BÁO CÁO THỰC HÀNH

Môn học: Pháp chứng kỹ thuật số

Lab 6: Final Test

GVHD: Đoàn Minh Trung

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT334.O11.ANTT1

STT	Họ và tên	MSSV	Email
1	Võ Thiên An	20520378	20520378@gm.uit.edu.vn
2	Nguyễn Quang Huy	20520546	20520546@gm.uit.edu.vn
3	Bùi Đức Hoàng	20520514	20520514@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

STT	Công việc	Kết quả tự đánh giá
1	memory.dmp	100%
2	dump.raw (Flag 1)	100%
3	dump.raw (Flag 2)	100%
4	dump.raw (Flag 3)	100%
5	Android_Chall.apk	100%

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

[1] Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

1. memory.dmp

Đầu tiên, ta sử dụng volatility với option imageinfo để scan profile phù hợp để đọc, tuy nhiên, volatility không phát hiện được hệ điều hành mà file dump đã được trích xuất. Có thể phỏng đoán rằng file có thể từ hệ điệu hành Linux

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./volatility_2.5_linux_x64 -f memory.dmp imageinfo
Volatility Foundation Volatility Framework 2.5
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : No suggestion (Instantiated with no profile)
                     AS Layer1 : LimeAddressSpace (Unnamed AS)
                     AS Layer2 : FileAddressSpace (/home/kali/Desktop/memory.dmp)
                      PAE type : No PAE
```

Để detect, ta có thể sử dụng lệnh strings để tìm kiếm với keyword " Linux version", từ đó, ta có thể biết được hệ điều hành được dump là Ubuntu 20.04 với kernel là 5.13.0-39-generic.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ strings memory.dmp | grep -i "Linux version"
 o  The intent is to make the tool independent of Linux version dependencies,
 o  The intent is to make the tool independent of Linux version dependencies,
MESSAGE=Linux version 5.13.0-39-generic (buildd@lcy02-amd64-080) (gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1)
untu SMP Thu Mar 24 16:43:35 UTC 2022 (Ubuntu 5.13.0-39.44~20.04.1-generic 5.13.19)
Linux version 5.13.0-39-generic (buildd@lcy02-amd64-080) (gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0,
 Thu Mar 24 16:43:35 UTC 2022 (Ubuntu 5.13.0-39.44~20.04.1-generic 5.13.19)
Linux version 5.13.0-39-generic (buildd@lcy02-amd64-080) (gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0,
 Thu Mar 24 16:43:35 UTC 2022 (Ubuntu 5.13.0-39.44~20.04.1-generic 5.13.19)9)
 o  The intent is to make the tool independent of Linux version dependencies,
Linux version 5.13.0-39-generic (buildd@lcy02-amd64-080) (gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0,
 Thu Mar 24 16:43:35 UTC 2022 (Ubuntu 5.13.0-39.44~20.04.1-generic 5.13.19)9)
Linux version 5.13.0-39-generic (buildd@lcy02-amd64-080) (gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0,
 Thu Mar 24 16:43:35 UTC 2022 (Ubuntu 5.13.0-39.44~20.04.1-generic 5.13.19)9)
Linux version 5.13.0-39-generic (buildd@lcy02-amd64-080) (gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0,
 Thu Mar 24 16:43:35 UTC 2022 (Ubuntu 5.13.0-39.44~20.04.1-generic 5.13.19)9)
Linux version 5.13.0-39-generic (buildd@lcy02-amd64-080) (gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0,
 Thu Mar 24 16:43:35 UTC 2022 (Ubuntu 5.13.0-39.44~20.04.1-generic 5.13.19)
```

Để có thể dump đươc profile cho volatility, ta có thể tham khảo phương pháp từ trang sau : https://n00bzunit3d.xyz/blog/loading-linux-profile-volatility2/

Khi này, ta có thể có được profile phù hợp trong volatility:

```
ubuntu@ubuntu:~/Desktop/volatility$ python vol.py --info | grep Ubuntu
Volatility Foundation Volatility Framework 2.6.1
LinuxUbuntu_5_13_0-39-generic_profilex64 - A Profile for Linux Ubuntu_5.13.0-39-
generic_profile x64
```

Theo gợi ý của đề bài, ta có thể check các lịch sử lệnh bash bằng option linux_bash

```
ubuntu@ubuntu:~/Desktop/volatility$ python vol.py --profile LinuxUbuntu_5_13_0-39-generic_profilex64 -f memory.dmp linux_bash
Volatility Foundation Volatility Framework 2.6.1
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timeliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apihooks (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.malware.servicediff (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.userassist (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getsids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shellbags (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.evtlogs (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.tcaudit (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.dumpregistry (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.lsadump (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.threads (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.mac.apihooks_kernel (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.registry.amcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.check_syscall_shadow (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.malware.svcscan (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.auditpol (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.ssdt (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.registry.registryapi (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.apihooks (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.envars (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shimcache (ImportError: No module named Crypto.Hash)
Pid      Name                 Command Time                   Command
-------- -------------------- ------------------------------ -------
  109633 bash                 2022-04-10 05:49:10 UTC+0000   echo "aW5zZWNsYWJ7dzNsYzBtM190MF9tM20wcllfZjByM25zMWM1fQ==" > Un33dt0r3@dth1s.txt
  109633 bash                 2022-04-10 05:49:10 UTC+0000   chmod 755 avml
  109633 bash                 2022-04-10 05:49:10 UTC+0000   sudo rm ~/.bash_history
  109633 bash                 2022-04-10 05:49:18 UTC+0000   sudo ./avml memory.dmp
```

Trong danh sách các lệnh bash, ta có 1 command đáng lưu ý khi echo một đoạn mã đã được encode với dấu hiệu của base 64 vào 1 file txt

```
Command
-------
echo "aW5zZWNsYWJ7dzNsYzBtM190MF9tM20wcllfZjByM25zMWM1fQ==" > Un33dt0r3@dth1s.txt
chmod 755 avml
sudo rm ~/.bash_history
sudo ./avml memory.dmp
```

Thực hiện decode đoạn mã trên, ta có được flag của bài

```
ubuntu@ubuntu:~/Desktop/volatility$ echo "aW5zZWNsYWJ7dzNsYzBtM190MF9tM20wcllfZ
jByM25zMWM1fQ==" | base64 --decode
inseclab{w3lc0m3_t0_m3m0rY_f0r3ns1c5}ubuntu@ubuntu:~/Desktop/volatility$
```

Flag : inseclab{w3lc0m3_t0_m3m0rY_f0r3ns1c5}

2. dump.raw (Flag 1):

Sử dụng option imageinfo và kdbgscan kiểm tra thông tin OS của file dump.raw ta có được profile là Win7SP1x64

```
┌──(kali㉿kali)-[~/Desktop/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f ../dump.raw imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64_23418, Win2008R2SP1x64, Win7SP1x64_23418
                     AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/kali/Desktop/dump.raw)
                      PAE type : No PAE
                           DTB : 0x187000L
                          KDBG : 0xf800029f2110L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0xfffff800029f3d00L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2022-04-08 19:05:12 UTC+0000
     Image local date and time : 2022-04-08 12:05:12 -0700
```

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~/Desktop/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f  ../dump.raw kdbgscan
Volatility Foundation Volatility Framework 2.6
**************************************************
Instantiating KDBG using: /home/kali/Desktop/dump.raw WinXPSP2x86 (5.1.0 32bit)
Offset (P)                    : 0x29f2110
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): Win7SP1x64
PsActiveProcessHead           : 0x2a29440
PsLoadedModuleList            : 0x2a47750
KernelBase                    : 0xfffff80002805000

**************************************************
Instantiating KDBG using: /home/kali/Desktop/dump.raw WinXPSP2x86 (5.1.0 32bit)
Offset (P)                    : 0x29f2110
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): Win7SP0x64
PsActiveProcessHead           : 0x2a29440
PsLoadedModuleList            : 0x2a47750
KernelBase                    : 0xfffff80002805000

**************************************************
Instantiating KDBG using: /home/kali/Desktop/dump.raw WinXPSP2x86 (5.1.0 32bit)
Offset (P)                    : 0x29f2110
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): Win2008R2SP1x64
PsActiveProcessHead           : 0x2a29440
PsLoadedModuleList            : 0x2a47750
KernelBase                    : 0xfffff80002805000

**************************************************
Instantiating KDBG using: /home/kali/Desktop/dump.raw WinXPSP2x86 (5.1.0 32bit)
Offset (P)                    : 0x29f2110
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): Win7SP1x64_23418
PsActiveProcessHead           : 0x2a29440
PsLoadedModuleList            : 0x2a47750
KernelBase                    : 0xfffff80002805000

**************************************************
Instantiating KDBG using: /home/kali/Desktop/dump.raw WinXPSP2x86 (5.1.0 32bit)
Offset (P)                    : 0x29f2110
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): Win2008R2SP0x64
PsActiveProcessHead           : 0x2a29440
PsLoadedModuleList            : 0x2a47750
KernelBase                    : 0xfffff80002805000

**************************************************
Instantiating KDBG using: /home/kali/Desktop/dump.raw WinXPSP2x86 (5.1.0 32bit)
Offset (P)                    : 0x29f2110
KDBG owner tag check          : True
```

Sử dụng lệnh

volatility -f dump.raw --profile= Win7SP1x64 filescan

Để liệt kê các file có trong hệ thống, Ta tìm được 2 file liên quan đến flag1

```
0x00000000071f1f20     16      0 R--r-d \Device\HarddiskVolume1\Windows\Fonts\ega40woa.fon
0x00000000071f38c0     16      0 R--rwd \Device\HarddiskVolume1\Users\NHK-InsecLab\Favorites\Links\Suggested Sites.url
0x00000000071f3a10     16      0 RW---- \Device\HarddiskVolume1\Users\TEMP\Desktop\h4lf-fl4g.rar
0x00000000071f3c30     16      0 RW-rwd \Device\HarddiskVolume1\Users\TEMP\AppData\Local\Google\Chrome\User Data\Default\Code Cache\js\b
0x00000000071f3e40     16      0 -W---- \Device\HarddiskVolume1\Windows\Performance\WinSAT\DataStore\2022-04-08 11.29.28.472 GraphicsMed
0x00000000071f4260     16      0 R--rwd \Device\HarddiskVolume1\Windows\System32\en-US\wshim6.dll.mui
```

```
0x000000013fc2fa80     16      0 R--rwd \Device\HarddiskVolume1\Windows\Globalization\MCT\MCT-US\Wallpaper\desktop.ini
0x000000013fc2fcd0     10      0 R--r-- \Device\HarddiskVolume1\Windows\Globalization\MCT\MCT-US\Wallpaper\US-wp3.jpg
0x000000013fc30070     16      0 R--rwd \Device\HarddiskVolume1\Users\NHK-InsecLab\Desktop\flag.txt
0x000000013fc31990     16      0 R--rwd \Device\HarddiskVolume1\Users\NHK-InsecLab\Favorites\Microsoft Websites\Microsoft At Work.url
0x000000013fc342a0     16      0 R--rwd \Device\HarddiskVolume1\Users\NHK-InsecLan\AppData\Local\Microsoft\Windows Mail\Stationery\Desk
```

Sử dụng option dumpfile để trích xuất 2 file tìm được

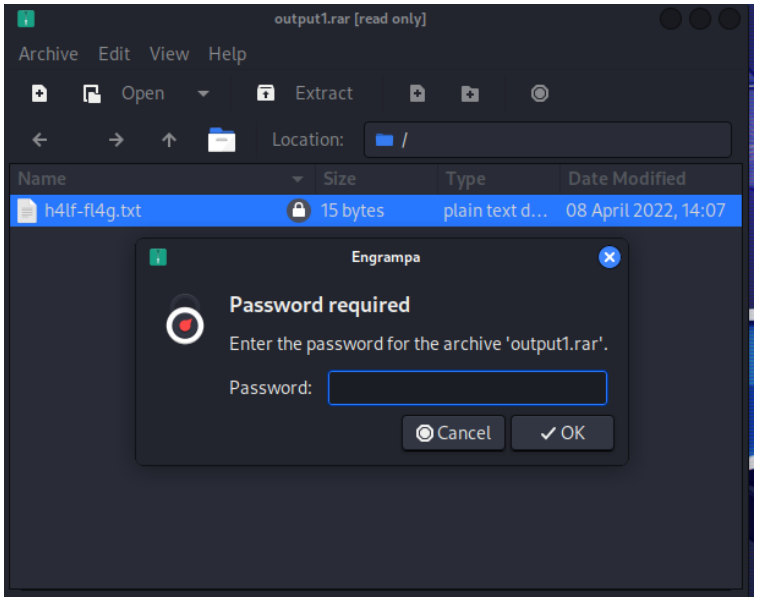
```
┌──(kali㉿kali)-[~/Desktop/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f  ../dump.raw --profile=Win7SP0x64 dumpfiles -Q 0x00000000071f3a10 -D ../
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x071f3a10   None   \Device\HarddiskVolume1\Users\TEMP\Desktop\h4lf-fl4g.rar
```

```
┌──(kali㉿kali)-[~/Desktop/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f   ../dump.raw --profile=Win7SP0x64 dumpfiles -Q 0x000000013fc30070 -D ../
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x13fc30070   None    \Device\HarddiskVolume1\Users\NHK-InsecLab\Desktop\flag.txt
```

Đọc file flag.txt ta được nửa đầu của flag

```
┌──(kali㉿kali)-[~/Desktop]
└─$ cat flag.txt
inseclab{w3lcom3_t0
```

Đối với file h4lf-fl4g.rar có yêu cầu nhập password để giải nén



Ta sử dụng rar2john và John the ripper để crack password, ta tìm được password là r0cky0u

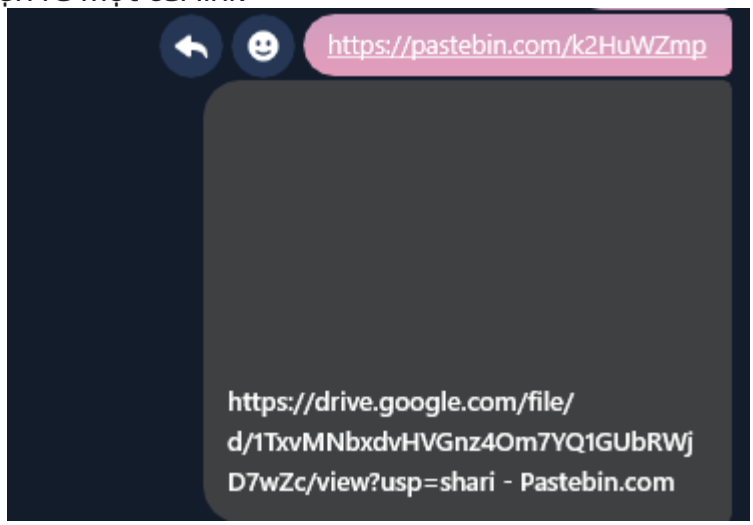
```
┌──(kali㉿kali)-[~/Desktop]
└─$ rar2john output1.rar > output.txt
Created directory: /home/kali/.john

┌──(kali㉿kali)-[~/Desktop]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt output.txt
Using default input encoding: UTF-8
Loaded 1 password hash (RAR5 [PBKDF2-SHA256 256/256 AVX2 8x])
Cost 1 (iteration count) is 32768 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
r0cky0u          (output1.rar)
1g 0:00:00:07 DONE (2023-12-03 21:10) 0.1338g/s 2021p/s 2021c/s 2021C/s drowssap1..lovemike
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Sử dụng password trên để giải nén, có được nửa sau của flag

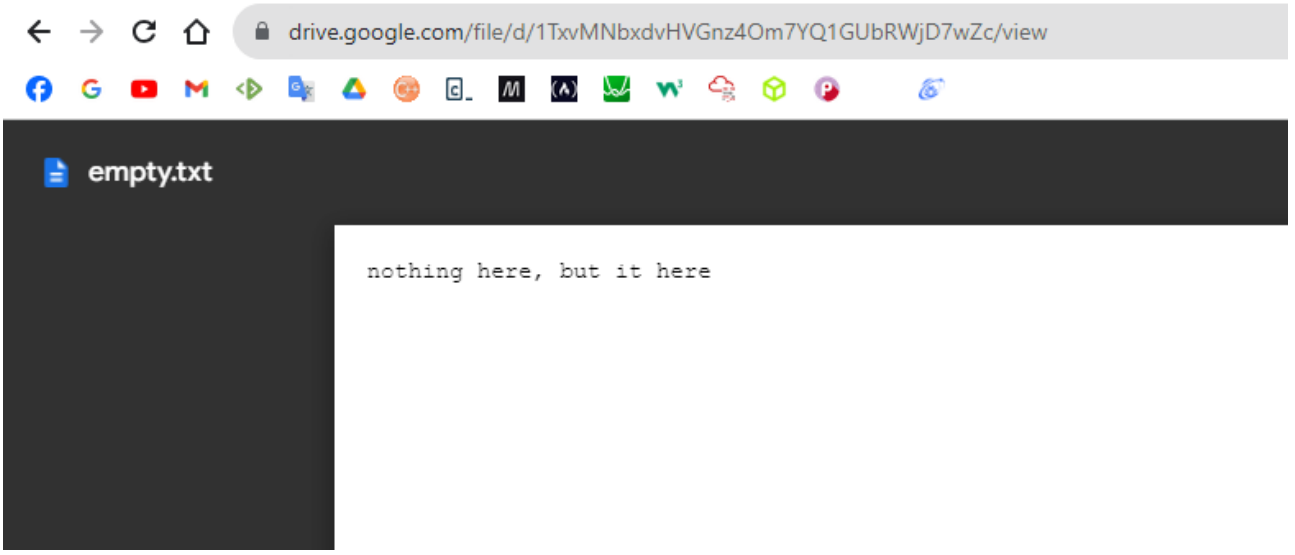
Flag1: inseclab{w3lcom3_t0_th3_w0rld_NHK}

3. **dump.raw (Flag 2):**

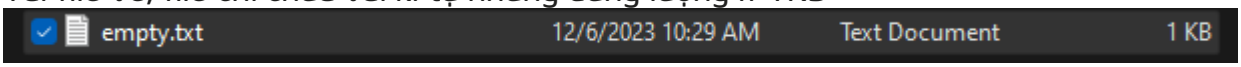
- Dựa theo google về việc tìm lịch sử trình duyệt bằng volatility, ta tìm được một phugins có tên là chromehistory có hỗ trợ. Cú pháp sử dụng: ./volatility_2.6_lin64_standalone --plugins=volatility-plugins/ -f ~/Desktop/dump.raw --profile=Win7SP1x64 chromehistory
- Sau khi chạy, ta sẽ nhận được một đường dẫn như sau: https://pastebin.com/k2HuWZmp
- Tuy nhiên, khi mở ở chrome thì ta không truy cập được, có thể là do lỗi proxy, IP.
- Tuy nhiên khi em gửi vào mess để chuyển qua điện thoại fake IP thử, thì em vô tình thấy nó hiện ra một cái link



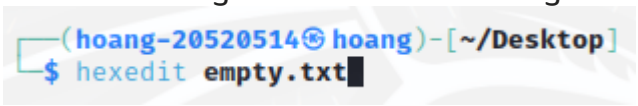
- Em thử hoàn thiện link, sau đó thử trên chrome, em được một trang như sau:



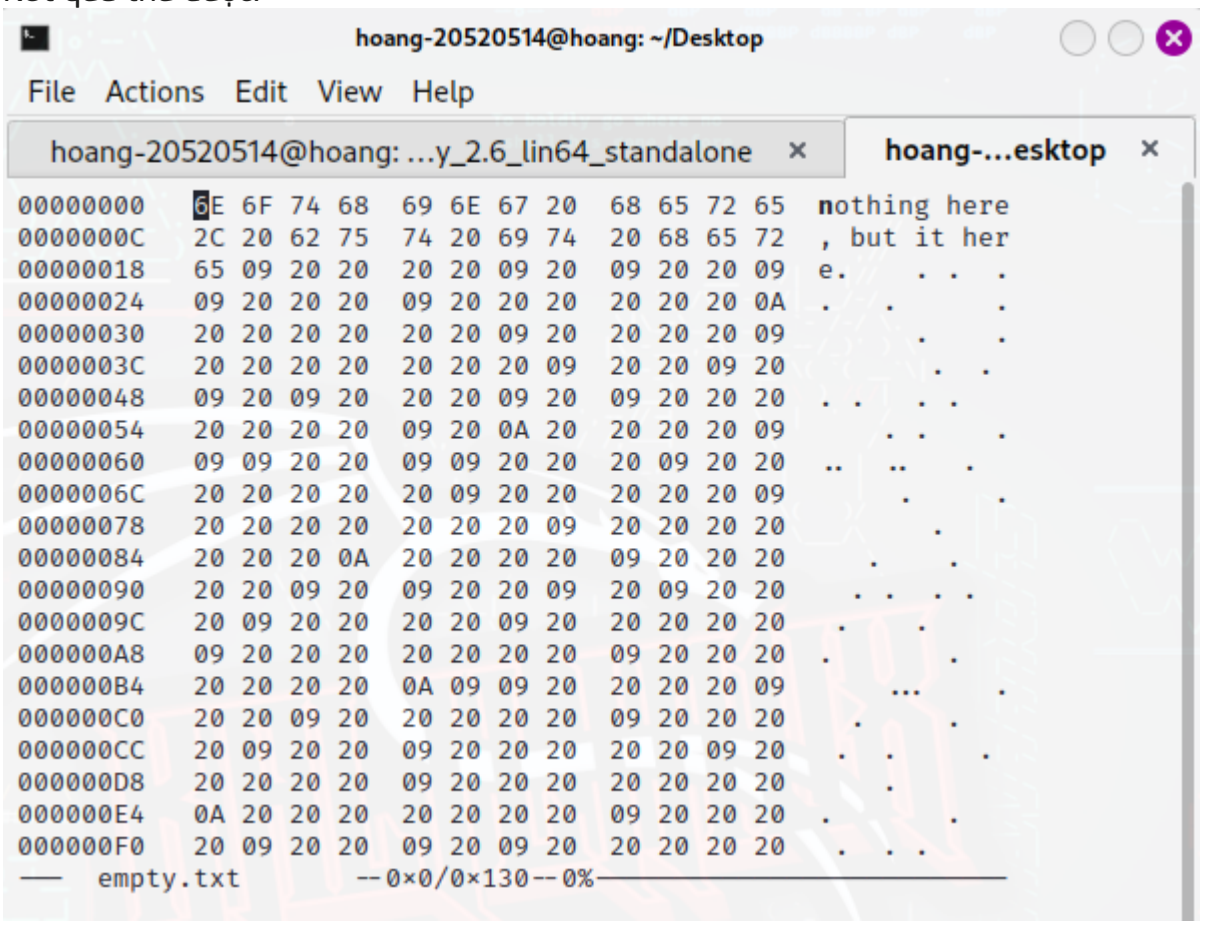
- Tải file về, file chỉ chứa vài kí tự nhưng dung lượng lf 1KB



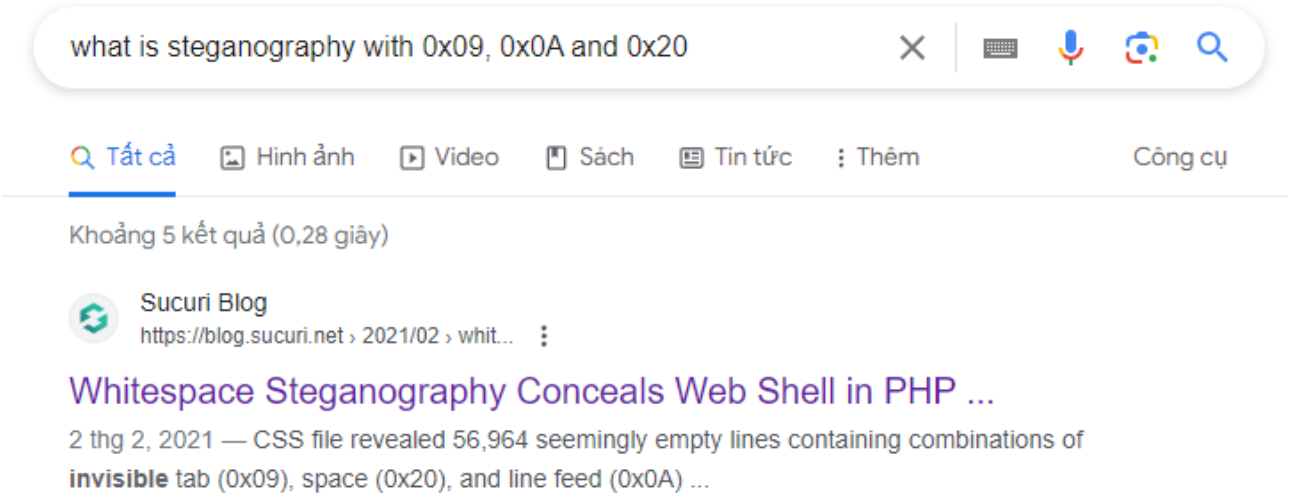
- Em sẽ thử dùng hexedit để xem thông tin file em có bị chèn gì vào không



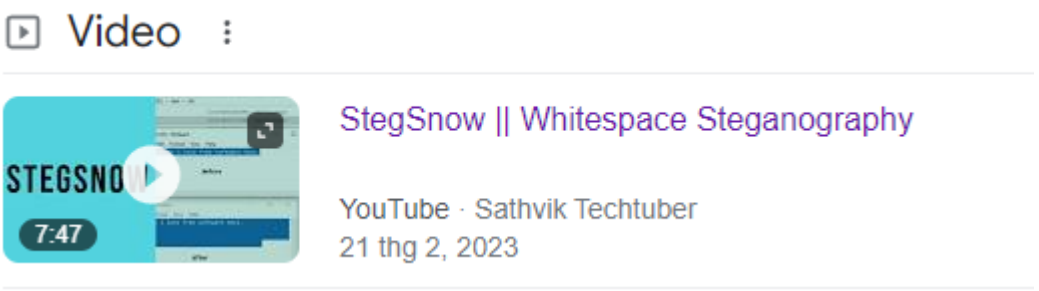
- Kết quả thu được:



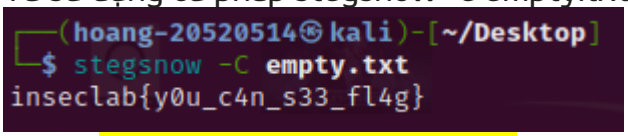
- Đây là một đoạn mã gì đó, chỉ bao gồm khoảng trắng (0x20), tab bị thành dấu chấm (0x9) và xuống dòng (0x0A).
- Em sẽ tìm thử trên google



- Sau một hồi tìm hiểu thì em biết được đây là Whitespace Steganography, em có tìm thì được một video hướng dẫn sử dụng công cụ stegsnow.



- Ta sử dụng cú pháp stegsnow -C empty.txt



- Flag2: inseclab{y0u_c4n_s33_fl4g}

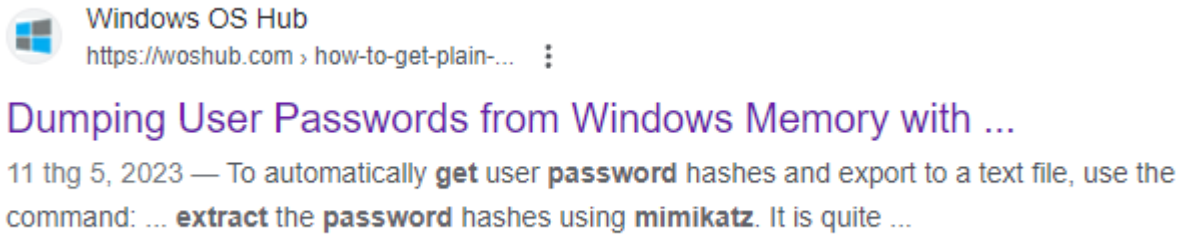
4. **dump.raw (Flag 3):**
   - Đầu tiên, ta sử dụng plugins hivelist để tra ra những vị trí file system

```
┌──(hoang-20520514㉿hoang)-[~/Downloads/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f  ~/Desktop/dump.raw --profile=Win7SP1×64 hivelist
Volatility Foundation Volatility Framework 2.6
Virtual            Physical           Name
------------------ ------------------ ----
0×fffff8a0012a6010 0×000000009e18b010 \??\C:\Users\sshd_server\ntuser.dat
0×fffff8a0012bb270 0×000000004829e270 \??\C:\Users\sshd_server\AppData\Local\Microsoft\Windows\UsrClass.dat
0×fffff8a0017f4010 0×0000000019cda010 \??\C:\Users\TEMP\ntuser.dat
0×fffff8a001882410 0×0000000021a41410 \??\C:\Users\TEMP\AppData\Local\Microsoft\Windows\UsrClass.dat
0×fffff8a0032eb010 0×000000011ff7a010 \??\C:\Windows\AppCompat\Programs\Amcache.hve
0×fffff8a00484c010 0×00000000a8ca5010 \Device\HarddiskVolume1\Boot\BCD
0×fffff8a004ecd010 0×00000000529bb010 \SystemRoot\System32\Config\DEFAULT
0×fffff8a004ed7010 0×0000000052913010 \SystemRoot\System32\Config\SAM
0×fffff8a00000e010 0×00000000a9537010 [no name]
0×fffff8a000024010 0×00000000a9742010 \REGISTRY\MACHINE\SYSTEM
0×fffff8a000063010 0×00000000a9683010 \REGISTRY\MACHINE\HARDWARE
0×fffff8a0005dc010 0×0000000054799010 \SystemRoot\System32\Config\SECURITY
0×fffff8a0005e6010 0×0000000013a00010 \SystemRoot\System32\Config\SOFTWARE
0×fffff8a000e2b010 0×00000000a4cc8010 \??\C:\System Volume Information\Syscache.hve
0×fffff8a000e61010 0×000000000dc00010 \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0×fffff8a000ef1010 0×000000004b8d9010 \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
```

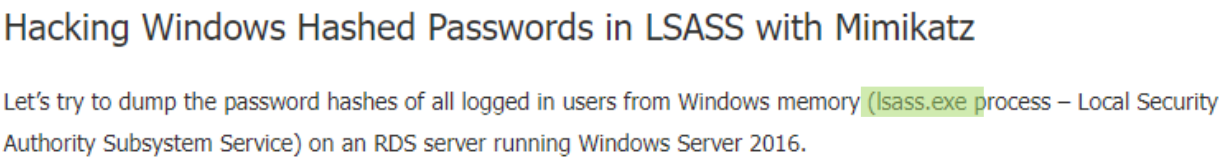
- Sử dụng hashdump để lưu vào file txt

```
┌──(hoang-20520514㉿hoang)-[~/Downloads/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f  ~/Desktop/dump.raw --profile=Win7SP1×64 hashdump -y 0×fffff8a000024010 -s 0×fffff8a004ed7010 > pwdhashes.txt
Volatility Foundation Volatility Framework 2.6

┌──(hoang-20520514㉿hoang)-[~/Downloads/volatility_2.6_lin64_standalone]
└─$ ls
AUTHORS.txt  CREDITS.txt  LEGAL.txt  LICENSE.txt  pwdhashes.txt  README.txt  volatility_2.6_lin64_standalone

┌──(hoang-20520514㉿hoang)-[~/Downloads/volatility_2.6_lin64_standalone]
└─$ cat pwdhashes.txt
Administrator:500:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
NHK-InsecLab:1000:aad3b435b51404eeaad3b435b51404ee:141be588e38b145c4e1f274b646898eb:::
sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16cfc061c3359db455d00ec27035:::
```

- Dựa vào đây, ta có thể lấy được NT hash và LM hash của user NHK-InsecLab
- Sau một khoảng thời gian crack bằng John và Hashcat vẫn không ra, em có tìm thử trên mạng, và người ta có một cách như trong blog sau:



- Cách này có chỉ chúng ta sẽ thực hiện bằng mimikatz, và có một điều kiện cần là tiến trình lsass.exe phải chạy.



- Ta sử dụng plugin pslist để in ra các tiến trình

```
0xfffffa800cdfc570 wininit.exe          392    332     3     76      0      0 2022-04-08 17:44:22 UTC+0000
0xfffffa800bdfa880 csrss.exe            404    384    16    279      1      0 2022-04-08 17:44:22 UTC+0000
0xfffffa8005a1cb10 services.exe         456    392     7    223      0      0 2022-04-08 17:44:22 UTC+0000
0xfffffa8005a276f0 lsass.exe            464    392     7    598      0      0 2022-04-08 17:44:22 UTC+0000
0xfffffa8005a1f750 lsm.exe              472    392     9    157      0      0 2022-04-08 17:44:22 UTC+0000
0xfffffa8005a248f0 winlogon.exe         496    384     3    110      1      0 2022-04-08 17:44:22 UTC+0000
0xfffffa8005a6db10 svchost.exe          616    456    10    356      0      0 2022-04-08 17:44:23 UTC+0000
0xfffffa8005ab5b10 svchost.exe          684    456     8    274      0      0 2022-04-08 17:44:23 UTC+0000
0xfffffa8005ae69c0 svchost.exe          736    456    19    437      0      0 2022-04-08 17:44:23 UTC+0000
0xfffffa8005b1bb10 svchost.exe          844    456    17    425      0      0 2022-04-08 17:44:23 UTC+0000
0xfffffa8005b64870 svchost.exe          888    456    18    633      0      0 2022-04-08 17:44:23 UTC+0000
0xfffffa8005a86b10 svchost.exe          924    456    29    937      0      0 2022-04-08 17:44:23 UTC+0000
0xfffffa8005b96b10 svchost.exe          980    456     6    136      0      0 2022-04-08 17:44:23 UTC+0000
0xfffffa8005c04870 svchost.exe          384    456    15    482      0      0 2022-04-08 17:44:23 UTC+0000
0xfffffa8005c9d8f0 spoolsv.exe         1048    456    13    270      0      0 2022-04-08 17:44:23 UTC+0000
0xfffffa8005cd9870 svchost.exe         1084    456    17    310      0      0 2022-04-08 17:44:23 UTC+0000
0xfffffa8005d1b320 svchost.exe         1204    456    13    334      0      0 2022-04-08 17:44:23 UTC+0000
```

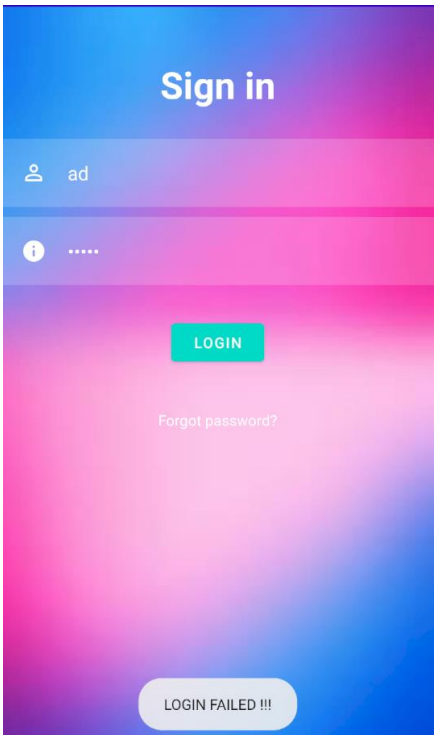
- Có sự xuất hiện của tiến trình này
- Cuối cùng, ta sử dụng mimikatz để lấy password với cú pháp

python vol.py --plugins=/home/ubuntu/community/FrancescoPicasso/ -f /home/ubuntu/dump.raw --profile=Win7SP1x64 mimikatz

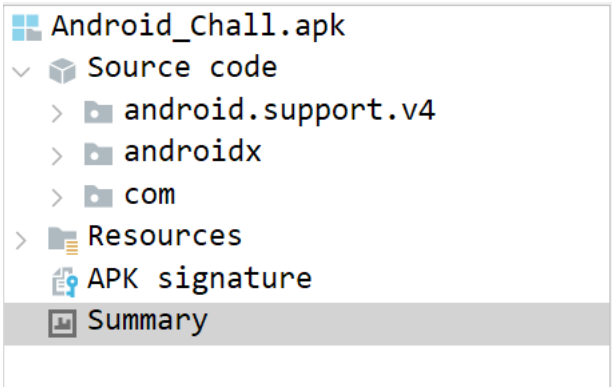
- Ta có được Flag là: AntiNHK
- Flag3: inseclab{AntiNHK}

5. Android_Chall.apk:

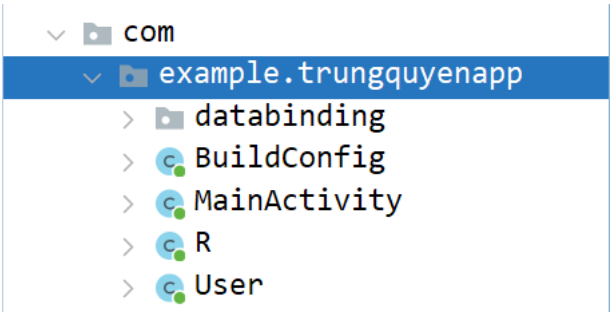
File apk cung cấp cho ta một giao diện đăng nhập cơ bản gồm username và password, nếu chúng ta đăng nhập sai sẽ chỉ hiển thị LOGIN FAIILED !!!



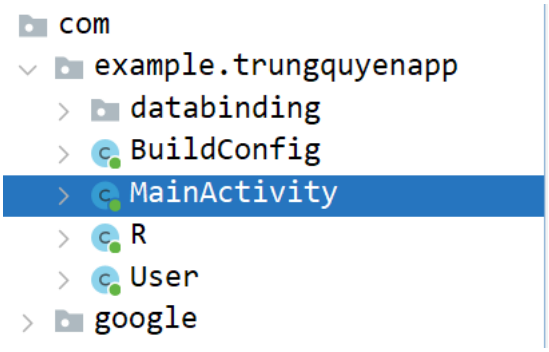
Sau đó, sử dụng tool jadx để có thể dump được resource của file apk ra để tiếp tục phân tích:



Tại trường source code/com/example.trungquyenapp, ta có được cấu trúc chính của file apk trên



Trong trường MainActivity, ta có 1 function đáng nghi




```
/* renamed from: lambda$onCreate$0$com-example-trungquyenapp-MainActivity  reason: not valid java name */
public /* synthetic */ void m26lambda$onCreate$0$comexampletrungquyenappMainActivity(TextView username, TextView pa
    if (username.getText().toString().equals("admin") && password.getText().toString().equals("admin")) {
        Toast.makeText(this, "LOGIN SUCCESSFUL", 0).show();
        Intent intent = new Intent(this, User.class);
        startActivity(intent);
        return;
    }
    Toast.makeText(this, "LOGIN FAILED !!!", 0).show();
}
```

Hàm trên đóng nhiệm vụ kiểm tra username và password mà người dùng nhập vào, nếu cả 2 username và password trên đều là admin, sẽ hiển thị thông báo LOGIN SUCCESSFUL và chuyển sang page khác qua Intent.

Vậy ta sử dụng “admin”- “admin” để đăng nhập vào ứng dụng và lấy được flag của file

Here your flag:

inseclab{w3lc0m3_t0_@ndr01d_Ch@ll3ng3!!!}

Flag : **inseclab{w31c0m3_t0_@ndr01d_Ch@ll3ng3!!!}**

Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hiện bài tập theo yêu cầu, hướng dẫn.
- Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

Báo cáo:

- File .DOCX và .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
- Đặt tên theo định dạng: [Mã lớp]-ExeX_GroupY. (trong đó X là Thứ tự Bài tập, Y là mã số thứ tự nhóm trong danh sách mà GV phụ trách công bố).

  *Ví dụ: [*NT101.K11.ANTT*]-Exe01_Group03.*
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm bài nộp.
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá:

- Hoàn thành tốt yêu cầu được giao.
- Có nội dung mở rộng, ứng dụng.

Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.

HẾT