12-Enumerating Security Controls

هنا بنشرح مثلا لو في security معمولة ممكن تمنعنا وكده on powershell

1-Windows Defender

هنا مثلا update اللي حصل في win10 2020 بيمنع مثلا ان استخدم PowerView tool

Windows Defender (or Microsoft Defender after the Windows 10 May 2020 Update) has greatly improved over the years and, by default, will block tools such as `PowerView`. There are ways to bypass these protections. These ways will be covered in other modules. We can use the built-in PowerShell cmdlet Get-MpComputerStatus to get the current Defender status. Here, we can see that the `RealTimeProtectionEnabled` parameter is set to `True`, which means Defender is enabled on the system.

Checking the Status of Defender with Get-MpComputerStatus

```
PS C:\htb> Get-MpComputerStatus

AMEngineVersion                 : 1.1.17400.5
AMProductVersion                : 4.10.14393.0
AMServiceEnabled                : True
AMServiceVersion                : 4.10.14393.0
AntispywareEnabled              : True
AntispywareSignatureAge         : 1
AntispywareSignatureLastUpdated : 9/2/2020 11:31:50 AM
AntispywareSignatureVersion     : 1.323.392.0
AntivirusEnabled                : True
AntivirusSignatureAge           : 1
AntivirusSignatureLastUpdated   : 9/2/2020 11:31:51 AM
AntivirusSignatureVersion       : 1.323.392.0
BehaviorMonitorEnabled          : False
ComputerID                      : 07D23A51-F83F-4651-B9ED-110FF2B83A9C
ComputerState                   : 0
FullScanAge                     : 4294967295
FullScanEndTime                 :
FullScanStartTime               :
IoavProtectionEnabled           : False
LastFullScanSource              : 0
LastQuickScanSource             : 2
NISEnabled                      : False
NISEngineVersion                : 0.0.0.0
```

```
NISSignatureAge                 : 4294967295
NISSignatureLastUpdated         :
NISSignatureVersion             : 0.0.0.0
OnAccessProtectionEnabled       : False
QuickScanAge                    : 0
QuickScanEndTime                : 9/3/2020 12:50:45 AM
QuickScanStartTime              : 9/3/2020 12:49:49 AM
RealTimeProtectionEnabled       : True
RealTimeScanDirection           : 0
PSComputerName                  :
```

2-AppLocker

An application whitelist is a list of approved software applications or executables that are allowed to be present and run on a system. The goal is to protect the environment from harmful malware and unapproved software that does not align with the specific business needs of an organization. AppLocker is Microsoft's application whitelisting solution and gives system administrators control over which applications and files users can run. It provides granular control over executables, scripts, Windows installer files, DLLs, packaged apps, and packed app installers. It is common for organizations to block cmd.exe and PowerShell.exe and write access to certain directories, but this can all be bypassed. Organizations also often focus on blocking the `PowerShell.exe` executable, but forget about the other PowerShell executable locations such as `%SystemRoot%\SysWOW64\WindowsPowerShell\v1.0\powershell.exe` or `PowerShell_ISE.exe`. We can see that this is the case in the `AppLocker` rules shown below. All Domain Users are disallowed from running the 64-bit PowerShell executable located at:

%SystemRoot%\system32\WindowsPowerShell\v1.0\powershell.exe

1. ما هو Application Whitelisting؟

- هو قائمة بالتطبيقات الموثوقة والمسموح تشغيلها على النظام.
- الهدف:
  - حماية النظام من البرامج الضارة (Malware).
  - منع تشغيل البرامج غير المصرح بها اللي مالهاش علاقة باحتياجات العمل.

2. ما هو AppLocker؟

- أداة من مايكروسوفت تمكن المسؤولين من:
  - التحكم في البرامج اللي يُسمح للمستخدمين بتشغيلها.
  - إنشاء قواعد (Rules) للتحكم في:

- الملفات التنفيذية (Executables).
- السكربتات (Scripts).
- ملفات التثبيت (Windows Installer files).
- مكتبات DLL.
- التطبيقات المعبأة (Packaged Apps).

3. كيفية الاستخدام في المؤسسات:

- المؤسسات غالبًا بتعمل حظر لتطبيقات خطيرة أو ممكن تُستغل، زي:
  - **cmd.exe** (سطر الأوامر).
  - **PowerShell.exe** (السكربتات القوية).
  - الكتابة في مسارات حساسة داخل النظام.

الثغرات في الحماية باستخدام AppLocker:

التركيز فقط على PowerShell.exe:

- العديد من المؤسسات بتحظر ملف **PowerShell.exe** الرئيسي الموجود في:
  `%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe`
- لكنهم بينسوا المواقع البديلة اللي فيها نسخ تانية من PowerShell، زي:
  `%SystemRoot%\SysWOW64\WindowsPowerShell\v1.0\powershell.exePowerShell_ISE.exe`
- الثغرة دي بتسمح للمهاجم تشغيل PowerShell من مواقع أخرى غير محظورة.

لماذا هذا يعتبر مشكلة؟

- **PowerShell** أداة قوية جدًا يمكن استخدامها لتنفيذ أوامر على النظام والوصول إلى بيانات حساسة.
- ونسوا النسخ الأخرى، يظل بإمكان المهاجم استغلالها لتجاوز الحماية PowerShell إذا حظروا نسخة واحدة من.

كيف يتم تجاوز الحظر؟

1. **تشغيل PowerShell من مسار غير محظور:**
   - يمكن تشغيل النسخة البديلة، PowerShell لو لم يتم حظر كل المواقع الممكنة لنسخ.
2. **استخدام PowerShell ISE (بيئة تطوير PowerShell):**
   - يمكن استخدام **PowerShell_ISE.exe** بنفس الكفاءة مثل **PowerShell.exe**.
3. **إنشاء نسخ بديلة:**
   - المهاجم قد ينسخ ملف PowerShell.exe إلى مسار غير محظور.

كيفية تعزيز الحماية:

1. **مراجعة جميع المواقع المحتملة:**
   - حظر كل PowerShell، بما في ذلك المسارات اللي تحتوي على:

- `%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe`
- `%SystemRoot%\SysWOW64\WindowsPowerShell\v1.0\powershell.exe`
- **PowerShell_ISE.exe**

2. **الاعتماد على أدوات إدارة متقدمة:**
   - استخدام **Windows Defender Application Control (WDAC)** بجانب AppLocker.
3. **التحديث الدوري للقواعد:**
   - مراجعة وتحديث القواعد بشكل دوري لمنع الطرق الجديدة لتجاوز الحماية.
4. **الحد من الامتيازات (Least Privilege):**
   - تقييد استخدام PowerShell للمستخدمين اللي يحتاجونه فعلاً.

باختصار:

- AppLocker أداة قوية لحماية الأنظمة، لكنها ممكن تُخترق إذا كانت القواعد غير كاملة.
- حماية PowerShell لازم تشمل كل المواقع والملفات البديلة لضمان عدم تجاوز الحظر.

Using Get-AppLockerPolicy cmdlet

```
PS C:\htb> Get-AppLockerPolicy -Effective | select -ExpandProperty
RuleCollections

PathConditions      : {%SYSTEM32%\WINDOWSPOWERSHELL\V1.0\POWERSHELL.EXE}
PathExceptions      : {}
PublisherExceptions : {}
HashExceptions      : {}
Id                  : 3d57af4a-6cf8-4e5b-acfc-c2c2956061fa
Name                : Block PowerShell
Description         : Blocks Domain Users from using PowerShell on
workstations
UserOrGroupSid      : S-1-5-21-2974783224-3764228556-2640795941-513
Action              : Deny

PathConditions      : {%PROGRAMFILES%\*}
PathExceptions      : {}
PublisherExceptions : {}
HashExceptions      : {}
Id                  : 921cc481-6e17-4653-8f75-050b80acca20
Name                : (Default Rule) All files located in the Program Files
folder
Description         : Allows members of the Everyone group to run
applications that are located in the Program Files folder.
UserOrGroupSid      : S-1-1-0
Action              : Allow
```

```
PathConditions      : {%WINDIR%\*}
PathExceptions      : {}
PublisherExceptions : {}
HashExceptions      : {}
Id                  : a61c8b2c-a319-4cd0-9690-d2177cad7b51
Name                : (Default Rule) All files located in the Windows folder
Description         : Allows members of the Everyone group to run
applications that are located in the Windows folder.
UserOrGroupSid      : S-1-1-0
Action              : Allow

PathConditions      : {*}
PathExceptions      : {}
PublisherExceptions : {}
HashExceptions      : {}
Id                  : fd686d83-a829-4351-8ff4-27c7de5755d2
Name                : (Default Rule) All files
Description         : Allows members of the local Administrators group to
run all applications.
UserOrGroupSid      : S-1-5-32-544
Action              : Allow
```

3-Constrained Language Mode

ما هو **Constrained Language Mode**؟

- هو وضع حماية يقوم بتقييد الأدوات والوظائف المتاحة في **PowerShell**.
- يتم تفعيله تلقائيًا في حالات معينة، مثل:
  - تشغيل PowerShell بحساب له صلاحيات محدودة.
  - وجود قيود إضافية مثل **AppLocker** أو **WDAC** (Windows Defender Application Control).

ماذا يقيد هذا الوضع؟

1. **حظر كائنات COM:**
   - يمنع استخدام كائنات **COM** (Component Object Model) التي تُستخدم للتفاعل بين البرامج، لأنها غالبًا ما تُستغل في تنفيذ هجمات.
2. **السماح فقط بأنواع محددة من .NET:**
   - يقيد الوصول إلى مكتبة **.NET Framework** ويسمح فقط باستخدام الأنواع الآمنة أو المصرح بها مسبقًا.
3. **منع Workflows المبنية على XAML:**
   - يحظر تشغيل Workflows المبنية على **XAML**، والتي تُستخدم غالبًا في تطوير الواجهات والتطبيقات.

4. **تعطيل الفئات (Classes):**
   - يمنع تعريف وإنشاء الفئات (Classes) داخل PowerShell، مما يقلل من مرونة البرمجة.
5. **قيود إضافية:**
   - منع تحميل مكتبات خارجية.
   - الحد من تنفيذ السكربتات الديناميكية المعقدة.

كيف تعرف الوضع الحالي للغة PowerShell؟

يمكنك معرفة إذا كان PowerShell يعمل في **وضع كامل الصلاحيات (Full Language Mode)** أو **وضع مقيد (Constrained Language Mode)** باستخدام الأمر التالي:

```
PS C:\htb> $ExecutionContext.SessionState.LanguageMode

ConstrainedLanguage
```

النتائج المحتملة:

- **FullLanguage:**
  - يعني أن الوضع كامل الصلاحيات، ويمكنك تنفيذ جميع الأوامر والوظائف بدون قيود.
- **ConstrainedLanguage:**
  - يعني أن الوضع مقيد، وهناك العديد من الميزات التي لا يمكنك استخدامها.

أهمية Constrained Language Mode:

- **حماية الأنظمة:** يقلل من فرص تنفيذ الهجمات التي تعتمد على PowerShell.
- **تقييد الحسابات:** يمنع الحسابات ذات الصلاحيات المحدودة من استخدام PowerShell بفعالية في الهجمات.

كيف تتخطى هذا الوضع؟

- إذا كان لديك صلاحيات إدارية، يمكنك تعديل الإعدادات لتشغيل PowerShell في **Full Language Mode**.
- ولكن هذا يتطلب أن تكون مسؤولاً على النظام (Administrator).

> **ملاحظة:** الالتزام بوضع **Constrained Language Mode** يعزز أمان النظام، وتجاوزه يجب أن يكون بحذر وبموافقة واضحة.

4-LABS (Local Administrator Password Solution)

تستخدم Microsoft (LAPS) لتوليد كلمات مرور عشوائية وتدويرها Administrator على الأجهزة التي تعمل بنظام Windows، وذلك لمنع الحركة الجانبية (lateral movement) بين الأجهزة في الشبكة. الميزة الأساسية هي أن LAPS تضمن أن لكل جهاز كلمة مرور مسؤول محلية فريدة ومحدثة بانتظام، مما يزيد من أمان الشبكة.

بخصوص **(enumeration)**، يمكننا تحديد أي مستخدمين في نطاق الدومين لديهم القدرة على قراءة كلمة مرور المسؤول المحلي لأي جهاز يحتوي على LAPS مُثبت، وأي الأجهزة التي لا تحتوي على LAPS.

LAPSToolkit: https://github.com/leoloobeek/LAPSToolkit هو مجموعة أدوات تساعد في تنفيذ هذه المهام بسهولة من خلال عدة وظائف، مثل:

1. **Parsing ExtendedRights :** بالإضافة إلى LAPS، هذه العملية تتيح لنا تحديد الأجهزة التي تحتوي على LAPS. معرفة المجموعات المُعينة التي لديها صلاحية قراءة كلمات المرور الخاصة بـ هذه المجموعات عادةً ما تكون مجموعات محمية، مثل مديري النظام أو المسؤولين.
2. **(Extended Rights):** عندما يقوم حساب بإضافة جهاز إلى الدومين، فإنه يحصل على حقوق موسعة على هذا الجهاز، وهو أمر يمكن أن يُستغل إذا LAPS على هذا الجهاز. هذا يعني أن الحساب يمكنه قراءة كلمات المرور الخاصة بـ كان الحساب الذي يمتلك هذه الحقوق في أيدٍ غير آمنة.

إذا قمنا بالبحث عن المستخدمين الذين يمكنهم قراءة كلمات مرور **LAPS**، فقد نتمكن من تحديد الحسابات التي يمكنها الوصول إلى كلمات مرور **administrator**، مما يسمح لنا باستهداف هؤلاء المستخدمين في الهجمات المحتملة.

باختصار، الهدف من **Enumeration** هو اكتشاف أي الحسابات في الدومين يمكنها قراءة كلمات مرور **administrator** للأجهزة المُثبت عليها **LAPS**، مما يساعد في تحديد الثغرات الأمنية المحتملة أو تسليط الضوء على الحسابات المعرّضة للخطر.

Using Find-LAPSDelegatedGroups

```
PS C:\htb> Find-LAPSDelegatedGroups

OrgUnit                                                 Delegated Groups
-------                                                 ----------------
OU=Servers,DC=INLANEFREIGHT,DC=LOCAL                    INLANEFREIGHT\Domain
Admins
OU=Servers,DC=INLANEFREIGHT,DC=LOCAL                    INLANEFREIGHT\LAPS
Admins
OU=Workstations,DC=INLANEFREIGHT,DC=LOCAL               INLANEFREIGHT\Domain
Admins
OU=Workstations,DC=INLANEFREIGHT,DC=LOCAL               INLANEFREIGHT\LAPS
Admins
OU=Web Servers,OU=Servers,DC=INLANEFREIGHT,DC=LOCAL     INLANEFREIGHT\Domain
Admins
OU=Web Servers,OU=Servers,DC=INLANEFREIGHT,DC=LOCAL     INLANEFREIGHT\LAPS
Admins
OU=SQL Servers,OU=Servers,DC=INLANEFREIGHT,DC=LOCAL     INLANEFREIGHT\Domain
Admins
OU=SQL Servers,OU=Servers,DC=INLANEFREIGHT,DC=LOCAL     INLANEFREIGHT\LAPS
Admins
OU=File Servers,OU=Servers,DC=INLANEFREIGHT,DC=L...     INLANEFREIGHT\Domain
Admins
OU=File Servers,OU=Servers,DC=INLANEFREIGHT,DC=L...     INLANEFREIGHT\LAPS
```

```
Admins
OU=Contractor Laptops,OU=Workstations,DC=INLANEF... INLANEFREIGHT\Domain
Admins
OU=Contractor Laptops,OU=Workstations,DC=INLANEF... INLANEFREIGHT\LAPS
Admins
OU=Staff Workstations,OU=Workstations,DC=INLANEF... INLANEFREIGHT\Domain
Admins
OU=Staff Workstations,OU=Workstations,DC=INLANEF... INLANEFREIGHT\LAPS
Admins
OU=Executive Workstations,OU=Workstations,DC=INL... INLANEFREIGHT\Domain
Admins
OU=Executive Workstations,OU=Workstations,DC=INL... INLANEFREIGHT\LAPS
Admins
OU=Mail Servers,OU=Servers,DC=INLANEFREIGHT,DC=L... INLANEFREIGHT\Domain
Admins
OU=Mail Servers,OU=Servers,DC=INLANEFREIGHT,DC=L... INLANEFREIGHT\LAPS
Admins
```

The `Find-AdmPwdExtendedRights` checks the rights on each computer with LAPS enabled for any groups with read access and users with "All Extended Rights." Users with "All Extended Rights" can read LAPS passwords and may be less protected than users in delegated groups, so this is worth checking for.

```
PS C:\htb> Find-AdmPwdExtendedRights

ComputerName                Identity                    Reason
------------                --------                    ------
EXCHG01.INLANEFREIGHT.LOCAL INLANEFREIGHT\Domain Admins Delegated
EXCHG01.INLANEFREIGHT.LOCAL INLANEFREIGHT\LAPS Admins   Delegated
SQL01.INLANEFREIGHT.LOCAL   INLANEFREIGHT\Domain Admins Delegated
SQL01.INLANEFREIGHT.LOCAL   INLANEFREIGHT\LAPS Admins   Delegated
WS01.INLANEFREIGHT.LOCAL    INLANEFREIGHT\Domain Admins Delegated
WS01.INLANEFREIGHT.LOCAL    INLANEFREIGHT\LAPS Admins   Delegated
```

Using Get-LAPSComputers

We can use the `Get-LAPSComputers` function to search for computers that have LAPS enabled when passwords expire, and even the randomized passwords in cleartext if our user has access.

```
PS C:\htb> Get-LAPSComputers

ComputerName                Password       Expiration
------------                --------       ----------
DC01.INLANEFREIGHT.LOCAL    6DZ[+A/[]19d$F 08/26/2020 23:29:45
EXCHG01.INLANEFREIGHT.LOCAL oj+2A+[hHMMtj, 09/26/2020 00:51:30
```

```
SQL01.INLANEFREIGHT.LOCAL    9G#f;p41dcAe,s 09/26/2020 00:30:09
WS01.INLANEFREIGHT.LOCAL     TCaG-F)3No;18C 09/26/2020 00:46:04
```