Solving Remote Challenge

Mungsul

Local and remote

- Local

Being able to solve, after connection to challenge server with terminal.

You can get IP, Port in description

- Remote

IP, Port, Binary

Difference

fd - 1 pt [writeup]

Mommy! what is a file descriptor in Linux?

* try to play the wargame your self but if you are ABSOLUTE beginner, follow this tutorial link: https://www.youtube.com/watch?v=blAxTfcW9VU

ssh fd@pwnable.kr -p2222 (pw:guest)

pwned (14380) times. early 30 pwners are : go_for_it

Flag? : auth

Local Challenge

brain fuck - 150 pt [writeup]

I made a simple brain-fuck language emulation program written in C.
The [ ] commands are not implemented yet. However the rest functionality seems working fine.
Find a bug and exploit it to get a shell.

Download : http://pwnable.kr/bin/bf
Download : http://pwnable.kr/bin/bf_libc.so

Running at : nc pwnable.kr 9001

pwned (874) times. early 30 pwners are : setuid0

Flag? : auth

Remote Challenge

Connecting to server

- Nc(netcat)

Netcat is tool that help you to connect any opened server.

Command : nc [SERVER ADDR] [Port]

Ex) nc 192.168.0.1 80

Connecting to server

선택 C:₩WINDOWS₩system32₩cmd.exe

```
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

C:₩Users₩Mungsul>nc
Cmd line: wrong^C
C:₩Users₩Mungsul>nc 192.168.0.1 80
GET / HTTP/1.0

HTTP/1.0 200 OK
Date: Tue, 23 Jan 2018 23:55:47 GMT
Server: Httpd/1.0
Connection: close
Content-Length: 112
Last-Modified: Mon, 25 Dec 2017 21:44:10 GMT
Content-Type: text/html

<html>
<head>
<meta http-equiv=refresh content="0; URL=login/login.cgi">
<title></title>
<body>
</body>
</html>

C:₩Users₩Mungsul>
```

Server and client

```
win32virus@ubuntu:~$ nc -lvp 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from localhost 47656 received!
qweqwe
```

server

```
win32virus@ubuntu:~$ nc localhost 4444
qweqwe
```

client

Network Programming

```c
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include<netinet/in.h>

int main()
{
    char buf[] = "Hello Client\n";
    char rbuf[256] = {0,};
    struct sockaddr_in server_addr, client_addr;
    int server_fd, client_fd, len;
    server_fd = socket(AF_INET,SOCK_STREAM,0);
    memset(&server_addr,0,sizeof(server_addr));;
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = htonl(INADDR_ANY);
    server_addr.sin_port = htons(4444);
    bind(server_fd,(struct sockaddr *)&server_addr, sizeof(server_addr));
    listen(server_fd,5);

    len = sizeof(client_addr);
    client_fd = accept(server_fd, (struct sockaddr *)&client_addr, &len);
    send(client_fd, buf, strlen(buf),0);
    recv(client_fd, rbuf, 256, 0);
    printf("%s\n",rbuf);
    close(client_fd);
    close(server_fd);
}
```

Server c code

```c
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include<netinet/in.h>

int main()
{
    char buf[] = "Hello Server!\n";
    char rbuf[256] = {0,};
    int client_len;
    int client_fd;

    struct sockaddr_in client_addr;
    client_fd = socket(AF_INET,SOCK_STREAM,0);
    client_addr.sin_family = AF_INET;
    client_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
    client_addr.sin_port = htons(4444);

    client_len = sizeof(client_addr);

    connect(client_fd, (struct sockaddr *)&client_addr, client_len);
    recv(client_fd, rbuf, 256, 0);
    printf("%s\n",rbuf);
    send(client_fd, buf, strlen(buf),0);
    close(client_fd);
}
```

Client c code

Network Programming

- Server flow

Make socket -> bind -> listen -> accept (wait until client is connected)

- Client flow

Make socket -> connect

Network Programming

- Server side

Output(send, write, printf etc..) is to send data to client.

Input(recv, read, scanf etc..) is to receive data from client.

- Client side

Output(send, write, printf etc..) is to send data to server.

Input(recv, read, scanf etc..) is to receive data from server.

Network Programming

```
win32virus@ubuntu:~/whoisLec/seminar$ ./server
Hello Server!

win32virus@ubuntu:~/whoisLec/seminar$
```

server

```
win32virus@ubuntu:~/whoisLec/seminar$ ./client
Hello Client

win32virus@ubuntu:~/whoisLec/seminar$
```

client

In python

```python
#!/usr/bin/python

from socket import *

s = socket(AF_INET,SOCK_STREAM)
s.bind(("0.0.0.0",4444))
s.listen(5)
conn, addr = s.accept()
print addr
conn.send("Hello Client")
print conn.recv(1024)
conn.close()
s.close()
```

server

```python
#!/usr/bin/python

from socket import *

s = socket(AF_INET,SOCK_STREAM)
s.connect(("localhost",4444))
print s.recv(1024)
s.send("Hello Server")
s.close()
```

client

Simple Remote Challenge

```python
#!/usr/bin/python

import SocketServer
import random
import sys

class MyTCPHandler(SocketServer.BaseRequestHandler):

    def handle(self):
        s = self.request
        for i in xrange(0,200):
            num1 = random.randrange(0,10000)
            num2 = random.randrange(0,10000)
            answer = num1 + num2
            s.send("{} + {} = ".format(num1,num2))
            d = s.recv(100)
            d = d.replace("\n","")
            if d != str(answer):
                s.send("Wrong")
                s.close()
                return
        with open("flag.txt","rb") as f:
            dd = f.read()
        s.send(dd)
        s.close()


if __name__ == '__main__':

    server = SocketServer.TCPServer(("0.0.0.0",int(sys.argv[1])), MyTCPHandler)
    server.serve_forever()
```

Random number addition, 200 times

nc badcoffee.kr 8888

Q&A