BÁO CÁO BÀI TẬP

**Môn học: Lập trình an toàn và khai thác lỗ hổng phần mềm**

**Tên chủ đề: Off-by-one & Return to Lib C**

*GVHD: Phan Thế Duy*

1. **THÔNG TIN CHUNG:**

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

Lớp: **NT521.N11.ATCL**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Vương Đinh Thanh Ngân | 20521649 | 20521649@gm.uit.edu.vn |
| 2 | Lê Minh Nhã | 20521690 | 20521690@gm.uit.edu.vn |
| 3 | Vũ Hoàng Thạch Thiết | 20521957 | 2051957@gm.uit.edu.vn |

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

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

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

BÁO CÁO CHI TIẾT

MỤC LỤC

[A. BÁO CÁO CHI TIẾT 3](#_Toc119970114)

[1. Setup environmen 3](#_Toc119970115)

[a) Tắt tính năng ngẫu nhiên địa chỉ 3](#_Toc119970116)

[b) Vulnerable program 3](#_Toc119970117)

[2. Task 4](#_Toc119970118)

[c) Finding out the addresses of libc functions 4](#_Toc119970119)

[d) Putting the shell string in the memory 5](#_Toc119970120)

[e) Exploiting the buffer-overflow vulnerability 6](#_Toc119970121)

[f) Turning on address randomization 8](#_Toc119970122)

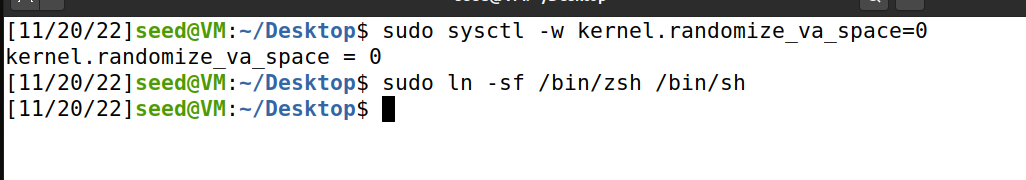
[B. TÀI LIỆU THAM KHẢO 8](#_Toc119970123)

[**YÊU CẦU CHUNG** 10](#_Toc119970124)

# BÁO CÁO CHI TIẾT

## Setup environmen

### Tắt tính năng ngẫu nhiên địa chỉ



### Vulnerable program

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Đây sẽ là chương trình chúng ta khai thác
* Chương trình có lỗ hổng buffer overflow
* Nó sẽ đọc đầu vào có kích thước 40 bytes từ 1 tệp gọi là badfile vào 1 bộ đệm có kích thước 12
* Vì hàm fread() không kiểm tra ranh giới bộ đệm, nên buffer overflow sẽ xảy ra
* Chương trình này là chương trình set uid do root sở hữu vì vậy ney61 người dùng bình thường khai thác được buffer over flow thì người dùng có thể có được root shell
* Cần lưu ý vì chương trình lấy đầu vào từ 1 tệp có tên badfile do người dùng cung cấp. Vì vậy chúng ta có thể xây dựng tệp theo cách sao cho khi 1 chương trình dễ bị tổng thương sao chép nội dung tệp vào bộ đệm của nó

## Task

### Finding out the addresses of libc functions

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Đầu tiên ta tạo 1 vulnerable program
* Sau đó biên dịch nó bằng lệnh: gcc -fno-stack-protector -z noexecstack -o retlib retlib.c
* Sau khi nó được compile thì ta cần đổi chủ của file thành root dùng command sudo chown root retlib và sau đó cấp quyền chó nó có thể thực thi được bằng lệnh chmod 4755 retlib



* Sau đó ta tạo 1 file tên là badfile có thể để trống cũng được

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Sau đó ta gdb để để xem chương trình retlib
* Ta cần tìm địa chỉ của system và exit vì trong return to libc ta cần nhảy đến 1 số code đã có sẵn trong chương trình

### Putting the shell string in the memory

- Ta cần nhảy vào 1 hàm có sẵn trong hệ thống và yêu cầu nó thực thi 1 lệnh bất kỳ

- Vì chúng ta cần mổ shell nên ta sẽ cần system() thực thi lệnh “/bin/sh”

- Vì vậy chúng ta cần đặt /bin/sh vào bộ nhớ và biết địa chỉ của nó để có thể truyền đến hàm system()

- Ta có thể thiết lập biến môi trường mới tên là MYSHELL and để nó chứa chuỗi /bin/sh bằng lệnh export MYSHELL=/bin/sh

- Ta sẽ dùng địa chỉ của MYSHELL như là đối số để hàm system() gọi đến

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Sau khi biên dịch và chạy chương trình trên ta sẽ có được địa chỉ của /bin/sh

Ảnh có chứa văn bản

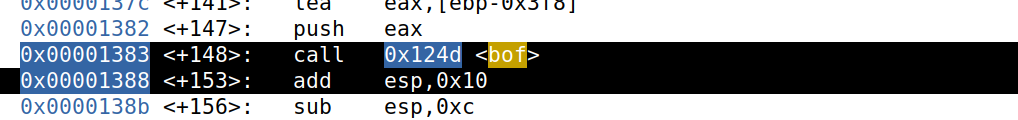
Mô tả được tạo tự động

### Exploiting the buffer-overflow vulnerability

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Bên trên là chương trình exploit.c
* Ta đã có được địa chỉ của system(), exit() và /bin/sh
* Ta sẽ đưa những địa chỉ này vào chương trình exploit.c
* Giờ ta cần tìm giá trị của X, Y, Z
* Ta sẽ dùng gdb để debug



* Sau khi ta disassemble main thì ta thấy có hàm bof và sau khi gọi bof thì địa chỉ trả về sẽ là 0x00001388

Ảnh có chứa bàn

Mô tả được tạo tự động

* Ta sẽ xem qua hàm bof ta cần overwirte địa chỉ trả về của ta với function string copy

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Sau khi đặt break point ta thấy nó lấy 2 argument và argument 1 sẽ là địa chỉ bắt đầu của stack

Ảnh có chứa văn bản

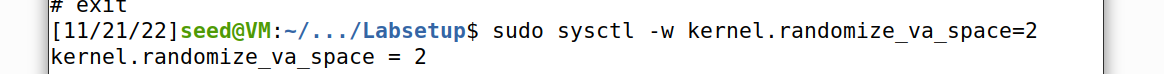
Mô tả được tạo tự động

* Ta in ra giá trị hex và ta thấy giá trị trả về ở trên và khoảng cách 28 byte nên Y=28 để có overwirte return address với system\_addr
* Sau khi bof return thì nó sẽ nhảy tới system đã bị ghi đè
* Và ta cần đặt exit function ngay sau system vì sau khi nó thực thi xong function nó sẽ cố nhảy tới function tiếp theo và tan đặt ngay sau nếu không nó sẽ xảy ra lỗi segmentaion fault

Ảnh có chứa văn bản

Mô tả được tạo tự động

### Turning on address randomization

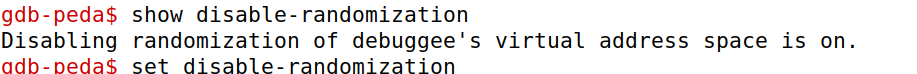


* Ta bật tính năng random địa chỉ và thực hiện lại cuộc tấn công

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Ta thấy địa chỉ thay đổi và báo lỗi segmentation fault



* Sau khi bật tính năng randomize thì ta xem lại địa chỉ của hàm system và exit

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Ta thấy nó vẫn không đổi
* Nhưng khi thực hiện tấn công thì các giá trị X, Y, Z không đổi chỉ có địa chỉ của nó thì đổi

# TÀI LIỆU THAM KHẢO

---

***Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này***

# **YÊU CẦU CHUNG**

* Sinh viên tìm hiểu và thực hiện bài tập theo yêu cầu, hướng dẫn.
* Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
* Sinh viên báo cáo kết quả thực hiện và nộp bài.

**Báo cáo:**

* File .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
* Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
* Đặt tên theo định dạng: [Mã lớp]-ExeX\_GroupY. (trong đó X là Thứ tự Bài tập, Y là mã số thứ tự nhóm trong danh sách mà GV phụ trách công bố).

*Ví dụ: [*NT101.K11.ANTT*]-Exe01\_Group03.*

* Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
* Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm bài nộp.
* Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

**Đánh giá**:

* Hoàn thành tốt yêu cầu được giao.
* Có nội dung mở rộng, ứng dụng.

*Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.*

**HẾT**

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