bi-sec
IT- & Informationssicherheit

Slides will be available afterwards

Bypassing App Control via Managed Installer

The Managed Installer „works as designed“ Case

E-Mail: Christian.Biehler@bi-sec.de
Twitter/X: @cssec
LinkedIn: bi-sec GmbH

bi-sec
IT- & Informationssicherheit

About: me

- Christian Biehler – IT-Security Expert
  - Master in IT Security
  - Working since 2012 as hacker, pentester, consultant
  - More than 400 projects regarding the topics
    - Pentesting / Vulnerability Management
    - Network security
    - Building and maintaining ISMS
    - Windows-, Linux-, Web Security
  - Trainer for Web Security, Windows Security, Microsoft 365 Security, AI assissted security testing
  - Author and speaker for different (German) papers and conferences
  - Trained and certified
    - ISO 27001 Lead Implementer, ISO 27035 Incident Manager, ISO 27005 Risk Manager
    - Certified Information Systems Security Professional (CISSP)
    - Certified Professional Penetration Tester (eCPPT)

Our topics for today

bi-sec
IT- & Informationssicherheit

Breaking App Control using Managed Installer

Fundamentals :-)

- Windows Security
- AppLocker & App Control

The Trick

- Setup
- Bypassing policies with PowerShell as Managed Installer

Our goal for today

bi-sec
IT- & Informationssicherheit

We want to:

- bypass a restrictive (signed) App Control policy
- by abusing built-in Windows functionality
- to run any desired software as an attacker.

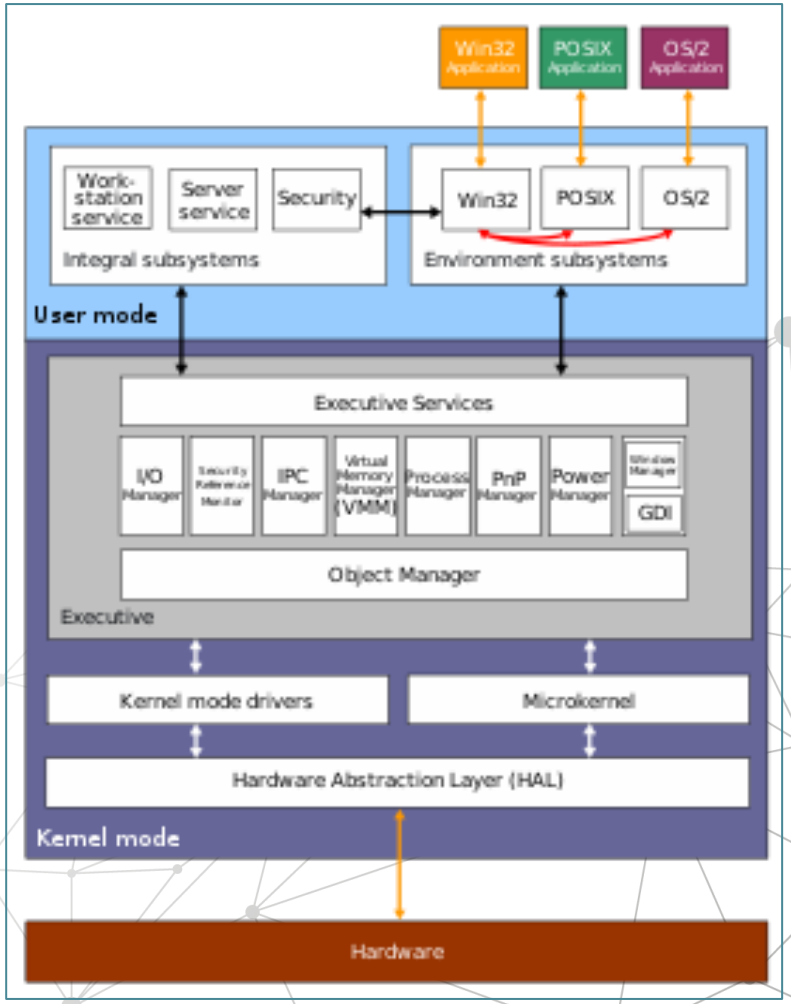
The really very shortest version of all times

Fundamentals :-)

Windows Fundamentals

bi-sec
IT- & Informationssicherheit

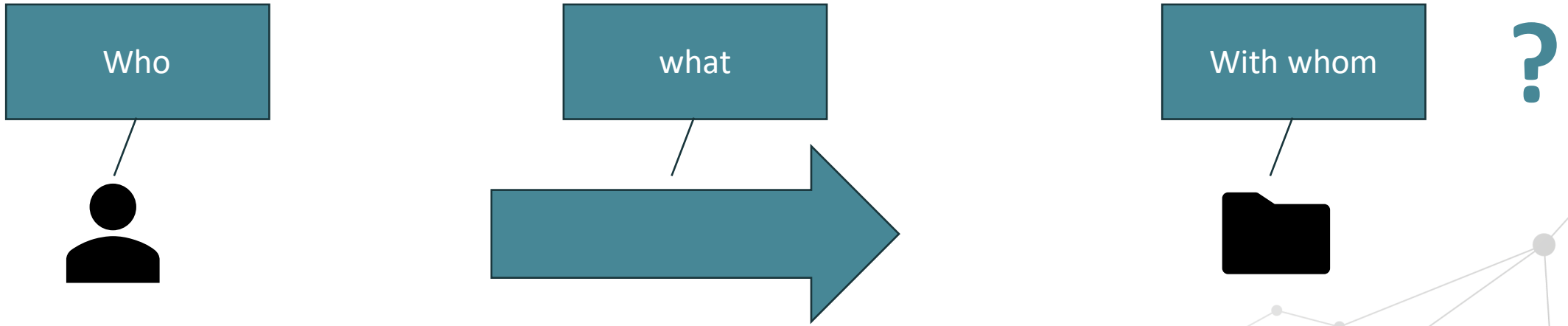
- Windows was developed in a time, were object-oriented programming was cool
  (some people think it is until today…)
- So, everything in Windows is an object
  - Counts as well for the Active Directory
  - In contrast: everything in Linux is a file
- Every object has attributes including a security descriptor and security-relevant information for the system



Windows security is based on just one single question!

There is only one question...

bi-sec
IT- & Informationssicherheit



Security Principal:
- User
- Group
- Computer
- Services
- …

Activity:
- Read
- Write
- ***Execute***
- …

Securable Object:
- File / Folder
- Process
- Session
- Registry Key
- Driver
- …

Why this matters…

bi-sec
IT- & Informationssicherheit

- Modern malware executes code in the context of the user
- Most cloud access tokens and cookies can be accessed without admin / system privileges
- Clickfix-family attacks are emerging

➢Software that users (and administrators) can run should be controlled …

Now, let‘s talk about …

bi-sec
IT- & Informationssicherheit

- App Control, aka:
  - App Control for Business
  - Application Control for Windows
  - Windows Defender Application Control (WDAC)
  - Microsoft Defender Application Control (MDAC)
  - Configurable Code Integrity
  - System Integrity Policy
  - Device Guard
  - Code Integrity Policy

(missed one?)

Intro: AppLocker & App Control

Summary: AppLocker vs. App Control

bi-sec
IT- & Informationssicherheit

	AppLocker (aka SRPv2)	App Control
Implementation	Working in user land	Working in kernel land
Rule storage location	Registry Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\SrpV2\	OS / Kernel / UEFI Unsigned: C:\Windows\System32\CodeIntegrity\CiPolicies\Active\ Signed (in addition): \EFI\Microsoft\Boot\CiPolicies\Active\
Security against malicious administrators	Admin can change, delete, …	Rules can be signed and UEFI „locked“ impeding admin changes
Flexibility	Changes work with gpupdate	Changes may require reboot
Protects from	Software only	Software and drivers
Implementation effort	Medium	High

Running App Control with signed Policy & Secure Boot

bi-sec
IT- & Informationssicherheit

Bootloader loads CI-Policy

- CI-Policy is located in the EFI

Policy signature is validated

- Signer must match policy signature / update signer

Policy is enforced

- Any new policy will be rejected if not signed (-> Secure Boot lock screen)
- Policy from EFI partition will be used if OS path policy is deleted

What that means is …

bi-sec
IT- & Informationssicherheit

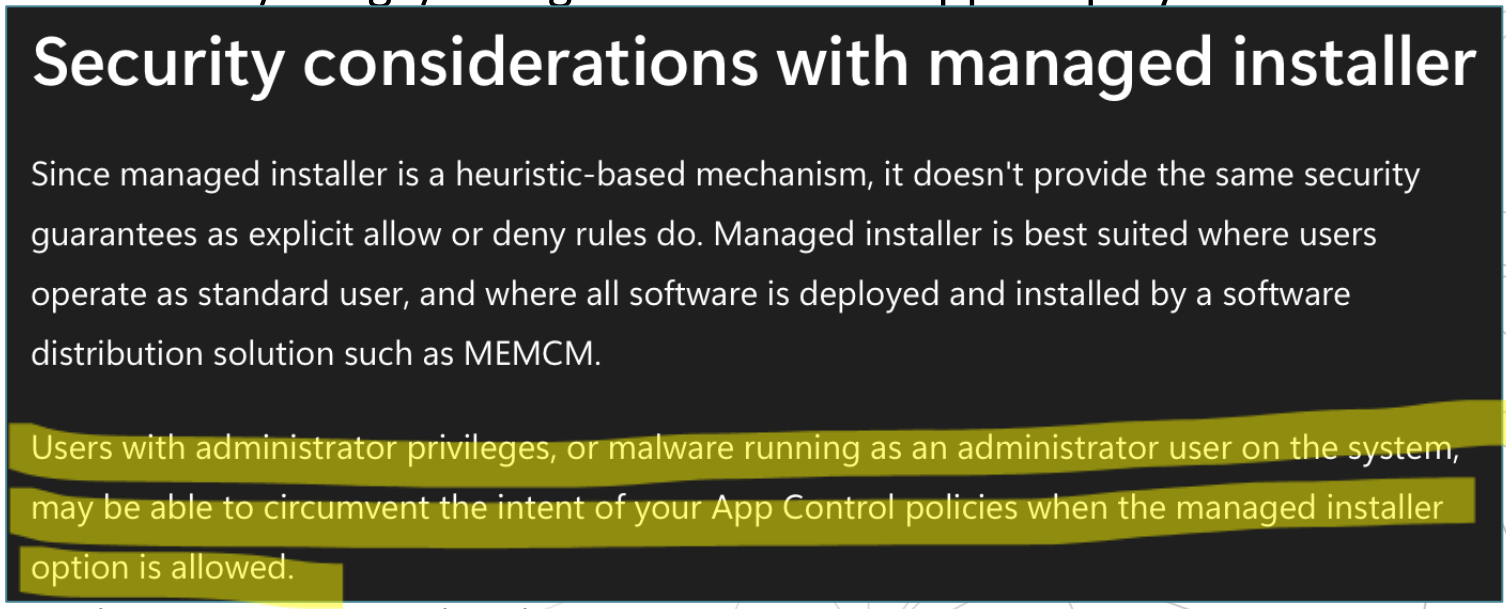
- … in theory, a signed App Control policy prevents bypasses from
  - … local administrator
  - … malware with system privileges
- … as long as the policy does not include that one feature …

The Trick

Sources / References

bi-sec
IT- & Informationssicherheit

- Please kindly note, that this is not a vulnerability
  - The behavior was officially described by Microsoft
  - Official MS guide:
    - https://learn.microsoft.com/en-us/windows/security/application-security/application-control/app-control-for-business/design/configure-authorized-apps-deployed-with-a-managed-installer



Security considerations with managed installer

Since managed installer is a heuristic-based mechanism, it doesn't provide the same security guarantees as explicit allow or deny rules do. Managed installer is best suited where users operate as standard user, and where all software is deployed and installed by a software distribution solution such as MEMCM.

Users with administrator privileges, or malware running as an administrator user on the system, may be able to circumvent the intent of your App Control policies when the managed installer option is allowed.

Managed Installer

bi-sec
IT- & Informationssicherheit

- Microsoft allows to define **Managed Installer** when using AppControl (MDAC / WDAC / UMCI / KMCI) on a device
  - All applications installed by the Managed Installer are allowed to run
  - All child apps (updates, …) created by the apps that were installed using **Managed Installer** are allowed to run as well … (!)
  - Security is propagated by file attributes that can only be written by System
- Defining Managed Installers requires AppLocker to run
  - A „dummy“ policy for .exe & .dll is enough
  - AppLocker RuleCollections can be used in AuditOnly Mode

Setup Part 1 – (legitimate) App Control Policies

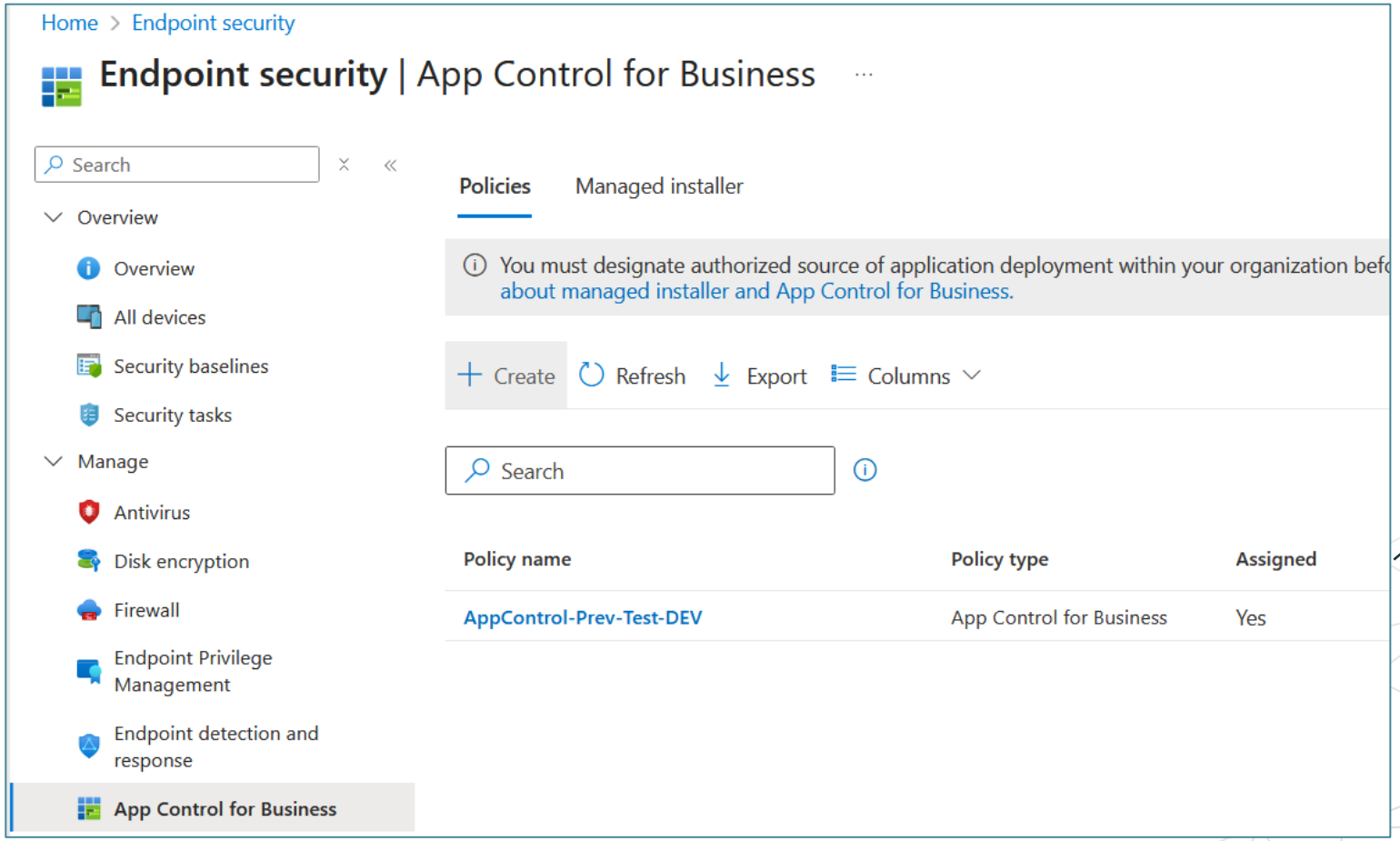
App Control – Built a demo policy in block mode

bi-sec
IT- & Informationssicherheit

- Create App Control policy using the Wizard
  - Add Program Files paths to reduce issues in the first place
    - *Remove before flight ..*
  - Make sure to have –Option 13 / Managed Installer enabled
- Deploy using GPO / Intune / Locally
  - GPO:
    - ConvertFrom-CIPolicy -XmlFilePath $AppControlPolicyXMLFile -BinaryFilePath $env:USERPROFILE\Desktop\$PolicyBinary
  - Intune:
    - Copy / Paste the XML ☺
  - Locally:
    - Copy / Paste .p7b / .cip file into the directory
      - C:\Windows\System32\...

App Control - Intune

bi-sec
IT- & Informationssicherheit



(we already deployed that one to prevent „Cloud speed“ issues)

Setup Part 2 – (malicious) AppLocker Policy

AppLocker – Defining Trusted App for Execution

bi-sec
IT- & Informationssicherheit

- An AppLocker policy for executables and dlls is required to define custom Managed Installers like the **Intune Management Agents**
  - Creation of the Managed Installer rule is not possible via GPO Editor / PowerShell
    - We need to change the plain XML App Locker rules file …
  - Only trusted signed .exe files can be defined as Managed Installer
- Microsoft provides the PowerShell command to help:
  - Get-ChildItem ${env:ProgramFiles(x86)}'\Microsoft Intune Management Extension\Microsoft.Management.Services.IntuneWindowsAgent.exe' | Get-AppLockerFileInformation | New-AppLockerPolicy -RuleType Publisher -User Everyone -Xml > AppLocker_MI_PS_ISE.xml
    - Take care of OS language for groups when using the PowerShell command
      - The RuleCollection itself will contain the SID UserOrGroupSid="S-1-1-0“ later

AppLocker – Building a Managed Installer Policy

bi-sec
IT- & Informationssicherheit

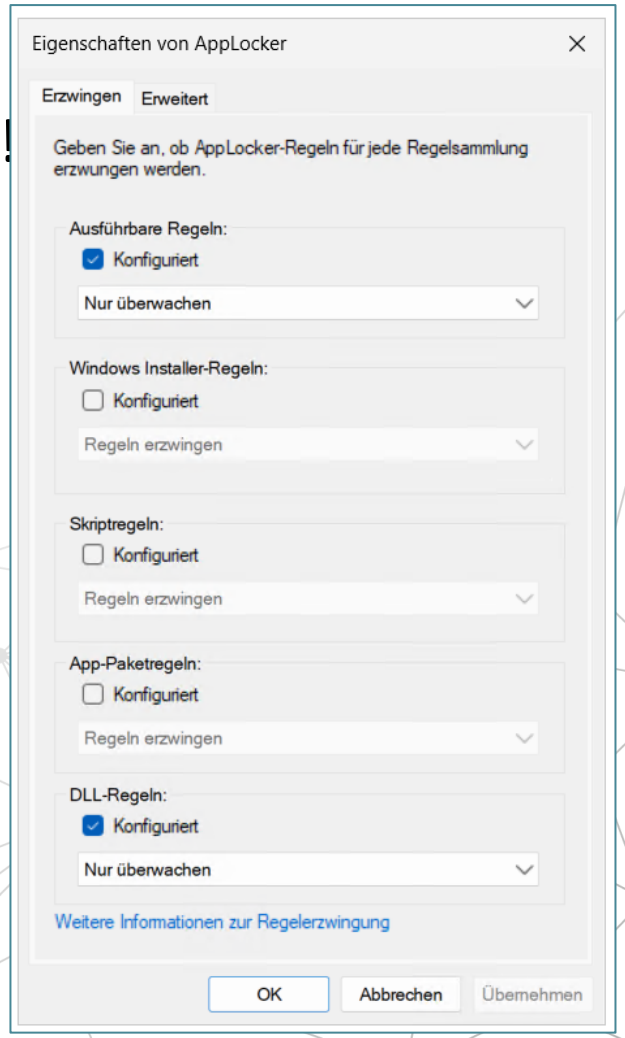
- Resulting Policy for trusted execution must be changed to be a Managed Installer Policy:
  - <AppLockerPolicy Version="1"><RuleCollection Type="Exe" EnforcementMode="NotConfigured"><FilePublisherRule Id="8e347ce8-a441-4224-bb7b-0b006237273f" Name="MICROSOFT.MANAGEMENT.SERVICES.**INTUNEWINDOWSAGENT.EXE**, Version 1.96.154.0 genau, aus MICROSOFT® INTUNE™, von O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US" Description="" UserOrGroupSid="S-1-1-0" Action="Allow"><Conditions><FilePublisherCondition PublisherName="O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US" ProductName="MICROSOFT® INTUNE™" BinaryName="MICROSOFT.MANAGEMENT.SERVICES.INTUNEWINDOWSAGENT.EXE"><BinaryVersionRange LowSection="1.96.154.0" HighSection="1.96.154.0" /></FilePublisherCondition></Conditions></FilePublisherRule></RuleCollection></AppLockerPolicy>
  - <AppLockerPolicy Version="1"><RuleCollection Type="ManagedInstaller" EnforcementMode="AuditOnly"><FilePublisherRule Id="8e347ce8-a441-4224-bb7b-0b006237273f" Name="MICROSOFT.MANAGEMENT.SERVICES.**INTUNEWINDOWSAGENT.EXE**, Version 1.96.154.0 genau, aus MICROSOFT® INTUNE™, von O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US" Description="" UserOrGroupSid="S-1-1-0" Action="Allow"><Conditions><FilePublisherCondition PublisherName="O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US" ProductName="MICROSOFT® INTUNE™" BinaryName="MICROSOFT.MANAGEMENT.SERVICES.INTUNEWINDOWSAGENT.EXE"><BinaryVersionRange LowSection="1.96.154.0" HighSection="1.96.154.0" /></FilePublisherCondition></Conditions></FilePublisherRule></RuleCollection></AppLockerPolicy>

AppLocker - Deployment

bi-sec
IT- & Informationssicherheit

- Import XML policy to GPO
  - The Managed Installer configuration is not visible in the UI (!
  - Only the two dummy deny rules are visible
    - Gpresult / Registry will show the real policies
  - Executable and dll rules can run in audit mode
    - No need to enforce anything
- Or - as an attacker - deploy the regkey locally
  - Regkeys can be created and exportet anywhere

.. done with AppLocker for now.

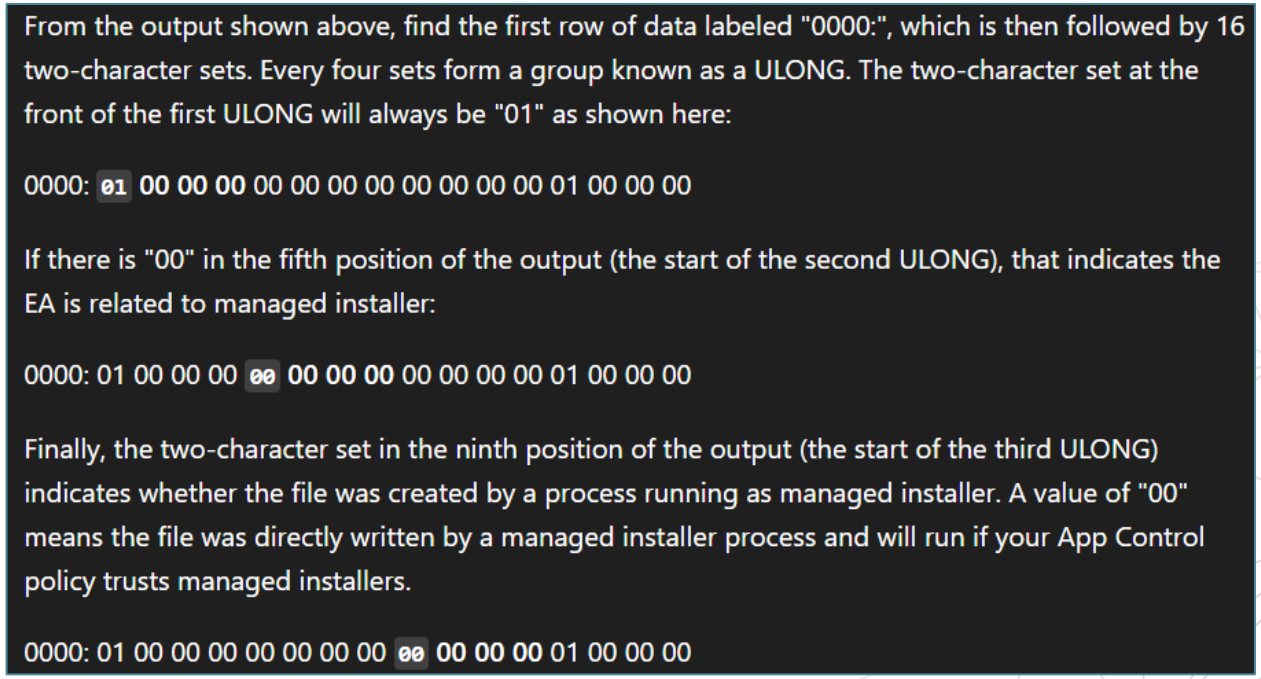


Setup done – Understand „Trusted“ Installer

Managed Installer - File Attribute

bi-sec
IT- & Informationssicherheit

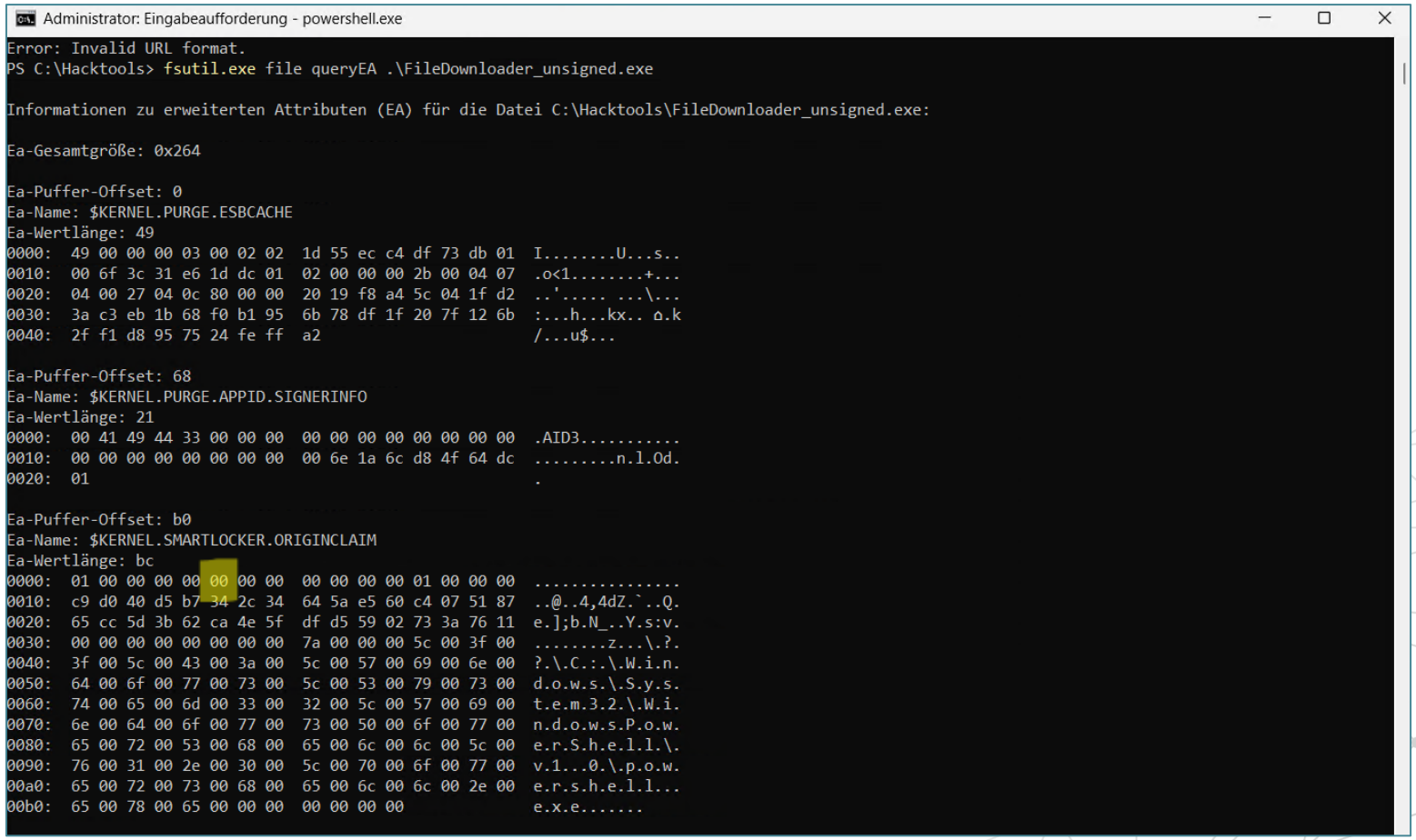
- Finding out if a file was written / installed using Managed Installer is … quiet uncomfortable
  - fsutil.exe file queryEA *filename*



https://learn.microsoft.com/en-us/windows/security/application-security/application-control/app-control-for-business/operations/configure-appcontrol-managed-installer

Managed Installer – View attributes with fsutil

bi-sec
IT- & Informationssicherheit



Managed Installer – can be defined

- The Managed Installer was defined via AppLocker
  - Managed Installer must be digitally signed
  - Signature must be trusted by Windows - $ 300 for Code-Signing-Cert

or …

```
PS C:\bi-sec\OD\bi-sec GmbH\Schulung-Windows - General\Infos-Todos-Beschreibungen\apppackage> Get-ChildItem "C:\Windows\
System32\WindowsPowerShell\v1.0\powershell.exe" | Get-AppLockerFileInformation | New-AppLockerPolicy -RuleType Publisher
 -User Jeder -Xml > AppLocker_MI_PS_ISE.xml
```

```
<RuleCollection Type="Exe" EnforcementMode="NotConfigured">
    <FilePublisherRule Id="c9730924-234a-4a5e-bfa3-fe365d1236cc" Name="POWERSHELL.EXE, Version 10.0.26100.5074 genau, aus MICROSOFT® WINDOWS® OPERATING SYSTEM, von O=MICROSOFT
    CORPORATION, L=REDMOND, S=WASHINGTON, C=US" Description="" UserOrGroupSid="S-1-1-0" Action="Allow">
        <Conditions>
            <FilePublisherCondition PublisherName="O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US" ProductName="MICROSOFT® WINDOWS® OPERATING SYSTEM" BinaryName=
            "POWERSHELL.EXE">
                <BinaryVersionRange LowSection="10.0.26100.5074" HighSection="10.0.26100.5074"/>
            </FilePublisherCondition>
        </Conditions>
    </FilePublisherRule>
</RuleCollection>
```

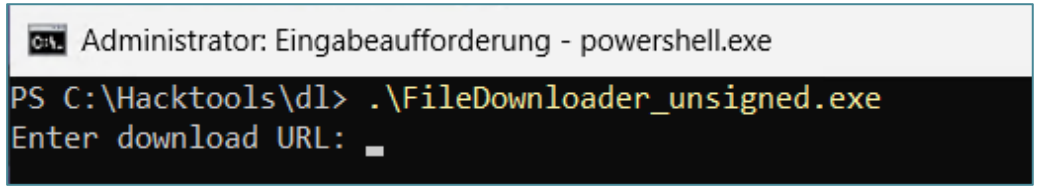
Let the magic happen

Conlusion

Abusing AppLocker to bypass App Control

bi-sec
IT- & Informationssicherheit

- Everything that the Managed Installer touches can be executed



- The definition of a Managed Installer happens via GPO / Registry
  - Every Administrator / attacker with admin privileges can write those
  - Secure Boot / EFI / App Control Signatures are basically ineffective
- Built-in tools like PowerShell are excellent Managed Installers
  - At least for an attacker, including Invoke-Webrequest..

Microsoft Learn says ...

bi-sec
IT- & Informationssicherheit

Security considerations with managed installer

Since managed installer is a heuristic-based mechanism, it doesn't provide the same security guarantees as explicit allow or deny rules do. Managed installer is best suited where users operate as standard user, and where all software is deployed and installed by a software distribution solution such as MEMCM.

Users with administrator privileges, or malware running as an administrator user on the system, may be able to circumvent the intent of your App Control policies when the managed installer option is allowed.

If a managed installer process runs in the context of a user with standard privileges, then it's possible that standard users or malware running as standard user may be able to circumvent the intent of your App Control policies.

Some application installers may automatically run the application at the end of the installation process. If the application runs automatically, and the installer was run by a managed installer, then the managed installer's heuristic tracking and authorization will extend to all files that are created during the first run of the application. This extension could result in unintentional authorization of an executable. To avoid that, ensure that the method of application deployment that is used as a managed installer limits running applications as part of installation.

The one Rule to ruin them all

bi-sec
IT- & Informationssicherheit

<Rule>

<Option>Enabled:Managed Installer</Option>

</Rule>

Don‘t use it!

Thank you (and good luck!)

bi-sec
IT- & Informationssicherheit

LinkedIn

OUR CONTACT

bi-sec
IT- & Informationssicherheit

christian.biehler@bi-sec.de

+49 7134 53 439 62

www.bi-sec.de

bi-sec GmbH
Am Autobahnkreuz 2A, 74248 Ellhofen
Tel: (+49) 7134 53439-60 I www.bi-sec.de I info@bi-sec.de
Geschäftsführung: Christian Biehler
Amtsgericht Stuttgart | Handelsregister: HRB771205
Steuer-Nr.: 65201/61219 I USt-ID: DE326708757
Commerzbank I IBAN: DE53 7004 0041 0587 1546 00 I BIC: COBADEFFXXX