BÀI TẬP THỰC HÀNH SỐ 2

MÔN: CƠ CHẾ HOẠT ĐỘNG CỦA MÃ ĐỘC

1 Mục tiêu

- Hiểu được cơ chế chèn mã vào file PE của viruses.
- Thực hiện chèn một đoạn mã đơn giản để hiển thị một message box khi mở file NOTEPAD.exe

2 Yêu cầu

- Cài đặt CFF Explorer để xem và chỉnh sửa các tham số trong file PE.
- Cài đặt HxD để đọc và sửa nội dung file PE.
- Cài đặt IDA Pro để kiểm tra mã hợp ngữ của chương trình muốn chèn.

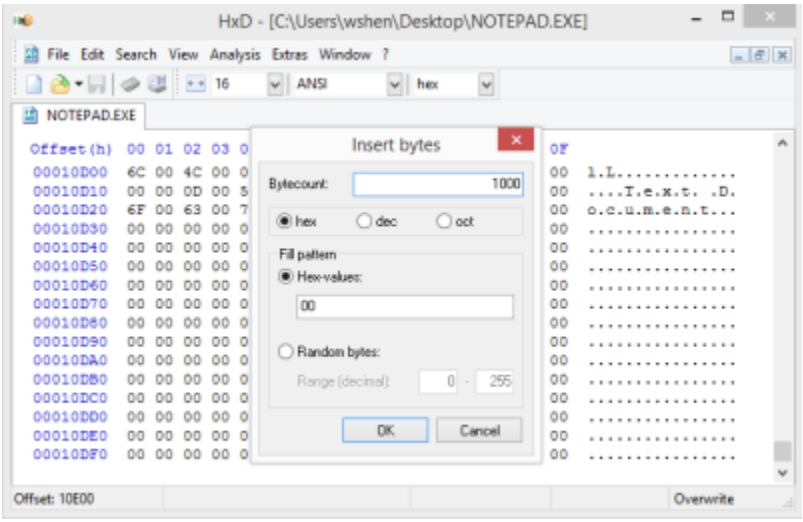
3 Hướng dẫn

Mở CFF Explorer để đọc nội dung của Notepad.exe. Nhìn vào các nội dung được hiển thị.

- Chọn Optional Header, tìm AddressOfEntryPoint trong bảng bên phải chứa giá trị 0x0000739D. Cũng trong bảng bên phải, tìm ImageBase chứa giá trị 0x01000000. Cộng 2 giá trị này lại ta được 0x0100739D. Đây chính là địa chỉ ảo (virtual address) xác định vị trí thực thi của chương trình.
- Chọn Section Header, ta thấy 3 sections: text, data và rsrc. Để đơn giản, chúng ta sẽ mở rộng rsrc và chèn code vào đây. Mỗi section có chứa 4 thông tin:
  - **VirtualSize**: kích thước của section khi được load vào bộ nhớ.
  - **VirtualAddress(VA)**: địa chỉ của section khi được load vào bộ nhớ.
  - **RawSize**: kích thước của section trong PE file.
  - **RawAddress(RA)**: địa chỉ của section trong PE file.

3.1 Tạo vùng nhớ trống trên file PE

Sử dụng HxD để mở Notepad.exe. Đặt trỏ chuột vào cuối file, chọn Edit→Insert Bytes, nhập giá trị 0x1000 (chúng ta có thể chèn nhiều hơn đối với các đoạn mã phức tạp, tuy nhiên sẽ làm tăng kích thước của PE file) và nhấn OK.



3.2 Tạo chương trình cần chèn

Để đơn giản, ta có một chương trình hiển thị MessageBox như sau:

```
#include <windows.h>

int main(int argc, char * argv[])
{
	MessageBox(NULL, L"Info", L"Code injected", MB_OK);
	return 0;
}
```

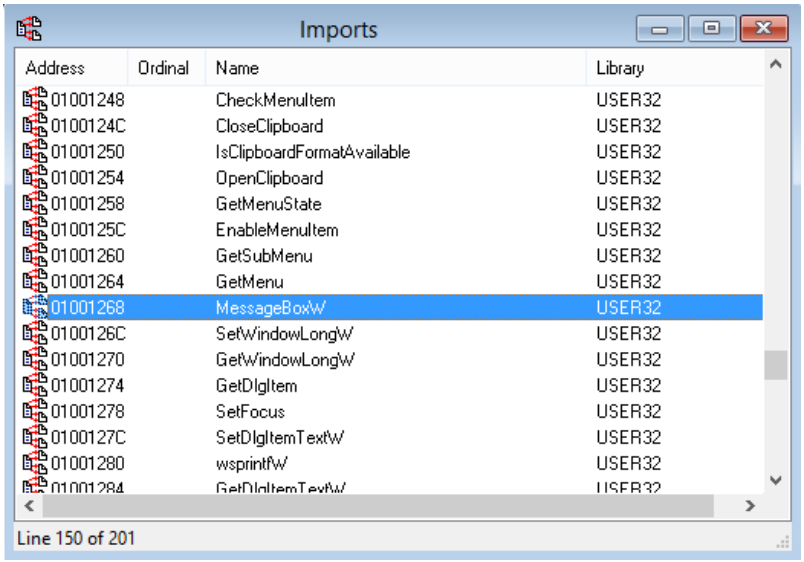
Biên dịch chương trình dưới chế độ Release, Not Using Precompiled Headers. Sử dụng IDA Pro để mở file PE và xem mã hợp ngữ của chương trình vừa biên dịch.

Về cơ bản, chương trình gồm 5 dòng lệnh (sử dụng chức năng hexview để xem mã hex của từng lệnh).

```
push 0                  ; 6a 00
push Caption            ; 68 X
push Text               ; 68 Y
push 0                  ; 6a 00
call [MessageBoxW]      ; ff15 Z
```

Để chèn đoạn code này vào Notepad.exe, ta phải đi tìm các giá trị (X, Y, Z) phù hợp.

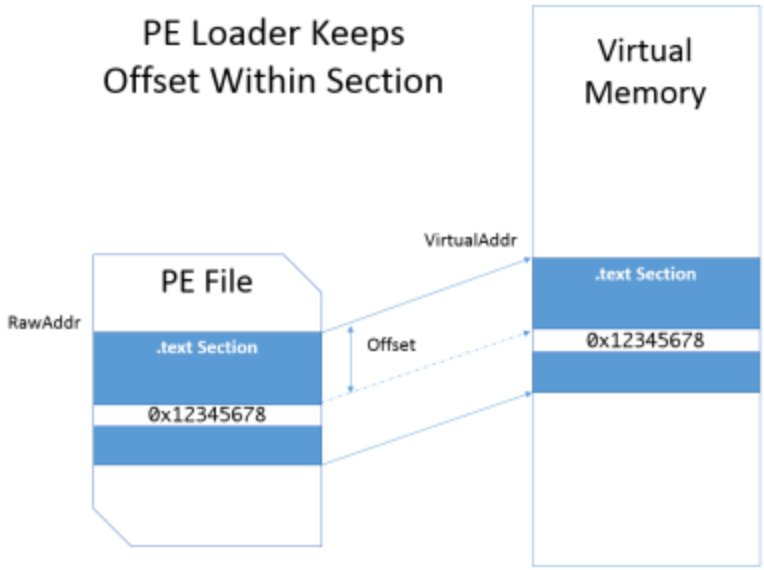
Giá trị Z chính là địa chỉ của hàm MessageBoxW được import từ thư viện USER32.dll. Trong IDA Pro, mở Notepad.exe, chọn View→Open Subviews→Imports và ta thấy địa chỉ của hàm MessageBoxW chính là 01001268.



Trong HxD, ta chọn địa chỉ 0x00011000 trong vùng nhớ đã được mở rộng (bước 3.1) để lưu trữ mã hợp ngữ, 0x00011040 để lưu trữ Caption và 0x00011060 để lưu trữ Text. Ta có thể tùy chọn những vị trí khác tùy thích.

Đối với mỗi section, loader sẽ copy section tại RA trong PE file sang bộ nhớ ảo tại VA trong khi vẫn giữ đúng offset bên trong section đó.

Offset = RA – Section RA = VA – Section VA (1)



Giá trị X có thể được tìm dựa vào công thức (1)

0x00011040 – 0x00008400 = X – 0x000B000

X = 0x00013C40

Cộng thêm ImageBase, suy ra X = 0x01013C40.

Tương tự, Y = 0x01013C60.

Như vậy, đoạn code này thực hiện chức năng như mong đợi và có địa chỉ mới là:

new_entry_point = 0x00011000 – 0x00008400 + 0x000B000 = 0x00013C00

3.3 Thiết lập lệnh quay về AddressOfEntryPoint ban đầu

Để chương trình Notepad.exe tiếp tục được thực thi sau khi đã chạy đoạn code trên, ta cần chèn dòng lệnh quay về AddressOfEntryPoint cũ ngay sau đoạn code ở bước 3.2.

jmp relative_VA

Đối với lệnh jmp, đích đến (old_entry_point) sẽ được tính bằng cách cộng giá trị relative_VA vào thanh ghi PC khi lệnh được thực thi. Bởi vì PC luôn trỏ đến vị trí đầu của câu lệnh kế tiếp, cho nên cần phải tính 5 bytes của câu lệnh jmp nữa. Ta có công thức sau:

old_entry_point = jmp_instruction_VA + 5 + relative_VA (2)

Nếu đặt lệnh jmp sau 5 câu lệnh ở bước 3.2 thì jmp_instruction_VA = 0x01013C14.

old_entry_point = 0x0100739D chính là giá trị AddressOfEntryPoint ban đầu đã cộng ImageBase.

Suy ra, relative_VA = 0x0100739D – 5 – 0x01013C14 = 0xFFFF3784.

Đến đây, ta đã có một đoạn mã hợp ngữ hoàn chỉnh để chèn vào Notepad.exe. Các địa chỉ được biểu diễn theo thứ tự little endian (x86).

```
push 0                      ; 6a 00
push Caption                ; 68   403C0101
push Text                   ; 68   603C0101
push 0                      ; 6a00
call [MessageBoxW]          ; ff15 68120001
jmp  Origianl_Entry_Point   ; e9   8437FFFF
```

3.4 Chèn vào Notepad.exe

Sử dụng HxD để chèn đoạn mã cùng với giá trị Caption và Text vào Notepad.exe. Lưu lại file.

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00010FF0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00011000   6A 00 68 60 3C 01 01 68 40 3C 01 01 6A 00 FF 15   j.h`<..h@<..j.ÿ.
00011010   68 12 00 01 E9 84 37 FF FF 00 00 00 00 00 00 00   h...é„7ÿÿ.......
00011020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00011030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00011040   43 00 6F 00 64 00 65 00 20 00 69 00 6E 00 6A 00   C.o.d.e. .i.n.j.
00011050   65 00 63 00 74 00 65 00 64 00 00 00 00 00 00 00   e.c.t.e.d.......
00011060   49 00 6E 00 66 00 6F 00 00 00 00 00 00 00 00 00   I.n.f.o.........
00011070   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00011080   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00011090   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
000110A0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

3.5 Hiệu chỉnh các tham số trong PE header

Sử dụng CFF Explorer để thay đổi các giá trị sau:

- Trong Section Headers, thay đổi .rsrc Section Header.

Name	Virtual Size	Virtual Address	Raw Size	Raw Address
Byte[8]	Dword	Dword	Dword	Dword
.text	00007748	00001000	00007800	00000400
.data	00001BA8	00009000	00000800	00007C00
.rsrc	00009958	0000B000	00009A00	00008400

- Trong Optional Headers, tăng SizeOfImage lên 0x1000.
- Trong Optional Headers, chỉnh sửa AddressOfEntryPoint thành 0x00013C00.

Lưu lại file.

3.6 Kiểm tra kết quả

Khi thực thi Notepad.exe, một cửa sổ xuất hiện như sau: