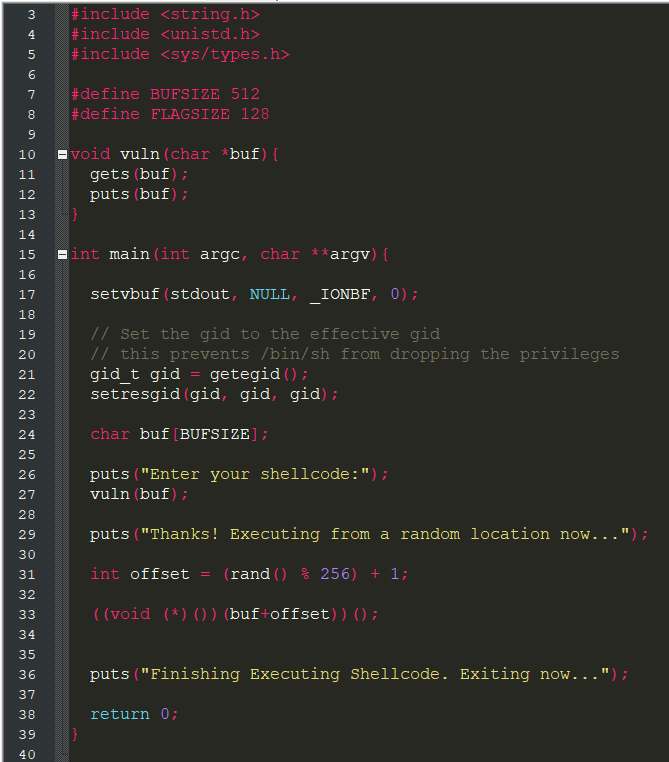
Slippery-shellcode

1. Đọc hiểu source code:



Chương trình gồm 2 hàm chính:

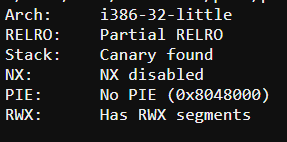
* Vuln() : nhận tham số là con trỏ buf, nhập vào buf bằng gets(buf) , sau đó in ra buf.
* Main() : gọi hàm vuln( ) . Sau đó random 1 giá trị offset nằm trong khoảng từ (1 -> 256) , đổi ra hexa là từ 0x1 đến 0xff.

Tiếp theo là thực thi lệnh ở buf + offset.

1. Phân tích lỗi và khai thác:
   1. Hướng giải:

Bài này không cho chúng ta hàm flag nữa, nên chắc là gọi shell để lấy flag rồi.

* 1. Checksec binary file:



* 1. Phân tích lỗi:

Trong hàm vuln( ) , hàm gets(buf) không kiểm tra số lượng kí tự được phép nhập vào nên gây ra lỗi buffer overflow ( tràn bộ đệm) . Với lỗi này, ta có thể lợi dụng làm tràn stack, overwrite địa chỉ return tại ebp+4 của vuln để làm gì đó. Nhưng đó là cách giải không đúng ý tác giả, chúng ta sẽ cố gắng đi theo hướng giải của bài này, chừng nào bí thì mới làm hướng mở rộng.

* 1. Khai thác:

Sau khi đọc code, ta để ý rằng sau khi nhập buf thì chương trình sẽ thực thi những gì ta đã nhập ở địa chỉ [buf + offset] , vậy nếu ta nhập vào shellcode thì khi chương trình gọi sẽ gọi shell cho chúng ta. Nhưng offset là 1 giá trị random, làm sao biết được nó bao nhiêu mà tính toán.

|  |  |  |
| --- | --- | --- |
| Buf | offset | Buf+offset |

Nhập ở đây

Thực thi ở đây

Đọc kĩ lại code 1 xíu, ta sẽ thấy offset luôn < 257 . Vậy nếu buf của ta có format như sau:

Buf = “\x90”\*257+ shellcode

Với lệnh “\x90” trong asm nghĩa là lệnh nop, nếu gặp lệnh này thì chương trình sẽ không làm gì cả và đi đến lệnh tiếp theo.

|  |  |  |
| --- | --- | --- |
| Buf | offset | Buf+offset |

|  |  |  |
| --- | --- | --- |
| “\x90” | “\x90” | shellocde |

Thì dù offset có bao nhiêu đi chăng nữa, shellcode của chúng ta luôn được thực thi vì offset nhỏ hơn 257 nên buf + offset luôn chỉ nằm trong đoạn “\x90”.

Stack:

Nop  
 Nop  
 .

.

.

Shellcode

Thế là lên được shell.