BÁO CÁO BÀI TẬP

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

**Tên chủ đề: File Infecting Virus**

*GVHD: Nguyễn Công Danh*

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT230.O22.ATCL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Bùi Nguyên Phúc | 21522469 | 21522469@gm.uit.edu.vn |
| 2 | Lê Hoàng Oanh | 21521253 | 21521253@gm.uit.edu.vn |
| 3 | Bùi Hoàng Trúc Anh | 21521817 | 21521817@gm.uit.edu.vn |
| 4 | Nguyễn Ngọc Trà My | 21520353 | 21520353@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-2)**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Yêu cầu 1 | 100% |
| 2 | Yêu cầu 2 | 33% |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

BÁO CÁO CHI TIẾT

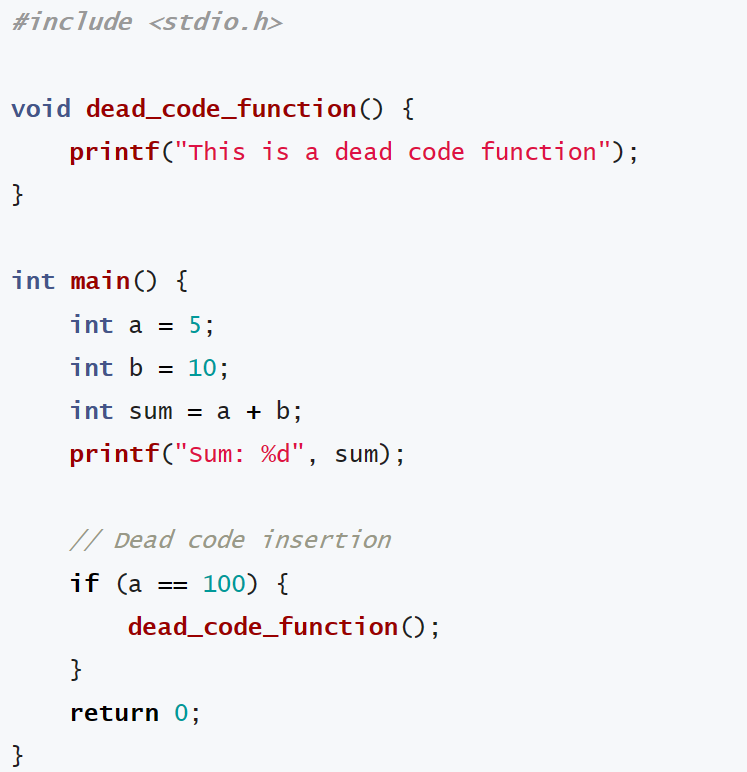
**Yêu cầu 1:** Trình bày hiểu biết của em về các kỹ thuật làm rối mã cơ bản, được liệt kê trong buổi 6. Ví dụ minh họa về các kỹ thuật đó

Dưới đây là một trình bày về các kỹ thuật làm rối mã cơ bản, cùng với ví dụ minh họa cho mỗi kỹ thuật:

**1. Dead-Code Insertion:**

- Kỹ thuật này bao gồm việc thêm vào mã nguồn các đoạn mã không hoạt động, không có tác dụng trong quá trình thực thi.

- Ví dụ minh họa:



**2. XOR:**

- XOR (hoặc phép XOR) là một phép toán logic thường được sử dụng trong mã hóa và làm rối mã bằng cách kết hợp giá trị của hai biến với nhau.

- Ví dụ minh họa:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Description automatically generated

**3. Register reassignment:**

- Kỹ thuật này liên quan đến việc thay đổi cách sử dụng các thanh ghi trong mã, làm cho việc theo dõi các giá trị và hoạt động của chúng trở nên khó khăn hơn.

- Ví dụ minh họa:

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, hàng

Description automatically generated

**4. Subroutine Reordering:**

- Kỹ thuật này là việc sắp xếp lại thứ tự các hàm hoặc các phần của mã, làm cho việc theo dõi luồng đi của chương trình trở nên khó khăn.

- Ví dụ minh họa:

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, thiết kế

Description automatically generated

**5. Instruction Substitution:**

- Kỹ thuật này thay thế một lệnh trong mã bằng một lệnh tương đương khác, làm cho việc hiểu cấu trúc và chức năng của mã trở nên khó khăn hơn.

- Ví dụ minh họa:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Description automatically generated

**6. Code Transposition:**

- Kỹ thuật này liên quan đến việc hoán đổi vị trí của các phần của mã, làm cho việc theo dõi luồng đi hoặc cấu trúc của mã trở nên phức tạp.

- Ví dụ minh họa:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Description automatically generated

**7. Code integration:**

- Kỹ thuật này kết hợp mã từ nhiều nguồn khác nhau vào một tệp duy nhất, làm cho việc phân tích và theo dõi nguồn gốc của mã trở nên khó khăn hơn.

- Ví dụ minh họa:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Description automatically generated

**8. Base64:**

- Base64 là một phương pháp mã hóa dữ liệu dựa trên 64 ký tự ASCII, thường được sử dụng để mã hóa dữ liệu nhạy cảm hoặc mã nguồn.

- Ví dụ minh họa:

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, hàng

Description automatically generated

**Yêu cầu 2:** Viết chương trình lây nhiễm virus vào tập tin thực thi (tập tin thực tin trên Windows – PE file 32 bits) có tính năng đơn giản (mục đích demo giáo dục như yêu cầu bên dưới)

Dưới đây là cách thực hiện sử dụng section .reloc trong tập tin thực thi để tiêm payload:

1. Xác định Kích thước của Payload:

- Tính toán kích thước của payload bằng cách cộng kích thước của caption và text cần hiển thị trong message box, sau đó cộng thêm một số lượng cố định (ví dụ: 0x20) để lưu trữ shellcode gọi hàm message box và nhảy về vị trí ban đầu của chương trình sau khi thực thi.

2. Mở rộng Phần Cuối của Chương Trình:

- Mở rộng phần cuối của chương trình thêm một khoảng bằng kích thước của payload tính ở bước trước.

3. Thay đổi Giá trị của Size Of Raw Data và Misc\_VirtualSize:

- Thay đổi các giá trị Size Of Raw Data và Misc\_VirtualSize của section cuối cùng của chương trình để phản ánh sự thay đổi kích thước đã thực hiện ở bước 2.

4. Tìm Địa chỉ của Hàm MessageBox:

- Tìm địa chỉ của hàm MessageBox trong thư viện Windows API.

5. Tính toán Giá trị của Entry Point Gốc:

- Tính toán giá trị của entry point gốc của chương trình.

6. Chuyển Caption và Text thành Byte:

- Chuyển caption và text cần hiển thị trong message box thành dạng byte.

7. Truyền các Giá trị vào Shellcode:

- Truyền các giá trị đã tính được (địa chỉ của hàm MessageBox, caption và text dưới dạng byte) vào shellcode.

8. Chuyển Shellcode thành Byte:

- Chuyển shellcode thành dạng byte để có thể chèn vào chương trình.

9. Chèn Shellcode vào Chương Trình:

- Chèn shellcode vào vị trí kế tiếp sau phần cuối cùng của chương trình.

10. Tăng Giá trị Size Of Image:

- Tăng giá trị Size Of Image của chương trình thêm một khoảng bằng kích thước của payload.

11. Thay đổi Entry Point của Chương Trình:

- Thay đổi entry point của chương trình để trỏ vào vị trí bắt đầu của shellcode. Khai thác sử dụng section .reloc để chèn payload. Sử dụng section .reloc để chèn payload bằng cách thực hiện các bước trên và chèn shell code vào địa chỉ kế tiếp sau phần cuối cùng của section .reloc.

Qua các bước trên, chúng ta sẽ có thể sử dụng section .reloc để tiêm payload vào một tập tin thực thi.

Ảnh có chứa văn bản, phần mềm, ảnh chụp màn hình, máy tính

Description automatically generated

---

**HẾT**

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-2)