BÁO CÁO BÀI TẬP 01

**Môn học: Cơ chế hoạt động của mã độc**

**Kỳ báo cáo: Buổi 01 (Session 01)**

**Tên chủ đề: Virus đơn giản trên tập tin PE**

*GVHD: Phan Thế Duy*

*Ngày báo cáo: 08/05/2021*

1. **THÔNG TIN CHUNG:**

Lớp: NT230.L21.ANTT

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Nguyễn Quang Trường | 18520182 | 18520182@gm.uit.edu.vn |
| 2 | Đào Trung Nguyên | 18521156 | 18521156@gm.uit.edu.vn |
| 3 | Bùi Chí Trung | 18521544 | 18521544@gm.uit.edu.vn |

BÁO CÁO CHI TIẾT

1. **Viết 1 chương trình bằng ngôn ngữ Python có khả năng thêm mới một Section, xóa một Section trong file PE. Kiểm tra lại chức năng của file PE sau khi thao tác để đảm bảo chương trình hoạt động bình thường.**

* Ngôn ngữ được sử dụng cho bài này là python 2.7, file pe dùng để thêm, xóa section là putty.exe được cung cấp trong phần resource.
* Chương trình gồm có 3 hàm main, add, delete. Trong đó: hàm add thêm mới 1 section, hàm delete xóa 1 section trong file pe và hàm gọi thực hiện hai hàm này.
* Đầu tiên là hàm main: nhận input đầu vào là 1 số nguyên dùng để lựa chọn hành động thêm hay xóa một section, đường dẫn của file pe sẽ được gán cho biến exe\_path.

Text

Description automatically generated

* Hàm add: nhận tham số đầu vào là đường dẫn của file pe, thực hiện thêm mới một section theo các bước sau đây:
* Bước 1: Thay đổi kích thước file pe, vì ta cần thêm mới section nên cần làm cho file có kích thướt lớn hơn ban đầu

Text

Description automatically generated

* Bước 2: Thêm section header mới.
* Thêm offset, tính toán các giá trị raw\_size, virtual\_size,… mới cho section.

Text

Description automatically generated

* Đặt thuộc tính cho section với giá trị các cờ: chứa executable code: 0x00000020, execute code: 0x20000000, read: 0x40000000, write: 0x80000000. Tổng là: 0xE0000020
* Nhập tên cho section, tuy nhiên tên này phải đủ 8 byte

Text

Description automatically generated

* Tạo section, gán tên vừa nhập, đặt giá trị cho các trường trong section table.

Text

Description automatically generated

* Bước 3: Sửa lại header chính, tăng số section lên, tính lại sizeofimage. Sau khi hoàn tất ghi những thay đổi trở lại file pe theo đường dẫn cung cấp.

Text

Description automatically generated

* Hàm delete: nhận tham số đầu vào là đường dẫn của file pe, thực hiện xóa section ở cuối theo các bước sau đây:
* Bước 1: xác định vị trí cần xóa(section cuối), có thể chỉ định index là một vị trí section bất kì. Tuy nhiên để không ảnh hưởng tới chức năng của file, nhóm chọn xóa section vừa được thêm vào cuối của hàm trên.

Text

Description automatically generated

* Bước 2: Xóa các dữ liệu liên quan, nếu index là vị trí của section bất kì thì cần kiểm tra xem section này có tồn tại không.

Text

Description automatically generated

* Bước 3: Thay đổi số section, ghi những chỉnh sửa xuống pe file được chỉ định

Text

Description automatically generated

* Tiến hành thực hiện thêm và xóa section vừa thêm:
* File pe trước khi thêm:

Graphical user interface, application

Description automatically generated

* Thêm: nhập input là một số khác không để chọn hành động thêm. Nhập tên section là ahihi. Kết quả chạy chương trình và kiểm tra lại bằng pestudio:

Text

Description automatically generated

Table

Description automatically generated with low confidence

* Vẫn mở được file pe bình thường:

Graphical user interface, application

Description automatically generated

* Xóa: nhập input là 0 để chọn hành động xóa section vừa thêm ở trên. Kết quả chạy trương trình và kiểm tra lại bằng pestudio:

Text

Description automatically generated

Graphical user interface, application, table, Excel

Description automatically generated

* Vẫn mở file được bình thường:

A picture containing text, screenshot, indoor

Description automatically generated

1. **Viết một virus máy tính đơn giản bằng cách thực hiện thao tác chèn 1 shellcode đơn giản (hiển thị Popup chào mừng có nội dung: “MSSV1-MSSV2-MSSV3” – và thanh tiêu đề của popup có nội dung là: “NT230”) vào phần Section mới thêm & thay đổi Entry Point (để đảm bảo chức năng chương trình gốc hoạt động bình thường). Kiểm tra lại chức năng sau khi thao tác (nếu có lỗi lúc chạy, thử tìm lí do). Quan sát kết quả thực thi và giải thích.**

* Để có thể popup của sổ với thông báo như yêu cầu, dùng payload để tạo shellcode như messagebox đúng như yêu cầu:

Text

Description automatically generated

* Đã có shellcode, để có thể chạy xong shellcode mà app vẫn dùng được bình thường thì ta cần phải đổi 6 bytes cuối của shellcode thành địa chỉ Entry point lúc đầu

A picture containing graphical user interface

Description automatically generated

*6 bytes cuối của shellcode*

* Với entrypoint đã tìm được trước khi thây đổi file PE

A picture containing text, black

Description automatically generated

* Bây giờ sẽ đổi 6 bytes cuối shellcode thành call tới entrypoint này

Text

Description automatically generated with low confidence

*6 bytes cuối mới của shellcode*

* Chúng ta cần phải cộng thêm 0x400000 ( 0x400000 + 0x33e0c) sẽ ra được là 0x433e0c vì file được load vào bộ nhớ, loader sẽ thêm trường image-base trong Optional Header với image\_base là 0x400000.
* Để thêm vào shellcode thì ta cần phải đảo ngược lại vì trong nền tảng intel sử dụng giá trị theo dạng little-endian.
* Sau khi có được entry point cần phải trỏ tới rồi thì có ta có thể sửa trực tiếp trên shellcode nhét vào code và chạy hoặc có thể code thêm để entry point tự động sửa trong shellcode.
* Ta sẽ dùng cách thêm trực tiếp vào shellcode. Vì địa chỉ của original entry point luôn như vậy nên ta có thể dùng cách này.
* Chạy code và kiểm tra kết quả. Thấy có hiện message và app vẫn dùng bình thường

A screen shot of a computer

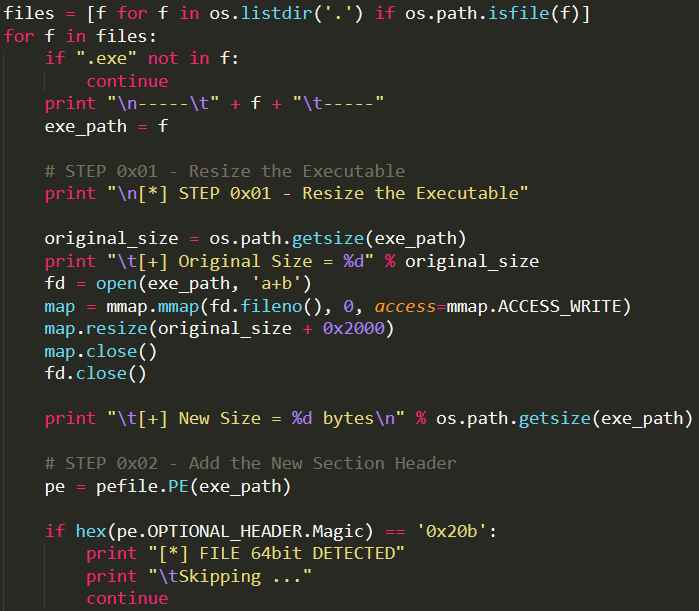
Description automatically generated with low confidence

Graphical user interface, text, application

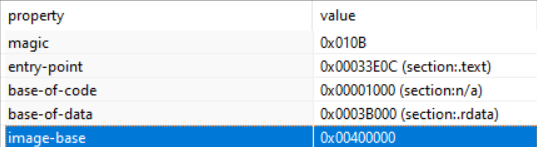
Description automatically generated

1. **Thực nhiễm lây nhiễm tất cả các file PE khác trong cùng một thư mục khi tập tin virus ở câu b được thực thi (mang file virus sang một thư mục khác bất kỳ và mở lên).**

* Để có thể lây nhiễm được các PE khác trong cùng một thư mục, ta cần nắm rõ một điều đó là mỗi file PE sẽ có các Entry Point mặc định khác nhau. Lời gọi địa chỉ trả về Entry Point mặc định vì thế cũng phải thay đổi theo từng file.
* Việc đầu tiên ta cần làm đó là quét hết một toàn bộ thư mục hiện đang chứa file virus để tìm ra tất cả file PE đang tồn tại. Để thực hiện điều này, ta sử dụng hàm **listdir()** thuộc thư viện os kết hợp cùng hàm **isfile()** cùng thư viện.
* Sau đó, với mỗi file được load lên, ta kiểm tra nếu đó có phải file PE và file đó thuộc loại 32bit hay 64bit. Do shellcode hiện tại chỉ chạy được cho file 32bit nên file 64bit sẽ được bỏ qua.



* Để shellcode có thể linh động với từng Entry Point mặc định, ta sẽ xóa đi 6 bytes cuối cùng của đoạn shellcode được dùng trong câu B, chỉ chừa lại đúng 1 byte **“\xb8”**, tức lệnh mov.
* Tiếp theo, ta viết một hàm với dữ liệu đầu vào là Original Entry Point (oep) của file PE trước khi bị thay đổi trong bước 3. Hàm này sẽ cộng thêm vào oep 0x400000. Vì khi file được load vào bộ nhớ, loader sẽ thêm trường image-base trong Optional Header vào Entry Point.



* Sau đó, hàm sẽ tiến hành đảo được từng bit trong chuỗi oep vì các platfrom nền Intel sử dụng giá trị theo Little-Endian. Ta sẽ thêm vào sau chuỗi đã được đảo ngược 2 bytes **“ffd0”**, ứng với lệnh call để call đến Original Entry Point.
* Lúc này, oep đang ở dạng string và chưa thể đưa vào shellcode. Ta sẽ dùng hàm unhexlify() thuộc thư viện binascii để chuyển chuỗi string này thành dạng binary.
* Cuối cùng, hàm sẽ trả về đoạn shellcode mới, với dữ liệu gồm đoạn shellcode cũ mặc định + đoạn call Original Entry Point tương ứng với file đang chỉnh sửa.

Text

Description automatically generated

* Giờ ta chỉ việc đưa file virus này vào bất cứ file nào ta mong muốn và bật lên. Những đoạn code bên trong sẽ lo phần còn lại.
* Chạy và kiểm tra thử khi trong thư mục **không** có file PE nào:

Text

Description automatically generated

* Chạy và kiểm tra thử khi trong thư mục có 2 file PE:

Text

Description automatically generated