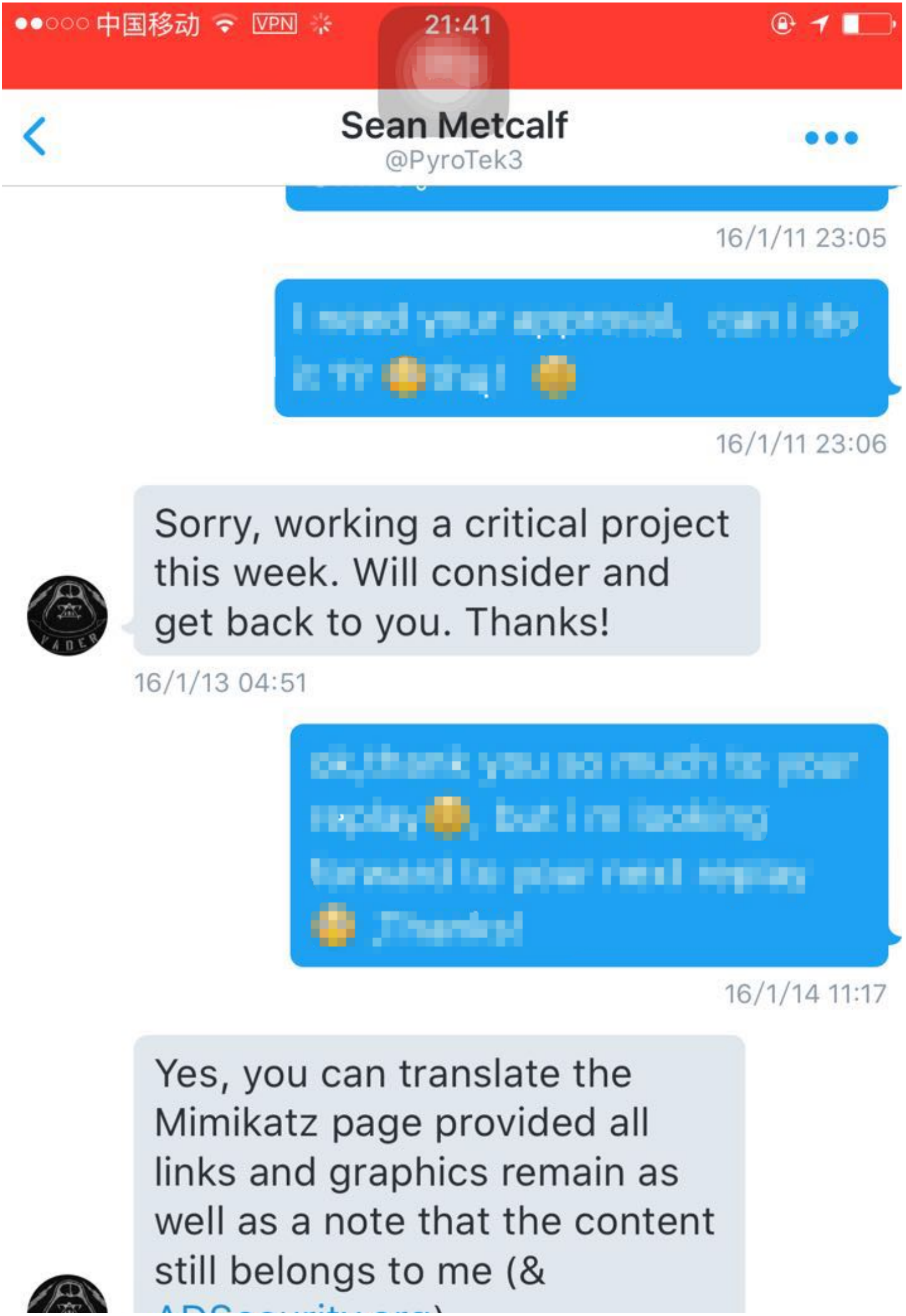
Mimikatz 非官方指南和命令参考_Part1

Her0in (/author/Her0in) · 2016/01/28 10:26

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 (&

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。此文是译文的第一部分，主要阐述了 Mimikatz 的基本信息和检测 Mimikatz 使用的方法，在第二部分中会包含大量常用或不常用的 Mimikatz 命令的具体用法。

0x00 简介

看起来 Red（攻） & Blue（防） Team 里的大多数人并不熟悉 Mimikatz 的大部分功能，所以我收集了所有我可以找到的可用的命令并将它们整理在此文中。如果，日后我又找到了一些新的有用的命令，我也会更新此文。这样， Red（攻） & Blue（防） Team 的黑客就都可以更好的理解 Mimikatz 的所有功能，同时也能把雇佣其所保护的企业的安全做的更好。

在与很多人，包括被雇佣的攻防两方的黑客，网络从业者交谈之后，我了解到除了最常用的几个 Mimikatz 命令外，大多数人并不知道 Mimikatz 的全部功能。本文将尽可能的详述每一个命令，它是什么，它又是如何工作的，以及运行它所需的权限，参数（必需的和可选的），另外还将使用相关的截图及附加的内容进行说明（如果可能的话）。然而还有一些内容我没有研究过，期望在不久的将来能深入探究。我会继续将关于使用 Mimikatz 的各个方面的文章发表在 ADSecurity.org 上，不过我打算一直更新此文，并尽可能的全面。

> 本文的内容是为了帮助企业更好地了解 Mimikatz 的功能，且不能被用于非法活动。请不要在未获批准的计算机上使用 Mimikatz ，总之，不要使用 Mimikatz 进行渗透测试攻击。
> 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。

我没有参与编写 Mimikatz，因此没有特别的见解。本文的所有内容都是从使用 Mimikatz，阅读 Mimikatz 源代码，与 Benjamin 交流以及查阅他的博客和 GitHub 上的页面，还有我自己的研究中获得的。

本文中的的任何错误都是我自己的错误而已。请在这里 (https://adsecurity.org/?page_id=293)发表评论。

非常感谢 Benjamin Delpy 编写并不断更新 Mimikatz。他所做的工作极大地提高了 Windows 的安全性，尤其是 Windows10。

0x02 Mimikatz 概述

Mimikatz 是从 Windows 系统中收集凭证数据最好的工具之一。事实上，我个人认为 Mimikatz 是收集 Windows 系统凭证数据的“瑞士军刀”（多个利器的集合） - 一个可以做任何事情的工具。由于 Mimikatz 的作者 Benjamin Delpy 是法国人，所以描述关于 Mimikatz 用法的资源都是法语的，至少在他的博客 (http://blog.gentilkiwi.com/)中是这样的。 Mimikatz 的 GitHub 页面 (https://github.com/gentilkiwi/mimikatz) 是英文的，包括了命令的用法等有用信息。

Mimikatz 是 Benjamin Delpy (@gentilkiwi) 在 2007 年使用 C 语言编写的一个 Windows x32/x64 程序，用于了解更多关于 Windows 的凭据数据（并作为 POC）。

有两个可选的组件能提供一些额外的功能，mimidrv（与 Windows 内核交互的驱动程序）和 mimilib（绕过 AppLocker，验证包/SSP，密码过滤器以及用于 WinDBG 的 sekurlsa）。

Mimikatz 需要管理员或 SYSTEM 权限，通常使用 DEBUG 权限执行某些操作，与 LSASS 进程（取决于所做的操作的要求）进行交互。

Mimikatz 可以通过编译并运行你自己的版本 (https://github.com/gentilkiwi/mimikatz)，运行 Mimikatz 可执行文件 (https://github.com/gentilkiwi/mimikatz/releases)，利用 Metasploit 脚本 (https://www.offensive-security.com/metasploit-unleashed/mimikatz/)，和官方的 PowerShell 版本— Invoke-Mimikatz (https://github.com/PowerShellMafia/PowerSploit)，或 Mimikatz 的十多个 PowerShell 变种（我比较偏爱用 PowerShell 写的 Empire (https://raw.githubusercontent.com/PowerShellEmpire/Empire/master/data/module_source/credentials/Invoke-Mimikatz.ps1)，因为它真的很棒！）。

Mimikatz 的源代码和二进制版本可在 GitHub 上找到，并遵守 Creative Commons 许可，具体细节如下：

您可以自由：

- 分享 - 在任何媒体或以任何格式复制和发布相关的源文件
- 适应 - 在源文件的基础上进行任何改造
- 用于任何目的，甚至是商业化。只要你遵守了该许可条款，许可证就不能废除这些自由。
- 署名 - 你必须给予适当的说明，提供了一个链接到许可证，并注明是否进行了更改。你可以这样做以任何合理的方式，但并不是在暗示许可条款认可您或您所使用的任何方式。
- 没有额外的限制 - 您可能不适用于在法律上限制他人做任何事情的许可牌照的法律条款或技术措施。

0x03 Mimikatz 作者介绍

- Benjamin DELPY gentilkiwi, 可以在 Twitter 上关注他 (@gentilkiwi) or 邮件联系 (benjamin [at] gentilkiwi.com)
- lsadump 模块的 DCSync 功能是 Benjamin 与 Vincent LE TOUX 一起合作写的, 可以邮件联系 vincent (vincent.letoux [at] gmail.com) or 访问他的主页 (http://www.mysmartlogon.com (http://www.mysmartlogon.com/))

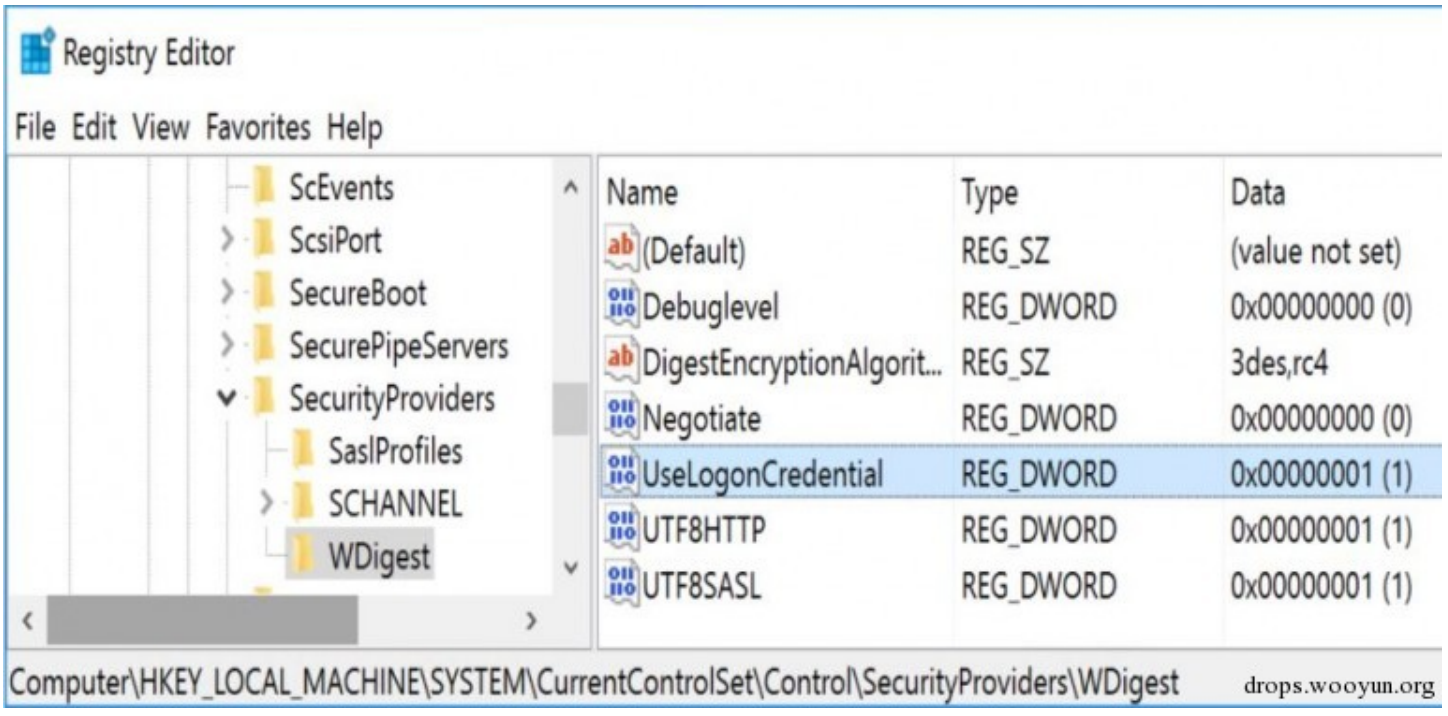
0x04 Mimikatz 官方链接

- Mimikatz GitHub (https://github.com/gentilkiwi/mimikatz) (源码)
- Mimikatz Releases (https://github.com/gentilkiwi/mimikatz/releases) (包含了编译好的二进制文件)
- Mimikatz GitHub Wiki (https://github.com/gentilkiwi/mimikatz/wiki) (包含了一些说明文档)
- GentilKiwi Blog (http://blog.gentilkiwi.com/mimikatz) (博客里的大多数内容是用法语写的,请使用 Chrome 浏览器进行自动翻译)

0x05 Mimikatz 与 凭证

在用户登录之后，会生成很多凭证数据并存储在本地安全权限服务的进程（LSASS (http://en.wikipedia.org/wiki/Local_Security_Authority_Subsystem_Service)）内存中。其目的是为了方便单点登录（SSO）在每次对资源进行访问请求时确保用户不会被提示。凭证数据包括 NTLM 密码哈希，LM 密码哈希（如果密码长度小于 15 个字符），甚至明文密码（以支持其他的 WDigest 和 SSP 认证）。虽然可以阻止 Windows 创建 LM 哈希到本地计算机的 SAM 数据库（或 AD 数据库） (http://support.microsoft.com/kb/299656)，但这并不能阻止系统在内存中生成 LM 哈希。默认情况下，在 Windows Server 2008 和 Windows Vista 中不再生成用户的 LM 哈希 (https://technet.microsoft.com/en-us/magazine/2006.08.securitywatch.aspx)，除非明确的启用了该功能。从 Windows 8.1 和 Windows Server 2012 R2 开始，LM 哈希和“纯文本”密码将不在内存中生成。此功能也被“移植”到了较早版本的 Windows 中，Windows 7/8/2008 R2/2012 需要打 kb2871997 补丁。为了防止在 LSASS 进程中放置“明文”密码，下面的注册表键需要被设置为“0”（禁用摘要）：

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest
“UseLogonCredential”(DWORD)
```



此注册表项值得你在你的环境中进行监控，因为攻击者可能希望将它设置为 1 ，启用摘要密码支持，以便在任何版本的 Windows 中，从 Windows 7/2008 R2 到 Windows 10/2012 R2，强制将“明文”密码放置在 LSASS 进程中。在 Windows 8.1/2012 R2 及以后的 Windows 系统中，没有 “UseLogonCredential” 的 DWORD 值，所以必须手动创建。如果在这些系统中存在了该键，可能意味着该系统存在一些问题。

需要注意的是，对于攻击者来说直接在目标系统上运行代码的希望很小，因此 Mimikatz 不断使用新的能够远程运行的功能进行更新。这包括运行 Mimikatz 并对远程系统进行远程转储凭证，使用 PowerShell 远程管理执行 Invoke-Mimikatz 以及 DCSync (https://adsecurity.org/?page_id=1821#DCSync)，最新的一个特性是针对某台 DC 进行远程抓取在域中的任何 Active Directory 帐户的密码数据时，不需要在 DC 中运行任何 Mimikatz 代码（它使用了微软官方的域控制器复制 API ，需要正确的权限来执行此功能 (https://adsecurity.org/?p=1729)）。

0x06 不同版本的操作系统中可用的凭证

Benjamin Delpy 在 OneDrive 上传了一个 Excel 图表（如下图所示），显示了在内存（LSASS）中可用的证书数据类型，其中包括了在 Windows 8.1 和 Windows 2012 R2 中增强的“减少保存在内存中的凭证数量和类型”的保护机制。

	Primary			CredentialKeys				tspkg		wdigest		kerberos				livessp	ssp	dpapi	credman 6
	LM	NTLM	SHA1	NTLM	SHA1	Root	DPAPI	off	on	off	on	pass 1	PIN 4	tickets	eKeys				
	Windows XP/2003																		
Local Account								2											
Domain Account								2					5						
	Windows Vista/2008 & 7/2008r2																		
Local Account																			
Domain Account																			
	Windows 8/2012																		
Microsoft Account																			
Local Account																			
Domain Account																			
	Windows 8.1/2012r2																		
Microsoft Account									3		3								
Local Account									3		3	7							
Domain Account									3		3								
Domain Protected Users									3		3								

	Windows 8.1 vault for user's authentication				
	PIN		Picture		Fingerprint
	code	pass	gestures	pass	pass
Microsoft Account					
Local Account					

not applicable
data in memory
no data in memory

1. can need an unlock on NT5, not available with smartcard
2. tspkg is not installed by default on XP, not available on 2003
3. tspkg is off by default (but needed for SSO with remoteapps/ts), wdigest too
   http://technet.microsoft.com/library/dn303404.aspx
4. PIN code when SmartCard used for native Logon
5. PIN code is NOT encrypted in memory (XP/2003)
6. When accessed/used by owner
7. When local admin, UAC and after unlock

0x07 Powershell 与 Mimikatz:

Mimikatz 中的大多数功能在 PowerSploit (https://github.com/mattifestation/PowerSploit)（PowerShell 渗透框架）中都是可用的。通过 “Invoke-Mimikatz” (https://github.com/mattifestation/PowerSploit/blob/master/Exfiltration/Invoke-Mimikatz.ps1) 这个 PowerShell 脚本（作者是Joseph Bialek (https://twitter.com/JosephBialek)），此脚本“利用Mimikatz 2.0 和 Invoke-ReflectivePEInjection 反射式的将 Mimikatz 完全加载到内存运行。这使得你在转储凭证时无需将 Mimikatz 的二进制文件写到磁盘中。PowerSploit 框架现在托管在 “PowerShellMafia” (https://github.com/PowerShellMafia/PowerSploit) GitHub 库中。

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。

防御者应该预料到包含在 Mimikatz 中的任何功能在 Invoke-Mimikatz 中都是可用的。

0x08 检测 Mimikatz 的方法

有几种方法可以潜在地检测 Mimikatz 在网络上的使用，虽然这些方法不能保证一定可行。由于 Mimikatz 的源代码在 GitHub 上已经公开，所以任何人都可以使用 Visual Studio 编译他们自己的版本。我构建了我自己的 Mimikatz 版本叫做 “kitkatz”，将所有的 “mimikatz” 实例替换为 “kitikatz”后，在 VirusTotal 上的检测率并不理想（4/54）。在我的 Windows 10 中的 Windows Defender 检测到了它。之后，我使用相同的字词将 “Benjamin Delpy” 和 “gentilkiwi” 替换了一下，仅仅是把 “e” 替换为 “3” ，“i” 替换为 “1”。检测率仍然比较差（4/54）
(https://www.virustotal.com/en/file/84b6cd8ccaa60a89c1375bec5044e6240d8a2db3f19fecd763749fa9a530470b/analysis/1449967355/)。然而在我的 Windows 10 中的 Windows Defender 却没有检测到它。所以，你的检测情况会有所不同。

- Benjamin Delpy 在 Mimkatz 的 GitHub 库 (https://github.com/gentilkiwi/mimikatz)中发布了 Mimikatz 的 YARA 规则 (https://plusvic.github.io/yara/)。
- 运行最新版的防病毒软件。 VirusTotal 在 2015年11月11日使用 AV 引擎对 mimikatz.exe（32位＆64位）的检测率为 35/35。
- Mimikatz (截至 10 月) 在 BusyLights 的附属活动 (http://blog.cobaltstrike.com/2015/11/11/revolutionary-device-detects-mimikatz-use/)。[使用 Mimikatz version 2.0 alpha 20151008 (oe.eo) edition 实现]
- 充分利用安全软件来鉴定与 LSASS 进程交互的过程。安全软件对注入过程进行监视可能同样也可以定期去检测 Mimikatz 的使用。
- 在企业中的多台计算机的内存中植入特殊凭据包括 HoneyTokens/HoneyHashes (https://isc.sans.edu/diary/Detecting+Mimikatz+Use+On+Your+Network/19311)。这些凭据是被标记过的，所以当有人试图使用它们时，系统会发出严重的警报。这需要某种推送方法以及对攻击者有吸引力的凭证。在理论上，这可能能够检测出凭证窃取并且能够在环境中使用此方法。
- WDIGEST 注册表键（ `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest` ）应该被设置为“0”，以防止在 LSASS 进程中存储“明文”密码。如果在企业内部的系统中，该注册表键被设置为“1”，这可能表示有凭证窃取活动发生。此注册表项值得你在你的环境中进行监控，因为攻击者可能希望将它设

置为 1，启用摘要密码支持，以便在任何版本的 Windows 中，从 Windows 7/2008 R2 到 Windows 10/2012 R2，强制将“明文”密码放置在 LSASS 进程中。

- 检测伪造 Kerberos 票证的方法我在 2015 年年初发表过，这些方法可以检测出黄金票证，白银票证，以及信任票证的伪造 (https://adsecurity.org/?p=1515)。对如何检测 MS14-068 Kerberos 漏洞攻击 (https://adsecurity.org/?p=763)我也给出了一些信息。
- 启用所有的 Windows 版本中所支持的企业 LSA 保护。这可以防止 Mimikatz 工作方式的“即开即用”，并需要使用Mimikatz驱动程序，记录事件，当它与LSASS交互的。
- 在所有的企业版的 Windows 中启用 LSA 保护。这可以防止 Mimikatz 工作方式的 “即开即用” ，以及阻止其在运行时需要使用 Mimikatz 驱动程序的要求，当它与 LSASS 进行交互时，也会记录事件日志。

0x09 Mimikatz 与 LSA 保护方式

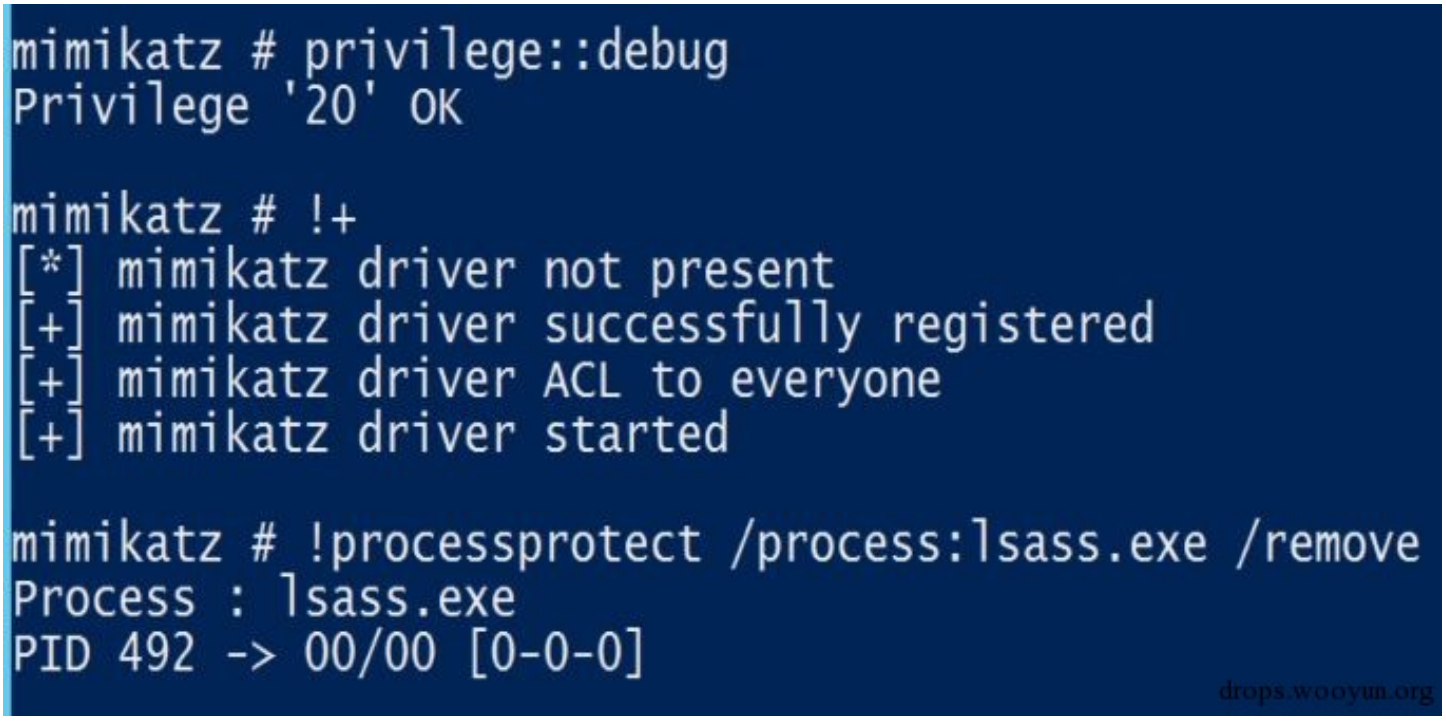
在 Windows Server 2012 R2 和 Windows 8.1 中包含了一个名为 “LSA 保护” 的新功能，其中包括在 Windows Server 2012 R2 上启用 LSASS 作为一个受保护的进程（Mimikatz 可以使用驱动程序绕过此保护措施，但会在事件日志中产生一些日志）：

LSA 包括本地身份验证服务子系统（LSASS）进程，验证用户的本地和远程登录并强制实施本地安全策略。Windows 8.1 操作系统为 LSA 提供了额外的保护阻止非受保护的进程读取内存和代码注入。这为 LSA 存储和管理的凭证信息提供了额外的安全性。

启用 LSA 保护

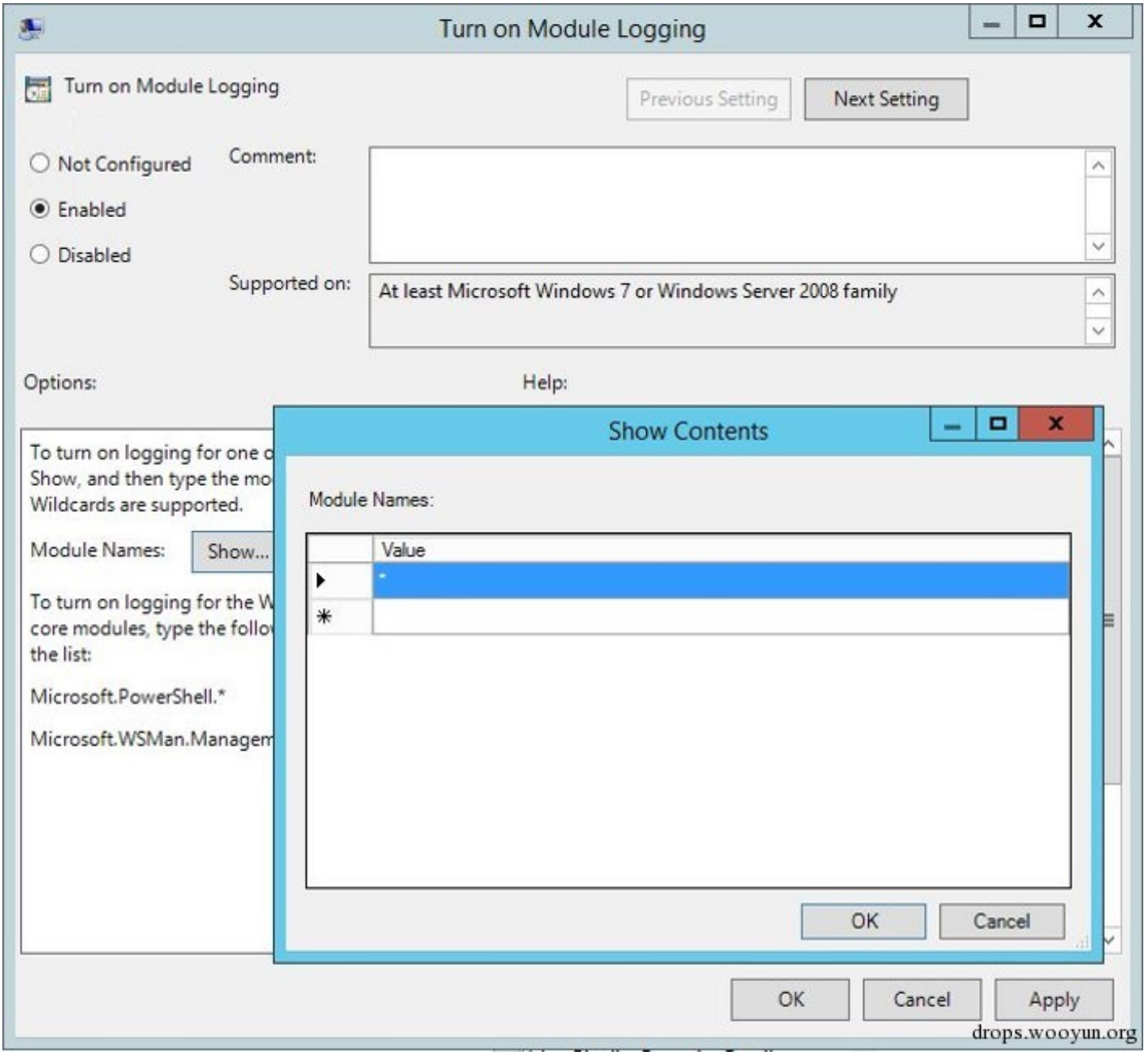
1. 打开注册表编辑器（Regedit.exe），并定位到注册表项： `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa` ，设置注册表键的值为： `“RunAsPPL”=dword:00000001`
2. 创建一个新的 GPO 并浏览到 “计算机配置”—“首选项”—“Windows设置”。右键单击注册表—新建，然后单击注册表项。弹出新的注册表属性对话框。在配置单元列表中，单击 `HKEY_LOCAL_MACHINE` 在注册表键路径列表中浏览到 `SYSTEM\CurrentControlSet\Control\Lsa` ，在值名称框中，输入 “RunAsPPL” ，在值类型框中单击 “REG_DWORD” ，在数值数据框中，输入 “00000001” ，最后单击“确定”。

LSA 保护阻止了一个非受保护的进程与 LSASS 进行交互。但是 Mimikatz 仍然可以使用驱动程序绕过（“!+”）。



0x0A 检测 Invoke-Mimikatz

- 确保所有的 Windows 系统都安装了 PowerShell v3 或更高版本。新版本的 PowerShell 有更好的日志记录功能，特别是PowerShell V5。
- 开启通过组策略记录 PowerShell 模块运行日志：计算机配置 — 策略 — 管理模板 — Windows组件 — Windows PowerShell，打开模块记录功能。输入“*”，然后单击确定。这将记录所有的 PowerShell 活动，包括所有的 PowerShell 模块。



- PowerShell 的活动将被记录到 PowerShell 的操作日志中。将这些事件推送到中央日志服务器（通过 Windows 事件转发或类似的方法）或 SIEM。
- 使用如下方法解析 PowerShell 的事件：
    - “System.Reflection.AssemblyName”
    - “System.Reflection.Emit.AssemblyBuilderAccess ”
    - “System.Runtime.InteropServices.MarshalAsAttribute”
    - “TOKEN_PRIVILEGES”
    - “ `SE_PRIVILEGE_ENABLED` ”

注意：虽然有可能通过提醒中的 “mimikatz”，“Delpy”，或 “gentilkiwi” 识别出 Mimikatz 的使用，但是一个“牛逼”的攻击者可能会推出自己的 Mimikatz 或 Invoke-Mimikatz 版本并且没有这些关键字。

检测带有攻击性的 PowerShell 工具

许多 PowerShell 的攻击工具都使用了如下被记录到 PowerShell 模块日志记录的调用方法。

- “GetDelegateForFunctionPointer”
- “System.Reflection.AssemblyName“
- “System.Reflection.Emit.AssemblyBuilderAccess“
- “System.Management.Automation.WindowsErrorReporting”
- “MiniDumpWriteDump”
- “ `TOKEN_IMPERSONATE` ”
- “ `TOKEN_DUPLICATE` ”
- “ `TOKEN_ADJUST_PRIVILEGES` ”
- “ `TOKEN_PRIVILEGES` ”