Alina Malware Analysis Report
By Olotu Praise Jah

May 2022

Contents

Introduction

Executive Summary

Analyzing the Attack

This section details the analysis performed on a component of the malware – spark.exe.

Static Analysis

Checking the file type of the malware:
```
$ file Spark.exe
Spark.exe: PE32 executable (GUI) Intel 80386, for MS Windows
```

Getting the hash function:
```
$ python3 hashes.py
SHA256 Hash = 1fabbd3d6fb5bf868ef07be4774649c4dd3f90959ef1e4477edd08f96de47f03
```

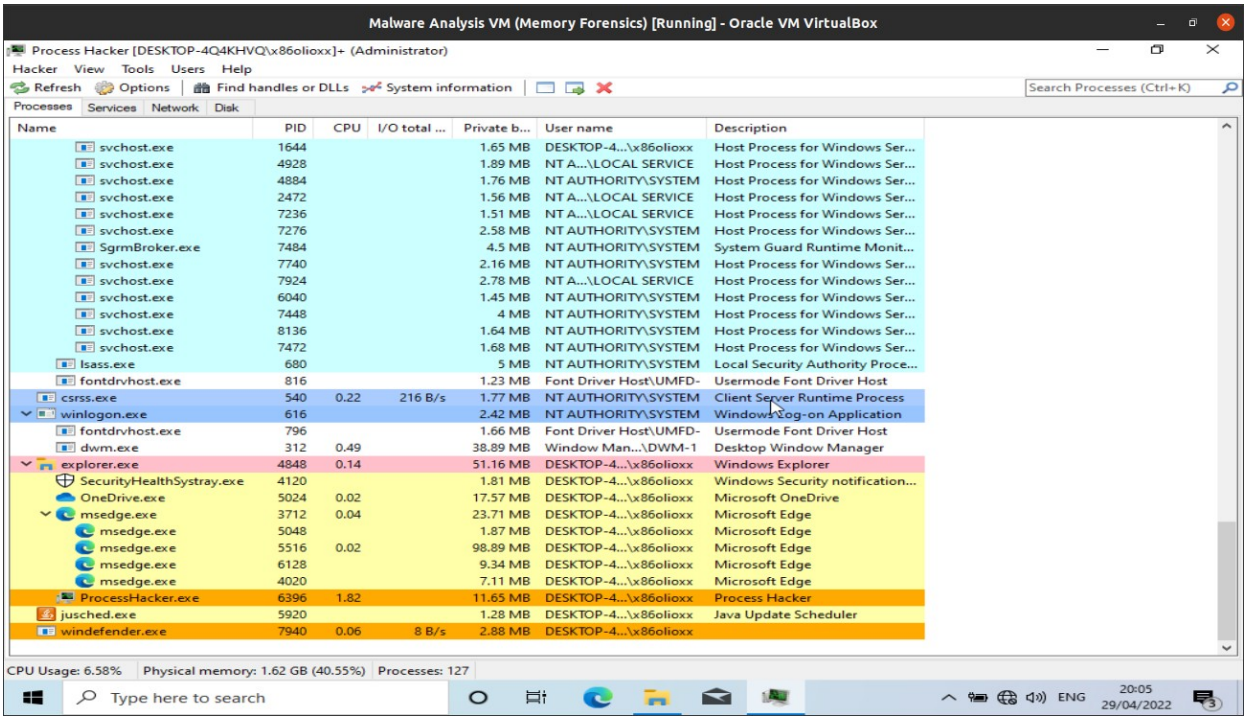
Checking the compile date of the malware:
```
$ python3 comptime.py
2014-05-23 06:51:59
```

Checking the DLL imports:
```
$ python3 imports.py Spark.exe
b ‘KERNEL32.DLL’
b ‘ADVAPI32.DLL’
b ‘SHELL32.DLL’
b ‘URLMON.DLL’
```

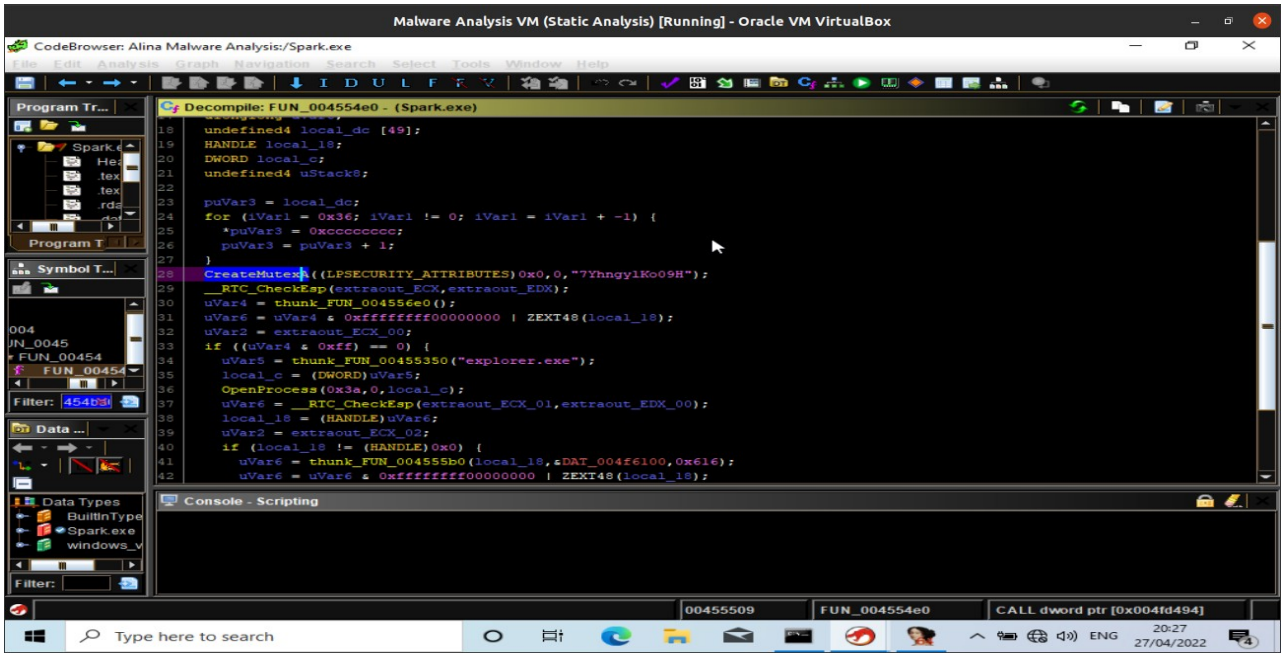
Utilizing the strings tool that comes with Linux on Spark.exe displays the list of legimate programs to avoid when scraping card infomation, the user that compiled the malware, the C2C servers used by the malware, default location of the rootkit once initialized etc

Dynamic Analysis

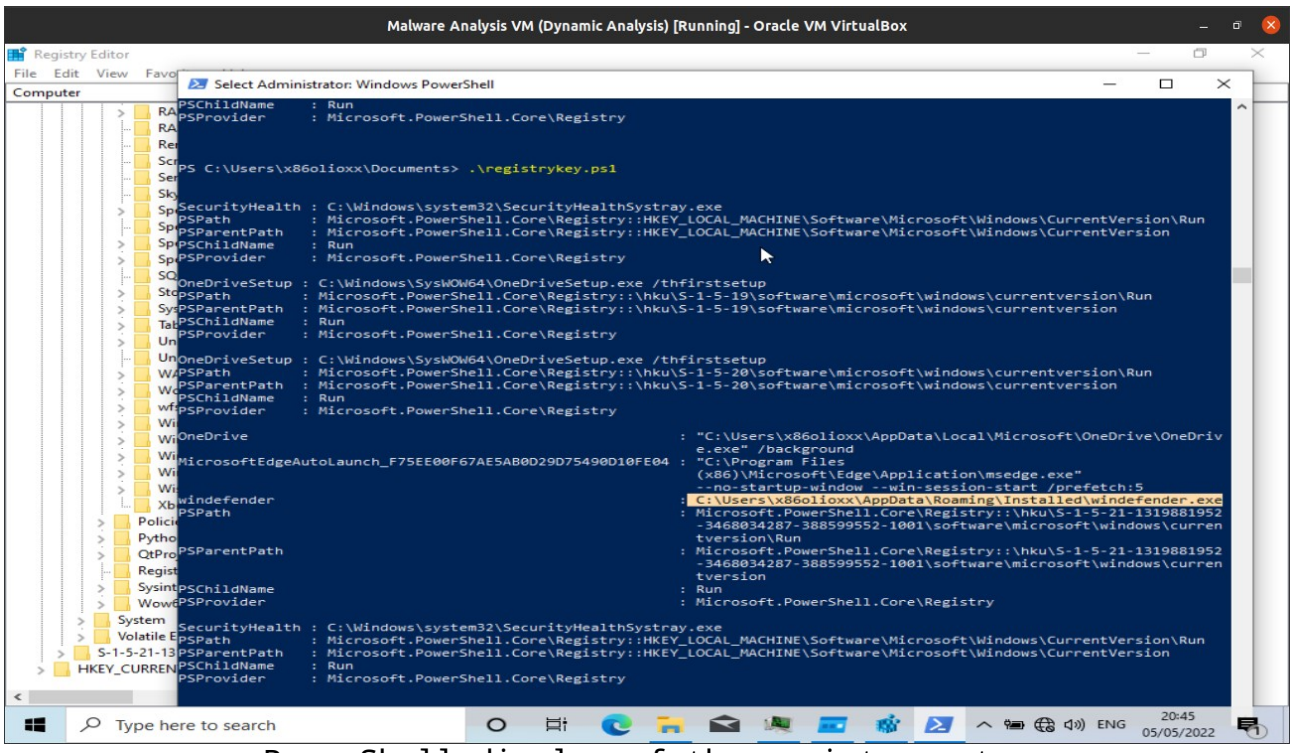
When Spark.exe is running on a machine, it installed(saves) an encrypted copy of itself in %APPDATA\ntkrnl%, then after it has been decrypted with a hardcoded password of 7YhngylKo09H, windefender.exe is created and executed by ShellExecute.



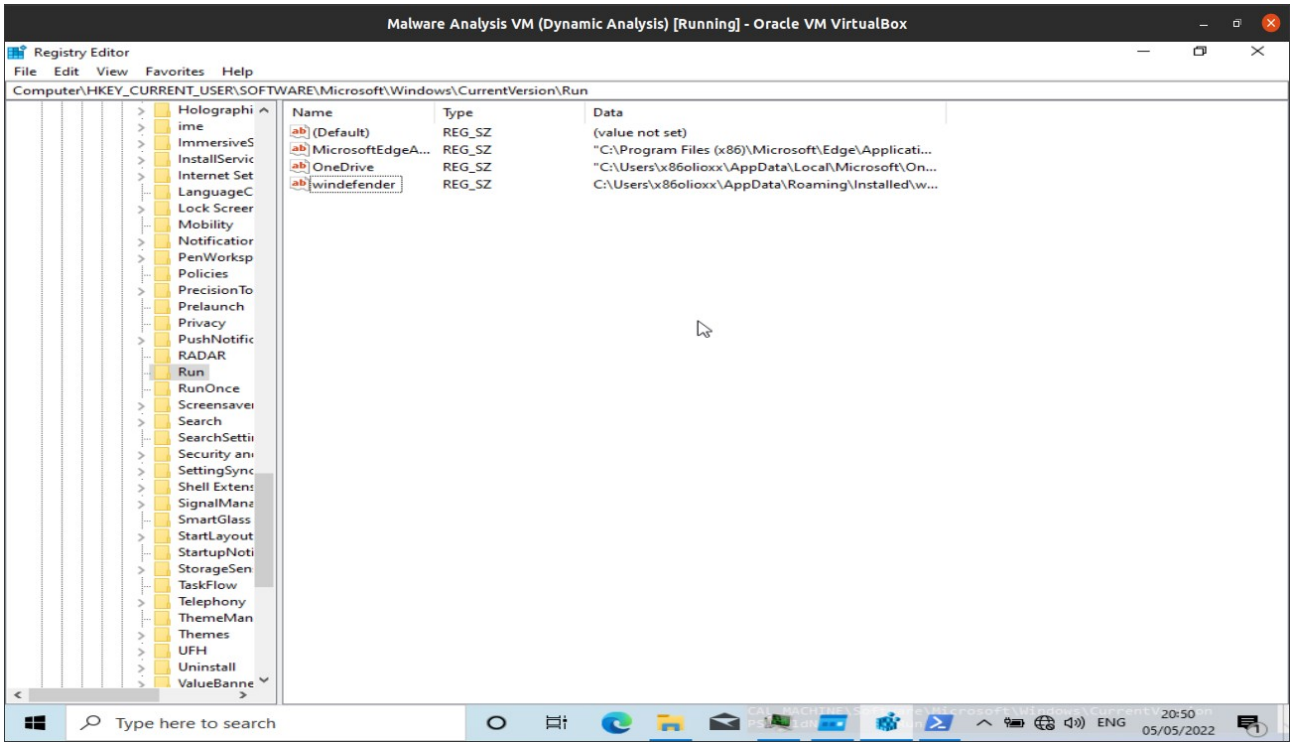
Process Hacker showing windefender.exe being created and executed



Password to decrypt the malware



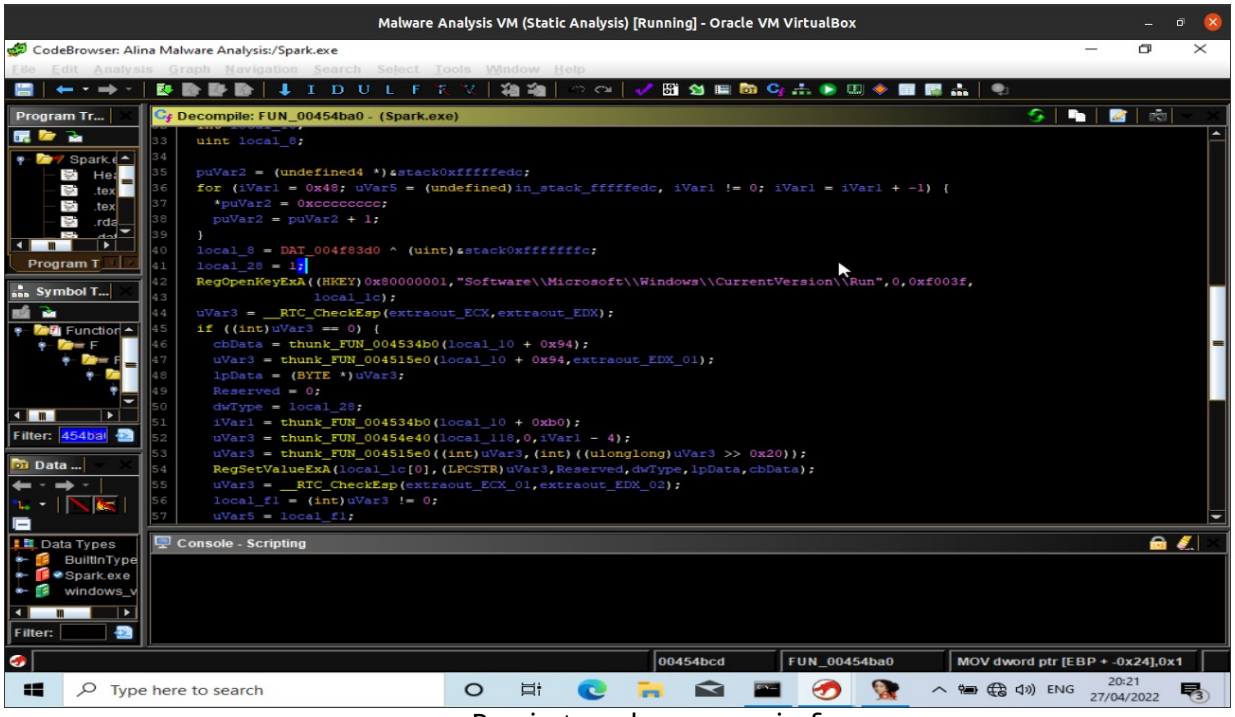
PowerShell display of the registry entry



windefender.exe

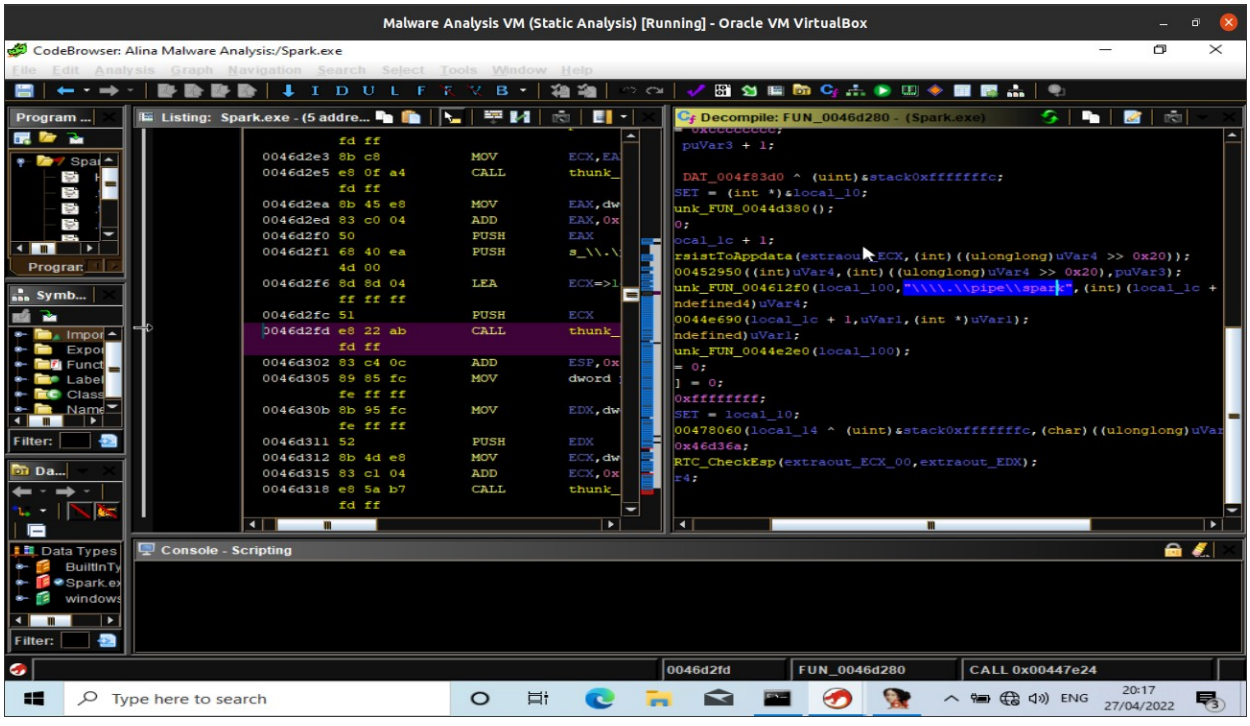
Manual Code Analysis(Dissasembly)

In order to achieve persistence, the malware(Spark.exe) add some entries(programs) into the Run registry keys. \\Software\\Microsoft\\Windows\\CurrentVersion\\Run



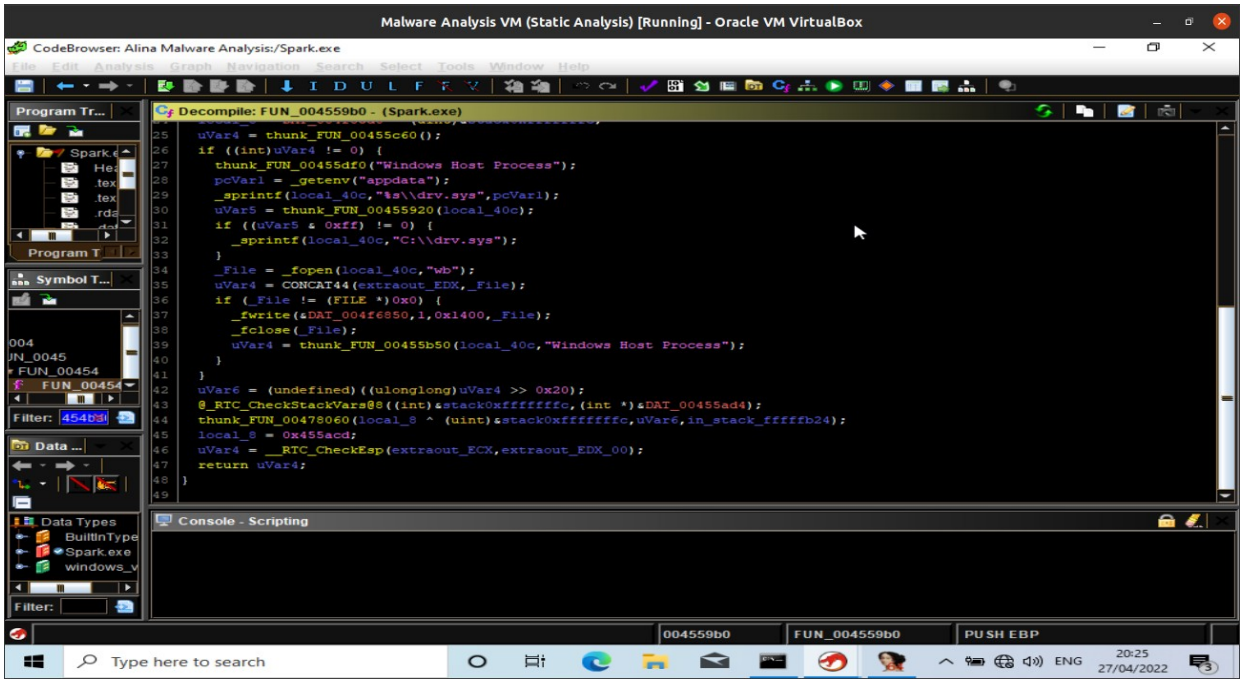
Registry key openinf

Spark.exe also utilizes a named pipe which is an **Inter-Process Communication** mechanism. The IPC allows threads to communicate together.



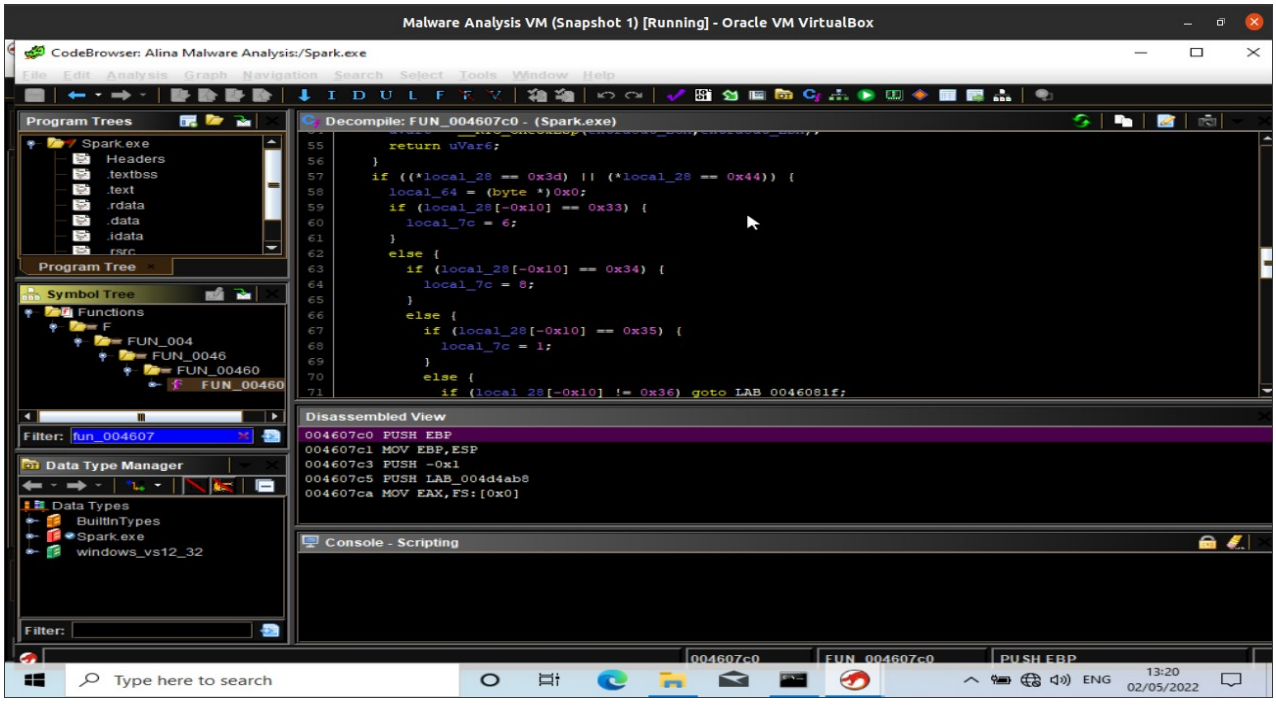
Named Pipe

The malware checks if the user has administrative priviledges in order to implement the kernel driver as a service. If they(user) do, the malware delete previous rootkit services, and then writes a new rootkit to disk, and finally, it creates a new service with the rootkit. The service name is "***Windows Host Process***", it is installed at "C:\" or %APPDATA% with a name of "drv.sys".



Rootkit Service

In order to authenticate the card numbers read from memory, it utilizes the Luhn Algorithm.



Implementation of Luhn Algorithm

Indicator of Compromise

Hashes

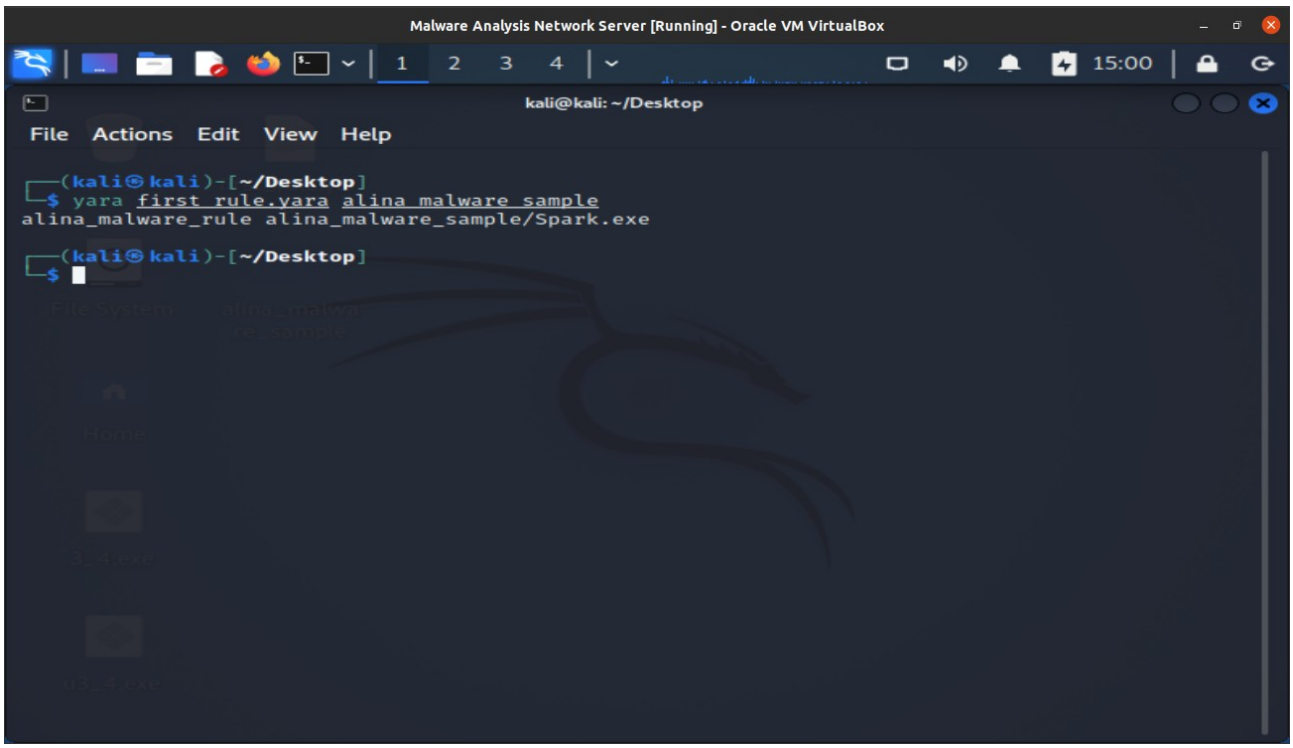
Hashes	File Name	File Type
d431f54201251619c07e4d5bf39e01cd (MD5) 1fabbd3d6fb5bf868ef07be4774649c4dd3f90959ef1e44 77edd08f96de47f03 (SHA256)	Spark	.exe
553d1afa824c34f348f8c53d1b043d3b671d946a (SHA1) 145a50d309bc9397baabf707aa396d4e (MD5) 905170f460583ae9082f772e64d7856b8f609078af9823e 9921331852fd07573 (SHA256) 72543a155d47a0845ee42fdbf9dfc93226effb11 (SHA1)	drv	.sys

Domains

adobeflasherup1[.]com
javaoracle2[.]ru

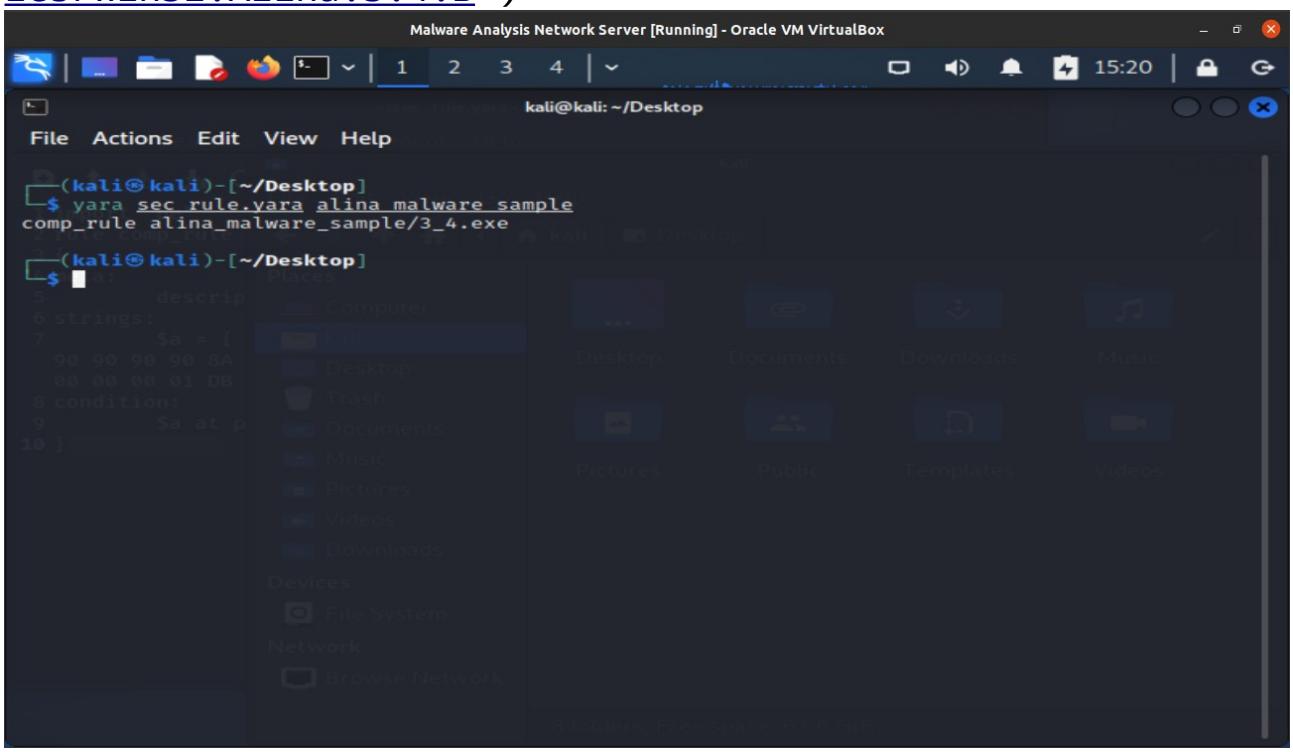
Yara Rule

After analyzing the malware the next step is to create Yara rules to help with identifying and classifying the malware sample.
The first rule identifies the Spark.exe file.



Yara Rule (Spark.exe)

Sometimes a malware might be packed, it is also helpful to identify packed malware. The below malware was downloaded from theZoo GitHub repository. (https://github.com/ytisf/theZoo/tree/master/malware/Binaries/Win32.Alina.3.4.B)



Compressed malware.