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

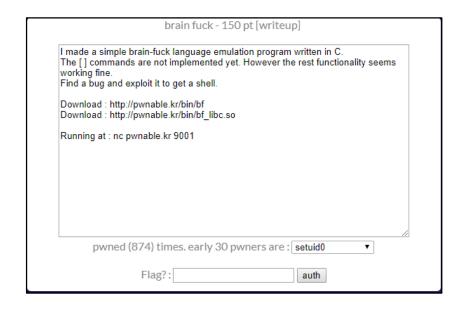
Remote

IP, Port, Binary

Difference



Local Challenge



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

win32virus@ubuntu:~\$ nc localhost 4444 qweqwe

server client

```
8 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);
```

```
2 #include
6 #include<netinet/in</pre>
8 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);
```

Server c code

Client c code

Server flow

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

Client flow

Make socket -> connect

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.

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

win32virus@ubuntu:~/whoisLec/seminar\$

win32virus@ubuntu:~/whoisLec/seminar\$./client

server client

Hello Client

In 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

```
1 #!/usr/bin/python
2
3 from socket import *
4
5 s = socket(AF_INET,SOCK_STREAM)
6 s.connect(("localhost",4444))
7 print s.recv(1024)
8 s.send("Hello Server")
9 s.close()
```

client

Simple Remote Challenge

```
.<mark>#</mark>!/usr/bin/python
 import SocketServer
4 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()
               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