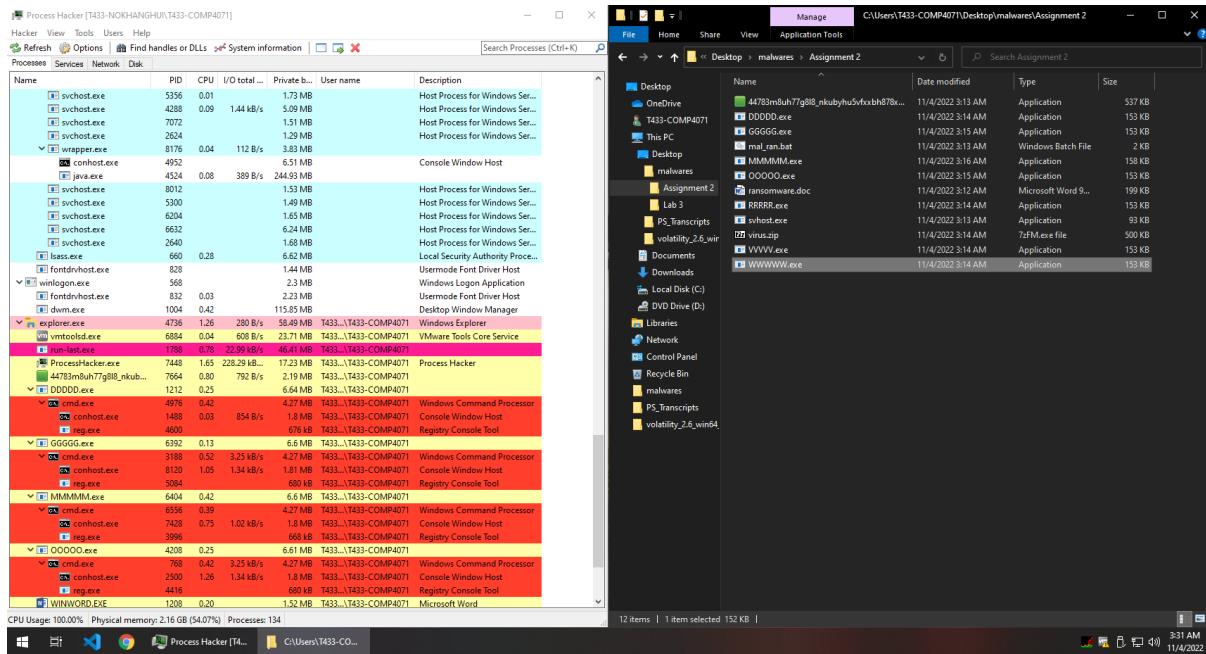


# Forensics Lab

**Author Name: Hui Nok Hang**

## Downloading and running the malwares

The malware executables from <https://app.box.com/s/btzloulf1f82ql4io5d4spbmy19i1ror> are downloaded into the Windows VM and run. They can be seen below in process hacker.



Memdump was then created and transferred to Kali Machine for investigation.

## Volatility Analysis

In Kali, volatility 3 modules pslist, pstree, psscan and malfind are run on the memdump we just created.

The screenshot shows two terminal windows side-by-side, both titled "t433-student@T433-NokHangHui-COMP4071: ~/Desktop/Tools/volatility3".

**Left Terminal:**

```
[t433-student@T433-NokHangHui-COMP4071:~/Desktop/Tools/volatility3]
$ sudo python3 vol.py -f memdump.mem windows.pslist.pList > plist.txt
[sudo] password for t433-student:
[t433-student@T433-NokHangHui-COMP4071:~/Desktop/Tools/volatility3]
$ sudo python3 vol.py -f memdump.mem windows.malfind.Malfind > malfind.txt
[t433-student@T433-NokHangHui-COMP4071:~/Desktop/Tools/volatility3]
$
```

**Right Terminal:**

```
[t433-student@T433-NokHangHui-COMP4071:~/Desktop/Tools/volatility3]
$ head plist.txt
Volatility 3 Framework 2.4.0

PID PPID ImageFileName Offset(V) Threads Handles SessionId Wow64 CreateTime ExitTime F
ile output

4 0 System 0xe38e27e7fe0+0 152 - N/A False 2022-11-04 05:27:52.000000 N/A Disa
bled
92 4 Registry 0xe38e27e7fe0+0 4 - N/A False 2022-11-04 05:27:46.000000 N/A Disa
bled
312 4 smss.exe 0xe38e26f7304+0 2 - N/A False 2022-11-04 05:27:52.000000 N/A Disa
bled
428 416 csrss.exe 0xe38e26f7041+0 10 - 0 False 2022-11-04 05:27:59.000000 N/A Disa
bled
504 496 csrss.exe 0xe38e26fb7200+0 13 - 1 False 2022-11-04 05:27:59.000000 N/A Disa
bled
524 516 wininit.exe 0xe38e2bfc3240+0 1 - 0 False 2022-11-04 05:27:59.000000 N/A Disa
bled
isabled
[...]
```

**Bottom Terminal:**

```
[t433-student@T433-NokHangHui-COMP4071:~/Desktop/Tools/volatility3]
$ head malfind.txt
Volatility 3 Framework 2.4.0

PID Process Start VPN End VPN Tag Protection CommitCharge PrivateMemory File output Hexd
ump Disasm

3408 s1host.exe 0x7ff7c0460000 0x7ff7c07e9fff VadS PAGE_EXECUTE_READWRITE 906 1 Disabled
4d 50 00 00 00 00 00 MZ.....
04 00 00 00 ff ff 00 00 .....
b8 00 00 00 00 00 00 00 .....
40 00 00 00 00 00 00 00 .....!
```

```
(t433-student㉿T433-NokHangHui-COMP4071) [~/Desktop/Tools/volatility3]
File Actions Edit View Help

(t433-student㉿T433-NokHangHui-COMP4071) [~/Desktop/Tools/volatility3]
$ head pscan.txt
Volatility 3 Framework 2.4.0

PID PPID ImageFileName Offset(V) Threads Handles SessionId Wow64 CreateTime ExitTime F
line output

301124 2824 SenseNdr.exe 0xa0f0346e0080 5 - 0 False 2022-11-04 07:52:40.000000 N/A
disabled
        0 System 0x38e27ef7f040 152 - N/A False 2022-11-04 05:27:52.000000 N/A
        bled
92 4 Registry 0x38e27ec2080 4 - N/A False 2022-11-04 05:27:46.000000 N/A
disabled
1736 4 MemCompression 0xe38e27ede040 142 - N/A False 2022-11-04 05:28:02.000000 N/A
disabled
1676 640 svchost.exe 0x38e27f18088 9 - 0 False 2022-11-04 05:28:02.000000 N/A
disabled
1886 640 svchost.exe 0x38e27fc9c080 2 - 0 False 2022-11-04 05:28:02.000000 N/A
disabled
[...]
$ head psreve.txt
Volatility 3 Framework 2.4.0

PID PPID ImageFileName Offset(V) Threads Handles SessionId Wow64 CreateTime ExitTime F
line output

4 0 System 0x38e27ef7f040 152 - N/A False 2022-11-04 05:27:52.000000 N/A
4 312 4 SenseNdr.exe 0x38e282f73040 2 - N/A False 2022-11-04 05:27:52.000000 N/A
+ 92 4 Registry 0x38e27ec2080 4 - N/A False 2022-11-04 05:27:46.000000 N/A
+ 1736 4 MemCompression 0x38e27ede040 142 - N/A False 2022-11-04 05:28:02.000000 N/A
+ 428 416 csrss.exe 0x38e2b704140 10 - 0 False 2022-11-04 05:27:59.000000 N/A
504 496 csrss.exe 0x38e2b7b7200 13 - 1 False 2022-11-04 05:27:59.000000 N/A
[...]
$ 

(t433-student㉿T433-NokHangHui-COMP4071) [~/Desktop/Tools/volatility3]
File Actions Edit View Help

(t433-student㉿T433-NokHangHui-COMP4071) [~/Desktop/Tools/volatility3]
$ sudo python3 vol.py -f memdump.mem windows.psTree > psreve.txt
[...]
(t433-student㉿T433-NokHangHui-COMP4071) [~/Desktop/Tools/volatility3]
$ sudo python3 vol.py -f memdump.mem pscan > pscan.txt
[...]
(t433-student㉿T433-NokHangHui-COMP4071) [~/Desktop/Tools/volatility3]
$ sudo python3 vol.py -f memdump.mem psscan > psscan.txt
[...]
(t433-student㉿T433-NokHangHui-COMP4071) [~/Desktop/Tools/volatility3]
$ sudo python3 vol.py -f memdump.mem psscan > psscan.txt
[...]
(t433-student㉿T433-NokHangHui-COMP4071) [~/Desktop/Tools/volatility3]
$ sudo password for t433-student:
[...]
(t433-student㉿T433-NokHangHui-COMP4071) [~/Desktop/Tools/volatility3]
$ 
```

## Malfind

Malfind has a long list of outputs of potentially suspicious processes.

Trimming down the file with grep and simple regex, their process ID and names are:

3408	sihost.exe
3436	svchost.exe
3548	svchost.exe
3640	taskhostw.exe
5188	svchost.exe
5688	RuntimeBroker.
5916	RuntimeBroker.
6884	vmtoolsd.exe ( <b>Eliminated</b> )
2072	svchost.exe
4928	dllhost.exe
5212	ApplicationFra
8100	dllhost.exe
5044	svchost.exe
7448	ProcessHacker. ( <b>Eliminated</b> )
12828	svchost.exe
16068	SearchApp.exe

All of these processes are named after legit processes in the windows environments. However, malwares are known to disguise themselves as legit process names, so we definitely cannot dismiss these findings. However, we can reasonably conclude that vmtoolsd.exe and ProcessHacker. are false positives, since these are most likely related to the assignment and virtual machine and they are too niche to be used as a disguise.

## Strings analysis

After getting a list of our target processes, now is the time to create the memory dump for these specific processes and doing strings analysis on them.

First, create yara rules in the thor-lite/custom-signatures/yara directory.

```
[t433-student@T433-NokHangHui-COMP4071] -[~/Desktop/Tools/volatility3]
ls -l ./greas pid
-rw-r--r-- 1 t433-student 1000 100000000.dmp
pid_12828.nokhang
pid_12828.nokhang
pid_15068.0x0.dmp
pid_15068.nokhang
pid_2072.0x7f7edee60000.dmp
pid_3408.0x7ff7e1b70000.dmp
pid_3408.nokhang
pid_3436.0x7f7edee60000.dmp
pid_3436.nokhang
pid_3548.0x7f7edee60000.dmp
pid_3548.nokhang
pid_3640.0x7f7edee60000.dmp
pid_3640.nokhang
pid_4928.0x7f718ff0000.dmp
pid_5044.0x7f7edee60000.dmp
pid_5044.nokhang
pid_5188.0x7f7edee60000.dmp
pid_5188.nokhang
pid_5188.0x7f7edee60000.dmp
pid_5212.0x7f7edee60000.dmp
pid_5212.nokhang
pid_5688.0x7ff7d82f0000.dmp
pid_5688.nokhang
pid_5916.0x7ff7d82f0000.dmp
pid_5916.nokhang
pid_8100.0x7f718ff0000.dmp
pid_8100.nokhang

[t433-student@T433-NokHangHui-COMP4071] -[~/Desktop/Tools/volatility3]
cd ..\..\Thor-lite\custom-signatures\yara

[t433-student@T433-NokHangHui-COMP4071] -[~/..\\Tools\\Thor-lite\\custom-signatures\\yara]
touch pid-(2072,3408,3436,3548,3640,4928,5188,5212,5688,5916,8100)-nokhang.yar

[t433-student@T433-NokHangHui-COMP4071] -[~/..\\Tools\\Thor-lite\\custom-signatures\\yara]
cd ..

pid-2072-nokhang.yar pid-3548-nokhang.yar pid-5188-nokhang.yar pid-5916-nokhang.yar
pid-3408-nokhang.yar pid-3640-nokhang.yar pid-5212-nokhang.yar pid-8100-nokhang.yar
pid-3436-nokhang.yar pid-4928-nokhang.yar pid-5688-nokhang.yar

[t433-student@T433-NokHangHui-COMP4071] -[~/..\\Tools\\Thor-lite\\custom-signatures\\yara]
```

```
t433-student@T433-NokHangHui-COMP4071: ~/Desktop/Tools/volatility3
$ sudo python3 vol.py -f memdump.mem windows.plist.PsList --pid 5212 --dump
VOLATILITY 3 Framework 2.4.0
Progress: 100.00          PDB scanning finished
PID          PPID        ImageFileName      Offset(V)    Threads Handles SessionId      Wow64      CreateTime      ExitTime      File output
5212         776        ApplicationFrame 0xe38e2e780800 8      -       1      False   2022-11-04 07:02:36.0000000  N/A      pid.5212.0*7ff778a10000.dmp

[t433-student@T433-NokHangHui-COMP4071: ~/Desktop/Tools/volatility3]
$ sudo strings pid.5212.0*7ff778a10000.dmp > pid.5212.nokhang

[t433-student@T433-NokHangHui-COMP4071: ~/Desktop/Tools/volatility3]
$ cat pid.5212.nokhang
[This program cannot be run in DOS mode.
Readelf
.Rela
.text
._impnsiv
.rdata
.B.data
.B.strtab
.B.rsrc
.B.reloc
.t$ UH
D$PH
.DH
D$PH
D$OL
Mph3
D$RH
D$RH
D$RH
I$H
I$UNAWH
F9<Bu
FA9<>Bu
FA9<<Bu
D$RH
D$PH
D$OL
D$PE3
L$KI
I$H
Mph3
A^_]
I$ UWAWH
FA9<>u
FA9<>u
FA9<>u
D$^_
D$PH
D$OL
D$PE3
I$H
I$H
Mph3
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
```

As we can see below, there are some interesting strings output in some of the processes. However, not every process outputs meaningful results.

```
t433-student@T433-NokHangHui-COMP4071: ~/Desktop/Tools/volatility3

File Actions Edit View Help
failureId
failureCount
function
FailureError
PartA_Privlags
willResult
HRESULT
fileName
lineNumber
module
failureType
message
threadId
threadStart
originatingContextId
originatingContextName
originatingContextMessage
currentContextId
currentContextName
currentContextMessage
FeatureUsage
featureId
featureVersion
featureBaseVersion
featureStage
enabled
kind
addend
FeatureVariantUsage
featureId
featureVersion
featureBaseVersion
featureStage
ended
VariantKind
variant
addend
Microsoft.Windows.Wil.FeatureLogging
Acls
RtlNtStatusToDosErrorNoTeb
RtlDlShutdownInProgress
RtlSubscribeWnFStateChangeNotification
RtlUnsubscribeWnFNotificationWaitForCompletion
pcshell\shell\applicationframe\host\classfactory.cpp
ApplicationFrameHost.pdb
.text
.text$di
.text$p01ApplicationFrameHost.exe!20_pr17
.text$e
.text$em$00
.text$np
.text$x
.text$yd
.text$y
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

```
t433-student@T433-NokHangHui-COMP4071: ~/Desktop/Tools/volatility3

File Actions Edit View Help
api-ms-win-core-synch-l1-1-0.dll
api-ms-win-core-heap-l1-1-0.dll
api-ms-win-core-errorhandling-l1-1-0.dll
api-ms-win-eventing-provider-l1-1-0.dll
api-ms-win-core-threadpool-l1-1-0.dll
api-ms-win-core-localization-l1-2-0.dll
api-ms-win-core-debug-l1-1-0.dll
api-ms-win-core-handle-l1-1-0.dll
api-ms-win-core-com-l1-1-0.dll
api-ms-win-core-rtlsupport-l1-1-0.dll
api-ms-win-core-profile-l1-1-0.dll
api-ms-win-core-sysinfo-l1-1-0.dll
combase.dll
DXVAdeclareAdapterRemovalSupport
drt.dll
malloc
_callnewh
??_except@?@EAA@?@EAEQEBD@Z
??_except@?@EAA@?@EAEQEBDH@Z
?HandleException@@UEBAPEB@Z
_CxxThrowException
memcp
memmove
memcmp
memset
<xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Copyright (c) Microsoft Corporation -->
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
  <assemblyIdentity
    version="5.1.0.0"
    processorArchitecture="amd64"
    name="Microsoft.ApplicationFrameHost"
    type="win32"
  />
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
    <security>
      <requestedPrivileges>
        <requestedExecutionLevel
          level="asInvoker"
          uiAccess="false"
        />
      </requestedPrivileges>
    </security>
  </trustInfo>
  <application xmlns="urn:schemas-microsoft-com:asm.v3">
    <windowsSettings>
      <dpiAware xmlns="http://schemas.microsoft.com/SMI/2005/WindowsSettings">True/Pm</dpiAware>
    </windowsSettings>
  </application>
</assembly>
```

(t433-student@T433-NokHangHui-COMP4071) - [~/Desktop/Tools/volatility3]

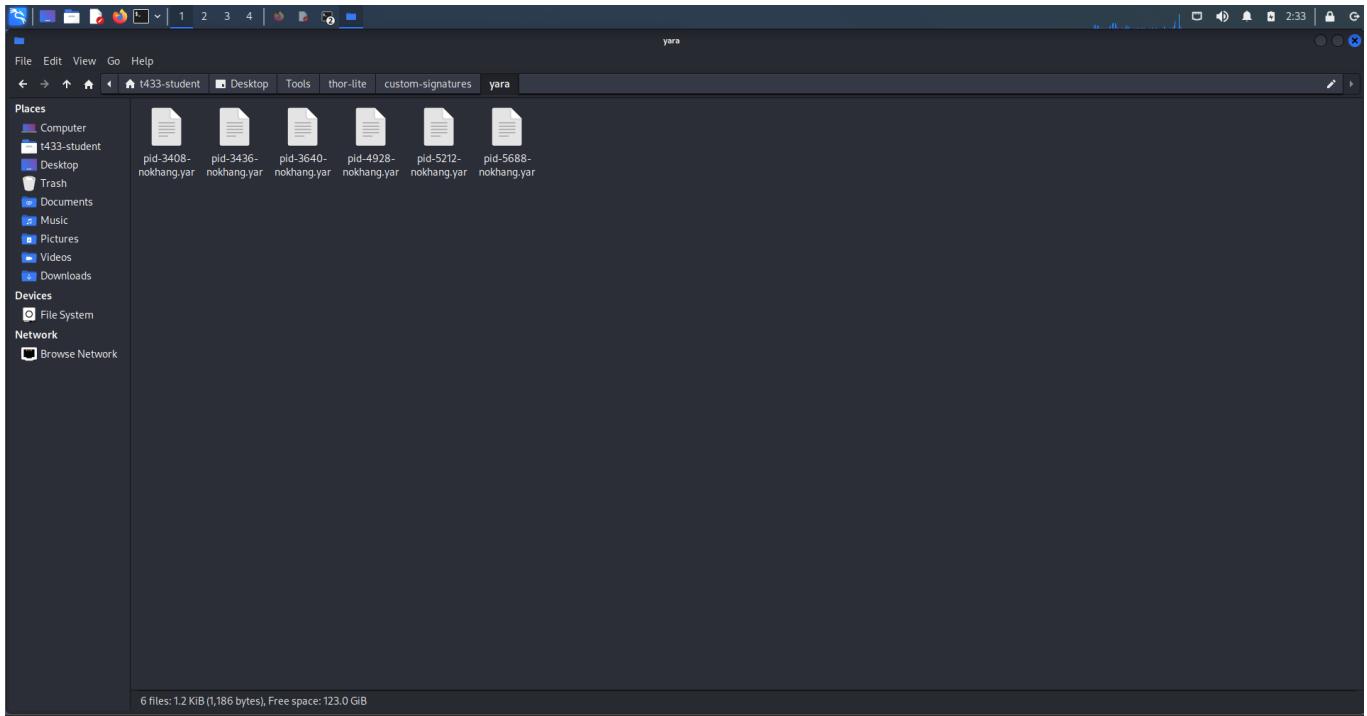
I repeated the steps for all of the listed processes above. Process 5044, 12828, 16068 yield no result from strings, therefore they are being excluded from having a yara rule created for them.

Since svchost.exe, RunTimeBroker and dllhost.exe have multiple instances in the list, which all yield the same strings result, I am only keeping the first instance of each of these processes in the list.

```
t433-student@T433-NokHangHui-COMP4071: ~/Desktop/Tools/volatility3
File Actions Edit View Help
ls | grep pid
pid.11282.0x1400000000.dmp
pid.11282.nokhang
pid.11668.0x0.dmp
pid.11668.nokhang
pid.2072.0xfffffedee60000.dmp
pid.2072.nokhang
pid.3340.0x7ffff7db70000.dmp
pid.3340.nokhang
pid.3436.0xfffffedee60000.dmp
pid.3436.nokhang
pid.3548.0xfffffedee60000.dmp
pid.3548.nokhang
pid.3640.0xfffffedee60000.dmp
pid.3640.nokhang
pid.4928.0xfffff18ff0000.dmp
pid.4928.nokhang
pid.5304.0xfffffedee60000.dmp
pid.5304.nokhang
pid.5188.0xfffffedee60000.dmp
pid.5188.nokhang
pid.5212.0xffff78a10000.dmp
pid.5212.nokhang
pid.5568.0xfffff270000.dmp
pid.5568.nokhang
pid.5916.0xffffd8f2f0000.dmp
pid.5916.nokhang
pid.8100.0xfffff18ff0000.dmp
pid.8100.nokhang
```

## Yara rules

Now that we have readable strings from every suspicious process we flagged, it is time to create yara rules to detect them in thor lite.



svchost.exe:

A screenshot of a YARA rule editor window titled "pid-3408-nokhang.yar - Mousepad". The window shows a YARA rule for "svchost". The code is as follows:

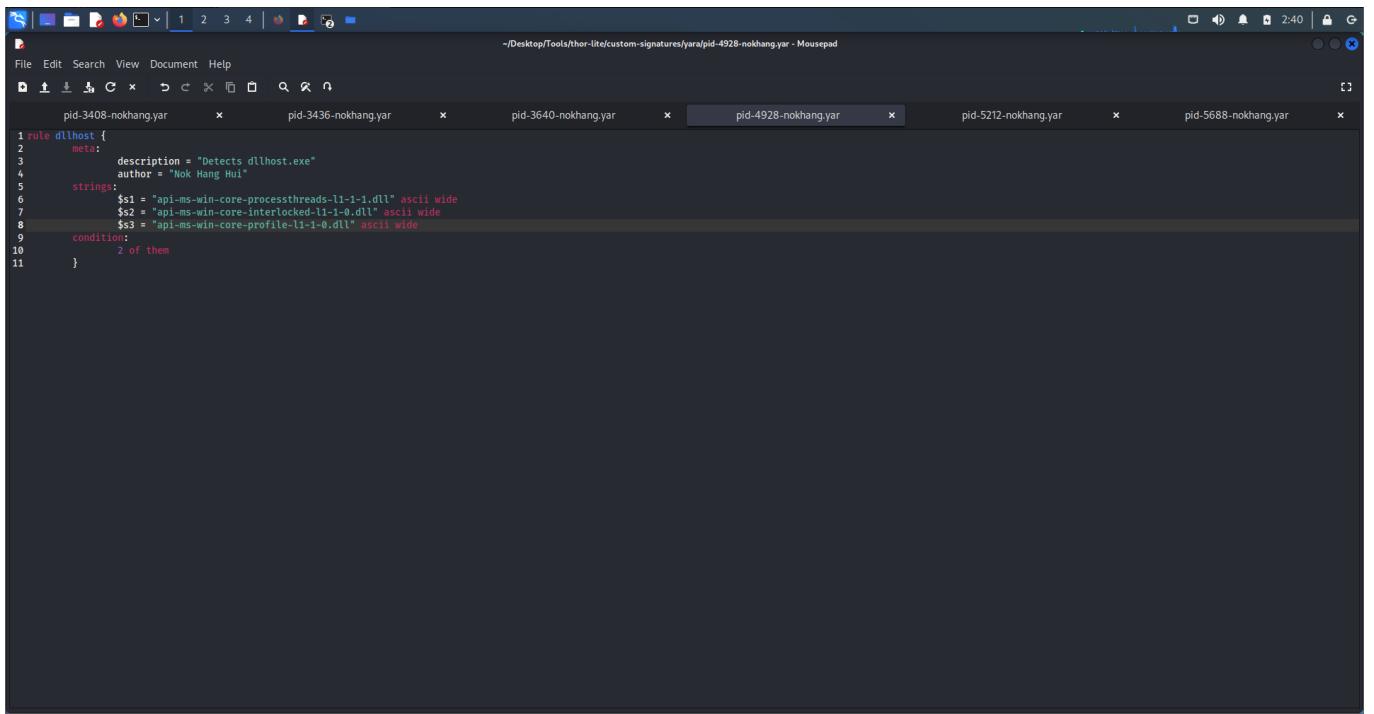
```
1 rule svchost {
2     meta:
3         description = "Detects svchost.exe"
4         author = "Nok Hang Hui"
5     strings:
6         $s1 = "SvchostPushServiceGlobal$Ex" ascii wide
7         $s2 = "h WATAUAVAWH" ascii wide
8         $s3 = "\$ UVAW$H" ascii wide
9     condition:
10        2 of them
11 }
```

## taskhostw.exe

A screenshot of a YARA rule editor window titled "pid-3640-nokhang.yar - Mousepad". The window shows a YARA rule for "taskhostw". The code is as follows:

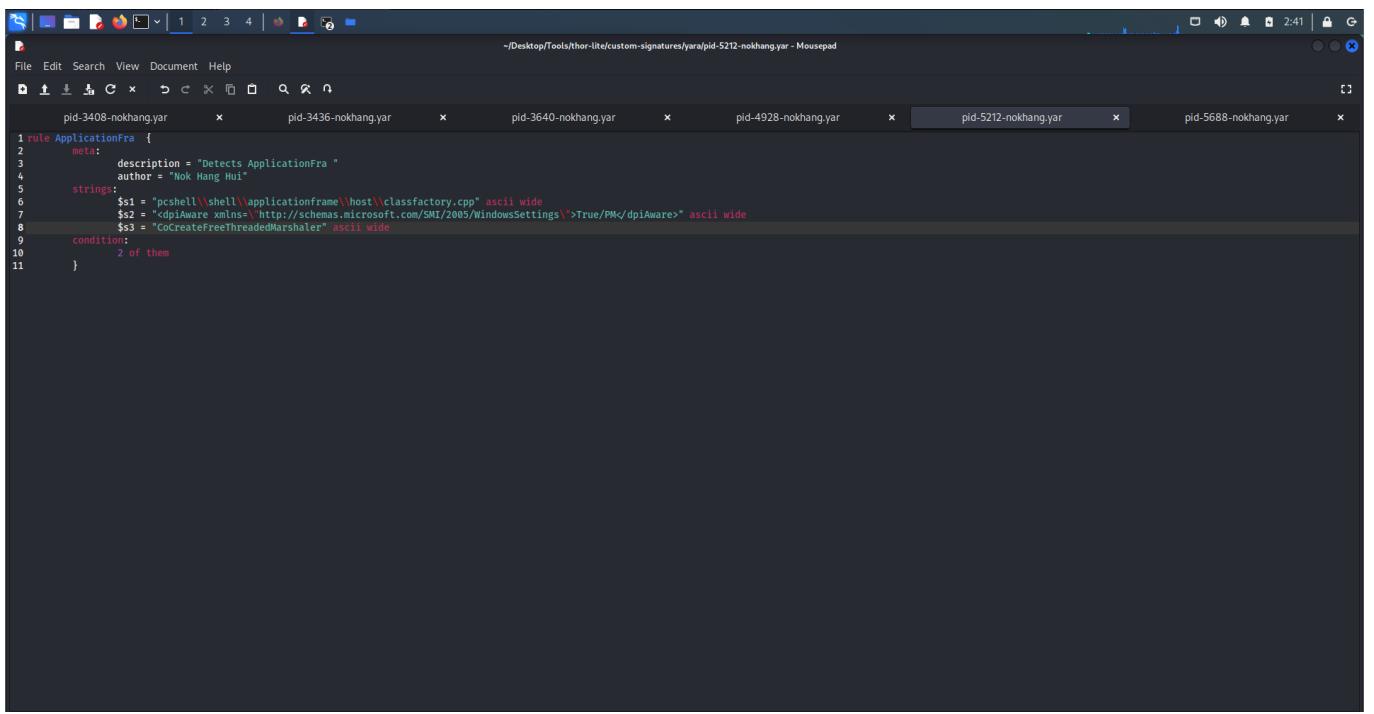
```
1 rule taskhostw {
2     meta:
3         description = "Detects taskhostw.exe"
4         author = "Nok Hang Hui"
5     strings:
6         $s1 = ".AVGenericException@mi@@" ascii wide
7         $s2 = ".AVinvalid_argument@std@@" ascii wide
8         $s3 = ".AVlogic_error@std@@" ascii wide
9     condition:
10        2 of them
11 }
```

## dllhost.exe:



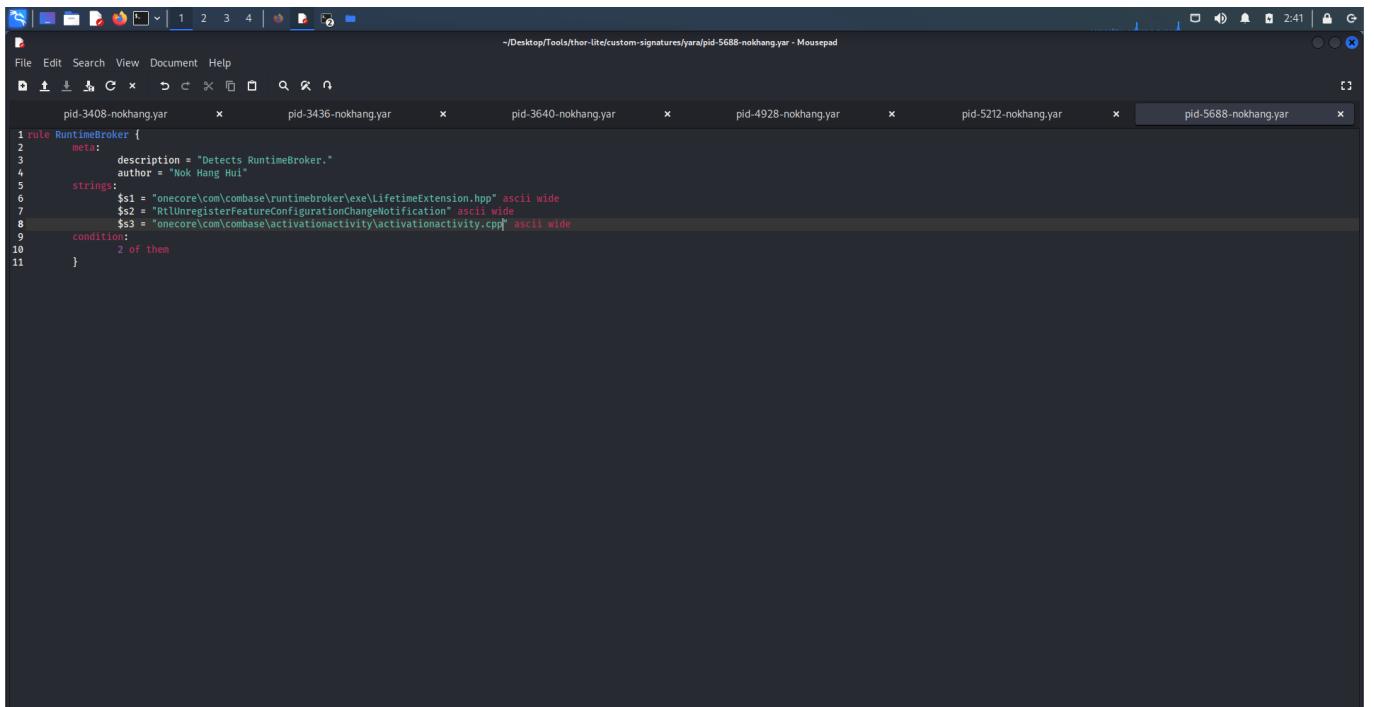
```
File Edit Search View Document Help
pid-3408-nokhang.yar x pid-3436-nokhang.yar x pid-3640-nokhang.yar x pid-4928-nokhang.yar x pid-5212-nokhang.yar x pid-5688-nokhang.yar x
-/Desktop/Tools/thor-lite/custom-signatures/yara/pid-4928-nokhang.yar - Mousepad
1 rule dllhost {
2     meta:
3         description = "Detects dllhost.exe"
4         author = "Nok Hang Hui"
5     strings:
6         $s1 = "$api-ms-win-core-processenvironment-l1-1-1.dll" ascii wide
7         $s2 = "$api-ms-win-core-interlocked-l1-1-0.dll" ascii wide
8         $s3 = "$api-ms-win-core-profile-l1-1-0.dll" ascii wide
9     condition:
10        2 of them
11 }
```

## ApplicationFra:



```
File Edit Search View Document Help
pid-3408-nokhang.yar x pid-3436-nokhang.yar x pid-3640-nokhang.yar x pid-4928-nokhang.yar x pid-5212-nokhang.yar x pid-5688-nokhang.yar x
-/Desktop/Tools/thor-lite/custom-signatures/yara/pid-5212-nokhang.yar - Mousepad
1 rule ApplicationFra {
2     meta:
3         description = "Detects ApplicationFra"
4         author = "Nok Hang Hui"
5     strings:
6         $s1 = "pshell\shell\applicationframe\host\classfactory.cpp" ascii wide
7         $s2 = "dpiAware xmlns="http://schemas.microsoft.com/SMI/2005/WindowsSettings\>True/FM</dpiAware" ascii wide
8         $s3 = "CoCreateFreeThreadedMarshaled" ascii wide
9     condition:
10        2 of them
11 }
```

## RunTimeBroker:

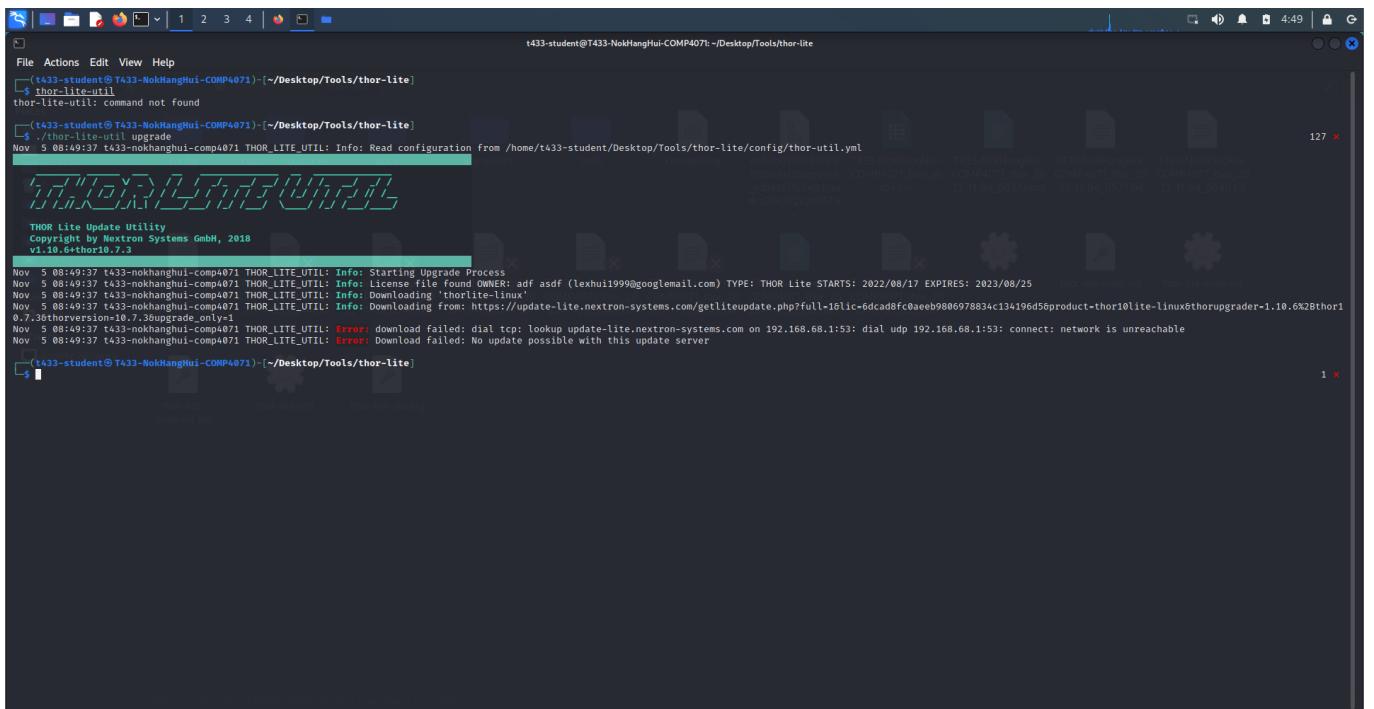


The screenshot shows a terminal window with several tabs open, each containing a YARA rule file. The tabs are labeled: pid-3408-nokhang.yar, pid-3436-nokhang.yar, pid-3640-nokhang.yar, pid-4928-nokhang.yar, pid-5212-nokhang.yar, and pid-5688-nokhang.yar. The content of the tabs is as follows:

```
1 rule RuntimeBroker {
2     meta:
3         description = "Detects RuntimeBroker."
4         author = "Nok Hang Hui"
5     strings:
6         $s1 = "onecore\com\cmbase\runtimebroker\exe\LifetimeExtension.hpp" ascii wide
7         $s2 = "RtlUnregisterFeatureConfigurationChangeNotification" ascii wide
8         $s3 = "onecore\com\cmbase\activationactivity\activationactivity.cpp" ascii wide
9     condition:
10        2 of them
11 }
```

## THOR lite

After creating all the respective yara rules, it is time to test it on our Kali machine.



The screenshot shows a terminal window with the following command and its output:

```
$ ./thor-lite-util upgrade
```

```
t433-student@T433-NokHangHui-COMP4071:~/Desktop/Tools/thor-lite
```

```
(t433-student@T433-NokHangHui-COMP4071) [~] /Desktop/Tools/thor-lite
```

```
thor-lite-util: command not found
```

```
(t433-student@T433-NokHangHui-COMP4071) [~] /Desktop/Tools/thor-lite
```

```
Nov 5 08:49:37 t433-nokhanghui-comp4071 THOR_LITE_UTIL: Info: Read configuration from /home/t433-student/Desktop/Tools/thor-lite/config/thor-util.yml
```

```
THOR Lite Update Utility
```

```
Copyright by Nextron Systems GmbH, 2018
```

```
v1.10.6+thor10.7.3
```

```
Nov 5 08:49:37 t433-nokhanghui-comp4071 THOR_LITE_UTIL: Info: Starting Upgrade Process
```

```
Nov 5 08:49:37 t433-nokhanghui-comp4071 THOR_LITE_UTIL: Info: License file found OWNER: adf asdf (lexhui1999@googlemail.com) TYPE: THOR Lite STARTS: 2022/08/17 EXPIRES: 2023/08/25
```

```
Nov 5 08:49:37 t433-nokhanghui-comp4071 THOR_LITE_UTIL: Info: Downloading 'thorlite-linux'
```

```
Nov 5 08:49:37 t433-nokhanghui-comp4071 THOR_LITE_UTIL: Info: Downloading from: https://update-lite.nextron-systems.com/getliteupdate.php?full=1&lic=6dcad8fc0aeeb9806978834c134196d56product=thor10lite-linux&thorupgrader=1.10.6%2Bthor10.7.3&thorversion=10.7.3&upgrade_only=1
```

```
Nov 5 08:49:37 t433-nokhanghui-comp4071 THOR_LITE_UTIL: Error: download failed: dial tcp: lookup update-lite.nextron-systems.com on 192.168.68.1:53: connect: network is unreachable
```

```
Nov 5 08:49:37 t433-nokhanghui-comp4071 THOR_LITE_UTIL: Error: Download failed: No update possible with this update server
```

```

t433-student@T433-NokHangHui-COMP4071: ~/Desktop/Tools/thor-lite
$ sudo ./thor-lite-linux-64 --customonly
[sudo] password for t433-student:
Notice Some modules and features are not available in Lite version and will be disabled
Notice This THOR Lite license permits non-commercial use only. It is strictly prohibited to sell THOR Lite or sell services that include the use of THOR Lite. For details, see the EULA in the ./docs folder. For a special license that covers these cases and suppresses this message, please contact our sales via https://www.nextron-systems.com/get-started/
[THOR] Thor Lite Version 10.7.3 (2022-07-27 16:44:51)
[THOR] (c) Nextron Systems GmbH
[THOR] Lite Version
[THOR] Scan Information
[THOR] Filescan 33
[THOR] Statistics
[THOR] Help
[THOR] Filters
[THOR] Hint 1
[THOR] Hint 2

```

> Scan Information

```

Info Thor Version 10.7.3
Info Thor Build: 11e6727eb83b (2022-07-27 16:44:51)
Info Run on system: T433-NokHangHui-COMP4071
Info Running as user: root
Info User has Admin rights: yes
Info Working Directory: /home/t433-student/Desktop/tools/thor-lite
Info Thor Scan started TIME: Sat Nov 5 02:52:47 2022 HOSTNAME: T433-NokHangHui-COMP4071
Info Effective argument list: [-customonly --dbfile /var/lib/thor/thor10-lite.db]
Info Platform: Kali GNU/Linux Rolling
Info Kernel: 5.15.0-kali2-amd64
Info Language: en_US.UTF-8
Info Locale: en_US.UTF-8
Info System Uptime: 0.00 days
Info CPU Count: 2
Info Memory in Megabytes: 3098
Info False positive filters applied: 0
Info False positive filters applied: 0 False positive filters TYPE: log filter
Info Signature Database: 2022/11/04-080717
Info Writing report file to: T433-NokHangHui-COMP4071_thor_2022-11-05_0252.txt
Info Writing csv report file to: T433-NokHangHui-COMP4071_files_md5s.csv
Info No json report file will be written
Info Writing html report file to: T433-NokHangHui-COMP4071_thor_2022-11-05_0252.html
Info IP Address 1: 192.168.68.129
Info ScanID: S-VLhMpB3LPZo
Info System is not a domain controller
Info Max file size can be scanned is 31.5 MB, use --max_file_size to increase the limit
Info Selected modules: Buttons, Cron, EnvCheck, Filescan, Firewall, Hosts, IntegrityCheck, LoggedIn, ProcessCheck, ServiceCheck, Timestamp, UserDir, Users
Info Deselected modules: DeepDive, Dropout, Rootkit, Thunderstrum
Info Selected features: Amcache, Archive, ArchiveScan, AtJobs, AuthorizedKeys, Bifrost2, C2, CPUlimit, CheckString, CronParser, EVTX, Eml, EnrichFileInfo, ExeDecompress, FilenameIocs, Filescan, KeywordIocs, Lnk, LogScan, MagicHeader, ParseCobaltStrike, Prefetch, ProcessConnections, ProcessHandles, ProgressTracker, RecycleBin, Registryhive, Rescontrol, SHIMCache, Sigma, SignalHandler, Stix, TeamViewer, ThorB, WER, WMIPersistence, WebdirScan, Yara
Info Deselected features: Action, Bifrost, DoublePulsar, DumpScan, GroupsXML, ProcessIntegrity, VulnerabilityCheck
Info No filters applied

```

Below is the HTML report. It confirmed the detection of multiple rules and gave 33 warnings.

Scan Information		Modules	Statistics
Scanner	Thor	Filescan	33
Version	10.7.3	Alerts	0
Run on System	T433-NokHangHui-COMP4071	Warnings	33
Argument list	--customonly --dbfile /var/lib/thor/thor10-lite.db	Notice	3
Signature Database	2022/11/04-080717	Info	384
Start Time	Sat Nov 5 02:52:47 2022	Errors	0
End Time	Sat Nov 5 03:00:30 2022		
IP Addresses	192.168.68.129		
Run as user	root		
Admin rights	yes		
Platform	Kali GNU/Linux Rolling		
Log File Name	T433-NokHangHui-COMP4071_thor_2022-11-05_0252.txt		
False Positive Filters Applied	0		
Scan ID	S-VLhMpB3LPZo		

**Help**

**Filters**

**Hint 1**

**Hint 2**

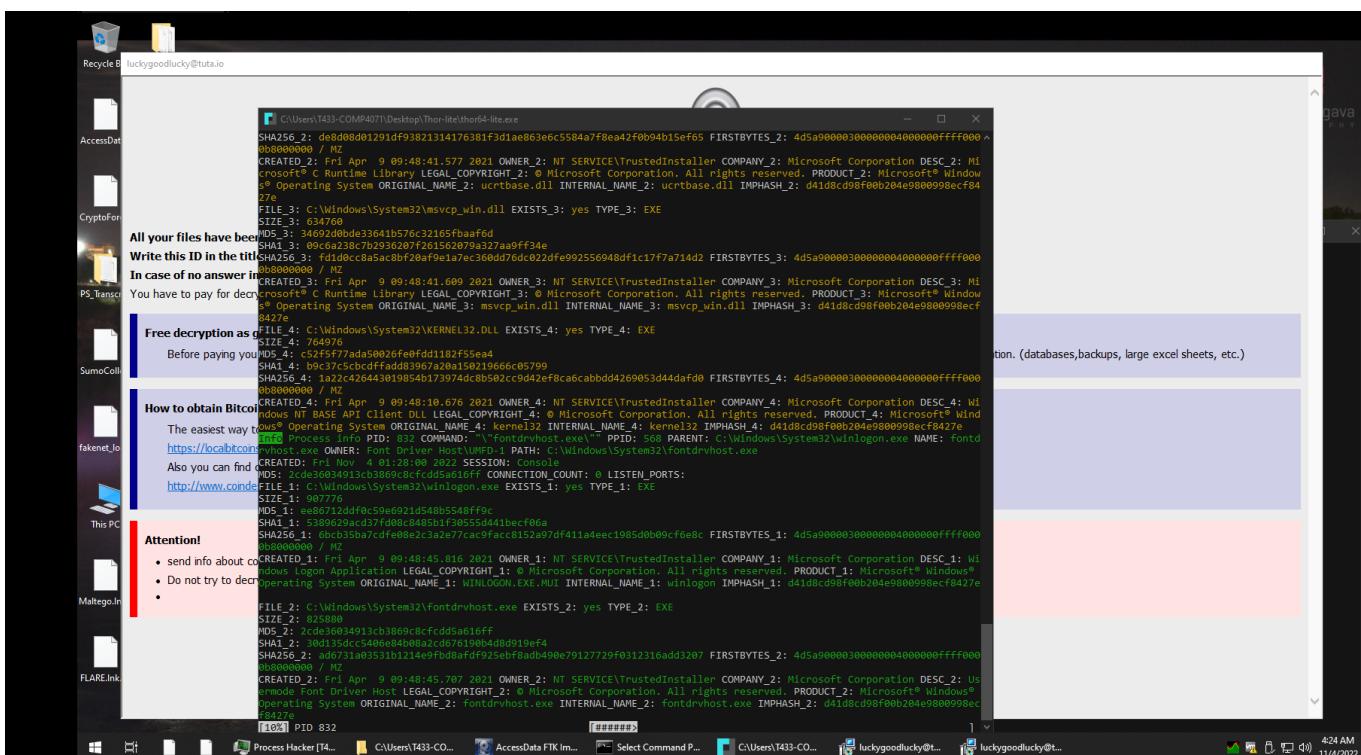
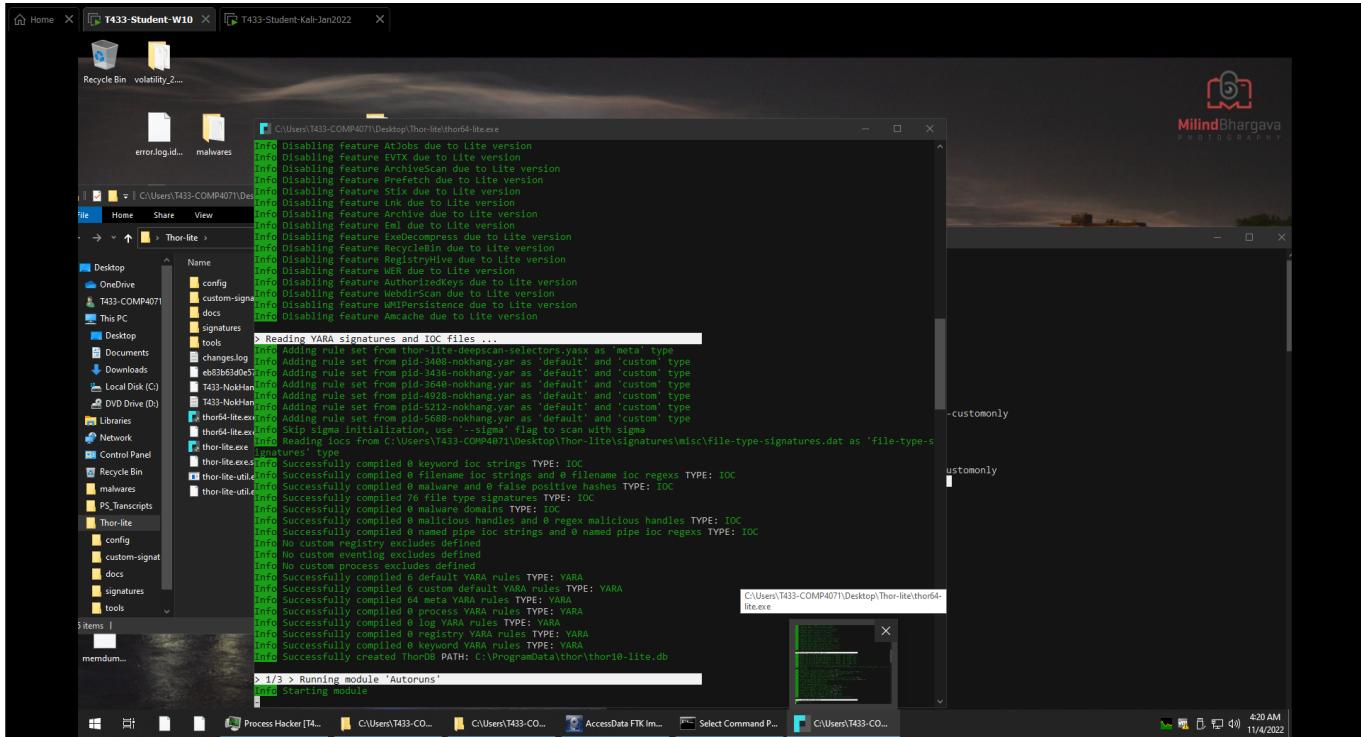
**Alerts**

**Warnings**

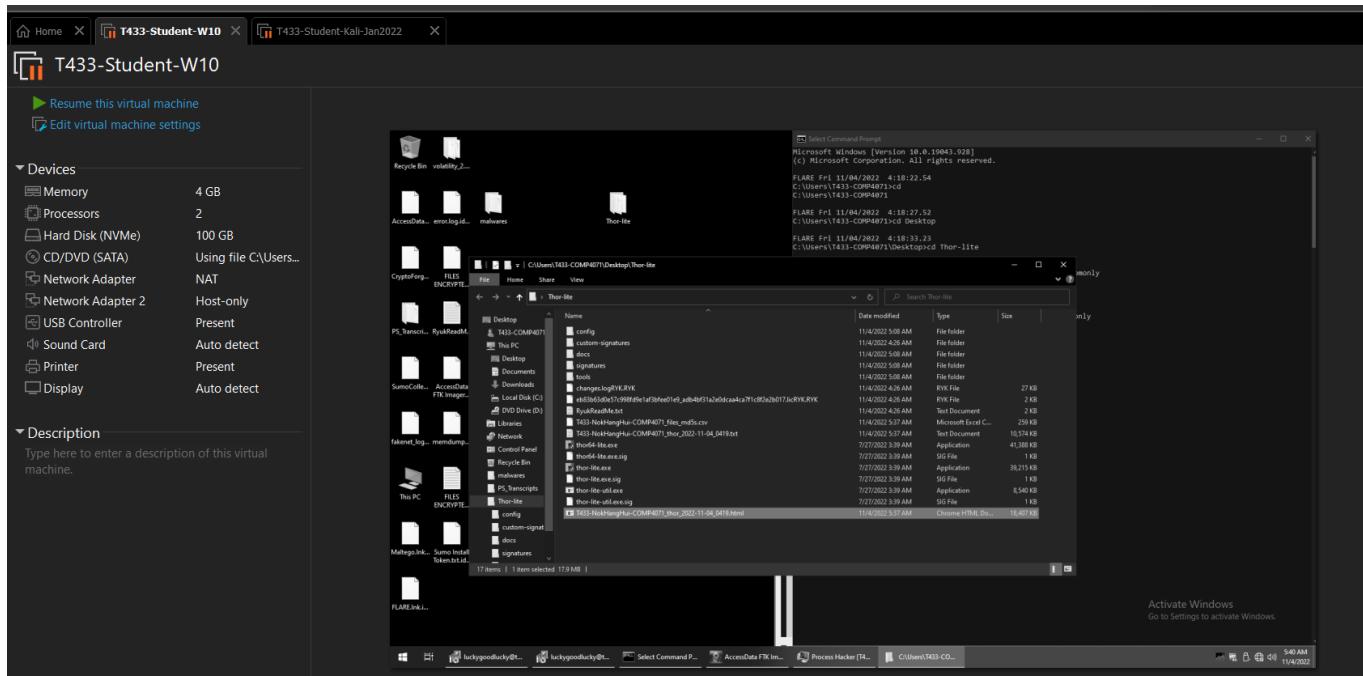
Warning 1  
Nov 5 06:54:02 T433-NokHangHui-COMP4071/192.168.68.129  
MODULE: Filescan  
MESSAGE: Possibly Dangerous file found  
FILE: /home/t433-student/.local/share/Trash/files/pid-2072-nokhang.var

No filters applied

Running thor-lite in the infected Windows VM.



HTML report is generated.



The screenshot shows a web browser window with the URL 'file:///C:/Users/lexhu/Desktop/T433-NokHangHui-COMP4071\_thor\_2022-11-04\_0419.html'. The page title is 'THOR Scan Report'. The content of the page is a detailed scan report from the THOR Lite scanner. It includes sections for 'Scan Information', 'Modules', 'Statistics', 'Help', 'Alerts', 'Warnings', and a log file section. The 'Scan Information' table contains various system details and configuration parameters. The 'Statistics' table provides a summary of findings. The 'Help' section contains links to keyboard shortcuts and filtering options. The 'Alerts' and 'Warnings' sections are currently empty. The log file section shows a single entry: 'Warning 1 Nov 4 08:22:32 T433-NokHangHui-COMP4071/192.168.171.128'.

## THOR Scan Report

This THOR Lite license permits non-commercial use only. It is strictly prohibited to sell THOR Lite or sell services that include the use of THOR Lite. For details, see the EULA in the ./docs folder. For a special license that covers these cases and suppresses this message, please contact our sales via <https://www.nextron-systems.com/get-started/>

Scan Information		Modules	Statistics
Scanner	Thor	Filescan	3195
Version	10.7.3	ProcessCheck	412
Run on System	T433-NokHangHui-COMP4071		
Argument list	--customonly --dbfile %ProgramData%\thor\thor10-lite.db		
Signature Database	2022/11/04-080717		
Start Time	Fri Nov 4 04:19:49 2022		
End Time	Fri Nov 4 05:37:09 2022		
IP Addresses	192.168.171.128		
Run as user	T433-NOKHANGHUI\T433-COMP4071		
Admin rights	yes		
Platform	Windows 10 Enterprise		
Log File Name	T433-NokHangHui-COMP4071_thor_2022-11-04_0419.txt		
False Positive Filters Applied	0		
Scan ID	S-aL2xeJKPfDk		

Alerts	
Warnings	
Warning 1	Nov 4 08:22:32 T433-NokHangHui-COMP4071/192.168.171.128

Statistics	
Alerts	0
Warnings	2861
Notice	749
Info	996
Errors	0

Help	
Shortcuts	Use Ctrl+I (Windows/Linux) or ⌘+I (macOS) to return to the top of the page
Filters	You can provide a file (~filter file) with regular expressions to suppress false positives
Hint 1	Select text and use the context menu to filter / select / lookup strings
Hint 2	Click on a module to filter for all events from that module.

No filters applied