker



Name	Description	Company Name	Path
sorttbls.nls			C:\WINDOWS\system32\sorttbls.nls
sxs.dll	Fusion 2.5	Microsoft Corporation	C:\WINDOWS\system32\sxs.dll
tapi32.dll	Microsoft® Windows(TM) Telepho...	Microsoft Corporation	C:\WINDOWS\system32\tapi32.dll
unicode.nls			C:\WINDOWS\system32\unicode.nls
urlmon.dll	OLE32 Extensions for Win32	Microsoft Corporation	C:\WINDOWS\system32\urlmon.dll
user32.dll	Windows XP USER API Client DLL	Microsoft Corporation	C:\WINDOWS\system32\user32.dll
userenv.dll	Userenv	Microsoft Corporation	C:\WINDOWS\system32\userenv.dll
uxtheme.dll	Microsoft UxTheme Library	Microsoft Corporation	C:\WINDOWS\system32\uxtheme.dll
version.dll	Version Checking and File Installati...	Microsoft Corporation	C:\WINDOWS\system32\version.dll
wininet.dll	Internet Extensions for Win32	Microsoft Corporation	C:\WINDOWS\system32\wininet.dll
winmm.dll	MCI API DLL	Microsoft Corporation	C:\WINDOWS\system32\winmm.dll
wintrust.dll	Microsoft Trust Verification APIs	Microsoft Corporation	C:\WINDOWS\system32\wintrust.dll
wldap32.dll	Win32 LDAP API DLL	Microsoft Corporation	C:\WINDOWS\system32\wldap32.dll
ws2_32.dll	Windows Socket 2.0 32-Bit DLL	Microsoft Corporation	C:\WINDOWS\system32\ws2_32.dll
ws2help.dll	Windows Socket 2.0 Helper for Wi...	Microsoft Corporation	C:\WINDOWS\system32\ws2help.dll
wshtcpip.dll	Windows Sockets Helper DLL	Microsoft Corporation	C:\WINDOWS\system32\wshtcpip.dll
wsock32.dll	Windows Socket 32-Bit DLL	Microsoft Corporation	C:\WINDOWS\system32\wsock32.dll
xpsp2res.dll	Service Pack 2 Messages	Microsoft Corporation	C:\WINDOWS\system32\xpsp2res.dll

CPU Usage: 3.79% | Commit Charge: 4.48% | Processes: 25 | Physical Usage: 16.03%

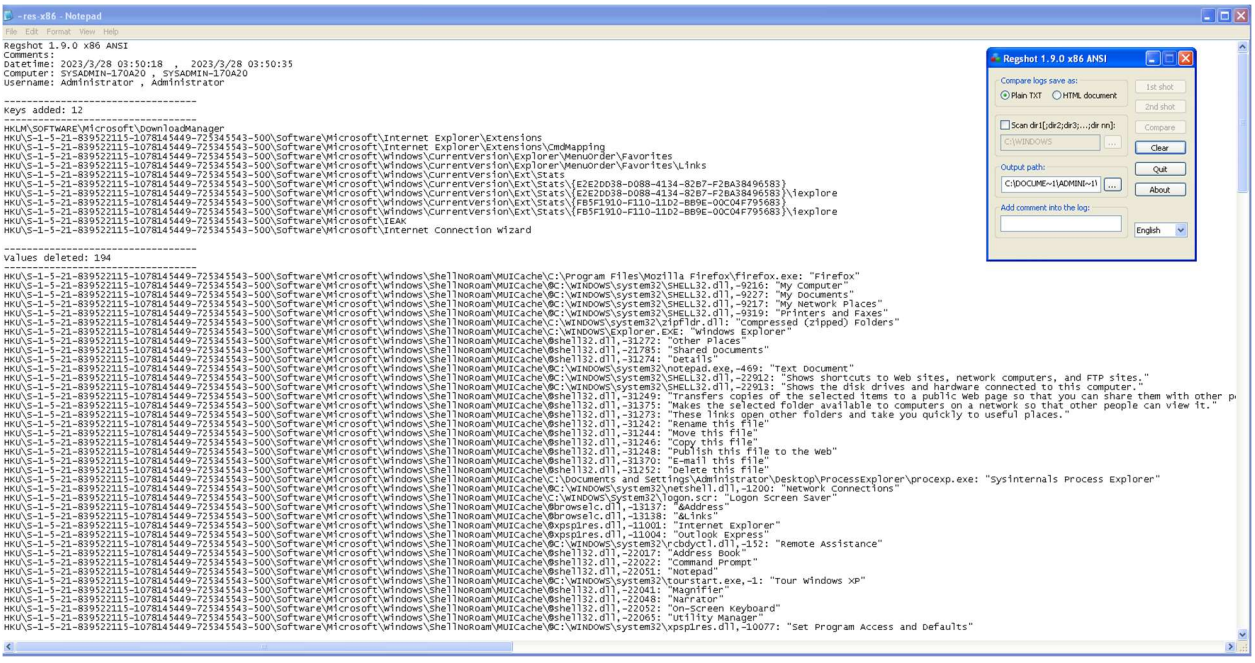
Insights gained: When the malware is run, Process Manager briefly shows the sample malware running before it closes and the new program that is running is a IEXPLORE.EXE program under svchost.exe. When this program is looked into further some of the DLL libraries that are imported deal with the Windows Socket for connection. The mshtml.dll, shdocvw.dll, and wininet.dll references are commonly used with Internet Explorer. The strings found in the memory of the file show that the malware is creating a new process for browsing with BrowseNewProcess and this means it may be attempting to execute code within this new process. The references to KERNEL32.dll, NTDLL.DLL, USER32.dll, SHLWAPI.dll can also indicate that the malware might be attempting to interact with operating system components. The strings related to error handling and debugging also suggest that the malware is attempting to avoid detection and analysis. There are also references to cryptography packages which means the malware may be trying to do operations that require encryption. In strings, CreateWindowExA and CreateMenu both suggest that the malware may be trying to display a fake or misleading user interface to trick the user into interacting with it.

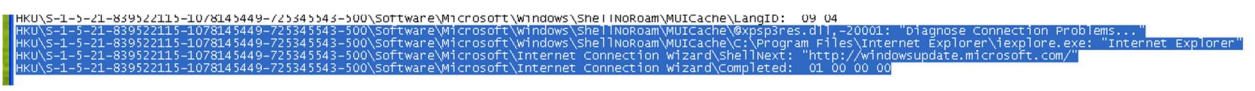
Lab 2 – Dynamic Analysis
Gunnar Yonker

Tool: Regshot

Purpose of Tool: Regshot is a tool that can take snapshots of the system registry and then they can be used to compare the system registry before and after malware execution. This is extremely useful to be able to identify any registry entries that have been modified by the malware execution.

Relevant information from tool:





Insights gained: By using this tool I was able to see that the malware made a lot of changes to the registry, 230 in total. There were 17 keys added and 7 values modified that were related to system settings such as Download Manager, Internet Explorer extensions, Explorer menu order, and Internet Connection Wizard. There were modifications made to the UserAssist registry which could be an indication that the malware is trying to monitor the user's activity on the system. There were 194 values deleted from the registry that was removing the language values for system functions such as My Computer and My Documents. It also was deleting values related to file operations such as renaming,

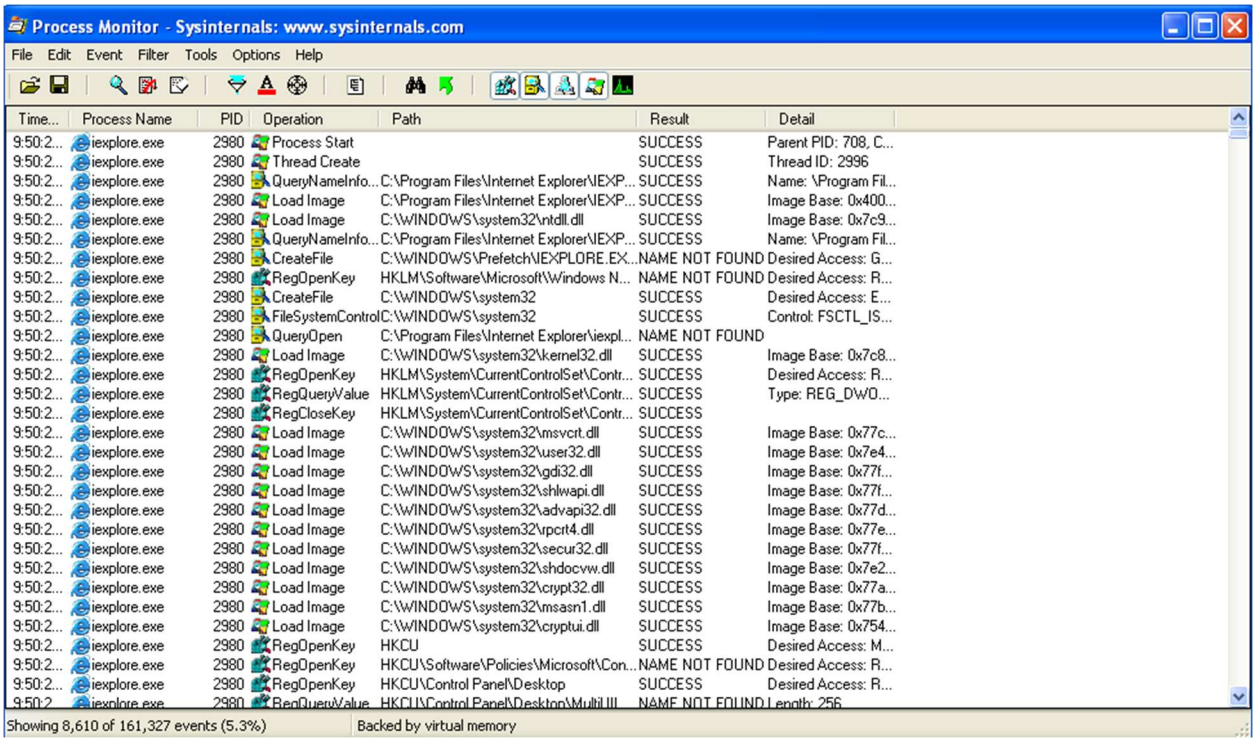
moving, and copying file. There were also modifications to values associated with web publishing. With all of the deletions and modifications by the malware I suspect that it was attempting to disrupt or disable system functions to make the user believe that they needed to update their windows system by then opening the windowsupdate.microsoft.com/ webpage in the Internet Explorer window.

Lab 2 – Dynamic Analysis
Gunnar Yonker

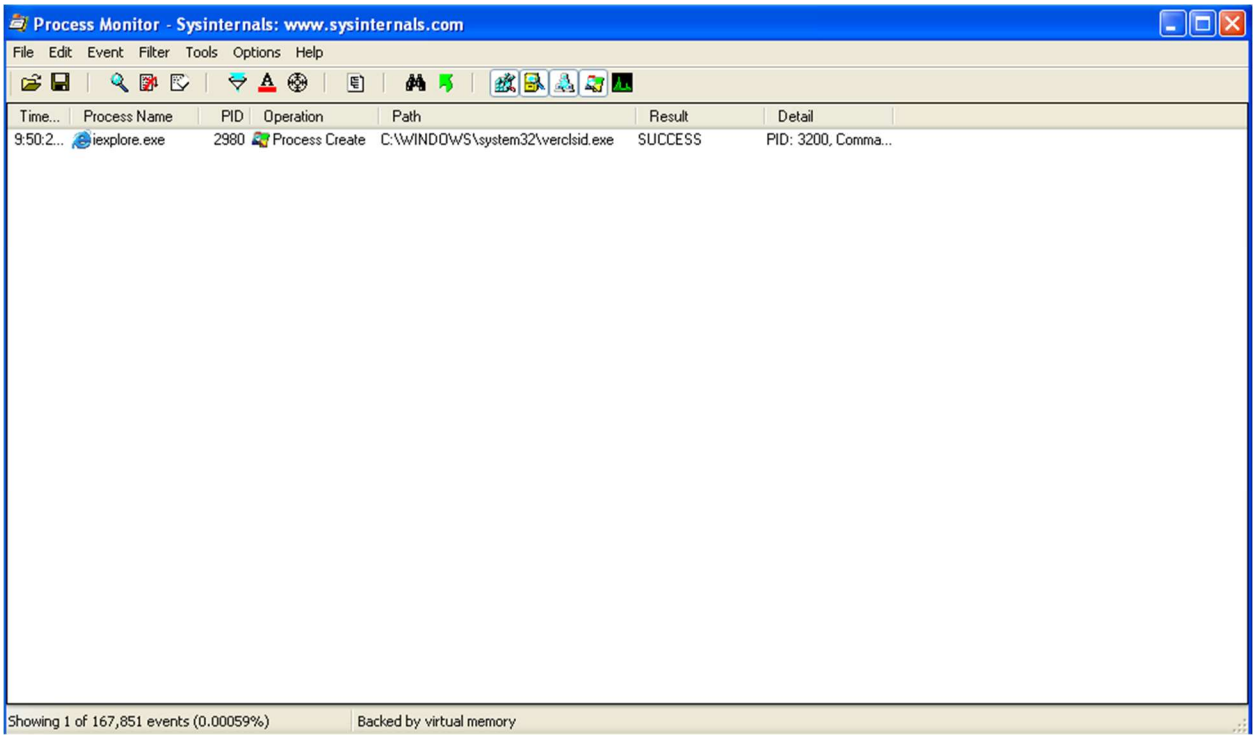
Tool: Process Monitor

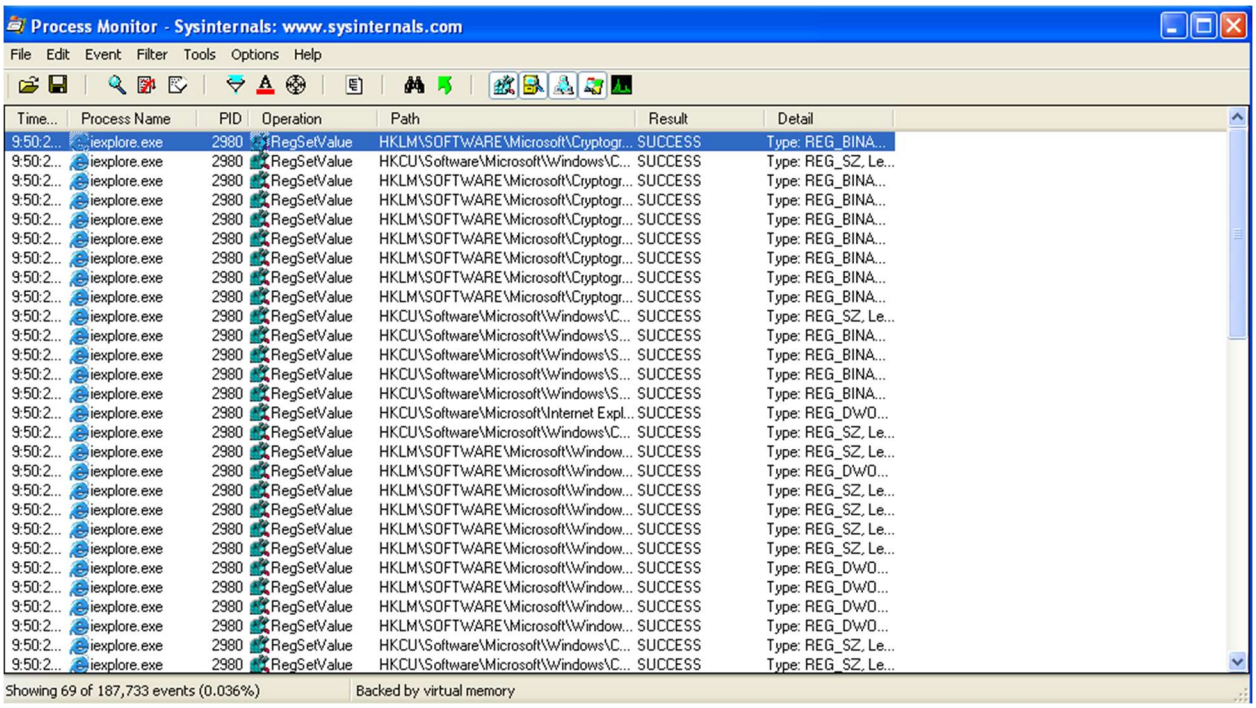
Purpose of Tool: The Process Monitor tool is used to monitor registry, filesystem, network, process and thread activities. With this tool network activity may not be reliably logged across the different versions of windows. It will monitor all systems calls when it is run and the event logs consume memory leading to the potential of exhausting all system RAM quickly. This tool can capture detailed information about these events, the timestamp of the event, and any relevant details about the event.

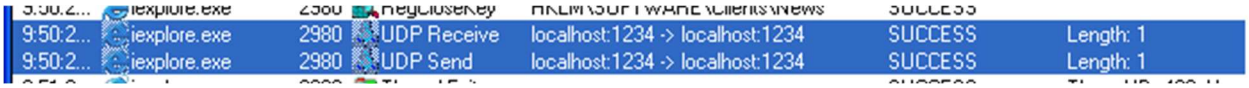
Relevant information from tool:



Lab 2 – Dynamic Analysis
Gunnar Yonker







Insights gained:

No WriteFile operations are taking place so that means that the malware is not creating any files. ProcessCreate operation is found and creating the verclsid.exe file. It also appears that there are many

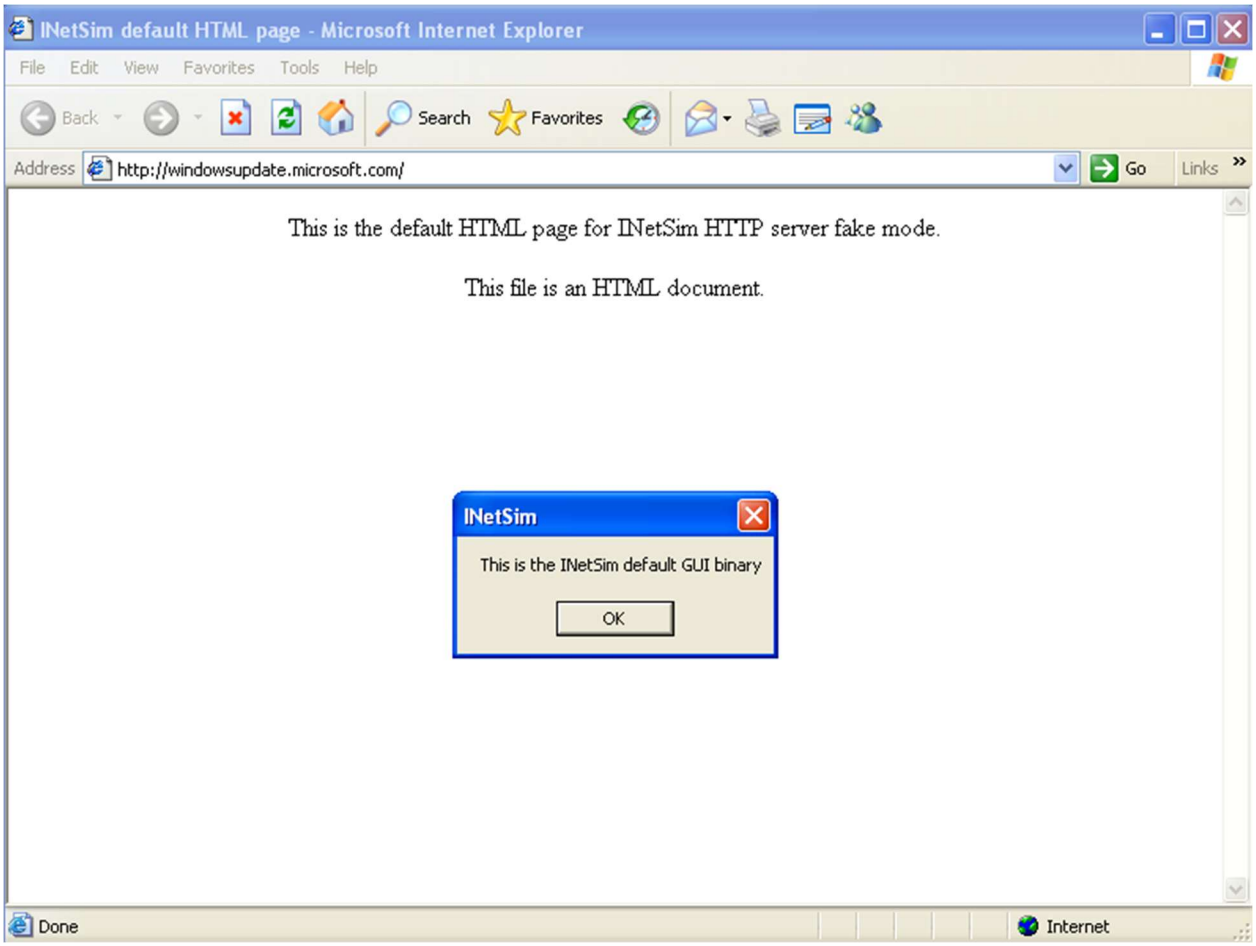
operations involving the registry such as RegSetValue. Towards the end of the Process Monitor list after the malware is run there is a UDP Receive and UDP send operation. There are also many operations where files are being read. There are many events that are taking place with many of them pertaining to the registry being modified to some capacity. I had difficulty filtering through the noise with this tool and need to expand my knowledge of what operations I should pay particular attention to so that I can filter through the noise more efficiently. I also noted that there was a Thread Create event that was then followed by a Thread Exit at the end of the Process Monitor List.

Lab 2 – Dynamic Analysis
Gunnar Yonker

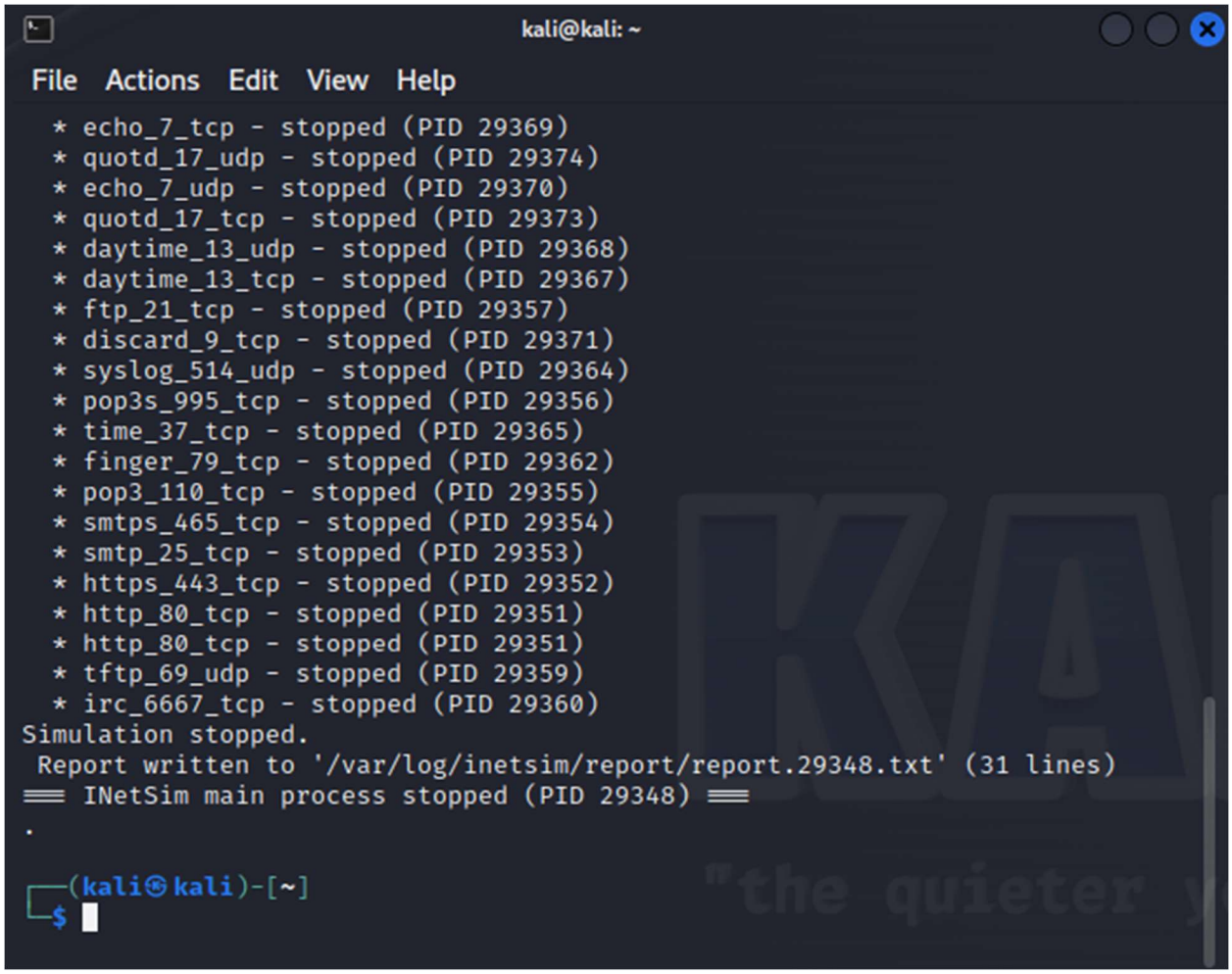
Tool: INetSim

Purpose of Tool: INetSim is a Linux-based software package for simulating various internet services such as providing fake services to incoming requests such as HTTP, DNS, SMTP, and HTTPS. It also records all incoming requests and connections which is useful for dissecting network behavior of a malware.

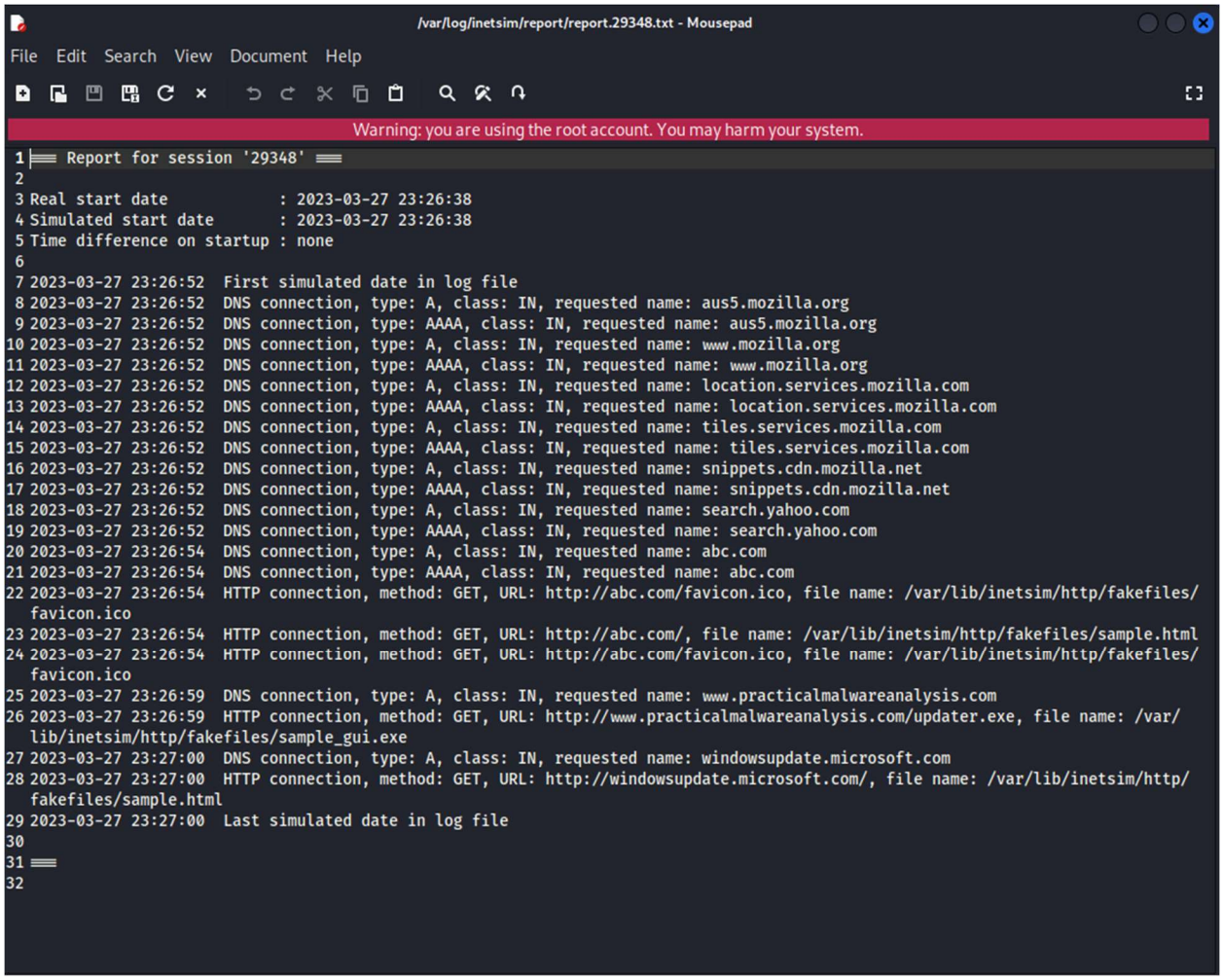
Relevant information from tool:



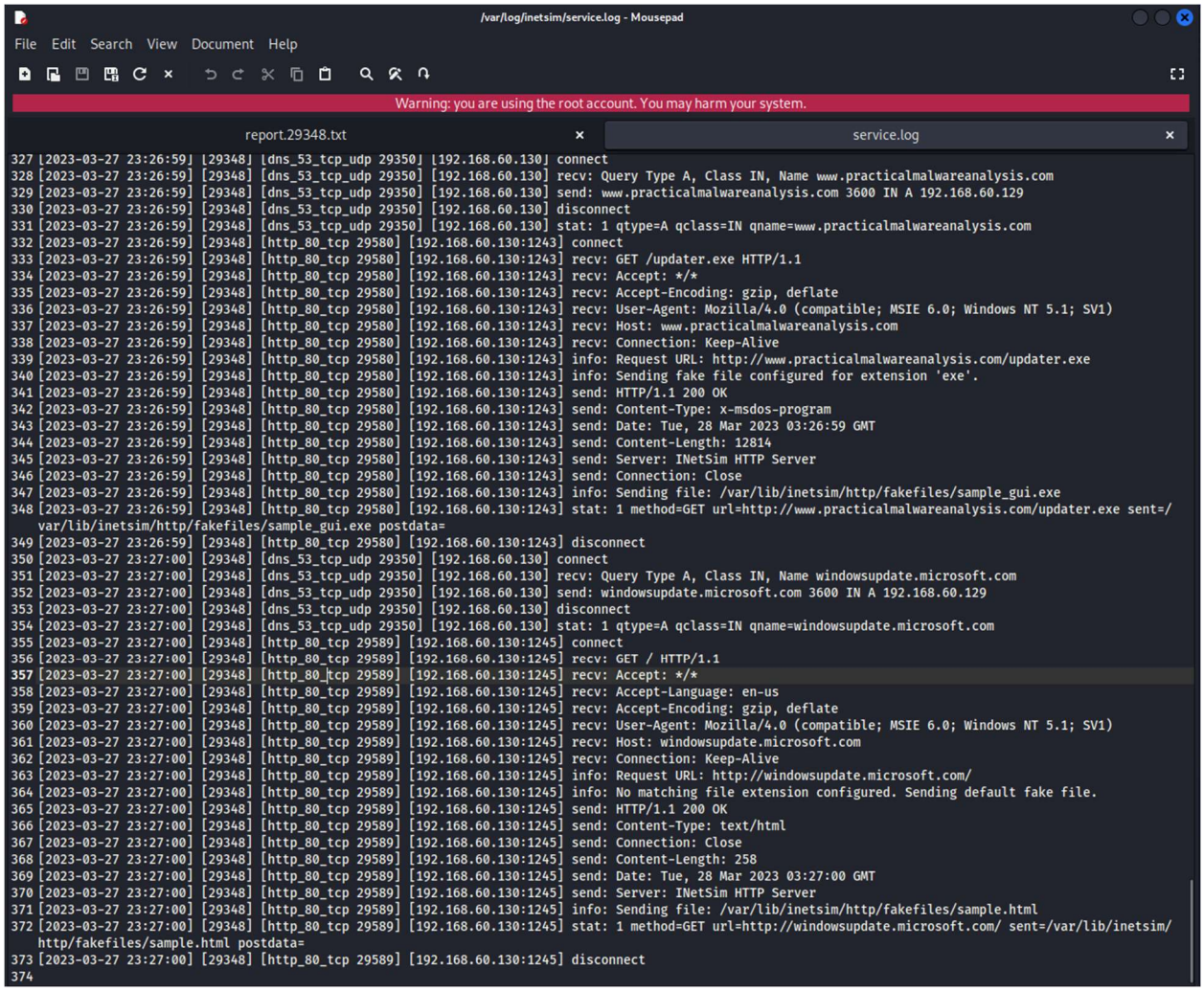
Lab 2 – Dynamic Analysis
Gunnar Yonker



Lab 2 – Dynamic Analysis
Gunnar Yonker



Lab 2 – Dynamic Analysis
Gunnar Yonker



Insights gained:

This tool provider a lot of valuable information about the nature of this malware. The malware attempts to get a file called updater.exe from practicalmalwareanalysis using a DNS connection and then a HTTP connection to retrieve the updater.exe file which is suspected to be a malicious file. It then requests the host of windowsupdate.microsoft.com/ to make the user think that the file is from Microsoft and that they need to update their windows.

Lab 2 – Dynamic Analysis
Gunnar Yonker

Conclusion:

The tools used in this lab to analyze the Lab 2 malware sample provided valuable insights into what processes the malware created using Process Explorer. I was able to see that the malware executed, started a internet explorer process, and then exited so that I would not be able to see that the malware was executed and still running. Using this tool I was also able to see the string that was accompanied by the malware created process, this section provided information as to what the malware was attempting to do with this process with functions such as CreateWindowExA and BrowseNewProcess. From this tool I knew which process to track when using the Process Monitor tool. Using the Regshot tool I was able to see what registry files were deleted/modified/created by comparing the windows registry before the malware was executed, and then after. This malware made a number of changes to the registry, with 230 operations taking place and a majority of them being deletions. Many of the modifications, deletions, and additions appeared to deal with system settings and system operations which gave me insight into the idea that the malware was attempting to change or hide system settings to then trick the user into further actions. Next, using the Process Monitor tool to look at events pertaining to the process created by the malware that I found using Process Explorer I was able to see all of the registry events taking place and other events such as a UDP connection attempt and ProcessCreate. Using the INetSim tool by setting up a Kali Linux VM that could run the tool and routing the network connection of my Windows XP infected system through the Kali Linux VM running INetSim I was able to get what I think is some of the most valuable information about this malware. I was able to see that when the malware was run that it was creating a DNS connection to practicalmalwareanalysis.com and then an HTTP connection to retrieve a file called updater.exe. It then tried to connect to the windowsupdate.microsoft.com/ url and that is the page that would be displayed to the user.

Taking all of these insights gathered from the use of the dynamic tools used, I think that the objective of the malware was to modify the system settings to make the user believe that their system needs a critical windows update and trick them into believing that. At the same time the malware would run, and then execute the Internet Explorer process which would first retrieve the updater.exe file from the original connection of practicalmalwareanalysis.com, then it would display to the user the windowsupdate.microsoft.com/ page in the Internet Explorer program that was opened. This combined with the modification of the windows system settings would trick the user into believing that the update was needed and that the updater.exe file was from the Microsoft page that was opened, and they would run it which would further infect the system. Once that file was executed, I am unsure as to what would happen after that file was executed, but if it was the goal of the malware to have the user update using the retrieved file then I would assume it would further infect the system in some capacity.

Key Takeaways:

I think that this reinforced the idea for me that first the static malware analysis should be completed because now with an understanding of the tools for the static analysis, it is a very quick process and can provide some initial insights. The malware can be further analyzed using dynamic analysis tools that we learned about in this module. One of my key takeaways is that I found it difficult to filter through the noise events in Process Monitor, and that as I use it more and understand more of the commonly used operations that it will become a powerful tool in analyzing malware samples. I really enjoyed learning about the use of INetSim by having the infected system route the connection to the Kali Linux VM running the INetSim. This is a very useful tool if it is observed that the malware is trying to connect to

the internet and retrieve any files or if the malware is trying to communicate with a Command & Control center by sending sensitive information that was gathered from the infected system. It was also interesting to see how the tools can be used with each other such as pairing Process Explorer and Process Monitor together to make sure that the correct process is being analyzed.