

## LAB 05

Thầy Mai Hoàng Đình  
Trường đại học FPT

Người thực hiện

Đặng Hoàng Nguyên

## Public AV Scanners

VirusTotal là một công ty con của Google chuyên phân tích tệp và URL. Ngoài giao diện miễn phí, VirusTotal cũng có cả API riêng tư và công khai.

Các kết quả từ VirusTotal bao gồm việc phát hiện kết quả của phần mềm độc hại bởi công cụ chống vi-rút được hỗ trợ. Chúng ta sẽ download file tại đường link này:

<https://wildfire.paloaltonetworks.com/publicapi/test/pe>

Nếu như phân tích kĩ, ta có thể thấy được ban đầu, con exe này bị dính một lỗi CVE-2020-0601 đây là một lỗi liên quan tới việc đánh lừa CryptoAPI để nhằm giả mạo độ uy tín của phần mềm để khiến chúng trở nên đáng tin cậy trên hệ điều hành windows

946a42effe1253fe5aea3c4b42bb848320af40b5fc17e0bb2775ec319b577f

38 / 71

38 security vendors and no sandboxes flagged this file as malicious

wildfire-test-pe-file.exe

Size: 54.00 KB | Last Analysis Date: a moment ago | EXE

peexe cve-2020-0601 spreader exploit

Community Score

DETECTION DETAILS BEHAVIOR COMMUNITY

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to [automate checks](#).

Popular threat label: trojan.bebloh/pancar Threat categories: trojan, pua Family labels: bebloh, pancar, cve20200601

Security vendors' analysis

Vendor	Detection	Vendor	Detection
Antiy-AVL	Trojan/Win32.BTSGeneric	Avira (no cloud)	SPR/PanCar.A
Bkav Pro	W32.AIDetect/Malware	ClamAV	Win Dropper Bebloh-9954185-0
CrowdStrike Falcon	Win/grayware_confidence_60% (D)	Cylance	Unsafe
Cynet	Malicious (score: 100)	Cyren	W32/Trojan.DFQ.gen/Eldorado
DeepInstinct	MALICIOUS	DrWeb	BackDoor.Bebloh.375

DeeplInStinct	⚠ MALICIOUS	DrWeb	⚠ BackDoor.Bebloh.375
Elastic	⚠ Malicious (high Confidence)	F-Secure	⚠ PrivacyRisk.SPR/PanCar.A
Fortinet	⚠ Riskware/WildFireTestFile	Google	⚠ Detected
Gridinsoft (no cloud)	⚠ Trojan.Win32.Gen.vblst1	Ikarus	⚠ Trojan.Win32.Agent
Jiangmin	⚠ Exploit.Multi.ar	K7AntiVirus	⚠ Riskware ( 0040eff71 )
K7GW	⚠ Riskware ( 0040eff71 )	Kaspersky	⚠ HEUR:Trojan.Win32.Generic
Malwarebytes	⚠ Exploit.CVE20200601	McAfee-GW-Edition	⚠ BehavesLike.Win32.Backdoor.gh
Microsoft	⚠ Trojan.Win32/Sabisk.EN.Dlml	NANO-Antivirus	⚠ Trojan.Win32.Bebloh.gdorj1
QuickHeal	⚠ Trojan.Wacatac.RI.S12026051	Rising	⚠ Trojan.Zpvedol8.F912 (RDMK:cmRlazzq0...
Sangfor Engine Zero	⚠ Trojan.Win32.Save.a	SentinelOne (Static ML)	⚠ Static AI - Suspicious PE
Sophos	⚠ Troj/AutoG-JY	SUPERAntiSpyware	⚠ Trojan.Agent/Gen-Crypt
TACHYON	⚠ Trojan/W32.Agent.55296.ALN	Tencent	⚠ Malware.Win32.Gen:circ.10bde52a
Trapmine	⚠ Malicious.moderate.ml.score	VBA32	⚠ Backdoor.Bebloh
VinIT	⚠ Backdoor.Win32.Bebloh.OL	Yandex	⚠ Trojan.Agent/q5HLRo863dA
Zillya	⚠ Exploit.CVE20200601.Win32.65	ZoneAlarm by Check Point	⚠ HEUR:Trojan.Win32.Generic
Acronis (Static ML)	✅ Undetected	AhnLab-V3	✅ Undetected

Như ta thấy ở đây, khi đem lên virustotal, trang web đã phân tích cho chúng ta biết rằng đây là một file chứa rất nhiều mã độc, có bao gồm cả CVE exploit,...

Qua bên phần detail, ta có thể kiểm tra MD5 hash của nó, những thông tin cơ bản, như là viết bằng ngôn ngữ gì. Ở đây là được viết bằng C++ trên Visual Studio 2010, được tạo trong khoảng thời gian nào, lần đăng tải đầu tiên, lần phân tích cuối cùng.

<b>Basic properties</b>	
MD5	005f4a3cfb6805579e90be8b23d18e8a
SHA-1	ad3315dee0188d5886cd739d1c8aa778023fc1
SHA-256	946a42effe1253fe5a5ea3c4b42bb848320a40b5fc17e0bb2775ec319b577f
Vhash	054048f5f111038243tz
Authenthash	b58095e2d9e817e8f35a99e38e443a5c689cfb9af0c1d32174d8b25852dd7b
Imphash	318cc6ba22de5640b5a89a3bd3b774c
Rich PE header hash	abd45a9dd93a63d0346522b93fcb3f
SSDEEP	768 x/EAAqxG0QqLccK+xl.7scaOZlcGs8WbwnWh+6AXT2qEDnXbPGEDUXnpT0rJmnU.OAc0QqgHW7/ZwcF8c6jELX+PupTNj
TLSH	T125435B253594C032DCA215300978D2A25A7F78326678858B7FE8677DAFF17C09B2937B
File type	Win32 EXE
Magic	PE32 executable (console) Intel 80386, for MS Windows
TrID	Win32 Executable MS Visual C++ (generic) (47.3%)   Win64 Executable (generic) (15.9%)   Win32 Dynamic Link Library (generic) (9.9%)   Win16 NE executable (generic) (7.6%)   Win32 Executable (generic) (6.8%)
DetectItEasy	PE32 Compiler: EP Microsoft Visual C/C++ (2008-2010) [EXE32]   Compiler: Microsoft Visual C/C++ (2010) [libcmf]   Linker: Microsoft Linker (10.0) [Console32,console]
File size	54.00 KB (55296 bytes)
<b>History</b>	
Creation Time	2012-12-20 19:14:11 UTC
First Submission	2023-05-29 09:23:09 UTC
Last Submission	2023-05-29 09:23:09 UTC
Last Analysis	2023-05-29 09:23:09 UTC
<b>Names</b>	
wildfire-test-pe-file.exe	
<b>Portable Executable Info</b>	

Ngoài ra khi phân tích rteen đây, ta có thể thấy rằng file EXE này sử dụng hai thư viện là Kernel32.dll và ADVAPI32.dll

Nhìn dưới đây, khi phân tích sâu hơn về từng dll. Ta thấy rằng, với

- ADVAPI32.dll, con malware sẽ gọi đến hàm “RegCloseKey”, “RegCreateKeyExW”, “RegSetValueExW”. Đây nói nôm na có thể là sử dụng để tạo 1 registry set value cho nó và sau đó đóng lại
- KERNEL32.dll, ta có thể thấy là nó sử dụng hàm **CloseHandle** để đóng một handle của một đối tượng đã mở trước đó. **CreateFileA** và **CreateFileW** đều là mở file nhưng mà với các kiểu khác nhau. **ExitProcess** là một hàm dùng để thoát tiến trình

The screenshot shows the VirusTotal interface for a file with SHA256 hash 946a42effe1253feaf5aea3c4b42bb848320af40b6fc17e0bb2775ec319b577f. The 'Imports' section is expanded, showing two categories of imported functions:

- ADVAPI32.dll:**
  - RegCloseKey
  - RegCreateKeyExW
  - RegSetValueExW
- KERNEL32.dll:**
  - CloseHandle
  - CreateFileA
  - CreateFileW
  - DecodePointer
  - DeleteCriticalSection
  - EncodePointer
  - EnterCriticalSection
  - ExitProcess
  - FlushFileBuffers
  - FreeEnvironmentStringsW

The Windows taskbar at the bottom shows the date and time as 4:48 PM on 5/29/2023.

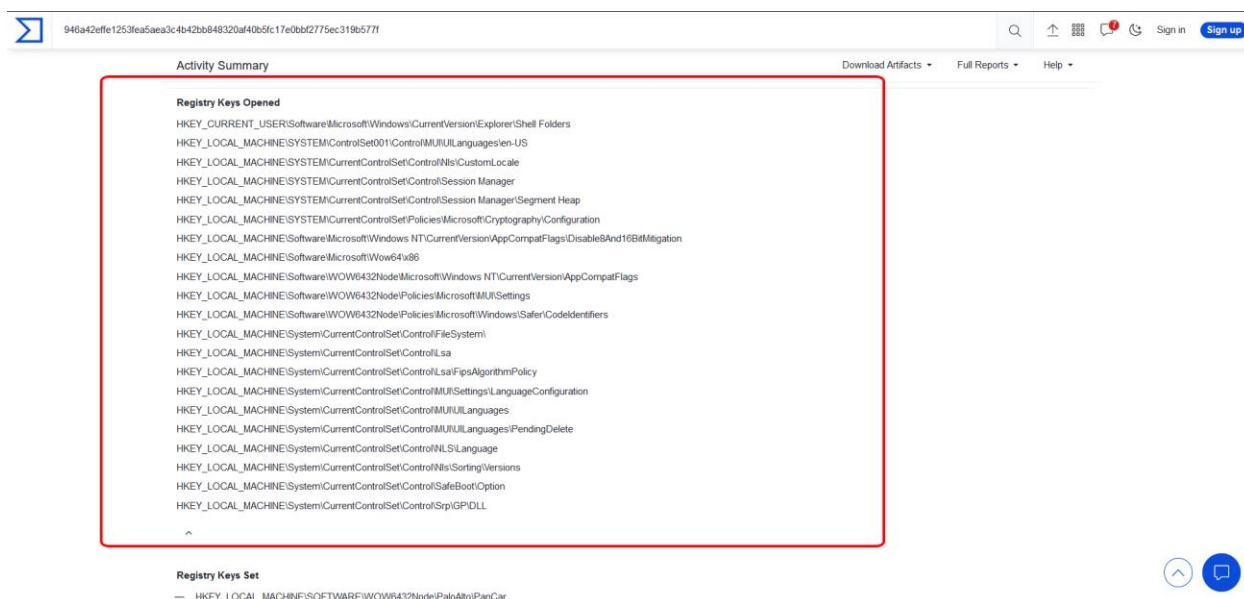
Theo như ta thấy malware đã gọi ra khá nhiều dll và gọi thêm một số tiến trình hệ thống như là conhost.exe

The screenshot shows the 'Activity Summary' section of the VirusTotal interface for the same file. The 'Files Opened' list is expanded, showing a long list of files accessed by the malware:

- C:\Users\user\Desktop\wildfire-test-pe-file.exe
- C:\Windows\AppPatch\systemmain.sdb
- C:\Windows\Globalization\Sorting\sortdefault.nls
- C:\Windows\SYSTEM32\ntmarta.dll
- C:\Windows\SYSTEM32\ole32.dll
- C:\Windows\SYSTEM32\wintypes.dll
- C:\Windows\System\WinSxS\ADVAPI32.dll
- C:\Windows\System\WinSxS\CRYPTBASE.dll
- C:\Windows\System\WinSxS\KERNEL32.DLL
- C:\Windows\System\WinSxS\KERNELBASE.dll
- C:\Windows\System\WinSxS\RPCRT4.dll
- C:\Windows\System\WinSxS\SspiCli.dll
- C:\Windows\System\WinSxS\apphelp.dll
- C:\Windows\System\WinSxS\bcryptPrimitives.dll
- C:\Windows\System\WinSxS\msvcrt.dll
- C:\Windows\System\WinSxS\ntdll.dll
- C:\Windows\System\WinSxS\sechost.dll
- C:\Windows\System32\ComMessaging.dll
- C:\Windows\System32\ComUIComponents.dll
- C:\Windows\System32\TextInputFramework.dll
- C:\Windows\System32\en-US\user32.dllmui
- C:\Windows\WinSxS\x-ww\microsoft.windows.common-controls\_6595b64144ccf1df\_6.0.17134.1304\_none\_d3be61b7c93d9f0
- C:\Windows\WinSxS\x-ww\microsoft.windows.common-controls\_6595b64144ccf1df\_6.0.17134.1304\_none\_d3be61b7c93d9f0\comctl32.DLL
- C:\Windows\system32\IMM32.DLL
- C:\Windows\system32\conhost.exe
- C:\Windows\system32\dwapi.dll

The Windows taskbar at the bottom shows the date and time as 4:48 PM on 5/29/2023.

Ở đây, kéo xuống phần phía dưới của summary, ta có thể thấy được rằng application đã mở rất nhiều file registry của hệ thống, bao gồm những thông tin cơ bản của hệ thống

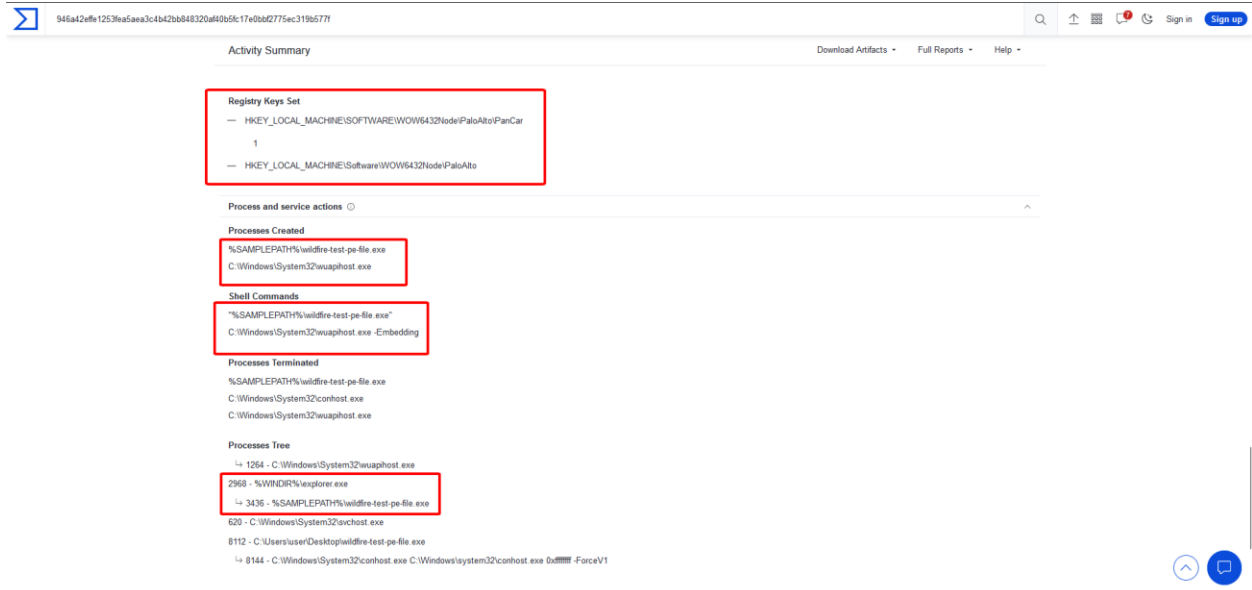


Như chúng ta đã dự đoán, thư viện ADVAPI32.dll được gọi nhằm để tạo ra những registry. Ở đây tạo 1 Registry có tên là PanCar. Sau khi khởi chạy chương trình, theo virustotal, có hai tiến trình được tạo ra đó chính là **wildfire-test-pe-file.exe** và **wuapihost.exe**.

- **wuapihost.exe** là một tiến trình hệ thống nên chúng ta cũng sẽ không bàn tới
- **wildfire-test-pe-file.exe** là một tiến trình được khởi chạy khá là đáng ngờ ngay sau khi nhấn chạy chương trình

Đặc biệt lưu ý tới phần Processes Tree, cho ta biết những tiến trình nay đang chạy, khởi động bởi những Parent process nào. Cho ví dụ như là wildfire-test-pe-file.exe nó được chạy với PID là 3436 và được chạy bởi Parent nó chính là 2968

Đặc biệt ngoài ra, nó còn phát hiện ra rằng, khi malware được chạy, nó còn khởi động cmd lên và sử dụng command để tắt đi một số tính năng bảo mật của máy thông qua cmd



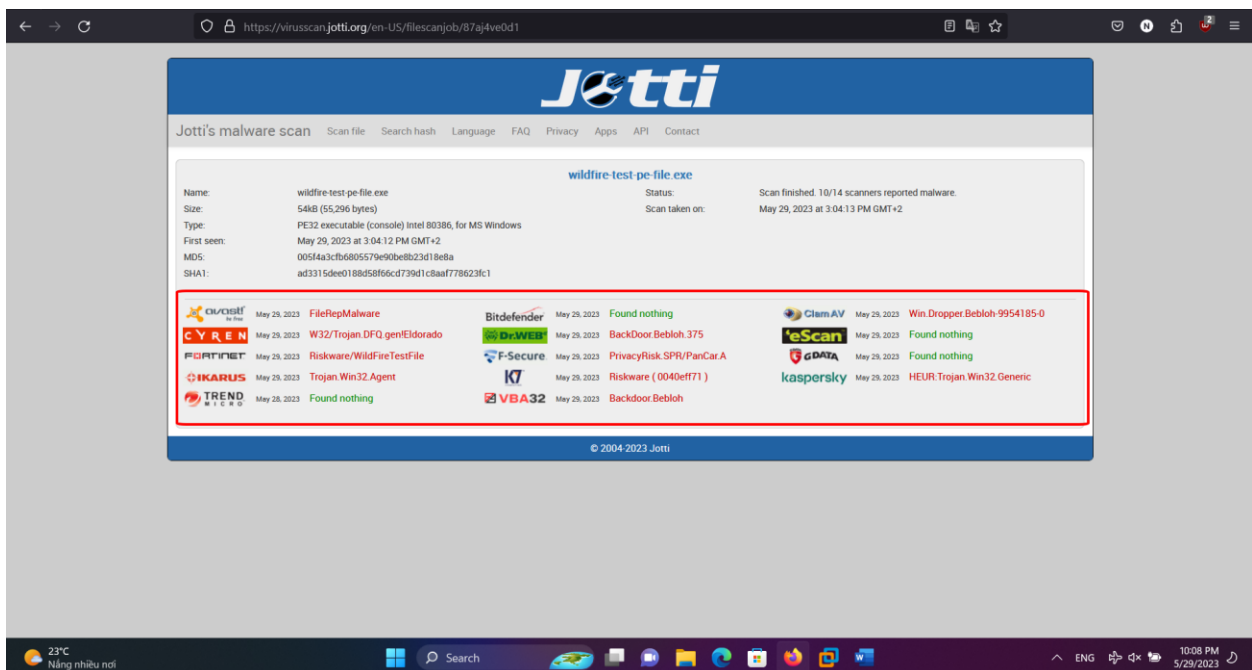
The screenshot shows the VirusShare Activity Summary for a specific file. The file is identified by the hash 946a42ef12539a5a53c4b42b848320a40b5c17e0b42775ec3196d771. The summary is divided into several sections:

- Registry Keys Set:** A red box highlights the registry keys set, showing two entries: `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\PaloAlto` and `HKEY_LOCAL_MACHINE\Software\WOW6432Node\PaloAlto`.
- Process and service actions:** This section is expanded to show further details.
- Processes Created:** A red box highlights the processes created, listing `%SAMPLEPATH%\wildfire-test-pe-file.exe` and `C:\Windows\System32\wuaphost.exe`.
- Shell Commands:** A red box highlights the shell commands executed, showing `"%SAMPLEPATH%\wildfire-test-pe-file.exe"` and `C:\Windows\System32\wuaphost.exe -Embedding`.
- Processes Terminated:** Lists the processes that were terminated, including `%SAMPLEPATH%\wildfire-test-pe-file.exe`, `C:\Windows\System32\conhost.exe`, and `C:\Windows\System32\wuaphost.exe`.
- Processes Tree:** A red box highlights the process tree, showing the hierarchy starting from `1264 - C:\Windows\System32\wuaphost.exe`, which spawned `2968 - %WINDIR%\explorer.exe`, which in turn spawned `3436 - %SAMPLEPATH%\wildfire-test-pe-file.exe`, `620 - C:\Windows\System32\conhost.exe`, `8112 - C:\Users\User\Desktop\wildfire-test-pe-file.exe`, and `8144 - C:\Windows\System32\conhost.exe`.

Ngoài ra chúng ta sẽ sử dụng một phần mềm scan trên mạng khác với virustotal đó chính là Jotti, phần mềm jotti này cho phép chúng ta có thể scan với nhiều loại antivirus khác nhau

Link scan của file mọi người có thể xem tại đây: <https://virusscan.jotti.org/en-US/filescanjob/87aj4ve0d1>

Sử dụng web scan này, ta có thể thấy có 10/14 ứng dụng tìm ra được đây là một con virus có chứa mã độc bên trong đó



The screenshot shows the Jotti's malware scan results for the file `wildfire-test-pe-file.exe`. The scan was completed on May 29, 2023, at 3:04:13 PM GMT+2. The file is identified by the hash 946a42ef12539a5a53c4b42b848320a40b5c17e0b42775ec3196d771. The scan results are as follows:

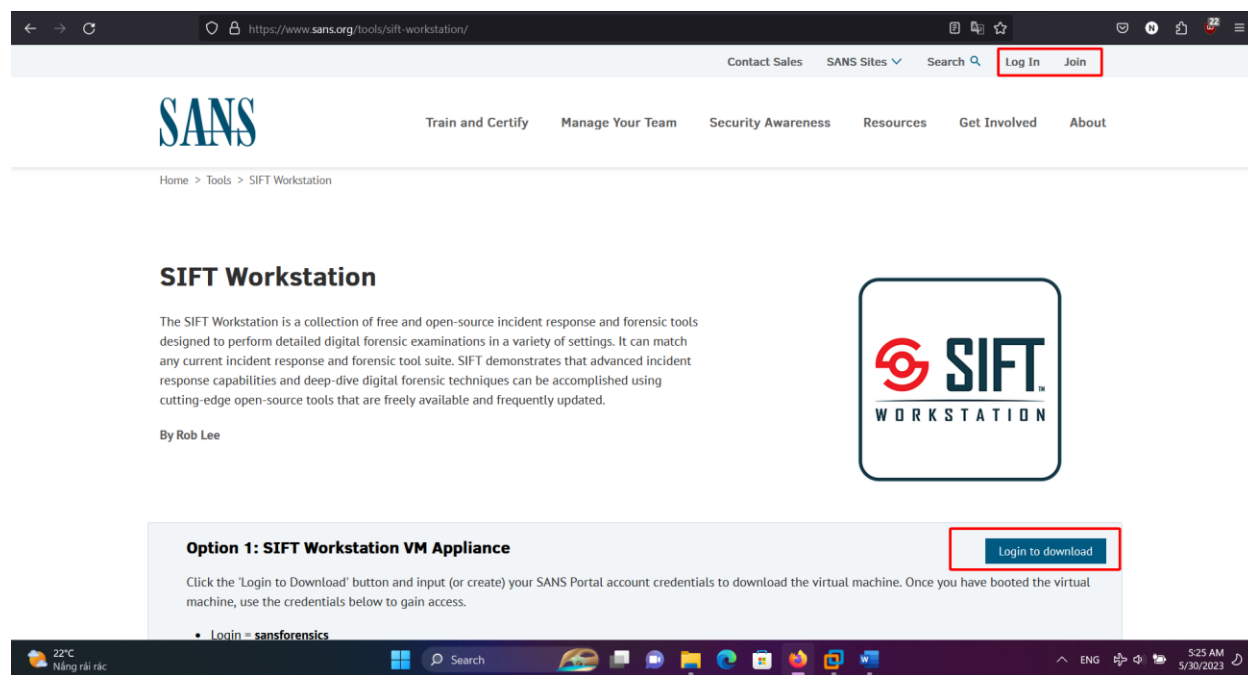
Antivirus	Result
Cybereason	FileRep/Malware
Cyren	W32/Trojan.DFQ.gen/Eldorado
Fortinet	Riskware/WildFireTestFile
Ikarus	Trojan.Win32.Agent
Trend Micro	Found nothing
Bitdefender	Found nothing
Dr.Web	BackDoor.Bebloh.375
F-Secure	PrivacyRisk.SPR/PanCar.A
K7	Riskware (0040eff71)
VBA32	Backdoor.Bebloh
ClimAV	Win.Dropper.Bebloh.9954185-0
eScan	Found nothing
GDATA	Found nothing
Kaspersky	HEUR.Trojan.Win32.Generic

## Sandbox Setup and Configuration

Tại đây, ta sẽ cài đặt San sift và thực hiện phân tích tính con Virus trên môi trường Ubuntu của Sans sift. Sans sift là một môi trường chuyên để thực hiện việc forensics.

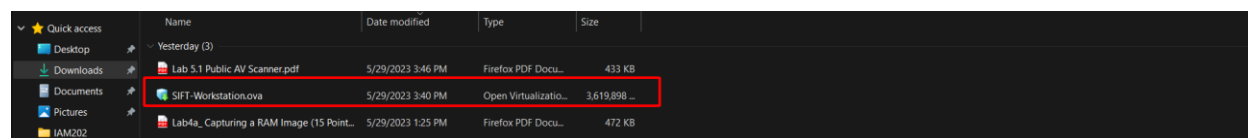
Chỉ cần vào <https://www.sans.org/tools/sift-workstation/> và thực hiện đăng nhập hoặc tạo tài khoản trước khi cài đặt, Sans cần yêu cầu chúng ta cần tài khoản trước khi cài đặt.

Sau khi tạo tài khoản / login xong, chúng ta bắt đầu thực hiện download



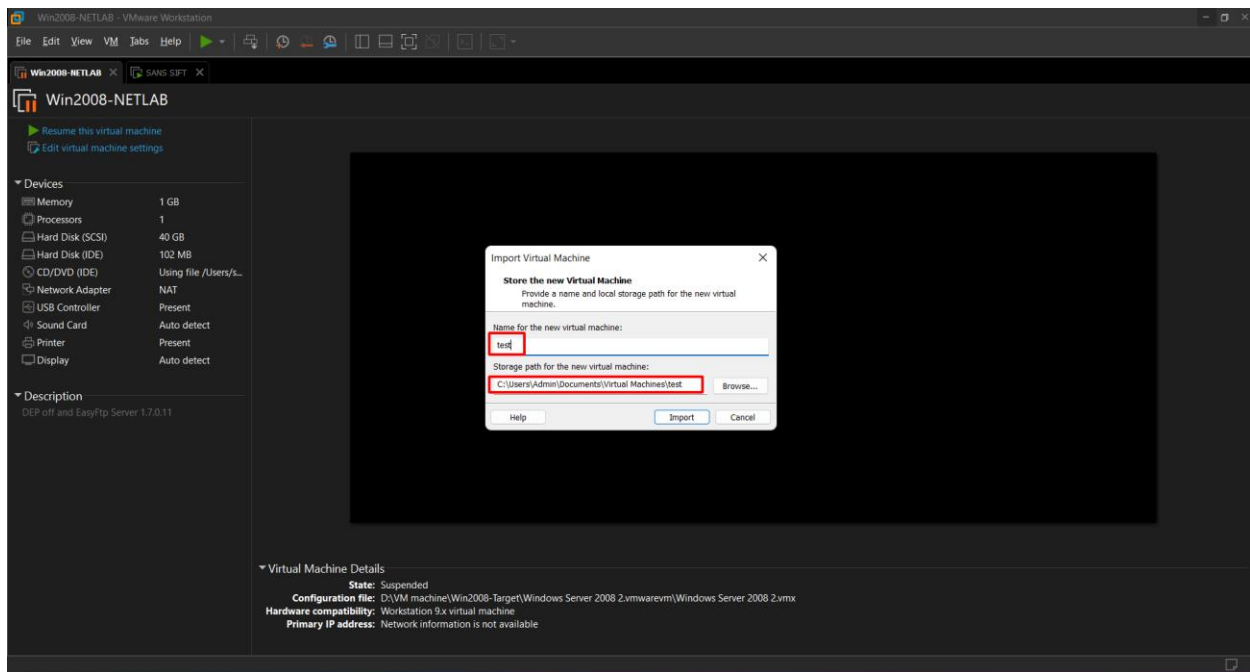
The screenshot shows the SANS website's SIFT Workstation page. The top navigation bar includes links like 'Contact Sales', 'SANS Sites', 'Search', 'Log In', and 'Join'. The 'Log In' button is highlighted with a red box. The main content area features the SANS logo, a description of the SIFT Workstation, and a section titled 'Option 1: SIFT Workstation VM Appliance'. Within this section, a 'Login to download' button is highlighted with a red box. Below the button, there is a note about using SANS Portal account credentials to download the virtual machine.

Lúc tải xong chúng ta sẽ có một file có đuôi **ova**. Đây là file máy ảo đã được cấu hình sẵn, chúng ta chỉ cần **double click** để máy tự chạy trong VMware.

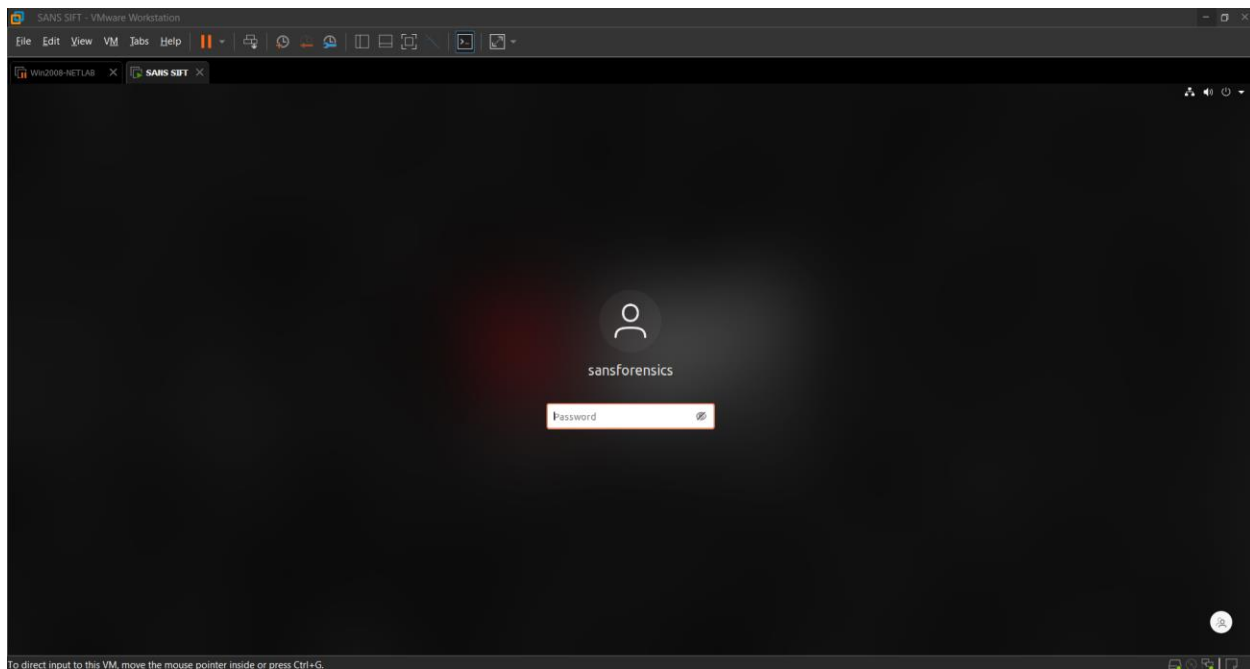


Name	Date modified	Type	Size
Yesterday (3)			
Lab 5.1 Public AV Scanner.pdf	5/29/2023 3:46 PM	Firefox PDF Docu...	433 KB
SIFT-Workstation.ova	5/29/2023 3:40 PM	Open Virtualizatio...	3,619,898 ...
Lab4a_Capturing a RAM Image (15 Point...	5/29/2023 1:25 PM	Firefox PDF Docu...	472 KB

Như tại đây, trong trường hợp này, chúng ta sẽ set tên máy ảo tên là **test** và lưu tại **C:\Users\Admin\Documents\Virtual Machines\test**. Sau khi set up xong, ta sẽ nhấn nút import để cho nó tự import vào trong máy



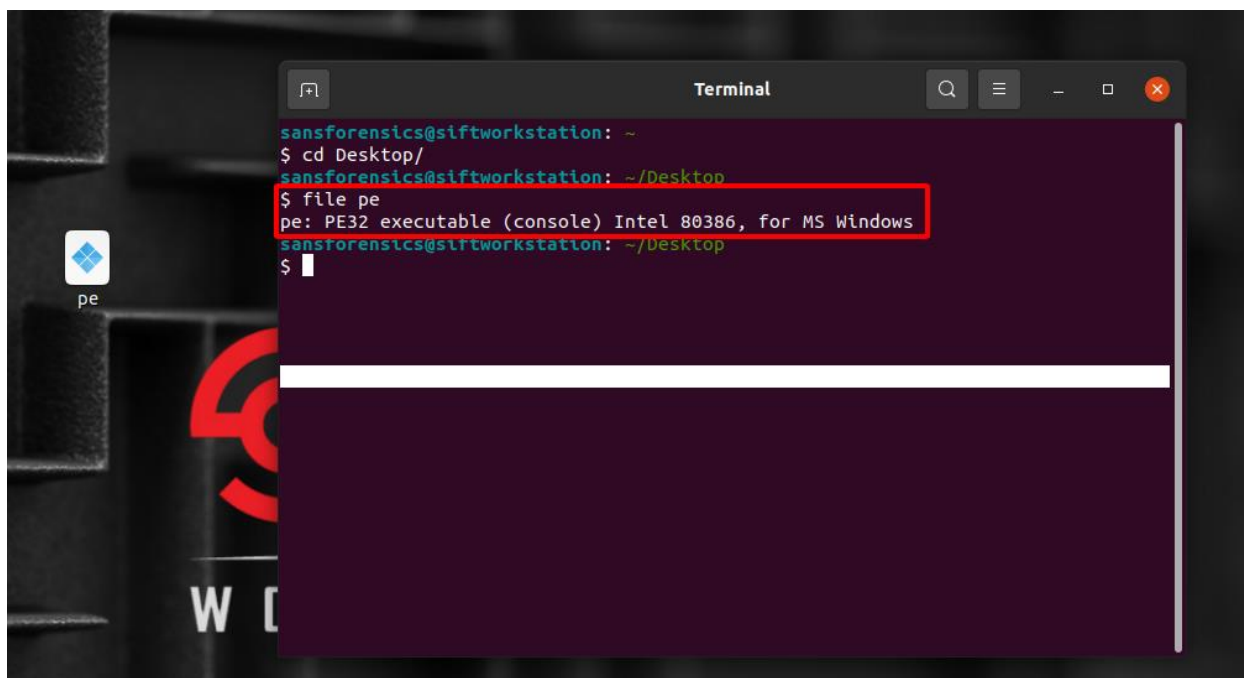
Sau khi import xong, ta sẽ vào login với tên user là **sansforensics** với password là **forensics**



Bắt đầu phân tích tĩnh bằng Sans sift, việc đầu tiên ta xác định bằng command **file** xem file này như thế nào.

- Sau khi bỏ vào file, thì ta thấy được đây là một file thực thi 32bit, sử dụng Intel 80386, dùng để thực thi trên hệ điều hành windows



A terminal window titled "Terminal" is shown over a dark desktop background. The terminal shows the following commands and output:

```
sansforensics@siftworkstation: ~  
$ cd Desktop/  
sansforensics@siftworkstation: ~/Desktop  
$ file pe  
pe: PE32 executable (console) Intel 80386, for MS Windows  
sansforensics@siftworkstation: ~/Desktop  
$
```

The output line "pe: PE32 executable (console) Intel 80386, for MS Windows" is highlighted with a red rectangular box. On the desktop, a file icon labeled "pe" is visible.

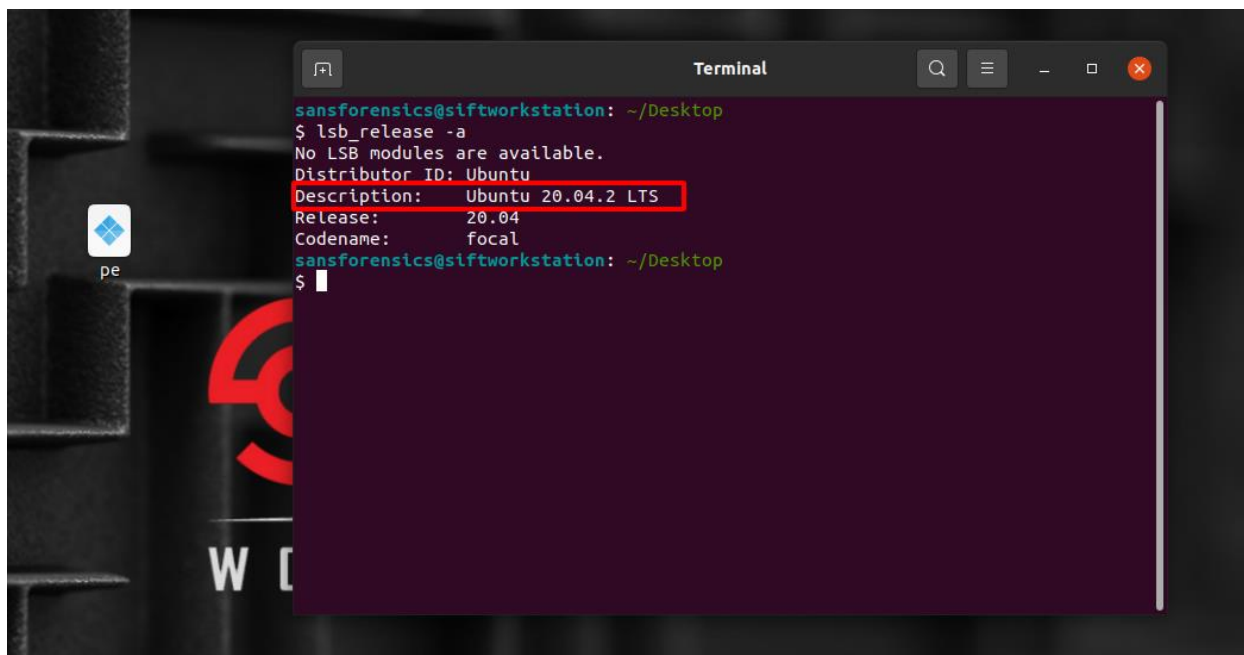
## Detect it easy

Thay vì bên Windows, chúng ta sẽ phân tích bằng PeID, trên linux, chúng ta sẽ có một công cụ mang tên là Detect it easy (DIE). Đây là một công cụ khá là phổ biến trên thế giới hiện nay để thực hiện việc phân tích tĩnh. DIE hiện có ở trên ba hệ điều hành là Windows và Linux và MacOS.

Để download trên Linux, ta sẽ vào trang github dưới đây và chọn bản phù hợp

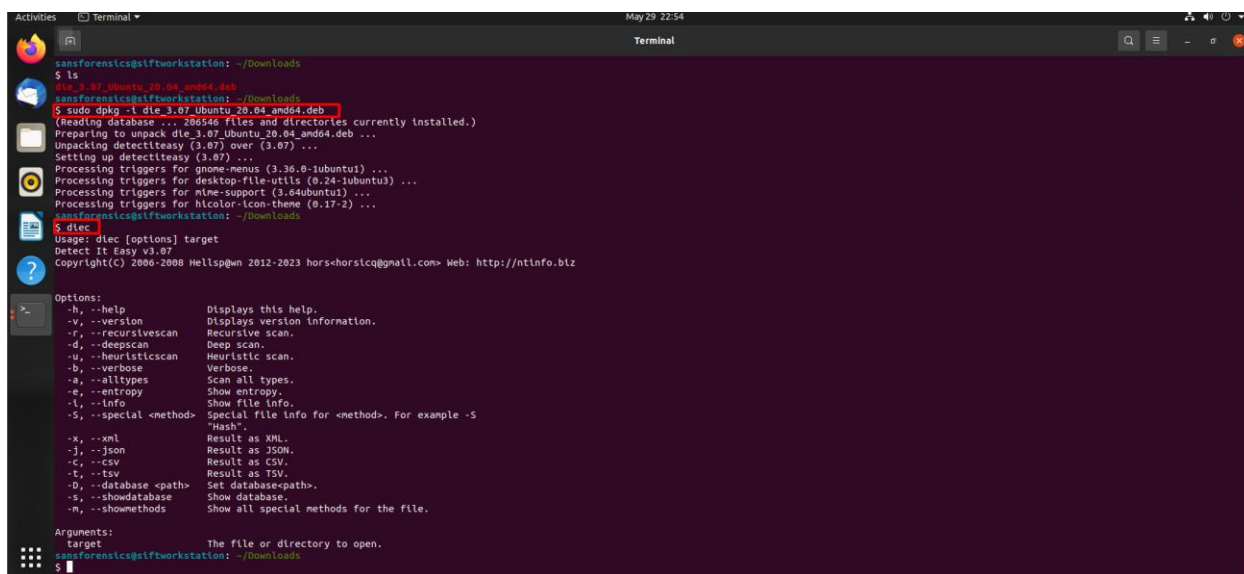
- [https://github.com/horsicq/DIE-engine/releases?fbclid=IwAR3RdV392LQtXdWWhco2QwH918cB7gPC4Iroy2Tr8Dulee9Y22f2jQ\\_KwXQ](https://github.com/horsicq/DIE-engine/releases?fbclid=IwAR3RdV392LQtXdWWhco2QwH918cB7gPC4Iroy2Tr8Dulee9Y22f2jQ_KwXQ)

Để muốn biết chúng ta đang dùng bản ubuntu hiện tại là gì, ta sẽ dùng command **lsb\_release -a**



Theo như Sans SIFT lúc làm bài lab này, hiện đang sử dụng Ubuntu bản 20.04 LTS. Vì thế việc của chúng ta chỉ cần lên trang github trên và download bản của Ubuntu 20.04 về và cài đặt.

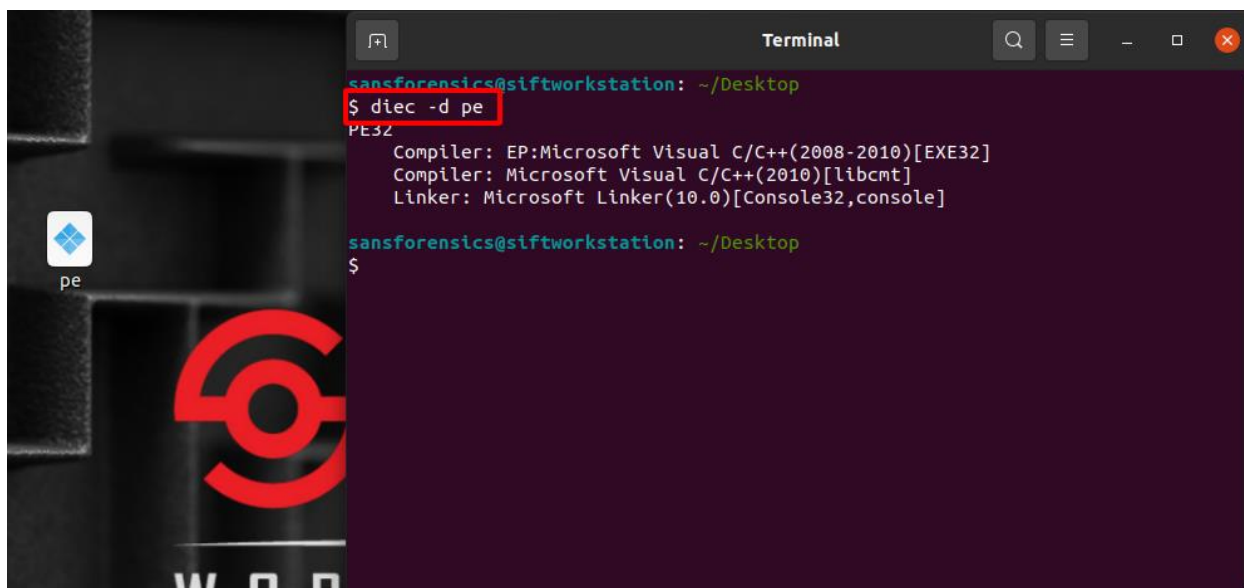
Bắt đầu tiến hành cài đặt file deb với câu lệnh **sudo dpkg -i <filename.deb>**



Vì DIE còn một bản dành cho GUI giống trên Windows, nhưng vì lý do nào đó thì link tải của GUI đã bị xóa, nên là chúng ta sẽ phân tích dựa trên CLI của chúng

Sử dụng câu lệnh **diec -d <filename>** để xem chương trình được viết bằng chương trình gì.

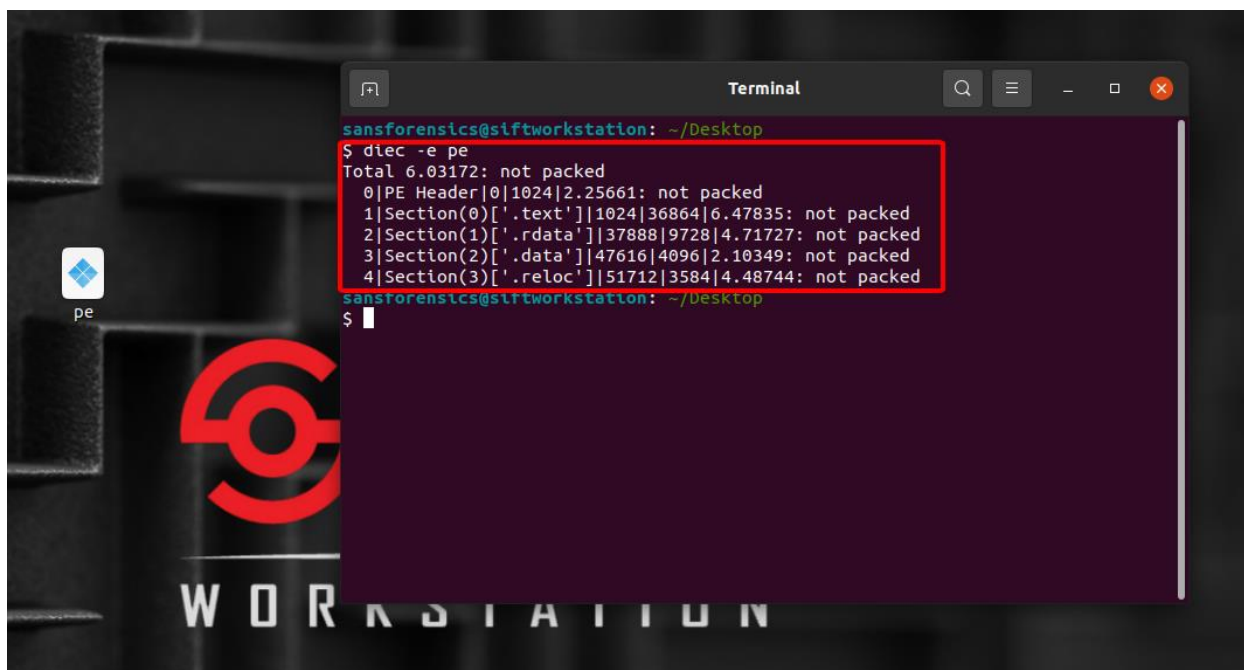
- Theo đúng giống như trên VirusTotal, DIE cũng scan ra được chương trình này là chương trình 32bit được viết bằng ngôn ngữ C/C++



```
sansforensics@siftworkstation: ~/Desktop
$ diac -d pe
PE32
Compiler: EP:Microsoft Visual C/C++(2008-2010)[EXE32]
Compiler: Microsoft Visual C/C++(2010)[libcm]
Linker: Microsoft Linker(10.0)[Console32,console]
sansforensics@siftworkstation: ~/Desktop
$
```

Sử dụng câu lệnh **diac -e <filename>** để check xem rằng file có bị packed hay không

- Theo như ta thấy được hình bên dưới thì các file không bị packed lại, vì thế không cần dùng các công cụ như **upx** để unpacked.



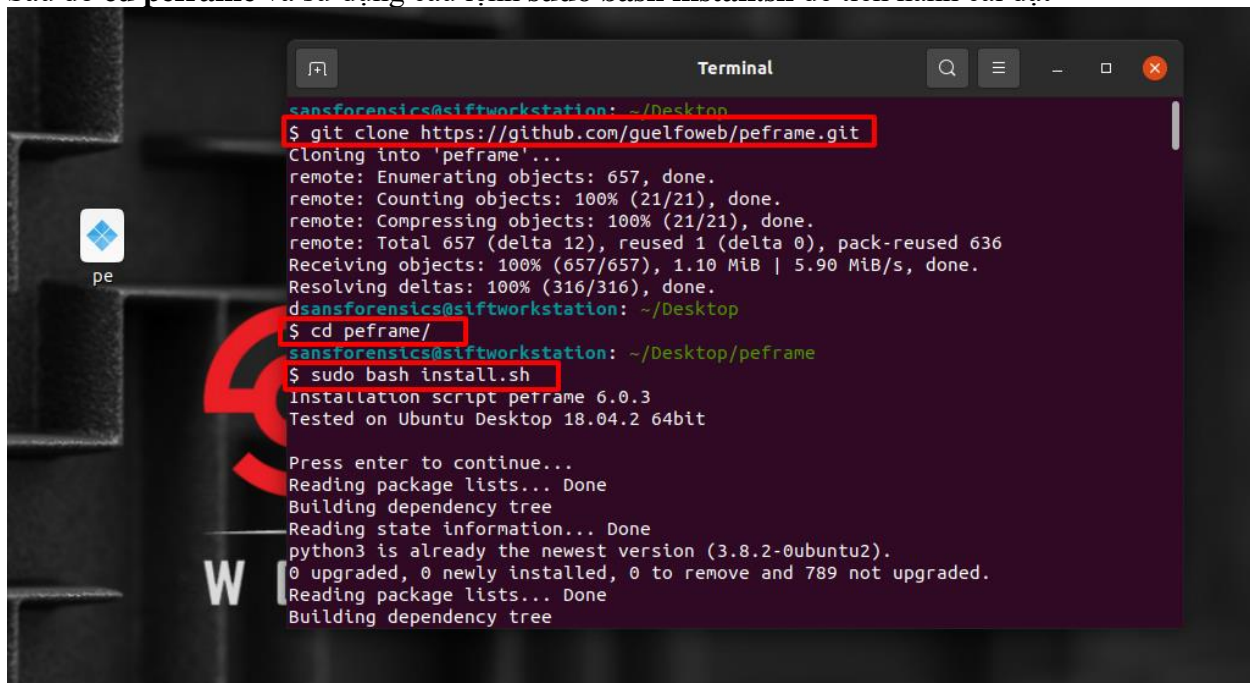
```
sansforensics@siftworkstation: ~/Desktop
$ diac -e pe
Total 6.03172: not packed
0|PE Header|0|1024|2.25661: not packed
1|Section(0)|['.text']|1024|36864|6.47835: not packed
2|Section(1)|['.rdata']|37888|9728|4.71727: not packed
3|Section(2)|['.data']|47616|4096|2.10349: not packed
4|Section(3)|['.reloc']|51712|3584|4.48744: not packed
sansforensics@siftworkstation: ~/Desktop
$
```

## PEFRAME

Trên Sans Sift, chúng ta sẽ tiến hành tải PE Frame – một công cụ giúp chúng ta có thể phân tích tĩnh một file window trên môi trường sandbox của linux.

Sử dụng câu lệnh **git clone** <https://github.com/guelfoweb/peframe.git>

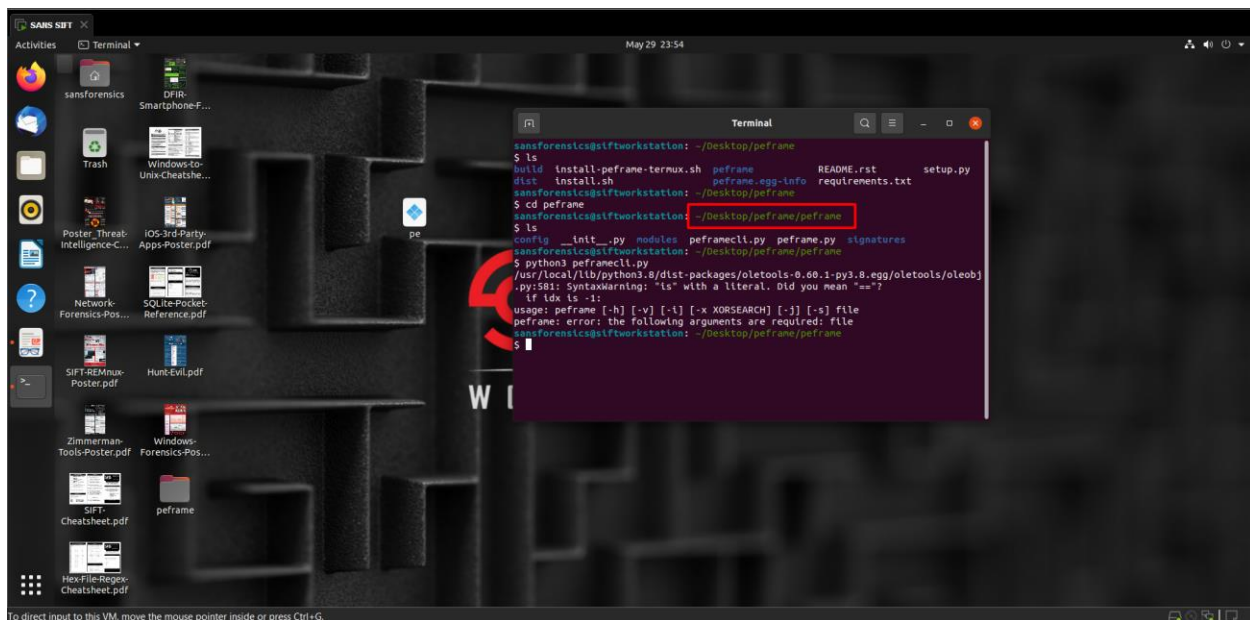
Sau đó **cd peframe** và sử dụng câu lệnh **sudo bash install.sh** để tiến hành cài đặt



```
sansforensics@siftworkstation: ~/Desktop
$ git clone https://github.com/guelfoweb/peframe.git
Cloning into 'peframe'...
remote: Enumerating objects: 657, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 657 (delta 12), reused 1 (delta 0), pack-reused 636
Receiving objects: 100% (657/657), 1.10 MiB | 5.90 MiB/s, done.
Resolving deltas: 100% (316/316), done.
sansforensics@siftworkstation: ~/Desktop
$ cd peframe/
sansforensics@siftworkstation: ~/Desktop/peframe
$ sudo bash install.sh
Installation script perrame 6.0.3
Tested on Ubuntu Desktop 18.04.2 64bit

Press enter to continue...
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3 is already the newest version (3.8.2-0ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 789 not upgraded.
Reading package lists... Done
Building dependency tree
```

Sau khi cài đặt xong, vào bên trong thư mục **peframe** trong thư mục **peframe** bằng câu lệnh **cd**



```
sansforensics@siftworkstation: ~/Desktop/peframe
$ ls
build  install-peframe-termux.sh  peframe  README.rst  setup.py
dist  install.sh  peframe.egg-info  requirements.txt
sansforensics@siftworkstation: ~/Desktop/peframe
$ cd peframe
sansforensics@siftworkstation: ~/Desktop/peframe/peframe
$ ls
config  __init__.py  modules  peframecli.py  peframe.py  signatures
sansforensics@siftworkstation: ~/Desktop/peframe/peframe
$ python3 peframecli.py
/usr/local/lib/python3.8/dist-packages/oletools-0.60.1-py3.8.egg/oletools/oleobj.py:581: SyntaxWarning: "is" with a literal. Did you mean "=="?
  if idx is -1:
usage: peframe [-h] [-v] [-i] [-x XORSEARCH] [-j] [-s] file
peframe: error: the following arguments are required: file
sansforensics@siftworkstation: ~/Desktop/peframe/peframe
$
```

Sử dụng câu lệnh **python3 peframecli.py <tên file> -i** để bắt đầu phân tích tĩnh

```
sansforensics@lftworkstation: ~/Desktop/peframe/peframe
$ python3 peframecli.py ~/Desktop/pe -l
/usr/local/lib/python3.8/dist-packages/oletools-0.00.1-py3.8.egg/oletools/oleobj.py:581: SyntaxWarning: "is" with a literal. Did you mean "=="?
if idx is -1:

File Information (time: 0:00:01.551701)
-----
filename      pe
filetype      PE32 executable (console) Intel 80386, for MS Windows
filesize      55296
hash_sha256    5a7a6d8705c0b14f4527b06ebbb43f20dab91ebacc293dff7af04b1a209270
virustotal     /
imagebase      0x400000
entrypoint     0x14ac
imphash        318cc0baf22de5640b5a89a3bd3b774c
datetime       2012-12-20 19:14:11
dll            False
directories     import, tls, relocations
sections       .rdata, .data, .reloc, .text *
features       antiddbg, packer

Interactive mode (press TAB to show commands)
-----
[peframe]>
```

Đây là thông tin ban đầu của file, chúng ta đã phân tích quá nhiều về phần này rồi, chúng ta sẽ phân tích các thứ khác như dll, hành vi của chúng. Nhấn **tab** để chọn những options khác

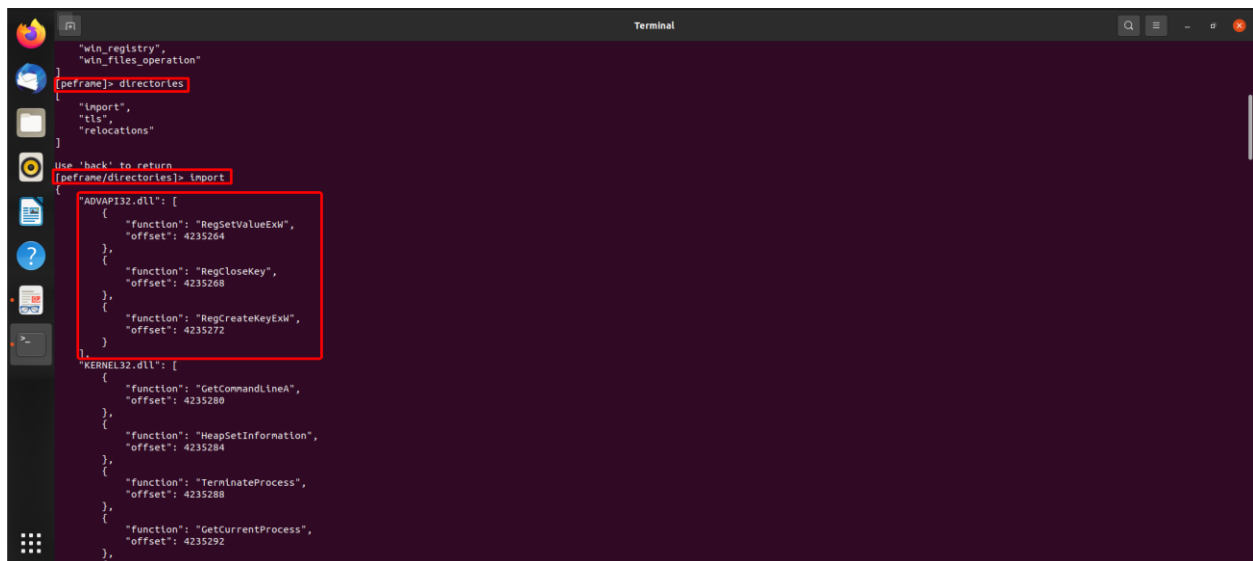
```
File Information (time: 0:00:01.551701)
-----
filename      pe
filetype      PE32 executable (console) Intel 80386, for MS Windows
filesize      55296
hash_sha256    5a7a6d8705c0b14f4527b06ebbb43f20dab91ebacc293dff7af04b1a209270
virustotal     /
imagebase      0x400000
entrypoint     0x14ac
imphash        318cc0baf22de5640b5a89a3bd3b774c
datetime       2012-12-20 19:14:11
dll            False
directories     import, tls, relocations
sections       .rdata, .data, .reloc, .text *
features       antiddbg, packer

Interactive mode (press TAB to show commands)
-----
[peframe]> behavior breakpoint directories exit features hashes info sections strings virustotal yara_plugins
[peframe]> behavior
{
  "anti_dbg",
  "xor",
  "win_registry",
  "win_files_operation"
}
[peframe]>
```

Vào bên trong phần behavior, ta thấy rằng, chương trình này có thực hiện hành vi như sau:

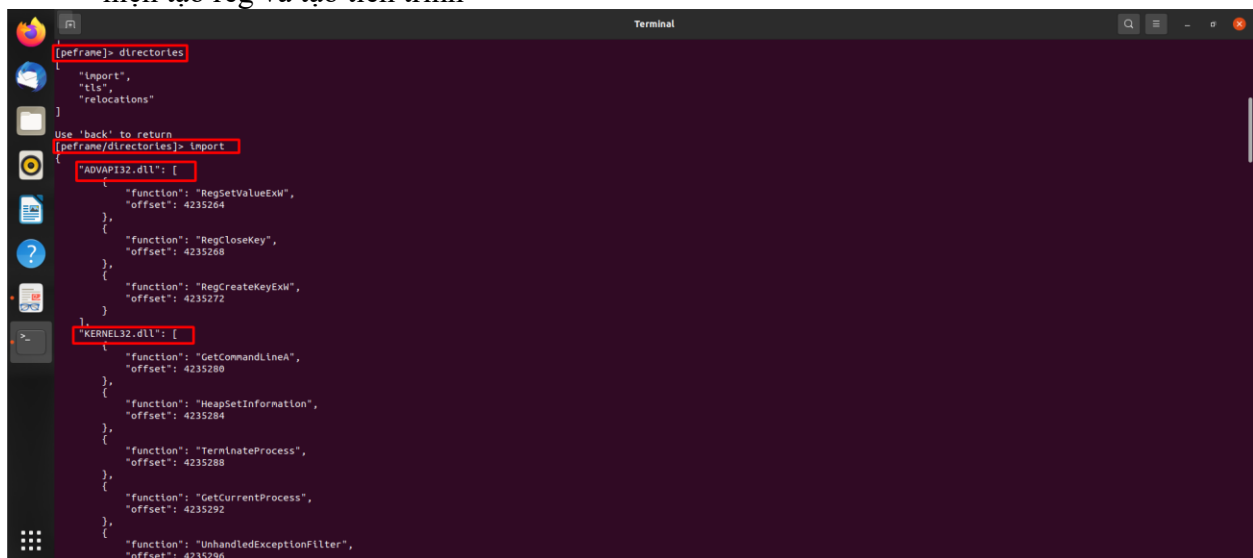
- Anti\_dbg: là một trình chạy anti debug, sẽ làm chúng ta khi debug sẽ đưa ra những thông tin sai lệch, gây khó dễ cho các trình debugger lúc đang thực hiện quá trình RE
- Xor: có lẽ đây là kiểu xor dữ liệu với nhau
- Win\_registry: Khi phân tích trên virusTotal, ta dễ dàng thấy được, phần mềm này khi chạy sẽ tạo ra registry sử dụng dynamic library có sẵn của Windows, chính nó cũng sẽ tự set value cho registry vừa tạo
- Win\_files\_operation: Sử dụng Kernel32.dll để xóa các file, tạo các tiến trình, tạo file mới như trên virusTotal đã phân tích báo cáo cho chúng ta

Vào bên trong phần Directory → import để xem các dll đã được gọi những hàm nào



```
"win_registry",
"win_files_operation"
}
(peframe)> directories
{
  "Import",
  "tls",
  "relocations"
}
Use 'back' to return
(peframe/directories)> import
{
  "ADVAPI32.dll": [
    {
      "function": "RegSetValueExW",
      "offset": 4235264
    },
    {
      "function": "RegCloseKey",
      "offset": 4235268
    },
    {
      "function": "RegCreateKeyExW",
      "offset": 4235272
    }
  ],
  "KERNEL32.dll": [
    {
      "function": "GetCommandLineA",
      "offset": 4235280
    },
    {
      "function": "HeapSetInformation",
      "offset": 4235284
    },
    {
      "function": "TerminateProcess",
      "offset": 4235288
    },
    {
      "function": "GetCurrentProcess",
      "offset": 4235292
    }
  ],
}
```

- Quả thật, bên trong chương trình gọi đến hai thư viện dll đó là ADVAPI32.dll và KERNEL32.dll
- Như đã phân tích ở trên phần VirusTotal, hàm ADVAPI32.dll và KERNEL32.dll sẽ thực hiện tạo reg và tạo tiến trình



```
(peframe)> directories
{
  "Import",
  "tls",
  "relocations"
}
Use 'back' to return
(peframe/directories)> import
{
  "ADVAPI32.dll": [
    {
      "function": "RegSetValueExW",
      "offset": 4235264
    },
    {
      "function": "RegCloseKey",
      "offset": 4235268
    },
    {
      "function": "RegCreateKeyExW",
      "offset": 4235272
    }
  ],
  "KERNEL32.dll": [
    {
      "function": "GetCommandLineA",
      "offset": 4235280
    },
    {
      "function": "HeapSetInformation",
      "offset": 4235284
    },
    {
      "function": "TerminateProcess",
      "offset": 4235288
    },
    {
      "function": "GetCurrentProcess",
      "offset": 4235292
    },
    {
      "function": "UnhandledExceptionFilter",
      "offset": 4235296
    }
  ],
}
```

Vào trong mục featured chúng ta sẽ xét tới những hàm mà featured nó gọi tới. Ở đây có hai hàm là antdbg và packer



```
Activities Terminal May 30 00:37
$ python3 peframecli.py ~/Desktop/pe -l
/usr/local/lib/python3.8/dist-packages/oletools-0.08.1-py3.8.egg/oletools/oleobj.py:581: SyntaxWarning: "is" with a literal. Did you mean "=="?
if idc.is_1:

-----
File Information (time: 0:00:01.528358)
-----
filename      pe
filetype      PE32 executable (console) Intel 80386, for MS Windows
filesize      55296
hash sha256    5a7a6d8705c0b14f4527b06ebbb43f20dab91ebacc293dffa7af04ba1a209270
virustotal     /
imagebase      0x400000
entrypoint     0x14ac
inphash        318cc0ba722de5640b5a89a3bd3b774c
datetimestamp 2012-12-20 19:14:11
dll            False
directories     import, tls, relocations
sections       .rdata, .data, .reloc, .text *
features       antdbg, packer

-----
Interactive mode (press TAB to show commands)
-----
[peframe]>
[peframe]> breakpoint directories exit features hashes info sections strings virustotal yara_plugins
[peframe]> features
{
  "antdbg",
  "packer"
}

Use 'back' to return
[peframe/features]> antdbg
{
  "GetLastError",
  "IsDebuggerPresent",
  "IsProcessorFeaturePresent",
  "TerminateProcess",
  "UnhandledExceptionFilter"
}
[peframe/features]>
```

- **Đối với antdbg:** ta có thể thấy hàm khá phổ biến khi phân tích của antdbg như là IsDebuggerPresent. Hàm này theo như em biết sẽ check xem rằng là có trình debug đang hoạt động không. Nếu có thì sẽ gây khó dễ cho chúng ta trong quá trình debug. Vì thế khi muốn chạy đúng phải set đúng Instruction Point để chương trình truyền biến về một cách đúng khi gặp antdbg

Vào bên trong phần strings chúng ta sẽ phân tích những strings bên trong hexdump có những gì

- Khi vào bên trong phần strings, cũng chả có gì thú vị, không thấy dấu hiệu của gửi đến một địa chỉ nào như các lab trước, chỉ thấy những hàm mà dll gọi tới. Đặc biệt, có thể quan tâm tới KeyOpenFailed.txt và KeyValuedFailed.txt. Hai file này khá là đặc biệt và chúng ta có thể cần để tâm tới chúng

```
Terminal
" \\\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNPQRSTUWXYZ[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~",
"('SPW",
"700pp",
"b bbb",
"xpppp",
" \\\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNPQRSTUWXYZ[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~",
"C:\\KeyOpenFailed.txt",
"C:\\KeyValuedFailed.txt",
"RegCreateKeyEx",
"RegSetValueEx",
"RegCloseKey",
"ADVAPI32.dll",
"GetCommandLine",
"HeapSetInformation",
"TerminateProcess",
"GetCurrentProcess",
"UnhandledExceptionFilter",
"SetUnhandledExceptionFilter",
"IsDebuggerPresent",
"GetLastError",
"HeapFree",
"ClosesHandle",
"EncodePointer",
"DecodePointer",
"EnterCriticalSection",
"LeaveCriticalSection",
"InitializeCriticalSectionAndSpinCount",
"RtlUnwind",
"GetProcAddress",
"GetModuleHandleW",
"ExitProcess",
"WriteFile",
"GetStdHandle",
"GetModuleFileNameW",
"GetModuleFileNameA",
"FreeEnvironmentStringsW",
"WideCharToMultiByte",
"GetEnvironmentStringsW",
"SetHandleCount",
"GetFileType",
"GetStartupInfo",
"DeleteCriticalSection",
"IsValid",
"IsValidValue",
"IsValidValue"
```