BÁO CÁO THỰC HÀNH

**Môn học: Cơ chế hoạt động mã độc**

**Kỳ báo cáo: Buổi 05 (Session 05)**

**Tên chủ đề: Return-Oriented Programming (ROP)**

*GVHD: Nghi Hoàng Khoa*

1. **THÔNG TIN CHUNG:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Đoàn Nguyễn Quốc Huy | 18520065 | 18520065@gm.uit.edu.vn |
| 2 | Thái Hoàng Nhân | 18520326 | 18520326@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:**

Dùng lệnh file và checksec để kiểm tra file rop2: giống như bài rop-basic với NX (Non executable) 🡪 Để khai thác cần sử dụng ROP

Text

Description automatically generated with medium confidence

Sử dụng IDA đọc Pseudo code:

Trong hàm main:

Text

Description automatically generated

Trong hàm vuln():

Graphical user interface, text

Description automatically generated

Ta thấy được hàm gets() để nhập dữ liệu vào, bây giờ sử dụng gdb để debug hàm vuln:

2 break point lần lượt đặt tại hàm gets (0x080488d0) và lệnh ret (0x080488dc)

Graphical user interface, text

Description automatically generated

Bắt đầu chạy bằng lệnh run để hiểu quá trình:

Graphical user interface, text

Description automatically generated

Nhập chuỗi “aaaaaaa” thì ta thấy được chuỗi được lưu tại thanh ghi EAX với địa chỉ **0xffffd1a0**

A picture containing text, monitor, screen, flat

Description automatically generated

Địa chỉ hàm ret là **0xffffd1bc**

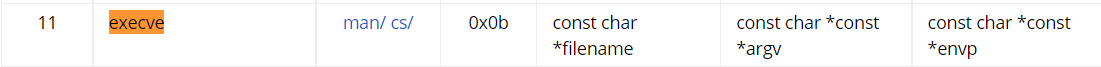
Hiệu 2 địa chỉ lại để tìm được số kí tự tối đa chèn được vào trước khi bị tràn:

Graphical user interface

Description automatically generated

* Cần nhập 28 bytes để làm tràn

Do là lệnh execve của kiến trúc 32 bit:



* eax = 0x0b,
* ebx là 1 filename, ở đây là ‘/bin/sh’
* ecx và edx lần lượt là NULL

Sử dụng ROPgadget để tìm các vị trí tương ứng để trả về các thanh ghi eax, ebx, ecx, edx

Graphical user interface, text

Description automatically generated

Thanh ghi eax trả về nằm ở địa chỉ: **0x080a8e36**

A picture containing text, scoreboard

Description automatically generated

Tương tự bài ROP-Basic: Khi tìm thanh ghi ebx thì hàm trả về cả 3 thành ghi **lần lượt theo thư tự là edx, ecx và ebx** nằm ở địa chỉ **0x0806ee91**

Tìm lời gọi hàm Interupt (int):

A picture containing text

Description automatically generated

Nằm ở địa chỉ **0x08049563**

Tìm filename /bin/sh có được sử dụng trong rop2 hay không với lệnh:

Graphical user interface

Description automatically generated

**/bin/sh không được tìm thấy trong rop2**

Chèn các đoạn payload cần thiết trước:

Graphical user interface, text, application, chat or text message

Description automatically generated

**Vậy làm thế nào để lấy được /bin/sh vào trong file rop2 và thực thi?**

* Ta sẽ đưa câu lệnh /bin/sh vào các vùng data **có thể ghi được (writable)**

Tham khảo: <https://failingsilently.wordpress.com/2017/12/14/rop-chain-shell/>

Đầu tiên tìm kiếm section có thể writable, file rop2 là một file elf (từ lệnh file ở trên), sử dụng **readelf** để đọc các section có tên:

A screenshot of a computer

Description automatically generated with medium confidence

* Ta sẽ chèn vào section **.data** (writable), ý tưởng là sẽ chèn câu lệnh **/bin/sh** vào section **.data** ở địa chỉ **0x080da060**

Sau đó cần tìm các gadget cần thiết để chèn /bin/sh:

**‘pop|ret’ | grep ‘edx’ mục đích là load thanh ghi edx ra để thanh ghi edx load section .data ra để thực hiện việc ghi vào section .data**

Text

Description automatically generated

Địa chỉ được gán lệnh trên là: **0x0806ee6b**

**xor eax, eax ; ret (trả eax về 0) với mục đích để kết thúc chuỗi**

**Text

Description automatically generated**

Địa chỉ được gán lệnh trên là: **0x08056420**

**mov dword ptr [edx], eax ; ret với mục đích để đưa chuỗi eax được gán vào vô địa chỉ của thanh ghi edx, khi này thanh ghi edx đã load section .data để hoàn tất việc ghi vào section .data.**

**A screenshot of a computer

Description automatically generated with medium confidence**

Địa chỉ được gán câu lệnh trên là: **0x08056e65**

**Lý do tại sao chọn các câu lệnh chứa các thanh ghi trên? (VD thay vì pop|ret edx thì sao lại không chọn thanh ghi khác edx):** Vì ở file rop2 câu lệnh mov dword ptr [edx], eax ; ret thì không có ptr[ebx], ptr[ecx] nên ta chọn thanh ghi edx.

Đoạn code python phục vụ việc ghi /bin/sh vào .data section:

Text

Description automatically generated

Sau khi chạy file py:

Text

Description automatically generated

Kết quả bị lỗi và không khai thác được.

Lý do?

Tham khảo: <https://resources.infosecinstitute.com/topic/stack-based-buffer-overflow-in-win-32-platform-part-6-dealing-with-bad-characters-jmp-instruction/>

Text

Description automatically generated

Theo nguồn tham khảo thì: 0x0a hay ở shellcode là \x0a tương đương với kí hiệu \n hay xuống dòng, điều này dẫn đến việc ngăn không cho đoạn code trở về sau không chạy được

Ở đầu file rop2.py:

Graphical user interface, text, application, chat or text message

Description automatically generated

Ngay câu lệnh pop eax ; ret giá trị được gán là 0x080a8e36 vậy nếu ghi thành shellcode sẽ là **\x08\x0a\x8e\x36** có **\x0a** 🡪 phần còn lại của file không chạy được 🡪 sử dụng ROPgadget với lệnh pop eax ; ret có giá trị không có bytes 0x0a:

Text

Description automatically generated

Địa chỉ **0x08056334** pop cả 3 giá trị là eax, ebx và edx

Text

Description automatically generated

Địa chỉ **0x0806ee92** pop ecx và ebx, ebx đã gán ở trên, ta gán lại giá trị tương tự

Đoạn code đầu file mới:

Text

Description automatically generated

Đoạn code chèn /bin/sh vào **section .data**: **eax** sẽ lần lượt chứa **/bin và //sh** (4 bytes mỗi lượt) và đưa từng đoạn trên vào **edx (mov\_eax\_mem\_edx\_ret)**

Text

Description automatically generated

Đoạn code chèn payload:

Text

Description automatically generated

**Nhóm 24: nc 45.122.249.68 10009**

Tổng hợp lại thứ tự đoạn code khai thác:

Các biến khởi tạo: bao gồm tiến trình, các biến pop, xor, mov,…

Text

Description automatically generated

Thêm chuỗi “/bin//sh” vào section .data, mỗi lượt lần lượt 4 bytes vì mỗi thanh ghi chỉ chứa được 4 bytes (kiến trúc x86)

Text

Description automatically generated

Text

Description automatically generated

Sau đó thêm chuỗi null để kết thúc chuỗi:

Text

Description automatically generated

Sau đó thêm payload để khai thác giống bài ROP\_Basic:

Text

Description automatically generated

Cuối cùng gửi và tương tác với process trên:

Shape

Description automatically generated with medium confidence

Chạy lệnh:

Graphical user interface, text

Description automatically generated

* flag{65941411226023053042}