

LAB 04

Thầy Mai Hoàng Đình
Trường đại học FPT

Người thực hiện

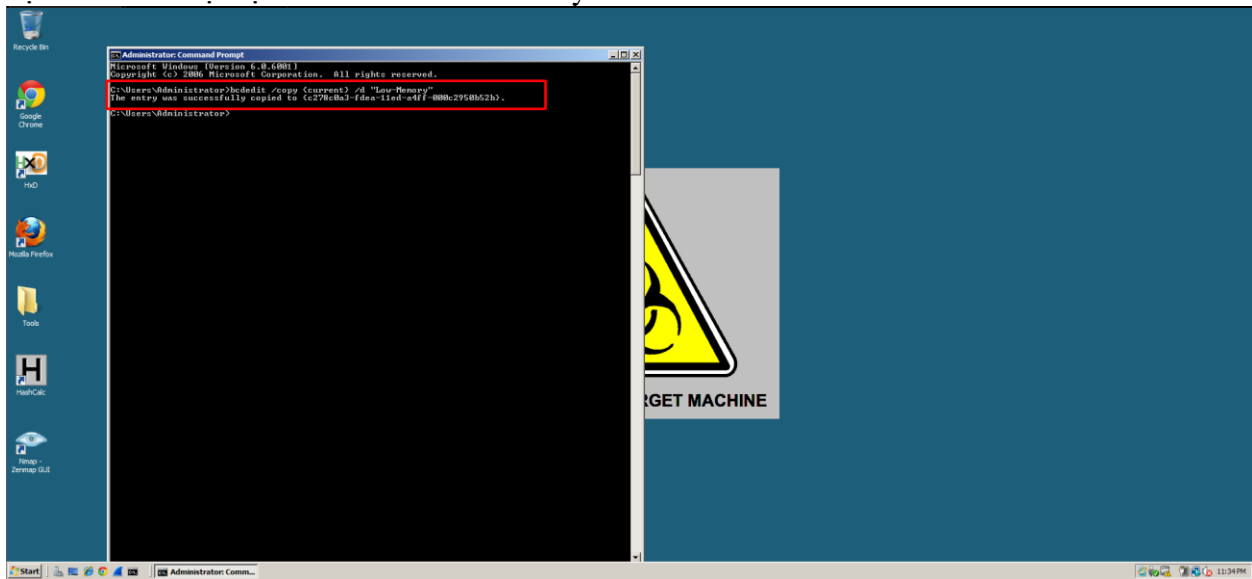
Đặng Hoàng Nguyên

Reducing the Available RAM

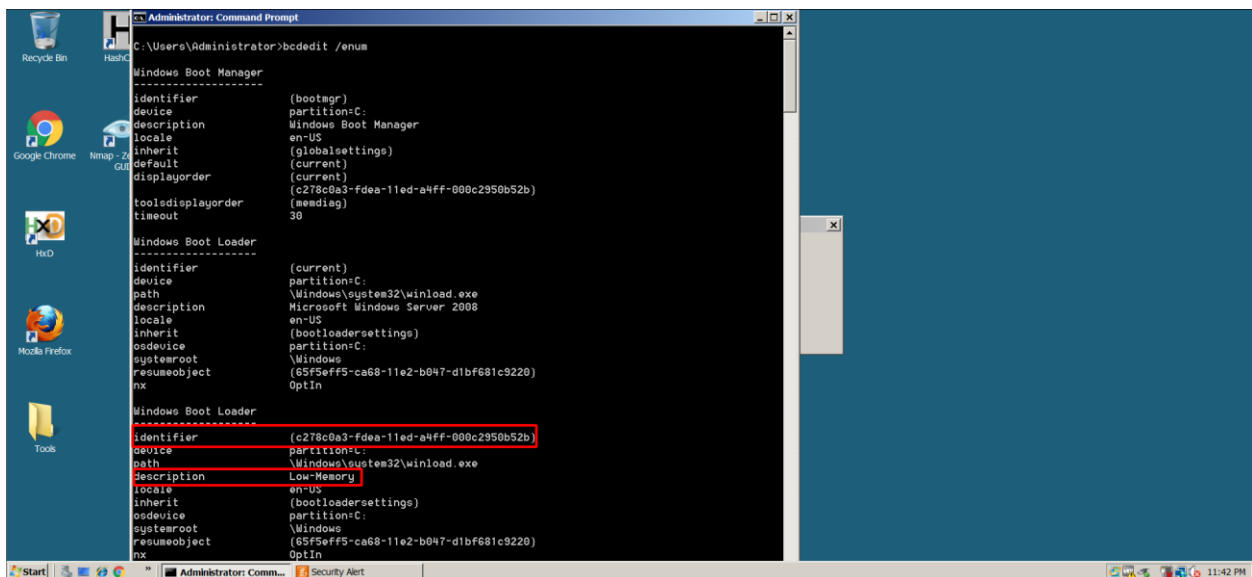
Nhấn vào start, hoặc nhấn CTR + R rồi nhấn CMD và chạy dưới quyền admin, sau đó nhập câu lệnh sau đây:

```
bcdedit /copy {current} /d "Low-Memory"
```

Câu lệnh này cho phép chúng ta sẽ tạo ra một boot entry bên trong Boot Configuration Data của hệ điều hành hiện tại với tên là Low-Memory.

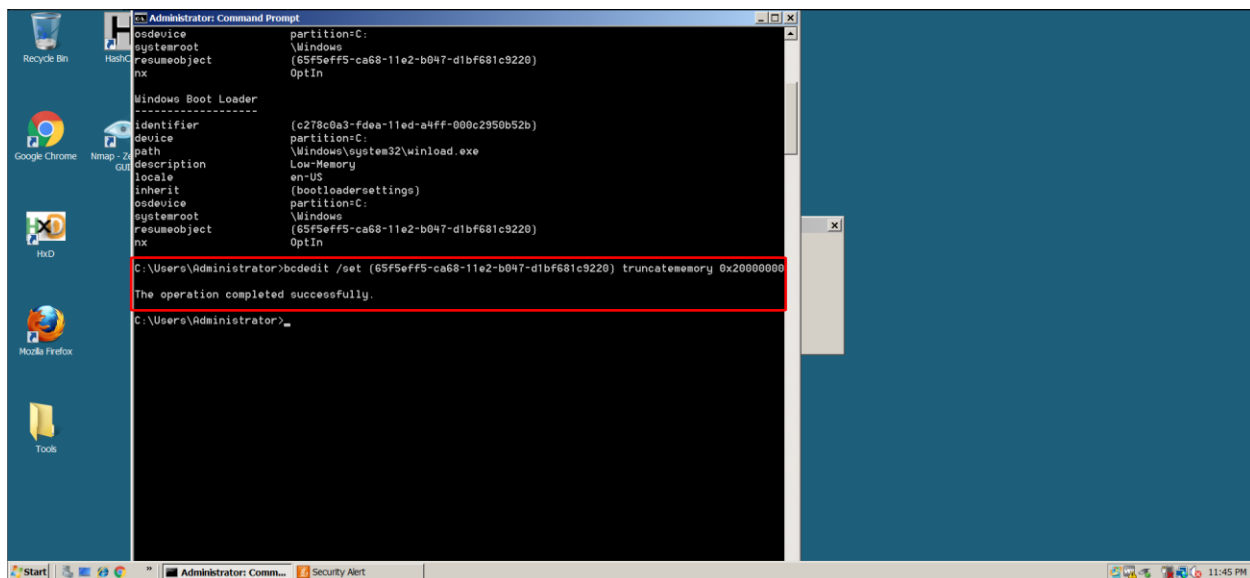


Sử dụng câu lệnh `bcdedit /enums` để check lại xem là có chưa:

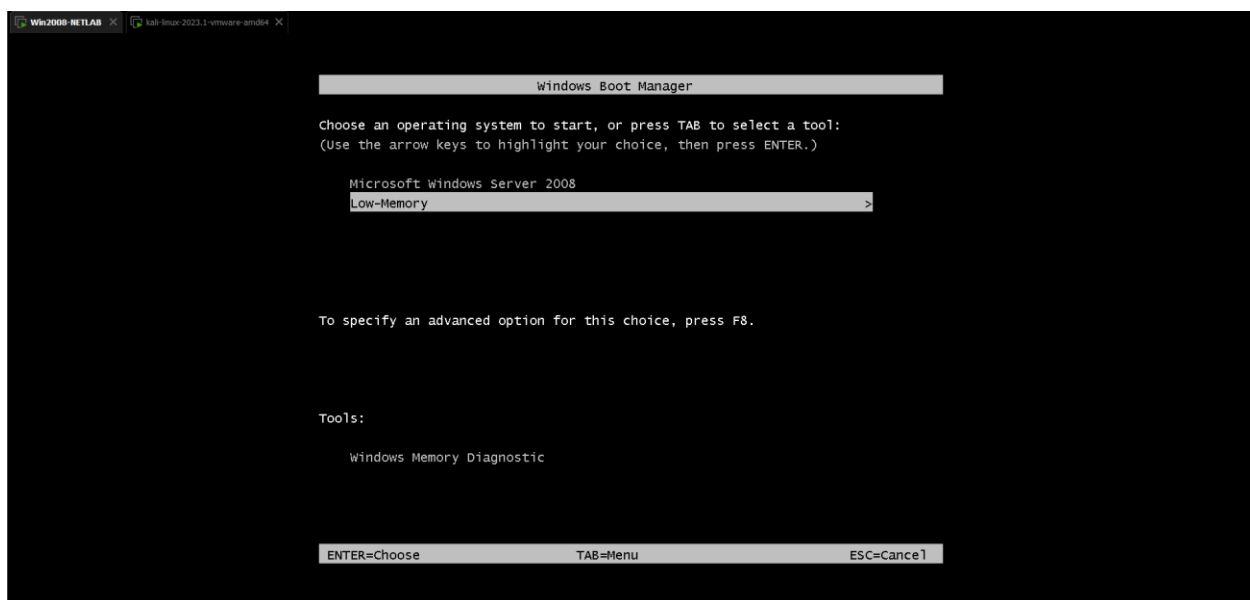


Như ta có thể thấy được là chúng ta đã tạo thành công. Sau đó sử dụng câu lệnh `bcdedit /set {identifier của máy bạn} truncatememory 0x20000000`

- Câu lệnh này có nghĩa là set bộ nhớ ram cho identifier của BCD đó là **0x20000000**. Ở đây 0x20000000 có nghĩa là 512MB. → Set bộ nhớ ram cho identifier đó là 512MB



Sau đó boot vào bên trong low-memory trong phần boot của Windows



Vào thành công chúng ta sẽ tới bước tiếp theo là tạo bằng chứng bằng công cụ ftkimager

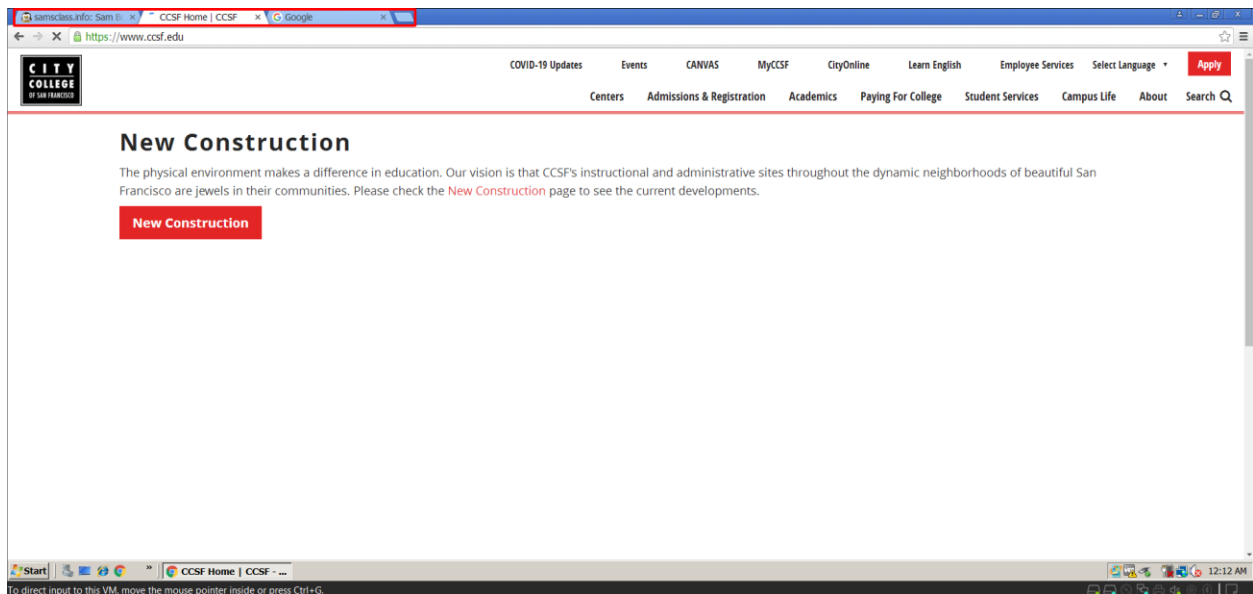
Creating Evidence

Lên trang web của FTK: <http://www.accessdata.com/support/product-downloads> tải bản mới nhất của FTK về. Hoặc nếu đang dùng window 2008 server, chúng ta sẽ lên trang web https://download.informer.com/win-1192693607-764df888-632a2e35-27868c05d6c0543f17-aff9fe3f5b80d0e2e-6412598663-1188631454/accessdata_ftk_imager.exe để tải file về



Sau đó lên Google Chrome và tìm kiếm ba trang web sau:

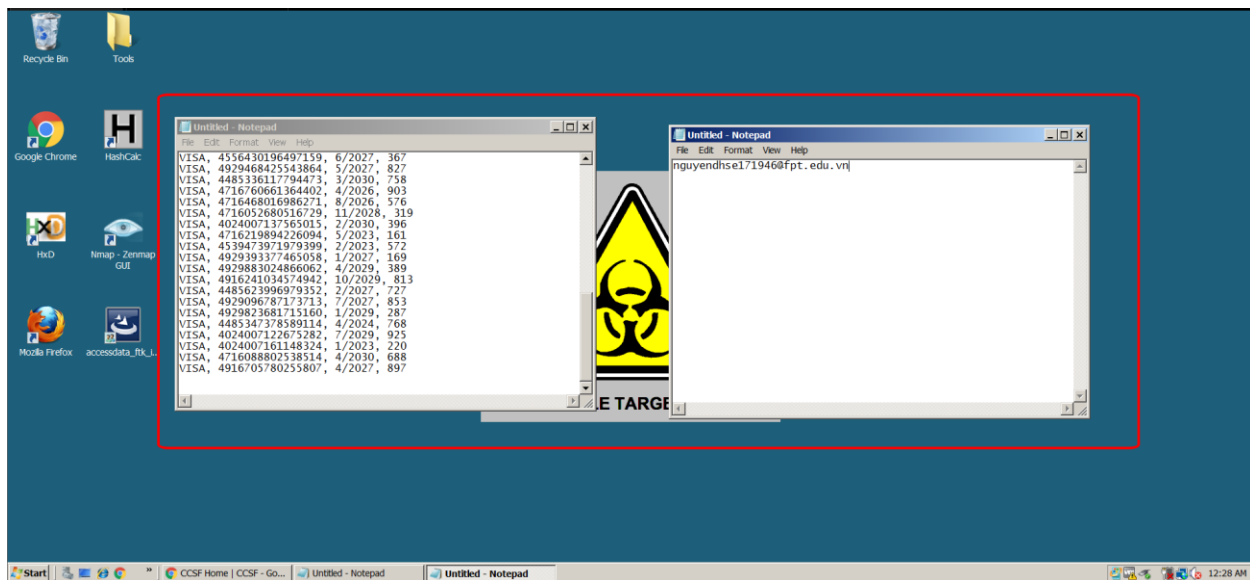
- samsclass.info
- ccsf.edu
- google.com



Sau đó, chúng ta mở notepad lên, và paste thông tin của những số thẻ visa fake vào trong notepad.

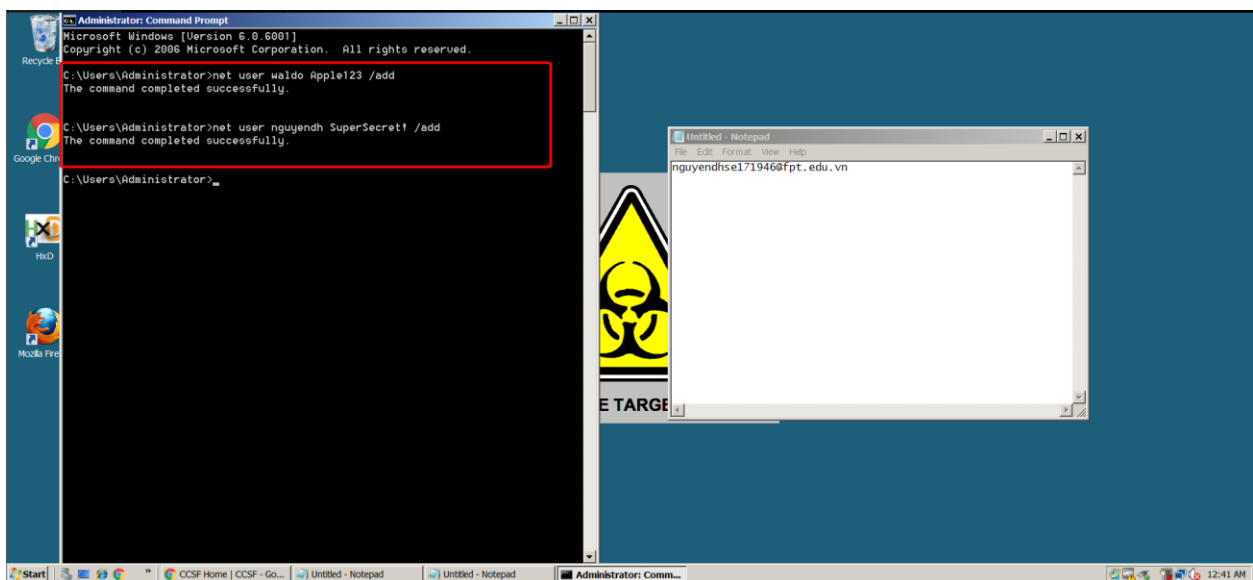
Thông tin của những thẻ visa fake có thể lên một số trang web là có. Lưu ý là chỉ paste chứ không save file lại

Tiếp theo mở thêm một notepad và điền email của mình vào như hình dưới đây

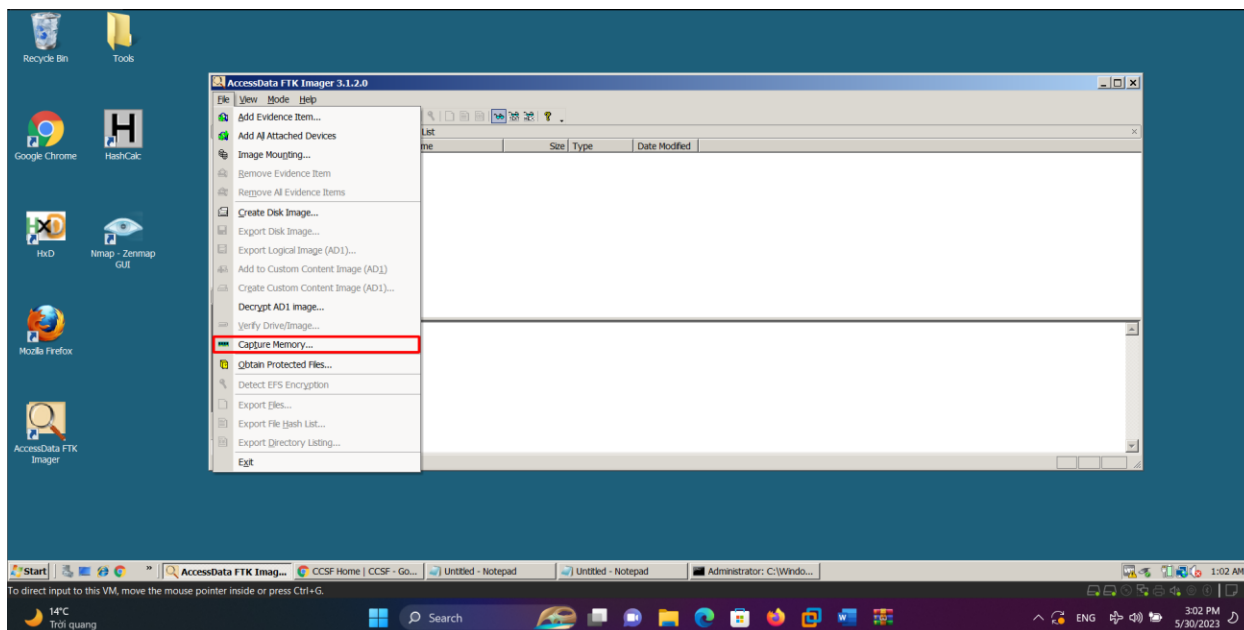


Mở thêm một cmd nữa và bắt đầu tạo thêm hai user nữa, một tên là waldo và một là tên của chính mình theo câu lệnh **net user waldo Apple123 /add** và **net user nguyendh SuperSecret! /add**

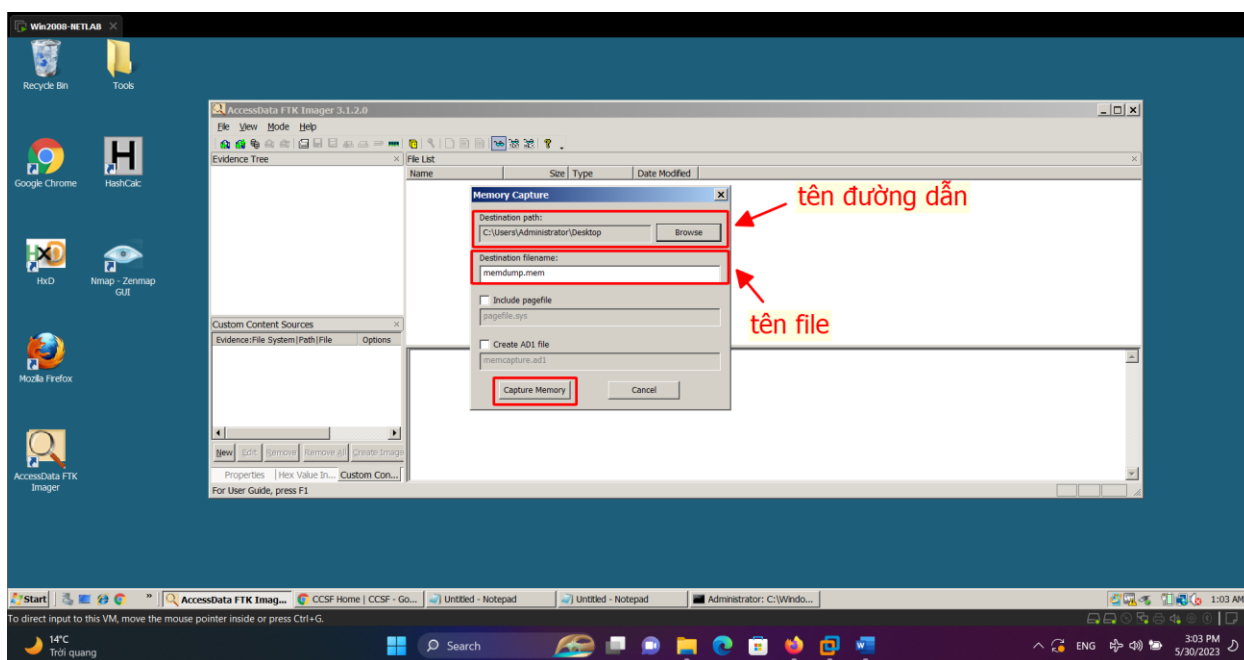
- Câu lệnh này có nghĩa là tạo hai user có tên là waldo và nguyendh với password lần lượt là Apple123 và SuperSecret!



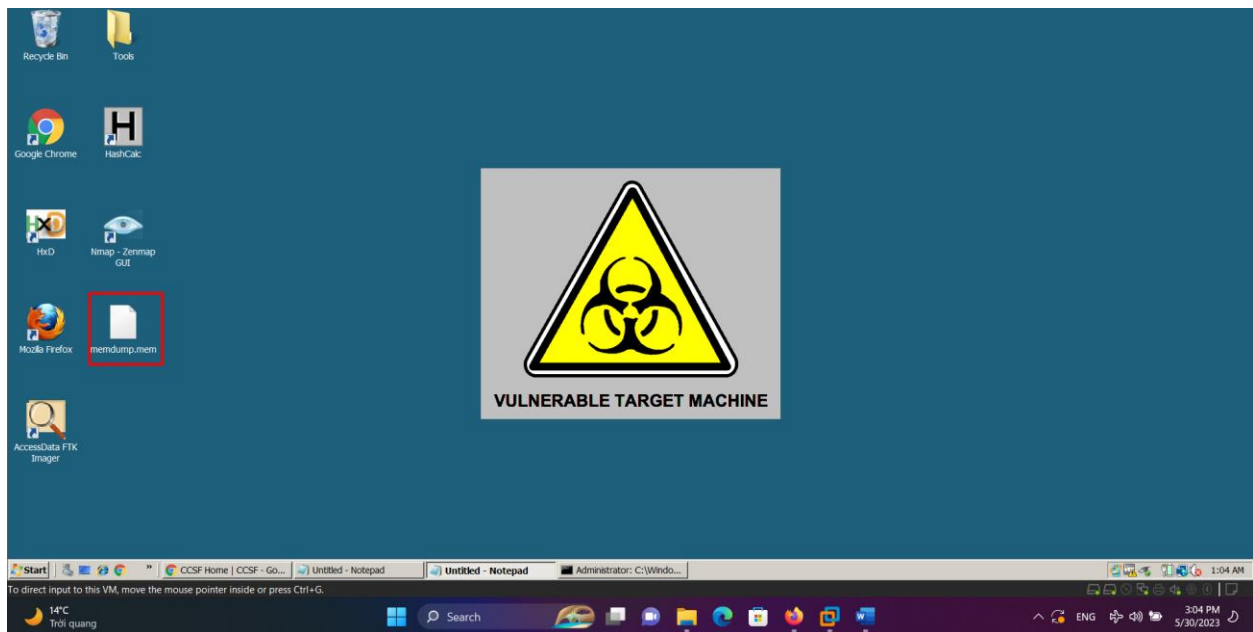
Sau khi set up xong, chúng ta sẽ bắt đầu dump memory bằng ổ đĩa thông qua FTK Imager. Vào trong toolbar → File → Capture Memory như trong hình



Sau đó sẽ bắt chúng ta chọn đường dẫn và tên file, trong trường hợp này chúng ta sẽ lưu ở Desktop với tên là memdump.mem



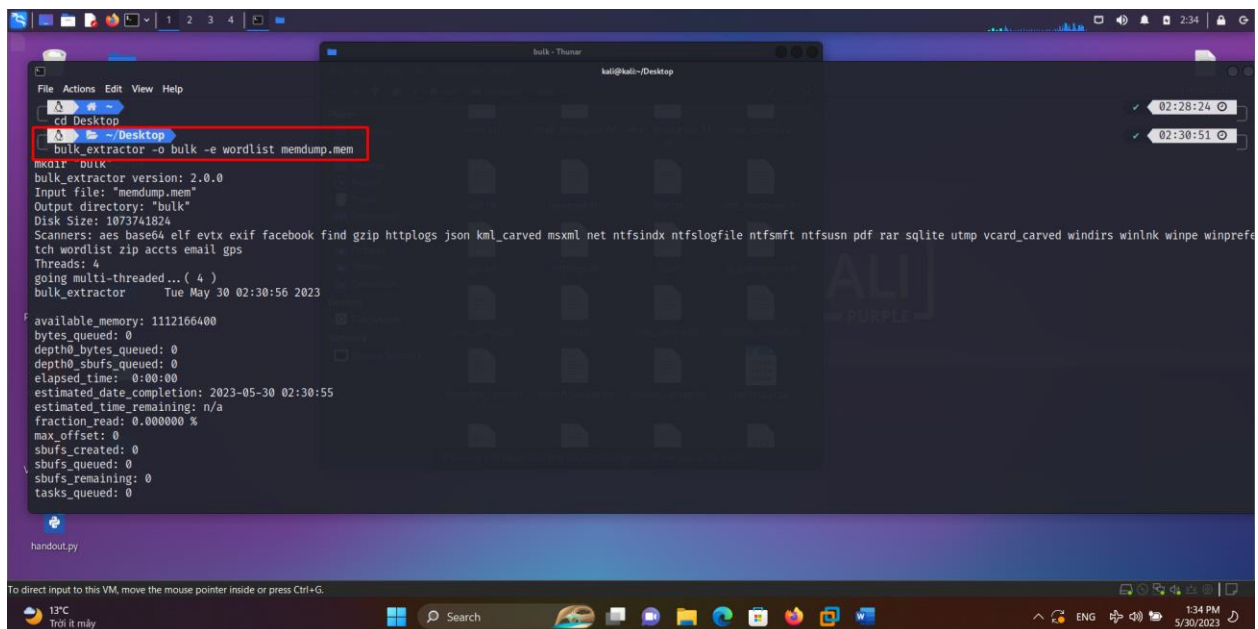
Sau khi xong chúng ta có thể kiểm tra lại trong đường dẫn đã có file chưa

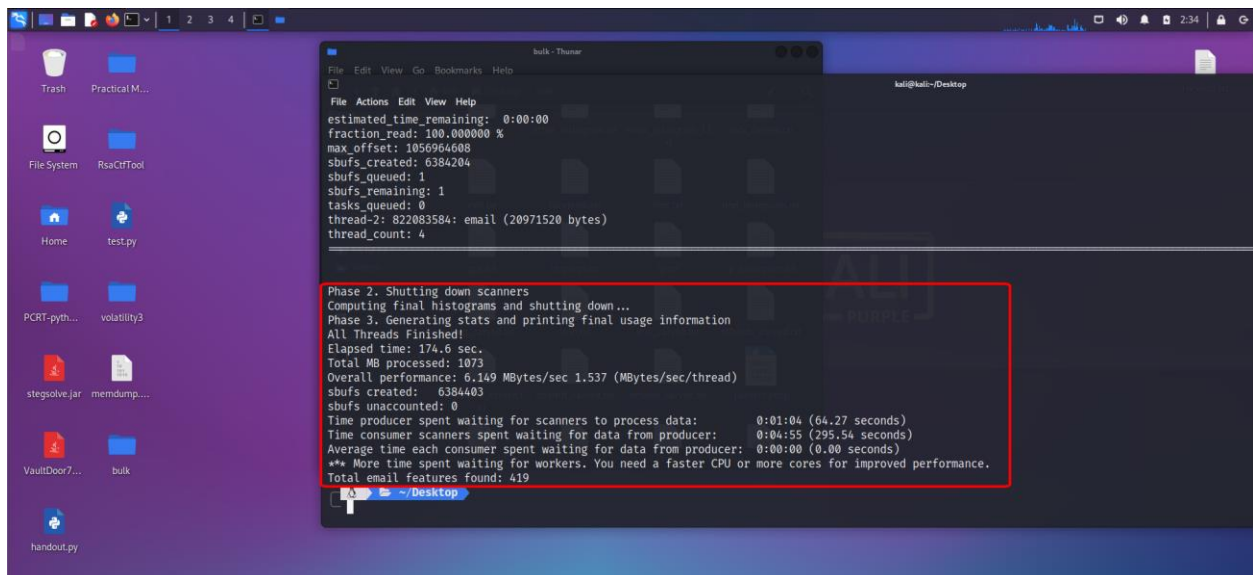


Như ta đã thấy ở đây sau khi kiểm tra đã có sẵn file memdump.

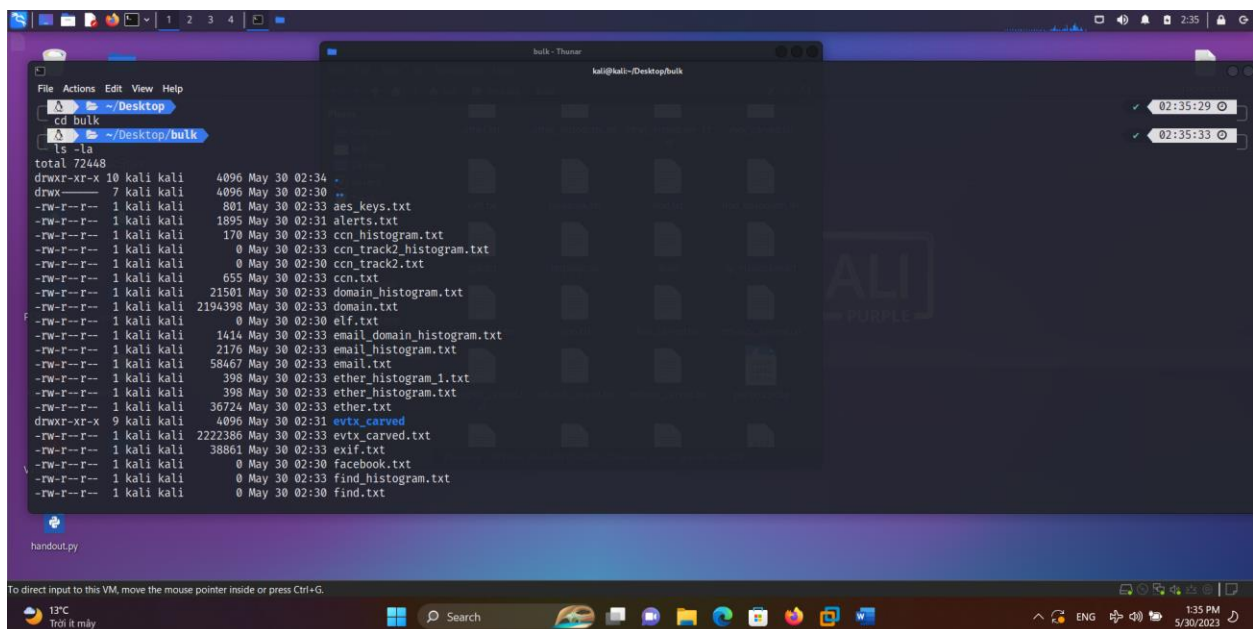
Vào bên trong kali, import file memdump vào bằng cách kéo thả. Sau khi xong chúng ta sẽ sử dụng **bulk_extractor** để có thể xem bên trong file memdump đó chứa gì bằng cách sử dụng câu lệnh sau đây: **bulk_extractor -o bulk -e wordlist memdump.mem**

Các kết quả tìm kiếm sẽ được lưu vào một thư mục có tên là **bulk**. **Wordlist** là sử dụng nhwunxg tên phổ biến để có thể scan trong đó có gì



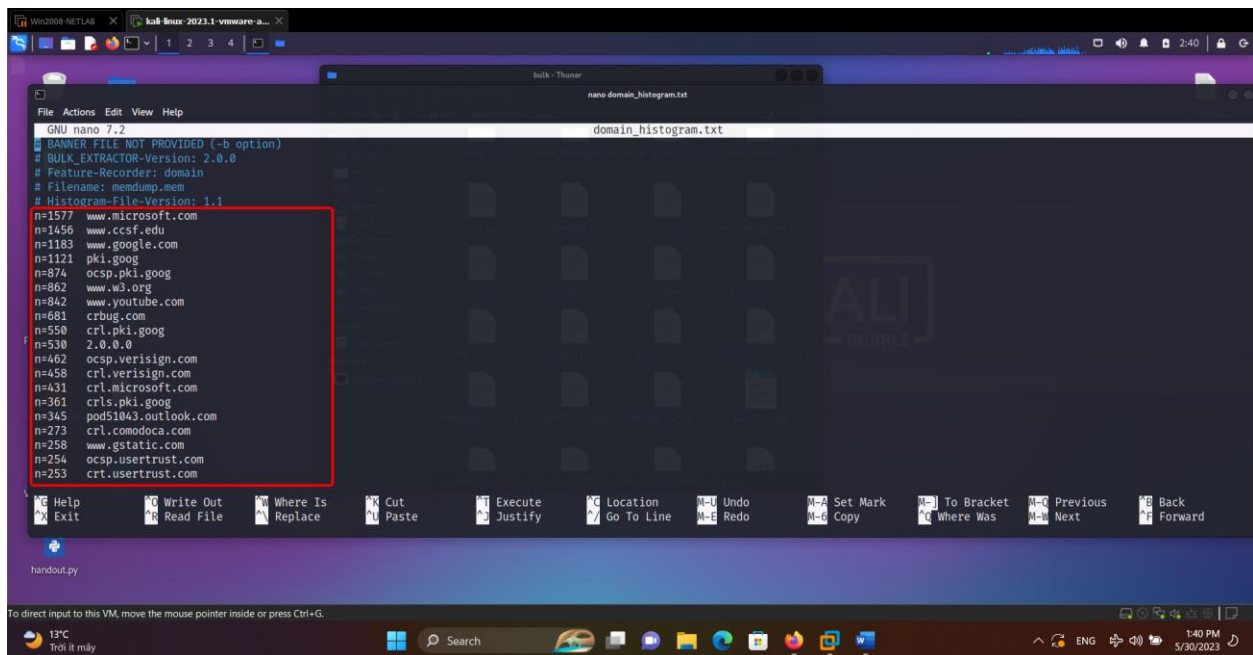


Như vậy là chúng ta đã scan hoàn tất, vào bên trong thư mục bulk xem có gì



Ở đây chúng ta có thể thấy được khá là nhiều file txt đã được lấy ra từ bên trong file memdump đấy. Chúng ta sẽ nhìn qua một số file như là Domain Names, Telephone Numbers, Credit Card Numbers, Word List, Email Addresses:

Chúng ta có thể xem số lần mà chúng ta truy cập, history truy cập trang web thông qua câu lệnh sau đây: **nano/vi/cat domain_histogram.txt**

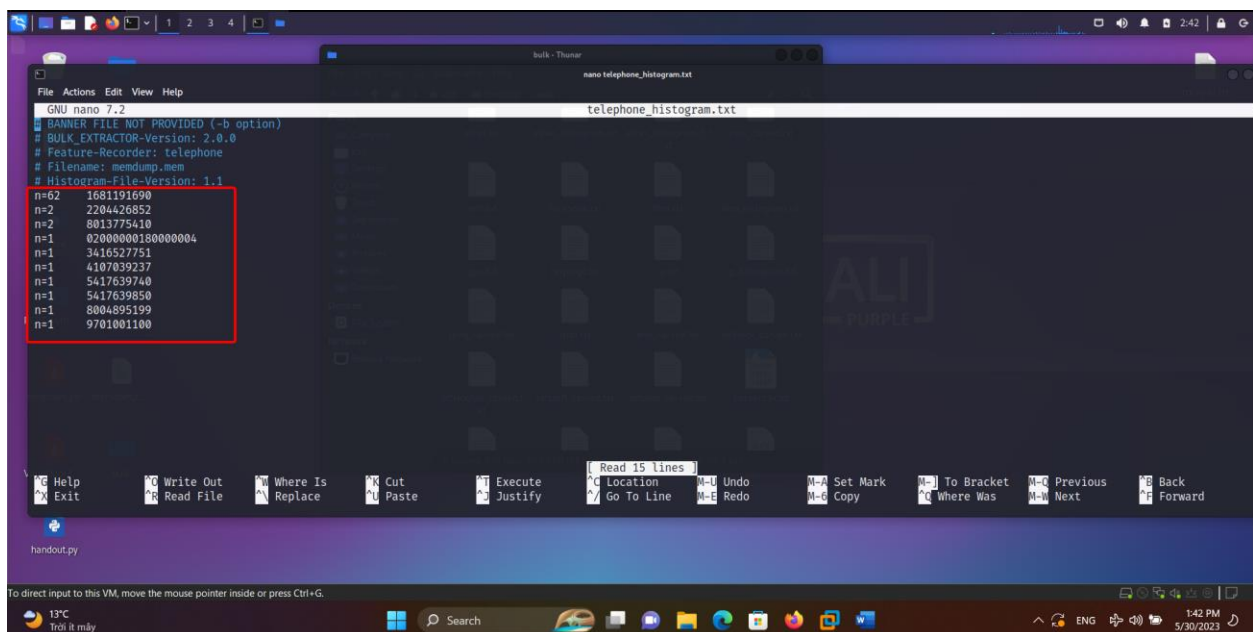


```
GNU nano 7.2
# BANNER FILE NOT PROVIDED (-b option)
# BULK_EXTRACTOR-Version: 2.0.0
# Feature-Recorder: domain
# Filename: memdump.mem
# Histogram-File-Version: 1.1
n=1577 www.microsoft.com
n=1456 www.ccsf.edu
n=1183 www.google.com
n=1121 pki.goog
n=874 ocsp.pki.goog
n=862 www.w3.org
n=842 www.youtube.com
n=681 crbug.com
n=550 crl.pki.goog
n=530 2.0.0.0
n=462 ocsp.verisign.com
n=458 crl.verisign.com
n=431 crl.microsoft.com
n=361 crls.pki.goog
n=345 pod51043.outlook.com
n=273 crl.comodoca.com
n=258 www.gstatic.com
n=254 ocsp.usertrust.com
n=253 crt.usertrust.com
```

- Như ta thấy bên trái là số lần chúng ta vào trang web nào đó, bên phải là tên trang web chúng ta đã vào.

Chúng ta sẽ thấy số điện thoại của mình, khi chúng ta nhập số đó vào form AccessData yêu cầu điền để tải xuống FTK Imager. Thông qua câu lệnh: nano/vi/cat telephone_histogram.txt

- Bên trái là số lần sử dụng, bên phải là tên số điện thoại dung để đăng kí



```
GNU nano 7.2
# BANNER FILE NOT PROVIDED (-b option)
# BULK_EXTRACTOR-Version: 2.0.0
# Feature-Recorder: telephone
# Filename: memdump.mem
# Histogram-File-Version: 1.1
n=62 1681191690
n=2 2204426852
n=2 8013775410
n=1 020000001800000004
n=1 3416527751
n=1 4107839237
n=1 5417639740
n=1 5417639850
n=1 8004895199
n=1 9701001100
```

Ngoài ra chúng ta còn có thể xem số thẻ tín dụng dưới command sau đây: nano/vi/cat ccn_histogram.txt

- Bên trái là số lần sử dụng, bên phải là tên số thẻ dùng để đăng kí

```

GNU nano 7.2
# BANNER FILE NOT PROVIDED (-b option)
# BULK_EXTRACTOR-Version: 2.0.0
# Feature-Recorder: ccn
# Filename: memdump.mem
# Histogram-File-Version: 1.1
n=4 4916705780255807
  
```

Chúng ta thấy các từ được tìm thấy và số lần từng từ được tìm thấy. Danh sách này hữu ích như một từ điển khi bẻ khóa các tệp hoặc thư mục được mã hóa. Thông qua câu lệnh sau: **nano/cat/vi wordlist.txt**

- Bên trái là số lần sử dụng, bên phải là tên đã được sử dụng

```

GNU nano 7.2
# BANNER FILE NOT PROVIDED (-b option)
# BULK_EXTRACTOR-Version: 2.0.0
# Feature-Recorder: wordlist
# Filename: memdump.mem
# Feature-File-Version: 1.1
31971 TCPAu$
31995 fsfsfu
32060 fy[Zfvyf
32148 occurred
32159 BOOTMGR
32170 missing
32180 BOOTMGR
32191 compressed
32210 Ctrl+Alt+Del
32226 restart
33282 fsfPQqVfw
33300 f*f*fyf
33399 fTVgf
33422 fPfPgf
33465 fZfYfBfQfV
33503 fYfZfQfVf
34061 fRfQfRf
34191 f*fPQfQf3
34412 fPfsfQf
  
```

Chúng ta sẽ thấy các địa chỉ email được sử dụng trên máy tính này và số lần từng được truy cập. Thông qua câu lệnh sau: **nano/cat/vi email_histogram.txt**

- Bên trái là số lần sử dụng, ở giữa là email đã được sử dụng, bên phải là định dạng mà mail đó sử dụng

```

GNU nano 7.2 email_histogram.txt
# BANNER FILE NOT PROVIDED (-b option)
# BULK_EXTRACTOR-Version: 2.0.0
# Feature-Recorder: email
# Filename: memdump.mem
# Histogram-File-Version: 1.1
n=40 premium-server@hawte.com (utf16=2)
n=38 info@valicert.com (utf16=3)
n=29 eay@cryptsoft.com
n=27 support@accessdata.com (utf16=27)
n=16 pki@sk.ee
n=14 appro@openssl.org
n=11 cps-requests@verisign.com
n=10 chambersignroot@chambersign.org
n=10 feste@feste.org
n=10 info@cert.at
n=9 info@dignotar.nl
n=9 info@netlock.hu
n=8 acrs@economia.gob.mx
n=8 certificate@trustcenter.de
n=8 chambersroot@chambersign.org
n=8 cps@netlock.net
n=8 info@netlock.net
n=8 server-certs@hawte.com
n=7 ellenorzes@netlock.net
  
```

Sử dụng volatility3, chúng ta sẽ bắt đầu đi phân tích. Để xem nhữn thông tin cơ bản đang chạy của memdump, chúng ta sẽ dùng câu lệnh **vol.py -f <memdump> windows.info**

Bản volatility2 so với 3 thì vol2 sẽ hiện ra những profile có thể phù hợp và sử dụng profile đó để cho sử dụng cho các câu lệnh về sau. Đến với vol3 thì đã bỏ đi chuyện scan profile, thay vào đó là những thông tin cơ bản về máy.

```

kali@kali:~/Desktop/volatility3
python3 vol.py -f ~/Desktop/memdump.mem windows.info
Volatility 3 Framework 2.4.2
Progress: 100.00 PDB scanning finished
Variable Value
Kernel Base 0x8181b000
DTB 0x122000
Symbols file:///home/kali/Desktop/volatility3/volatility3/symbols/windows/ntkrpamp.pdb/37D328E3BAE5460F8E662756ED80951D-2.json.xz
Is64Bit False
IsPAE True
layer_name 0 WindowsIntelPAE
memory_layer 1 FileLayer
KdDebuggerDataBlock 0x81912c90
NTBuildLab 6001.18000.x86fre.longhorn_rtm.0
CSDVersion 1
KdVersionBlock 0x81912c68
Major/Minor 15.6001
MachineType 332
KeNumberProcessors 3405774849
SystemTime 2023-05-30 03:24:38
NtSystemRoot C:\Windows
NtProductType NtProductServer
NtMajorVersion 6
NtMinorVersion 0
PE MajorOperatingSystemVersion 6
PE MinorOperatingSystemVersion 0
PE Machine 332
  
```

Tiếp tới chúng ta sẽ xem bên trong file memdump này đang chạy những tiến trình nào bằng 1 trong câu lệnh sau đây:

- **vol.py -f <file memdump> windows.pslist:** cho phép chúng ta xem dạng list
- **vol.py -f <file memdump> windows.psscan:** cho phép chúng ta xem theo dạng lộn xộn
- **vol.py -f <file memdump> windows.pstree:** cho phép chúng ta xem theo dạng cây

Ở đây ta có thể thấy sau khi scan thì nó sẽ bắt đầu hiện ra khá là nhiều tiến trình, đây là những tiến trình hiện đang chạy trên máy, khi đang dump file memdump. Có những chú thích về cột như sau:

- ImageFileName : Tên process, như nó sẽ được hiển thị trong task manager
- PID : ID của process
- PPID: Parent process ID có nghĩa là tiến trình cha, gọi tiến trình con
- CreateTime: Thời gian tiến trình đó khởi chạy

PID	PPID	ImageFileName	Offset(V)	Threads	Handles	SessionId	Wow64	CreateTime	ExitTime	File output
4	0	System	0x83155d90	105	524	N/A	False	2023-05-29 06:48:56.000000	N/A	Disabled
416	4	smss.exe	0x840322d0	4	28	N/A	False	2023-05-29 06:48:56.000000	N/A	Disabled
492	480	csrss.exe	0x8413c750	11	423	0	False	2023-05-29 06:48:57.000000	N/A	Disabled
536	480	wininit.exe	0x840a1d90	3	180	0	False	2023-05-29 06:48:57.000000	N/A	Disabled
544	536	csrss.exe	0x8440e490	10	349	1	False	2023-05-29 06:48:57.000000	N/A	Disabled
576	528	winlogon.exe	0x84421d90	3	118	1	False	2023-05-29 06:48:57.000000	N/A	Disabled
624	536	services.exe	0x84439020	6	239	0	False	2023-05-29 06:48:58.000000	N/A	Disabled
636	536	lsass.exe	0x8444d988	10	603	0	False	2023-05-29 06:48:58.000000	N/A	Disabled
644	536	lsm.exe	0x84450d90	9	160	0	False	2023-05-29 06:48:58.000000	N/A	Disabled
796	624	svchost.exe	0x844b5d90	6	297	0	False	2023-05-29 06:48:58.000000	N/A	Disabled
860	624	svchost.exe	0x844c2808	6	273	0	False	2023-05-29 06:48:58.000000	N/A	Disabled
896	624	svchost.exe	0x844d3700	15	286	0	False	2023-05-29 06:48:58.000000	N/A	Disabled
1016	624	svchost.exe	0x844f8d90	5	148	0	False	2023-05-29 06:48:58.000000	N/A	Disabled
1032	624	svchost.exe	0x844fc998	36	914	0	False	2023-05-29 06:48:58.000000	N/A	Disabled
1048	624	Slsvc.exe	0x84502478	4	99	0	False	2023-05-29 06:48:58.000000	N/A	Disabled
1076	624	svchost.exe	0x84507188	17	596	0	False	2023-05-29 06:48:58.000000	N/A	Disabled
1164	624	svchost.exe	0x84515180	20	254	0	False	2023-05-29 06:48:58.000000	N/A	Disabled
1188	624	svchost.exe	0x845012b8	16	489	0	False	2023-05-29 06:48:58.000000	N/A	Disabled
1344	624	svchost.exe	0x8455b278	17	263	0	False	2023-05-29 06:48:59.000000	N/A	Disabled
1448	1032	taskeng.exe	0x845b0020	5	136	0	False	2023-05-29 06:49:11.000000	N/A	Disabled
1532	624	spoolsv.exe	0x8454a020	16	290	0	False	2023-05-29 06:49:14.000000	N/A	Disabled
1572	624	ftpbasicssvr.exe	0x834148d8	2	52	0	False	2023-05-29 06:49:14.000000	N/A	Disabled
1608	624	svchost.exe	0x83420d90	5	124	0	False	2023-05-29 06:49:14.000000	N/A	Disabled
1624	624	svchost.exe	0x84482d48	3	73	0	False	2023-05-29 06:49:14.000000	N/A	Disabled
1664	624	vmtoolsd.exe	0x83578020	7	277	0	False	2023-05-29 06:49:14.000000	N/A	Disabled
1752	624	svchost.exe	0x845b8d90	4	43	0	False	2023-05-29 06:49:14.000000	N/A	Disabled
1056	624	dllhst.exe	0x848bf6c0	13	240	0	False	2023-05-29 06:49:22.000000	N/A	Disabled
1280	624	msdtc.exe	0x848d8368	11	166	0	False	2023-05-29 06:49:23.000000	N/A	Disabled
2608	1032	taskeng.exe	0x8494d550	9	239	1	False	2023-05-29 06:51:22.000000	N/A	Disabled

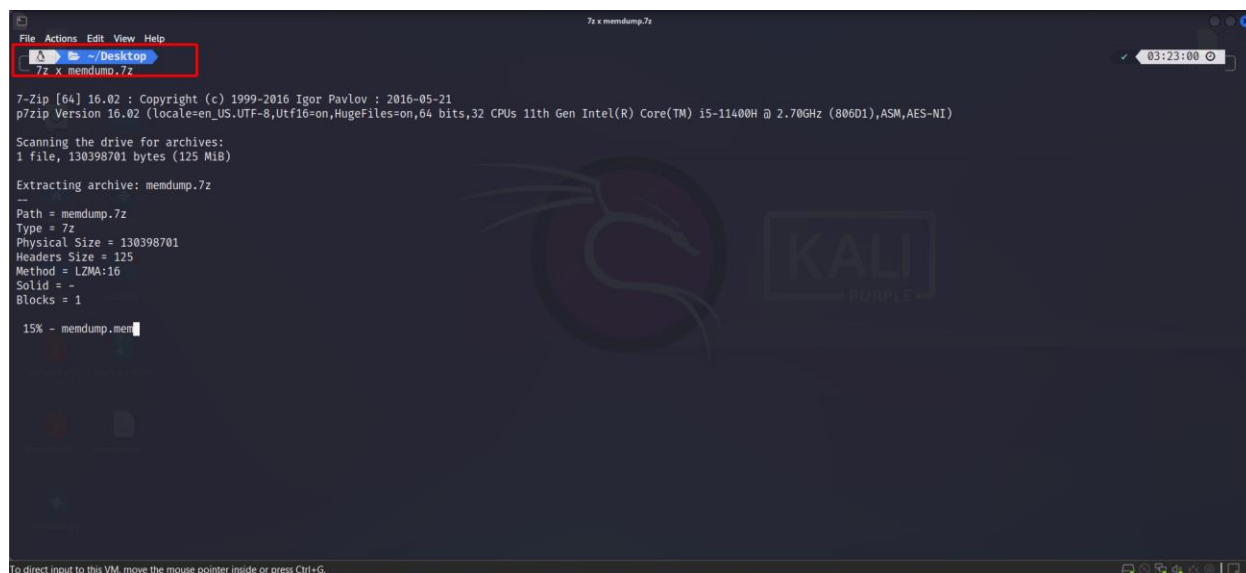
Ngoài ra ta có thể thực hiện scan network thông qua câu lệnh netscan sau đây: **vol.py -f <memdump file> windows.netscan**

Offset	Proto	LocalAddr	LocalPort	ForeignAddr	ForeignPort	State	PID	Owner	Created
0x3e405bf8	UDPv4	0.0.0.0	123	*	0	1076	svchost.exe	2023-05-30 03:00:40.000000	
0x3e405bf8	UDPv6	::	123	*	0	1076	svchost.exe	2023-05-30 03:00:40.000000	
0x3e40a840	UDPv4	0.0.0.0	5355	*	0	1188	svchost.exe	2023-05-30 03:00:40.000000	
0x3e40a840	UDPv6	::	5355	*	0	1188	svchost.exe	2023-05-30 03:00:40.000000	
0x3e412ae8	TCPv4	-	1254	224.0.0.22	443	CLOSED	964	chrome.exe	-
0x3e42fe38	UDPv4	0.0.0.0	5355	*	0	1188	svchost.exe	2023-05-30 03:00:40.000000	
0x3e43b7a0	UDPv4	192.168.15.137	138	*	0	4	System	2023-05-30 03:00:40.000000	
0x3e48e730	UDPv4	0.0.0.0	123	*	0	1076	svchost.exe	2023-05-30 03:00:40.000000	
0x3e4a2d30	TCPv4	192.168.15.137	139	0.0.0.0	0	LISTENING	4	System	N/A
0x3e4ad558	UDPv4	192.168.15.137	137	*	0	4	System	2023-05-30 03:00:40.000000	
0x3e4e3680	TCPv4	192.168.15.137	1274	216.58.208.238	443	ESTABLISHED	964	chrome.exe	-
0x3e539b00	TCPv4	-	1275	192.168.15.255	443	CLOSED	964	chrome.exe	-
0x3e539cfc	TCPv4	-	1276	192.168.15.255	443	CLOSED	964	chrome.exe	-
0x3e687a60	TCPv4	0.0.0.0	445	0.0.0.0	0	LISTENING	4	System	N/A
0x3e687a60	TCPv6	::	445	::	0	LISTENING	4	System	N/A
0x3e6973e8	UDPv4	0.0.0.0	3702	*	0	1076	svchost.exe	2023-05-30 03:00:55.000000	
0x3e6973e8	UDPv6	::	3702	*	0	1076	svchost.exe	2023-05-30 03:00:55.000000	
0x3e6aa680	UDPv4	0.0.0.0	0	*	0	1188	svchost.exe	2023-05-30 03:00:40.000000	
0x3e6aa680	UDPv6	::	0	*	0	1188	svchost.exe	2023-05-30 03:00:40.000000	
0x3e6ac488	TCPv4	0.0.0.0	1030	0.0.0.0	0	LISTENING	624	services.exe	N/A
0x3e6ac488	TCPv6	::	1030	::	0	LISTENING	624	services.exe	N/A
0x3e6ad20	TCPv4	0.0.0.0	1030	0.0.0.0	0	LISTENING	624	services.exe	N/A
0x3e6c3988	UDPv4	0.0.0.0	0	*	0	1164	svchost.exe	2023-05-29 06:49:22.000000	
0x3eab7d68	TCPv4	0.0.0.0	1028	0.0.0.0	0	LISTENING	636	lsass.exe	N/A
0x3eac79b0	TCPv4	0.0.0.0	135	0.0.0.0	0	LISTENING	860	svchost.exe	N/A
0x3eac9500	TCPv4	0.0.0.0	135	0.0.0.0	0	LISTENING	860	svchost.exe	N/A
0x3eac9500	TCPv6	::	135	::	0	LISTENING	860	svchost.exe	N/A
0x3ead3640	TCPv4	0.0.0.0	1025	0.0.0.0	0	LISTENING	536	wininit.exe	N/A
0x3ead3640	TCPv6	::	1025	::	0	LISTENING	536	wininit.exe	N/A

Extracting Password Hashes

Bây giờ chúng ta sẽ download một file memdump khác và tiến hành phân tích file memdump đó dựa trên vol3.

Vì đây là một file 7z nên chúng ta sẽ phải dùng command **7z x <file name>** để có thể giải nén chúng ra



Sau đó ta sẽ bắt đầu đi phân tích registry hive của file memdump thông qua câu lệnh

- `vol.py -f "/path/to/file" windows.registry.hivescan`: Chỉ hiện ra offset
- `vol.py -f "/path/to/file" windows.registry.hivelist`: Hiện hết tất cả những gì cần thấy'

Tại đây chúng ta phải lưu ý tới hai chỗ là SYSTEM và SAM vì hai chỗ này chứa những thứ cần thiết để chúng ta có thể lấy được mã hash password. Ngoài ra còn phải chú ý tới offset của chúng, vì chúng ta sẽ rất cần nó trong việc dump file hoặc process ra.

```
File Actions Edit View Help
0x9ded6a8
0x9fa5008
0x10e59008
0x12e8e008
0x155be008
0x158b46b0
0x1973520
0x1b2c008
0x1b2eca20
0x1b33f450
0x1b40b008
0x1b4b1148
0x1c8e6008
0x1de39148
~/Desktop/volatility3 P develop
python3 vol.py -f ~/Desktop/mendump.mem windows.registry.hivelist
Volatility 3 Framework 2.4.2
Progress: 100.00 PDB scanning finished
Offset FileFullPath File output
0x812eb6b0 \Device\HarddiskVolume1\Windows\ServiceProfiles\NetworkService\NTUSER.DAT Disabled
0x81321008 \Device\HarddiskVolume1\Windows\ServiceProfiles\LocalService\NTUSER.DAT Disabled
0x86211008 Disabled
0x86226008 \REGISTRY\MACHINE\SYSTEM Disabled
0x86248008 \REGISTRY\MACHINE\HARDWARE Disabled
0x89c7f148 \Device\HarddiskVolume1\Windows\System32\config\DEFAULT.DAT Disabled
0x89c30450 \Device\HarddiskVolume1\Windows\System32\config\SMM Disabled
0x89c36008 \Device\HarddiskVolume1\Windows\System32\config\SECURITY Disabled
0x89c47008 \Device\HarddiskVolume1\Windows\System32\config\COMPONENTS Disabled
0x89c47a20 \Device\HarddiskVolume1\Windows\System32\config\SOFTWARE Disabled
0x89cd1a20 \Device\HarddiskVolume1\Boot\BCD Disabled
0x9465f6a8 \Device\HarddiskVolume1\Users\Administrator\NTUSER.DAT Disabled
0x946ae008 \Device\HarddiskVolume1\Users\Administrator\AppData\Local\Microsoft\Windows\UsrClass.dat Disabled
~/Desktop/volatility3 P develop
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
```

Để scan xem là bên trong bộ password chứa những gì, thì ta sẽ sử dụng câu lệnh **python3 vol.py -f <filename> windows.hashdump.Hashdump**

```
File Actions Edit View Help
~/Desktop/volatility3 P develop
python3 vol.py -f ~/Desktop/mendump.mem windows.hashdump.Hashdump
Volatility 3 Framework 2.4.2
Progress: 100.00 PDB scanning finished
User rid lmhash nthash
Administrator 500 aad3b435b51404eeaad3b435b51404ee e19ccf75ee54e06b06a5907af13cef42
Guest 501 aad3b435b51404eeaad3b435b51404ee 31ddcf0dd19ae931b73c59d7e0c089c0
student 1000 aad3b435b51404eeaad3b435b51404ee e19ccf75ee54e06b06a5907af13cef42
probe 1002 aad3b435b51404eeaad3b435b51404ee e19ccf75ee54e06b06a5907af13cef42
waldo 1004 aad3b435b51404eeaad3b435b51404ee cfeac129dc5e61b7eb9b7e7131fc7e2b
YOUR-NAME 1005 aad3b435b51404eeaad3b435b51404ee 958c8526e4252b277d8d70aadb2ea2ce
~/Desktop/volatility3 P develop
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
```

Ở đây để dưới dạng mã hash. Nếu chúng ta đem chúng vào trogn crackstation.net – một trang web chuyên có những mã hash để có thể decrypt về dạng ban đầu. Tuy là hash là không thể nào trở lại ban đầu được nhưng mà web này sẽ sử dụng những cái đã có check xem là mã hash này có nằm trong những mã hash hay được sử dụng hay không

CrackStation Password Hashing Security Defuse Security

CrackStation

Defuse.ca · Twitter

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

958c8526e4252b277d8d70adb2ea2ce

I'm not a robot

reCAPTCHA

Privacy · Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, rpeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
958c8526e4252b277d8d70adb2ea2ce	NTLM	SuperSecret!

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

[Download CrackStation's Wordlist](#)

How CrackStation Works

CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash. If the hash is present in the database, the password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our [hashing security page](#).

Crackstation's lookup tables were created by extracting every word from the Wikipedia databases and adding with every password list we could

Thử với mã của tài khoản YOUR-NAME thì ta biết được rằng, password đó chính là SuperSecret! Đúng với nhuwxg gì mà chúng ta đã đặt trước khi lấy ra file memdump.

Sau đó chúng ta sẽ extract những gì mà bên trong console, cụ thể ở đây là cmd xem llucs mà dump memory thì trong cmd có nhwunxg gì thông qua câu lệnh: **vol.py -f <filename>windows.cmdline**

Không biết gì lý do gì, windows.cmdline bên vol3 chỉ hiện được tên process, nên chúng ta sẽ dung lại vol2 để có thể scan lại ./volatility2 consoles --profile=<tên profile> -f <filename>

```
Windows PowerShell
-a---- 12/27/2016 11:02 PM 15794079 volatility_2.6_win64_standalone.exe

PS C:\Users\Admin\Desktop\volatility_2.6_win64_standalone> .\volatility_2.6_win64_standalone.exe consoles --profile=Win2
008SP1x86 -f memdump.mem
Volatility Foundation Volatility Framework 2.6
*****
ConsoleProcess: csrss.exe Pid: 472
Console: 0x14b4944 CommandHistorySize: 50
HistoryBufferCount: 1 HistoryBufferMax: 4
OriginalTitle: taskeng.exe
Title: -
*****
ConsoleProcess: csrss.exe Pid: 472
Console: 0x14b65ec CommandHistorySize: 50
HistoryBufferCount: 1 HistoryBufferMax: 4
OriginalTitle: C:\Windows\system32\schtasks.exe
Title: -
*****
ConsoleProcess: csrss.exe Pid: 516
Console: 0x28389c CommandHistorySize: 50
HistoryBufferCount: 1 HistoryBufferMax: 4
OriginalTitle: C:\Program Files\VMware\VMware Tools\TPAutoConnSvc.exe
Title: -
AttachedProcess: TPAutoConnect.e Pid: 2480 Handle: 0x214
-----
CommandHistory: 0x284508 Application: TPAutoConnect.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x214
-----
Screen 0x284040 X:80 Y:25
Dump:
ThinPrint AutoConnect component, Copyright (c) 1999-2012 Cortado AG, 8.8.734.1
*****
ConsoleProcess: csrss.exe Pid: 516
Console: 0x671427c CommandHistorySize: 50
HistoryBufferCount: 3 HistoryBufferMax: 4
OriginalTitle: Command Prompt
```



```
Windows PowerShell
??(osoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>net user waldo qwerty
The user name could not be found.

More help is available by typing NET HELPMSG 2221.

C:\Users\Administrator>net user /?
The syntax of this command is:

NET USER
[username [password | *] [options]] [/DOMAIN]
username {password | *} /ADD [options] [/DOMAIN]
username [/DELETE] [/DOMAIN]
username [/TIMES:{times | ALL}]

C:\Users\Administrator>net user waldo qwerty /add
The password does not meet the password policy requirements. Check the minimum password length, password complexity and password history requirements.

More help is available by typing NET HELPMSG 2245.

C:\Users\Administrator>net user YOUR-NAME letmein /add

C:\Users\Administrator>net user waldo Apple123 /add
The command completed successfully.

C:\Users\Administrator>net user YOUR-NAME SuperSecret! /add
The command completed successfully.

C:\Users\Administrator>
C:\Users\Administrator>exit
PS C:\Users\Admin\Desktop\volatility_2.6_win64_standalone>
```

Kéo xuống dưới ta thấy được rằng trong cmd, user đã hỏi xem cách thêm người dung và password như thế nào, sau đó đã thêm 2 user với 2 password ban đầu và sau đó chuyển password của 2 user đó thành 2 từ khác đó chính là Apple123 và SuperSecret!