

Application Control 2018.3

Release Notes

Components in this Release

Component	Version
Application Manager Agent	10.2.86.0
Application Manager Console	10.2.86.0
Analysis Server & Web Console	10.2.86.0

What's new?

We've reviewed the Product Ideas that you've submitted via the [Ideas Portal > Security Ideas > Application Control](#) on the Community Website and we've addressed several of these in the 2018.3 release. Please continue to submit your Product Ideas or vote on ideas that have already been submitted so that we get a better understanding of the improvements that you would like to see. Your feedback is important and we will endeavor to focus on the ideas that get the most votes as we plan each release.

Self-Authorization Usability Improvements

Self-Authorization is a logical step on the journey from Audit to Restricted modes. On the security slider, it sits right between the two.



When you've been in audit mode for a period of time and think you are ready to move to restricted, moving to the Self-Authorizing mode first provides the following benefits:

- From an administrator perspective, there is often a sense of nervousness moving from audit to restricted modes. While you've been monitoring the logs in audit mode and have created rules to cater for these log events, there may still be some concern that you've missed something and moving to restricted could result in an influx of support calls and impact to user productivity. Moving to self-authorizing mode ensures that users can authorize and applications that would otherwise be blocked and allows the administrator to continue to monitor the log events for any anomalies. If there are no log events (or a relatively small number of events), it means that the rules are correct and you are now confidently ready to move into restricted mode.
- From an end-user perspective, while in audit mode, Application Control has really had no impact on the user. They are not prevented from installing or running applications that would otherwise be blocked in restricted mode. However, in self-authorizing mode, these applications are presented to the user for them to make an authorization decision. This starts the education process that there are changes on the way and also helps to prevent the spread of malware. If the user authorizes an unknown executable which infects their endpoint, it only infects their endpoint. Other endpoints are unaffected unless those users also decide to authorize this file.

To date, the problem with self-authorizing mode has been that it presents **all** executable files for self-authorization that fail Trusted Ownership checking or are not covered by another rule. This includes not only the initial application executable itself. It also includes any subsequent dlls that are loaded by the application. Also, because these dlls are initially blocked while awaiting self-authorization, it can cause the application to crash, even though the dll has been authorized by the user. The user must then restart the application and could potentially go through this exercise multiple times before the application is fully authorized and functional.

So, rather than being a positive experience in a transition from audit to restricted modes, self-authorization has been somewhat of a negative experience for customers.

We've tackled this issue in the 2018.3 release and self-authorization is now much improved. Once the initial application exe is self-authorized, any subsequent child dlls are automatically authorized so that applications can be self-authorized with just a single click (or maybe a couple of clicks in some cases). This will be a much better experience for both administrators and end-users and really allow self-authorization to be that transitional step between audit and restricted modes. Try it out and tell us what you think.

Option to silently block executables

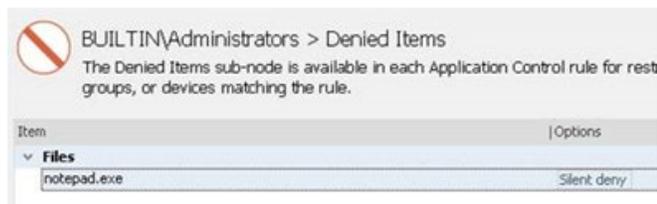
Certain executables, such as application updaters or driver updaters, are often intentionally blocked by administrators because they don't want these executables to make updates on-the-fly and have untested or unauthorized application versions installed on endpoints. Instead, they prefer to push these updates out centrally.

These updaters typically run on a schedule (e.g. once a week) and, if they are blocked by Ivanti Application Control, this will cause an access denied message box to be presented to the end user. The user then has to dismiss this message. This experience is confusing for the user and can result in unnecessary support calls.

In 2018.3, we've added an option, when creating a rule, to not show an access denied message when a blocked executable is denied. The "Silent deny" option will also be displayed in the Options column when viewing the list of rules so that you can easily see which rules have the silent deny option enabled.

Options

- Do not show access denied message when denied
- Ignore Audit Event Filtering



Disable vs Remove Configuration Rule

When troubleshooting a rule, the only option available up to now has been to remove the rule to determine whether it is the source of the problem. If it turns out that the rule was not the issue, the administrator then has to add the rule back to the configuration. This contains the risk that the rule will be misconfigured when it is being restored if the administrator doesn't remember exactly which options were selected.

In 2018.3, we've added a right-click option to disable a rule which should prove very useful for troubleshooting. The disabled rule will be greyed out as shown below and can easily be re-enabled via right-click.



Per-item auditing

Turning on logging for 9001 events can result in large volumes of log events which could quickly overwhelm your database depending on the scale of your implementation. However, over time, the AC configuration can become a bit unwieldy and it can sometimes be unclear why a particular rule was added and whether it is even being used any longer. The rule may have been added by a previous administrator who has since left the company. What would be really useful is to understand if this rule is being used but, I can't enable logging just for this rule.

So, we've changed this in 2018.3, whereby logging can be enabled for individual rules. We've done this by adding an option to ignore the audit event filtering (which is used to determine for which file types (e.g. *.exe) events get raised when you select an event ID (e.g. 9001) to raise). In the example below, we've checked the 9001 checkbox in the Auditing dialog (Manage menu):

Select events to raise

Check the events you want to raise locally. Central event filtering defined with the Ivanti Management Console takes priority over this selection in a centralized deployment scenario.

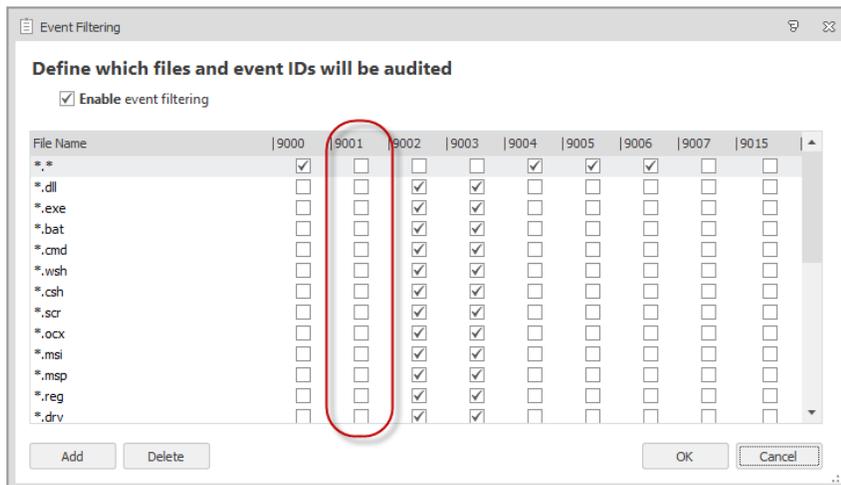
[Toggle selected](#)

ID	Name	Description	Log Locally
9000	Denied execution	Denied execution request.	<input type="checkbox"/>
9001	Allowed execution	Allowed execution request.	<input checked="" type="checkbox"/>
9002	Overwrite changed owner	Overwrite of an allowed executable.	<input type="checkbox"/>
9003	Rename changed owner	Rename of a denied executable.	<input type="checkbox"/>
9004	Application limit denial	Application limit denial.	<input type="checkbox"/>
9005	Time limit denial	Time limit denial.	<input type="checkbox"/>

Warning! You have selected a high volume event



Although this dialog provides a warning when you select 9001 events, the event filtering dialog is used to define which filetypes the 9001 events should be raised for. The default for 9001 events is that no filetypes are selected, in which case no 9001 events will be raised. This is how this feature has always worked.



In 2018.3, we've added a new option to ignore Audit Event filtering. When this option is checked for a specific rule, it means that if an event ID (e.g. 9001) is selected on the Auditing dialog, this event will be raised for this rule regardless of the event filtering settings. So, even if no filetypes are selected, as in the example above, 9001 events will still be raised for this rule.

As in the case of the Silent Deny feature, the Ignore Event Filtering option will also be displayed in the Options column when viewing the list of rules so that you can easily see which rules have this option enabled.



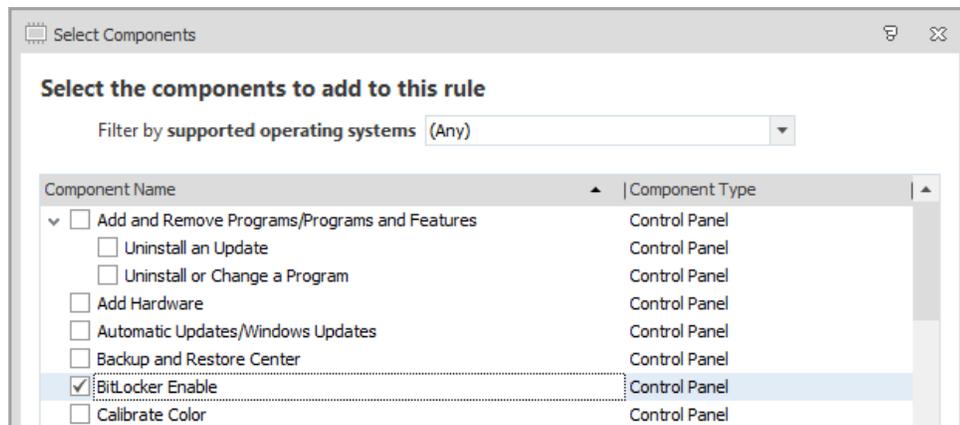
Add Network Port variable to AC message box

In previous releases, when a network port was blocked, the AC message box did not show which port was blocked. While this information was included in the log event, it made troubleshooting issues more difficult as, rather than simply extracting the information from the message box, the administrator would have to gather and search through the audit logs to find the port number.

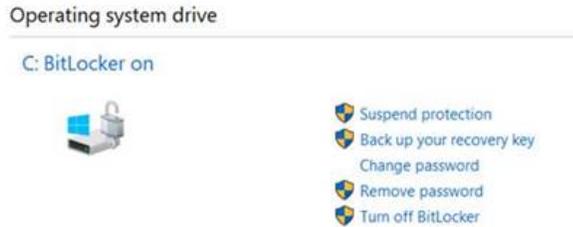
In 2018.3, we've added the port number to the message box when a network port is blocked so this information is more readily available.

Extended BitLocker support

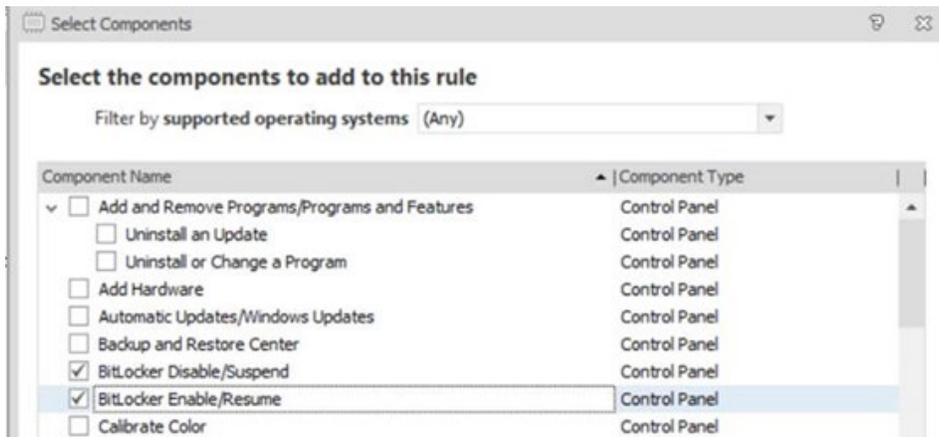
The Privilege Management feature set provides an option to allow BitLocker to be enabled.



However, once it has been enabled, there are additional options available within the Control Panel to suspend or disable BitLocker. This is sometimes necessary to install or update applications.



However, up to now, it has not been possible to provide these privileges to users via Ivanti Application Control. In 2018.3, we've added an additional option under the User Privileges > Components tab to disable or suspend BitLocker. We've also updated the Enable option to Enable/Resume. Customers now have more granular control over BitLocker.



Bugs Fixed

The following customer support issues have been resolved in this release:

ID	Title	Details
14241	AsModLdr.sys causing BSoD with Device Guard	DRIVER_VERIFIER_DETECTED_VIOLATION When enabling Device Guard Virtualization Based Protection of Code Integrity, this enabled Driver Verified and can blue screen the box citing AsModLdr as the cause.
64394	File Owner reported as '<Unknown>' in audit data - and missing in Rules Analyzer	When AC is in audit-only mode and is unable to read the file descriptor for file ownership, it tries a second time without impersonation. This succeeds but the event is still raised as with an unknown file owner.
64617	Audit logs for "9000" Events (Access Denied) sometimes miss the Parent Process	Audit events for "9000" Events may be missing Parent Process meta data, such as Parent Process Name. This occurs if the target file does not contain "File Version" info under the file's property -> details tab.
64668	Denied Active Setup processes can be launched from within a XenApp published application	If the Advanced Setting "Ignore restrictions during Active Setup" is enabled and one of the processes listed within Active Setup is also added to the Process Denied list e.g. cmd.exe. It is possible to still launch the process within a XenApp published Application.
64829	PowerShell Script validation does not work if PowerShell v2.0 is removed	PowerShell script validation does not work when PowerShell v2.0 is removed on Windows 10. Removing PowerShell v2.0 still leaves PowerShell v5.0 installed
64883	Policy Change Request items do not appear until Scripted Rules have completed.	Policy Change Request items to not appear (e.g. Desktop Icon, Right Click context menu) until Scripted Rules have completed. This occurs even if the Scripted Rule does not have any Policy Change Request items enabled.
65587	PowerShell scripted rule validation does not work for 32bit processes on a 64bit system	When "Validate PowerShell scripts" is enabled, PowerShell scripts launched by a 32bit process on a 64bit system are not validated. This results in the script being denied from running.
65630	AC prevents Microsoft Patch (.msp) files from running with Self-Authorizing	When Applying an MSP while in self-authorising mode the windows installer usage screen is displayed instead of the installer.
65809	Event ID 5038 raised events as AMLdrApplnit.dll fails Code Integrity checking	Event 5038 is repeatedly raised on Windows 10 endpoints (and potentially 2K16) indicating AMLdrApplnit.dll is failing Code Integrity checking
66211	Console Exception on Save running as non-privileged user	When attempting to save a configuration to the Management Center in the Application Control Console as a standard user, the following error is displayed and the Application Control Console may crash. An error occurred while discovering the Custom Rule condition schemas System.UnauthorizedAccessException: Access to the path 'C:\ProgramData\AppSense\User SxS assemblies' is denied.
66503	VMware View desktops hang on startup after deploying Application Control	VMware View desktops hang on startup after deploying the Application Control agent.

Known Issues and Limitations

The known issues and limitations are detailed on our [Community site](#).

Supported Operating Systems and Technologies

The supported Operating Systems and Technologies are now detailed in the [Application Control Maintained Platforms Matrix](#).

Required Utilities and Components

Component	Required Utilities and Components
Console	Microsoft Windows Installer 5.0 Microsoft .NET Framework 4.6.x Microsoft Visual C++ 2017 Redistributable Package
Agent	Microsoft Windows Installer 5.0 Microsoft Visual C++ 2017 Redistributable Package
Web Service	Microsoft Windows Installer 5.0 Microsoft Visual C++ 2017 Redistributable Package Microsoft .NET Framework 4.6.x

Note: Please ensure that your endpoints are fully up to date with the latest patches from Microsoft. See [Document Number: 43407](#) for more details.

Further Help and Information

Information about installing, configuring, and using Application Control is available from our [product help](#).