

**Alina Malware Analysis Report**  
**By Olotu Praise Jah**

**May 2022**

# Contents

---

Executive Summary.....

Analyzing the Attack .....

    Spark .....

    drv .....

Indicators of Compromise .....

Yara Rule .....

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'
```

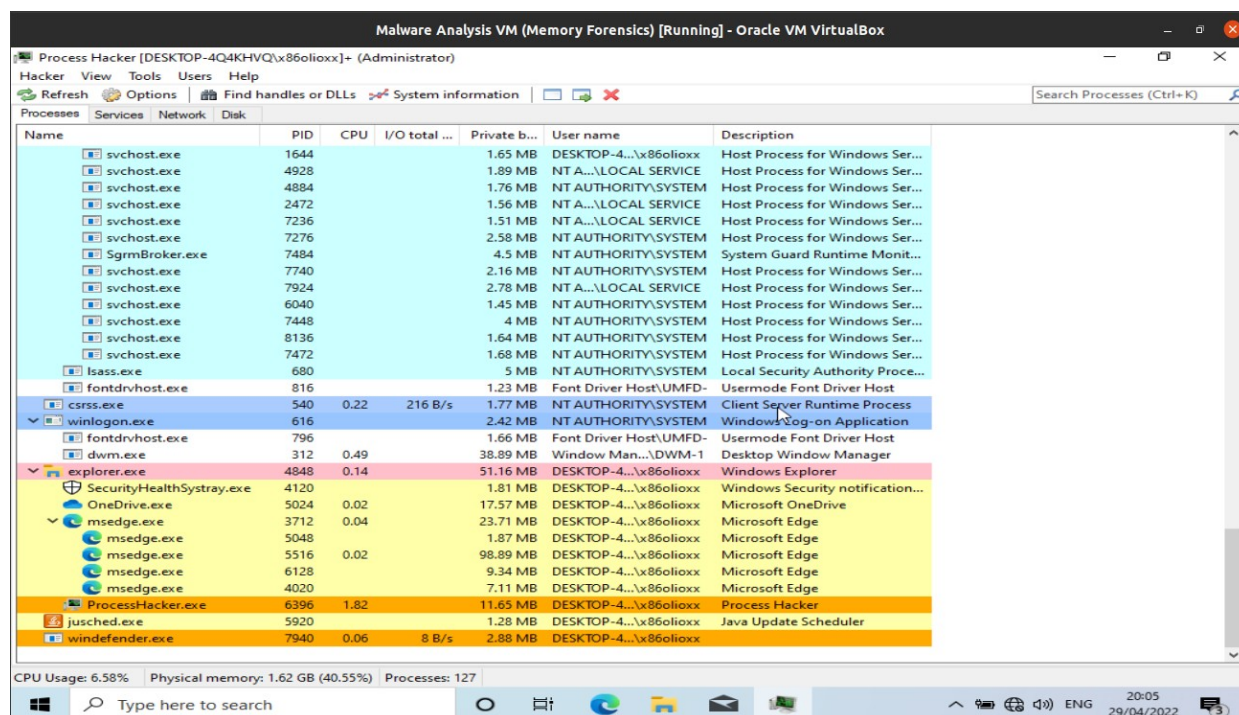
```
b 'SHELL32.DLL'
```

```
b 'URLMON.DLL'
```

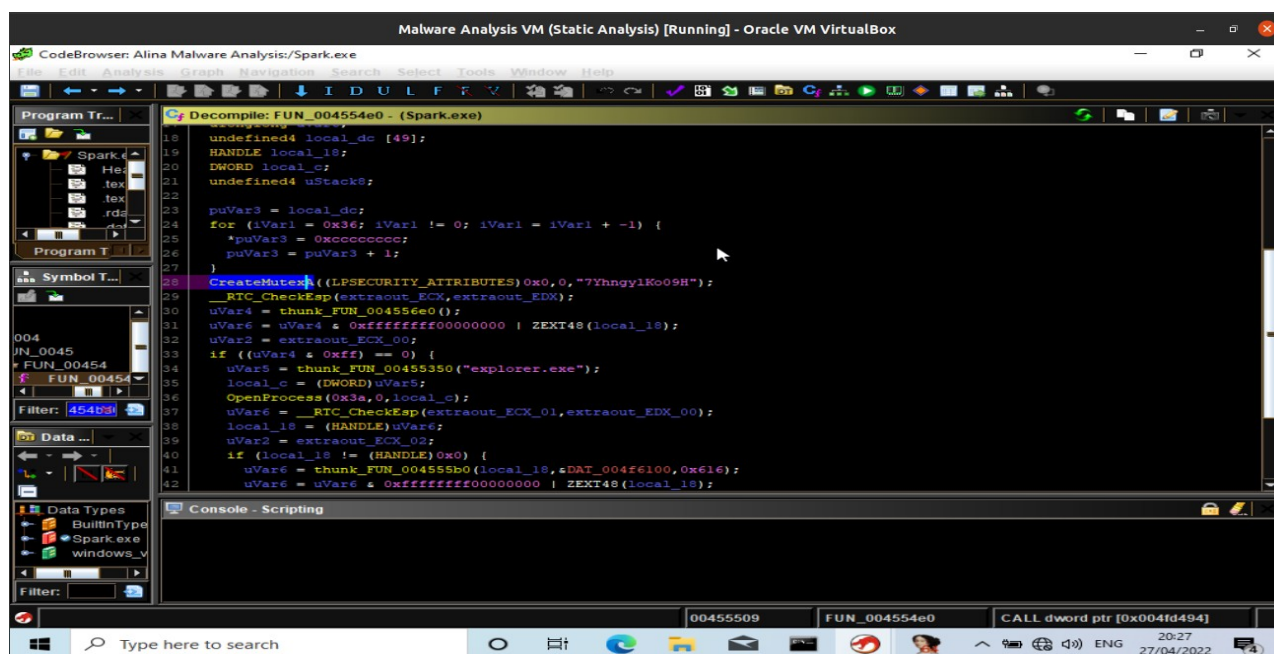
Utilizing the strings tool that comes with Linux on Spark.exe displays the list of legitimate programs to avoid when scraping card information, the user that compiled the malware, the C2C servers used by the malware, default location of the rootkit once initialized etc

# Dynamic Analysis

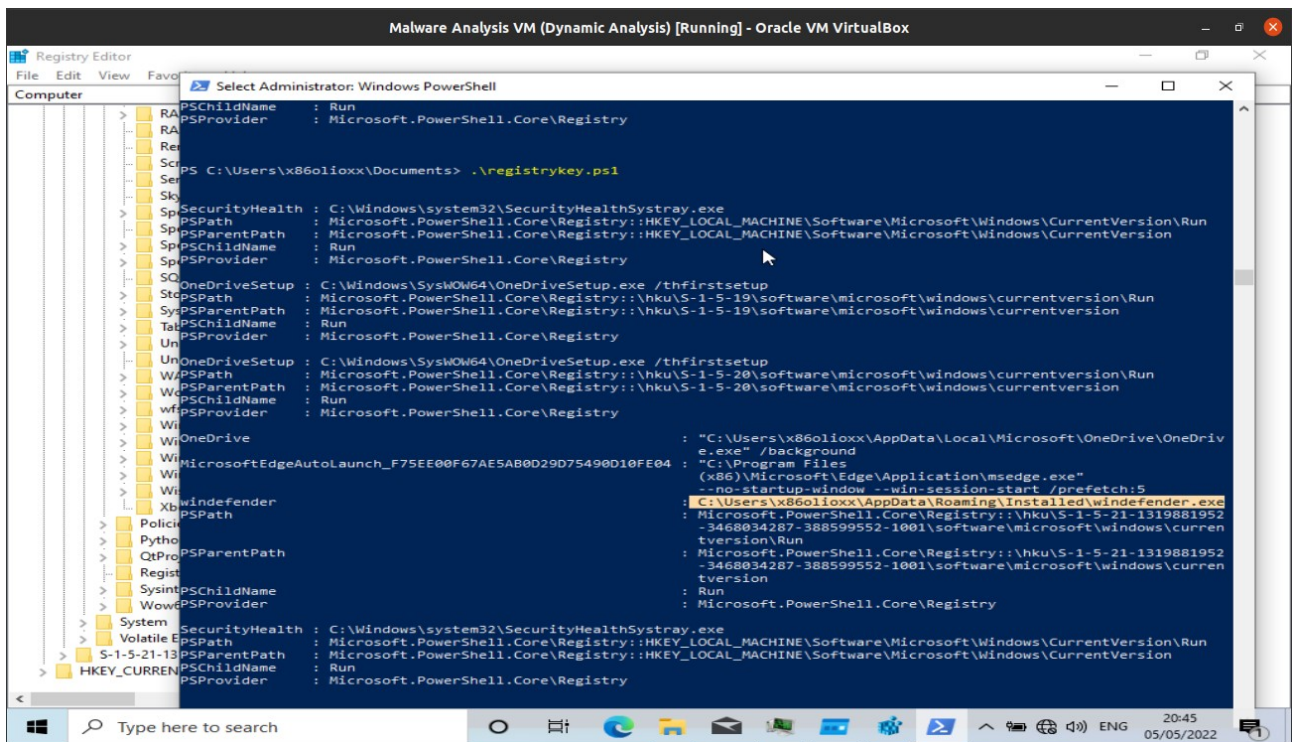
When Spark.exe is running on a machine, it installed(saves) an encrypted copy of itself in %APPDATA%\ntkrnl%, then after it has been decrypted with a hardcoded password of 7YhngylKo09H, windefender.exe is created and executed by ShellExecute.



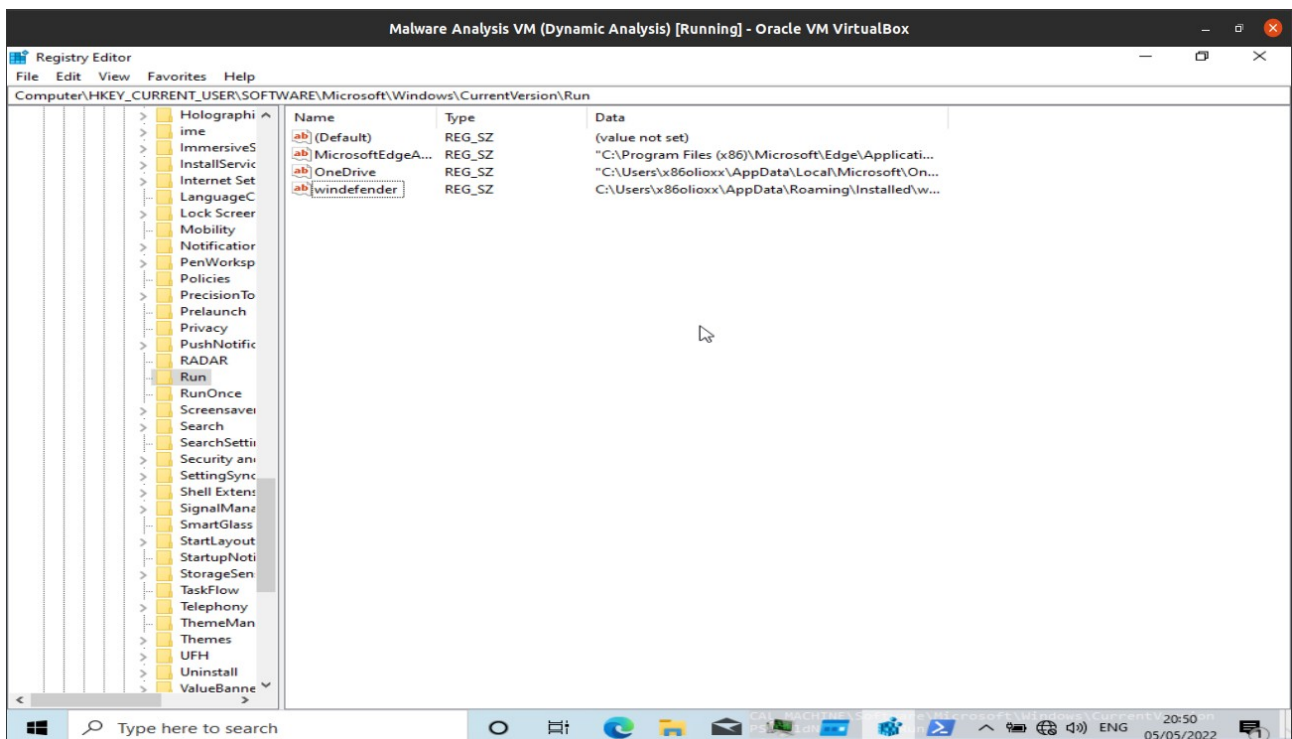
Process Hacker showing windefender.exe being created and executed



Password to decrypt the malware



PowerShell display of the registry entry

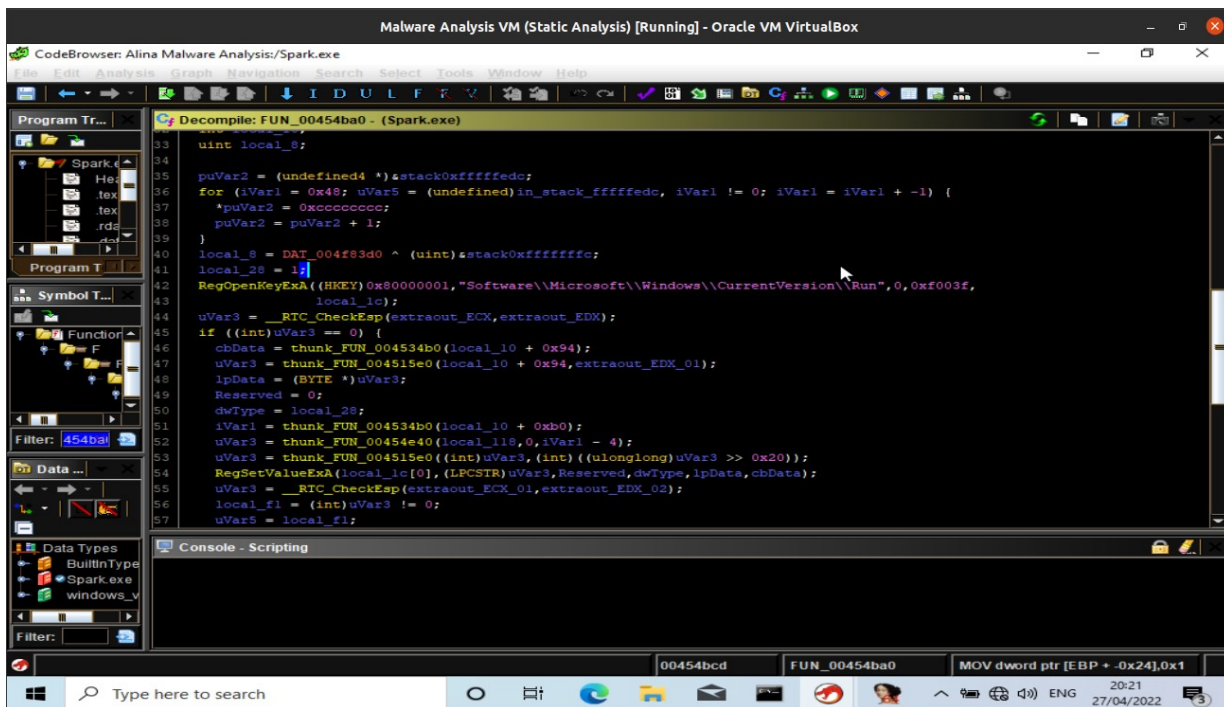


windefender.exe



# Manual Code Analysis(Dissassembly)

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



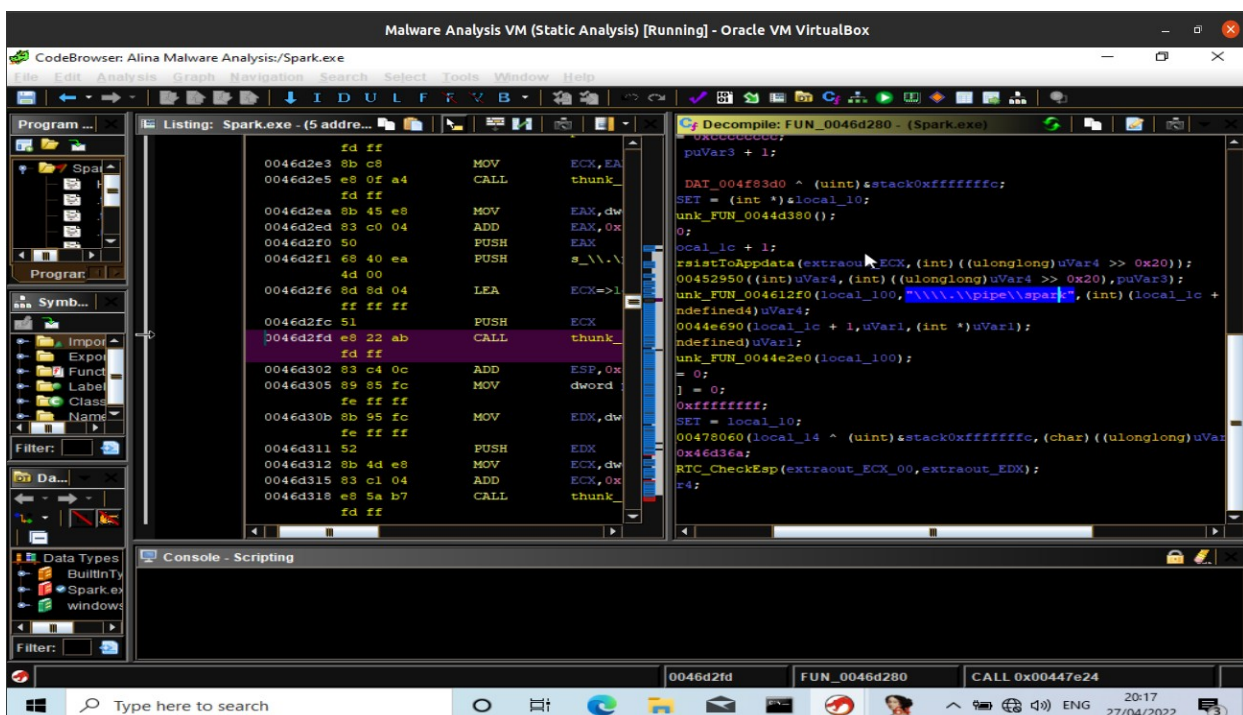
```
CodeBrowser: Alina Malware Analysis/Spark.exe
File Edit Analysis Graph Navigation Search Select Tools Window Help

Program Tr...
C:\Decompile: FUN_00454ba0 - (Spark.exe)
33  uint local_8;
34
35  puVar2 = (undefined4 *) &stack0xffffffff;
36  for (iVar1 = 0x48; uVar5 = (undefined) &in_stack_ffffdc, iVar1 != 0; iVar1 = iVar1 - 1) {
37      *puVar2 = 0xffffffff;
38      puVar2 = puVar2 + 1;
39  }
40  local_8 = DAT_004f83d0 ^ (uint) &stack0xffffffff;
41  local_28 = 1;
42  RegOpenKeyExA((HKEY) 0x80000001, "Software\\Microsoft\\Windows\\CurrentVersion\\Run", 0, 0xf003f,
43              local_1c);
44  uVar3 = _RTC_CheckEsp(extraout_ECX, extraout_EDX);
45  if ((int) uVar3 == 0) {
46      cbData = thunk_FUN_004534b0(local_10 + 0x94);
47      uVar3 = thunk_FUN_004515e0(local_10 + 0x94, extraout_EDX_01);
48      lpData = (BYTE *) uVar3;
49      Reserved = 0;
50      dwType = local_28;
51      iVar1 = thunk_FUN_004534b0(local_10 + 0xb0);
52      uVar3 = thunk_FUN_00454e40(local_118, 0, iVar1 - 4);
53      uVar3 = thunk_FUN_004515e0((int) uVar3, (int) ((ulonglong) uVar3 >> 0x20));
54      RegSetValueExA(local_1c[0], (LPCSTR) uVar3, Reserved, dwType, lpData, cbData);
55      uVar3 = _RTC_CheckEsp(extraout_ECX_01, extraout_EDX_02);
56      local_f1 = (int) uVar3 != 0;
57      uVar5 = local_f1;
58  }

Console - Scripting
00454bcd  FUN_00454ba0  MOV dword ptr [EBP + -0x24], 0x1
```

Registry key openinf

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



```
CodeBrowser: Alina Malware Analysis/Spark.exe
File Edit Analysis Graph Navigation Search Select Tools Window Help

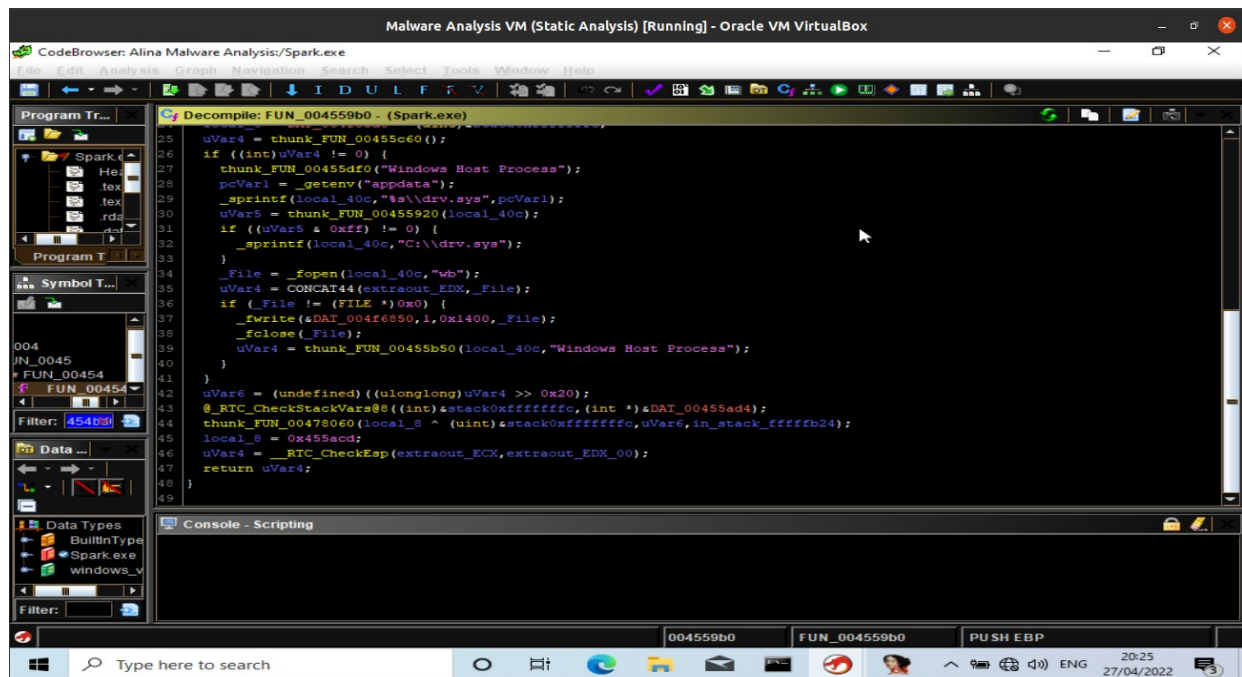
Program ...
Listing: Spark.exe - (5 address...)
0046d2e3 8b c8 MOV ECX, EAX
0046d2e5 e8 0f a4 CALL thunk_
0046d2ea 8b 45 e8 MOV EAX, dw
0046d2ed 83 c0 04 ADD EAX, 0x4
0046d2f0 50 PUSH EAX
0046d2f1 68 40 ea PUSH s_\\.\
0046d2f6 8d 8d 04 LEA ECX=>1
0046d2fc 51 PUSH ECX
0046d2fd e8 22 ab CALL thunk_
0046d302 83 c4 0c ADD ESP, 0xc
0046d305 89 85 fc MOV dword,
0046d30b 8b 95 fc MOV EDX, dw
0046d311 52 PUSH EDX
0046d312 8b 4d e8 MOV ECX, dw
0046d315 83 c1 04 ADD ECX, 0x4
0046d318 e8 5a b7 CALL thunk_

C:\Decompile: FUN_0046d280 - (Spark.exe)
0046d280:
puVar3 + 1;
DAT_004f83d0 ^ (uint) &stack0xffffffff;
SET = (int *) &local_10;
unk_FUN_0044d380();
0;
local_1c + 1;
RaiseToAppdata(extraout_ECX, (int) ((ulonglong) uVar4 >> 0x20));
00452950((int) uVar4, (int) ((ulonglong) uVar4 >> 0x20), puVar3);
unk_FUN_004612f0(local_100, "\\.\pipe\\spark.exe", (int) (local_1c +
undefined4) uVar4;
0044e690(local_1c + 1, uVar1, (int *) uVar1);
undefined4 uVar1;
unk_FUN_0044e2e0(local_100);
= 0;
j = 0;
0xffffffff;
SET = local_10;
00478060(local_14 ^ (uint) &stack0xffffffff, (char) ((ulonglong) uVar
0x46d36a;
RTC_CheckEsp(extraout_ECX_00, extraout_EDX);
E4;

Console - Scripting
0046d2fd  FUN_0046d280  CALL 0x00447e24
```

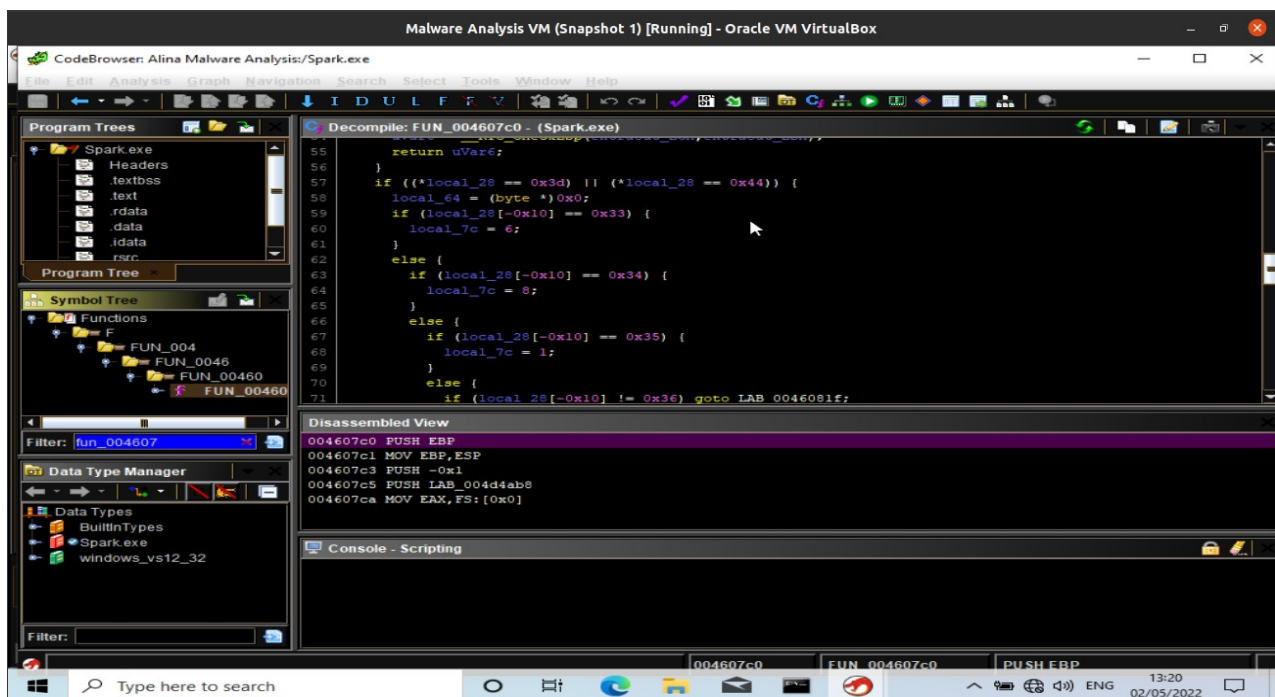
Named Pipe

The malware checks if the user has administrative privileges 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)		
1fabbd3d6fb5bf868ef07be4774649c4dd3f90959ef1e4477edd08f96de47f03 (SHA256)	Spark	.exe
553d1afa824c34f348f8c53d1b043d3b671d946a (SHA1)		
145a50d309bc9397baabf707aa396d4e (MD5)		
905170f460583ae9082f772e64d7856b8f609078af9823e9921331852fd07573 (SHA256)	drv	.sys
72543a155d47a0845ee42fdbf9dfc93226effb11 (SHA1)		

## Domains

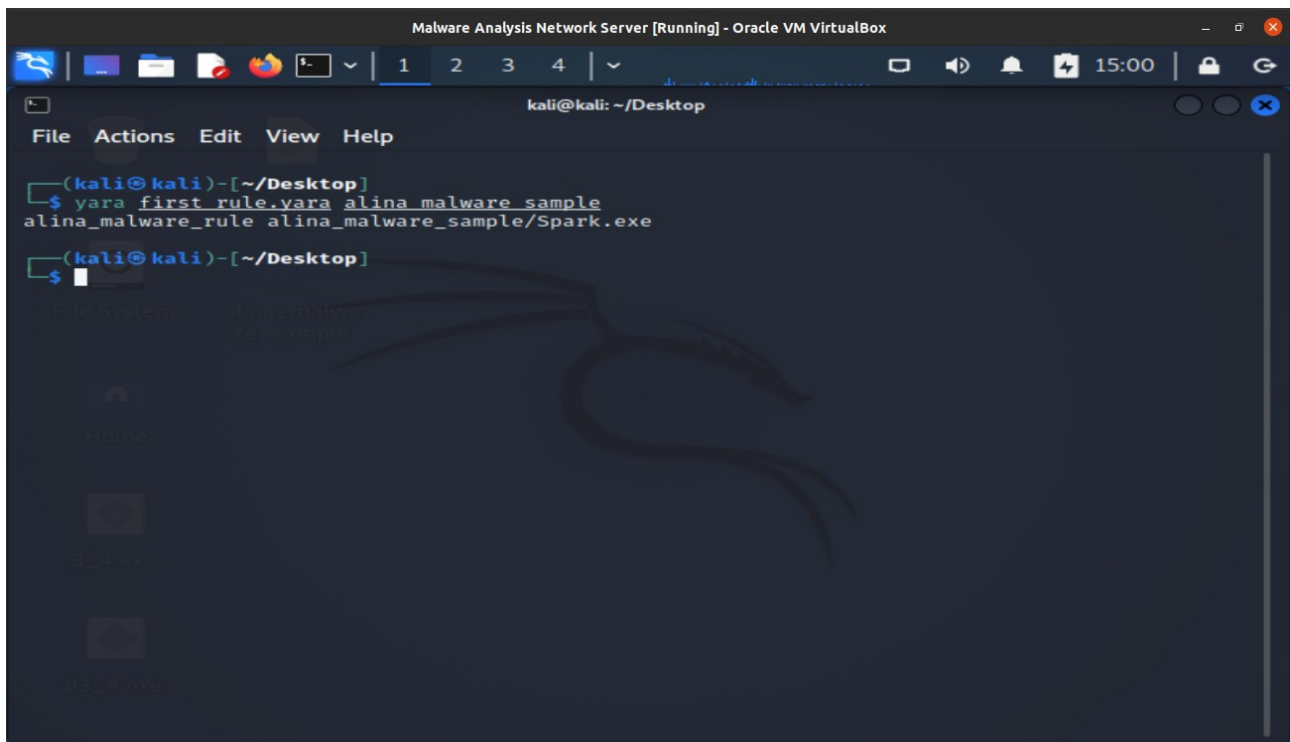
adobeflasherup1[.]com  
javaoracle2[.]ru

## Yara Rule

---

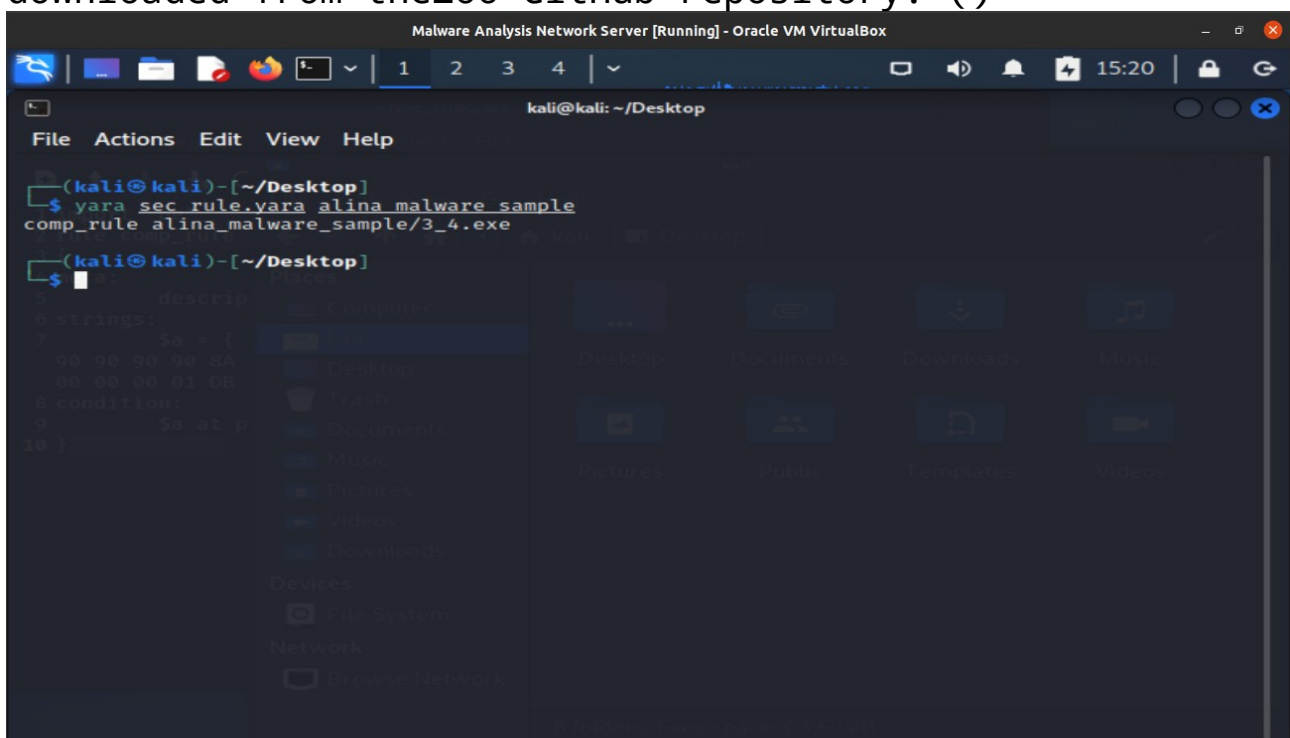
After analyzing the malware the next step is to create Yara rules to help with identifying and classifying the malware sample.

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. ()



Compressed malware.