BÁO CÁO THỰC HÀNH

**Môn học: Lập trình hệ thống**

**Kỳ báo cáo: Buổi 06**

**Tên chủ đề: Buffer Overflow Attack (Buffer Bomb)**

*GVHD: Đỗ Thị Hương Lan*

*Ngày báo cáo: 21/06/2022*

1. **THÔNG TIN CHUNG:**

Lớp: NT209.M21.ATCL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Trương Đình Trọng Thanh | 20520766 | 20520766@gm.uit.edu.vn |
| 2 | Trần Đức Minh | 20521617 | 20521617@gm.uit.edu.vn |

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

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Level 0 | 100% |
| 2 | Level 1 | 100% |
| 3 | Level 2 | 0% |
| 4 | Level 3 | 0% |

BÁO CÁO CHI TIẾT

## 1. Level 1

**E1.1: Vẽ stack hàm getbuf() để xác định vị trí chuỗi buf lưu input**

Hàm getbuf(): có mã assembly ở địa chỉ 0x802F2A25

Graphical user interface, application, table

Description automatically generated

Stack:

|  |  |  |
| --- | --- | --- |
|  | Ebp + 4 | Return address |
|  | Ebp |  |
|  | … |  |
| Eax | Ebp - 520 | Buf |
|  | … |  |
|  | Ebp - 532 |  |
|  | Ebp - 536 |  |
| Esp |  |  |

**E1.2: Xác định đặc điểm của chuỗi exploit nhằm ghi đè lên địa chỉ trả về của hàm getbuf:- Kết nối giữa IDAPro và Linux để kiểm tra từng dòng code**

- Chuỗi exploit cần có kích thước bao nhiêu bytes?

=> Chuỗi exploit cần có kích thước 48 bytes.

- 4 bytes ghi đè lên 4 bytes địa chỉ trả về sẽ nằm ở vị trí nào trong chuỗi exploit?

=> 4 bytes ghi đè lên 4 bytes địa chỉ trả về sẽ nằm ở vị trí 4 bytes cuối trong chuỗi exploit.

**E1.3: Xác định địa chỉ hàm smoke để làm 4 bytes ghi đè lên địa chỉ trả về.**

Table

Description automatically generated

Địa chỉ hàm smoke: 0x802F224B

**E1.4: Xây dựng chuỗi exploit với độ dài và nội dung đã xác định trước đó.**

- Chuỗi exploit dài 48 bytes (với 44 bytes rác và 4 bytes địa chỉ cần ghi đè).

- Địa chỉ của hàm smoke là 0x802F224B, byte ordering Little Endian nên ta cần viết

các byte lần lượt là 4B 22 2F 80.

- Các byte còn lại tùy ý nên ta nhập 524 bytes đầu là 00 (chỉ cần khác 0x0A) và 4

bytes cuối là 4B 22 2F 80.

Table

Description automatically generated

**E1.5: Thực hiện truyền chuỗi exploit cho bufbomb và báo cáo kết quả.**

Graphical user interface, text, application

Description automatically generated

## 2. Level 1

**E.2: Khai thác lỗ hổng để bufbomb thực thi đoạn code của fizz thay vì hàm test. Đồng thời truyền giá trị cookie của sinh viên làm tham số fizz.**

Graphical user interface, text

Description automatically generated with medium confidence

Địa chỉ hàm fizz: 0x802F2278.

- Chuỗi exploit dài 56 bytes (với 44 bytes rác ở đầu, 4 bytes địa chỉ hàm fizz + **4 bytes rác** + 4 bytes tham số đầu vào).

(Giải thích lý do cần 4 bytes rác in đậm: khi lệnh ret thực thi, con trỏ esp lúc này đang nằm ở return address của hàm getbuf, nếu gọi hàm fizz ngay sau đó thì lúc này ebp của fizz sẽ nằm tại return address (thu hồi stack frame của hàm getbuf và cấp phát cho hàm fizz), mssv (đối số) sẽ nằm ngay bên trên ebp của fizz và chương trình sẽ hiểu cookie là return address. Vì vậy nên cần thêm 4 bytes vào để vị trí nhập cookies sẽ là ebp + 8, chương trình sẽ hiểu đây là đối số của hàm fizz).

- Địa chỉ của hàm fizz là 0x802F2278, byte ordering Little Endian nên ta cần viết các byte lần lượt là 78 22 2F 80 và tham số đầu vào cookie có giá trị là 0x158DD48F (vừa lấy được ở level 0), ta cần viết 8F D4 8D 15.

- Các byte còn lại tùy ý nên ta nhập 524 bytes đầu là 00 (chỉ cần khác 0x0A), 4 bytes tiếp theo là 78 22 2F 80, 4 bytes kế nhập 01 và 4 bytes cuối cùng 8F D4 8D 15.

Table

Description automatically generated

Kết quả:

Graphical user interface, text, application

Description automatically generated

## 3. Level 2 (Bỏ)

## 4. Level 3 (Bỏ)

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