-

Bài thực hành: Dịch ngược ngôn ngữ Python và phân tích mã độc Trojan cơ bản

**TÌM HIỂU BÀI THỰC HÀNH**

**PTIT - Resource**

**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

Bài thực hành này tập trung vào việc giúp sinh viên làm quen với việc dịch ngược ngôn ngữ Python và phân tích mã độc Trojan cơ bản thông qua sử dụng các công cụ như PyInstaller Extractor và uncompyle6. Mục đích chính là giúp sinh viên hiểu được cách thức hoạt động của các công cụ này và cách chúng có thể được áp dụng để khai thác mã nguồn Python được bảo vệ.

PyInstaller Extractor là một công cụ quan trọng trong quá trình dịch ngược Python. Nó cho phép chúng ta trích xuất mã nguồn từ các ứng dụng Python được đóng gói bằng PyInstaller. Bằng cách này, chúng ta có thể xem mã nguồn gốc của một ứng dụng mà không cần phải có mã nguồn ban đầu.

Ngoài ra, uncompyle6 là một công cụ dịch ngược Python mạnh mẽ, giúp chúng ta chuyển đổi mã bytecode của Python (các file .pyc) thành mã nguồn Python có thể đọc được. Điều này giúp chúng ta phân tích và hiểu được mã nguồn của một ứng dụng Python mà không cần phải có mã nguồn ban đầu.

Đối với những sinh viên muốn tìm hiểu sâu hơn về uncompyle6, có thể tham khảo trang web chính thức của dự án tại <https://pypi.org/project/uncompyle6/>.

* 1. Yêu cầu đối với sinh viên

Có kiến thức cơ bản về mã độc trojan và ngôn ngữ lập trình Python.

* 1. Nội dung thực hành

**Khởi động bài lab:**

- Vào terminal, gõ:

*labtainer ptit-resource*

*(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: Khởi tạo môi trường

+ Sau khi khởi động xong 2 terminal ảo sẽ xuất hiện. Ở terminal ptit-resource-attack thực hiện các câu lệnh:

*cd server*

*sudo ./server*

+ Trên terminal ptit-resource, thực hiện các câu lệnh sau:

*cd google*

*sudo ./google*

- Task 2: Sử dụng Wireshark để bắt gói tin

+ Trên terminal ptit-resource, thực hiện câu lệnh để chạy công cụ Wireshark:

*sudo wireshark &*

+ Giao diện Wireshark xuất hiện, chọn card mạng eth0

+ Thực hiện lại câu lệnh: *sudo ./google*

=> Theo dõi gói tin trên Wireshark, phát hiện các gói tin nghi ngờ

* Những dữ liệu nào được gửi đi?

- Task 3: Kiểm tra String để kiểm tra các hàm, thư viện được sử dụng

+ Trên terminal ptit-resource, thực hiện kiểm tra strings của file google

*strings google*

Lệnh strings trong Ubuntu được sử dụng để trích xuất chuỗi văn bản có thể đọc được từ một tập tin nhị phân. Cụ thể, nó sẽ tìm kiếm trong tệp tin đầu vào và trích xuất tất cả các chuỗi ký tự có thể đọc được mà không cần phải dịch ngược hay phân tích cú pháp.

* Ngôn ngữ gốc được sử dụng để tạo ra chương trình này là gì?

- Task 4: Dùng pyinstxtractor để biên dịch ngược thành file.pyc

+ Trên máy ảo ptit-resource có sẵn công cụ pyinstrxtractor. Thực hiện các câu lệnh sau:

*python pyinstxtractor.py google*

+ Bây giờ, PyInstaller Extractor đã tạo ra thư mục google\_extracted gồm nhiều file liên quan trong đó có file google.pyc

* Cho biết file .pyc là file gì?

- Task 5: Sử dụng uncompyle để chuyển file.pyc thành file.py

+ Trên terminal, sử dụng các câu lệnh sau:

*uncompyle6 /google \_extracted/google.pyc > google.py*

+ Cuối cùng, sinh viên thu được mã nguồn python từ file google

* Phân tích mã nguồn và cho biết mã độc đã gửi dữ liệu qua cổng nào?

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

*ssh ubuntu@172.25.0.10*

+ Mật khẩu là mã hash md5 của file thực thi có tên “google”. Sau đó, hãy thực hiện in kết quả của tệp tin *readme* ra màn hình:

*cat readme*

**Kết thúc bài lab:**

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

*stoplab ptit-resource*

- 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:

*stoplab ptit-resource*