Level 2:

* Level này yêu cầu thực thi hàm bang. Hàm này so sánh giá trị cookie với một biến toàn cục global\_value được gán bằng 0. Nếu 2 giá trị này bằng nhau thì sẽ khai thác được thành công. Vì giá trị này không nằm trên stack nên không thể thực hiện ghi đè giá trị.
* Ta sẽ tìm địa chỉ của biến global value bằng IDAPro

Jump->Jump by name->Search global\_value



* Ta thấy nó nằm tại địa chỉ 0x80270160
* Ta cần tìm địa chỉ trả về mới sẽ là địa chỉ bắt đầu lưu chuỗi exploit trong stack. Địa chỉ này chỉ xác định trong lúc chương trình chạy nên ta cần debug breakpoin ở getbuf để xem vị trí chính xác của buf

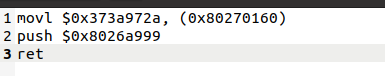
Ảnh có chứa văn bản

Mô tả được tạo tự động

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Ta thấy ebp=0x556836c0 ta dùng giá trị ebp này để tính tiếp vị trí của thể của chuỗi buf bằng cách trừ đi 0x28 = 0x55683698

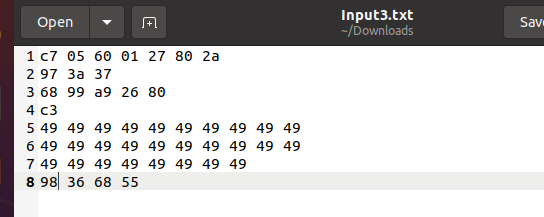


* Ta biết được địa chỉ hàm cần thực thi là 0x8026a999 bằng IDAPro
* Sau đó ta sẽ viết chuỗi exploit tương ứng để truyền cookie vào ô nhớ của biến toàn cục.
* Ta sẽ dùng objdump để chuyển chuỗi exploit về thành những byte code để đưa vào bufbomb

Ảnh có chứa văn bản, màn hình, ảnh chụp màn hình, đóng

Mô tả được tạo tự động

* Chuỗi exploit sẽ là



* Kết quả khu truyền chuỗi exploit vào bufbomb

Ảnh có chứa văn bản

Mô tả được tạo tự động

Level 4:

Như những level trc ta cần chuỗi exploit dài ít nhất 48 bytes, trong đó các byte từ 45 đến 48 chứa địa chỉ trả về trong đó nội dung exploit thực hiện các công việc sau:

* Gán cookie tương ứng giá trị trả về: movl $0x373a972a, %eax
* Khôi phục trạng thái bị gián đoạn là %ebp của hàm gọi getbuf – test.

Ảnh có chứa bàn

Mô tả được tạo tự động

* Từ đoạn code của hàm test ta có %ebp = %esp + 24 = %esp + 0x18
* Đoạn code tương ứng để khôi phục %ebp:

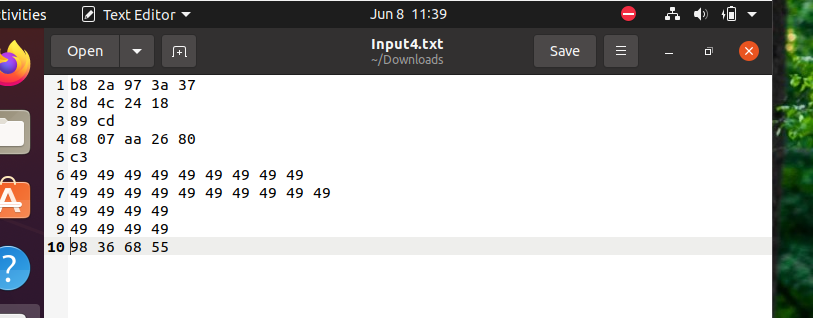
Lea $0x18(esp), %ecx

Mov %ecx, %ebp

* Đưa giá trị trả về đúng vào stack và thực hiện lệnh ret để trở về test
* Từ đoạn mã assembly trên ta cũng có được địa chỉ trả về đúng sau khi gọi getbuf là 0x8026AA07
* Dump ra ta có byte code sau
* Ảnh có chứa văn bản

  Mô tả được tạo tự động

Chuyển qua chuỗi exploit



* Và kết quả thu được:
* Ảnh có chứa văn bản

  Mô tả được tạo tự động