# AppLocker known issues

16 December 2018

This document lists issues observed during AaronLocker deployments and offers workarounds to apply until the issues are resolved.

### Microsoft Edge browser crashes on start when DLL rules are configured

**Symptom:** if you have AppLocker DLL rules configured on Windows 10 versions 1703, 1709, or 1803, Microsoft Edge crashes shortly after starting. The problem occurred whether DLL rules were configured for “Audit only” or “Enforce rules.”

**Resolution:** Ensure the affected version of Windows 10 is fully patched. Patches for each affected version were released by 30 August 2018. The fixes first appeared in the following updates:

For Windows 10 v1703: <https://support.microsoft.com/help/4343889>  
For Windows 10 v1709: <https://support.microsoft.com/help/4343893>  
For Windows 10 v1803: <https://support.microsoft.com/help/4340917>

Newer updates also include the fixes.

**Additional steps:** If you had earlier applied a workaround involving a registry edit, delete these registry values to ensure that you aren’t disabling any important security features:

|  |  |
| --- | --- |
| Key | HKEY\_LOCAL\_MACHINE\Software\Microsoft\Internet Explorer\Spartan |
| Value name | RAC\_LaunchFlags |
| Key | HKEY\_CURRENT\_USER\Software\Microsoft\Internet Explorer\Spartan |
| Value name | RAC\_LaunchFlags |

### Files are blocked when they shouldn’t be

On rare occasions, AppLocker blocks a file that it should allow, and the event data incorrectly reports the file as not signed. In addition, AppLocker caches the incorrect result indefinitely and never allows the file to run. This is because of a bug that is still being researched. Two workarounds are described here.

**Workaround 1:** One workaround is to copy the file to a new name in the same directory, delete the original file, then rename the copy to the original name. These steps invalidate the cached result for that file, so the next time the file is referenced, AppLocker will reevaluate it and most likely do so correctly.

Example:

OneDrive fails to start. AppLocker event log reports an error with TELEMETRY.DLL and shows the file as not signed. Error information captured with Get-AppLockerEvents.ps1:

GenericPath : %LOCALAPPDATA%\MICROSOFT\ONEDRIVE\17.3.6816.0313\TELEMETRY.DLL

GenericDir : %LOCALAPPDATA%\MICROSOFT\ONEDRIVE\17.3.6816.0313

OriginalPath : %OSDRIVE%\USERS\TOBY\APPDATA\LOCAL\MICROSOFT\ONEDRIVE\17.3.6816.0313\TELEMETRY.DLL

FileName : TELEMETRY.DLL

FileType : DLL

PublisherName : -

ProductName :

BinaryName :

FileVersion :

Hash : 0xB2FD0EC99D98D89CEB30C45D47F5418AA70CCCF78FC22CC3EABEF6F6E67AA17A

UserSID : S-1-5-21-3841777977-1772892211-860544140-1002

UserName : DESKTOP-L0DMFHV\Toby

MachineName : DESKTOP-L0DMFHV

EventTime : 2018-06-25T09:46:18.7067597

PID : 3476

EventType : Error

However, Test-AppLockerPolicy says that current policy should allow the file:

PS C:\> Test-AppLockerPolicy -PolicyObject (Get-AppLockerPolicy -Effective) -Path "C:\Users\Toby\AppData\Local\Microsoft\OneDrive\17.3.6816.0313\Telemetry.dll" | Format-List \*

FilePath : C:\Users\Toby\AppData\Local\Microsoft\OneDrive\17.3.6816.0313\Telemetry.dll

PolicyDecision : Allowed

MatchingRule : Microsoft OneDrive (partial): Signer/product rule for O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US/MICROSOFT ONEDRIVE

Implement the workaround:

copy .\Telemetry.dll .\Workaround.file

del .\Telemetry.dll

ren .\Workaround.file Telemetry.dll

**Workaround 2.** The second workaround is to disable AppLocker’s caching of “unsigned” results. Note however that this may cause performance degradation. To disable the result caching, configure the following registry value, and then reboot the computer:

[HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Control\Appid]

"IgnoreCacheUnsignedFiles"=dword:00000001

### Intel batch files

Example customer email:

*There is a batch file that we get a warning about when auditing AppLocker.*

*C:\Intel\GfxCPLBatchFiles\{A6D608F0-0BDE-491A-97AE-5C4B05D86E01}.bat*

Some Intel drivers install a service that creates randomly-named batch files in a user-writable C:\Intel subdirectory and then try to execute them as the interactive user. There is no good way for any application whitelisting technology to allow the file to run. Intel has fixed the problem with a recent update:

<https://downloadcenter.intel.com/download/27680>

Discussion about the issue:

<https://communities.intel.com/thread/119638?start=45&tstart=0>