2

Lab

Simple Worm

Thực hành Cơ chế hoạt động của Mã độc

Lưu hành nội bộ 2024

A. TỔNG QUAN

A.1 Mục tiêu

- Tìm hiểu Buffer Overflow
- Khai thác lỗ hổng Buffer Overflow trên máy bị lỗ hổng
- Khai thác lỗ hổng Buffer Overflow từ xa
- Tạo Simple Worm

A.2 Thời gian thực hành

- Tại lớp 5 tiết.
- Tại nhà 5 tiết.

B. CHUẨN BỊ MÔI TRƯỜNG

- 2 máy chủ Ubuntu Server 32 bits (một máy đóng vai trò máy chủ, một máy đóng vai trò client để khai thác lỗ hổng trên máy chủ)

  P/s:

  - *Ubuntu 14.04 32 bits*
  - *sysctl -p on server without reboot*
  - *add 5 byte to return address*
  - *-m32 when build*

- Cài đặt những gói ứng dụng hỗ trợ trong quá trình thực thi và debug code (lưu ý chạy với quyền root):

  – Cài đặt GCC (để biên dịch code)

  ```
  sudo apt-get update
  sudo apt-get install gcc
  ```

  – Cài đặt GDB (để debug code)

  ```
  sudo apt-get install libc6-dbg gdb valgrind  libc6-dev-i386
  ```

  – Tắt những thông số trên máy thực thi code để vô hiệu quá tính năng bảo vệ lỗ hổng Buffer Overflow

  ```
  sudo bash -c 'echo 0 > /proc/sys/kernel/randomize_va_space'
  ```

  – Khi thực thi chương trình, chúng ta truyền thêm những tham số như sau:

  ```
  gcc -mpreferred-stack-boundary=2 -m32 -z execstack -fno-stack-protector -o vul_server vul_server.c
  ```

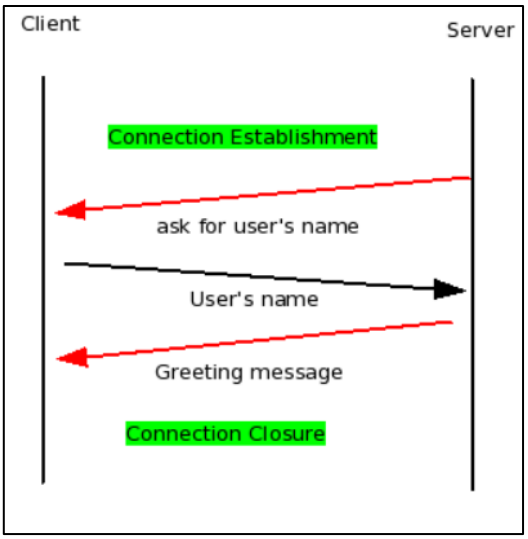
C. THỰC HÀNH

C.1 Local Buffer Overflow

Đọc cẩn thận tập tin *"stack_smashing"*. Chạy từng ví dụ để hiểu cơ chế khai thác lỗi buffer overflow **(các bạn chạy tới exploit3.c)** .

C.2 Remote Buffer Overflow

1. Trong trường hợp này chúng ta có máy chủ đang mở sẵn cổng 5000 để chờ client kết nối (biên dịch và thực thi tập tin vul_server.c trên máy chủ khi đó máy chủ sẽ mở cổng 5000 và chờ client kết nối).



Hình 1.

2. Trên máy chủ, chúng ta thao tác:

```
gcc -mpreferred-stack-boundary=2 -z execstack -fno-stack-protector -o vul_server vul_server.c
```

```
ubuntu@ubuntu:~$ gcc -mpreferred-stack-boundary=2 -m32 -z execstack -fno-stack-protector -o
 vul_server vul_server.c
vul_server.c: In function 'main':
vul_server.c:51:46: warning: implicit declaration of function 'atol' [-Wimplicit-function-d
eclaration]
   srv.sin_port = htons( (unsigned short int) atol(argv[1]));
                                              ^
vul_server.c:71:32: warning: implicit declaration of function 'inet_ntoa' [-Wimplicit-funct
ion-declaration]
       printf("client from %s", inet_ntoa(cli.sin_addr));
                                ^
vul_server.c:71:14: warning: format '%s' expects argument of type 'char *', but argument 2
has type 'int' [-Wformat=]
       printf("client from %s", inet_ntoa(cli.sin_addr));
              ^
vul_server.c:74:7: warning: implicit declaration of function 'close' [-Wimplicit-function-d
eclaration]
       close(c);
       ^
```

```
./vul_server 5000
```

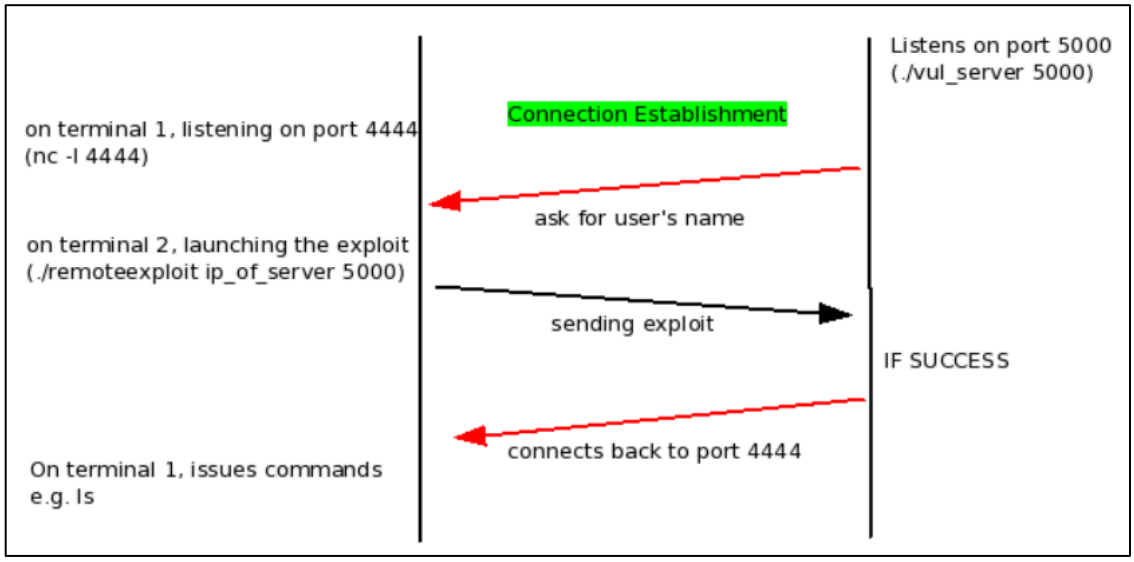
4. Trên máy client, chúng ta thao tác:

telnet *IP_Server* 5000

```
ubuntu@ubuntu:~$ telnet [redacted].20.189 5000
Trying [redacted].20.189...
Connected to [redacted].20.189.
Escape character is '^]'.
My name is: khoa
Hello :khoa, welcome to our siteConnection closed by foreign host.
```

Sau khi hoàn thành đoạn những thao tác trên. Chúng ta chuyển sang trường hợp phức tạp hơn.

5. Trên máy chủ sau khi biên dịch và thực thi *./vul_server 5000*. Máy chủ sẽ mở cổng 5000 chờ client kết nối tới. Tại client, chúng ta sẽ biên dịch và thực thi *./remoteexploit IP_Server 5000*. Nếu khai thác thành công lỗ hổng trên máy chủ, máy chủ sẽ tự động kết nối ngược lại client theo cổng *4444* (cổng 4444 là cổng mặc định). Để có thể kiểm soát máy chủ từ xa, trường hợp này chúng ta dùng phần chương trình netcat. Toàn bộ qui trình tấn công diễn ra như sau:



Hình 2.

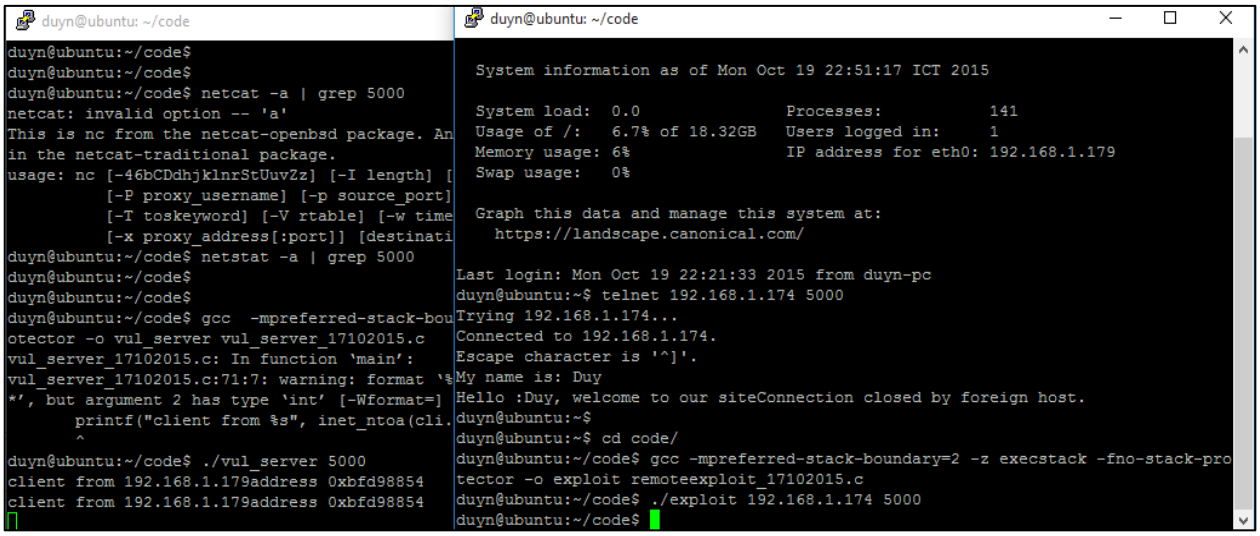
Lưu ý: để có thể thực hiện thành công cuộc tấn công này. Trong tập tin remoteexploit.c, chúng ta cần phần điều chỉnh địa chỉ IP cho phù hợp và địa chỉ trả về con trỏ hàm trong stack cho chính xác

Bước 1. Trên terminal 1 client - mở công 4444

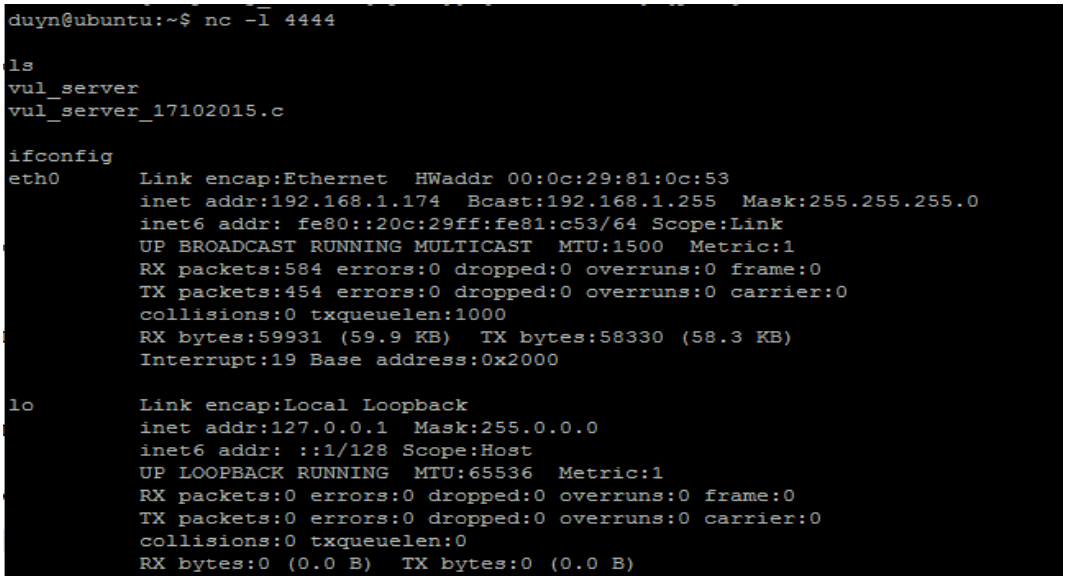
```
duyn@ubuntu:~$ nc -l 4444
```

Bước 2. Thực thi

- Trên máy chủ: ./vul_server 5000
- Trên terminal 2 client: ./exploit 192.168.1.174 5000



- Sau khi khai thác thành công, trên terminal 1 client mà chúng ta thấy chính là terminal trên máy chủ.



> **Yêu cầu 1** **Hoàn thành mục C.1 và C.2.**

C.3 Simple Worm

> **Yêu cầu 2** **Nhiệm vụ cuối cùng trong bài lab này là sau khi Worm đã lây nhiễm thành công trên máy chủ thứ 1 và attacker khai thác thành công thì Worm sẽ tự động đây lây nhiễm sang máy chủ khác và tự động thực thi ./vulner_server 5000**

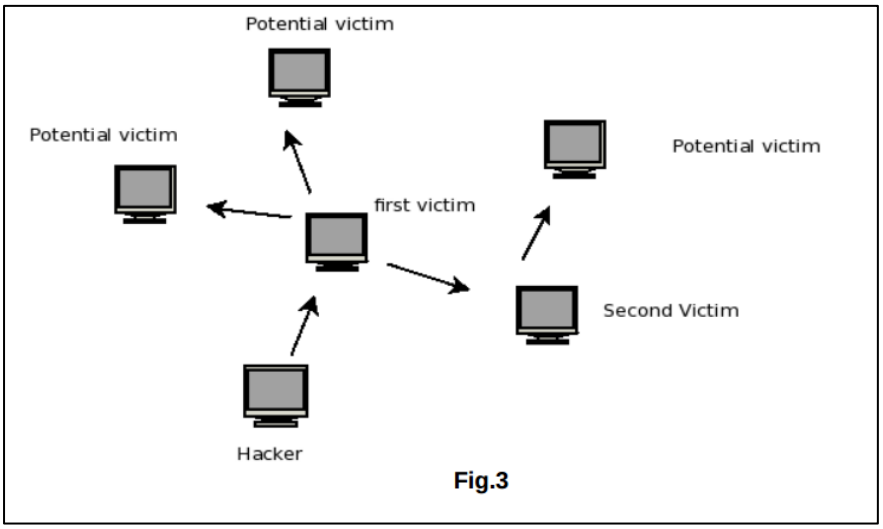


Fig.3

D. YÊU CẦU

- Sinh viên tìm hiểu và thực hành theo hướng dẫn theo nhóm từ 1-2 sinh viên.
- Báo cáo kết quả chi tiết những việc (**Report**) đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có), video demo (điểm cộng).

Báo cáo:

- Làm báo cáo trên file mẫu.
- File **.PDF**. Tập trung vào nội dung, không mô tả lý thuyết.
- Trong file báo cáo yêu cầu **ghi rõ** nhóm sinh viên thực hiện.
- Đặt tên theo định dạng: [Mã lớp]-Lab4_MSSV1-MSSV2.pdf
  Ví dụ: [NT230.N2X.ATCL]-Lab4_2052xxxx-2052yyyy.pdf
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá: Sinh viên hiểu và tự thực hiện được bài thực hành. Khuyến khích:

- Chuẩn bị tốt và đóng góp tích cực tại lớp.
- Có nội dung mở rộng, ứng dụng trong kịch bản phức tạp hơn, có đóng góp xây dựng bài thực hành.

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

-HẾT-