# 

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

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

------o0o------



**TÌM HIỂU BÀI THỰC HÀNH**

**PTIT – STATIC- ANALYSIS**

Giảng viên: Đinh Trường Duy

# Nội dung và hướng dẫn thực hiện bài thực hành

## 1.1. Mục đích

* Giúp sinh viên hiểu về quy trình, các công cụ trong quá trình phân tích mã độc và thực hiện phân tích tĩnh mã độc.

## 1.2. Yêu cầu đối với sinh viên

* Có kiến thức cơ bản về mã độc và cách sử dụng các công cụ để phân tích mã độc.

## 1.3. Nội dung thực hành

* Khởi động bài lab:
  + Vào terminal, gõ:

*Labtainer -r ptit-static-analysis*

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

Sau khi khởi động xong một terminal ảo sẽ xuất hiện.

* Các nhiệm vụ thực hành:
  + Nhiệm vụ 1: Thu thập thông tin file của mẫu mã độc.
  + Trên terminal ***ptit-static-analysis*** sử dụng lệnh “file”, xác định thông tin của mẫu. Từ đó có thể biết được những thông tin gì về sample.exe?
* Nhiệm vụ 2: Thu thập thông tin file của mẫu mã độc.
* Sử dụng lệnh “md5sum” để lấy chữ ký của mẫu.
* Mở trình duyệt (gõ firefox ở máy *ptit-static-analysis*), truy cập vào [*https://www.virustotal.com/gui/home/upload*](https://www.virustotal.com/gui/home/upload) để tìm kiếm thông tin từ chữ ký của mẫu. Có bao nhiêu nhà cung cấp và môi trường cho rằng đây là tệp tin độc hại?
  + Nhiệm vụ 3: Phân tích chuỗi
  + Thực hiện phân tích các chuỗi trong mẫu, sử dụng lệnh “strings” để xem xét các chuỗi đáng ngờ như các URL, tên tệp, tên các thư viện được sử dụng, chuỗi mã hóa, …
  + Sử dụng lệnh “echo” để in ra màn hình các chuỗi đáng ngờ.
  + Nhiệm vụ 4: Phân tích Upacking:
  + Từ các chuỗi thu được có thể biết được mẫu này đã sử dụng kỹ thuật gì để che dấu và tránh bị phát hiện?
  + Dùng công cụ UPX để unpack mẫu, sử dụng lệnh: “upx -d sample.exe -o unpack\_sample.exe”.
  + Phân tích lại các chuỗi trong mẫu sau khi đã unpack. Từ đó thu thập được các thông tin đáng ngờ nào?
  + Sử dụng lệnh “echo” để in ra những chuỗi đáng ngờ.
  + Nhiệm vụ 5: Xác định các hành vi độc hại của mẫu
  + Sử dụng công cụ Radare2 để tìm chức năng chính của mẫu, dùng lệnh:

r2 ./unpack\_sample.exe

* Lệnh “aaaaa”: Tự động phân tích một file nhị phân hoặc một đoạn mã.
* Liệt kê tất cả các hàm (Analysis Functions List), sử dụng lệnh “afl” sẽ quét qua mã máy và xác định tất cả các hàm có trong chương trình. Điều này bao gồm các hàm hợp lệ và các hàm có thể không hoàn chình hoặc không được sử dụng. Lệnh “afl” sẽ hiện thị thông tin của mỗi hàm gồm: đại chỉ bắt đầu, địa chỉ kết thúc, kích thước của hàm và thông tin khác liên quan.
* Để hiện thị mã hợp ngữ của một hàm, sử dụng lệnh “s <địa chỉ bắt đầu của hàm>” để chuyển vùng nhớ hiện tại đến đại chỉ bắt đầu của hàm cụ thể trong mã máy (ví dụ: s 0x00401190). Sau đó sử dụng lệnh “pdf” để hiện thị mã máy của hàm mà con trỏ vùng nhớ hiện tại đang trỏ tới. Nó tạo ra một phiên bản của mã hợp ngữ, cho phép xem và phân tích mã máy cụ thể của hàm đó.
* Quan sát tất cả các hàm để xác định hành vi chính của mẫu mã độc sample.exe là gì (phải xem 2 hàm chính mới có thể hoàn thành nhiệm vụ)?
  + Nhiệm vụ 6: Giải mã flag:
* Phân tích kỹ các hàm của mã độc trong nhiệm vụ 5. Từ đó xác định được các hành vi độc hại mà mã độc thực hiện lên máy của nạn nhân. Mã độc đã tạo một bộ hẹn giờ để chờ đến thời điểm thực hiện hành vi độc hại.
* Sử dụng công cụ 7zip để giải nén file Flag.zip với câu lệnh là 7z x Flag.zip -p<password>.password là năm mà mã độc đã tạo bộ hẹn giờ. Giải mã chuỗi ký tự trong file Flag.txt để thu được flag ở dạng bản rõ và in ra màn hình bản rõ của flag với lệnh “echo”.
* Kết thúc bài lab:
  + Trên terminal đầu tiên sử dụng câu lênh sau để kết thúc bài lab:

*stoplab ptit-static-analysis*

* + 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*