-

Bài thực hành: Tìm hiểu về quá trình phân tích tĩnh

**TÌM HIỂU BÀI THỰC HÀNH**

**ptit-ranalystic**

**Giảng viên:** Đinh Trường Duy

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**

* + 1. Nội dung và hướng dẫn bài thực hành
  1. Mục đích

Mục đích của bài thực hành là hiểu rõ về mã độc Wannacry. Đây là một loại ransomware (phần mềm đòi tiền chuộc) đã gây ra một cuộc tấn công mạng lớn vào năm 2017, lan rộng trên toàn cầu và gây ra thiệt hại nghiêm trọng cho hàng ngàn tổ chức và cá nhân. WannaCry là loại mã độc khi thâm nhập vào thiết bị, máy tính của người dùng hoặc máy tính trong hệ thống doanh nghiệp sẽ tự động mã hoá hàng loạt các tập tin theo những định dạng mục tiêu như văn bản tài liệu, hình ảnh... Người dùng cá nhân cũng như doanh nghiệp sẽ phải trả một khoản tiền không hề nhỏ nếu muốn lấy lại các dữ liệu đó.

Có thể kiểm tra mức độ lây nhiễm virus Wanna Cry tại link: <https://intel.malwaretech.com/WannaCrypt.html>

Việc nghiên cứu về Wannacry giúp sinh viên hiểu rõ về cách hoạt động của ransomware, cách mà nó lây lan, và biện pháp phòng chống để bảo vệ mình khỏi những cuộc tấn công tương tự trong tương lai.

* 1. Yêu cầu đối với sinh viên

Sinh viên cần có kiến thức cơ bản về hệ điều hành và mạng máy tính và các khái niệm cơ bản về phân tích malware để hiểu và áp dụng các công cụ PEV, IDA. Hiểu biết về các công cụ như VirusTotal có thể hữu ích để bổ sung kiến thức trong quá trình phân tích mã độc WannaCry.

* 1. Nội dung thực hành

Trong bài lab này, sinh viên sẽ sử dụng IDA phiên bản 8.4

**Khởi động bài lab:**

- Vào terminal, gõ:

*labtainer ptit-ranalystic*

*(chú ý: sinh viên sử dụng mã sinh viên của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm)*

**Các nhiệm vụ:**

- Task 1: Kiểm tra loại tệp mã độc

+ Thư mục hiện tại đã chuẩn bị sẵn các tệp tin .pdf, .txt, .png để sinh viên phân tích. Để có thể xem chữ ký tệp một cách chi tiết, sinh viên hãy sử dụng lệnh sau:

*hexdump -C <tên file> | less*

+ Ngoài ra, Ubuntu có một công cụ dễ dàng có thể phát hiện ra loại tệp là “file”. Sinh viên dùng lệnh đó để kiểm tra từng tệp:

*file <tên file>*

* Sinh viên hãy chỉ ra từng loại tệp và chữ ký tệp của các tệp tin có sẵn.
* Chỉ ra tệp nghi ngờ

- Task 2: Trích xuất chuỗi trong mẫu đáng ngờ.

+ Phân tích chuỗi có thể cung cấp gợi ý về chức năng của chương trình và các chỉ số liên quan đến một tệp nhị phân đáng nghi. Các chuỗi được trích xuất từ tệp nhị phân có chứa các tham chiếu đến tên tệp, URL, tên miền, địa chỉ IP, lệnh tấn công, registry,... thì khả năng cao là có dính mã độc.

+ Phân tích chuỗi cũng giúp cung cấp các thông tin quan trọng về cách hoạt động của tệp. Từ đó, ta có thể phân tích hành vi xem chúng có phải là độc hại không.

+ Các loại chuỗi mà chúng ta nên tìm kiếm bao gồm:

• Tên tệp.

• Địa chỉ URL (Miền mà mã độc kết nối tới).

• Địa chỉ IP.

• Khóa Registry.

+ Kẻ tấn công cũng có thể bao gồm các chuỗi giả mạo để gây trở ngại cho quá trình phân tích.

+Trên terminal sử dụng lệnh strings kết hợp các lệnh như grep để tìm kiếm đầu ra của lệnh strings:

a. Trích xuất chuỗi bằng :

*strings <file mã độc> | less*

Khi ta chạy lệnh này trong Terminal, nó sẽ hiển thị tất cả chuỗi mà nó tìm thấy trong tập tin mã độc. Điều này có thể hữu ích để tìm thông tin như các chuỗi văn bản, địa chỉ email, URL, tên tệp tin, và nhiều hơn nữa mà có thể bị ẩn trong các tập tin nhị phân.

Sinh viên có thể tìm được thấy các chuỗi khả nghi như sau:

115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn

* Hãy tìm kiếm thêm các chuỗi khả nghi như vậy và viết vào báo cáo?
* Sau đó, sinh viên hãy tìm kiếm chuỗi này trên Internet và hãy cho biết chuỗi đó là gì?

b. Kếp hợp với grep:

Dùng strings thì nó sẽ hiển thị tất cả các chuỗi khiến cho việc đọc, phân tích trở lên khó hơn. Vì vậy kết hợp với lệnh grep để tìm kiếm chuỗi đặc biệt trong file nghi ngờ:

* + Tìm kiếm các hệ thống tệp:

*strings <file mã độc> | grep -i file*

* Cho biết các API mã độc có thể lợi dụng và mục đích của nó?
  + Tìm kiếm các Registry:

*strings <file mã độc> | grep -i reg*

* Liệt kê các Registry được sử dụng trong file mã độc. Mục đích của chúng là gì?
  + Mã hóa dữ liệu:

*strings <file mã độc> | grep -i crypt*

* Đây là hành vi đặc trưng của loại mã độc nào?
  + Thao tác với tiến trình:

*strings <file mã độc> | grep -i process*

* + Trốn tránh phần mềm anti-virus:

*strings <file mã độc> | grep -i resource*

* Hãy liệt kê một vài các Windows API khả nghi và giải thích chi tiết công dụng của chúng?
  + Tìm kiếm các thư viện (dll) để thao tác với hệ thống:

*strings <file mã độc> | grep -i dll*

* Chỉ ra 2 thư viện hệ điều hành quan trọng được sử dụng trong file mã độc này.

- Task 3: Kiểm tra xáo trộn hoặc nén.

+ Kẻ tấn công có thể sử dụng các kỹ thuật xáo trộn hoặc nén mã độc để che giấu chúng khỏi các công cụ phân tích. Chúng có thể nén và xáo trộn hầu hết toàn bộ bản thân hay chỉ 1 phần dữ liệu và lưu trữ ở các section. Bằng cách này, mã độc có thể tránh được phát hiện và phân tích từ các công cụ phân tích tĩnh thông thường.

+ Kiểm tra xem file có bị nén hay không, sinh viên có thể sử dụng lệnh sau:

*pescan <tên file>*

* Entropy của file là bao nhiêu?
* File có khả năng bị nén hay không?

+ Để kiểm tra file bị nén toàn bộ hay 1 phần, chúng ta sẽ tiến hành phân tích PE Header ở task sau.

- Task 4: Phân tích PE Header

+ Phân tích PE (Portable Executable) header là một phần quan trọng trong quy trình phân tích mã độc tĩnh, đặc biệt khi làm việc với các tệp thực thi Windows (có định dạng PE). PE header chứa thông tin về cấu trúc của tệp exe, bao gồm thông tin về mã máy tính, phân 17 đoạn, địa chỉ nhập, tài liệu nguồn, và nhiều chi tiết quan trọng khác.

+ Sử dụng công cụ readpe để thực hiện phân tích PE header. Thực hiện câu lệnh sau:

*readpe --all <tên\_file>*

* Hãy tìm và cho biết thời điểm mã độc được tạo ra?

+ Hãy kiểm tra imports và exports của tệp tin và trả lời câu hỏi sau:

* Các hàm imports và exports có xuất hiện trong tệp tin này không?
* Hãy chỉ ra các thư viện liên kết động (.dll) được sử dụng?
* Hãy chỉ ra các function (API) được sử dụng để quản lý dịch vụ, registry, tiến trình, file, bộ nhớ ảo và thực hiện xáo trộn dữ liệu và thông tin?

+ Ở phần section, sinh viên hãy cho biết:

* Có các section nào xuất hiện trong tệp tin?
* Hãy cho biết kích thước của từng loại section.

+ Sinh viên có thể phát hiện ra rằng section .rcsc có kích thước lớn hơn rất nhiều so với những section khác. Section .rsrc thường chứa các tài nguyên như hình ảnh, âm thanh, biểu tượng, cấu trúc menu, hoặc các dữ liệu tùy chỉnh khác được sử dụng bởi chương trình. Kích thước lớn của section này so với các section khác có thể gợi lên sự nghi ngờ về tính độc hại của tệp tin.

+ Sinh viên hãy phân tích sâu hơn vào section này bằng cách sử dụng peres để trích xuất các tệp tin có trong section rsrc này. Thực hiện câu lệnh sau:

*peres -x <tên\_file>*

* Hãy kiểm tra từng file bằng các câu lệnh ở task 1 và cho biết có file nào khả nghi không?

+ Sau khi thực hiện bước trên, sinh viên có thể thấy được có 1 file đang bị nén dưới dạng PKZIP, hãy thử giải nén nó bằng “unzip”:

*unzip <tên file>*

+ Tại đây, sinh viên thấy rằng tệp tin đang yêu cầu mật mã để có thể giải nén. Chúng ta sẽ tìm kiếm mật mã này ở task tiếp theo.

- Task 5: Dùng IDA tìm kiếm mật khẩu file PKZIP

+ Để cài đặt IDA, sinh viên hãy chạy lệnh sau:

*./idafree84\_linux.run*

+ Sinh viên mở công cụ IDA

*./idafree-8.4/ida64 <tên file>*

+ Sau khi đã mở được tệp lên IDA, sinh viên xem qua mã hợp ngữ và cố gắng phân tích ngược chức năng của mã độc.

* Sinh viên cho biết mã độc thêm một registry value tại địa chỉ nào? Tìm hiểu trên mạng địa chỉ đầy đủ là gì?

+ Sinh viên hãy tìm kiếm hàm WinMain vì đây là hàm chính của chương trình này. Ở bước đầu, chương trình đã cố gắng tạo dịch vụ Windows, sau đó giải nén nội dung của file được mã hóa PKZIP được lưu trữ bên trong file .exe.

* Sinh viên hãy cho biết mật khẩu để giải mã là gì?
* Sinh viên cho biết mã độc tạo hai process nào?
* Sinh viên cho biết mã độc tạo mutex với tên là gì? Chức năng của mutex này là gì?

+ Các địa chỉ Bitcoin và danh sách các phần mở tệp tin được cố định sẵn trong tệp. Ngoài ra, danh sách các phần mở tệp tin xác định những tệp nào sẽ bị mã hóa.

* Sinh viên hãy cho biết mã độc này mã hóa các tập tin với phần mở rộng nào?

- Task 6: Mở file PKZip

+ Sinh viên giải nén tệp:

*unzip -P <password> <tên file> -d extracted*

+ Khi giải nén file PKZIP ở trong file EXE, sinh viên sẽ tìm thấy hướng dẫn gửi tiền, các file exe khác (taskdl.exe và taskse.exe)

+ Một trong những tệp này chứa danh sách địa chỉ TOR mà những kẻ tấn công sử dụng để liên lạc với các nạn nhân.

* Sinh viên tìm hiểu địa chỉ TOR là gì? Thường xuất hiện dưới dạng nào?
* Sinh viên hãy cho biết có bao nhiêu địa chỉ TOR và liệt kê các địa chỉ TOR này?

+ Sinh viên có thể phân tích sâu hơn các địa chỉ TOR này bằng việc gửi các địa chỉ này lên VirusTotal để phân tích.

+ Sau khi hoàn thành quá trình mã hóa, WannaCry cố gắng ngăn chặn các phương pháp phục hồi dữ liệu phổ biến bằng cách thực hiện một số lệnh trên hệ thống.

+ Sinh viên cần phân tích 2 file .exe sau khi giải nén tệp PKZip ở trên các bước tương tự như phân tích tệp .exe ban đầu (Kiểm tra loại tệp, tạo mã hash, trích xuất chuỗi, kiểm tra xáo trộn hoặc nén, phân tích PE Header)

- Task 7: Phân loại mã độc sử dụng Fuzzy Hashing.

+ Trong thực tế, có những dạng mã độc chưa bao giờ được phát hiện nhưng chúng lại có tính tương đồng với những mã độc đã biết. Vì vậy, để kiểm tra xem mã độc thuộc họ mã độc nào, sinh viên hãy dùng Fuzzy hashing - một phương pháp để so sánh sự tương đồng giữa các tập tin. Thư mục hiện tại bao gồm thư mục malwares có sẵn một số mã độc Ransomware. Trước tiên, sinh viên hãy tạo fuzzy hash cho thư mục đó bằng công cụ ssdeep và lưu thành 1 file output.txt. Sinh viên sử dụng lệnh sau:

*ssdeep -r <tên thư mục> > output.txt*

+ Sau đó, sinh viên hãy so sánh các fuzzy hash với file nghi ngờ bằng lệnh sau:

*ssdeep -m output.txt <tên file>*

* File mã độc trên tương đồng với family Ransomware nào và tỉ lệ là bao nhiêu?

+ Tiếp theo, sinh viên hãy thực hiện kết nối đến 1 container khác bằng ssh thông qua lệnh sau:

*ssh ubuntu@192.168.0.20*

+ Mật khẩu chính là 1 trong 2 mã hash md5 mà sinh viên đã lấy được ở các bước trên. Sau đó, hãy thực hiện in kết quả của tệp tin filetoview ra màn hình:

*cat filetoview*

**Kết thúc bài lab:**

- Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

*stoplab ptit-ranalystic*

- Khi bài lab kết thúc, một tệp zip lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab.

**Khởi động lại bài lab:**

- Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

*labtainer -r ptit-ranalystic*