1

Session

Memory Forensics

Điều tra bộ nhớ

Tài liệu Thực hành

Pháp chứng Kỹ thuật số

GVTH: ThS. Phan Thế Duy

Học kỳ II – Năm học 2018-2019

Tp. HCM, 3.2019

Lưu hành nội bộ

A. TỔNG QUAN

1. Điều tra pháp chứng kỹ thuật số

Trong lĩnh vực an toàn thông tin, Forensics hay còn gọi là điều tra số là công việc phát hiện, bảo vệ và phân tích thông tin được lưu trữ, truyền tải hoặc được tạo ra bởi một máy tính hoặc mạng máy tính, nhằm đưa ra các suy luận hợp lý để tìm nguyên nhân, giải thích các hiện tượng trong quá trình điều tra.

Mục tiêu của Forensic?

Mục tiêu cốt lõi của việc điều tra là phát hiện, bảo quản, khai thác và đưa ra kết luận về dữ liệu thu thập được. Cần lưu ý rằng dữ liệu phải đảm bảo tính xác thực, và được lấy mà không bị hư hại, nếu không dữ liệu đấy sẽ không còn ý nghĩa.

2. Điều tra bộ nhớ

Điều tra bộ nhớ (Memory Forensic) là gì?

Điều tra bộ nhớ là một hình thức điều tra phân tích quan trọng trong kỹ thuật điều tra số cho phép các nhà điều tra xác định các hành vi bất thường, không được phép (unauthorized) trên các máy tính hay máy chủ mục tiêu. Để thực hiện điều này, những công cụ, phần mềm sẽ được dùng để "chụp" lại snapshot tình trạng hiện tại của bộ nhớ hệ thống (còn gọi là memory dump). Sau đó, tập tin "chụp" được sẽ được lưu trữ, xử lý, tìm kiếm và phân tích bởi nhân viên điều tra.

Nhắc đến điều tra dữ liệu trên bộ nhớ, là nhắc đến kỹ thuật phân tích dữ liệu lưu trên RAM. Đây là kỹ thuật điều tra máy tính bằng việc ghi lại bộ nhớ RAM của hệ thống thời điểm có dấu hiệu nghi ngờ, hoặc đang bị tấn công để tiến hành điều tra, giúp cho việc xác định nguyên nhân cũng như các hành vi đã xảy ra trên hệ thống, cung cấp các chứng cứ phục vụ cho việc xử lý tội phạm. Kỹ thuật điều tra này được sử dụng khi quá trình phân tích tĩnh từ những gói tin thu được, cũng như các thông tin từ nhật ký hệ thống ghi lại, nhưng chưa xác định được nguồn gốc cũng như kỹ thuật tấn công, hoặc cung cấp các thông tin có được chưa đầy đủ, chưa đủ sức thuyết phục.

Như chúng ta biết, phân tích điều tra bộ nhớ RAM cũng giống như tất cả những nỗ lực điều tra số khác, nó liên quan đến việc thu thập thông tin, để có thể cung cấp các chứng cứ như bằng chứng sử dụng trong điều tra hình sự. Nhưng cụ thể hơn thì đây là kỹ thuật mà người điều tra cố gắng sử dụng kiến trúc quản lý bộ nhớ trong máy tính để ánh xạ, trích xuất các tập tin đang thực thi và cư trú trong bộ nhớ vật lý của máy tính. Những tập tin thực thi có thể được sử dụng để chứng minh rằng hành vi của tội phạm đã xảy ra hoặc để theo dõi nó đã diễn ra như thế nào. Tính hữu ích của loại hình điều tra này đó là trong thực tế, bất kỳ thông tin nào tìm thấy trong bộ nhớ RAM, sẽ được hiểu là gần đây nó đã được chạy trên hệ thống của nạn nhân.

Tại sao lại phải phân tích điều tra bộ nhớ? Memory dump chứa những gì?

Vì tất cả mọi thứ trước khi nạp vào hệ điều hành đều đi qua RAM, thông thường memory dump là bản sao đầy đủ của toàn bộ bộ nhớ của máy tính. Do đó dung lượng của memory dump sẽ là dung lượng bộ nhớ của máy đang sử dụng. Một memory dump đầy đủ sẽ chứa toàn bộ những dữ liệu được hệ điều hành nạp vào và sử dụng bao gồm các process đang chạy, các file đã được mở, các dữ liệu đã được thực thi, tính toán trên bộ nhớ và chưa bị xóa hoặc ghi đè bởi dữ liệu khác và cả các vùng nhớ còn trống chưa được sử dụng. Cụ thể:

- ***Các tiến trình đang chạy trên hệ thống***: Tất cả các tiến trình đang chạy được lưu trữ ở bộ nhớ và có thể được lấy ra bằng việc sử dụng các công cụ dựa vào cấu trúc của hệ thống đang điều tra. Từ khi được đưa vào hệ thống cho đến khi kết thúc tiến trình tồn tại ở các trạng thái khác nhau. Trong hệ điều hành có nhiều tiến trình khác nhau, tiến trình này có thể là tiến trình con, hay tiến trình cha của tiến trình kia, tất cả những tiến trình đó có thể được tìm thấy trong bộ nhớ RAM. Các tiến trình ẩn cũng có thể được trích xuất ra khỏi bộ nhớ đã được ghi lại. Khi các tiến trình đã kết thúc nó vẫn có thể được cư trú trong bộ nhớ bởi vì không gian cư trú vẫn chưa được phân bổ lại.
- ***Các tập tin đang mở và Registry Handles:*** Các tập tin đang mở cũng như bất kỳ một xử lý registry (registry handles) nào được truy xuất bởi một tiến trình đều được lưu trữ trong bộ nhớ. Thông tin về các tập tin đang mở hay các xử lý liên quan đến Registry rất có giá trị trong việc điều tra. Chúng ta có thể hiểu như thế này, giả sử có một tiến trình của phần mềm độc hại đang chạy trên hệ thống thì các tập tin đang mở có thể giúp người điều tra khám phá nơi mà mã độc hại đang được lưu trữ trên đĩa. Trong trường hợp phân tích điều tra bộ nhớ RAM thì với việc phân tích Registry chúng ta sẽ thực hiện việc tạo dựng lại dữ liệu registry, ví dụ như hiển thị các giá trị chứa trong trong Registry của **winlogon**, sử dụng lệnh *printkey* của **volatility**.
- ***Các kết nối mạng đang có trong hệ thống:*** Thông tin về các kết nối mạng bao gồm các cổng đang lắng nghe trên hệ thống, các kết nối đang được thiết lập và các thông tin liên kết giữa hệ thống với các kết nối từ xa, những thông tin này đều có thể được lấy ra từ bộ nhớ. Các thông tin về các kết nối mạng có thể cho ta biết các Backdoor đang kết nối trong hệ thống, các máy chủ điều khiển botnet… dựa vào các kết nối của nó. Thông tin về kết nối mạng là một trong những yếu tố rất quan trọng và đáng tin cậy khi điều tra một hệ thống bị thỏa hiệp.
- ***Mật khẩu và các khóa mật mã:*** Một trong những lợi thế quan trọng của phân tích điều tra bộ nhớ là việc phục hồi lại mật khẩu người dùng và các khóa mật mã có thể được dùng để giải mã các tập tin, chương trình mà người dùng sử dụng, truy

cập (firefox, yahoo…). Mật khẩu và khóa mật mã có quy luật chung là không bao giờ được lưu trên đĩa cứng mà không có bất kỳ sự bảo vệ nào khi sử dụng chúng. Tuy nhiên chúng lại được lưu trữ trong bộ nhớ RAM và một khi điều này xảy ra thì việc điều tra bằng việc ghi lại bộ nhớ RAM có thể giúp phục hồi dữ liệu, khôi phục mật khẩu và có thể truy cập vào tài khoản trực tuyến thuộc sở hữu của kẻ đang bị điều tra như email và dữ liệu lưu trữ.

- ***Các tập tin bị xóa:*** Dữ liệu ẩn thông thường chúng ta hay nghĩ đến các thực thể độc hại trên hệ thống, hay các dữ liệu mà chủ hệ thống muốn bảo vệ để lưu trữ dữ liệu trong bộ nhớ RAM thay vì trong đĩa cứng. Ngoài việc các tập tin được ẩn trong bộ nhớ, kẻ tấn công cũng có thể chạy mã độc hại từ bộ nhớ thay vì lưu trữ mã độc hại trên đĩa cứng. Việc này gây khó khăn cho việc dịch ngược mã, khó khăn khi thực hiện việc lấy mẫu các bản sao của chương trình độc hại và tìm ra cách thức nó làm việc như thế nào, để giảm thiểu được mối đe dọa do chúng gây ra, chính vì thế việc kiểm tra các tập tin ẩn chứa trong bộ nhớ RAM có thể đưa ra những phát hiện hay những thông tin rất quan trọng cho quá trình điều tra. Dữ liệu ẩn thông thường chúng ta hay nghĩ đến các thực thể độc hại trên hệ thống, hay các dữ liệu mà chủ hệ thống muốn bảo vệ để lưu trữ dữ liệu trong bộ nhớ RAM thay vì trong đĩa cứng. Ngoài việc các tập tin được ẩn trong bộ nhớ, kẻ tấn công cũng có thể chạy mã độc hại từ bộ nhớ thay vì lưu trữ mã độc hại trên đĩa cứng. Việc này gây khó khăn cho việc dịch ngược mã, khó khăn khi thực hiện việc lấy mẫu các bản sao của chương trình độc hại và tìm ra cách thức nó làm việc như thế nào, để giảm thiểu được mối đe dọa do chúng gây ra, chính vì thế việc kiểm tra các tập tin ẩn chứa trong bộ nhớ RAM có thể đưa ra những phát hiện hay những thông tin rất quan trọng cho quá trình điều tra.
- ***Mã độc hại và các tập tin bị lây nhiễm:*** Trong những năm gần đây việc tấn công sử dụng mã độc ngày càng phổ biến và gia tăng, kẻ tấn công có thể chạy mã khai thác từ bộ nhớ thay vì mã độc sẽ thực hiện việc lưu trữ chính nó trong ổ đĩa. Điều này mục đích chính là để tránh sự phát hiện, qua mặt các chương trình diệt virus. Việc kẻ tấn công thực hiện việc lưu trữ mã độc trong bộ nhớ sẽ làm cho những người trực tiếp phân tích gặp khó khăn trong việc phục hồi và dịch ngược mã của chúng.

  Các tập tin khi được xử lý sẽ được lưu lại trong bộ nhớ, nhưng nó không được liên kết lâu hơn thông qua danh sách duy trì bởi hệ điều hành, vì hệ điều hành phải tìm ra phương pháp sử dụng thay thế, quá trình liệt kê các tập tin và phục hồi lại các tập tin đó giống như xây dựng lại các tập tin trên đĩa cứng đã bị xóa. Đây có

thể là những bằng chứng mà kẻ tấn công muốn hủy đi để gây khó khăn cho việc điều tra khi tìm kiếm chứng cứ.

Làm thế nào để có được bộ nhớ RAM?

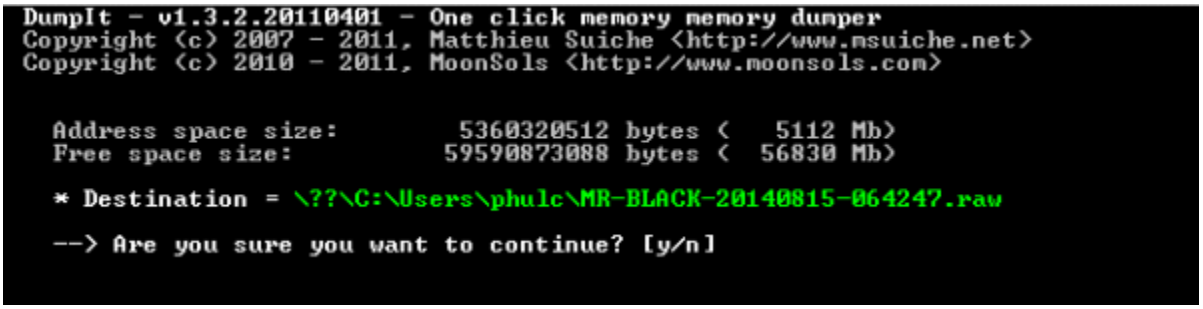
Để có được bộ nhớ RAM và phân tích nó thì đầu tiên chuyên viên phân tích phải biết sử dụng kỹ thuật để thu lại (Acquisition).

Có 2 phương pháp để thu lại bộ nhớ RAM: Thu lại dựa trên phần cứng và thu lại dựa trên phần mềm, cả hai phương pháp đều có những ưu và nhược điểm riêng, sẽ được mô tả chi tiết ở phần sau. Nhìn chung, theo quan điểm điều tra pháp lý trong điều tra số thì việc thu lại dựa trên phần cứng thường được đề cao hơn vì nó đáng tin cậy và thường khó khăn cho kẻ tấn công khi chúng cố ý làm sai lạc chứng cứ, tuy nhiên hiện tại thì kỹ thuật thu lại bộ nhớ RAM dựa vào phần mềm thường được sử dụng vì nó có chi phí phù hợp, dễ tìm kiếm.

Thu lại bộ nhớ RAM dựa trên phần cứng liên quan đến việc tạm ngưng quá trình xử lý của máy tính và thực hiện truy cập bộ nhớ trực tiếp để có được một bản sao của bộ nhớ, nó được coi là tin cậy hơn vì ngay cả khi hệ điều hành và phần mềm trên hệ thống đã bị xâm nhập hoặc bị làm sai bởi kẻ tấn công, chúng ta vẫn có thể nhận được một hình ảnh chính xác của bộ nhớ, bởi vì chúng ta không phụ thuộc vào các thành phần của hệ thống. Nhưng nhược điểm của phương pháp này là chi phí đắt đỏ. Một trong phần cứng chuyên dụng thiết kế cho mục đích này là Tribble Card, nó là một tấm mạch PCI được cài đặt sẵn trong hệ thống trước khi một thỏa hiệp được diễn ra, vì vậy cho phép các nhà điều tra ghi lại bộ nhớ một cách chính xác và đáng tin cậy nhất, hơn hẳn so với phương pháp thu hồi dựa trên phần mềm.

Thu lại bộ nhớ RAM dựa trên phần mềm hiện nay là kỹ thuật thường được sử dụng phổ biến bằng việc sử dụng bộ công cụ đáng tin cậy được phát triển và cung cấp bởi các chuyên gia điều tra số hàng đầu trên thế giới như **Memoryze, dumpit, win32dd, Mandiant Redline**,... Hoặc cũng có thể sử dụng những công cụ đã được tích hợp sẵn trong hệ thống để tiến hành thu lại bộ nhớ RAM như **dd, memdump** trong linux.

Hình 1 mô tả việc sử dụng công cụ DumpIt để ghi lại bộ nhớ RAM trong môi trường windows.



Hình 1. Cách thu lại bộ nhớ bằng công cụ DumpIt

Một số công cụ thường dùng khi thực hiện điều tra bộ nhớ RAM

- **Volatility** là một framework với rất nhiều plugins được dùng để phân tích và tìm kiếm thông tin từ chứng cứ thu được. Các plugins được viết để phân tích điều tra bộ nhớ theo kiến trúc của hệ điều hành windows.

  Việc phân tích sẽ tiến hành trong một môi trường độc lập với hệ thống ghi lại chứng cứ, volatility cài đặt và làm việc được cả trên hệ điều hành linux cũng như windows, trong lĩnh vực Memory Forensics thì đây là công cụ không thể thiếu để điều tra và phân tích, nó có thể thực hiện được các công việc như xác định các tiến trình đang chạy trên hệ thống đó, các registry handles, các sockets đang mở trên mạng, danh sách dlls được nạp vào mỗi tiến trình, các thư viện, hàm, API hooks trong người dùng và nhân (kernel)…

- **Mandiant Redline:** Mandiant Readline là công cụ miễn phí hàng đầu của Mandiant cung cấp khả năng điều tra máy chủ để người dùng tìm thấy dữ liệu hoạt động của các tập tin độc hại thông qua việc việc phân tích bộ nhớ và tập tin hệ thống.

  Công cụ này cung cấp các đánh giá sơ lược về các mối đe dọa trong hệ thống được phân tích. Lọc các dữ liệu thông qua khung thời gian, xác định các tiến trình đang hooks trong hệ thống, thực hiện việc phân tích các chỉ số thỏa hiệp, thu thập tất cả các tiến trình đang chạy, kiểm tra dữ liệu từ bộ nhớ,…

- **SANS Investigate Forensic Toolkit (SIFT) Workstation:** SANS SIFT là một môi trường điều tra số tuyệt vời, nó được tạo ra bởi các chuyên gia forensics hàng đầu trên thế giới, nó tập trung và chọn lọc lại các công cụ điều tra và tích hợp sẵn trong một hệ thống, nó chứng minh rằng điều tra chuyên sâu và ứng phó với sự cố xâm nhập có thể được thực hiện bằng việc sử dụng các công cụ mã nguồn mở và cập nhật thường xuyên, bộ công cụ chọn lọc này tích hợp những công cụ nổi tiếng như Thesleuth Kit, DFF, wireshark, Foremost, volatility, Hexeditor…

3. Mục tiêu

Bài thực hành này giúp sinh viên được làm quen, sử dụng, tăng cường kiến thức về các kỹ năng điều tra kỹ thuật số liên quan đến việc phân tích bộ nhớ.

4. Môi trường & cấu hình

- Sử dụng các thiết bị và tài liệu, khuyến cáo được cung cấp bởi GVTH, yêu cầu tác phong nghiêm túc trong quá trình thực hiện.
- Công cụ gợi ý: **Volatility**
- Tài liệu nên đọc: Sách ***"The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory"*** (tác giả: Michael Hale Ligh, Andrew Case, Jamie Levy, AAron Walters)

B. THỰC HÀNH

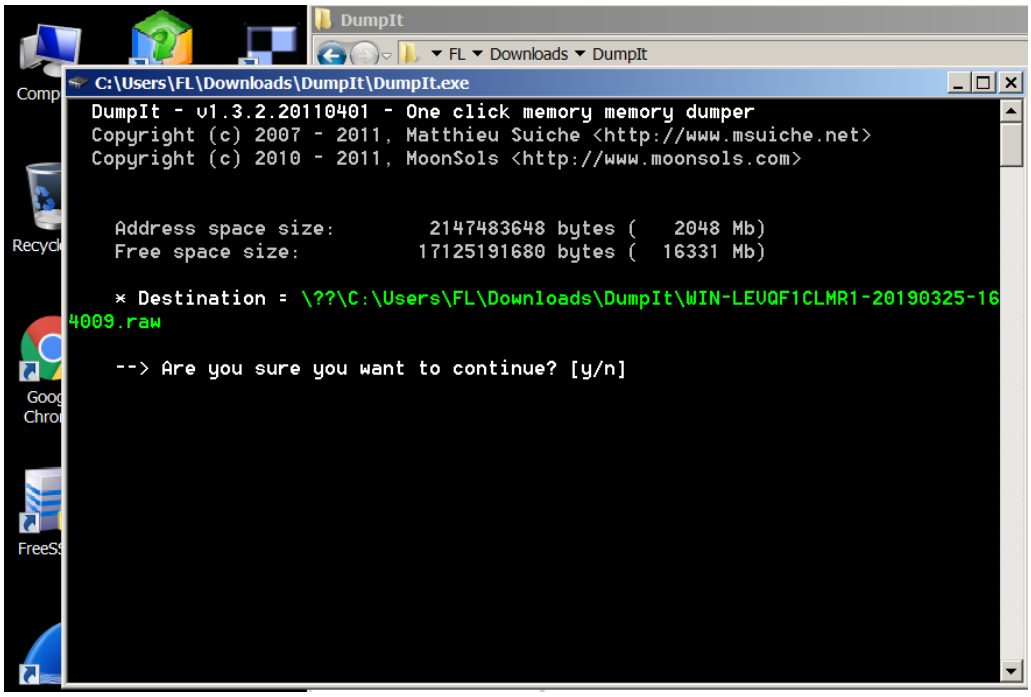
Sinh viên thực hiện điều tra theo yêu cầu của GVHD, làm theo nhóm thực hành đã đăng ký trên lớp trong buổi thực hành.

B1. Thu lại bộ nhớ (Memory Acquisition)

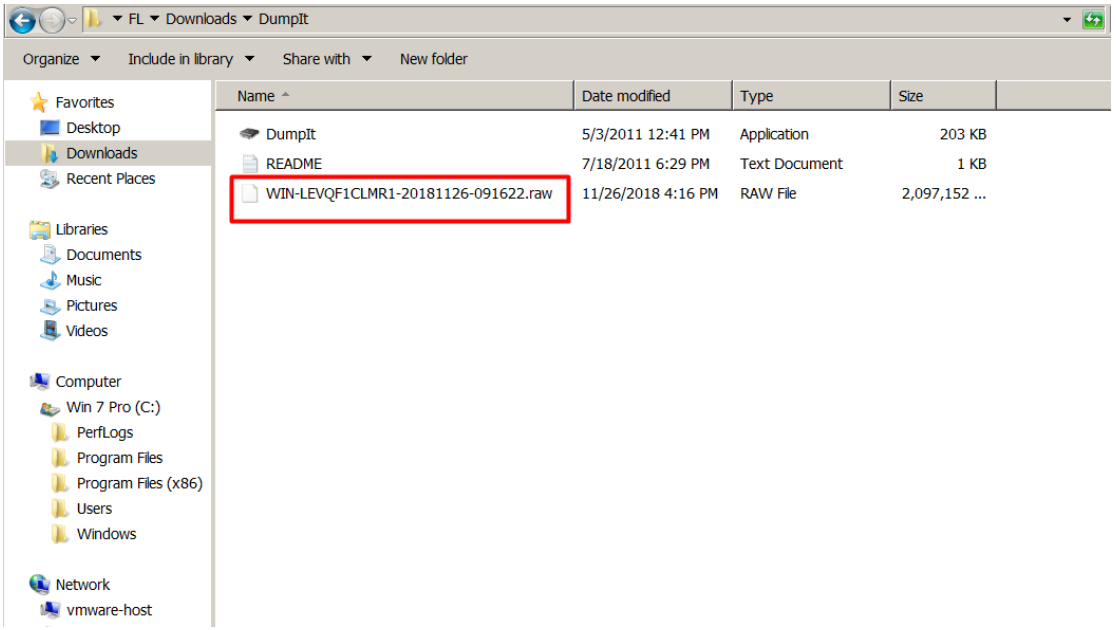
Sử dụng công cụ DumpIt (thu RAM)

- Download công cụ DumpIt

- Chạy chương trình và quan sát:



Tập tin thu được như sau:



B2. Phân tích bộ nhớ với công cụ Volatility

Volatility là công cụ dùng để phân tích bộ nhớ RAM ở các hệ điều hành 32/64 bit. Nó hỗ trợ các hệ điều hành như Linux, Windows, Mac, and Android. Nó có thể phân tích nhiều loại bộ nhớ khác nhau: raw dumps, crash dumps, VMware dumps (.vmem), virtual box dumps…

a. Kịch bản 1

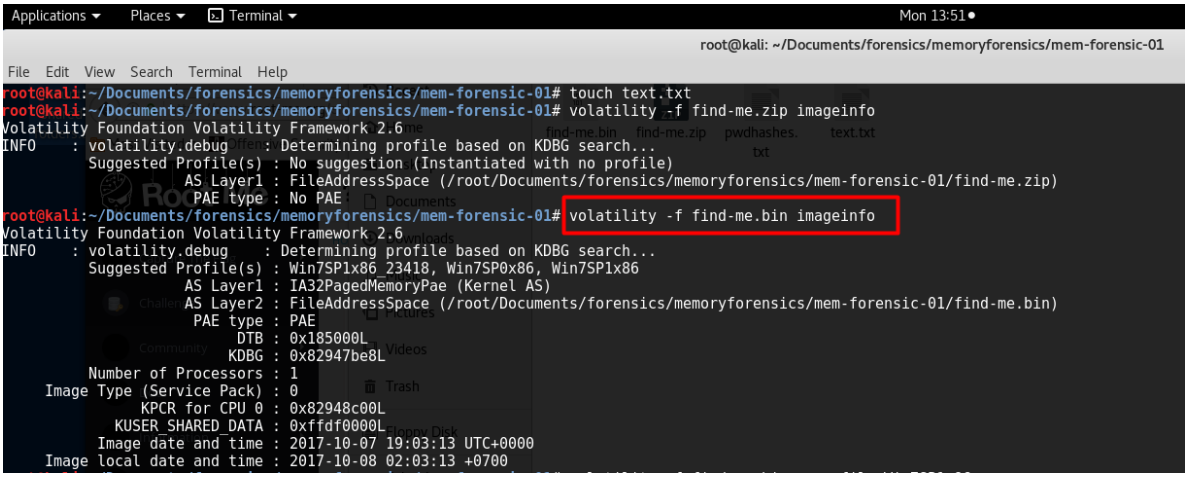
- Tài nguyên: find-me.bin
- Kiểm tra thông tin của file dump:

  volatility -f <tập_tin_ram_dump.raw> imageinfo

  hoặc

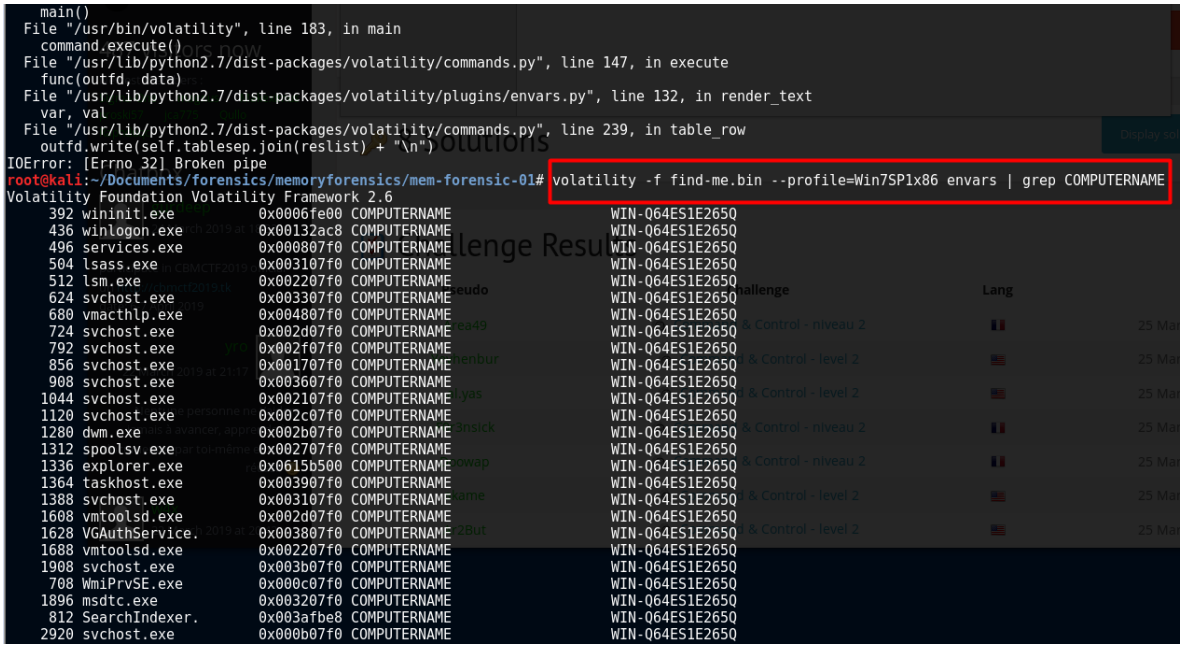
  volatility imageinfo -f <tập_tin_ram_dump.raw>

  volatility -f <tập_tin_ram_dump.raw> kdbgscan



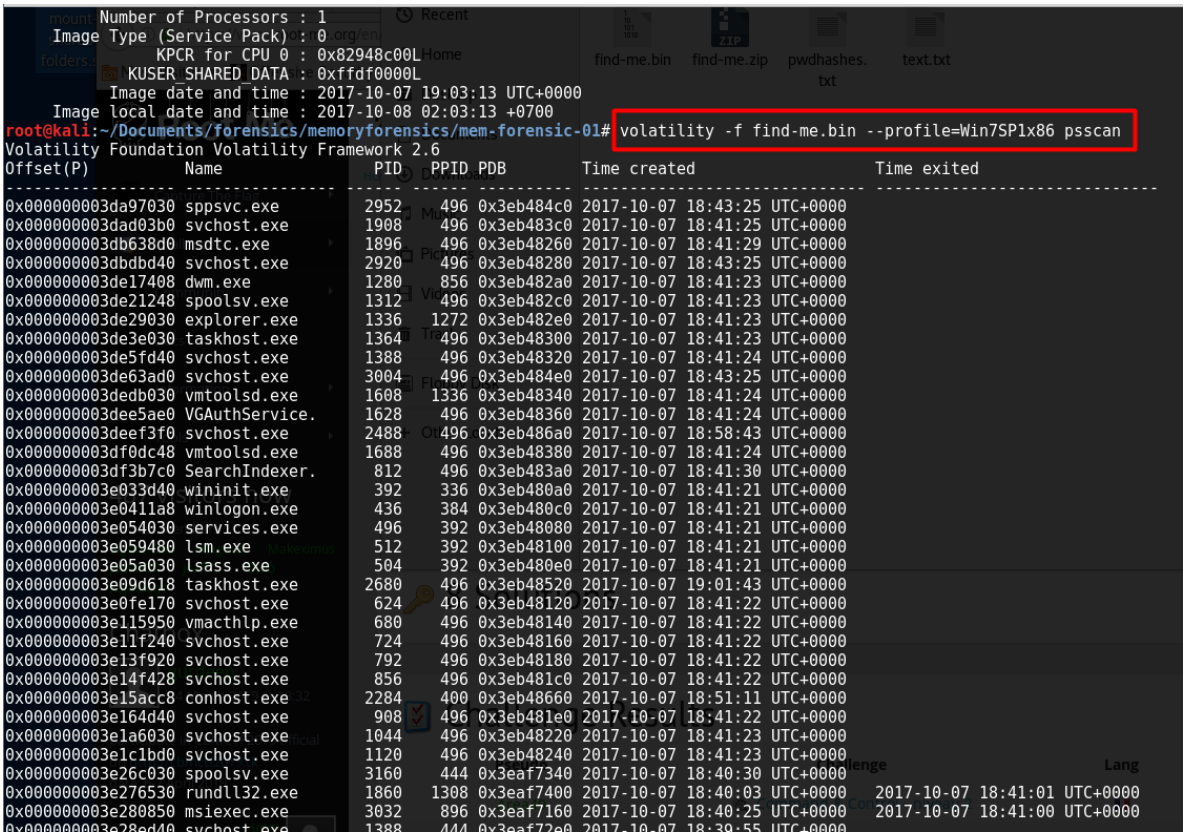
- Dựa vào kết quả đưa ra, ta có thể xác định profile của hệ thống đã được dump là Win7SP0x86 hoặc Win7SP1x86.
- Windows có rất nhiều biến môi trường để các process khi chạy có thể truy xuất dữ liệu tham chiếu như OS, TEMP, windir, Path… và tên máy đang sử dụng sẽ được lưu trong biến có tên COMPUTERNAME. Ta có thể xem biến môi trường COMPUTERNAME bằng cách gõ:

  volatility -f find-me.bin —profile=Win7SP1x86 envars | grep COMPUTERNAME



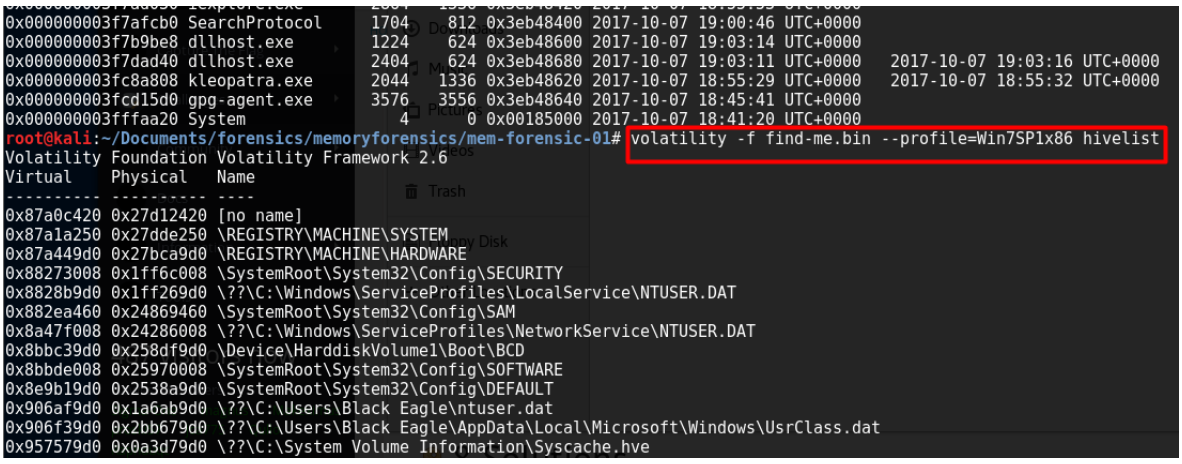
- Ngoài ra, sau khi xác định được hệ điều hành là Win7SP1x86, ta có thể thực hiện kiểm tra các process đang chạy, gõ lệnh:

  volatility -f find-me.bin --profile=Win7SP1x86 psscan



- Chúng ta cũng có thể lấy **hivelist**. Hiểu đơn giản thì đây là công đoạn 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.

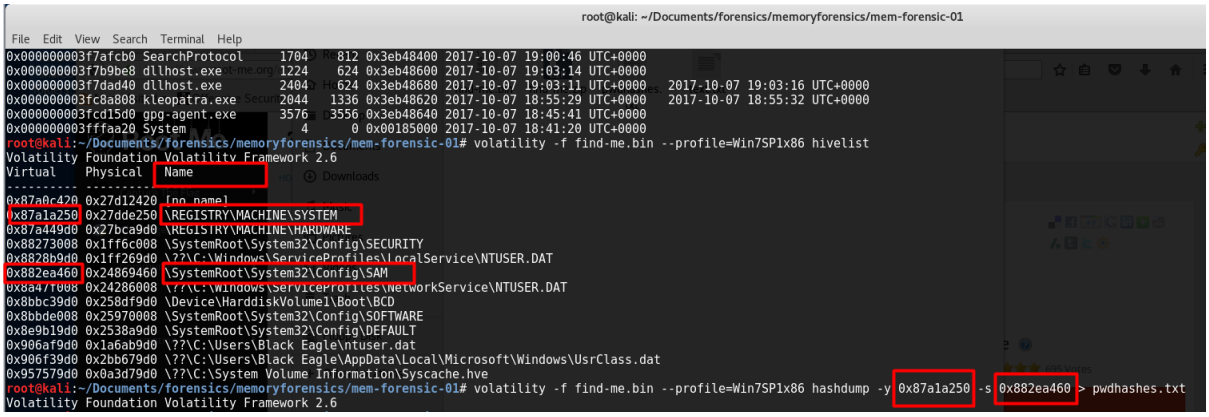
  volatility -f <tập_tin_ram_dump.raw> –profile=Win7SP1x64 hivelist



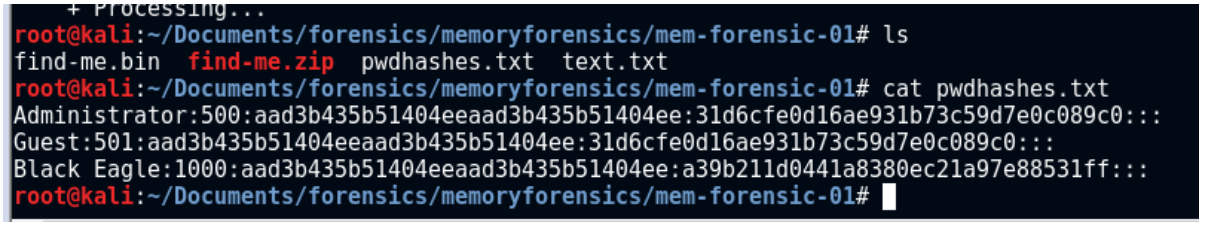
- Trong kết quả hiện ra, ta có được danh sách các key về user trên Windows đang được lưu trữ trên RAM. Công đoạn tiếp theo chỉ là tìm ra mã băm của mật khẩu. dựa vào giá trị key của hệ thống [system key] và giá trị key của tập tin SAM [SAM key].

- Trong các record trả về, ta quan sát có tổng cộng 3 cột: **Virtual | Physical | Name**. Nhìn ở cột Name ta thấy được system key là dòng **\REGISTRY\MACHINE\SYSTEM** và SAM key là **\SystemRoot\System32\Config\SAM**.
- Lấy ra giá trị Virtual tương ứng của 2 bản ghi trên và bỏ vào câu lệnh bên dưới. Đồng thời, ta sẽ trích xuất mã băm mật khẩu vào một tập tin text để tiện quan sát.

volatility -f <tập_tin_ram_dump.raw> –profile=Win7SP1x64 hashdump -y <Virtual_value_of_System_Key> -s <Virtual_value_of_SAM_Key> > pwdhashes.txt



- Và như vậy, Windows có bao nhiêu tài khoản, chương trình sẽ liệt kê toàn bộ ra, kể cả những tài khoản đang bị disable.



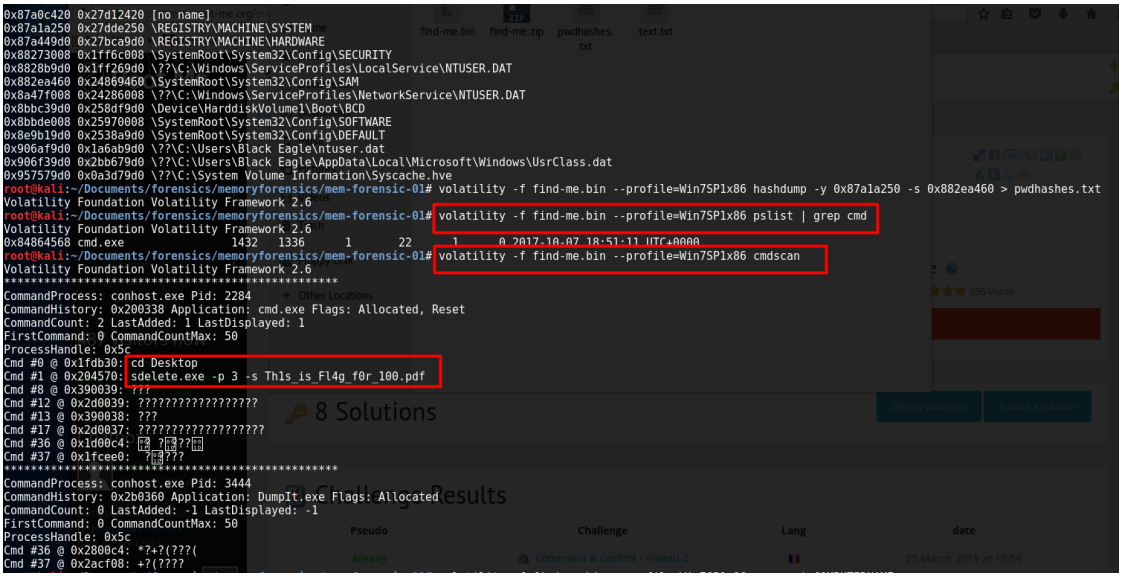
- Mỗi record lấy ra sẽ có các trường cụ thể ngăn cách nhau bởi dấu ":". Ý nghĩa các trường này là:

<Username>:<User ID>:<LM hash>:<NT hash>:<Comment>:<Home Dir>:

(Tham khảo thêm: https://www.yg.ht/blog/blog/archives/339/what-is-aad3b435b51404eeaad3b435b51404ee

http://www.adshotgyan.com/2012/02/lm-hash-and-nt-hash.html)

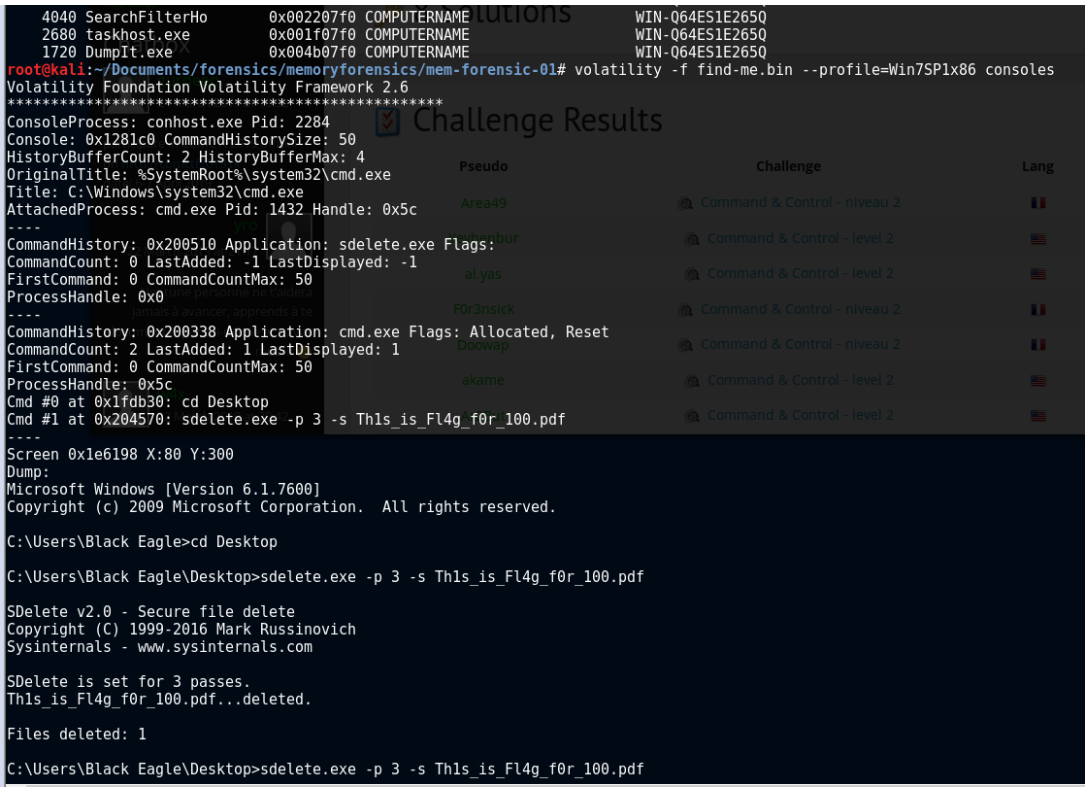
- Xem lịch sử tiến trình cmd trên máy đối tượng. Volatility có 2 plugin dùng để xem thông tin lịch sử của các lệnh đã được gõ vào cmd là ***cmdscan*** và ***consoles***. Thử với lệnh ***cmdscan***, gõ lệnh như hình:



- Thử xem lịch sử tiến trình cmd với plugin ***consoles***:

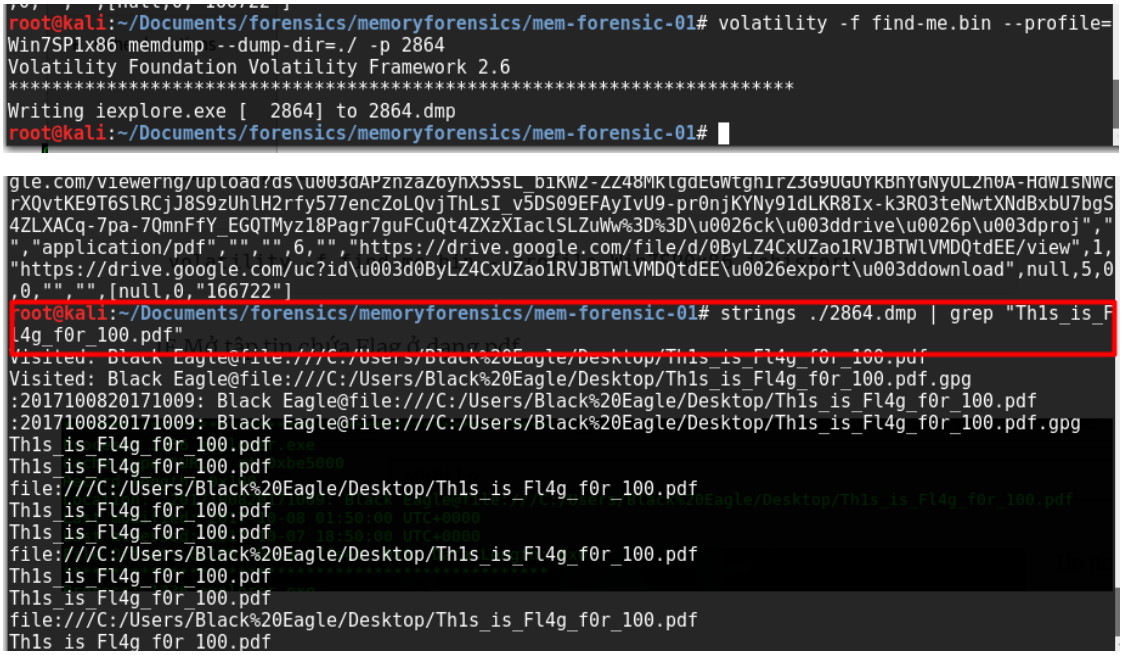
  volatility -f find-me.bin—profile=Win7SP0x86 consoles.

  Quan sát kết quả:



  Dùng plugin consoles cho ta thấy được nhiều thông tin hơn cmdscan.

- Thực hiện dump riêng tiến trình có PID là 2864 ra thành file riêng, gõ lệnh:



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. Thử nghiệm lấy thông tin mật khẩu từ đó.
- 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.
- Xem thông tin của các tiến trình: iexplore.exe, gpg-agent.exe.

Đáp án:

b. Kịch bản 02

- Tài nguyên: WIN-LEVQF1CLMR1-20190326-033038.raw

Yêu cầu 2. Thực hiện phân tích:

- Xem các tiến trình đang chạy
- Tìm thông tin tài khoản người dùng trên máy đối tượng.
- Lịch sử tiến trình cmd
- Xem nội dung một tập tin text do người dùng soạn thảo sử dụng notepad.
- Xem 2 URL mà người dùng truy cập gần nhất.

Đáp án:

c. Kịch bản 03

- Tài nguyên: Kb03-dp-e81.raw.lzma

> **Yêu cầu 3. Thực hiện phân tích:**
>
> - Cung cấp bằng chứng xác định file được cho là file dump từ bộ nhớ máy ảo. Xác định hệ điều hành của máy này.
> - Tìm flag cho file tài nguyên bên trên. Biết rằng flag có định dạng CTF{flag}.
>
> *Đáp án:*

d. Kịch bản 04

- Tài nguyên (Root-me.org): Tải memory dump tại: http://challenge01.root-me.org/forensic/ch2/ch2.tbz2
- Link challenge:
  + https://www.root-me.org/en/Challenges/Forensic/Command-Control-level-2
  + https://www.root-me.org/en/Challenges/Forensic/Command-Control-level-3
  + https://www.root-me.org/en/Challenges/Forensic/Command-Control-level-4
  + https://www.root-me.org/en/Challenges/Forensic/Command-Control-level-5
  + https://www.root-me.org/en/Challenges/Forensic/Command-Control-level-6

> **Yêu cầu 4. Thực hiện phân tích, hoàn thành các challenge trên:**
>
> - Thực hiện các bước điều tra, mô tả rõ ràng.
> - Có ảnh chụp, giải thích lí do.
>
> *Đáp án:*

e. Kịch bản 05

- Tài nguyên: Kb05-dp-E81.vmem

- Thực hiện các bước điều tra, mô tả rõ ràng.
- Có ảnh chụp, giải thích lí do các bước.

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

- Tìm tên và mật khẩu của tài khoản người dùng trong bộ nhớ
- Tìm tên (ComputerName) và địa chỉ IP của máy tính mục tiêu.
- Người dùng trên máy tính mục tiêu thích chơi một vài trò chơi điện tử cũ. Nêu tên trò chơi mà người này chơi. Cung cấp địa chỉ IP máy chủ của trò chơi.
- 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.
- 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.
- 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?
- Cho biết cách nào để mã độc xâm nhập và nhiễm vào máy tính của người này. Có phải do thói quen cũ?
- 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?
- Nhân viên điều tra xác định được mã độc là một ransomeware. Tìm địa chỉ ví Bitcoin của kẻ tấn công.
- Tìm mật khẩu mà kẻ tấn công dùng để mã hóa file.
- Trích xuất mật khẩu từ bộ nhớ, xem khả năng dùng mật khẩu này để giải mã file (do ransomeware mã hóa).

Đáp án:

C. THAM KHẢO

- Rekall Github: https://github.com/google/rekall
- Rekall: http://www.rekall-forensic.com
- Volatility: https://code.google.com/archive/p/volatility/downloads
- MoVP II - 1.2 - VirtualBox ELF64 Core Dumps: https://volatility-labs.blogspot.com/2013/05/movp-ii-12-virtualbox-elf64-core-dumps.html

D. YÊU CẦU

Bài thực hành được chia làm 2 phần riêng biệt.

- **Class Part (CP):** Sinh viên hoàn thành trên lớp (Bắt buộc).

  0% <= CP< 50%: 1đ

  50% <= CP < 90 %: 5đ

  90% <= CP <= 100%: 10đ

- **Home Part (HP):** Hoàn thành phần còn lại và làm báo cáo sau khi kết thúc buổi thực hành (nộp trên Course môn học theo deadline).
- Điểm Thực hành của mỗi Buổi (Session): **S = (CP + HP)/2**

– Sinh viên tìm hiểu và thực hành theo hướng dẫn.

– Nộp báo cáo kết quả gồm chi tiết những việc bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).

– Sinh viên báo cáo kết quả thực hiện và nộp bài.

Báo cáo:

– File .DOCX và .PDF. Tập trung vào nội dung, không mô tả lý thuyết.

– Chỉ dùng duy nhất 1 loại Font chữ (Times New Roman – cỡ chữ 12)

– Đặt tên theo định dạng: [Mã lớp]-SessionX_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành đã đăng ký với GVHD-TH).

Ví dụ: [NT101.H11.1]-Session1_Group3.

– Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.

– Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm bài Lab.

– Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá: Sinh viên hiểu và tự thực hiện được bài thực hành. Khuyến khích:

– Chuẩn bị tốt và đóng góp tích cực tại lớp.

– Có nội dung mở rộng, ứng dụng trong kịch bản phức tạp hơn, có đóng góp xây dựng bài thực hành.

Bài sao chép, nộp trễ, thực hiện không nghiêm túc ... sẽ được xử lý tùy mức độ vi phạm.

HẾT