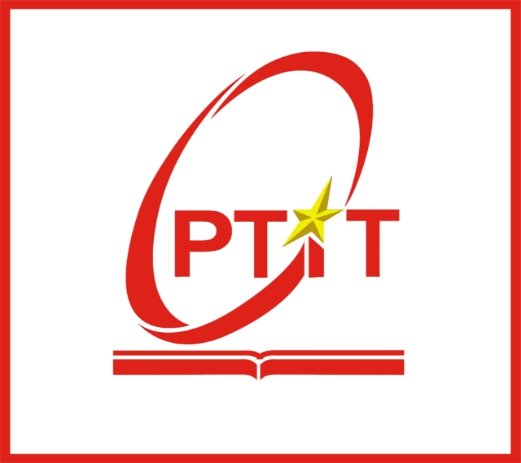


**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**



**ptit-static-analysis-on-windows**

Giảng viên: Đinh Trường Duy

**MỤC LỤC**

1. **Bài thực hành phân tích tĩnh trên Windows**
   1. **Nội dung và hướng dẫn bài thực hành**
      1. ***Mục đích***

* Giúp sinh viên tìm hiểu về mã, dữ liệu, cấu trúc của phần mềm độc hại
* Giúp sinh viên nắm được cách sử dụng các công cụ phân tích phần mềm độc hại cơ bản trên Windows
  + 1. ***Yêu cầu với sinh viên***
* Có kiến thức cơ bản về phân tích mã độc, nắm được quy trình phân tích tĩnh trong phân tích mã độc.
  + 1. ***Nội dung thực hành***
* Khởi động bài lab
* Vào terminal gõ:

*labtainer ptit-static-analysis-on-windows*

(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)

* Trước khi khởi động bài lab nếu máy Windows 7 và máy Ubuntu chưa được cấu hình mạng NAT, sinh viên cần cấu hình môi trường thực hành bằng cách mở máy Windows 7, cài đặt máy Windows 7 và máy Ubuntu cùng một Network Adapter là NAT bằng cách chọn:

*VM => Settings… => Network Adapter => NAT*

* Sau đó sinh viên đăng nhập vào máy Windows 7 với tài khoản (*user1:user1*), thực hiện kiểm tra địa chỉ IP của máy Windows 7 => Giả sử tìm được địa chỉ IP là *192.168.1.128*
* Khi khởi động bài lab, một cửa sổ ubuntu hiện lên, sinh viên thực hiện kết nối đến máy windows bằng lệnh

*./connect\_windows.sh.x < địa chỉ IP của máy Windows >*

* Sau khi kết nối thành công, cửa sổ Windows 7 sẽ hiện lên. Sinh viên cần mở một cửa sổ Windows powershell để thực hành và kiểm tra kết quả thực hành trên Windows, sinh viên thực hiện các lệnh trên cửa sổ powershell này trong suốt quá trình làm bài thực hành.
* Sau đó sinh viên xem phần mềm độc hại có tên financials-xls.exe tại đường dẫn C:\Users\user1\Downloads\financials-xls.

(Chú ý: Sau mỗi lần thực hiện một lệnh theo nhiệm vụ của bài thực hành cần chạy chương trình [*\\tsclient\C\_\check\_lab.ps1*](file:///\\tsclient\C_\check_lab.ps1) để ghi lại tiến độ làm bài thực hành)

* ***Nhiệm vụ 1:*** Xác định thông tin cơ bản về mẫu phần mềm độc hại. Sinh viên sử dụng công cụ CFF Explorer để kiểm tra thông tin về loại tệp của phần mềm độc hại financials-xls.exe.
* Để hoàn thành được nhiệm vụ này sinh viên cần xác định loại tệp của tệp độc hại tại mục File Type, và dạng thập lục phân của chuỗi ASCII cho biết định dạng của tệp độc hại được đặt tại mục Hex Editor (sử dụng lệnh echo trong cửa sổ Windows powershell để in ra màn hình thông tin tìm được tương ứng và chạy chương trình [*\\tsclient\C\_\check\_lab.ps1*](file:///\\tsclient\C_\check_lab.ps1) để ghi lại tiến độ làm bài thực hành).

*Để kiểm tra kết quả sinh viên thực hiện kiểm tra bằng checkwork tại mục file\_type*

* ***Nhiệm vụ 2:*** Xác định mã nhận dạng, dấu vân tay duy nhất cho mẫu phần mềm độc hại. Mã nhận dạng thường có dạng hàm băm mật mã MD5, SHA1 hoặc SHA256. Lấy dấu vân tay được sử dụng cho nhiều mục đích, bao gồm:
* Xác định và theo dõi các mẫu phần mềm độc hại
* Quét toàn bộ hệ thống để tìm sự hiện diện của phần mềm độc hại giống hệt nhau
* Chia sẻ với các bên liên quan dưới dạng IoC hoặc như một phần của báo cáo thông tin về mối đe dọa
* Để hoàn thành được nhiệm vụ này sinh viên cần lấy hàm băm MD5 và SHA256 của tệp độc hại financials-xls.exe bằng cách sử dụng lệnh Get-FileHash trên Windows PowerShell (chạy chương trình [\\tsclient\C\_\check\_lab.ps1](file:///\\tsclient\C_\check_lab.ps1) để ghi lại tiến độ làm bài thực hành)

*Get-FileHash -Algorithm MD5 <đường dẫn đến tệp độc hại>*

Để kiểm tra hàm băm SHA256 của phần mềm độc hại sinh viên sử dụng lệnh :

*Get-FileHash -Algorithm SHA256 <đường dẫn đến tệp độc hại>*

*Để kiểm tra kết quả tại nhiệm vụ này sinh viên thực hiện checkwork tại mục fingerprinting*

* ***Nhiệm vụ 3:*** Kiểm tra hàm băm tệp được tạo với VirusTotal một công cụ quét phần mềm độc hại trực tuyến, kết hợp với nhiều nhà cung cấp phần mềm chống virus khác nhau, cho phép tìm kiếm hàm băm của tệp. IMPHASH, viết tắt của "Import Hash", là hàm băm mật mã được tính từ các chức năng nhập của tệp Windows Portable Executable (PE).
* Để hoàn thành nhiệm vụ này sinh viên cần mở trang VirusTotal, tìm IMPHASH của mã độc financials-xls.exe và so sánh với IMPHASH tính được từ chương trình python imphash\_calc.py.

Sử dụng module Pefile trong python để tính toán IMPHASH cho tệp financials-xls.exe bằng cách tạo file imphash\_calc.py chứa đoạn mã python sau:

import sys

import pefile

import peutils

pe\_file = sys.argv[1]

pe = pefile.PE(pe\_file)

imphash = pe.get\_imphash()

print(imphash)

Sử dụng lệnh:

*python C:\path\imphash\_calc.py C:\path\ financials-xls.exe*

(chạy chương trình [*\\tsclient\C\_\check\_lab.ps1*](file:///\\tsclient\C_\check_lab.ps1) để ghi lại tiến độ làm bài thực hành)

*Để kiểm tra kết quả sinh viên thực hiện checkwork tại mục IMPHASH*

* ***Nhiệm vụ 4:*** Fuzzy Hashing (SSDEEP), còn được gọi là băm từng phần, là một kỹ thuật băm được thiết kế để tính toán giá trị băm biểu thị sự giống nhau về nội dung giữa hai tệp. Kỹ thuật này chia tệp thành các khối nhỏ hơn, có kích thước cố định và tính toán hàm băm cho mỗi khối. Các giá trị băm thu được sau đó được hợp nhất để tạo ra hàm băm mờ cuối cùng.
* Sinh viên sử dụng công cụ ssdeep để thực hiện tính mã SSDEEP cho tệp độc hại bằng lệnh:

*ssdeep.exe C:\Samples\MalwareAnalysis\ financials-xls.exe*

* Tại đây sinh viên so sánh kết quả SSDEEP thu được với SSDEEP trong mục Details lab trên VirusTotal, in ra màn hình mã ssdeep tìm được từ VirusTotal bằng lệnh echo trong Windows powershell.

(chạy chương trình [*\\tsclient\C\_\check\_lab.ps1*](file:///\\tsclient\C_\check_lab.ps1) để ghi lại tiến độ làm bài thực hành)

*Để kiểm tra kết quả sinh viên thực hiện checkwork tại mục fuzzy\_hashing*

* ***Nhiệm vụ 5:*** Section hashing, (băm các phần PE) là một kỹ thuật mạnh mẽ cho phép các nhà phân tích xác định các phần của tệp (PE) đã được sửa đổi. Bằng cách áp dụng section hashing, các nhà phân tích bảo mật có thể xác định các phần của tệp PE đã bị giả mạo hoặc thay đổi. Trong python có thể sử dụng module Pefile để thực hiện tính toán section hashing.
* Để hoàn thành nhiệm vụ này sinh viên sử dụng mô-đun pefile để truy cập và băm dữ liệu trong các phần riêng lẻ của tệp PE bằng việc chỉnh sửa file python có sẵn theo mẫu sau với đúng định dạng tên section\_hashing.py

import sys

import pefile

pe\_file = sys.argv[1]

pe = pefile.PE(pe\_file)

for section in pe.sections:

print (section.Name, "MD5 hash:", section.get\_hash\_md5())

print (section.Name, "SHA256 hash:", section.get\_hash\_sha256())

Bên cạnh đó sinh viên cần kiểm tra các hàm băm của file mã độc financials-xls.exe bằng cách sử dụng công cụ PEStudio. Sinh viên cần chỉ ra được hàm băm MD5 của file đặt tại mục footprints

(chạy chương trình [*\\tsclient\C\_\check\_lab.ps1*](file:///\\tsclient\C_\check_lab.ps1) để ghi lại tiến độ làm bài thực hành)

*Để kiểm tra kết quả sinh viên thực hiện checkwork tại mục section\_hashing*

* ***Nhiệm vụ 6:*** Mục tiêu của nhiệm vụ này là trích xuất các chuỗi (ASCII & Unicode) từ tệp nhị phân. Các chuỗi có thể cung cấp manh mối và thông tin chi tiết có giá trị về chức năng của phần mềm độc hại. Bên cạnh đó giải pháp phân tích chuỗi khác được gọi là FLOSS. FLOSS là viết tắt của "FireEye Labs Obfuscated String Solver", là một công cụ được phát triển bởi nhóm FLARE của FireEye để tự động giải mã các chuỗi trong phần mềm độc hại. Nó được thiết kế để bổ sung thay cho việc sử dụng các công cụ chuỗi truyền thống, thể bỏ sót các chuỗi bị xáo trộn thường được phần mềm độc hại sử dụng để tránh bị phát hiện
* Để hoàn thành nhiệm vụ này sinh viên cần sử dụng strings và floss trên powershell để hiển thị các chuỗi có trong phần mềm độc hại.
  + Sử dụng strings bằng lệnh:

*strings <đường dẫn đến tệp độc hại>*

* + Sử dụng floss bằng lệnh:

*floss <đường dẫn đến tệp độc hại>*

(chạy chương trình [*\\tsclient\C\_\check\_lab.ps1*](file:///\\tsclient\C_\check_lab.ps1) để ghi lại tiến độ làm bài thực hành)

* Tại đây sinh viên chỉ ra sự khác biệt giữa các chuỗi trích xuất bởi strings so với floss?

*Để kiểm tra kết quả sinh viên thực hiện checkwork tại mục strings*

* ***Nhiệm vụ 7:*** Trong phân tích tĩnh, có thể phát hiện một mẫu phần mềm độc hại đã được nén hoặc làm xáo trộn bằng kỹ thuật được gọi là đóng gói. Điều này có thể làm giảm khả năng phân tích chuỗi vì các tham chiếu đến chuỗi thường bị che khuất hoặc bị loại bỏ. Kết quả là, tệp phần mềm độc hại trở nên khó phân tích hơn vì không thể quan sát trực tiếp mã gốc. Một trình đóng gói phổ biến được sử dụng là Ultimate Packer for Executables (UPX).
* Từ kết quả Section Headers trong CFF Explorer khi thực hiện kiểm tra với tệp độc hại financials-xls.exe tại nhiệm vụ 1 và từ kết quả trích xuất chuỗi với lệnh strings ở nhiệm vụ 6 , thấy rằng đầu ra có chứa các chuỗi bao gồm UPX. Từ đây sinh viên rút ra kết luận gì về financials-xls.exe?
  + Sử dụng UPX bằng lệnh:

*upx -d -o <tên tệp sau khi giải nén> <đường dẫn đến tệp độc hại>*

(chạy chương trình [*\\tsclient\C\_\check\_lab.ps1*](file:///\\tsclient\C_\check_lab.ps1) để ghi lại tiến độ làm bài thực hành)

Sau đó sinh viên thực hiện lại lệnh strings để trích xuất lại các chuỗi từ tệp sau khi giải nén

Sinh viên rút ra được kết luận gì giữa kết quả chuỗi thu được trước và sau khi giải nén tệp độc hại?

*Sau khi hoàn thành nhiệm vụ sinh viên thực hiện checkwork tại mục unpacking\_upx*

* Để thúc bài lab :
* Trên terminal đầu tiên sử dụng lệnh sau để kết thúc bài lab:

*stoplab ptit-static-analysis-on-windows*

* 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-static-analysis-on-windows*