DriveLock Application Control

Documentation 2022.1

DriveLock SE 2022

Table of Contents

1 DriveLock Application Control

1.1 Licensing model DriveLock Application Control

DriveLock offers a range of licenses with a different scope of features.

If you have an EDR license, part of the application control functionality is available to you, which you can use to detect attacks.

	Application Control (Legacy)	Application Control	Application Behavior Control (ABC)	EDR
Whitelisting or blacklisting of applications	yes	yes	-	-
File properties rule	yes	yes	-	-
Hash database rule	yes	yes	-	-
Special rule	yes	yes	-	-
Whitelisting or blacklisting of DLLs	-	yes	-	-
Whitelisting or blacklisting of scripts	-	yes	-	-
Local whitelist	-	yes	-	-
Predictive whitelisting	-	yes	-	-

Application collections	-	yes	yes	yes
Local learning	-	yes	yes	-
Application behavior rules	-	-	yes	Reporting
• File accesses	-	-	yes	Reporting
• Registry accesses	-	-	yes	Reporting
• Script execution	-	-	yes	Reporting
• Starting applications	-	-	yes	Reporting
• Loading DLLs	-	-	yes	Reporting
Application behavior recording	-	-	yes	-

> ☑ Note: The legacy application control license cannot be combined. Both application control with machine learning function (Application Control) and application behavior control can be used individually or combined.

1.2 Features

Use DriveLock Application Control to specifically restrict or allow the use of applications on your corporate computers.

> ☑ Note: Note that DriveLock Application Control is not automatically included in the standard DriveLock installation package. If you have not entered a license for it, this node will not appear in your DriveLock Management Console. Depending on the

license, some functionalities, such as application behavior control, may not be available.

DriveLock Application Control includes the following features:

- Application rules: By blacklisting and/or whitelisting applications, you can define basic rules to determine which applications are allowed to run and which are blocked. This lets you control the use of any application on computers where DriveLock is installed. Different criteria determine whether access is allowed or blocked.

- Application behavior rules: Define exactly what applications are allowed to do, for example, the permissions they get, the directories they can write to, and the processes they can start. By recording application behavior via remote agent control, you can automatically generate application behavior rules.

- Local learning: In addition to the rules you define in policies, you can also make the DriveLock Agent learn locally what DriveLock Application Control allows.

2 Overview in the DriveLock Management Console

You can configure basic settings for application control in the taskpad view of the **Applications** node. From this overview, you can quickly set the scan and blocking modes, configure basic application rules (four special rules) and additional application rules, application behavior rules, application collections and script definitions.

You are also provided with samples of rules that are already preconfigured to represent useful scenarios. If you select the option **Add out-of-the-box recommended block rules** or **Add out-of-the-box sample rules**, the new **Recommended block rules** folder will be created to contain these blacklist rules.

Whenever you change the settings, such as the **scanning and blocking mode**, this is reflected in color (e.g. green, if the current mode is set to Whitelist).

You can also select the individual settings on the left in the DriveLock Management Console. Click **Advanced configuration** to open the corresponding subnode.

3 Settings

You can configure the following settings for DriveLock application control:

1. General settings:
    - Scanning and blocking mode

    - General hash algorithm

    - Always audit application execution

    - Custom user notification message

2. Troubleshooting settings (driver settings)

    > ☑ Note: We recommend using these settings only after consulting DriveLock support.

    - Application control caching

    - Cache lifetime ("time to live")

    - Paths without hash generation for executed applications

3. Setting for trusted processes

4. Activate local whitelist:
    - Local whitelist and predictive whitelisting

5. Settings for local learning:
    - Directories learned for the local whitelist

    - Additional extensions learned for the local whitelist

    - Upload local whitelist to DriveLock Enterprise Service

    - Start learning the local whitelist automatically

6. Settings for application behavior control
    - Duration of the learning phase for application behavior control

    - Ask user in case of unusual application behavior

> ☑ Note: Using conditional settings (configuration filters) is also possible within the application control. For more information, see the corresponding chapter of the Administration Guide at DriveLock Online Help.

3.1 Scanning and blocking mode

When scanning or blocking executables, DriveLock checks the file as the Windows operating system loads it into memory. Based on the results of the scan and the rules configured in the DriveLock policy, DriveLock will allow or deny program execution.

Basically, scanning/blocking DLLs works the same way. When programs load DLLs, all of them are checked as they load.

> ⚠ Warning: If you intend to enable application control in whitelist mode involving DLLs, make sure that you do not block any DLLs that are required for your system to function fully.

Note that Windows installs numerous DLLs that are not identified as part of the operating system or the .NET Framework. Also, not all of these DLLs are installed in the Windows system directory and some do not have a ("valid") Microsoft signature. This is why none of the special rules cover such DLLs.

Example:

Some versions of Windows come with Microsoft OneDrive installed as a standard feature. OneDrive is installed in the user profile and is not part of the operating system. However, the Windows Explorer reloads OneDrive DLLs. Windows Explorer will quit if these DLLs are not whitelisted in your rules.

Best practice:

We recommend that you enable predictive whitelisting or local whitelisting before you enable DLL blocking. Always start in simulation mode and evaluate the application control events in order to whitelist all DLLs required by the system.

3.1.1 Simulation

Use one of the two simulation modes, Whitelist (simulate) or Blacklist (simulate), to test templates or rules before actually blocking programs. In simulation mode the DriveLock Agent creates events when an application is started that is controlled by a template or rule, but no programs are blocked.

Use the simulation modes to identify applications that users are running before you enforce any blocking rules. You can easily use the Windows Event Viewer for analysis or use DriveLock Operations (DOC) or Control Center (DCC) to find relevant events quickly.

3.1.2 Whitelist or Blacklist

To fully enable application control, select Whitelist or Blacklist from the drop-down list.

If you select Whitelist, all applications will be blocked unless there is a suitable application rule that removes this block.

Blacklisted applications, by contrast, do not initially prevent any application from running unless there is a specific rule that blocks them.

3.1.2.1 Whitelist mode

In whitelist mode, all applications are allowed that match a whitelist rule. Using blacklist rules, you can block individual applications in this case as an exception to an existing whitelist rule or template.

Priority: blacklist rule - whitelist rule - other settings

To allow all users to run all programs in the Program Files folder, create a directory rule and allow all applications within this folder to run. To prevent one of these applications from running on one computer, create a blacklist rule for only this application and apply it to the computer.

3.1.2.2 Blacklist mode

When using the blacklist mode, all applications are allowed to run unless they are listed in blacklist rules or templates. Use blacklist rules or templates in this mode to specify the applications that users are not allowed to start. Use whitelist rules in this mode to define exceptions to blacklist templates or rules.

Priority: whitelist rule - blacklist rule - other settings

Example: Users in your organization are not allowed to run the program "Skype". However, your CEO must use Skype when being out of the office. To allow this, create a blacklist rule to block Skype for all uses. Then define a whitelist rule allowing the Skype application and configure it to apply to only the CEO's account.

3.2 General hash algorithm

Use this setting to specify a fixed hash algorithm that will be used for reporting, for the local whitelist, and for creating new rules.

You can change the hash algorithm later or use a different hash algorithm in rules. In this case, the agent may have to calculate multiple hashes of a file, which can lead to slight performance losses.

> ⊘ Warning: DriveLock Agents prior to version 2022.1 only use the configured hash algorithm, which means that rules with a different hash algorithm will not work on these agents.

We recommend the hash method SHA-256 shown in the example.
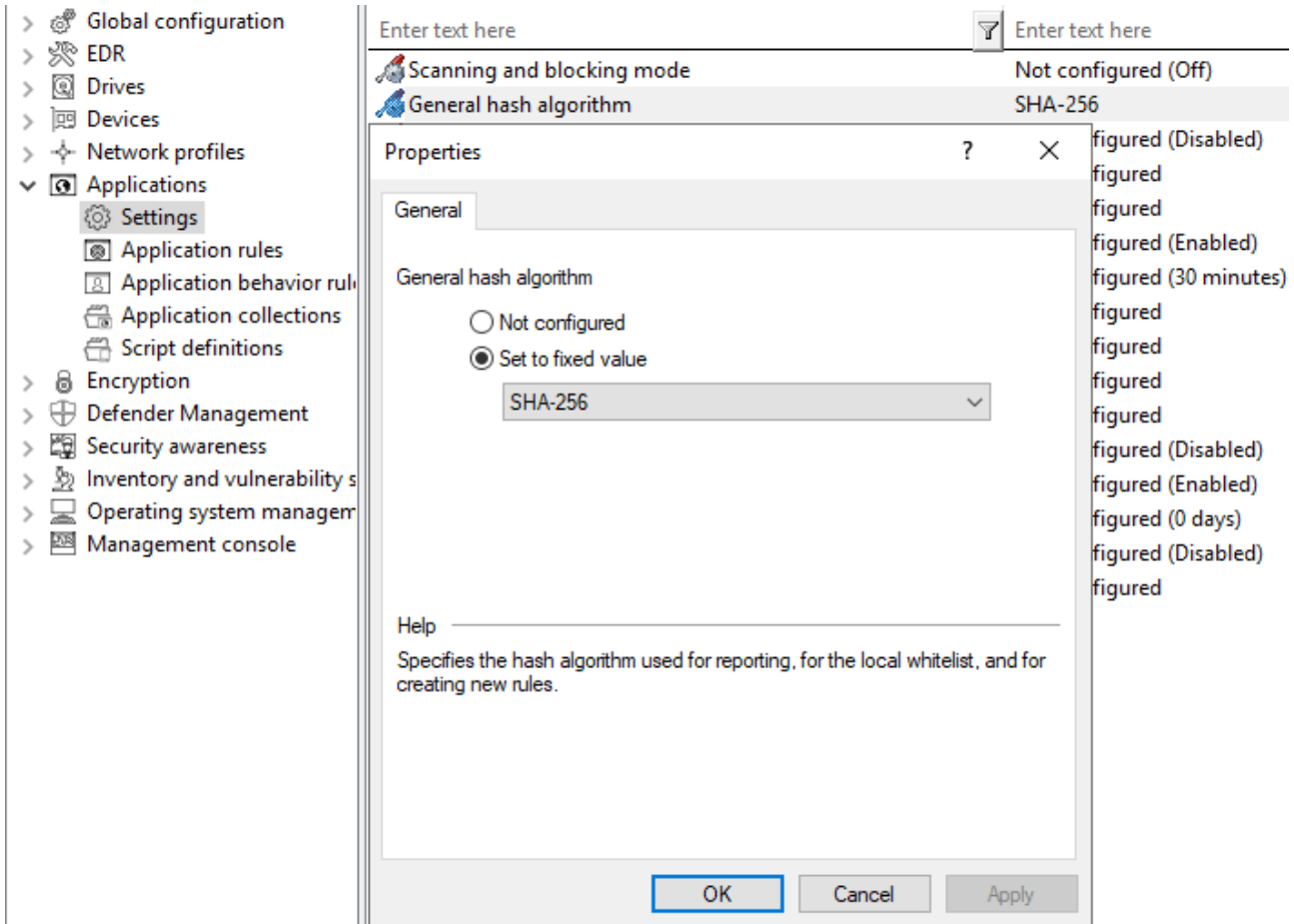
3.3 Always audit application execution

If you want to collect information as events about started programs independent of the selected operation mode, choose **Always audit application execution (independent of blocking mode)** and check **Enabled**.

Note: However, logging each successful program startup can slow down system performance. Sending all events to the DriveLock Enterprise Service also increases the network load and database size.

3.4 Custom user notification message

You can define a **custom user notification message** for each whitelist rule. Unless specified otherwise, DriveLock will display this message when the Application Control blocks an application.

If you configured a multilingual message text for the current language, DriveLock will display the standard messages defined for this language instead of the message configured in this dialog box.

Select **Display custom message** to enable the messages and type the message to be displayed to the user. Use the `%EXE%` variable in the message to inform the user of the name of the application that was blocked. It is replaced by the path and file name at runtime.

Click Test to preview the message.

3.5 Trusted processes

This setting can be configured if you are using client management software for software distribution in your company. On the Local Learning tab in some application and application collection rules, you can also specify whether this client management software is given special permissions (for example, whether it can start other programs that are not on the whitelist) and is therefore considered trustworthy.

The following configuration options are available:

1. **Not configured** is the default option.

2. **Set to configured list**:
   Add the name of the software. This software is checked when the DriveLock Enterprise Service starts.

3.6 Local whitelist and predictive whitelisting

This central setting enables or disables the use of the local whitelist.

The following configuration options are available:

1. **Enable local whitelist**:
   Once the policy with this setting is assigned, the DriveLock Agent starts the learning mode and afterwards activates the local whitelist with the learned applications.

2. **Enable predictive whitelisting** in connection with **Enable predictions based on publisher certificates**:
   Particularly during update processes, this option ensures the following automation: Files are automatically added to the local whitelist provided that they match the product description or are signed by a similar certificate as the ones of the files learned in the local whitelist.
   Use this option to quickly and easily allow update processes (e.g. of browsers). Creating well-defined rules for updating applications via local learning (for example, using learning behavior recording, using the recording results in application behavior rules, or specifying permissions accurately) is more time-consuming, but it gives you a more reliable result.

3.6.1 Display local whitelist via agent remote control

If you are using DriveLock Application Control with local learning, the system creates a database on the DriveLock Agent that contains the applications allowed for that computer (local whitelist). You can connect to an agent and view the contents of this database or delete individual entries.

Display application control whitelist:

1. Open the **Operating** node in the DriveLock Management Console and select **Agent remote control**.

2. Select **Display local application control whitelist** from the context menu of the relevant DriveLock Agent.
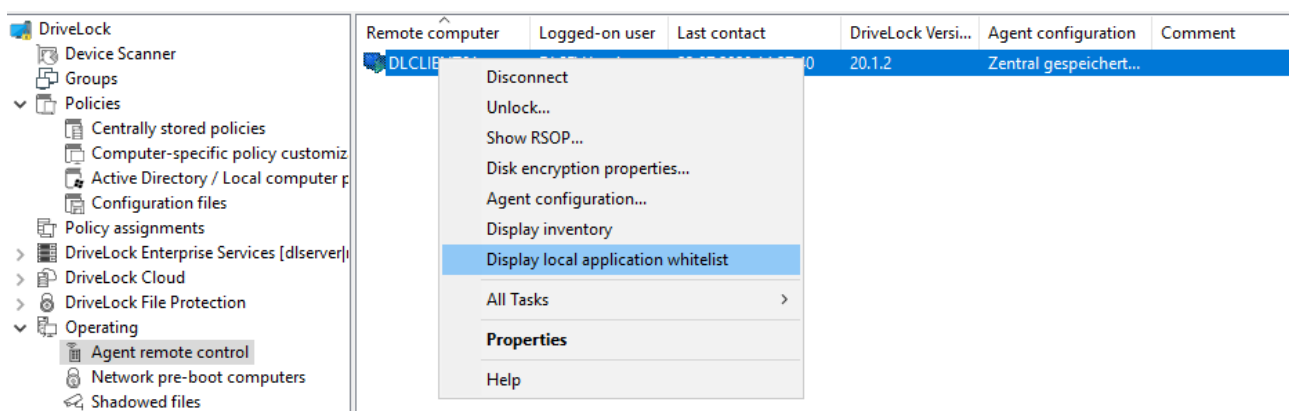
If you want to delete individual entries, possibly because too many applications have been learned, proceed as follows:

1. Double-click the relevant agent to display its properties.

2. On the **Application Control** tab, select the **Display...** button.

3. A window with a structure similar to Windows Explorer opens. Opening the database may take some time depending on its size.

4. You will see the learned applications here. Select the entry you want to delete.

> Note: Refer to the Administration Guide on DriveLock Online Help for more information about agent remote control.

3.6.2 Local learning

DriveLock Application Control features a learning functionality that can be used to learn the behavior of applications on DriveLock Agents.

This is accomplished by enabling the client computer to enter learning mode and creating a local whitelist (hash database) of installed programs and DLLs. This individual local whitelist then contains the approved files that have been learned locally. Once the learning mode is completed, the local whitelist is activated and only the "learned" programs can be executed. To ensure that programs that are installed or updated at a later time are not blocked by application control, the learning mode can be temporarily reactivated for installations or updates.

You can activate local whitelisting either by configuring the Local whitelist and predictive whitelisting setting or by creating a Predictive Whitelisting rule.

Local learning is triggered

- by specifying the corresponding learning settings in an application list rule or

- by using an application behavior rule that was automatically created from an application behavior recording.

When the local whitelist is activated, you can define additional settings to configure the learning functionality.

The local whitelist is merged incrementally with the application database on the DriveLock Enterprise Service (DES). When you create file properties rules, you can also select from this global application database.

3.6.2.1 Application behavior recording and control

There are two ways to partially or fully automate application behavior control.

1. Using a reference computer
   You can easily track and learn background actions, such as access from applications, running programs, or written files, with the help of behavior recording. The results of this recording can be stored in a file.
   - On a reference computer, enable application behavior recording for one or more applications using the Agent remote control functionality.

   - You will then work with these applications, making sure that all important actions are performed, especially updates and configuration changes. This involves recording the behavior of the applications, such as determining which files are written and which other programs are started.

   - Then you can generate application behavior rules from the recorded data.

2. Automatic learning on individual DriveLock Agents
   - With an application collection rule, you can specify that the behavior of an application is restricted to the actions that are learned during a learning phase. In this case, only the access modes Execute, Load DLL and Write file are supported.

   - During a learning phase, the system learns how the application behaves and after completing the learning phase, any deviating behavior will be blocked.

3.6.2.1.1 Configure application behavior recording

Start a behavior recording to find out how an application behaves.

> ☑ Note: Make sure that the application has been whitelisted.
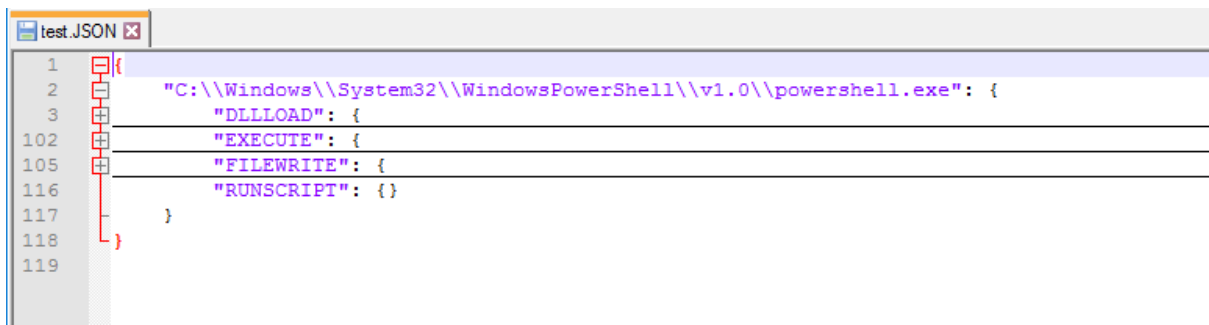
Once you have saved the behavior recording, you can then use it to generate application permissions that are restricted to precisely the learned behavior. This way, only the behavior that is actually needed will be allowed, everything else will be blocked.

Please do the following:

1. Open the **Operating** node in the DriveLock Management Console and select **Agent remote control**.

2. Double-click the relevant agent to display its properties.

3. Select the **Start** button on the **Application Control** tab in the **Application behavior recording** section.

4. Add directories or programs whose behavior you want to record.

5. Select which kind of accesses you want to record, see example.
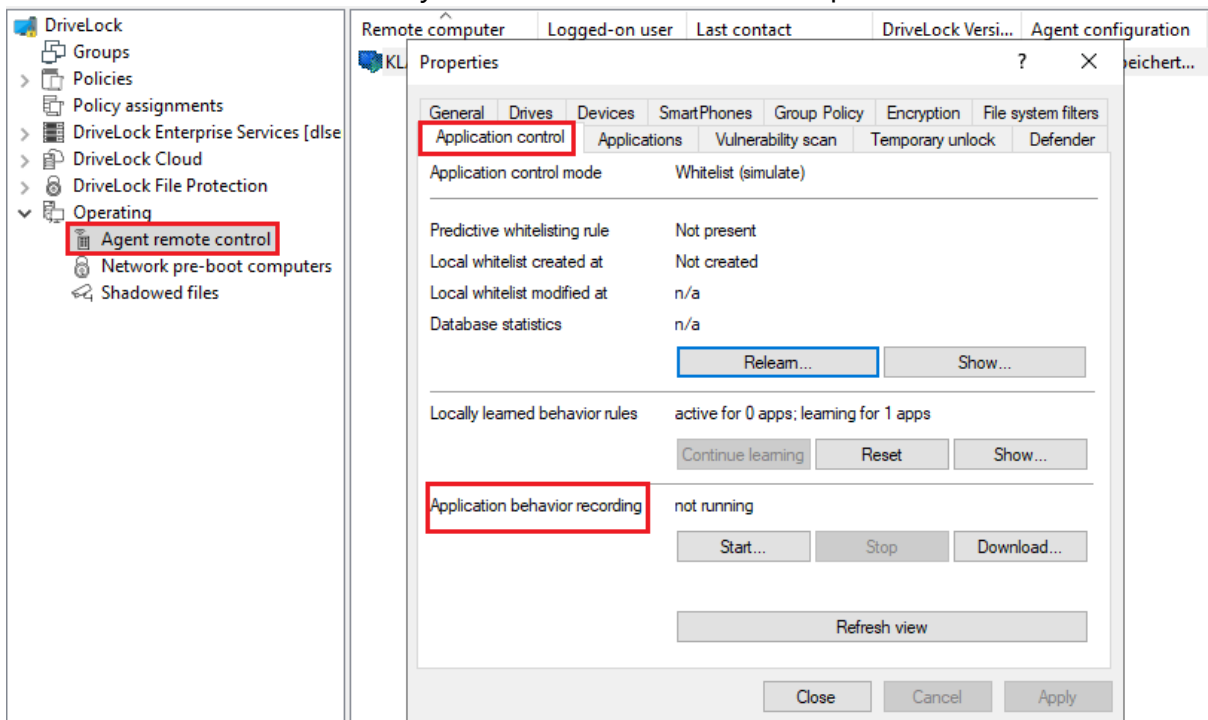
6. If you want to delete a recording that already exists, select the checkbox.

7. It is recommended to limit the recording to a certain period of time. You can enter a maximum of 10 days here, but we recommend a much shorter period.

8. Once you have tested the application, for example on a reference computer, for a certain period of time and collected a sufficient amount of data, click **Download...** to download the behavior recording in a JSON file and evaluate the results.

```
test.JSON
   1    {
   2        "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe": {
   3            "DLLLOAD": {
 102            "EXECUTE": {
 105            "FILEWRITE": {
 116            "RUNSCRIPT": {}
 117        }
 118    }
 119
```

9. You can now use this results file in an application behavior rule.

3.6.2.1.2 Locally learned behavior rules

The information you see in the **Locally learned behavior rules** section reflects the settings you defined in the application collection rules on the **Local Learning** tab. As soon as an agent uses a policy with these settings, a learning phase is started, thus activating application behavior control. The learning phase for the three modes (load DLL, execute, write files) are independent of each other.

The following states and buttons are available:

- **not active**: There are no applications specified yet that need to be learned or controlled.

- **active for**: The specified number of applications is blocked when a behavior is detected that has not been learned yet.

- **learning for**: The applications are still in the learning phase.

- **Continue learning**: The start time of the learning phase is reset, the list that has been learned so far is continued.

- **Reset**: The list that has been learned so far is deleted. The activity display returns to **not active**.

- **Show...**: Clicking on this button opens a dialog in which the learned entries are displayed, see figure.
  If you save the result in a JSON file, you can use it to have application behavior rules generated from it. To do this, proceed as described in chapter Generate application behavior rules from behavior recording.

3.7 Settings for local learning

You can configure the following settings for local learning:

Setting	Configuration options
Upload local whitelist to DriveLock Enterprise Service	Once created, you can have the local whitelist sent to the DriveLock Enterprise Service (DES), which maintains a list of all locally learned files. This list can then be used to generate hash rules. The default option is **Disabled**.
Start learning the local whitelist automatically	Use this setting to define whether local whitelist learning is started automatically (i.e. as soon as the corresponding policy is assigned to the DriveLock Agent) or by users. The default option is **Enabled**. Select **Disabled** if you want to wait until a user actively starts learning. This means that the user is responsible for the initial learning of the local whitelist. You can configure the settings of the agent user interface accordingly. To do so, go to the **Global configuration** node, select Settings and then the **User interface settings** sub-node and then **Task bar notification area settings**. Here you can add the context menu item **Initial local whitelist learning**. Note: Keep in mind that application blocking is disabled in this case until the user has initiated learning.
Additional extensions learned for the local whitelist	You can specify additional file types in addition to the standard file types to add to the local whitelist. This is useful if an application uses a different file extension for a file type, or in order to learn scripts that are already running on the system.

Setting	Configuration options
Directories learned for the local whitelist	Typically, the files are learned from all local hard drives. You can restrict the learning process to certain directories where the software you want DriveLock to learn is located. Enable the setting by specifying the directories in the list.

3.8 Settings for application behavior control

You can configure the following settings related to application behavior control:

Setting	Configuration options
Duration of the learning phase for application behavior control	This setting lets you specify a period of time during which an application learns and records everything it will do on the DriveLock Agent. The corresponding rules are generated based on the learned behavior. The default option is **Not configured**. Choose **Set to value** to specify a time period. Once the application is started the first time, a countdown begins. After the time is over, everything that does not comply with the learned behavior is blocked.
Ask user in case of unusual application behavior	When application behavior control is enabled for a DriveLock Agent and the learning process has been completed, any application behavior that differs from what was learned is considered 'unusual'. The default option is **Disabled**. Select **Enabled** if a user must confirm or reject the unusual behavior. Behavior confirmed is subsequently learnt.

4 Application rules

The following application rules are available:

- Application hash database:
  With the help of hash databases, you can allow or block all applications contained in the database with a single rule. Hash databases are created easily by automatically searching through predefined directories. Click Application hash database to create this type of database and enable it via a rule. For example, you can create a hash database from a reference PC that contains all your company programs. If you apply this rule to other computers in your organization, all applications that are also installed on the reference PC are automatically enabled, while all other programs are blocked by DriveLock.

- File properties rule:
  This rule allows filtering by a number of different file properties.
  The following rules from previous versions (before 2020.2) are combined into a single rule: file path, file owner, hash, and publisher certificate rule.

  > ⚠ Warning: If you have been using one or more of these individual rules in a policy in an older DriveLock version (before 2020.2), they are automatically converted to a file properties rule, taking over the properties set in each rule. The file properties rules are only compatible with pre-2020.2 DriveLock Agents, if the property combinations in the new rule exactly match the corresponding property options from the old rule types.

- Special rule:
  The special rules make it easy to identify all program files on a computer that meet a certain criterion such as whether a file is part of the Microsoft operating system, a part of DriveLock, or a .NET program. You can also use the special rule to override a blacklist rule, for example, so that some users, such as the service administrators, can run all programs.

- Predictive whitelisting rule:
  Use this rule to enable predictive whitelisting. The settings in this rule override the Local whitelist and predictive whitelisting setting.

- Application collection rule:
  Apply this rule (available from version 2020.1 and higher) if you want to use existing application collections, and especially to activate the learning settings for applications.

- (deprecated) Application template:
  This rule is only available for backward compatibility for older DriveLock versions.

- Create **folders** in the **Application rules** node to group rules by subject, for example, by vendor or type of software, for easier management.
  In order to control processes such as browser updates, for example, it is practical and convenient to store all the application rules required to do so in a folder named after the browser. You can also assign appropriate access rights.

4.1 Different rule types

When configuring application rules, you can specify different rule types:
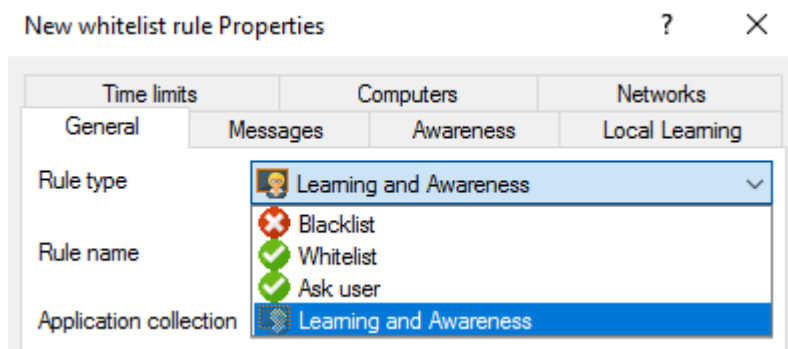
- **White or blacklist rules**: These rule types specify which applications are allowed to run on the DriveLock Agent or which are prohibited and blocked.

- **Ask user**: With this rule type, an application is allowed (whitelist), but the user must confirm its start.

- **Learning and Awareness**: This rule type ensures that only the learning settings on the **Local Learning** tab take effect or that the awareness campaigns specified on the **Awareness** tab are displayed. This means that you can configure settings for an application without actively allowing (whitelist) or blocking (blacklist) it.
    - The **Local Learning** tab appears in the following rules: File properties rule and Application collection rule.

    - Here you can find out how to use the settings on the **Local Learning** tab.

    - You can find a sample configuration for displaying an awareness campaign here.

4.2 File properties rule

This rule allows you to specify different file properties to filter by. Along with some additional options, this rule combines the file owner, file path, hash, and publisher certificate rule options from previous versions.

> ⚠ Warning: The file properties rules are only compatible with pre-2020.2 DriveLock Agents, if the property combinations in the new rule exactly match the corresponding property options from the old rule types. For example, if you combine the path with the owner and the publisher, the (old) agent cannot interpret the rule type accurately and will therefore ignore the rule.

Please do the following:

1. **Path**: Start here by selecting a path from which to start (or block) applications, or a specific file within a directory. To do so, click **...** . This option checks if the path of the file meets certain conditions.

> 📝 Note: The other boxes in the dialog will be filled in automatically as soon as you have made a selection here. Then, check the options you want to filter by.

You can also select an application from the list of currently started programs (option **From running processes...**) or from the application database (option **From application inventory...**).

To view information about currently running applications from another computer where DriveLock is installed and running via the remote connection, select the **on Agent** option, enter the name of the computer, and then click **Connect**.

Also select one of the two options in the drop-down list:

- **matches**: applies if the path corresponds to the specified text, where wildcards may be used. If the text does not contain backslashes, only the file name is checked.

- **contains**: applies if the specified text occurs anywhere in the file path.

2. Then assign a **rule name** and select the **rule type**, that is, the way the rule will be implemented. For more information, please visit here.

3. **Hash**: This option verifies that the hash value of the file contents matches the specified value. The system stores this value when creating the rule and compares it with the currently calculated value at runtime. If both match, the rule is activated. Use this option, for example, for a single application that you want to allow or block via whitelist or blacklist.

4. **Owner**: Use this option to restrict the starting of an application to a specific file owner. For example, you can use this setting to allow all programs installed by an administrator or by a trusted installer account, while blocking all applications that were installed by other users. This also allows for automatically blocking all applications that can be run without prior installation.

   The following options can be selected or are entered automatically depending on the selection:

   - **Administrators group**: This option covers all local administrators. To allow the file, the administrators group must be the explicit file owner.

   - **Trusted Installer** and **Local System**: These default Windows accounts must be file owners so that the file is allowed.

   - **AD user or group**: Select an AD user or group as file owner here. This is where the SID is checked.

- **Name (user / group)**: You can manually add a user or group here. Here the name is checked.

> 📝 Note: If you assign a group, the file owner must be the group, not a member of that group.

5. **Description**: Enter the file description here, e.g. 'Paint' for the mspaint.exe file.

6. **Version**: You can have the version checked to prevent users from running other or older program versions, e.g. you can allow Firefox version 83.0.0.7621 or higher and block all previous versions that might contain security vulnerabilities. Select the appropriate option from the drop-down menu, e.g. greater than or equal to.

7. **Product**: Enter the product name here, e.g. Microsoft Windows operating system.

8. **Certificate validation**: This option allows you to whitelist signed software or blacklist unsigned software.
   You can also use the browse button to select certificates via the application inventory.

> 📝 Note: Note that Windows files are not signed. You must also enter a file path here, for example.

9. **Subject**, **Issuer**, **Thumbprint** and **Serial number** are additional certificate properties. The serial number is only unique in combination with the publisher.

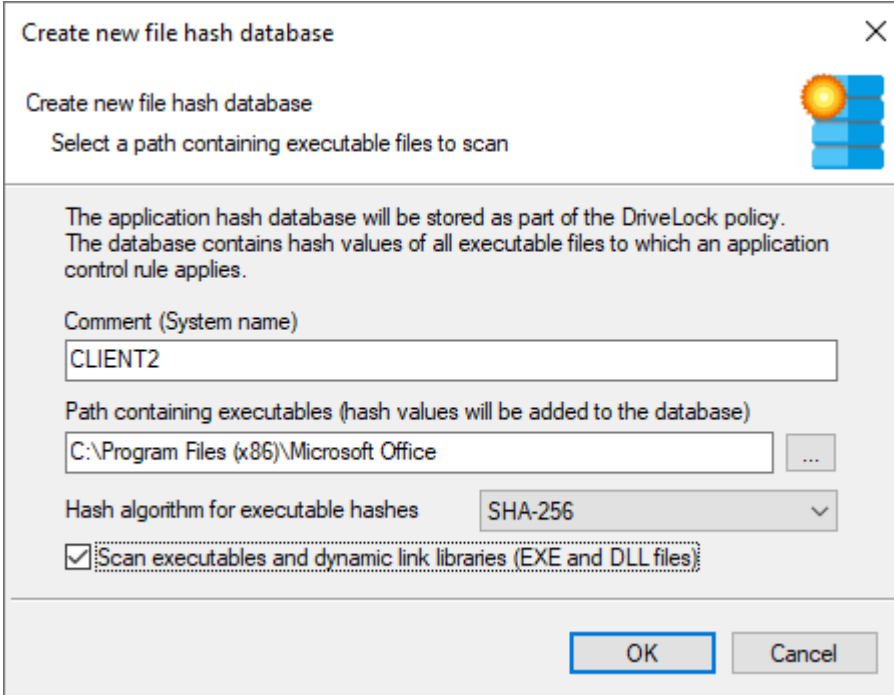### 4.3 Application hash database

For easier application control configuration, DriveLock Application Control provides the ability to create application hash databases and use them in white or blacklist mode. Hash databases can be created by automatically searching for applications in a directory or directories (and their child directories), calculating their hash values and saving them to a file. A hash database of all installed programs can also be created from the hard disk of a reference system.

Follow these steps to create an application hash database:

1. In the **Applications** node, select **Application rules**. Next, select **New** from the context menu and open the **Application hash database** dialog.

2. Initially no database is selected on the **General** tab. You can either create a new database file or select an existing one.

> ☑ Note: DriveLock provides a utility program **DriveLock Application Hash Database Tool** that can also be used to generate a hash database. The utility is located in the installation directory of DriveLock (C:\Program Files\CenterTools\DriveLock MMC\Tools\DLExeHasher.exe).

3. The value that is already preset in the hash procedure is listed in the **Hash algorithm used in database** section.

4. To create a new database, click **Database file** and then click **Create new**.

5. In the Comment (System name) box, type the name of the computer to be scanned. With this information, it is easier to assign multiple database files during a migration at a later date. Type or click **...** to select the directory to be scanned for applications.

> ☑ Note: You can scan a directory on a remote computer by specifying the UNC path for this directory.

The **Hash algorithm for executable hashes** defines the algorithm used for this database. Initially the general hash algorithm is set here. Select **Scan executables and dynamic link libraries** to scan DLL files in addition to EXE files.

6. Click **OK**. DriveLock starts a recursive scan of the specified directory and all child directories below it.

> ☑ Note: Please note that scanning larger directories or UNC paths may take some time. Please do not interrupt the process.

> ☑ Note: No duplicate entries are generated during the search. If it finds the same file in a different directory, DriveLock does not add the hash value to the hash database again. This has no effect on how the rule is applied because applications are evaluated based on their hashes and not a specific location. Also, this behavior allows for differential scanning, which only adds applications that are not already in the database.

7. When DriveLock has finished detecting all program files and has calculated all hashes, it adds all applications it detected to the template and displays the previous dialog box.

8. Add a description (**Rule name**) and enter additional information in the **Comment** text box if necessary.

9. Click **Database content** to view, edit or merge the programs that are included in the database.

10. Click **Database content** and then click **View / edit** to view the database content.
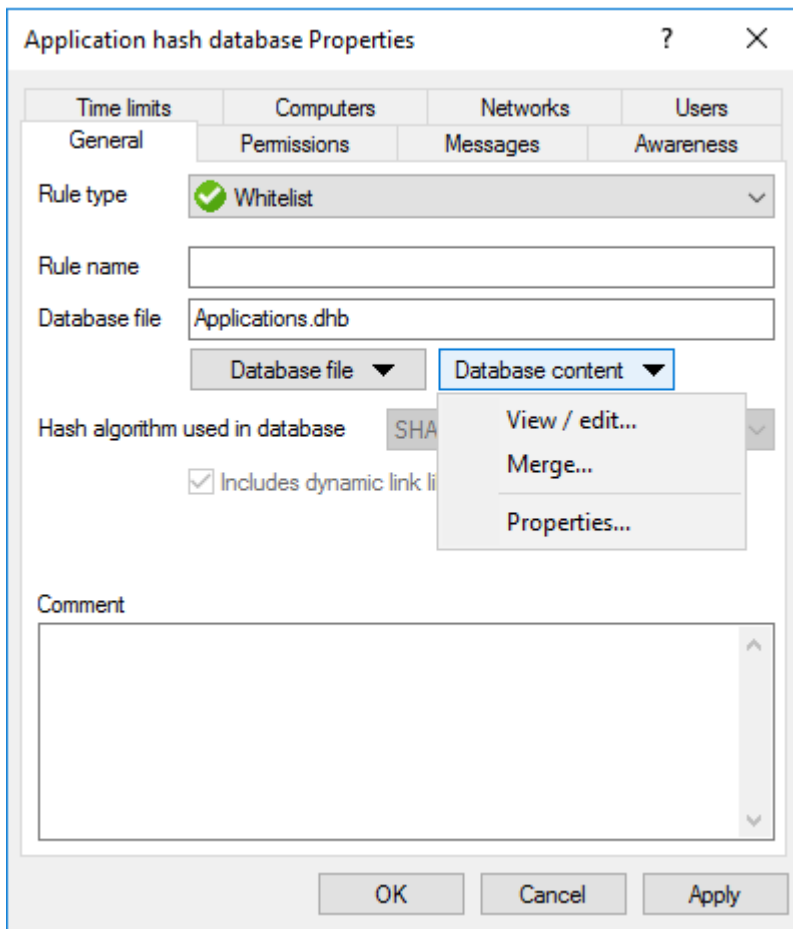
11. The left pane displays the folders that were scanned. Select a folder to display all programs that were found in this folder in the right pane.
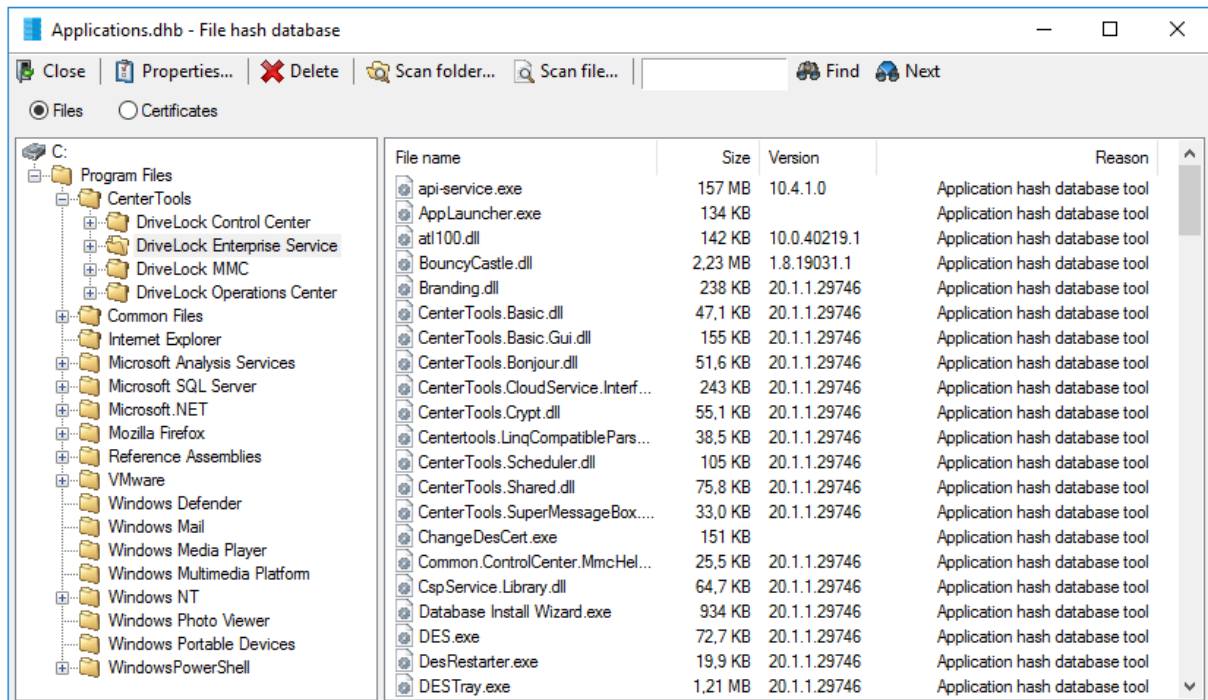
12. To add additional hashes, click **Scan folder** or **Scan file**. Click **Delete** to remove the selected application hash or folder. To view additional information about the hash database, click **Properties**.

13. To close the hash database viewer, click **Close**.

> ☑ Note: You can also use the standalone Application Hash Database Tool, DLExeHasher.exe, to view, edit and merge hash databases.

14. Click **Database content** and then click **Merge** to add the content of another database.

15. Type or select the path of the database file containing the entries to be added. Alternatively you can use the file selection dialog.

16. Click **OK** so that DriveLock merges the database content.

17. Then it displays the template properties again.

18. Click **OK** to exit the dialog and save the changes.

> ☑ Note: Even if you are using a whitelist rule based on a hash database of all installed applications to control a computer, it is recommended that you also use some special application rules for programs that are part of the operating system. DriveLock loads these special rules faster that data from the hash database and they are available earlier to the DriveLock Agent when Application Control starts.

4.4 Special rule

> ☑ Note: Special rules can only be used as whitelist rules.

You can select from the following options in the dialog:
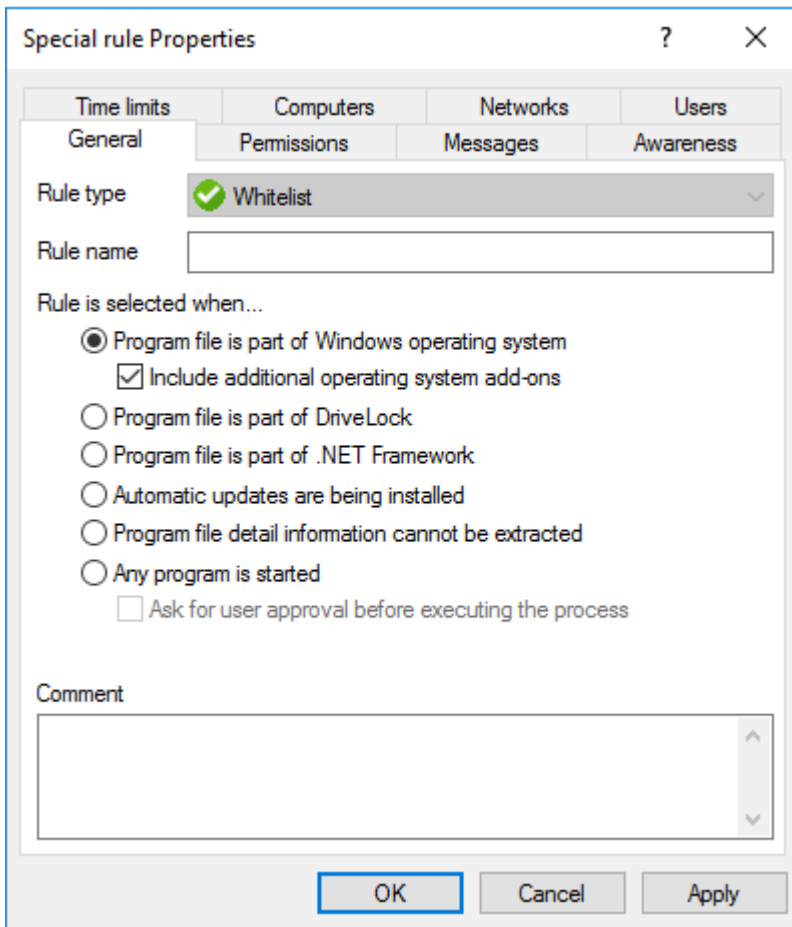
1. Program file is part of the Windows operating system
   - includes all programs protected by the Windows System File Protection (WFP)

   Include additional operation system add-ons addresses programs in
   - C:\windows

   - C:\windows\system32

   - C:\windows\servicing

   - C:\windows\pchealth\helpctr\binaries (Help Center)

   - C:\windows\application compatibility scripts

   - C:\windows\explorer.exe

   - C:\Programs\Internet Explorer

   - C:\Programs\Windows Defender

2. The program is a component of DriveLock
   - all programs in the DriveLock installation directories

3. The program is part of the .NET Framework
   - all programs in C:\Windows\Microsoft.NET

4. Windows Automatic Updates are being installed
   - all processes initialized by the Windows Update Agent

5. Program file detail information cannot be extracted
   - can be used as a fallback if for any reason DriveLock is not able to access or read information details from a specific file

6. Any program is started
   - can be used in conjunction with rule limitations for example, to allow access to all programs for the Administrators group, optionally including a user approval before executing the process.

   > Note: This user permission does not affect the priority of the rule.

4.4.1 Basic application rules

To create basic application rules, click **Change...** in the Taskpad view.

Select the type of rules to use and then click Finish. DriveLock creates the corresponding special rules.
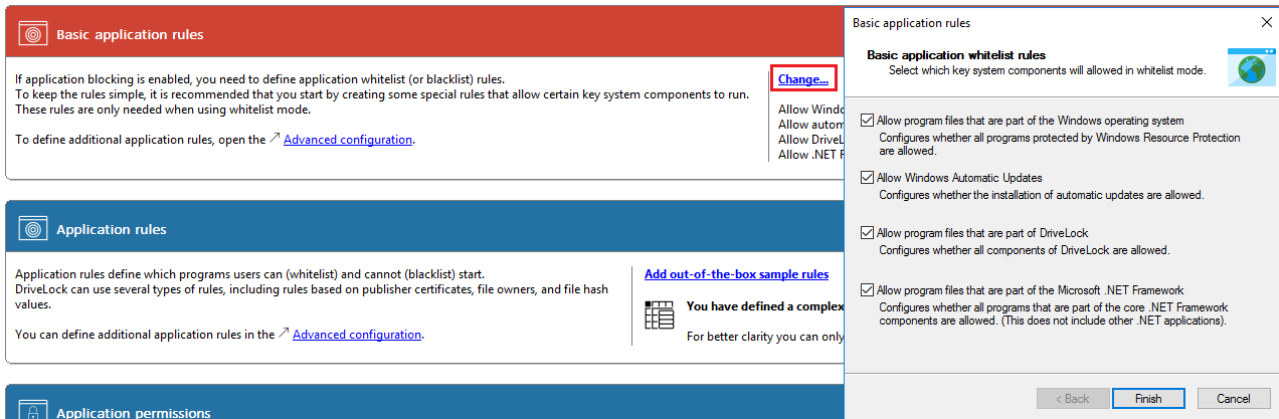
4.5 Predictive whitelisting rule

Predictive whitelisting rules are only applicable as a whitelist rule.

Specify the following options in the dialog:

35
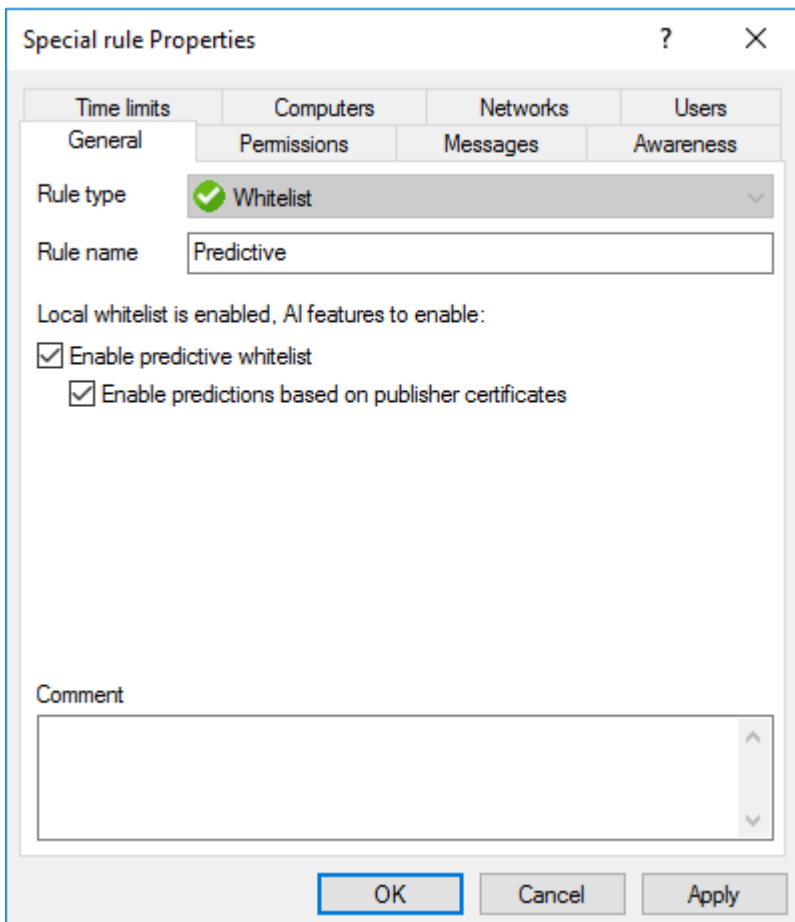
By selecting the **Enable predictions based on publisher certificates** option, DriveLock uses algorithms to detect new versions of signed software even if the certificates are not completely identical.

See also the Local Whitelist and predictive whitelisting setting.

> Note: Note that this setting only works if the new version can be recognized properly.

4.6 Application collection rule

> ✏️ Note: This rule has no user restrictions.

The task pad view provides you with two samples that you can use immediately. With one rule you can learn and control the behavior of different browsers and with the other one that of different e-mail clients (the corresponding application collections are created simultaneously in the **Application collections** folder).

Based on the behavior of browsers during updates, the following example explains the dialog options:

1. The **General** tab contains the following information:
   - **Rule type**: Learning and Awareness
     The **Learning and Awareness** option only controls the learning settings, but does not determine whether a specific program may be started or not (as would be the case with the white or black list options).

     > ✏️ Note: This decision is based on the hashes of the files (in hash rules), which are automatically managed by Application Control.

   - **Rule name**: Learn the behavior of browsers

   - **Application collection**: Browsers
     Make sure that the application collection contains all common browsers and exists already.

2. The following options are available on the **Local Learning** tab:
   - **The application may start programs that are not included in any whitelist**:
     By selecting this option, any service process that is to execute a browser update can be started, even if this service process is not explicitly whitelisted. This option also allows the service process to start the actual browser update, which is not whitelisted either.

   - **Learn all program files written by this application (including child processes)**
     To enable the browser update to terminate the actual browser and service process and to replace the corresponding files with the updated version of the browser, all child processes of the service process must be automatically added to a whitelist.

This means that the actual browser, being a child process of the service process, will be able to start programs that are not explicitly allowed. In addition, all the files that the browser writes are also automatically added to the whitelist.

As neither of these options are wanted for browsers, it is important to configure the browser so that such permissions are not passed on to the process. This is why you select the following option:

- **This application never gets the permissions listed above**

In the section **Learn and control application behavior** you also specify that browsers learn locally

- which programs they start,

- which DLLs they load and

- which directories they are allowed to write their files to.

Conclusion: With these settings, the applications that are specified in the rule get exactly the rights they need on the respective DriveLock Agent where the application behavior is recorded. In this way it is even possible to learn different download directories for applications on different agents.

4.7 Application template (deprecated)

Application templates can contain one or more applications that are either blocked (blacklist) or allowed (whitelist).

> ⚠ Warning: Please note that this application rule is obsolete and should not be used anymore. If you still need information on this, please refer to the Administration Guide. We recommend using application hash database rules instead.

5 Application behavior rules

Use application behavior control to accomplish the following results:

- Prevent an application (or process, script) from being started from within an allowed application, thus causing a potential danger to your system.

- Specify which type of access you want to grant a particular application (e.g. read or write access to files or the registry).

For this purpose, the following options are available. You can...

- determine in which order (priority) application behavior rules are processed,

- specify the action to be taken when a particular application is accessed (for example, the application is blocked or not),

- determine whether an application permission can be passed on to child processes,

- specify different file and folder filters or

- specify script types that are allowed for running scripts.

Also, starting with version 2020.1, you can create a behavior rule based on a stored recording of application behavior on the DriveLock Agent.

All application behavior rules can be arranged in the DriveLock Management Console in a user-defined folder structure.

5.1 Defining application behavior rules

You can create application behavior rules as follows:

1. In the Taskpad view, select **Add behavior rule...** or click **New** in the context menu of the **Application behavior rules** subnode, creating a new behavior rule. In this context menu you can also create **folders** to group related application behavior rules.

2. In either case, the properties dialog box as shown below will appear, allowing you to enter your details.

3. Enter a description on the **General** tab and add a comment if necessary.
   In the figure below you can see one of the supplied sample permissions.

4. The **Enable rule** option is set by default.

5. The **Priority** option provides you with several choices.
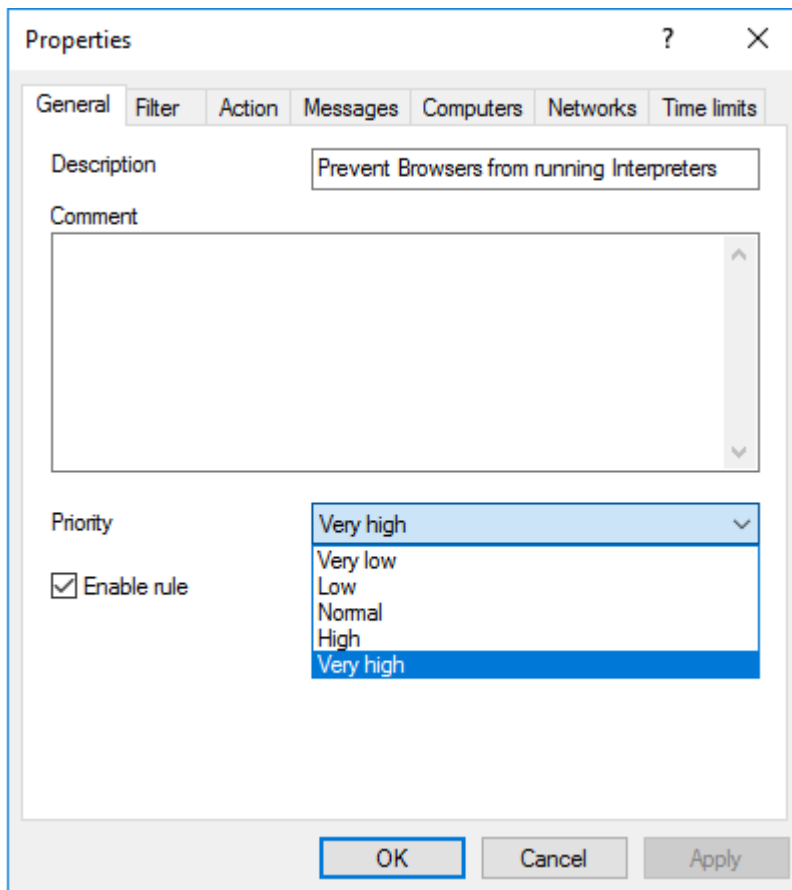
> Note: Generally valid application behavior rules get a lower priority, special ones a higher one. The priorities vary according to the use cases. High-priority rules are processed before low-priority rules. The system checks the rules in the specified order, and if a rule matches, it is applied.

You can reduce or increase the **priority** in the DriveLock MMC.

Example: Combine rules, e.g. create a rule that allows the Browser to start Windows Media Player with high priority and another rule that forbids the Browser to start any other programs with a lower priority.

6. Continue your input on the Filters, Action, Messages and general settings for rules and permissions (Computers, Networks, Times) tabs.

See the use cases described for specific examples.

5.1.1 Information on the Filter tab

The following settings are available here:

1. **Accessing application**

Here you can either specify the full path or the name of the application you want to control, e.g. C:\Program Files\Mozilla Firefox\firefox.exe or just firefox.exe. You may use wildcards.

Note that you can select application collections here, provided you've created them already. Please refer to the corresponding chapter for more information.

2. **Pass on to child processes**

Select this setting so that your application permission is valid not only for the processes that meet the **Accessing application** requirement, but also for all children. This setting affects not only the immediate child processes, but all of their children as well.

> Note: This is particularly useful if you select **Block** as an action on the **Filter** tab because it prevents your application behavior rules from being bypassed by starting another process.

Example: You create an application permission that prohibits your browser from starting Powershell. By selecting this option you can prevent Powershell from being started from the command line anyway (which is a child process).

3. **Access mode**

The access mode is a filter parameter for the application permission. Here you can define the action the accessing application should take.

4. Additional specifications (target)

Depending on the access mode you choose, you enter different targets in the next text box (a path can be specified in all cases).

> Note: Starting with version 2020.1 you can enter several specifications here. This reduces the number of rules.

Access mode	Target	Explanation
Execute	Started application	Enter the name of the application that is not supposed to be started (in this case, you would choose Block as an action). Optionally, you can specify a **command line parameter** here that will restrict the

Access mode	Target	Explanation
		execution of the called application to a greater extent. Use case 1 Note that you cannot enter parameters in Windows XP!
Load DLL	DLL name	Enter the DLL that may only be loaded from a specific directory, for example. Use case 2
Run script	Script name	Enter the script you want to restrict from running. Use case 3 Please note that DriveLock only considers the script types defined in the Script definition subnode.
Read / write file	File name	Enter a file name or a directory the accessing application is allowed (or not allowed) to read or write to. Use case 4 for read access Use case 5 for write access
Read / write registry	Registry key	Enter the respective registry key (e.g. `HKEY_LOCAL_MACHINE\SOFTWARE\Mi-`

Access mode	Target	Explanation
		`crosoft\`), that may or may not be accessed (read or write access). You can use wildcards. Use case 6 Please note that this access mode is only available for Windows 7 and higher!

5.1.2 Information on the Action tab

On this tab you determine how application control will respond to the entries on the **Filter** tab.

Please do the following:

1. Select the appropriate action:

Block	⌄
Do not block	
Block	
Ask user	
Modify reporting	

    - **Do not block**: Select this option if you do not require any further action. This setting corresponds to 'Allow'.

    - **Block**: Choose Block if you want to prevent specific events depending on the access mode or the target. For example, this action prevents an application or script from running, or a DLL from loading. This is the default setting.

    - **Ask user**: To let users decide which action they want to allow, select this option. Then, for example, it is up to the user to decide whether a Powershell script is run or not.

    > Note: Rule evaluation is stopped for these options (Do not block, Block and Ask user).

    - **Modify reporting**: No further action is taken with this option, it only changes the reporting. Further below you can indicate whether the command line will be

displayed in the event. Note that with this option the evaluation of the rules continues.

> 🖉 Note: Please note that these actions provide additional protection for particularly vulnerable processes. 'Do not block' can still be blocked by a setting in a white or black list, but 'Block' overwrites the setting in a whitelist rule!

2. Specify one of the following mechanisms that applies to targets other than the ones defined on the **Filter** tab:
   - **Block access to other targets**
     Allow access only to the targets that are explicitly allowed, and block all other targets.

   - **Block access by other applications**
     Only applications with explicit permission are allowed access, all other applications are blocked.
     Example: No other application may access the banking directory other than the banking software from use case no. 4.

3. Determine which events will be generated:
   The **Generate audit events when access is denied** is the default option. You can additionally or alternatively select the **Generate audit events when access is allowed** option. Use this option, for example, if you want to allow execution of specific scripts in a rule and want to generate the associated events. All events are displayed in the DriveLock Control Center (DCC) or DriveLock Operations Center (DOC). Both options are also suitable for the simulation mode.

> 🖉 Note: Please note that a large number of events will be created if you select both options.

4. The option **Include command line in event** determines that the corresponding event, which reports the process start (allowed or blocked), may also display command line parameters in the **EDR** section under **Application control**. The option is disabled by default.

> 🖉 Note: Please note that the command line may contain confidential data, such as passwords.

5.1.3 Information on the Messages tab

To find the default message texts for application control, that are displayed on the DriveLock Agent, open the **Global Settings** node, then select **Multilingual notification messages**, then **Languages / Standard messages** and open the **Applications** tab. Refer to the Administration Guide on DriveLock Online Help for more information about creating custom notification messages.

1. There is only one option on this tab available for **application behavior rules**, and it is enabled by default:

   - **Display message when access is denied**: Select a default text from the drop-down list or define your own text to be shown to the user when access is blocked.

   - Depending on the access mode, the following wildcards are allowed:
     - Access mode Execute:
       %EXE% for the name of the application; %PARENT% for the name of the program that starts the application.

     - All other access modes:
       %EXE% for the name of the application; %TARGET% for the access target.

2. There are three options for **application rules**:

   - **Display custom message in user notification**: Again, you can select a standard text or define your own text.

   - Check **Display no message when this rule is activated** if the user does not need to know when an application is blocked (by a blacklist).

   - By default, events are generated when applications are blocked. If these events are not required, check **Do not generate audit events when this rule is activated** .

5.1.4 General settings for rules

The following tabs appear in various application and behavior rules.

1. **Logged on users** tab:
   By default, the rule is active for all logged on users and groups.

2. **Computers** tab:
   - Select the computers the rule applies to.

   - For example, you can create a behavior rule only for a special group of computers that contains computers with a newer version of the DriveLock Agent.

3. **Messages** tab:

   For more information about the options on this tab for application rules or application behavior rules, click here.

4. **Networks** tab:

   Determine the network connections the rule applies to.

5. **Time limits** tab:
   - If you want the rule to apply only for a specific period of time, you can specify an individual time frame here (e.g. only on weekdays from 09:00 to 17:00)
   - It is also possible to specify a date for the start and end of the validity period.
   - Highlight the required period by either activating a single field or by clicking on a weekday on the left or a time at the top. In addition, check either **Rule active** or **Rule not active** for the times you selected.

6. **Permissions** tab:
   - Determine the users or groups the rule is active for.
   - Check **Selected users and groups** to activate the rule for a specific group of users only. To include another group or user in the list, click Add. Click Remove to delete the previously selected entry.

## 5.2 Generate application behavior rules from behavior recording

Whenever applications require access that is not apparent to the user (writing temporary files, creating configuration files or caches, etc.), DriveLock records these background actions and allows you to control them.

To have application behavior rules generated automatically from the result of the behavior recording, proceed as follows:

1. In the context menu of the **application behavior rules** under All Tasks, click the menu item **Generate behavior rules from the application behavior recording...** .
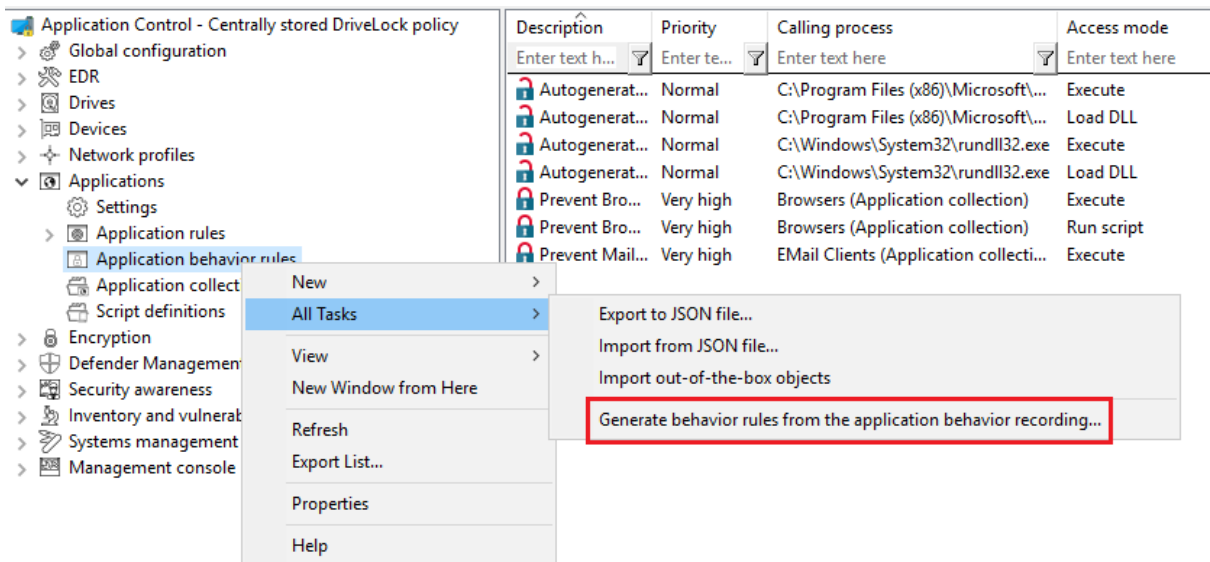
2. Select the data source for the recording results in the following dialog. This information can be obtained from the DriveLock Agent on the local or remote computer or from a pre-existing results file.
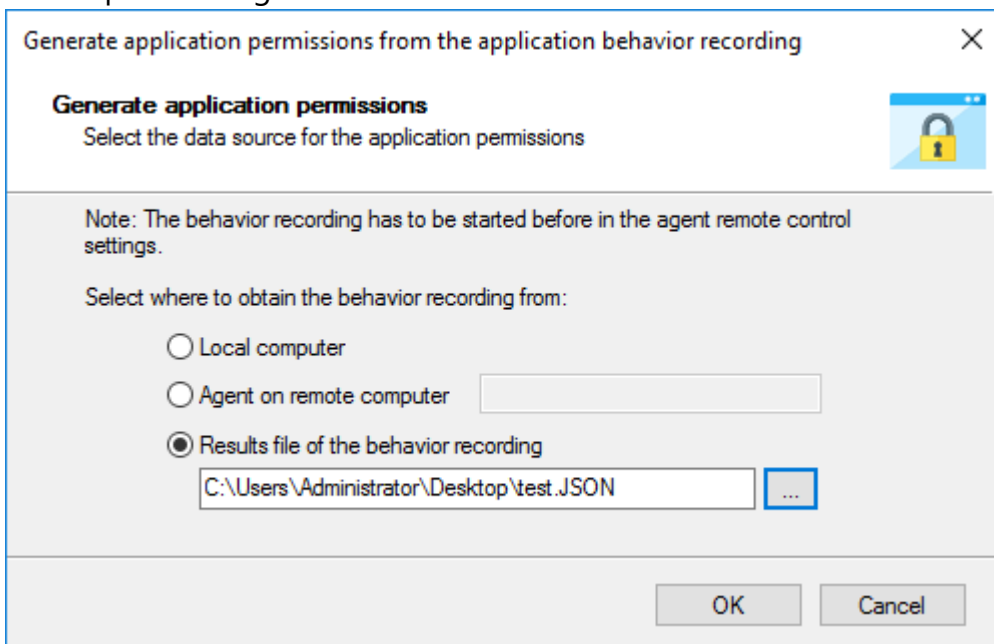

3. In the next dialog you configure the following:

- Select an application (or multiple applications) and specify whether to use the entire path or only the file regardless of where it is stored. For example, for browsers we recommend that you use the name without the path.

- Specify the access modes you want to create rules for and whether or not to combine multiple files using wildcards. **Never** is recommended for the **Execute** access mode, because it involves only a limited number of files (and rules to be created from them) that do not require combining. With **Write file**, on the other

hand, we recommend that you **always** use wildcards (even for small numbers) rather than having rules created for each individual file that is written.
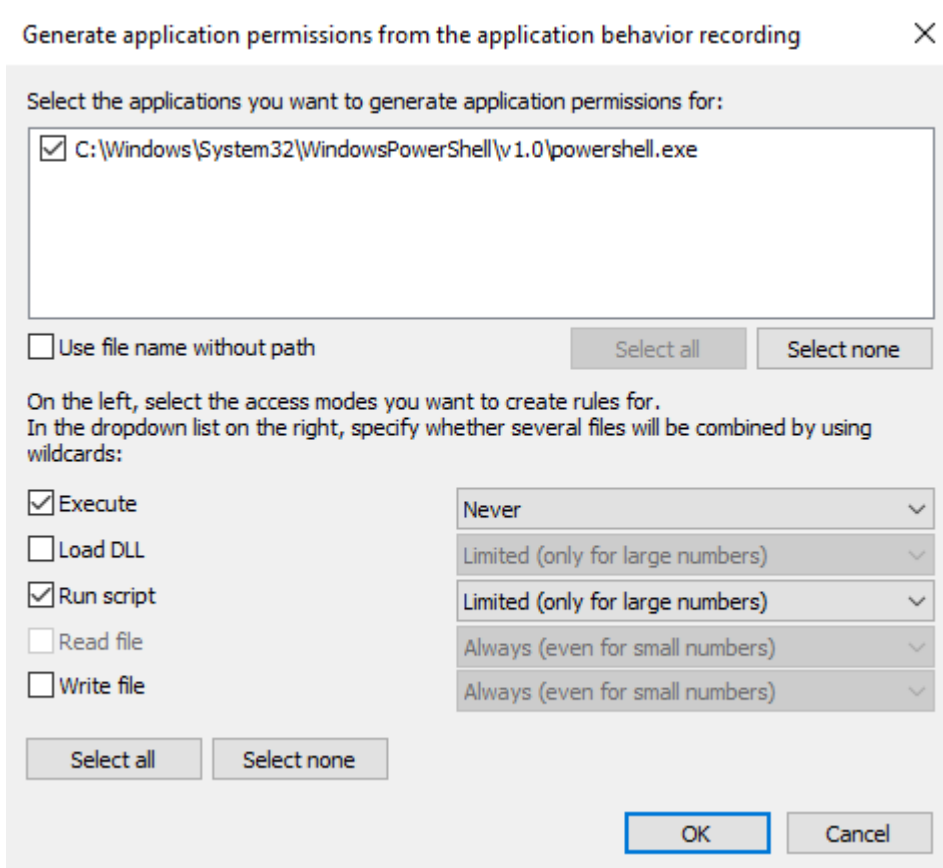
4. In the next step, the rules generated automatically are displayed as **Autogenerated rule** in the node **Application behavior rules**. On the **Action** tab you can see that the accessing application is allowed (Do not block), all other accesses are blocked.
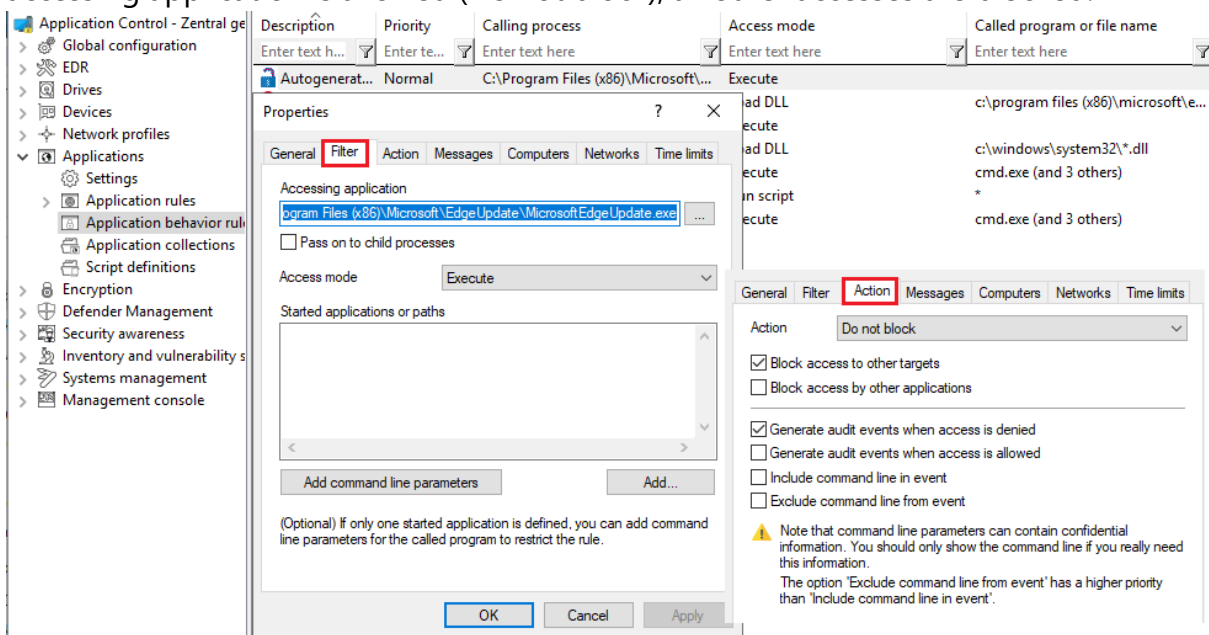
Tip: Create a separate folder for these application behavior rules so that they can be easily distinguished from the existing ones.

Summary: Creating application behavior rules automatically provides a much leaner and clearer set of rules and reduces the time spent on monitoring or analyzing events.

6 Application collections

Application collections are a set of applications that belong together in terms of subject matter or program. You can use them in the corresponding application behavior rules or application rules.

Rather than creating individual rules for each application, you can create a rule for multiple applications (on the application collection) at once. This reduces your set of rules and keeps it simple.

Example: Three application behavior rules should apply to three applications each:

- Rule no. 1 defines that no other applications are allowed to start from within a specific application.

- Rule no. 2 defines that applications are not allowed to write to a specific directory.

- Rule no. 3 defines that applications may only write text files to a specific directory.

> Note: By using lists, the number of rules can be reduced.

Create application collections based on the following example or use the provided application collections displayed in the taskpad view.

6.1 Application collection for Microsoft Office products

Scenario: You want to group different Microsoft Office products in an application collection to be able to use them in application behavior rules or application collection rules.
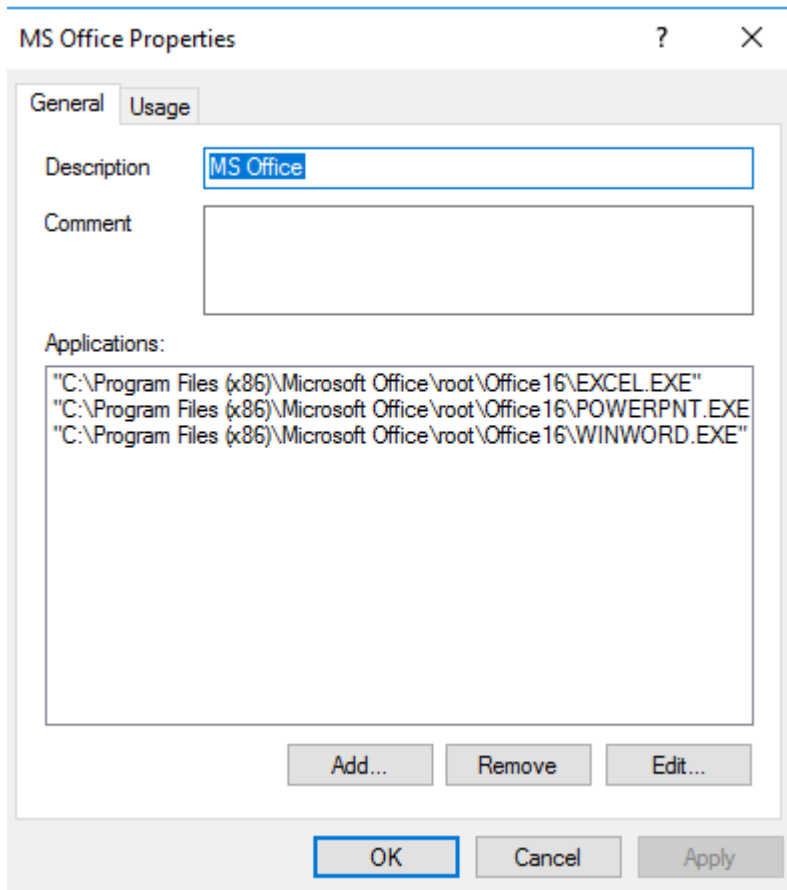
1. Select the **Application collections** sub-node and open the context menu.

2. Choose **New** and then **Application collection**.

3. Enter a unique description, here MS Office.

4. You can optionally enter a **comment**.

5. **Add** the paths to the applications you want to include. You can later remove applications or edit the paths.

6. Save your collection and use it now in application behavior rules.

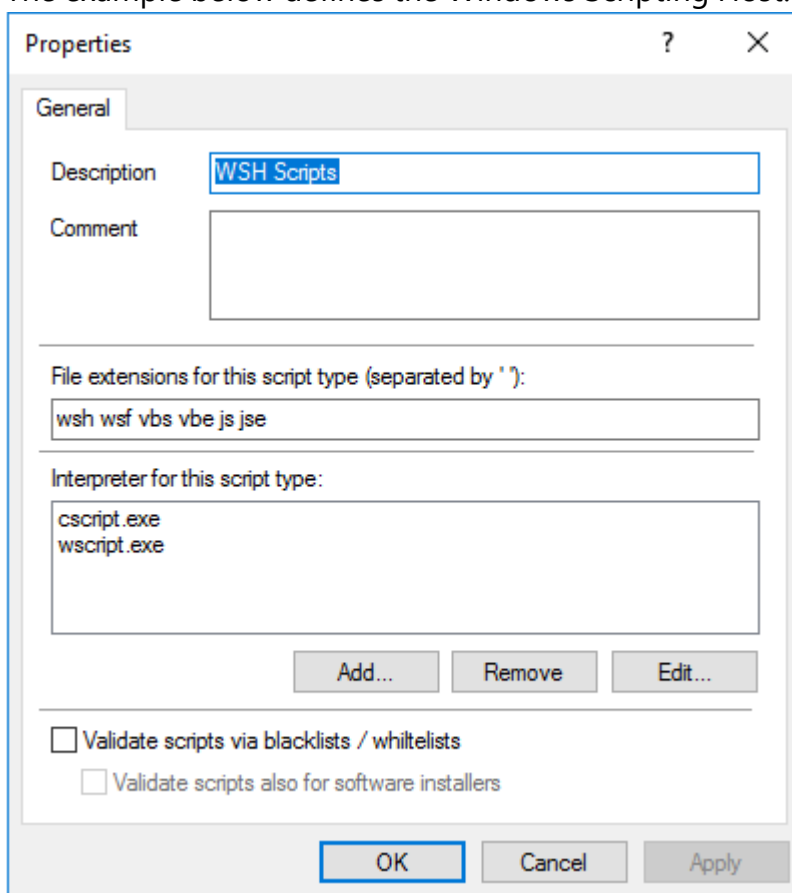The **Usage** tab displays the application rules where this collection is used.

7 Script definitions

To be able to use the Run script access mode with the application behavior rules, you must define the appropriate script types.

This definition tells application control which file accesses it should interpret as script execution.

Please do the following:

1. Open the context menu of **Script definitions**.

2. Click **New** and enter your definition in the following dialog.
   The example below defines the Windows Scripting Host.

3. Enter the extensions that apply to the script in the **File extensions for this script type** text box. Simply enter a space between the extensions.

4. Enter the interpreters that can interpret your script in the **Interpreter for this script type** text box.

5. With the **Validate scripts via blacklists / whitelists** option, you can specify to have scripts checked in blacklists or whitelists in the same way as DLLs or EXE files. For more information on blacklisting and whitelisting, see the corresponding chapters.

6. Select the **Validate scripts also for software installers** option if you want the validation to also apply to scripts started by software update processes.
Example: msiexec.exe is a trusted installer and may only be started if the corresponding MSI file is also trusted.
The Trusted process setting allows you to create a fixed list for such processes.

8 Use cases

8.1 Application behavior rules

8.1.1 Use case 1: Prevent PowerShell from starting

Scenario: You want to prevent Powershell from starting when a user launches a browser (here Internet Explorer), which could potentially install malware on the agent computers.

1. Start out with entering a description and a **Comment** if required on the **General** tab. As this is a rather general rule, enter a low **Priority** for it. Check **Enable rule** (default).

2. On the **Filter** tab, specify the following:
   - Enter the full path to the iexplore.exe in the **Accessing application** text box Alternatively, you could also use an application collection that contains different browsers.

   - Check **Pass to child processes** to prevent the browser from calling Power-shell.exe from the command line (cmd.exe) (this is a child process).

   - Since you want to prevent PowerShell from starting from Internet Explorer, specify Execute as **Access mode**.

   - Browse for a file or for a folder in the **Started applications or paths** text box, e.g. powershell.exe as file name in this example.

   > Note: We recommend specifying only the file name with blocking rules so that all instances can be included. When you specify the full path, please note that several program instances may exist, e.g. powershell.exe may be located in two different directories C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe or in C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe.

3. Specify the following on the **Action** tab:
   - The measure you want to use is to **block** the access.

4. For all other options, keep the default settings.

Conclusion: Every time the iexplore.exe is called and tries to start PowerShell, PowerShell will be blocked.

8.1.2 Use case 2: Restrict loading a DLL

Scenario: You want to specify that DLLs may only be loaded from certain directories.
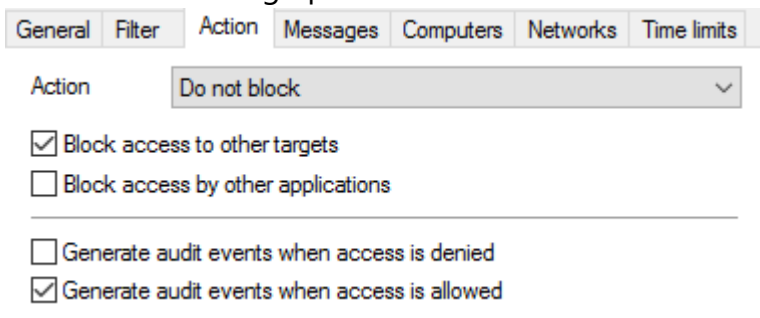
In this specific case, you want to prevent Windows Media Player from loading DLLs from network drives.

Proceed as shown in the figure:

1. Create an application permission where you define that the Windows Media Player application wmp.exe may only load DLLs from **\DLL4WMP\ .
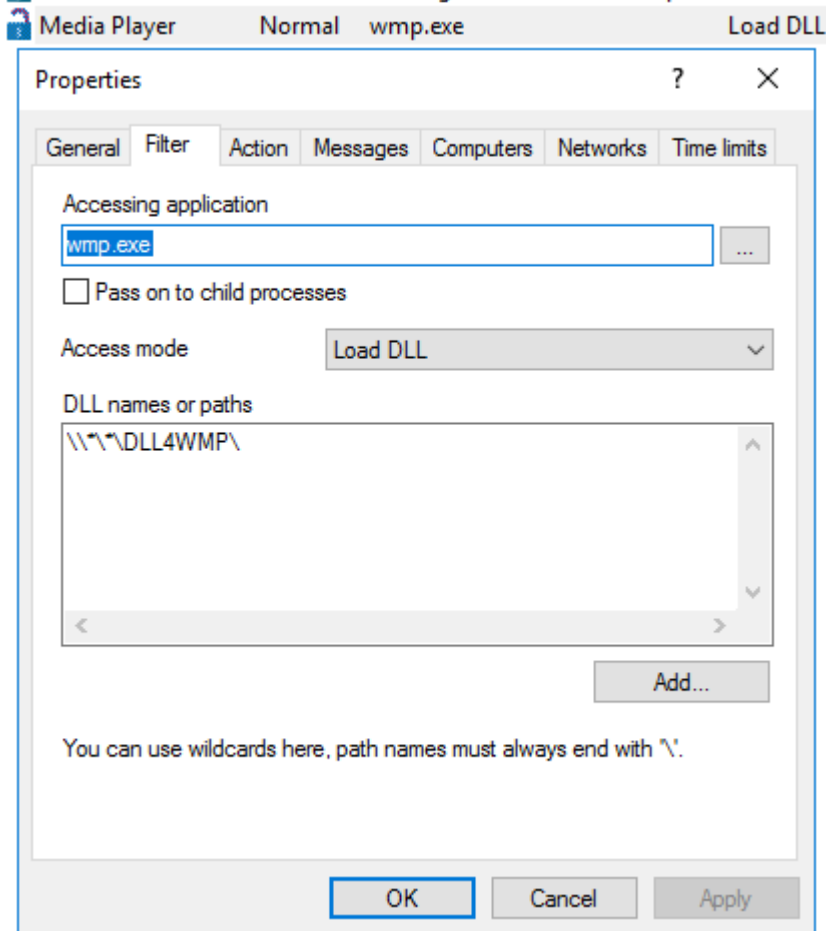
2. Select the following options on the **Action** tab:

- Select **Do not block** and check **Block access to other targets** to ensure that the DLL can only be loaded from the specified target.

- Select **Generate audit events when access is allowed**.

Note: Please note that rules with 'Do not block' (i.e. allow) have priority over 'Block'!

8.1.3 Use case 3: Run scripts

Scenario: You don't want browsers to run VB scripts (*.vbs).

Proceed as shown in the figure:

1. As **Accessing application**, select the application collection you created for your browser.

2. You can check the **Pass to child processes** option in this case. In this way it is possible to prevent the specified VB script from being started from a child process (e.g. from the command line).
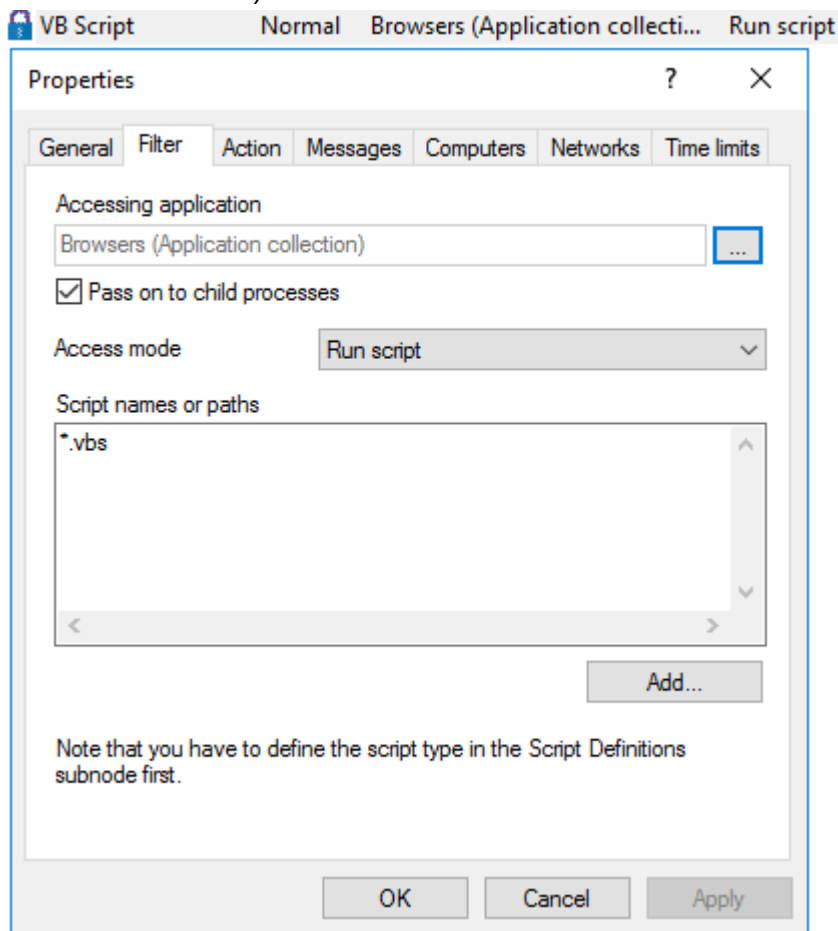
3. On the **Action** tab, select **Block** as the action.

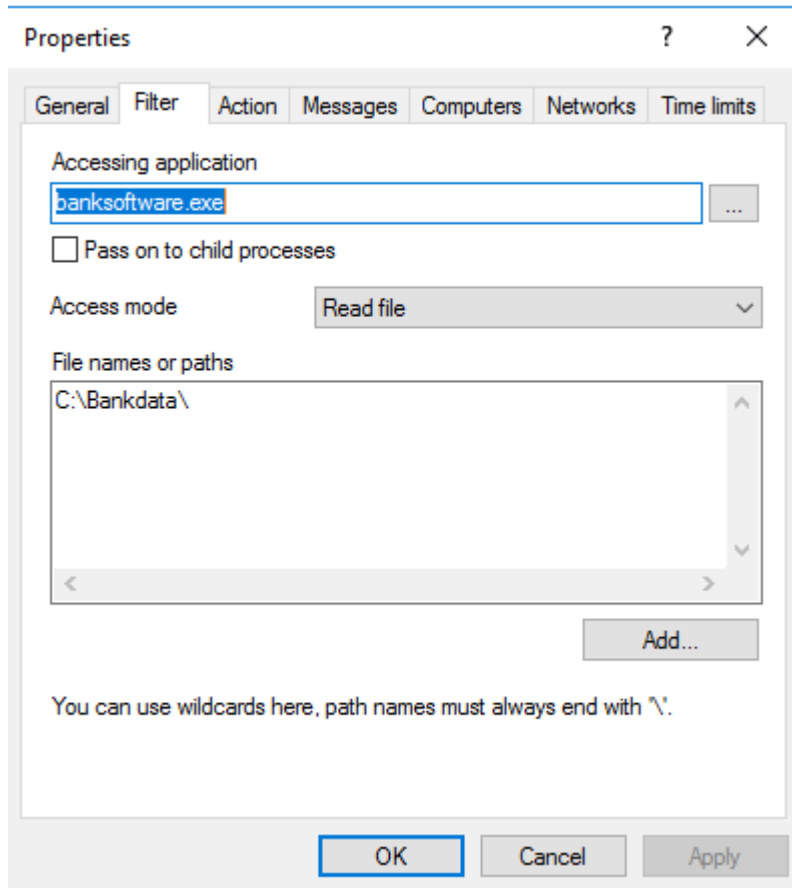4. For all other options, keep the default settings.

8.1.4 Use case 4: Read a specific directory

Scenario: You want to ensure that only your own banking software has read access to a specific directory. You do not want any other application to have read access to this directory. It

would be possible for malware to gain read access to this directory via a security vul-nerability in the browser and thereby read out your bank details. You need to prevent this from happening.
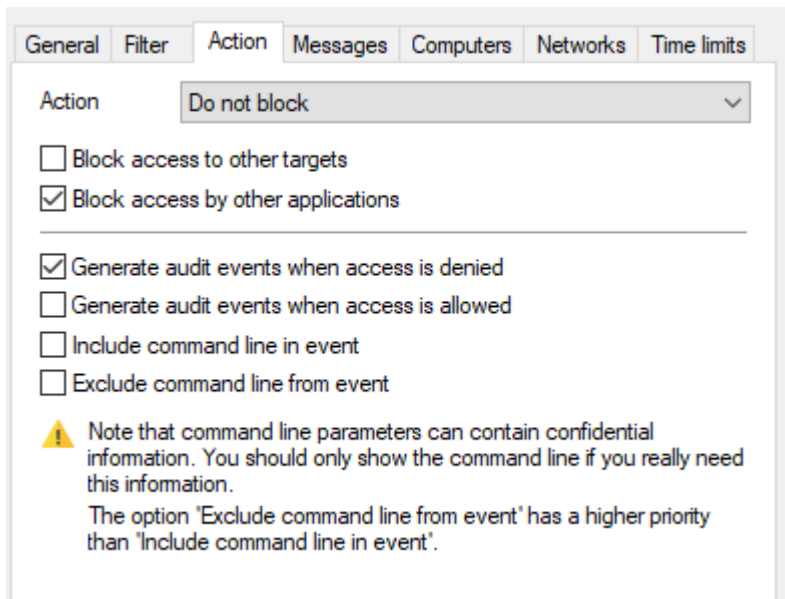
Proceed as shown in the figure:

1. Start out with entering a description and a **Comment** if required on the **General** tab.

2. On the **Filter** tab, enter Banksoftware.exe as **Accessing application**. As **Access mode** select **Read file** and under **File name** enter the path (in the example C:\Bankdata\).

3. Specify the following on the **Action** tab:
   - Select **Do not block** and check the **Block access by other applications** box to ensure that only your own banking software has read access to the specified tar-get.

- The **Generate audit events when access is denied** is the default option.

8.1.5 Use case 5: Write to a specific directory

Scenario: You want to specify that a particular browser (here it's Mozilla Firefox) is not allowed to write to the Documents folder. Since you want to specify this for all users and not just for some users, use a wildcard.

Proceed as shown in the figure:

1. Start out with entering a description and a **Comment** if required on the **General** tab.

2. On the **Filter** tab, enter the path to the browser as **Accessing application**.
   - To prevent the browser from being able to write to the directory via child processes anyway, check the option.

   - As **Access mode** select **Write file** and enter the path with wildcard (in the example C:\Users*\Documents\) in the **File name** text box.
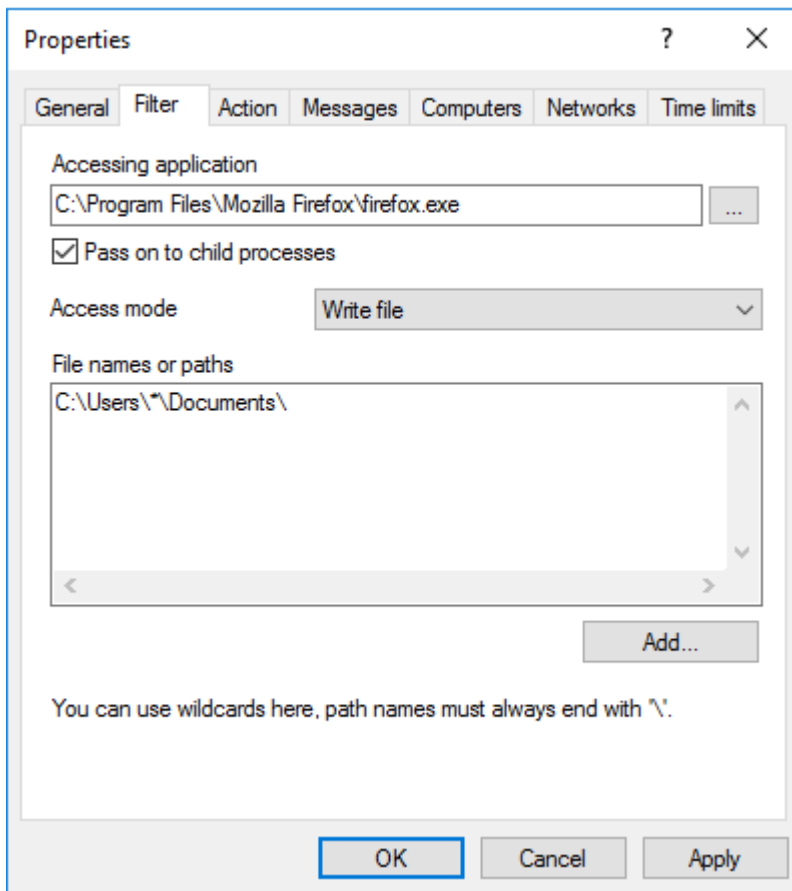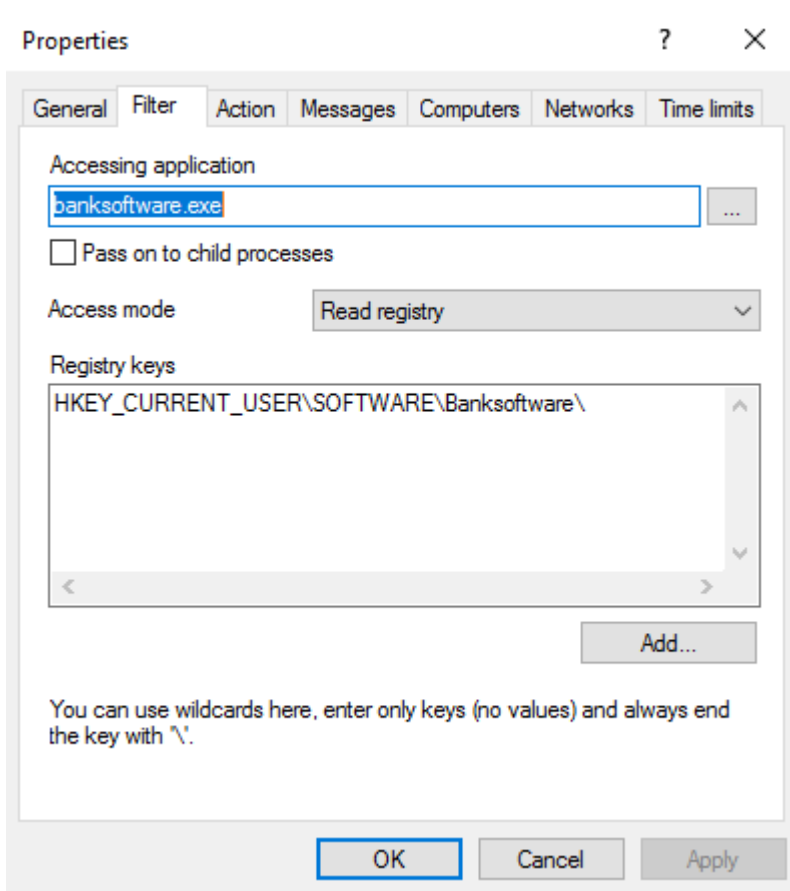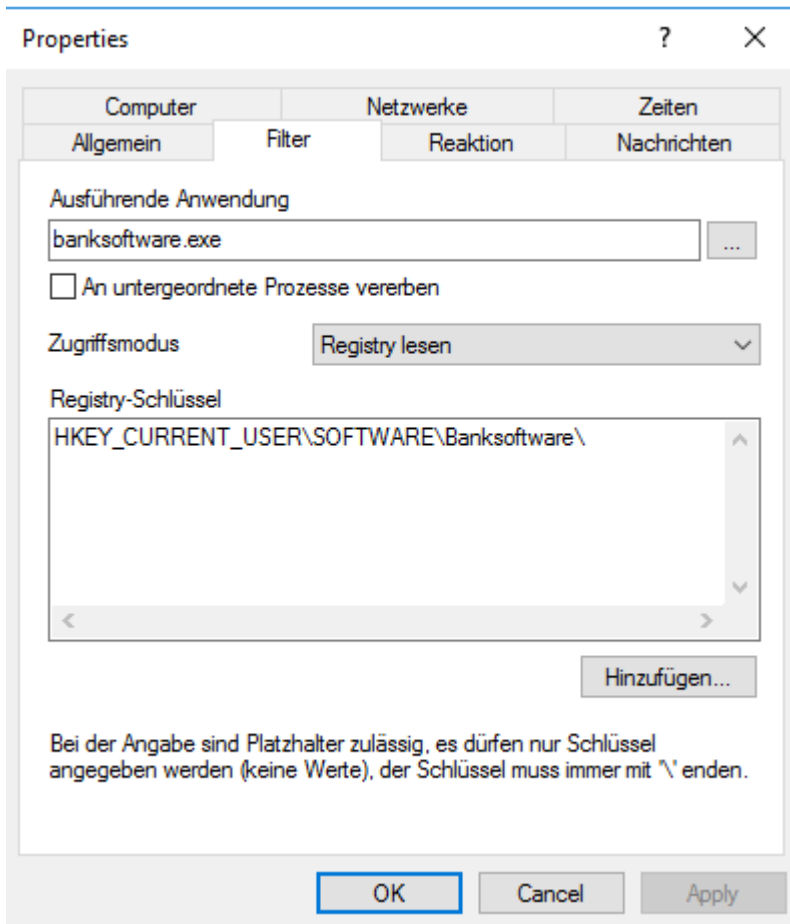
3. On the **Action** tab, select **Block**.

4. For all other options, keep the default settings.

8.1.6 Use Case 6: Restrict registry access

Scenario: You want to control registry access for your banking software from use case 4. Create two application permissions so that only the Banksoftware.exe is allowed to read the registry in the specified key.

Proceed as shown in the figure:

1. Start out with entering a description and a **Comment** if required on the **General** tab.

2. On the **Filter** tab, enter banksoftware.exe as **Accessing application**. As **Access mode** select **Read registry** and enter the key in the **Registry key** text box (in the example HKEY_CURRENT_USER\SOFTWARE\Bank Software\).

DriveLock

Properties [German dialog]

Computer	Netzwerke	Zeiten
Allgemein	Filter	Reaktion	Nachrichten

Ausführende Anwendung

`banksoftware.exe` [...]

☐ An untergeordnete Prozesse vererben

Zugriffsmodus   Registry lesen ▾

Registry-Schlüssel

```
HKEY_CURRENT_USER\SOFTWARE\Banksoftware\
```

Hinzufügen...

Bei der Angabe sind Platzhalter zulässig, es dürfen nur Schlüssel
angegeben werden (keine Werte), der Schlüssel muss immer mit '\' enden.

OK   Cancel   Apply

Properties [English dialog]

General	Filter	Action	Messages	Computers	Networks	Time limits

Accessing application

`banksoftware.exe` [...]

☐ Pass on to child processes

Access mode   Read registry ▾

Registry keys

```
HKEY_CURRENT_USER\SOFTWARE\Banksoftware\
```

Add...

You can use wildcards here, enter only keys (no values) and always end
the key with '\'.

OK   Cancel   Apply

3. Specify the following on the **Action** tab:

- Select **Do not block** and check the **Block access by other applications** box to ensure that only your own banking software has read access to the registry key.

- The **Generate audit events when access is denied** is the default option.
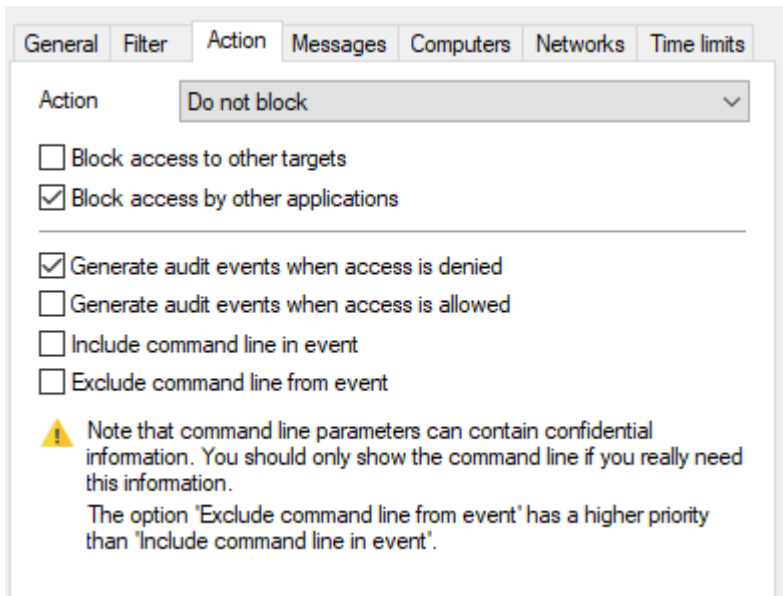
8.1.7 Use case 7: Detecting attacks with the example MITRE ATT&CK™ rules

DriveLock provides rules based on the MITRE ATT&CK framework. You can import these rules in the **EDR** node.

Some of these rules are stored in separate folders in the **Application behavior rules** node, see the figure below.

| | Note: The purpose of these rules is not to block or allow actions, but simply to report certain events on the particular computer, that are then processed by the event filters and alerts. |

8.2 Application rules

### 8.2.1 Use case 8: Show security awareness campaign when starting Outlook

Scenario: You want to display a security awareness campaign every time the user starts Outlook. Create a new file properties rule for this purpose.

Proceed as shown in the figure:

1. Specify the following on the **General** tab:

    - **Rule type**: Learning and Awareness

    - **Rule name**: Outlook

    - Choose the appropriate path. The other fields are filled in automatically.

    - Select the filters you want to create the rule by, and select the appropriate check-boxes.
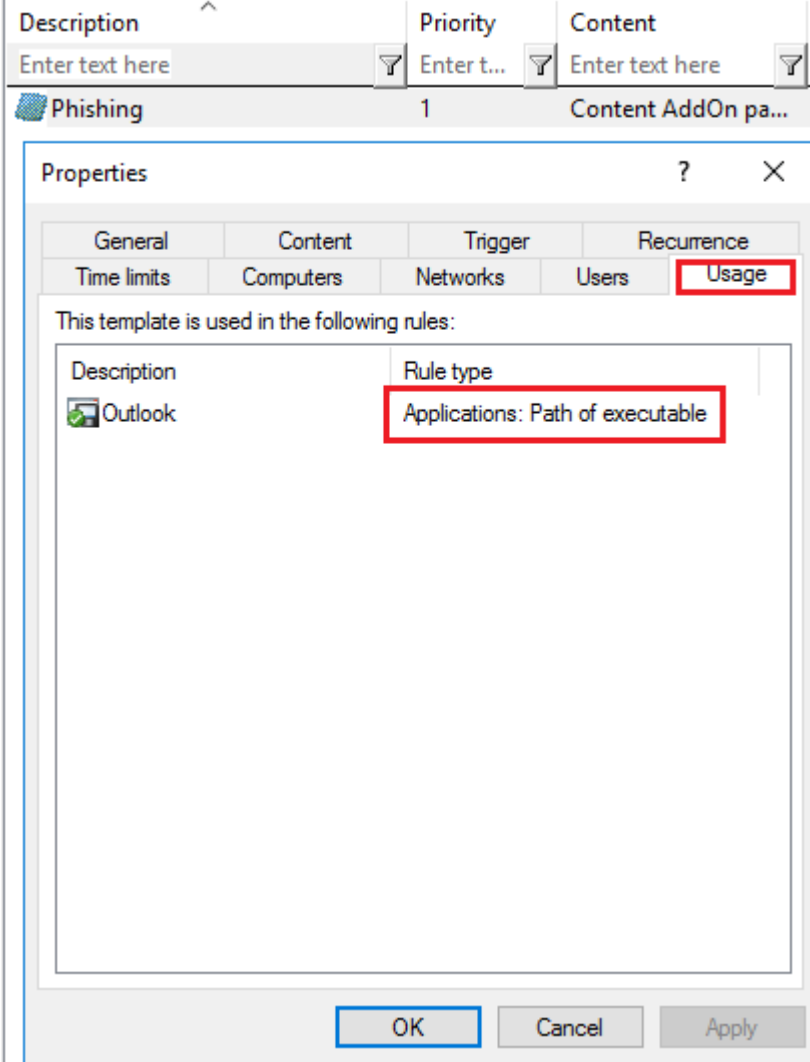
    - Add a **comment** if necessary.
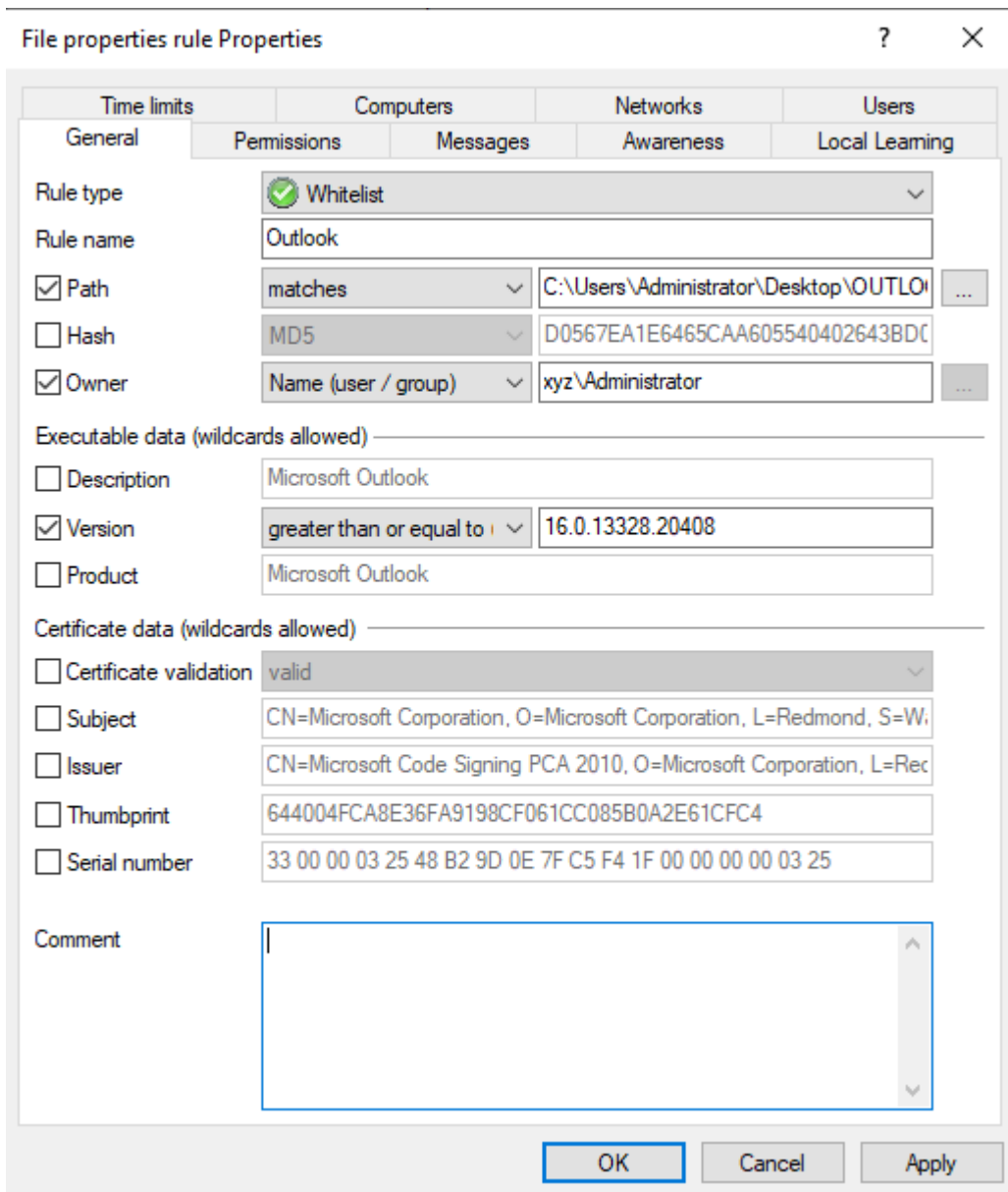
2. Open the **Awareness** tab.
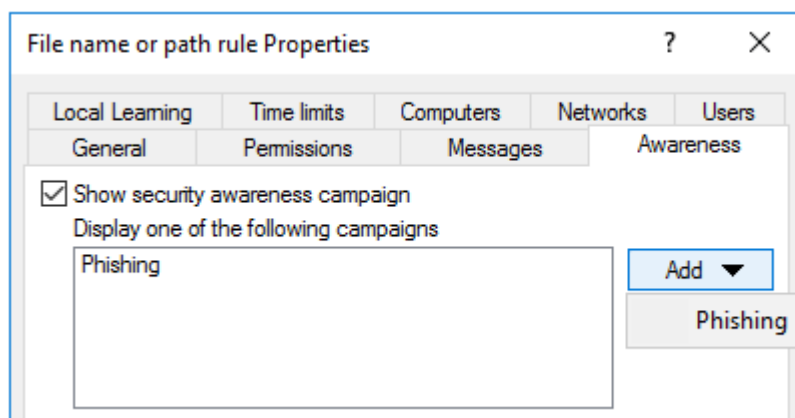
Select the campaign from the drop-down list under **Add**.

> ✐ Note: Make sure to set the **If used in rules** option on the **Trigger** tab in the properties dialog of the security awareness campaign.

For the **Phishing** campaign, the following information appears on the **Usage** tab:

> ✐ Note: For more information about creating security awareness campaigns, see the corresponding documentation on DriveLock Online Help.

3. For all other options, keep the default settings.

9 List of application control terms

Term	Explanation
Application collection	Grouping of several related applications in terms of subject matter or program. An application collection is used in application rules or in application behavior rules.
Application rules	Application rules can be used to allow or block individual applications, as well as configure local learning and the display of awareness campaigns.
Application behavior	Application behavior includes all actions an application executes, such as starting additional applications or DLLs or writing to specific directories.
Application Behavior Control	Monitoring the behavior of applications. DriveLock monitors and controls the activities of applications running on the agent.
Application behavior rules	Application behavior rules define the actions an application is allowed or not allowed to perform (for example launching other programs, loading DLLs, reading or writing files or the registry, executing scripts).
Blacklist	A negative list containing non-permissible and untrustworthy targets. By blacklisting it is possible to block specific applications.
Local learning	In the course of a learning phase, the DriveLock Agent learns what is allowed on the particular client computer: starting applications or DLLs, or performing actions such as writing to specific directories.
Local whitelist	The local whitelist is a hash database rule that is generated locally. It can be pre-filled with executables (allowed files) in certain directories and can be extended accordingly.
Simulation mode	During a simulation, DriveLock generates event messages

Term	Explanation
	for started or blocked applications based on configured rules, but the execution itself is neither allowed nor prevented.
Application behavior recording	Recording of application behavior on the DriveLock Agent; to be saved as a JSON file and to generate application behavior rules from it.
Whitelist	A positive list containing allowed and trusted targets. Only these may be executed.

Copyright