Nathan Ly CS-GY 6963

Registry Malware Scanner

Host: https://github.com/nate0203/Registry-Scanner

The Windows Registry contains a lot of set up information that every installed program uses. It is very overwhelming to comb through the entire thing and difficult to find what you may want. Since every program leaves its mark on the registry, we could potentially find hidden malware. However, it is difficult to detect without doing an extensive search which would be very time consuming.

Hive files were created for machines infected by malware. Using VirtualBox and a Windows 10 image, a safe environment was created. Going to KernelMode.info, a couple of simple malware samples were used in an attempt to infect the machine. The files, demo1 and demo2, are the results of the malware sample. The first file, demo1, created a folder with programs that run on start up. These executables were found and running the script reader.py on the infected machine with the option to use VirusTotal which marked those a few of those files as malware. The second file, demo2, was a sample with a very simple keylogger built using Visual Basic that was not found in common malware locations. Using reader.py, the location was found but there were no executables related to it. However, further analysis could be completed with the script find.py to check where else it exists in the registry and allows someone to determine what program it could have been installed with based on the timestamp associated with the registry key. The result was that it is a standalone program that was installed knowingly.

File – reader.py

SHA256: e7bc08f2aaaf3a6d1f8d6abadb7983dd17f0e0b04eabad7a686566f9c78c9ab4

MD5: 7eb4728232b434fde93d1208b8affb32

Usage – python2 reader.py

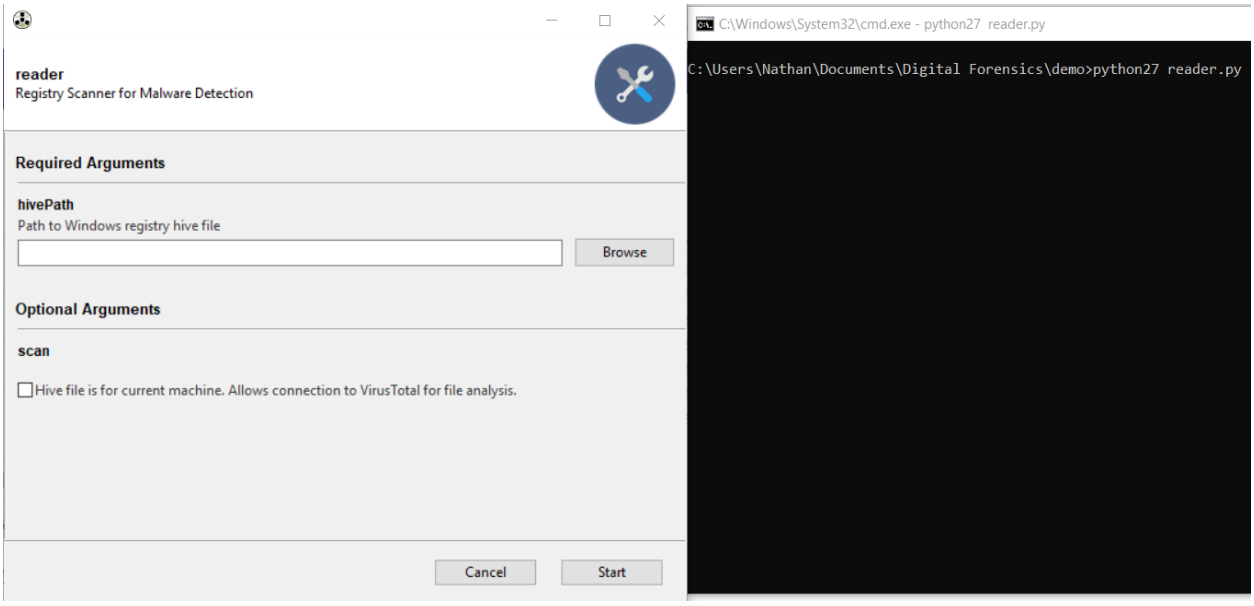
Hive File Tested – demo1

SHA256: 7f523c58e9b4ff2c203a9883c2439d7dc8f8e92e3ba4826b385928bf3d810f08

MD5: 0dc721bd12769a60292d5437ff214cbd

Output – SQL database and TXT file

The script will load a GUI that will allow browsing of the file system. One may choose the hive file and then an option to scan but this only works if the hive file belongs to the current machine. This script looks at common paths in registry where malware may insert itself. These values must be reviewed as malware may not be obvious in its naming conventions. The script attempts to open these paths in the hive file and then extract the value. There is also a scan (depth first search) for any executable or related files. The option is available to use the results of the scan and attempt to check if those files are malware comes from connecting to and submitting a request to the VirusTotal API. Again, this option only works if the hive file that was exported belongs to the current machine running the script as the value in the registry would be the path to the file in question. This path could either be an absolute path (C:\Windows\...) or a relative path which could be resolved using the winreg library (turning %SystemRoot% to C:\Windows). Output consists of a text file with a summary of what is in the database and a SQL database with all the data from the registry for someone to review.



File – find.py

Hashes –

SHA256: 9199af34cabc7dc83a3c01f29cd1b862d1fade47da7dc08fd1678dbfd90a0179

MD5: 56530bfa34d3d9c4930eaa71526cb6c3

Usage – python2 find.py

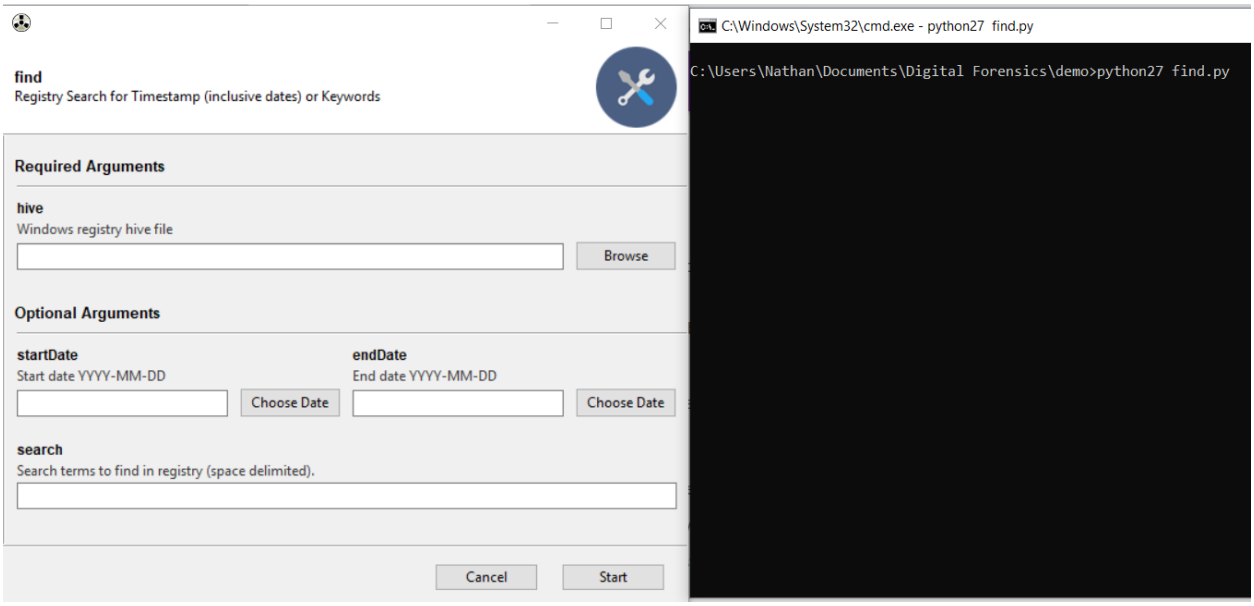
Hive File Tested – demo2

SHA256: cac2e04f5a8246ecd37d761682327e0aa5f395803538cec7ab31b0aa90c37f2b

MD5: faa7d2c80a54b5439f239ccbf5ba5b60

Output – SQL database

The script will create another GUI which allows someone to browse the filesystem for the hive file. This script searches the registry for terms provided and/or keys that fall on a range of dates. For any registry key, there is a timestamp associated with it and all those value/data entries that fall underneath that key is saved. The output is a SQL database with a single table showing the results of the search. It makes it easier for an investigator to query if their search leads to a significant number of items. Testing involved providing different types of search terms to see if it would appear.



Overall these scripts allow a user to analyze registry from external machines and could lead to discovering what type of malware exists or when it appeared. Currently, the scripts only work on Window machines as there were issues using the GUI in Linux.