Carbon Black App Control and ASD ACSC Essential 8

Australian Signals Directorate | Australian Cyber Security Centre | Application Control | Compliance Overview

Purpose of this document is to provide a comprehensive Carbon Black App Control implementation and configuration guide to comply with ASD ACSC Essential 8.

Carbon Black App Control is used to lock down endpoints, servers, and critical systems, prevent unwanted changes, and ensure continuous compliance with regulatory mandates, including the application control mitigation strategy of the ASD ACSC Essential 8.

This How-To Guide offers a comprehensive outline of the procedures an organisation can adopt to reach Maturity Level 1 (ML1), ML2, and/or ML3 of ASD ACSC's Essential 8 Mitigation strategy for application control. By adhering to this guide, users of Carbon Black App Control can formulate and implement a systematic plan for the gradual implementation of Carbon Black App Control at High Enforcement throughout their production environment.

For more operational tutorials on Carbon Black App Control, see the Carbon Black App Control Activity Path on Tech Zone.

Note: This guide is not intended as and should not be used as a replacement for a detailed implementation project plan. Carbon Black App Control Deployment service can enable Maturity Levels 1, 2, and 3 within your environment.

Essential Eight

The Essential Eight Maturity Model, first published in June 2017 and updated regularly, supports the implementation of the Essential Eight. It is based on ASD's experience in producing cyber threat intelligence, responding to cyber security incidents, conducting penetration testing and assisting organisations to implement the Essential Eight.

The mitigation strategies that constitute the Essential Eight are:

1. Patch applications
2. Patch operating systems
3. Multi-factor authentication
4. Restrict administrative privileges
5. **Application control**
6. Restrict Microsoft Office macros
7. User application hardening
8. Regular backups

Essential 8 Application Control

Application control is one of the most effective mitigation strategies in ensuring the security of systems. Application control is a security approach designed to protect against malicious code (also known as malware) executing on systems. When implemented robustly, it ensures only approved applications (e.g. executables, software libraries, scripts, installers, compiled HTML, HTML applications, control panel applets and drivers) can be executed.

While application control is primarily designed to prevent the execution and spread of malicious code, it can also prevent the installation or use of unapproved applications.

Implementing application control involves the following high-level steps:

- Identifying approved applications
- Developing application control rules to ensure only approved applications are allowed to execute

- Maintaining application control rules using a change management program
- Validating application control rules and their implementation on an annual or more frequent basis.

When determining how to enforce application control, the following methods are considered suitable if implemented correctly:

- Cryptographic hash rules
- Publisher certificate rules (combining both publisher names and product names)
- Path rules (ensuring file system permissions are configured to prevent unauthorised modification of folder and file permissions, folder contents and individual files).

Carbon Black App Control supports all of the aforementioned recommendations and with even more advanced features such as Yara based automation.

Note: Carbon Black App Control and Carbon Black Cloud offer other features that support Essential 8 as compensating control, the following responses are only for Application Control, reach out to your friendly Carbon Black Security Architect to discuss full capability to support defence in depth.

Carbon Black App Control ASD Essential 8 Implementation

These answers are modelled off the November 2023 revision of the Essential 8 Maturity Model.

Note: Please note that higher level requirements are in addition to lower levels. For example to reach maturity level two you must also meet the requirements of maturity level one.

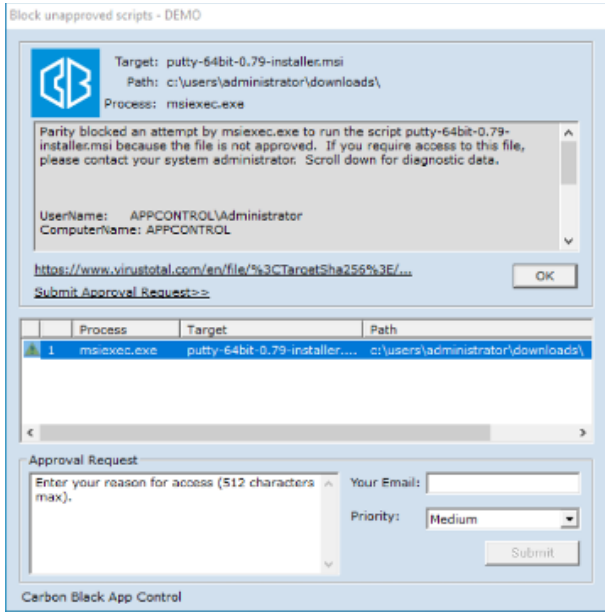
Application control	**Application control is implemented on workstations.**	Application control is implemented on workstations.	Application control is implemented on workstations.
	–	**Application control is implemented on internet-facing servers.**	Application control is implemented on internet-facing servers.
	–	–	**Application control is implemented on non-internet-facing servers.**
	Application control is applied to user profiles and temporary folders used by operating systems, web browsers and email clients.	Application control is applied to user profiles and temporary folders used by operating systems, web browsers and email clients.	Application control is applied to user profiles and temporary folders used by operating systems, web browsers and email clients.
	–	**Application control is applied to all locations other than user profiles and temporary folders used by operating systems, web browsers and email clients.**	Application control is applied to all locations other than user profiles and temporary folders used by operating systems, web browsers and email clients.
	Application control restricts the execution of executables, software libraries, scripts, installers, compiled HTML, HTML applications and control panel applets to an organisation-approved set.	Application control restricts the execution of executables, software libraries, scripts, installers, compiled HTML, HTML applications and control panel applets to an organisation-approved set.	Application control restricts the execution of executables, software libraries, scripts, installers, compiled HTML, HTML applications and control panel applets to an organisation-approved set.
	–	–	**Application control restricts the execution of drivers to an organisation-approved set.**
	–	**Microsoft's recommended application blocklist is implemented.**	Microsoft's recommended application blocklist is implemented.
	–	–	**Microsoft's vulnerable driver blocklist is implemented.**
	–	**Application control rulesets are validated on an annual or more frequent basis.**	Application control rulesets are validated on an annual or more frequent basis.

Example excerpt from Appendix D: Comparison of maturity levels

Automation and Interactive Approvals

Managing application control in end-user environments can be challenging due to frequent updates and diverse application requirements. Carbon Black App Control offers automation and interactive approval features to enhance efficiency in maintaining Essential 8 without hindering users' ability to carry out their tasks through automated workflows and approval processes.

Enable user involvement in policy tuning. Customise the pop-up notifier entirely; adjust the logo and text to align with your company policies and language preferences. Furthermore, request justifications from users during the approval request process.



Example App Control Pop. Simple User Interactions Pause or block applications from running Request input from user on what is changing and why Allow local approval (by policy.)

Approvals allow your users and applications to work without interruption, and when configured correctly allows for manual and automatic updates to approved applications.

Note: There are a number of additional approval methods not listed below. To review all options for approvals please refer to the Carbon Black App Control.

Note: Best practice is to use approvals and file creation control rules rather than execution control rules where possible. Execution control rules should only be used when other options are not viable.

Deployment

Carbon Black recommends grouping systems by the types of applications that will typically be run on those systems, for example, your Marketing users and your Developer users would typically run different types of software; and your Web Servers are different from your SQL servers or Exchange/mail servers.

It is recommended that you create a Low Enforcement and High Enforcement policy for each group of systems. This is because you want to put your systems in a monitoring mode first while you update the approval mechanisms unique to your environment, then move systems into High Enforcement in ever increasing waves once you're happy with the approval automation.

Moving a system from a Disabled policy to a Low Enforcement policy will initiate an "initialisation". The initialisation is a full disk inventory scan of all interesting files on that system. In the context of the Essential Eight Application Control mitigation strategy, interesting files include executables, software libraries, scripts, installers, compiled HTML, HTML applications and control panel applets.

The initialisation scan consumes CPU resources as it extracts file properties and hashes files (this process is similar to a typical Antivirus full disk scan in terms of performance impact). Therefore, it is recommended that you move systems from a Disabled policy to a Low Enforcement (monitor) policy in ever-increasing waves and, for critical servers, to do this during a scheduled window with consideration to your change control policy for critical systems.

Enforcement Levels

Enforcement level is the protection applied on a computer running the App Control Agent, the following diagram provides recommend process to implement enforcement levels and below table describes each enforcement level:

Example high-level deployment process

The basic process is to configure approvals that you already know about, for example enable SCCM approvals if you're using SCCM for software and patch deployment. Then add the required script types that are not enforced by default. While doing those two steps deploy the agent, ideally in Disabled mode. Then move a pilot group of systems to a Low Enforcement policy, monitor for unapproved files and make any required adjustments to automatic approvals. Repeat that process until you've configured all required automatic approvals. During this process you can start to move an ever-increasing wave of systems from Disabled to Low Enforcement policies. Once you're confident that you've configured all required automatic approvals then move systems into High Enforcement, at which point systems are in the desired Application Control mode.

High (Block Unapproved Files)	High enforcement level offers the highest protection by allowing only explicitly approved applications to run on computers. It is suitable for systems with static application configurations like servers or single-purpose systems. For computers with more dynamic configurations, High enforcement can still be utilized by pre-approving files through trusted directories, users, publishers, updaters, or reputation approvals. Additionally, any files existing on computers before installing the Carbon Black App Control Agent are locally approved to run under High enforcement, except those already identified and banned on the server.
Medium (Prompt Unapproved)	Medium enforcement provides a balanced approach to security, preventing unapproved files from executing without completely blocking them. It blocks all unapproved files from running but presents a dialog on client computers, allowing users to decide whether to run the file. If the user permits the file, it becomes locally approved and is always allowed to run on that computer. If an unapproved file is executed remotely and allowed by the user, it is temporarily approved for 14 days. Explicitly banned files are prohibited from running under Medium enforcement.
Low (Monitor Unapproved Files)	Low enforcement is suitable when only banning specific files is necessary without concern for unknown files. It blocks banned files while allowing the installation of both approved and unapproved software. While unapproved files are permitted to execute under low enforcement, they can be monitored, and if necessary, emergency lockdown measures can

	be applied.
None (Visibility)	Setting the Enforcement Level to None (Visibility) enables tracking of file activity without blocking it. In this mode, Carbon Black App Control's reporting and asset management features monitor executable file activity on computers, including drift reports, event reports, and file inventory, without enforcing any rules. It can serve as an initial step towards establishing a more controlled environment.
None (Disabled)	Choosing None (Disabled) mode stops all enforcement and tracking activities. When the agent is disabled for a computer, its file database is removed from the agent but retained on the server for one day. Computers in Agent Disabled mode reset their files when moved to a policy with another Enforcement Level. However, agents in None (Disabled) mode continue to monitor certain operations locally to prevent information gaps if reactivated later, although this typically has minimal resource impact unless a large number of writes are performed.

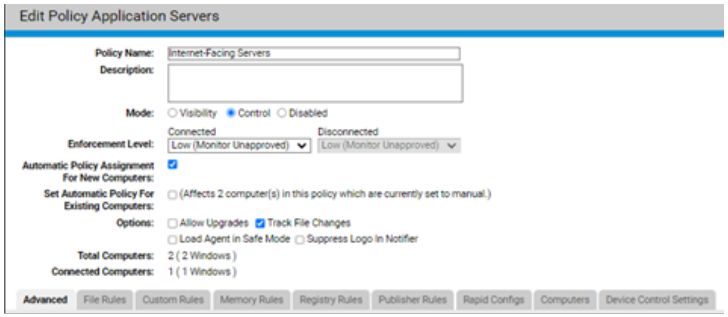
Essential 8 Maturity Levels

To assist organisations with their implementation of the Essential Eight, four maturity levels have been defined (Maturity Level Zero through to Maturity Level Three). With the exception of Maturity Level Zero, the maturity levels are based on mitigating increasing levels of tradecraft (i.e. tools, tactics, techniques and procedures) and targeting, which are discussed in more detail below. Depending on overall capability, malicious actors may exhibit different levels of tradecraft for different operations against different targets. For example, malicious actors capable of advanced tradecraft may use it against one target while using basic tradecraft against another. As such, organisations should consider what level of tradecraft and targeting, rather than which malicious actors, they are aiming to mitigate.

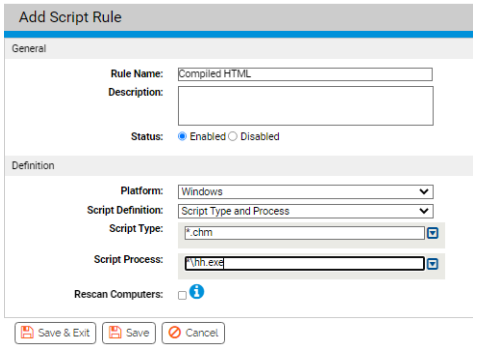
Organisations need to consider that the likelihood of being targeted is influenced by their desirability to malicious actors, and the consequences of a cyber security incident will depend on their requirement for the confidentiality of their data, as well as their requirement for the availability and integrity of their systems and data. This, in combination with the descriptions for each maturity level, can be used to help determine a target maturity level to implement.

Finally, Maturity Level Three will not stop malicious actors that are willing and able to invest enough time, money and effort to compromise a target. As such, organisations still need to consider the defence in depth detection and prevention strategies., reach out to your friendly Carbon Black Security Architect to discuss.

Maturity Level 1

Control	**Carbon Black App Control**
Application control is implemented on workstations.	App Control supports installation on workstation endpoints and makes it easy to administer application control from a single console. See App Control Agent Operating Environment Requirements and Windows Operating Systems and Respective Agents Deploy Carbon Black Sensor on all workstations in your environment including Windows, MacOSX and Linux. Policy Creation The following steps can be done prior to deploying any agents, during agent deployment or after agents have been deployed, assuming you have deployed agents using a disabled policy. 1. Go to Rules ->Policies and “click “Create Policy”. Use the below settings as a default.  Figure 4: New Policy creation window. a. Once the Low Enforcement policies have been created, move a pilot group of

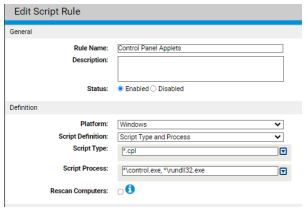
	systems for each policy into the respective Low Enforcement policy. i. This will result in those systems starting the one-time initialisation scan. This will utilise CPU while the agent crawls the system for files, generates hashes and collects file metadata. ii. Refer to the Carbon Black App Control User Guide for information on doing the initialisation scan once for VDI and servers cloned from a custom image. 2. Repeat the above step until you have created a Low Enforcement policy for every type of system. You can create High Enforcement policies later or repeat these steps, selecting “High Enforcement” for the Enforcement Level for both Connected & Disconnected.
Application control is applied to user profiles and temporary folders used by operating systems, web browsers, and email clients.	App Control performs application control on all folders (including user profiles and temporary folders) by default.
Application control restricts the execution of executables, software libraries, scripts, installers, compiled HTML, HTML applications and control panel applets to an organisation-approved set.	App Control enforces application control for all file types listed in the corresponding requirement. 1. Enable tracking of HTML applications. a. Go to Rules -> Software Rules -> Scripts b. Enable existing script rule for HTML Application (*.hta), by editing the rule and changing the Status from “Disabled” to “Enabled”. 2. Create a script rule to track and enforce compiled HTML files. a. Go to Rules -> Software Rules -> Scripts b. Create a new script rule: i. Rule Name: Compiled HTML ii. Status: Enabled iii. Platform: Windows iv. Script Type: *.chm v. Script Process: *\hh.exe



vi. *Example Compiled HTML Script Rule configuration.*

3. Create a script rule to track and enforce control panel applets.
    a. Go to Rules -> Software Rules -> Scripts
    b. Create a new script rule:
        i. Rule Name: Control Panel Applets
        ii. Status: Enabled
        iii. Platform: Windows
        iv. Script Type: *.cpl
        v. Script Process: *\control.exe

    Press enter after entering the above and then add: *\rundll.exe



Example Control Applet Rule configuration.

Maturity Level 2

Control	**Carbon Black App Control**
Application control is implemented on internet-facing servers.	This requirement is very straightforward. With the groundwork being performed when configuring Maturity Level 1. This requirement simply extends the type of endpoints to include having App Control installed and enforced on all Internet-facing servers in addition to all Workstations. To achieve this step, make sure you have policies configured to cater for the requirements of servers as covered for workstations in the Maturity Level 1 instructions, then deploy the agent to your Internet-facing servers and work towards High Enforcement.
Application control is applied to all locations other than user profiles and temporary folders used by operating systems, web browsers and email clients.	This requirement is an extension of the Maturity Level 1 requirement to apply application control to user profiles and temporary folders by extending those controls out to all other locations. As the ML1 requirement only applied to solutions that solely rely on path-based rules, (not those that use publisher and hash based rules), the requirement did not apply to users of Carbon Black App Control. If you have worked through the ML1 stages of this guide, you will already meet this requirement.
Microsoft's recommended application blocklist is implemented.	Carbon Black App Control allows you to configure these block rules in a Report mode initially to ensure that you know if there are any of the executables in the block rules that are used legitimately in your environment. This allows you to configure different rules for the systems that require these executables but block them on all other computers. Microsoft recommends that you block the following applications. An attacker can use these applications or files to circumvent application allow policies, including WDAC: There are two ways to implement this requirement. 1. Update the Carbon Black provided Suspicious Application Protection

	Rapid Configuration to add all of the executables in the current version of the Microsoft Recommended Block Rules. See here for information regarding using Rapid Configurations. 2. Create an Execution Control rule that specifies all of the executables in the current version of the Microsoft Recommended Block Rules. Refer to this Carbon Black User Community post that has an Execution Control rule that can be imported, and if needed edited or updated for your requirements (Carbon Black Community access required). To use and update the Suspicious Application Protection Rapid Config follow these steps: 1. Navigate to Rules -> Software Rules -> Rapid Configs. 2. Click on the Edit icon for the Suspicious Application Protection. 3. Change the Status to Enabled. 4. Add the following executables to the Report Or Block Execution of Suspicious Applications: addinprocess.exe addinprocess32.exe addinutil.exe aspnet_compiler.exe bash.exe bginfo.exe1 cscript.exe dbghost.exe dbgsvc.exe dotnet.exe fsi.exe fsiAnyCpu.exe infdefaultinstall.exe kd.exe kill.exe lxssmanager.dll lxrun.exe Microsoft.Build.dll Microsoft.Build.Framework.dll

Microsoft.Workflow.Compiler.exe
msbuild.exe
msbuild.dll
mshta.exe
ntkd.exe
ntsd.exe
powershellcustomhost.exe
runscripthelper.exe
texttransform.exe
visualuiaverifynative.exe
system.management.automation.dll
webclnt.dll
davsvc.dll
wfc.exe
windbg.exe
wmic.exe
wscript.exe
wsl.exe
wslconfig.exe
wslhost.exe

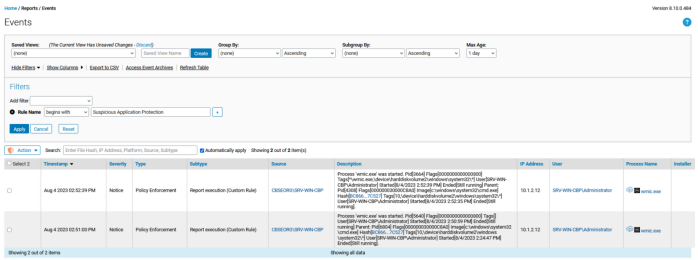
5. Once you added the above executables to the list you can click the Save & Exit button at the bottom of the screen.

NOTE: The above step leaves these block rules in Report Only mode. Please review the events for any hits on this rule to determine if these executables are used in your environment, which may require exceptions for specific systems.

To monitor the events for executions matching this Rapid Configuration do the following:

1. Go to Reports -> Events.
2. If required, change the Saved View to (none).
3. Click on Show Filters.
4. In the Add filter drop down, scroll down and select Rule Name.
5. Change from is to begins with.
6. Enter *Suspicious Application Protection*
7. Change the Max Age to *1 day* (or the amount of time that is

relevant for your environment).



Example of monitoring for events that match the Suspicious Application Protection Rapid Config.

8. If you want to monitor for any hits of this Rapid Configuration on a regular basis add the name of the view you want to save in the Saved View Name text box next to the Saved Views drop down and click the Create button.

i. *Naming & creating a custom view.*

9. This will create a Saved View you can select when in the Events view to monitor only for hits on this Rapid Configuration.

If you prefer to create a custom Execution Control Rule you can either import the rule provided in the Community post mentioned above and make any adjustments and/or updates for your environment, or you can create a new Execution Control rule.

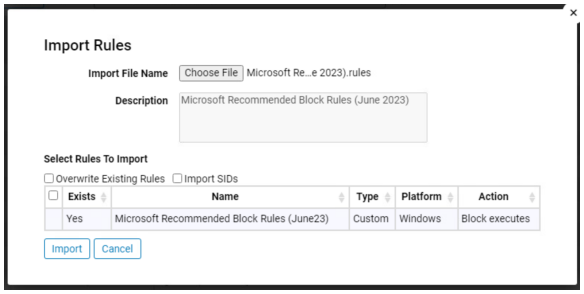
When creating a rule, use the following settings and add the list of Microsoft's recommendations.

To import the community posted rule simply download the zip file from the

community post and extract the .rules file.

1. Then navigate to Rules > Software Rules > Custom.
2. Click on the Import Rules button. 

3. In the Import Rules modal, click the Browse or Choose File button and select the .rules file you extracted previously.



a. *Import Rules Modal.*

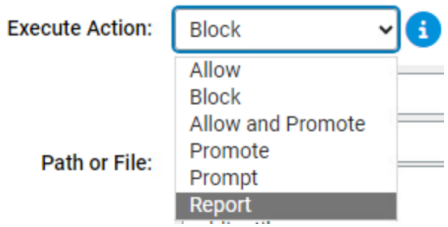
4. Make sure to check the rule you want to import.

5. Review, then click the Import button.

   NOTE: The rule provided in the community post is set to Block, but is disabled by default. The following steps show how to enable and change this rule to Report only for an initial monitoring phase.

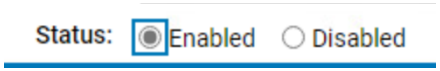
6. Click on the edit rule icon to edit the rule.

7. Since it is recommended to monitor for any of these executables that are legitimately used in your environment before blocking them, change the Execute Action from Block to Report.



Editing the allowed action for an Execution Control rule.

8. Then change the status to Enabled.

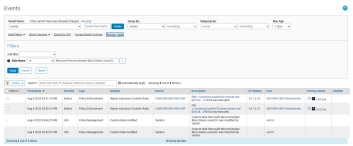


9.

9. Then click the Save & Exit button at the bottom of the page.

Monitoring for any executions of the Microsoft Recommended Block Rules executables or libraries.

1. Go to Reports -> Events.
2. Make sure the Saved View is set to (none).
3. Click Show Filters and in the Add filter drop down, scroll down, and select Rule Name.
4. Enter the name of your newly created rule.
5. Click Apply.



6.

Example configuration of a Saved View for the imported Microsoft

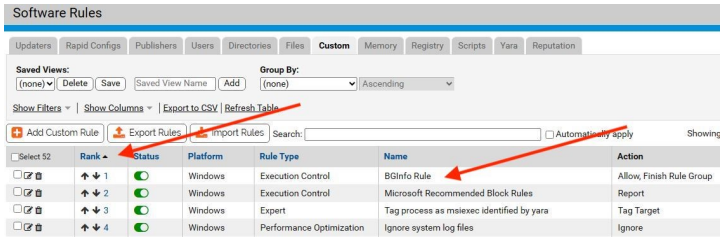
Recommended Block rules.

6. If you want to save this view to quickly review any rule hits simply add the name of the saved view to the Saved View Name text box at the top of the screens and then click the Create button.

Once you are satisfied that only unwanted executions are occurring you should change the Rule's Execute Action back to Block and save the rule.

Rule Prioritisation and Making Allowances for Approved Use Cases when using Execution Control rules.

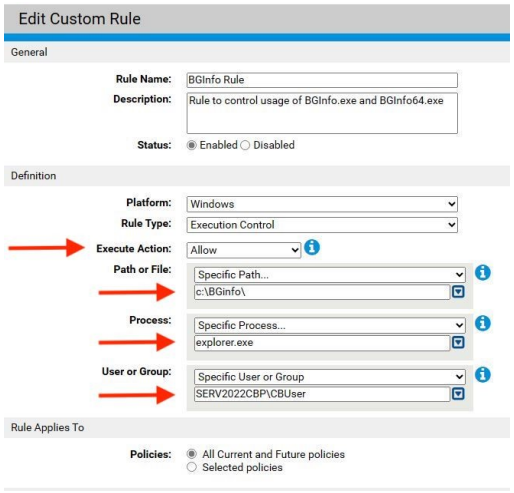
Custom Rules within Carbon Black App Control are applied in order of rank. For example, if we wanted to make a rule allowing BGInfo.exe (one of Microsoft's Recommended Blocked Applications), to run under certain conditions, then we can create a new rule above the rule that blocks it from running. If it has a higher Rank in the rule list, it will be checked first to determine if the application can run.



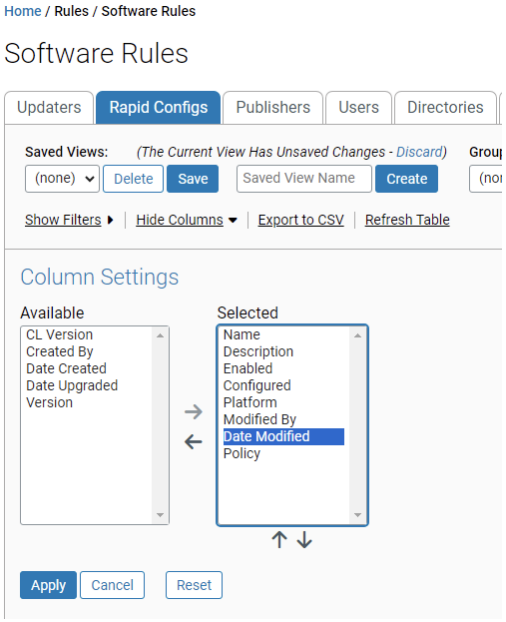
This rule can then be created with a number of conditions under which an application, in this case, BGInfo.exe is allowed to run. This can include the following:

- Directory location
- Parent Process
- Specific User or Group attempting to execute the application

Further to this, the rule can be applied to all policies or specifically selected policies. This allows you can further narrow down which systems the allow rule

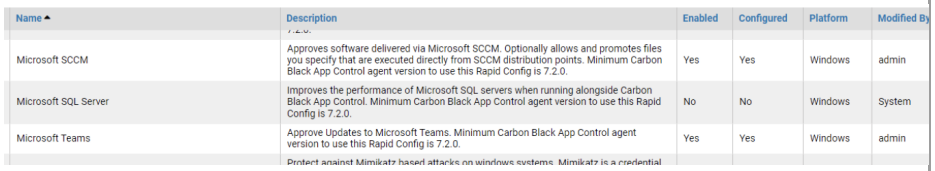
	would apply to on top of the conditions specified. 
Application control rulesets are validated on an annual or more frequent basis.	This requirement is based around creating processes or procedures to make sure that rules created when the solution is first deployed are still working as intended and have coverage to protect from any changes that may have occurred within your organisation. A simple example of this is to check annually for changes to Microsoft's recommended block rules to see if any new applications have been added or removed. Another example is to check that any rule changes have not resulted in a non-compliant configuration, such as allowing execution based only on path. It is likely that you may be making changes to the configuration of App Control as you continue using the solution, with rule changes occasionally being needed to

	accommodate new software or workflows within your organisation. An annual review is recommended to cover anything you may have missed and to validate that your configuration is still done according to best practice and maintains compliance to the Essential Eight mitigation strategy. Carbon Black App Control can assist you with this requirement through reporting provided within the console. The App Control server logs and displays by default both the Username and the Date Modified for each rule to help determine when changes have been made and who made them. This is shown in the rule or policy configuration pages as well as being logged, the latter allowing for monitoring either with the Carbon Black App Control server UI, creating an alert and/or monitoring in a SIEM. You can use this capability to focus your audit on changes that have been made since the previous year's audit, reducing the time it takes to complete the annual audit. Some rule views require that you add more columns to the default view to quickly see when a rules was either last modified or by whom it was last modified. To add columns to the default views follow this process: 1. For Rapid Configs, navigate to Rules -> Software Rules and then click on the Rapid Configs tab. 2. In the Saved Views section just below the tabs, click on the Show Columns hyperlink. 3. This will expand to show the Column Settings. 4. Select the Date Modified column from the Available list then click the right arrow to add that column to the Selected list.



a. Optional: You can re-order columns by selecting the column you want to move and then clicking the up or down arrow below the list.

5. Click Apply.

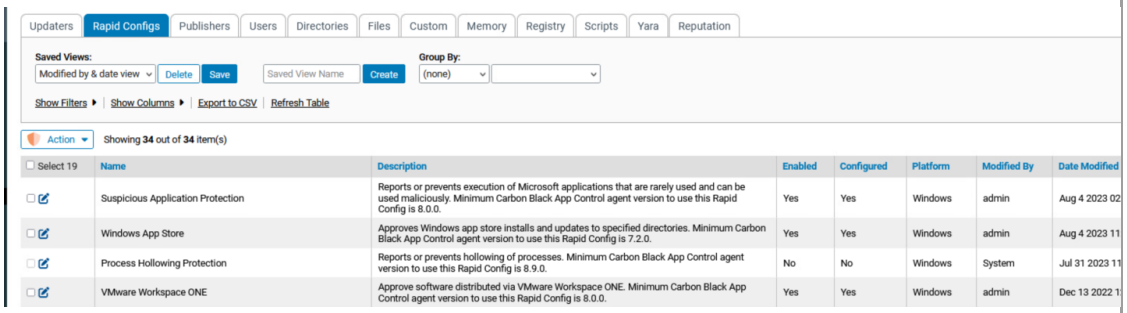


Name	Description	Enabled	Configured	Platform	Modified By
Microsoft SCCM	Approves software delivered via Microsoft SCCM. Optionally allows and promotes files you specify that are executed directly from SCCM distribution points. Minimum Carbon Black App Control agent version to use this Rapid Config is 7.2.0.	Yes	Yes	Windows	admin
Microsoft SQL Server	Improves the performance of Microsoft SQL servers when running alongside Carbon Black App Control. Minimum Carbon Black App Control agent version to use this Rapid Config is 7.2.0.	No	No	Windows	System
Microsoft Teams	Approve Updates to Microsoft Teams. Minimum Carbon Black App Control agent version to use this Rapid Config is 7.2.0.	Yes	Yes	Windows	admin

6. To save this view for future use, add a name for this view in the Saved View name text box.
7. Then click the Create button.

This allows you to select this view in the future from the Saved Views dropdown.

You can also order the list by specific columns by clicking on the column name. If you save your view with the order by the date modified you can see the most recent modified Rapid Config at the top with the oldest modification at the bottom every time you select that saved view.



Example Saved View with the Date Modified column ordered by the most recent modification at the top.

You can repeat the above process for Updaters, Files rules and Custom Rules (or all tabs if desired).

:Example file rules page with the date modified and last modified by columns added.

Windows	File Creation Control	Intune Software Deployment	Jul 24 2023 04:24:42 PM
Windows	Expert	Tag process as msiexec identified by yara	Feb 2 2023 10:01:26 AM

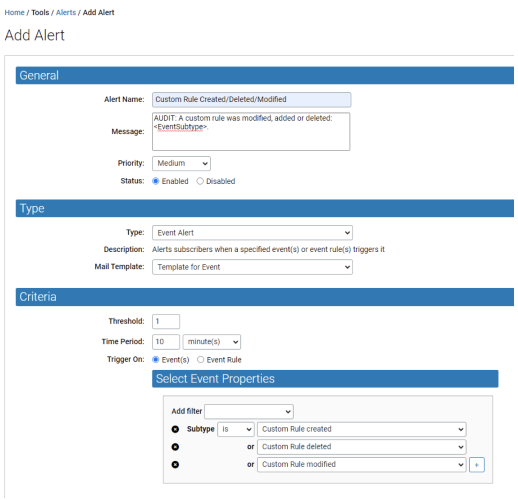
Example of the custom rules tab with the dated modified and modified by columns added.

Enabled: Yes	18 item(s)				
Adobe Reader	Mac	Yes	Feb 11 2022 02:22:41 PM	System	Jul 27 2023 11:17:16 AM
Allow Printer Installations	Windows	Yes	Feb 11 2022 02:22:38 PM	System	Feb 11 2022 02:22:38 PM
Apple System Performance	Mac	Yes	Feb 11 2022 02:22:53 PM	System	Feb 11 2022 02:22:53 PM
Carbon Black Cloud	Windows	Yes	Feb 11 2022 02:22:53 PM	System	Jul 27 2023 11:16:12 AM

Example of the Updaters tab with modified columns added.

To create an alert for any custom rules being modified, created or deleted:

1. Navigate to Tools -> Alerts.
2. Click the Add Alert button.
3. Name the rule and customise the message as needed.
4. In the Criteria section, change the Subtype to *Custom Rule modified*.
    a. Click the + button to add additional criteria.
    b. Select *Custom Rule created*.
    c. Click the + button to add additional criteria.
    d. Select *Custom Rule deleted*.

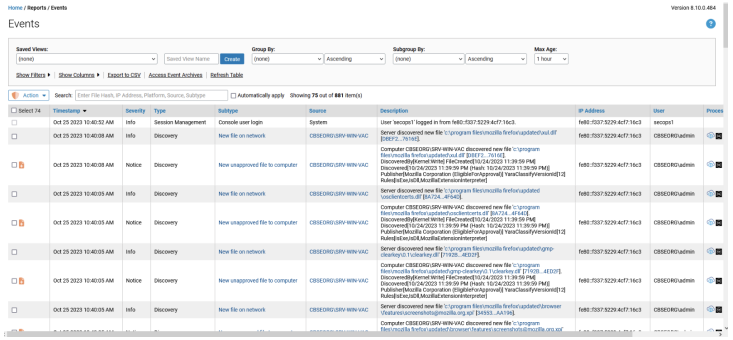
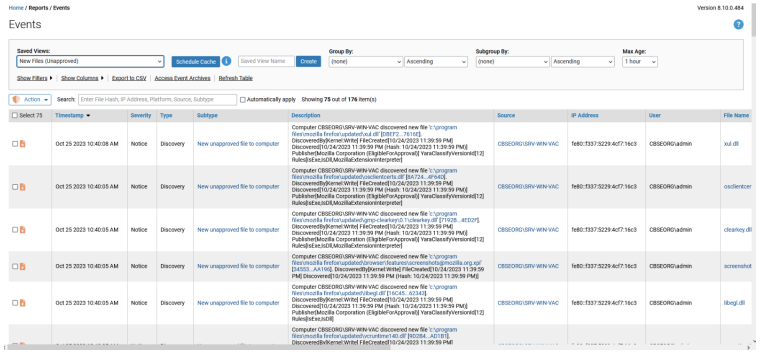


Add Alert configuration example for alerting on custom rule changes.

5. You can also change the threshold criteria if desired to reduce the volume of alerts when multiple changes are made in rapid succession.
6. Scroll down and click the Create & Exit button.

Carbon Black App Control will now generate an alert when a custom rule is created, deleted, or modified.

Editing that Alert will allow you to add subscribers who can receive an email notification when that alert is triggered.

Allowed and blocked application control events are centrally logged.	All blocked and first time allowed executions of a binary are logged for all agents, as well as all file writes and file write blocks (Plus many more event types). To view all these events, go to Reports -> Events. The Events page has some built-in saved views to allow you to quickly filter the events to show only the events that you want to see. For example, you can select the New Files (Unapproved) view to see all new files in the environment that have not been approved. This view is particularly useful during the initial phases of deployment to identify the approval rules you need to create for your environment.  *The default Events view.* 

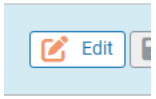
The New Files (Unapproved) view in Events.

Carbon Black recommends forwarding events to your SIEM or Log Management solution. This can be done in two ways:

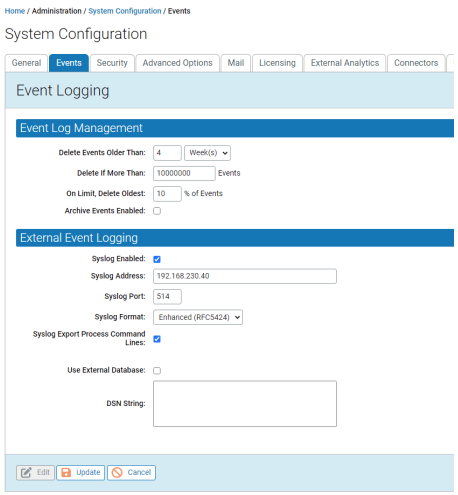
1. Syslog.
2. File forwarding with your SIEM or Log Manager's log forwarder.

Configuring syslog can be done using the following steps:

1. Go to Settings -> System Configuration -> Events
    a. if navigating with the left-hand menu you would go to Administration > System Configuration -> Events
2. Click the Edit button at the bottom of the page.



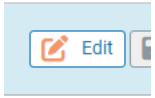
3. Click the Syslog Enabled checkbox.
4. Enter the address of your syslog receiver/server.
5. Change the syslog port if required.
6. Select the syslog format you require.
    a. This can be:
        i. Basic (RFC3164)
        ii. Enhanced (RFC5424)
        iii. CEF (ArcSight/HP)
        iv. LEEF (Q1 Labs/IBM)
7. If you want to export process command lines make sure to check that box.
8. Click the Update button at the bottom of the screen.



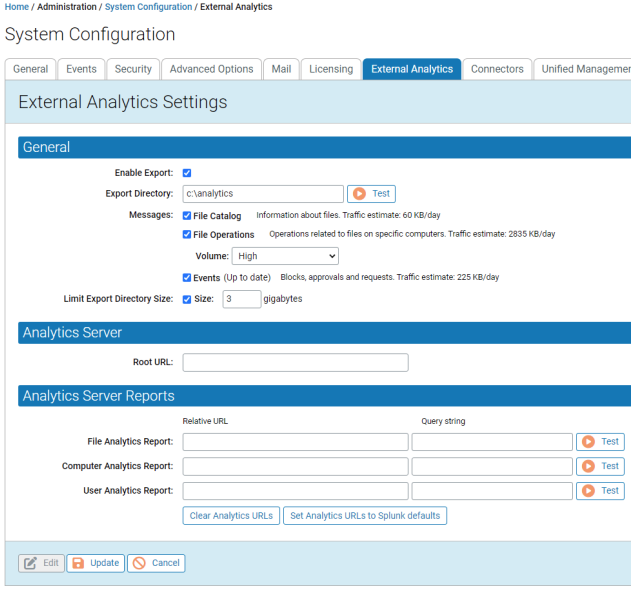
Example Syslog configuration.

Configuring the second option is called External Analytics Settings and can be done using the following steps:

1. Go to Settings -> System Configuration -> External Analytics (if navigating with the left-hand menu you would go to Administration -> System Configuration -> External Analytics)
2. Click the Edit button at the bottom of the page.



3. Click the Enable Export button.
4. Enter the directory to export the logs to.
    a. This should be a directory on the App Control server that is monitored by your SIEM/Log Management solutions file exporter tool (e.g. Splunk Universal Forwarder or Sumo Logic Collector).
    b. The directory should also be writable by the App Control Server service (ParityServer).
5. Enable the data you want to export. More information can be found in

	the product user guide. 6. If you want to limit the export directory size, click the check box and enter the maximum directory size in gigabytes. (min 3GB, max 10 petabytes) 7. Click the Update button at the bottom of the screen.  Example configuration of event export for External Analytics (e.g. SIEM).
Event logs are protected from unauthorised modification and deletion.	Carbon Black recommends that event logs are forwarded to your log management and/or SIEM solution and protected in those systems along with the rest of your logs. Carbon Black App Control uses a SQL database to store its records. Carbon Black recommends that you follow security recommendations from the vendor of your SQL solution to help protect the database and all event logs contained within. If you are using Microsoft SQL Server, Microsoft has guides for Securing SQL Servers and SQL Server Security Best Practices. Further to Microsoft's recommendations, Carbon Black provides recommendations

	for configuration of SQL Servers for use with Carbon Black App Control, including permissions needed for a dedicated service account to ensure permissions granted are kept to a minimum. See the CB App Control SQL Server Configuration Guide for more information. Microsoft's recommendations around securing SQL servers include aspects of system hardening and application control. It is recommended that you have a policy dedicated to your SQL Servers that runs in high enforcement and completely locks down the use of any application not required for the server to perform its tasks.
Event logs from internet-facing servers are analysed in a timely manner to detect cyber security events.	As discussed in the previous requirement, Event Logs are stored on a SQL Database. There are several ways to monitor logs both via App Control and via external methods. Having the Carbon Black App Control Agent deployed on the servers hosting both Carbon Black App Control and the SQL Server in High Enforcement is important. It is recommended that you export logs to an external log collection platform such as a SIEM. App Control supports exporting logs in the Syslog format and multiple SIEM vendors have applications to visualise and analyse App Control logs. Please refer to the Allowed and blocked application control events are centrally logged section for steps for configuring both Syslog export and event export for external analytics. External notifications can also be setup within Carbon Black App Control to notify the security team of any Blocked Files attempting to run on High Enforcement endpoints and review Carbon Black App Control API Reference

Cyber security events are analysed in a timely manner to identify cyber security incidents. Cyber security incidents are reported to the Chief Information Security Officer, or one of their delegates, as soon as possible after they occur or are discovered. Cyber security incidents are reported to ASD as soon as possible after they occur or are discovered. Following the identification of a cyber security incident, the cyber security incident response plan is enacted.	New requirement as of November 2023 These four requirements are focused on organisation policies and procedures that should be followed in the event of a cyber security events or incidents. The first requirement involves an organisation's process for assessing security events in a timely manner to be able to identify if a cyber security incident has occurred. Identification of events promptly can help detect threats in the early stages of an attack and can help limit the scope of an incident and the damage it can cause. It can also assist in obtaining assistance from third parties to help contain the threat if resources do not exist internally with the necessary skills. The second and third requirements are to make sure the CISO (Chief Information Security Officer), or one of their delegates; and the ASD's ACSC (Australian Signals Directorate's Australia Cyber Security Centre) are notified as soon as possible in the event of a cyber security incident being discovered. This often means alerting the relevant stakeholders before a full picture of an incident is available. Depending on the industry your organisation resides in, there may also be legal requirements to disclose such incidents within a certain timeframe. Your organisation should be aware of any regulatory requirements it is subject to. Additionally, your organisations should have clear and available information on how soon each party should be notified, who is responsible for reporting an incident, and in what time frame an incident should be reported. The fourth requirement is to make sure your organisation's incident response plan is enacted in the event of a cyber security incident. Clear and available documentation covering the organisation's process of what to do in the event of an incident should exist. This should cover what steps a member of the security team should follow once an incident is identified including the aforementioned reporting.

Maturity Level 3

Control	**Carbon Black App Control**
Application control is implemented on non-internet-facing servers.	To meet this requirement, you should install the Carbon Black App Control agent on all remaining servers. Assuming you have already deployed the agent to all workstations and internet-facing servers, and implemented a High Enforcement policy on those systems, the process for the remaining servers will be similar to the steps you followed for meeting the Maturity Level 2 requirements.
Application control restricts the execution of drivers to an organisation-approved set.	Carbon Black App Control includes drivers by default and are treated like any executable code and they need to be approved in High Enforcement. Following the above process for the rest of your servers will ensure you include drivers for those additional servers in your organisation-approved set of drivers, along with all the other types of files listed in the requirement.
Microsoft's vulnerable driver blocklist is implemented.	To ensure that drivers on the Recommended Driver Block rules are detected, and prevented from being loaded when they are first introduced to your environment, Carbon Black recommends blocking the hashes for these drivers. Any time a banned driver is loaded a specific event is generated (Execution allowed (file loaded before kernel) or Execution Allowed (file loaded before service)). Importing hashes can be done in two ways: 1. Import a list of hashes via the App Control console user interface. i. Pros: Easy to do via the UI.

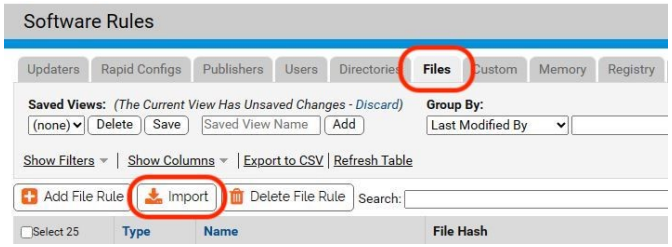
ii. Cons: Results in all hashes in a list having the same rule name.

2. Import hashes along with other properties via the App Control API.
    i. Pros: Allows for individual rule names for hashes and can also include filenames in addition to a rule name.
    ii. Cons: Requires that you format the data and script the import via the App Control API.
        1. There is a community project that has already created a supported .csv file and PowerShell script to import via the API.

Option 1 – Importing a list of driver hashes

To implement the first option, you can follow the below steps.

- First step is to import the hashes from Microsoft's list of Recommended Blocked Drivers. The hash values for these drivers need to be added to a text file.
- When you have the text file list ready, you need to go to Software Rules -> Files.
- On this screen, click on the Import button (see below screenshot).
- You will then get a pop-up window where you can choose a Rule Name, Select the text file to upload and choose the Rule Type. The option to completely Ban the drivers from loading can be chosen here. This will also stop the drivers from being introduced / copied to one of the endpoints. Alternatively, if you wish to test the rules first, you can choose the option to Ban (Report Only).



	 Option 2 – Importing via the API To implement the second option, you can utilise one of the API's available for the platform that allows you to import file information for approvals or bans. You can find more information on that API here. For information on utilising API's with Carbon Black App Control, you can find further details in the User Guide under Carbon Black App Control API. To assist with this process, you can utilise a Powershell script created by a member of the Carbon Black team that helps automate the process. You can find out more information about the process and links to download the script on this Carbon Black Community post (Carbon Black Community access required).
Event logs from non-internet-facing servers are analysed in a timely manner to detect cyber security events.	This requirement builds on the previous requirement from Maturity Level 2. This time, extending the requirement to analyse event logs to non-internet-facing servers. This may require changing your external notifications from Carbon Black App Control to increase the scope and cover these additional endpoints. It may also require adjustment to events that are being sent to your SIEM or log collection platform to ensure coverage of these additional devices. See also Events, Alerts and Metres.

Event logs from workstations are analysed in a timely manner to detect cyber security events.	As per the previous requirement, this requirement builds on the previous requirement from Maturity Level 2. This time, extending the requirement to analyse event logs to workstations. This may require changing your external notifications from Carbon Black App Control to increase the scope and cover these additional endpoints. It may also require adjustment to events that are being sent to your SIEM or log collection platform to ensure coverage of these additional devices. Other Carbon Black App Control capabilities to consider for reporting include Baseline Drift Reports:

Application Approval automation

Carbon Black App Control provides multiple methods for automatic approvals, reducing the administration overheads of Application Control or allow listing.

The following steps are recommended for most organisations, and you can use the examples below to create approvals specific to your organisation.

Note: There are a number of additional approval methods not listed below. To review all options for approvals please refer to the Carbon Black App Control.

Approvals allow your users and applications to work without interruption, and when configured correctly allows for manual and automatic updates to approved applications.

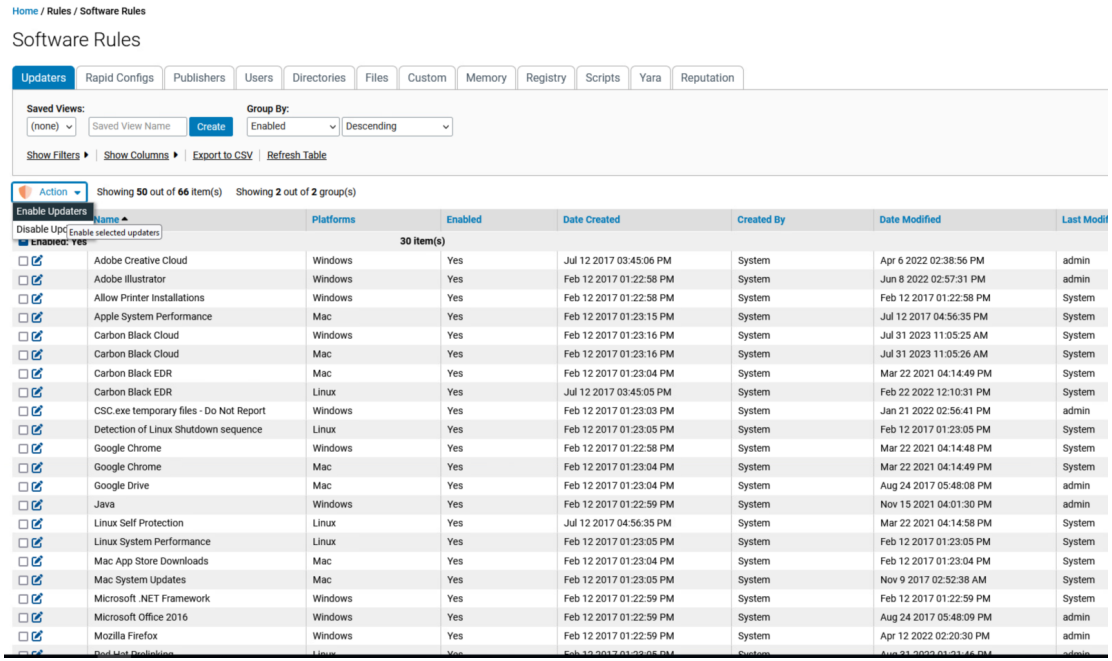
Note: Best practice is to use approvals and file creation control rules rather than execution control rules where possible. Execution control rules should only be used when other options are not viable.

Updaters – allow approved software to self-update

Trusted Updaters are used to determine which programs are allowed to self-update on a system, or be manually updated by an end-user, without interruption or manual approval from an administrator. Updaters are created and maintained by Carbon Black.

To enable relevant Updates for your environment:

1. Go to Rules -> Software Rules -> Updaters
    1. Select all the applications that apply for your organisation.



Trusted Updaters example

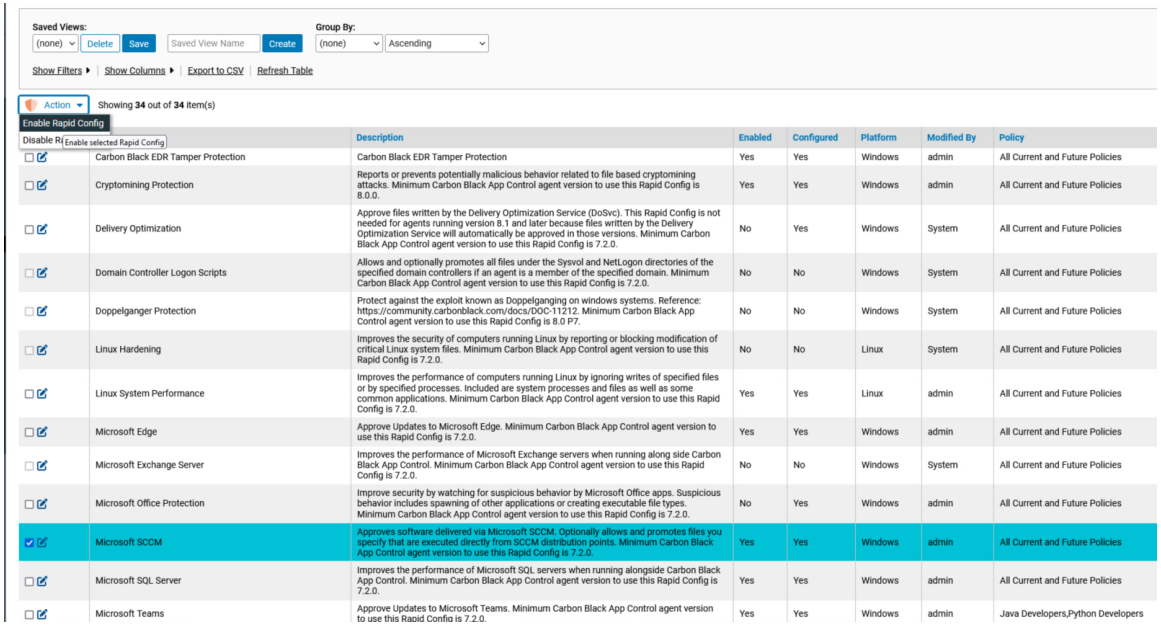
2. Click the Action button and select Enable Updaters

Software Deployment Tool Approvals

Most organisations deploy software and patches using an enterprise software deployment tool. By configuring Carbon Black App Control to automatically approve software and patches deployed via your software deployment tool you can streamline the allow listing process and ensure that, as long as the application and/or patch is approved to be installed by your software deployment tool, the application and/or patch will execute as expected.

Carbon Black provides several Rapid Configurations out of the box, which includes some Rapid Configs for common software deployment tools such as Microsoft SCCM and VMware Workspace ONE. In this section of this guide, we will refer to Microsoft SCCM as an example.

1. Go to Rules -> Software Rules -> Rapid Configs and select Microsoft SCCM from the list, then click the *Action* button and select *Enable Rapid Config*.

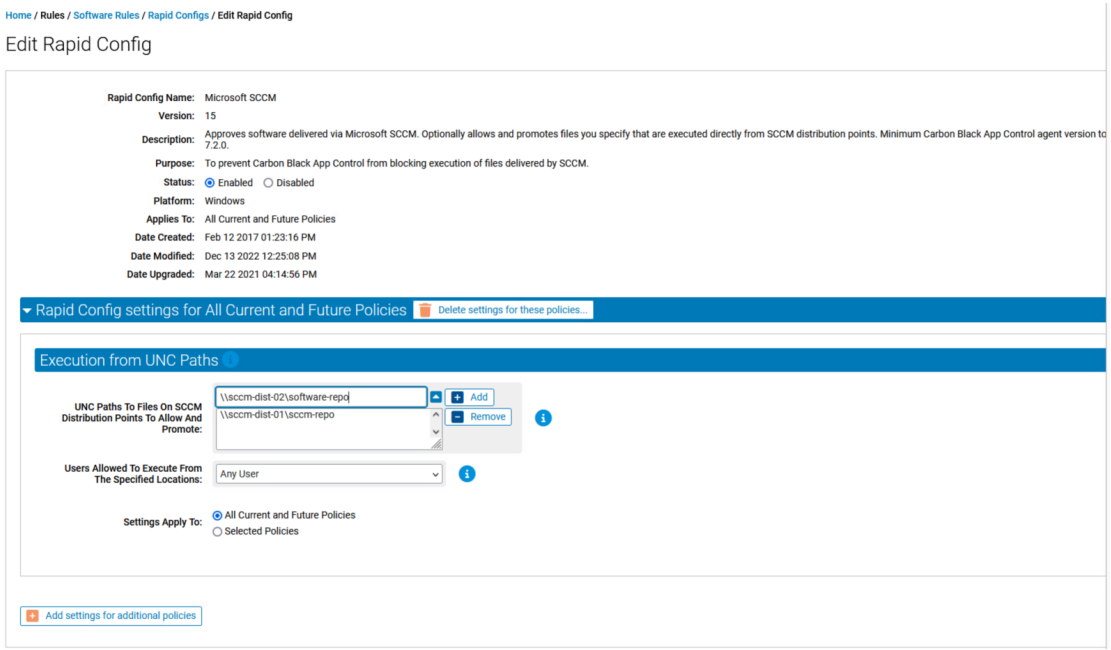


		Description	Enabled	Configured	Platform	Modified By	Policy
☐	Carbon Black EDR Tamper Protection	Carbon Black EDR Tamper Protection	Yes	Yes	Windows	admin	All Current and Future Policies
☐	Cryptomining Protection	Reports or prevents potentially malicious behavior related to file based cryptomining attacks. Minimum Carbon Black App Control agent version to use this Rapid Config is 8.0.0.	Yes	Yes	Windows	admin	All Current and Future Policies
☐	Delivery Optimization	Approve files written by the Delivery Optimization Service (DoSvc). This Rapid Config is not needed for agents running version 8.1 and later because files written by the Delivery Optimization Service will automatically be approved in those versions. Minimum Carbon Black App Control agent version to use this Rapid Config is 7.2.0.	No	Yes	Windows	System	All Current and Future Policies
☐	Domain Controller Logon Scripts	Allows and optionally promotes all files under the Sysvol and NetLogon directories of the specified domain controllers if an agent is a member of the specified domain. Minimum Carbon Black App Control agent version to use this Rapid Config is 7.2.0.	No	No	Windows	System	All Current and Future Policies
☐	Doppelganger Protection	Protect against the exploit known as Doppelganging on windows systems. Reference: https://community.carbonblack.com/docs/DOC-11212. Minimum Carbon Black App Control agent version to use this Rapid Config is 8.0 P7.	No	No	Windows	System	All Current and Future Policies
☐	Linux Hardening	Improves the security of computers running Linux by reporting or blocking modification of critical Linux system files. Minimum Carbon Black App Control agent version to use this Rapid Config is 7.2.0.	No	No	Linux	System	All Current and Future Policies
☐	Linux System Performance	Improves the performance of computers running Linux by ignoring writes of specified files or by specified processes. Included are system processes and files as well as some common applications. Minimum Carbon Black App Control agent version to use this Rapid Config is 7.2.0.	Yes	Yes	Linux	admin	All Current and Future Policies
☐	Microsoft Edge	Approve Updates to Microsoft Edge. Minimum Carbon Black App Control agent version to use this Rapid Config is 7.2.0.	Yes	Yes	Windows	admin	All Current and Future Policies
☐	Microsoft Exchange Server	Improves the performance of Microsoft Exchange servers when running along side Carbon Black App Control. Minimum Carbon Black App Control agent version to use this Rapid Config is 7.2.0.	No	No	Windows	System	All Current and Future Policies
☐	Microsoft Office Protection	Improve security by watching for suspicious behavior by Microsoft Office apps. Suspicious behavior includes spawning of other applications or creating executable file types. Minimum Carbon Black App Control agent version to use this Rapid Config is 7.2.0.	No	Yes	Windows	admin	All Current and Future Policies
☑	Microsoft SCCM	Approves software delivered via Microsoft SCCM. Optionally allows and promotes files you specify that are executed directly from SCCM distribution points. Minimum Carbon Black App Control agent version to use this Rapid Config is 7.2.0.	Yes	Yes	Windows	admin	All Current and Future Policies
☐	Microsoft SQL Server	Improves the performance of Microsoft SQL servers when running alongside Carbon Black App Control. Minimum Carbon Black App Control agent version to use this Rapid Config is 7.2.0.	Yes	Yes	Windows	admin	All Current and Future Policies
☐	Microsoft Teams	Approve Updates to Microsoft Teams. Minimum Carbon Black App Control agent version to use this Rapid Config is 7.2.0.	Yes	Yes	Windows	admin	Java Developers,Python Developers

Microsoft SCCM Rapid Config

a. In most organisations, that is all that is required to approve software deployed via SCCM. However, in some cases you may also need to specify your SCCM distribution points.

    i. To do this click on the edit icon for the Microsoft SCCM Rapid Config, then add the UNC path for your SCCM distribution points.
    Press Enter after each UNC path to add another UNC path to the list.



Adding SCCM Distribution Points to SCCM Rapid Config

ii. *Optional:* Modify the users allowed to execute from the defined SCCM distribution points.

2. As with any Rapid Configuration you can apply the one configuration to all policies or to specific policies, the latter allowing for more control.

Creating Custom Rules for Software Deployment Tools

If your software deployment tool does not have a Rapid Configuration you can replicate the SCCM and Workspace ONE Rapid Configurations using Custom Rules.

The following is an example of how to do this for Microsoft Intune.

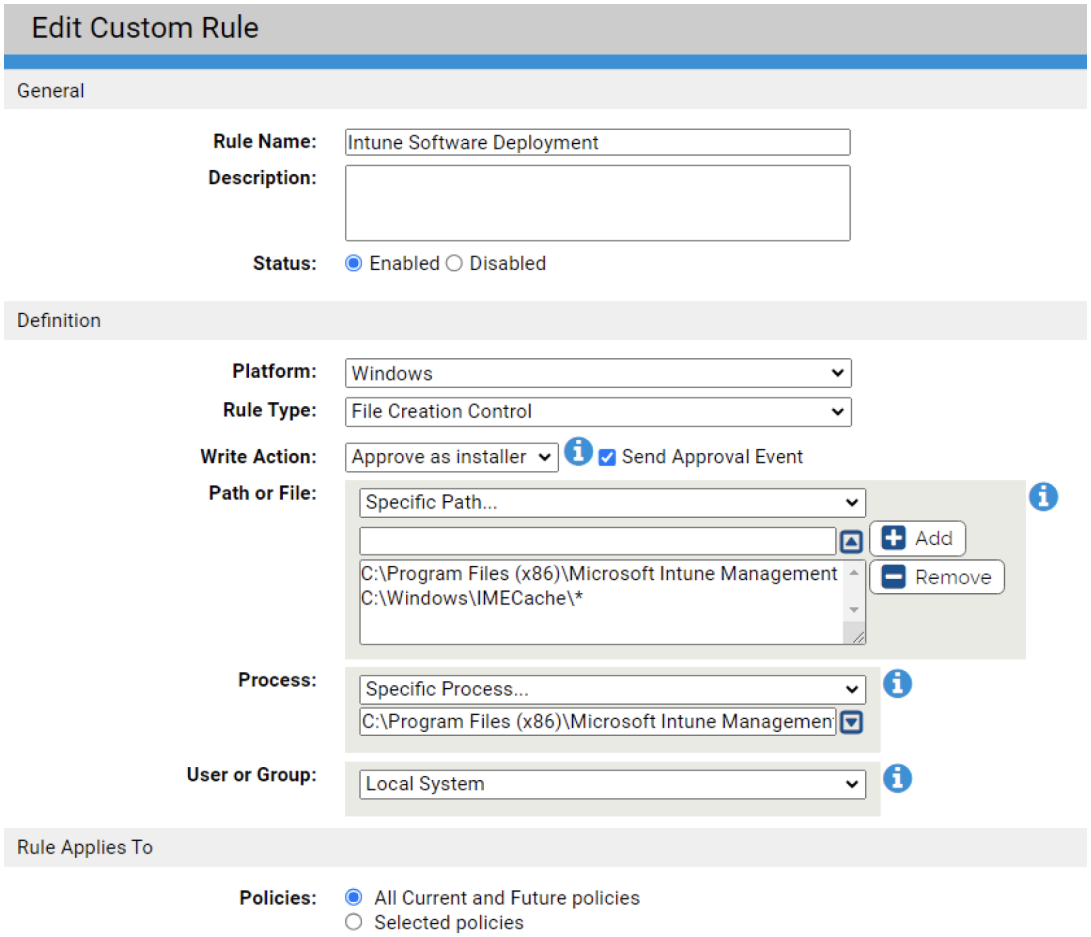
1. Go to Rules -> Software Rules -> Click on the Add Custom Rule button.

a. Name the Rule appropriately.

b. A clear description of the intent of this rule is highly recommended.

c. Select the Rule Type as: File Creation Control

d. Select the Write Action as: Approve as Installer

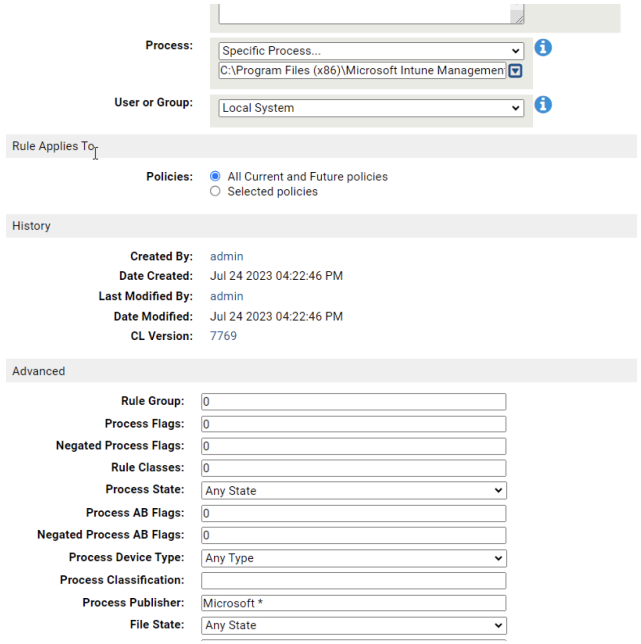


InTune Approvals Example (validate correct paths for your environment)

e. Click the Save & Exit button at the bottom of the page to save the rule.

f. If you want to specify Advanced settings such as the publisher of the process used to write installers to disk, you need to have a saved rule. Edit the rule again and scroll down to see the Advanced settings.

g. For this example (Microsoft Intune), Add Microsoft * to the Process Publisher. You can be more specific and enter the exact publisher name for the relevant processes. Process publishers need to be valid publishers with valid certificate chains in order for this value to apply.



InTune Approvals Example – Advanced settings – Process Publisher setting

PowerShell Constrained Language Mode

By default, Windows will attempt to execute a randomly generated PowerShell script from temporary folders every time PowerShell is executed. This feature is used to check if Windows Defender Application Control or AppLocker is enabled. If these are not enabled, those PowerShell scripts will execute but if those tools are enabled, those scripts will be blocked and will default that PowerShell session to use Constrained Language Mode.

If you enable App Control High Enforcement, these scripts will be blocked, and by default, the user will see a block notification.

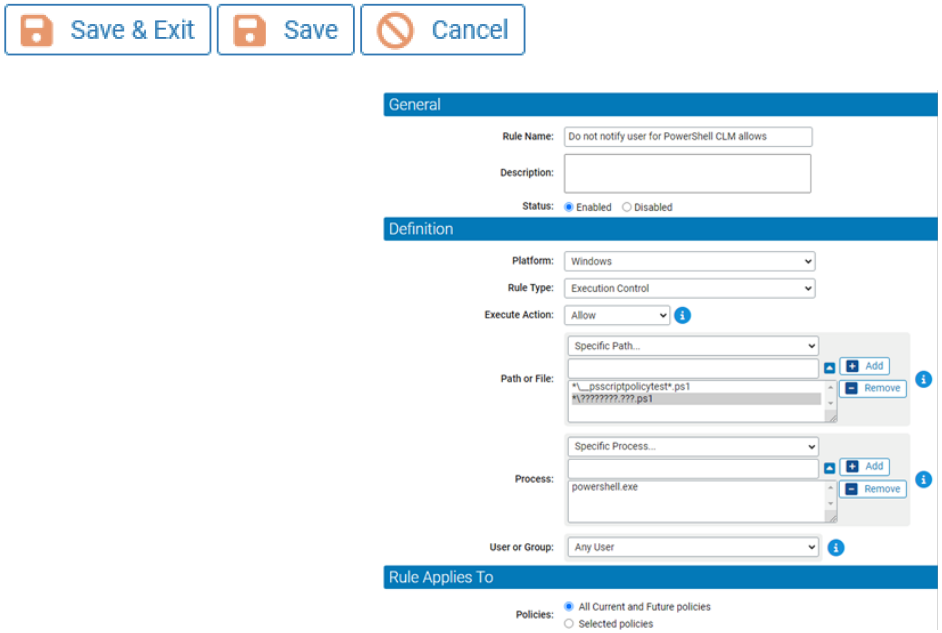
Since PowerShell Constrained Language Mode is not required for Maturity Level 2 you can configure App Control to allow the execution of these temporary scripts by creating a custom rule. These are still unsigned and run from a temporary folder. You can use the below monitoring procedure to identify these scripts and create your own custom rule. While it is also possible to completely bypass the monitoring of these temporary scripts it is not recommended to configure App Control in this way as that may result in an Essential Eight non-compliant configuration.

Allow execution.

1. Create a new Execution Control rule.
    a. Make sure the platform is set to Windows.
    b. Set the Execution Action to Allow.
    c. In the Path add the following two paths:
        i. *__psscriptpolicytest*.ps1
        ii. *\????????.???.ps1
    d. Enter the Process as: powershell.exe
    e. Set the User or Group to Any User.
        i. Note that when a user executes a PowerShell script these files are created in the context of that user but when the OS creates these scripts they are created under the context of LOCAL SYSTEM. Some applications may create files under the context of the service account.
    f. Select the policies to apply this rule to (should be all current and future policies but you may have a reason to apply

to specific policies.

g. Click the Save & Exit button at the bottom of the page to save the rule.



Example Allow rule for PowerShell Constrained Language Mode test scripts.

Block execution silently.

If you choose to just block these scripts, you do not need a custom rule as they would be blocked by default. However, to ensure users do not get these block notifications you will want to create a custom rule with no notifier so you can block these scripts silently.

1. Create a new Execution Control rule.
    a. Make sure the platform is set to Windows.

b. Set the Execution Action to Block.
c. Uncheck the Use Policy Specific Notifier checkbox.
d. For the Custom Execute Notifier dropdown, select <none>.
e. In the Path add the following two paths:
    i. *__psscriptpolicytest*.ps1
    ii. *\????????.???.ps1
f. Enter the Process as: powershell.exe
g. Set the User or Group to Any User.
    i. Note that when a user executes a PowerShell script these files are created in the context of that user but when the OS creates these scripts they are created under the context of LOCAL SYSTEM. Some applications may create files under the context of the service account.
h. Select the policies to apply this rule to (should be all current and future policies but you may have a reason to apply to specific policies.
i. Click the Save & Exit button at the bottom of the page to save the rule.

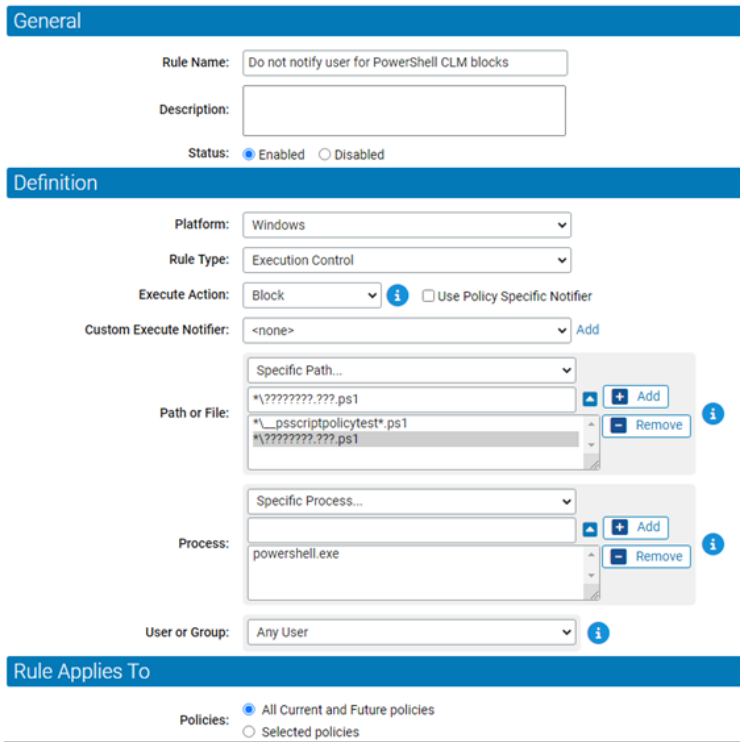
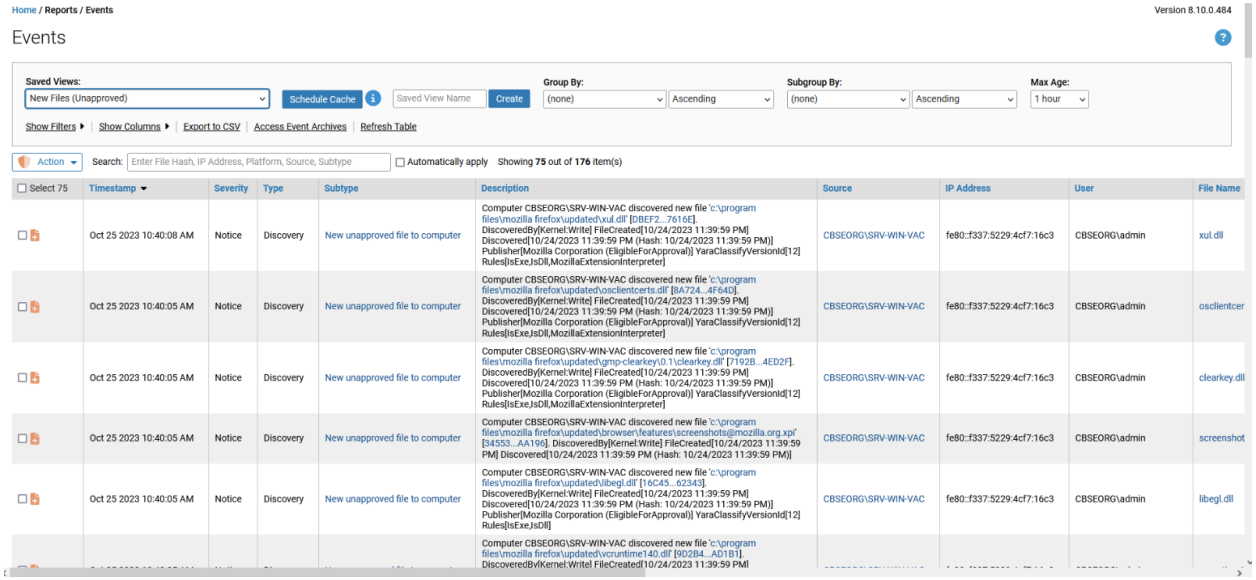


Figure 11: Example Silent Block rule for PowerShell Constrained Language Mode test scripts.

Monitoring for unapproved files

Your systems should now be in a policy with a Low Enforcement mode. This allows you to monitor your environment for some time to ensure that the Approvals you've configured in the previous steps are covering all the required software instals, patches, and operating system updates.

The goal of this monitoring period is to identify if you require any additional approval automation. You can use a default saved view, called New Files (Unapproved), of the Events to determine if there are any new, unapproved files in your environment.



The New Files (Unapproved) view in Events.

Adding more columns to the default view can provide additional information you can use to create a new approval automation, such as the process that wrote the file to disk, user account that process was executed with, the path of the new file, the file prevalence, and the publisher information.

Moving to High Enforcement

Before moving systems to High Enforcement, or default deny, you should be comfortable that your existing configuration is automatically approving new and updated executables, software libraries, scripts, installers, compiled HTML, HTML applications and control panel applets as expected. This should include automatic approvals of application and operating system patching. At this stage you can start to move systems from Low Enforcement (monitoring) to High Enforcement (Application Control, default deny).

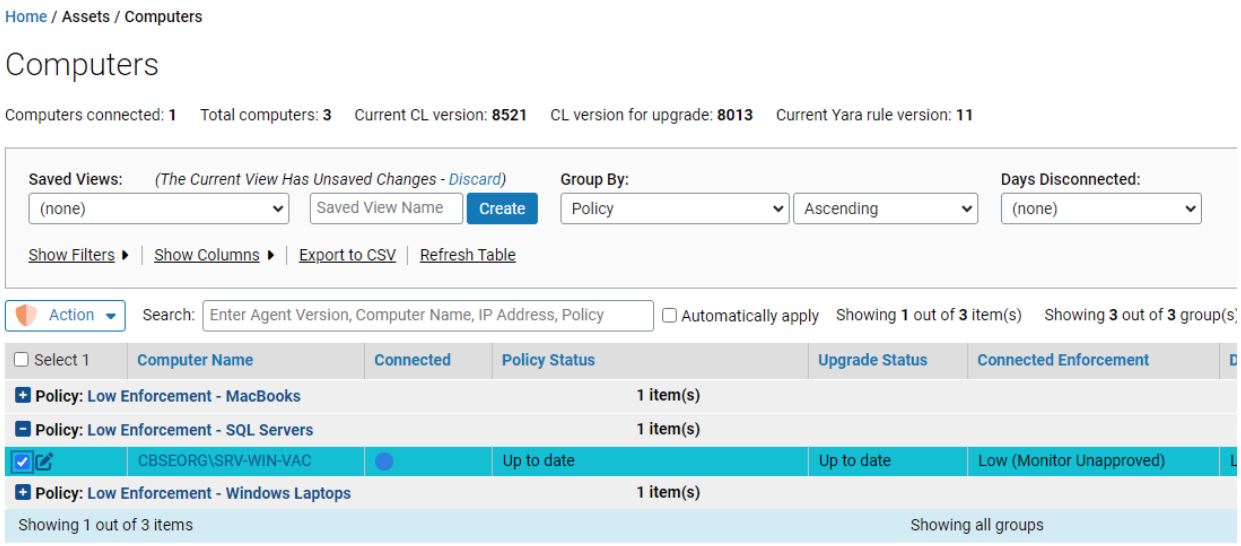
Carbon Black recommends that you move systems from one policy to another, starting with a small pilot group and then monitoring

for any unwanted denials of execution. Adjust the automatic approvals as needed and then move a slightly larger group of systems into High Enforcement.

Repeat this process in successive deployment stages until you have moved all systems into high enforcement.

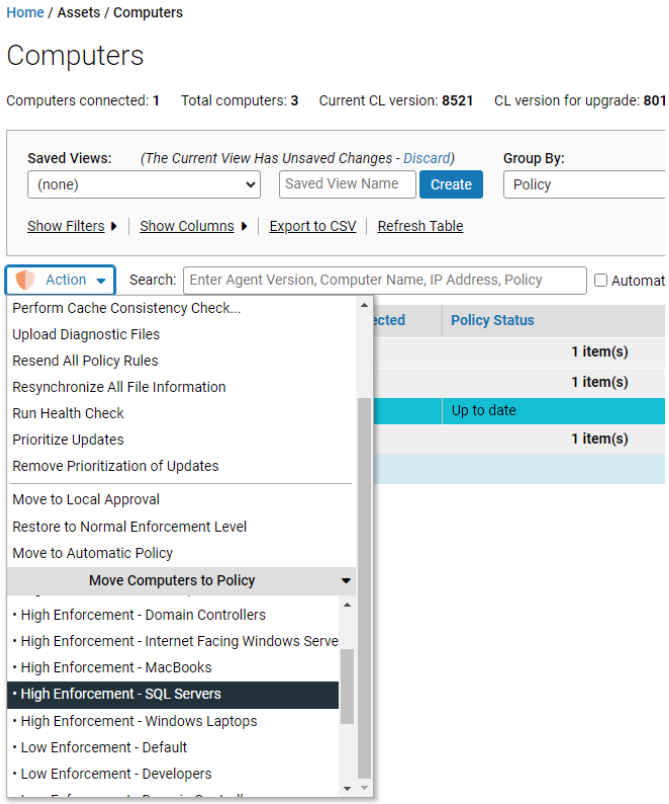
To change a policy on a system, follow these steps:

- Go to Assets -> Computers
- Use the Group By feature to group systems as you feel is best for you.
    - The following example is grouping by Policy, but you can group by Platform, Operating System, Machine Model, Virtualised, Computer Name or many other options.
- Expand the group where you want to select a subset of computers to move policy.
- Select several computers using the checkbox.



Example of selecting specific systems based on Group by Policy with the intent to change their policy.

- Click the Action button.
- Scroll down and select the policy you want to move those computers to.



Example of changing computers' policy to a High Enforcement policy.

Performance Optimisations

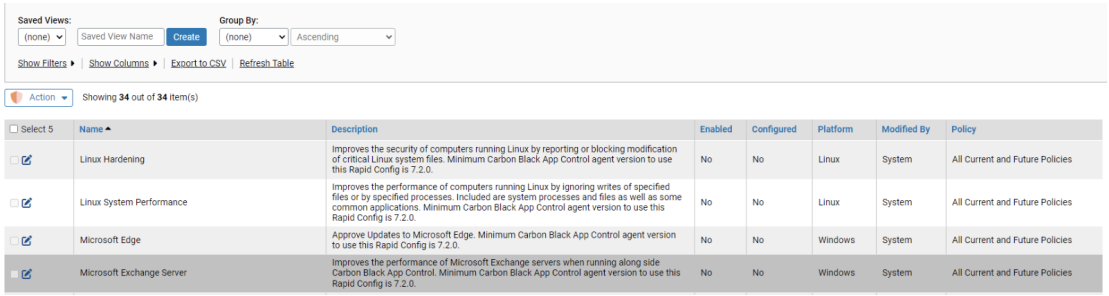
In every organisation there will be some applications that continuously write files to disk that are important for that application but are not important for Application Control. Some examples include log files and database files. Carbon Black App Control provides a method to ignore these files to reduce the performance overhead of hashing every file called Performance Optimisation Rules.

Performance Optimisation rules ignore file Reads, Writes, Creates and Renames. Executions are *not* ignored by Performance Optimisation rules.

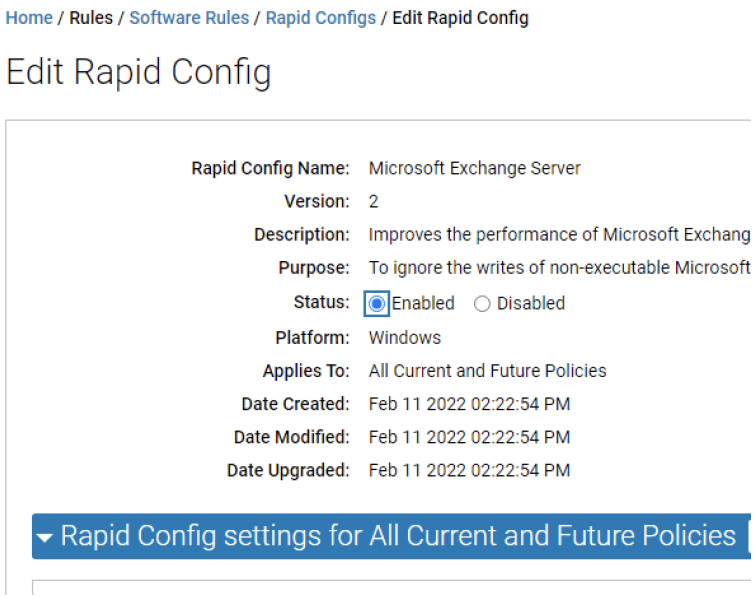
Carbon Black also maintains some Rapid Configurations that are focused on performance optimisation, and recommends leveraging a combination of these Rapid Configurations, where relevant, and custom Performance Optimisation rules to minimise the impact to systems without impacting on the ability to control applications in your environment.

Performance Optimising Rapid Configurations: Exchange Server

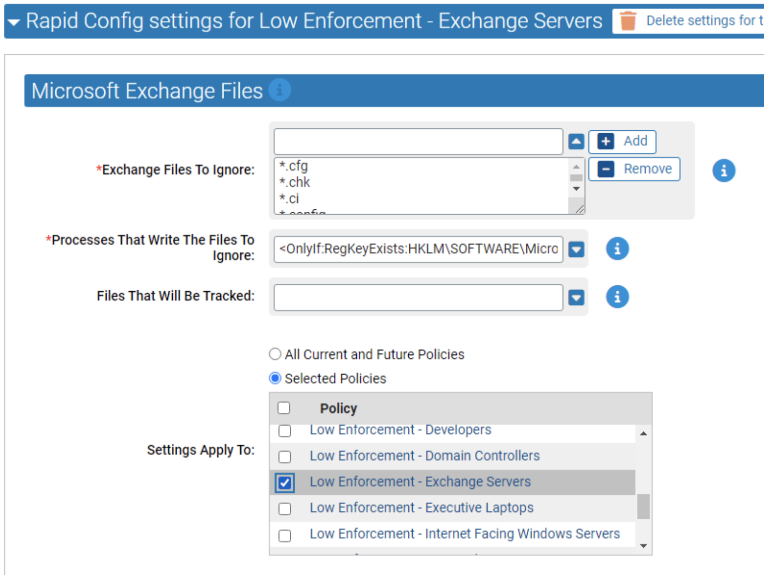
1. Navigate to Rules > Software Rules and click on the Rapid Configs tab.
2. Scroll down to Microsoft Exchange Server and click the View Details/Edit button.



3. Change the Status from Disabled to Enabled.



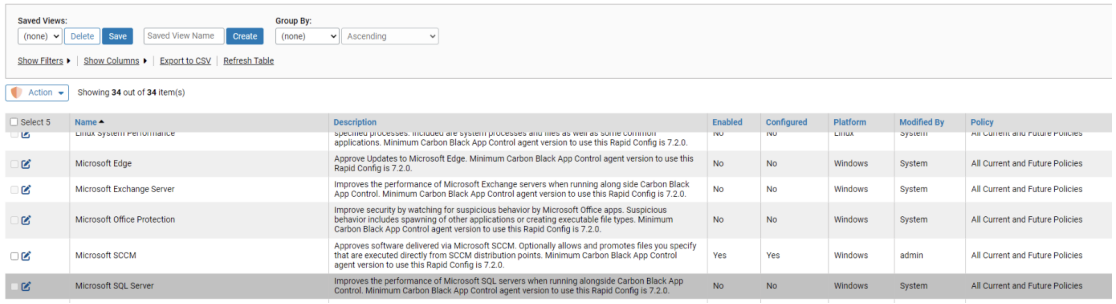
4. Change the Settings Apply To field from All Current and Future Policies to Selected Policies.
5. Scroll down to find the relevant policy (for Exchange Servers) and select that policy.



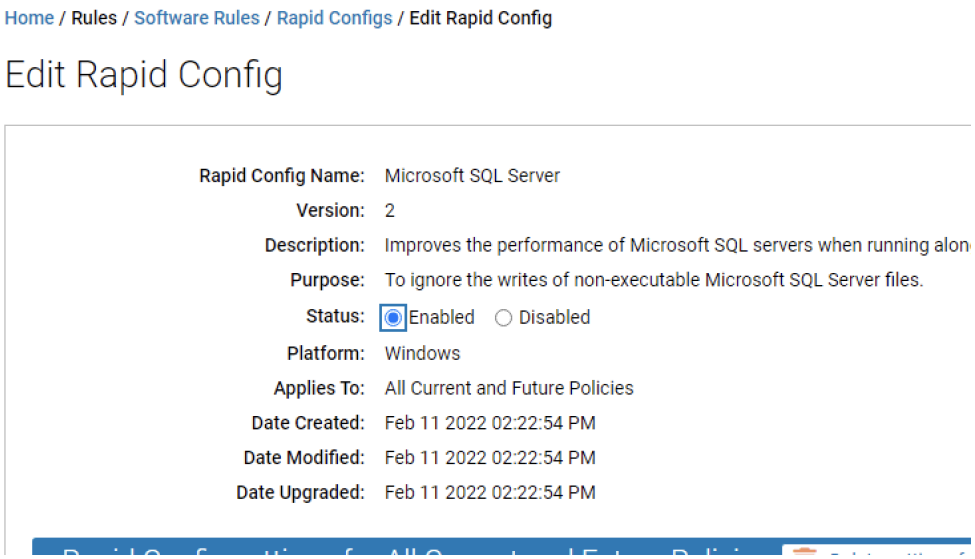
6. Click the Save & Exit button at the bottom of the screen to save the changes to the Rapid Config.

Microsoft SQL Server

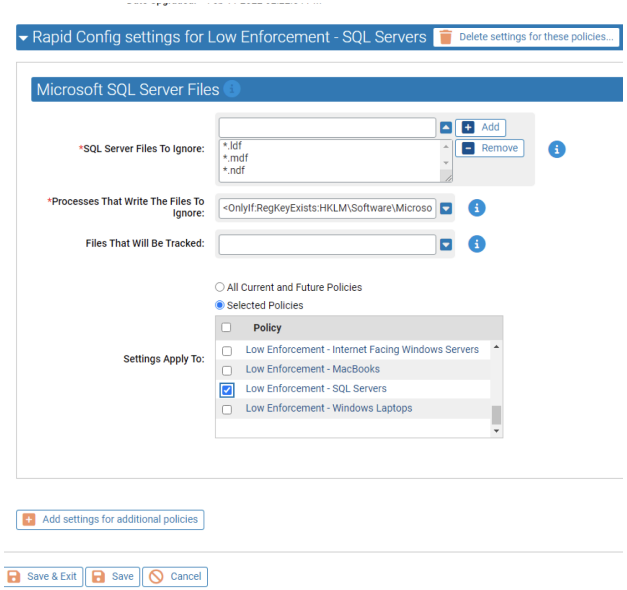
1. Navigate to Rules > Software Rules and click on the Rapid Configs tab.
2. Scroll down to Microsoft SQL Server and click the View Details/Edit button.



3. Change the Status from Disabled to Enabled.



4. Change the Settings Apply To field from All Current and Future Policies to Selected Policies.
5. Scroll down to find the relevant policy (for Microsoft SQL Servers) and select that policy.

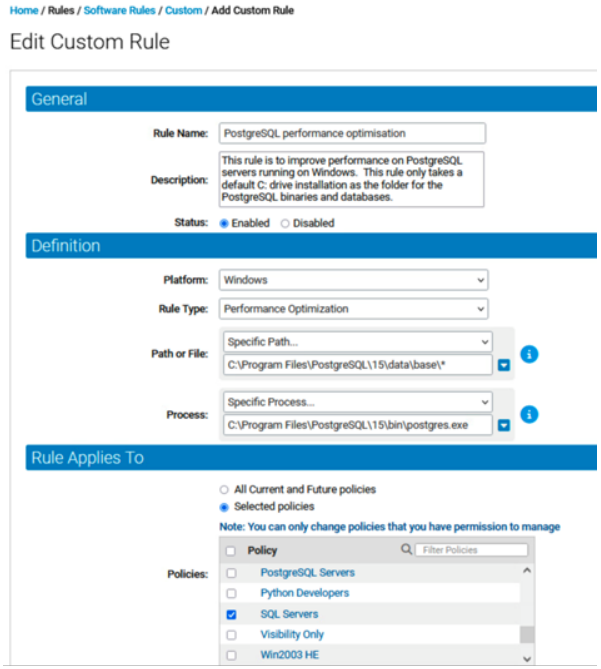


6. Click the Save & Exit button at the bottom of the screen to save the changes to the Rapid Config.

Custom Performance Optimisation rule examples: PostgreSQL on Windows

1. Navigate to Rules > Software Rules and click on the Custom tab.
2. Click the Add Custom Rule button.
3. Name the rule – use a descriptive name so that others know what the rule is for more easily.
4. Add a description that details more information about what this rule is for.
    a. While this is optional it will help with rule auditing in the future. You can also add information on rule edits to the description.
5. Set the Status to Enabled.
6. Select the Rule Type as Performance Optimization.
7. In the Path or File text box, enter the path for the database files for your PostgreSQL installation.
    a. The example screenshot is showing the default location for a Windows install on the C drive. It is possible to use macros here for the path, e.g. <ProgramFiles>\PostgreSQL\15\data\base*

8. In the Process dropdown select Specific Process...
    a. Then enter the full path and filename of the PostgreSQL executable.
    b. Again the example screenshot is showing a default install location on Windows in the C drive. Your path will likely be different.
9. In the Rule Applies To section, change to Selected Policies, then scroll until you find the relevant policies and click the checkbox.
    a. You can select multiple policies if required.
10. Click the Save & Exit button.



Example custom performance optimisation rule

NOTE: The process for this example is not signed by a publisher so there is no point in specifying the Process Publisher in the Advanced section of the custom rule. Where possible, you should pair the process path with the Process Publisher setting.

Additional Resources

For more information about Carbon Black App Control, you can explore the following resources:

- Carbon Black TechZone
- Carbon Black's website
- Carbon Black User eXchange
- Carbon Black App Control Product Documentation

Author and Contributors

- DetectX.com.au

Changelog

The following updates were made to this guide:

Date	**Changes**
08/12/2023	Updates applied for November 2023 changes to the Essential 8 Recommendations.
20/10/2023	Added rule examples for PowerShell Constrained Language Mode test script executions.
02/03/2024	Rewrote the Configuration guide as a simple to read practical implementation guide that follows E8 Maturity level format.