Reverse Engineering and Malware Analysis Home Lab

Description

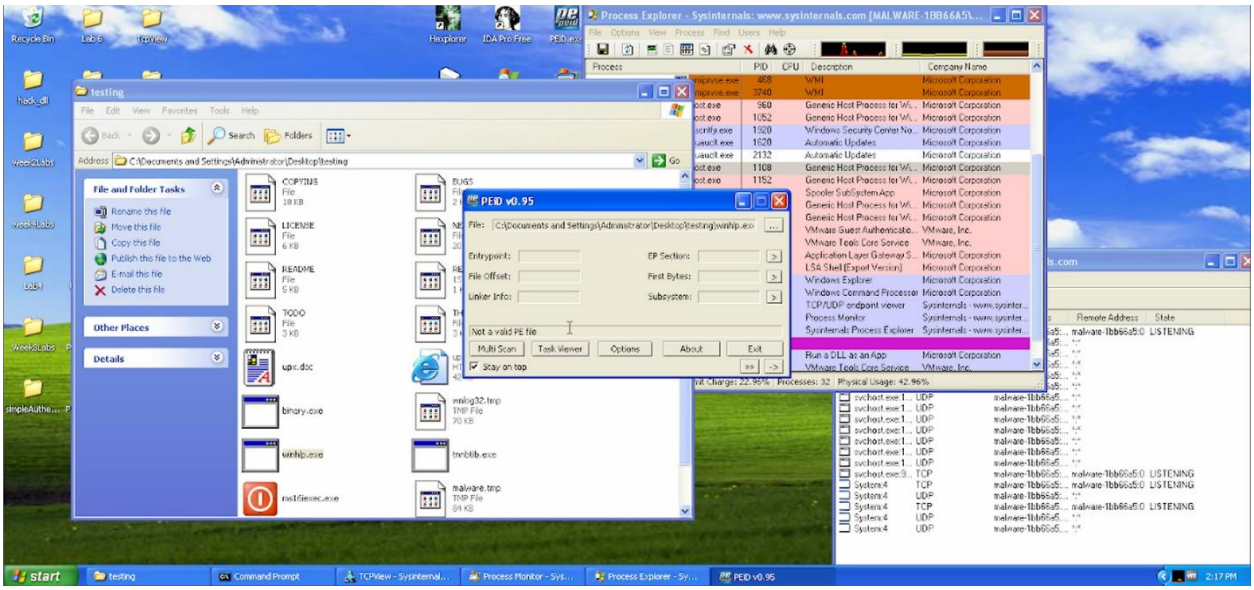
The main purpose of this lab is to gain hands-on experience with utilizing a program for reverse engineering and malware analysis. The lab involves analyzing and executing a malware sample to observe its malicious behaviors and identify potential activities in a controlled virtual environment.

Programs and Environments Used

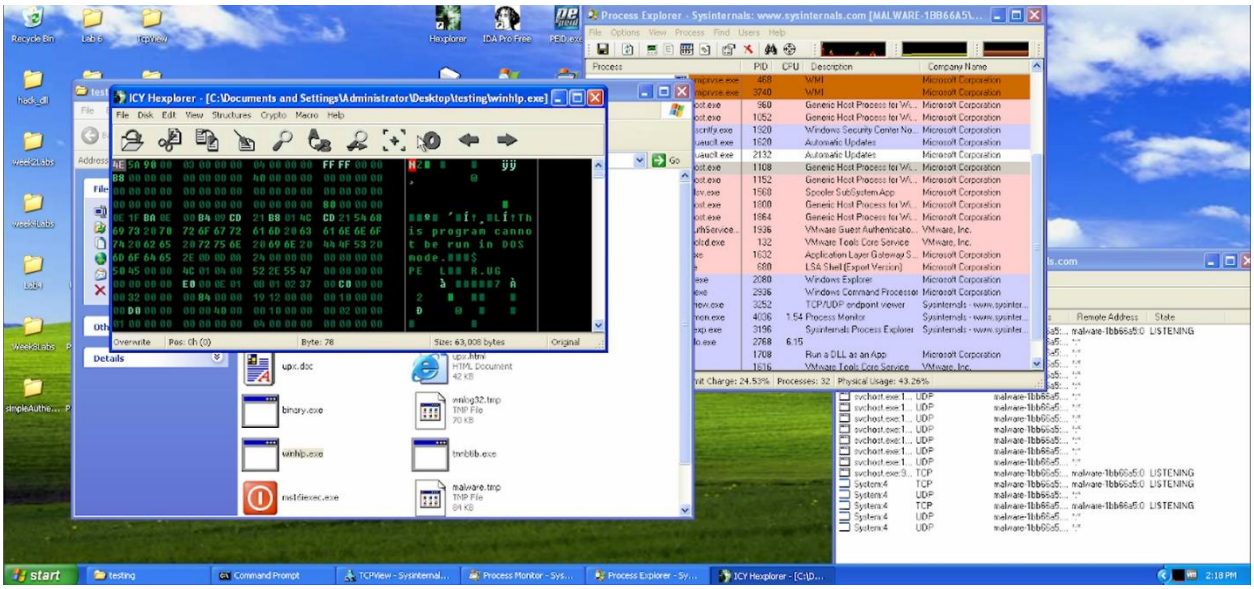
- Hypervisor: VMWare Workstation 16 Pro
- Environment: Windows XP Home Edition
- Malware Sample: winhlp.exe (MD5: 793c46c01ff4f4bb5670bdb07c61fe89)
- Programs Used for Static Analysis: Hexplorer, PEiD, IDA Pro
- Programs Used for Dynamic Analysis: Process Explorer, Process Monitor, TCPView

Walkthrough

Before we analyze the malware sample (winhlp.exe), we want to check if it is a valid PE file to see if it will function as intended when executing for dynamic analysis. Winhlp.exe will be inserted into a program called PEiD to verify if it is a valid PE file as seen below:

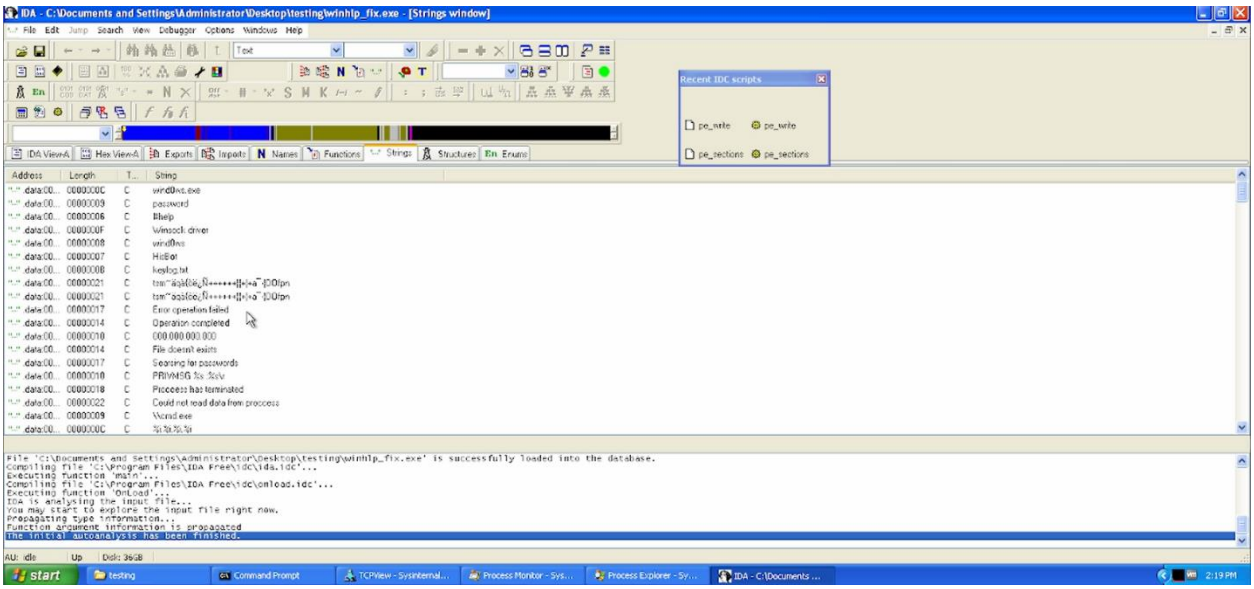


Because winhlp.exe is not a valid PE file, we will then insert the malware sample into a text editor (Hexplorer) to check for any errors within the PE header as seen below:



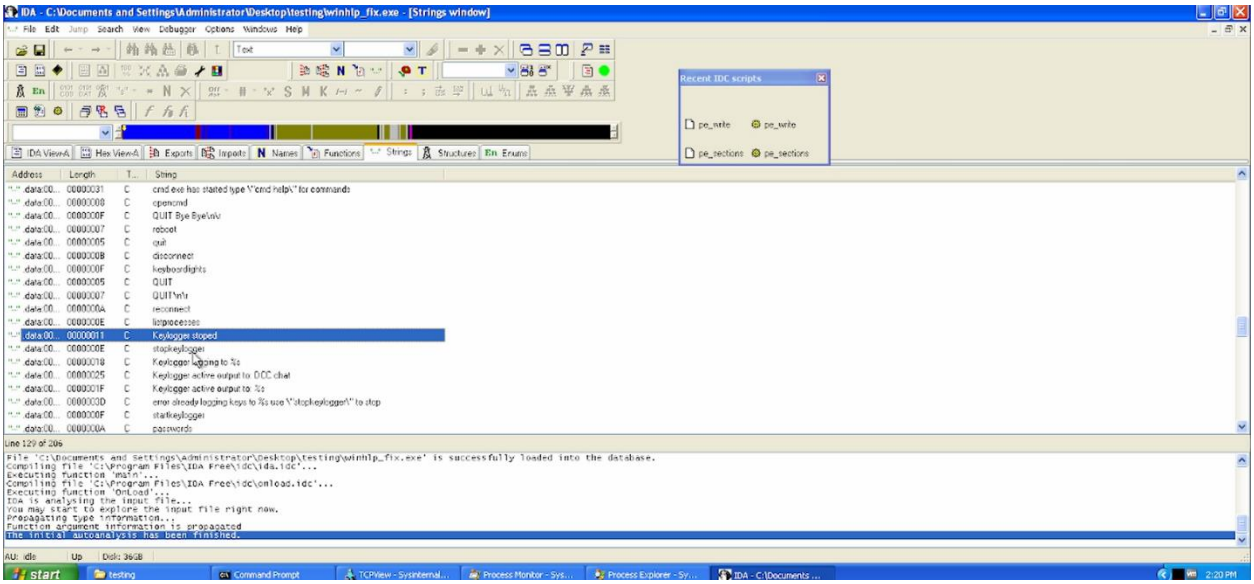
From the text editor, we can see that the magic number of the malware sample is incorrect as it should be MZ and not NZ. When correcting the magic number and saving it as a separate file (winhlp_fix.exe), putting that new file back into PEiD will result in a working PE file.

After fixing the PE header, we will insert winhlp_fix.exe into IDA Pro to conduct static analysis. We will first analyze the strings that IDA Pro detects within winhlp_fix.exe to identify possible malicious behaviors as seen below:

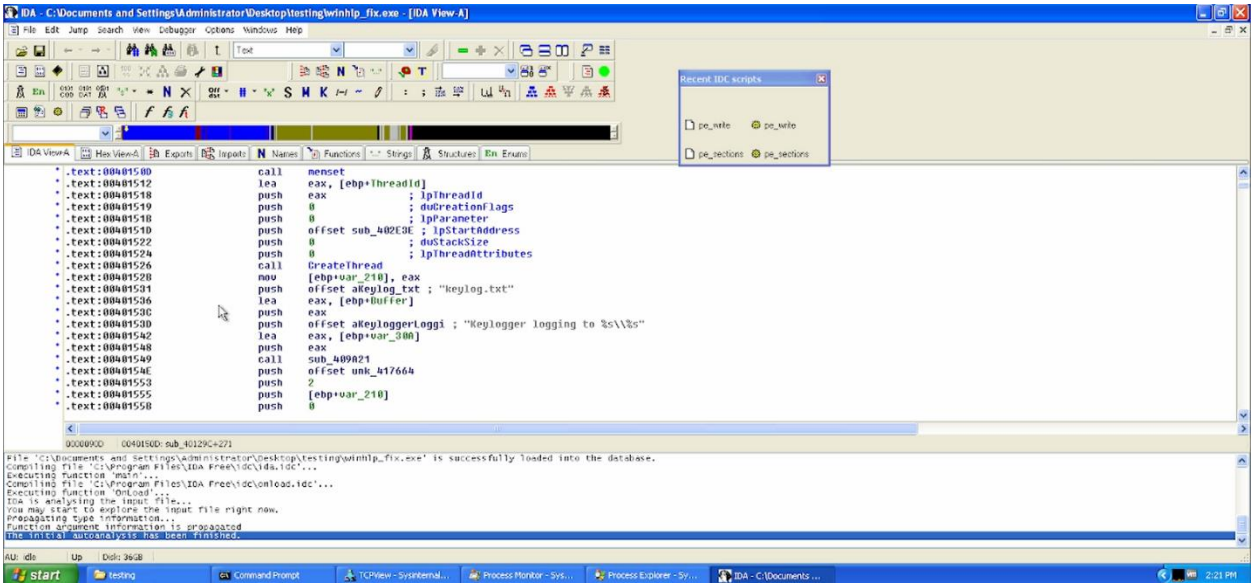


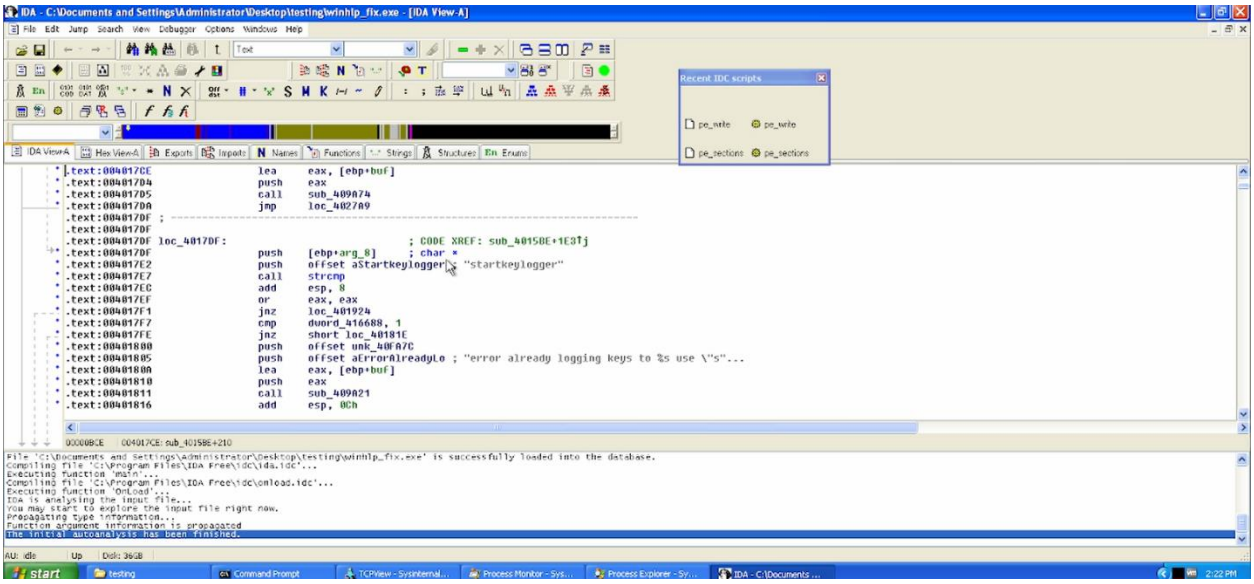
In the image above, we can see a couple of interesting strings identified within the malware sample. The first string we see is wind0ws.exe, which is clearly a malicious file that will likely be created once we execute winhlp_fix.exe. Another string that stands out is keylog.txt, which is very likely to contain records of keylogging activities monitored by the malware. When scrolling through the large list of strings of winhlp_fix.exe, one can

view a plethora of other miscellaneous strings related to port scanning, password searching, system calls, domain names, server information, and more. For the focus of this lab, we will only focus on the keylogging portion of the malware sample in which one can view a screenshot of more strings related to keylogging as seen below:

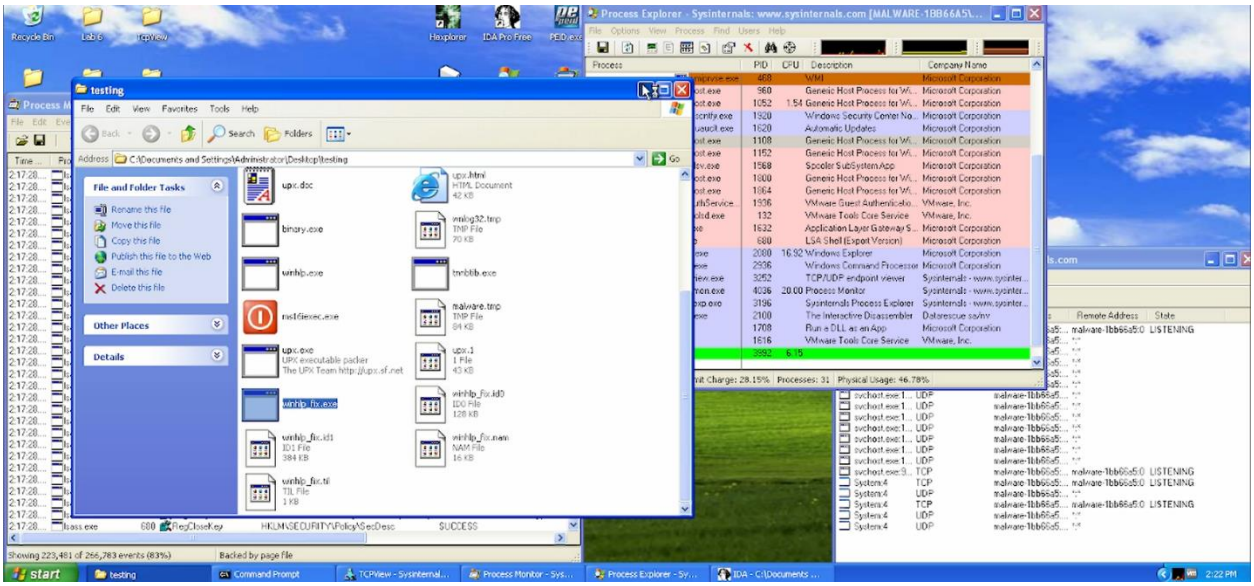


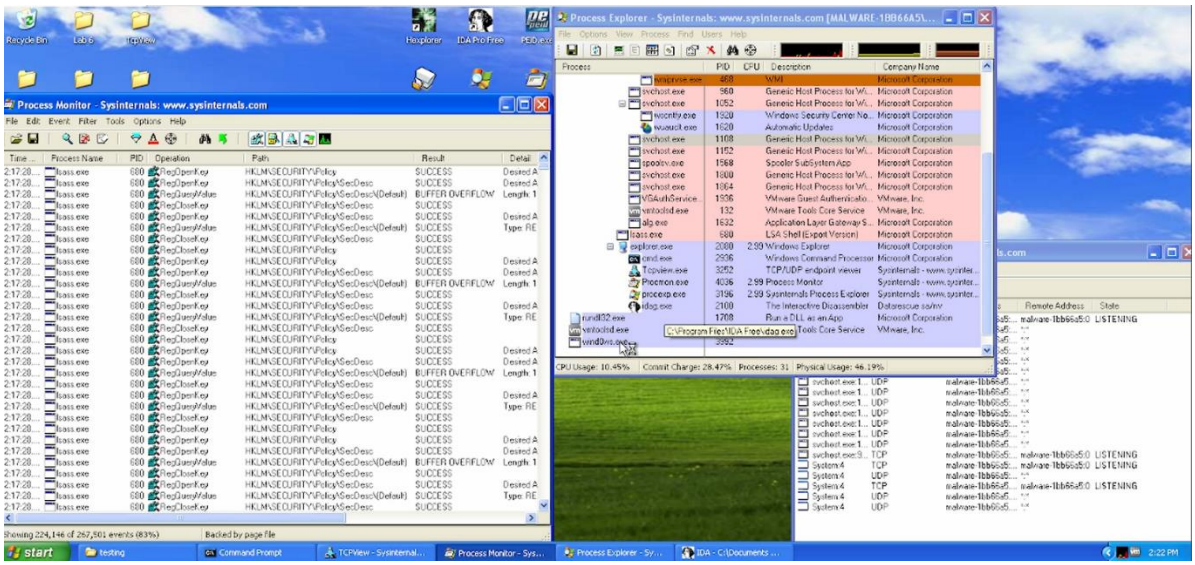
For further static analysis, we can view the assembly code of winhlp_fix.exe and see the same set of strings from earlier associated with the set of assembly code instructions next to it as seen below:





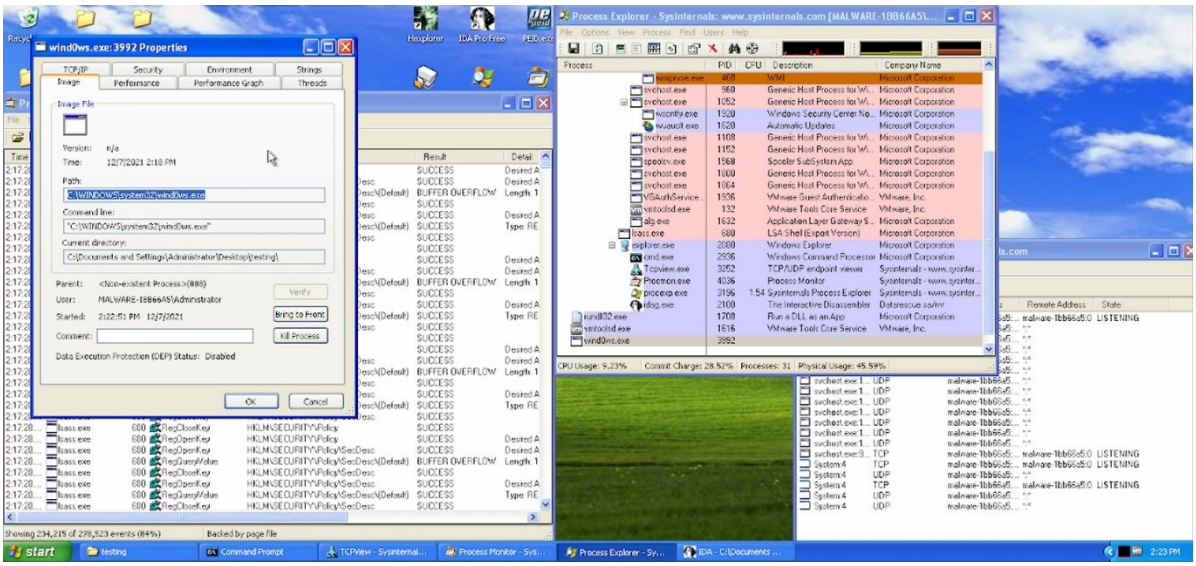
After utilizing IDA Pro to conduct static analysis of winhlp_fix.exe, we will then conduct dynamic analysis of the malware sample via three programs: Process Explorer, Process Monitor, and TCPView. To do so, we will execute winhlp_fix.exe and observe the programs for any signs of malicious behaviors as seen below:

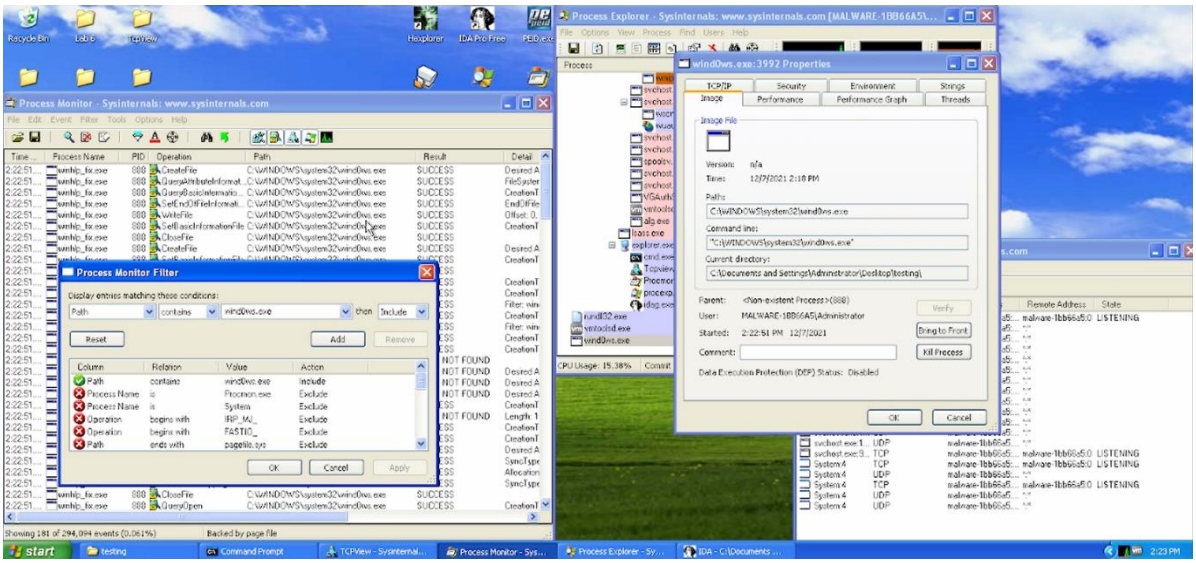




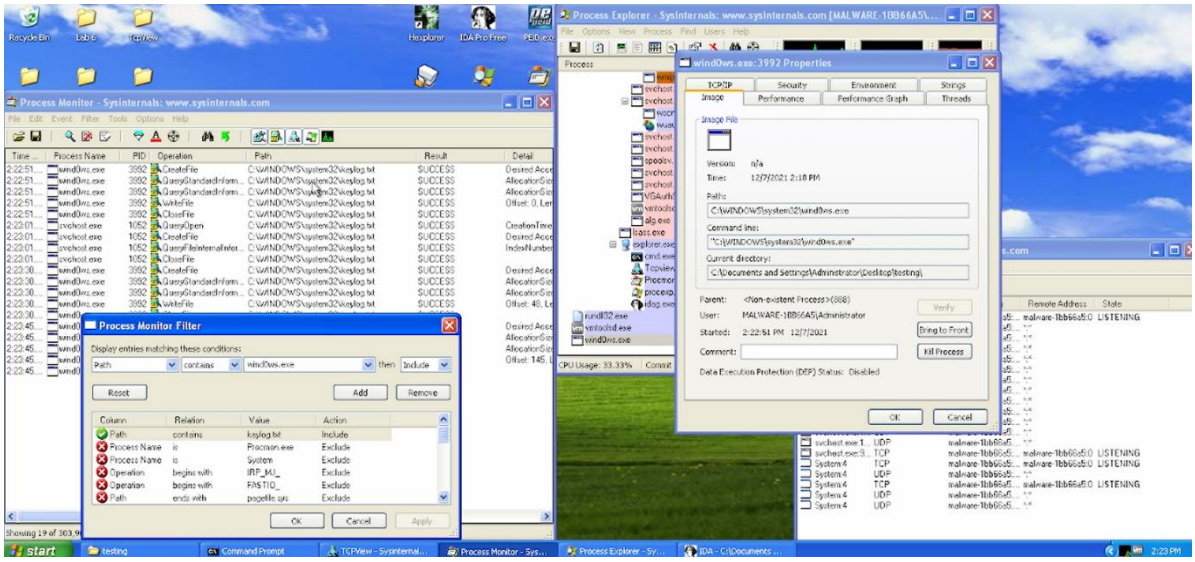
As we can see in the second image above, Process Explorer detects that wind0ws.exe was created right after executing winhlp_fix.exe, which we predicted from our static analysis. However, TCPView did not detect any new network connections being made, which implies that the malware is not trying to communicate with any outside network or failed to do so.

When viewing the properties of wind0ws.exe from Process Explorer, we can see the directory path of where wind0ws.exe is located. We can also use Process Monitor to filter our search in locating the same directory path of wind0ws.exe as well as confirm that wind0ws.exe was created by winhlp_fix.exe from below:

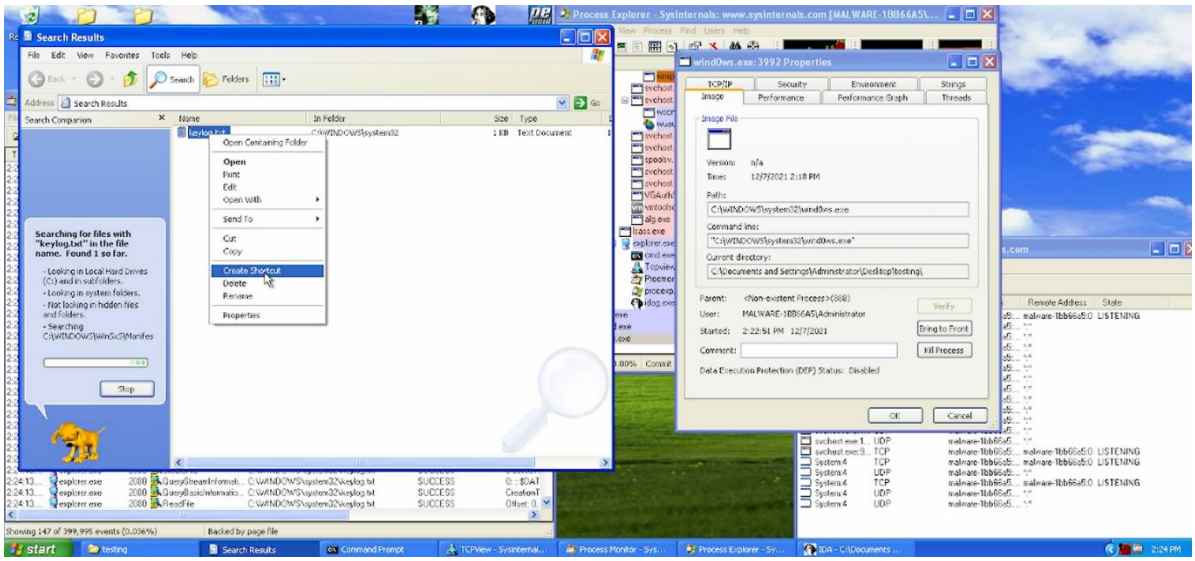




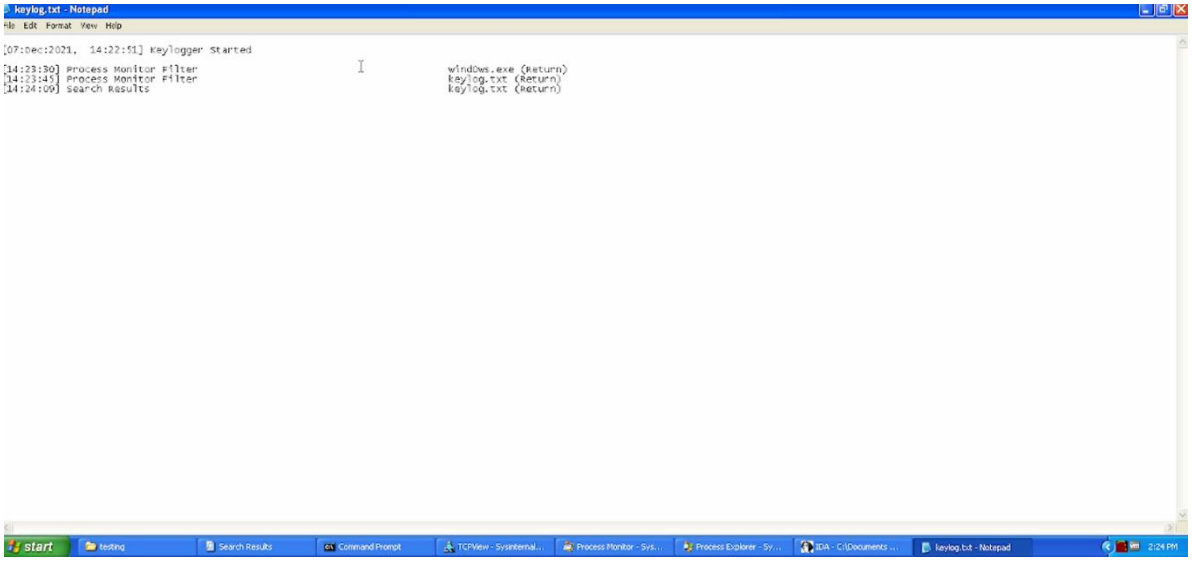
With wind0ws.exe's directory path located and confirmed through Process Monitor, we can do a similar search to locate the directory path for keylog.txt as that is a big indicator for malicious activities of winhlp_fix.exe. When searching for keylog.txt, the directory path of it turns out to be the exact same path as wind0ws.exe as seen below:



From the results above, we can infer that keylog.txt was created through wind0ws.exe as shown by the operation CreateFile by wind0ws.exe. With the directory path of keylog.txt located, we can conduct a search on our local Windows XP machine for keylog.txt and create a shortcut for us to access it on our desktop as shown below:



When opening the shortcut of keylog.txt from our desktop, we can view the different types of actions we have done previously when conducting dynamic analysis as shown in the keylogging file from below. We can infer that keylog.txt started functioning once we executed winhlp_fix.exe, and it was able to record the actions of our searches during the dynamic analysis from Process Monitor and the local system search:



Next Steps

There are multiple parts of this lab that can be further analyzed for better understanding of the malware sample. Some future steps we can take to conduct further analysis of winhlp.exe would be:

- Check for hidden/obfuscated strings in text editor/IDA Pro assembly code

- Reverse engineer the assembly code to view where the data is being pushed and stored in memory
- Conduct further analysis of the system calls and operations made by winhlp_fix.exe based on the results of Process Monitor to better understand what it is trying to do