Some Shall Pass

Common missteps in application control

@rustla

Hi

I'm Russ

I'm a Penetration Tester at Trustwave in Perth

Application Control Experience:

- Deployment and config
- Assessing config

Summary

Application Control Overview

- What is Application Control?
- Method Overview and Details

Weaknesses and Remediation

- Common Weaknesses or Misconfigurations
- Demonstrations
- Remediations

Application Control?

Application Control prevents code from running unless it's explicitly approved (Allow Listing)

Can also be used to enforce SOE (e.g. block Spotify)

Today will be looking at Windows implementations

Allow List Methods

Three Methods:

- Cryptographic Hashes
- Publishers
- File Paths

Policies may consist of all three

Hashes

Hash of the file contents are added to the allow list

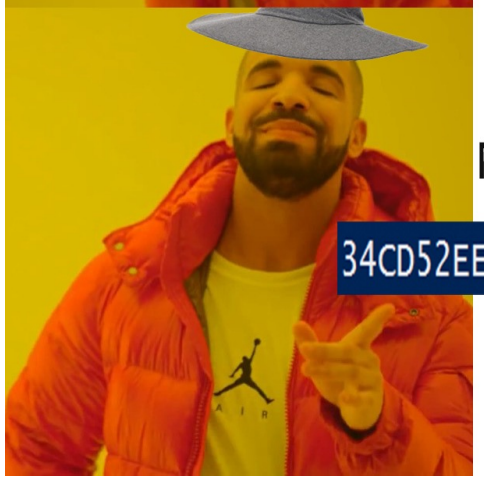
Changes to the file contents mean the hash changes and the file won't run

Useful for SOE images or published apps (e.g. SCCM)

Allow: 34CD52EEDA44A3F616EB52FE38B54529
Filename: regedit.exe

Name: regedit.exe

AC1FBAD43D4E8EA8B8DE614EEC66807C



Name: whatever.exe

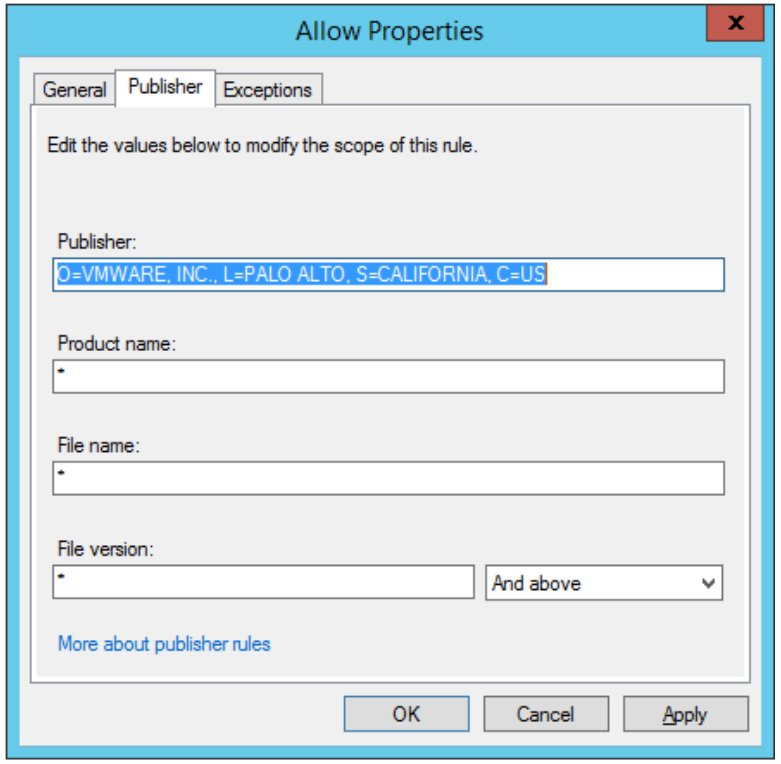
34CD52EEDA44A3F616EB52FE38B54529

Publishers

Applications and libraries can be digitally signed

The publisher (e.g. VMware) that signs the applications can be added to the allow list

Useful for signed files that change frequently
e.g. OS and browsers



File Paths

Executables matching file path and name added to the allow list

Wildcards can be used to trust all subfolders and files

Useful for scripts that change frequently (e.g. CI/CD, sysvol scripts)

Exploiting File Path Rules

Goal is to prevent non-privileged users running unapproved code

- Admins may uninstall endpoint controls or stop services

Wildcards in user-writable folders can be exploited

Example user-writable folder found during policy config review
C:\ProgramData*

Other user-writable folders by default include C:\Windows\Tasks

Example File Path Rule

Truncated output from:
(get-applockerpolicy -effective).RuleCollections

```
PathConditions      : {C:\ProgramData\*}
PathExceptions      : {}
PublisherExceptions : {}
HashExceptions      : {}
Id                  : 05472081-8f91-45dc-ae8e-74a0b3875c81
Name                : Additional allowed path: C:\ProgramData\*
Description         : Allows Everyone to execute from C:\ProgramData\*
UserOrGroupSid      : S-1-1-0
Action              : Allow
```

S-1-1-0 is all users

Demo Time

Mitigating File Path Rule Exploitation

Use hashes or publisher rules where you can

Use file path rules as a last resort

Avoid path rules for user-writable folders

Check user-writable folders regularly

Attack Scenarios

File Path Use Case

- Find user-writable folders in allow list policy
- Run any executable (e.g. mimikatz)
- Use mimikatz to access saved user creds (e.g. remote desktop) to move laterally

Exploiting LOLBins

Using signed and/or trusted Microsoft binaries to execute unapproved code

Living Off the Land Binaries (LOLBins)

- lolbas-project.github.io

Affects trusted signed files, hashes and paths

Example LOLBin – MSBuild.exe

Demo Time

Mitigating LOLBin Exploitation

Implement “Microsoft recommended block rules”

Blocklist of known LOLBins

Essential Eight Maturity Level Three Requirement

Be careful – log, monitor, tailor to your environment

Microsoft recommended block rules

08/23/2021 • 27 minutes to read • +12

Applies to:

- Windows 10
- Windows 11
- Windows Server 2016 and above

Note

Some capabilities of Windows Defender Application Control are only available on specific Windows versions. Learn more about the **Defender App Guard feature availability**.

Members of the security community* continuously collaborate with Microsoft to help protect customers. With the help of their valuable reports, Microsoft has identified a list of valid applications that an attacker could also potentially use to bypass Windows Defender Application Control.

Unless your use scenarios explicitly require them, Microsoft recommends that you block the following applications. These applications or files can be used by an attacker to circumvent application allow policies, including Windows Defender Application Control:

- addinprocess.exe
- addinprocess32.exe
- addinutil.exe

Attack Scenarios

File Path Use Case

- Find user-writable folders in allow list policy
- Run any executable (e.g. mimikatz)
- Use mimikatz to access saved user creds (e.g. remote desktop) to move laterally

LOLBin Use Case

- Execute a C# executable (e.g. Rubeus) using MSBuild
- Compromise AD service accounts with Rubeus
- Or run SharpHound (also C#) to find paths to DA

Exploiting Interpreters

Interpreted (e.g. PowerShell) scripts can be allow listed

Some implementations require scripts to touch the disk

Invoke-Expression (iex) is your friend

Demo Time

Mitigating Interpreter Exploitation

Role-based policy – only allow Python to run for dev users

Modern PowerShell features:

- Script block logging
- Constrained Language Mode (enabled in PS 5.1 when AppLocker Script Rules configured?)
- Antimalware Scan Interface (AMSI) – endpoint protection visibility

Block PowerShell v2 to prevent bypassing modern features

Attack Scenarios

File Path Use Case

- Find user-writable folders in allow list policy
- Run any executable (e.g. mimikatz)
- Use mimikatz to access saved user creds (e.g. remote desktop) to move laterally

LOLBin Use Case

- Execute a C# executable (e.g. Rubeus) using MSBuild
- Compromise AD service accounts with Rubeus
- Use Seatbelt (also C#) to look for interesting DPAPI creds

Interpreter Use Case

- Download PowerShell script (e.g. PowerUp) to memory and execute
- Requires CLM disabled or PS v2 enabled
- Identify local privesc opportunities using PowerUp
- Search for user writable weaknesses in policy

What Next?

Those Using Allow Listing

- Review your config for improvements
- Implement LOLbin block rules where you can

For The Curious

- Today's (Vulnerable) policy published on GitHub
- It's a playground, not production
- Apply the policy on a VM (Windows Server recommended)
- Admins group can run anything

Questions?

Reach out: @rustla

References:

- lolbas-project.github.io
- github.com/bohops/GhostBuild

AppLocker Policy:

- github.com/rustla/AppLockerLab