# BÁO CÁO THỰC HÀNH LẬP TRÌNH HỆ THỐNG LAB4

# Nhóm 12:

22520825 Nguyễn Đức Luân

22520661 Vũ Ngọc Quốc Khánh

22521110 Đoàn Hoàng Phúc



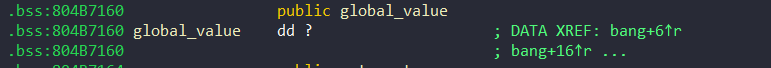
Quan sát hàm bang bằng IDA pro :

A screenshot of a computer program

Description automatically generated

Để in ra được dòng chữ như yêu cầu thì biến global phải bằng với giá trị cookie nhập vào. Ta kiểm tra hai biến này :  
A black background with white text

Description automatically generated



Ta thấy rằng hai biến trên đều là biến toàn cục chưa được khởi tạo giá trị(vùng segment .bss). Vậy cách để giải quyết vấn đề là thay đổi giá trị của biến global\_value theo giá trị của cookie là được. Do biến global\_value không thể ghi đè trực tiếp thông qua lỗi Buffer overflow vì vùng nhớ của biến này nằm thấp hơn so với stack. Chính vì vậy chúng ta sẽ thực hiện việc truyền mã thực thi thông qua buffer để thay đổi giá trị của global\_value đồng thời nhảy vào hàm bang.

Đầu tiên ta phải xác định được địa chỉ bắt đầu của chuỗi buffer ta nhập vào chương trình. Việc xác định cũng rất đơn giản, ta có thể làm theo cách trong file hướng dẫn thực hành, tuy nhiên lần này ta sẽ thực hiện theo một cách khác là dùng PwnGDB để xác định vị trí:

Ta thực hiện câu lệnh sau : gdb ./bufbomb

A screen shot of a computer program

Description automatically generated

Tiếp đến thực hiện lệnh sau:

A computer screen with text

Description automatically generated

Kết quả ta được như sau:

A screen shot of a computer program

Description automatically generated

Lúc này nhập lệnh “n” đến khi mũi tên màu xanh lục trỏ đến lệnh call Gets. Nhập tiếp n để nhập input:

A screenshot of a computer

Description automatically generated

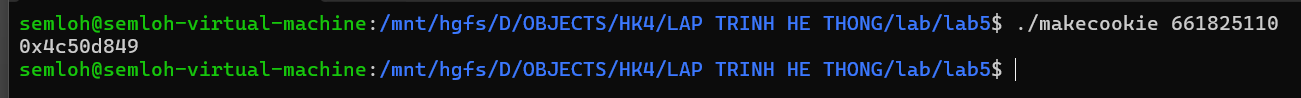
Lúc này trên giao diện của gdb xuất hiện thông tin như sau:

A computer screen with text

Description automatically generated

Ta thấy chuỗi input đã nhập được lưu bắt đầu tại địa chỉ 0x556833de. Đây cũng chính là địa chỉ mà ta cần tìm.

Tiếp đến ta xác định giá trị của cookie thông qua tên user được tạo bằng mssv của thành viên trong nhóm là 661825110. Dùng chương trình makecookie để lấy giá trị cookie:\



Vậy giá trị cookie là 0x4c50d849

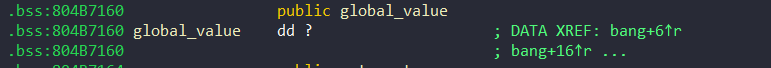
Tiếp đó ta cần xác định địa chỉ của hàm bang thông qua IDA:

A screen shot of a computer

Description automatically generated

Vậy địa chỉ của hàm bang sẽ là 0x804B2259

Cuối cùng xác định địa chỉ của biến global\_value là 0x804B7160



Sau khi có đủ các thông tin cần thiết ta tạo file shellcode như sau:

A black background with purple numbers

Description automatically generated

Thực hiện biên dịch và dump ra byte code:  
gcc -m32 -o shellcode.o -c shellcode.s

objdump -D shellcode.o

A screenshot of a computer

Description automatically generated

Đưa toàn bộ byte code vào file txt. Lưu ý phải kèm đủ 54 byte cùng với 4 byte return address là địa chỉ bắt đầu lưu của buffer. Ta được kết quả sau:

A screen shot of a computer

Description automatically generated

Chạy chương trình và kiểm tra kết quả:

A black screen with blue text

Description automatically generated

A close up of a text

Description automatically generated

Xác định được các yếu tố bị thay đổi (do Buffer overflow) và phục hồi nó. Theo dõi hàm

test:

A computer screen shot of a program

Description automatically generated

Theo luồng hoạt động bình thường của chương trình thì ta có nhập cái gì đi nữa thì

hàm getbuf() sẽ trả về 1 và v2 != cookie → In ra dòng chữ “Dud: getbuf returned …”

Biến v1 là kết quả trả về của hàm uniqueval(). Ta xem thử hàm này :

A computer screen shot of a program code

Description automatically generated

Kết quả trả về là một số ngẫu nhiên được tạo thông qua seed là biến v0. Do dùng một seed là biến v0 không thay đổi giá trị nên hàm này sẽ trả về các giá trị giống nhau sau mỗi lần chạy. Chính vì lý do đấy nên biến v0 được dùng để kiểm tra xem chương trình có bị lỗi do hàm getbuf ghi đè quá lố hay không. Nếu có thì sẽ in ra dòng chữ. “Sabotaged!: the stack has been corrupted”.

Vậy ta mong muốn quay về hàm test và gán v2 == cookie (để chứng minh là quay về hàm test thành công ) và thực hiện đến khi in ra “Boom!: getbuf returned … “ . Vậy công việc shellcode lần này là:

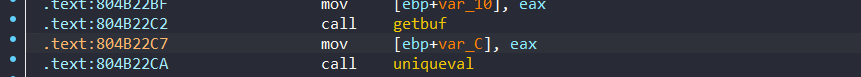
+Gán giá trị cookie trả về

+Khôi phục trạng thái thanh ghi/bộ nhớ bị thay đổi của hàm mẹ (test)

+Đẩy địa chỉ trả về đúng vào stack(câu lệnh thực thi tiếp theo của test)

+Thực thi lệnh ret và trở về hàm test

Đâu tiên giá trị của cookie đã được xác định ở bài trên nên ta sẽ bỏ qua, tiếp đến là khôi phục trạng thái thanh ghi/bộ nhớ của hàm mẹ test (khôi phục ebp của hàm test sau khi bị ghi đè vởi buffer) bằng lệnh shellcode, địa chỉ lệnh tiếp theo của test là :0x804B22C7



Ta có file shellcode như sau:

A screenshot of a computer

Description automatically generated

Xem mã assembly sau để dễ hiểu hơn:

A screen shot of a computer

Description automatically generated

Thông qua hình trên ta xác định được giá trị ebp của hàm test được xác định bằng giá trị của esp cộng với 0x18

Biên dịch lại chương trình thành file object và dump ra byte code:

**A screen shot of a computer

Description automatically generated**

Đưa byte code vào file txt và thực hiện như bài trên. Ta thu được file như sau:

A black screen with blue text

Description automatically generated

Kiểm tra :

A black screen with blue text

Description automatically generated