



DriveLock Linux Agent

Documentation 2022.1

DriveLock SE 2022




Table of Contents

| | |
|--|-----------|
| 1 DRIVELOCK LINUX AGENTS | 4 |
| 2 SYSTEM REQUIREMENTS | 5 |
| 2.1 Supported Linux distributions | 5 |
| 2.2 DriveLock configurations | 5 |
| 3 INSTALLING THE DRIVELOCK AGENT | 6 |
| 3.1 Installation instructions | 6 |
| 3.2 Installation parameters | 7 |
| 3.3 Installing the DriveLock Agent on IGEL clients | 7 |
| 3.3.1 Configuring the UMS server | 8 |
| 4 CONFIGURATION SETTINGS | 13 |
| 4.1 Recommended procedure | 13 |
| 4.2 Policy settings for DriveLock Linux Agents | 14 |
| 4.2.1 Global configuration | 15 |
| 4.2.2 EDR | 15 |
| 4.2.2.1 EDR: Event settings | 15 |
| 4.2.2.2 Event filter definitions | 16 |
| 4.2.2.2.1 Create event filter definitions | 17 |
| 4.2.3 Drives | 18 |
| 4.2.3.1 Drive settings | 18 |
| 4.2.3.2 Drive whitelist rules | 18 |
| 4.2.4 Devices | 20 |
| 4.2.4.1 Supported device classes for Linux agents | 20 |
| 4.2.4.2 Device settings | 20 |
| 4.2.4.2.1 Device whitelist rules (for USB controllers) | 22 |
| 4.2.4.2.2 Device whitelist rules (for devices) | 23 |
| 4.2.4.2.3 Android and Apple devices | 24 |

| | |
|---|-----------|
| 4.2.4.2.4 Devices collections | 25 |
| 4.2.4.2.4.1 Create device collections | 25 |
| 4.2.5 Applications | 26 |
| 4.2.5.1 Prerequisites for Application Control on Linux Agents | 26 |
| 4.2.5.2 Scanning and blocking mode | 27 |
| 4.2.5.3 File properties rule (for Linux) | 28 |
| 4.2.5.4 Special rule (for Linux) | 29 |
| 4.3 Agent remote control | 30 |
| 5 LINUX AGENTS IN THE DCC | 32 |
| 5.1 DCC: Help Desk actions | 32 |
| 6 LINUX AGENTS IN THE DOC | 34 |
| 7 LIST OF EVENTS | 35 |
| 8 COMMAND LINE TOOL | 47 |
| COPYRIGHT | 49 |

1 DriveLock Linux Agents

DriveLock supports assigning centrally stored policies to DriveLock agents running the Linux operating system.

The functionality of Linux support is currently limited to locking external devices and drives connected to Linux clients via a USB interface, plus some application control functions. This gives administrators control over the usage of devices, drives and applications, on DriveLock Linux agents as well, so that these client computers are reliably protected from malware attacks. In addition, the EDR functionality can be used to evaluate some DriveLock events and create corresponding event filter definitions.

2 System requirements

2.1 Supported Linux distributions

DriveLock supports the following 64-bit Linux distributions (as listed below and higher):

- CentOS Linux 8
- Debian 7
- Fedora 31
- IGEL OS starting with version 10
- Red Hat Enterprise Linux 5
- SUSE 15.1
- Ubuntu 18.04

2.2 DriveLock configurations

The following configuration requirements must be met to manage DriveLock Linux Agents in a DriveLock environment and control the use of their USB interfaces.

Full installation and configuration of a DriveLock Suite with

- DriveLock Management Console (DMC): Version 2019.2 and higher
- DriveLock Enterprise Service (DES): Version 2019.2 SP1 and higher
- DriveLock Linux-Agent (on Linux clients): Version 2019.2 SP1 and higher



Note: Please ensure that the same DriveLock version (or higher) is installed on the DES and on the DriveLock Agent.

3 Installing the DriveLock Agent

3.1 Installation instructions

Follow these steps to install the DriveLock Linux Agent on your Linux clients.


 Note: Please note that the installation is different for [IGEL clients](#).

1. Copy and extract the **drivelock.tgz** file on your Linux clients. It is included on the DriveLock ISO image.
2. The file contains the **drivelockd-install.sh** installation script . Run this script (see also [Installation parameters](#)).

 Warning: To run scripts on the Linux client, you must have administrator rights (see figure).

```
test@debian10:~$ sudo ./drivelockd-install.sh
[sudo] password for test:
Drivelock self extract installer
extracting archive...
install to path [suggest: '/opt/drivelock']:
drivelock server url [format: http(s)://<server>:<port>]: https://192.168.8.249:6067
drivelock tenant [default: root]: kav
installing drivelock linux agent to: '/opt/drivelock'
setting server to: 'https://192.168.8.249:6067'
setting tenant to: 'kav'
starting agent ...
```

3. Enter the following:
 - Installation path: The default is `/opt/drivelock`, but you can also specify a different path.
 - DES and port: Enter the server URL in the format `'https://<Server-Port>:<Port>'` here.
 - Tenant: The default is `'root'`, but you can also specify a different tenant (in the figure `kav`).
4. The DriveLock Service starts as soon as the DriveLock Linux Agent has been completely installed.
5. If you experience errors during installation, we recommend restarting the Linux client to ensure that all DriveLock messages are displayed in the Linux client's user interface.

 Note: The Linux client only displays messages when devices are connected or disconnected (as popups), the DriveLock Agent does not have its own user interface here.

3.2 Installation parameters

To install the DriveLock Linux Agent on your Linux clients, you can optionally use installation parameters. To display the individual parameters, open the installation script with the parameter `-h` (see figure).

```

test@debian10:~$ sudo ./drivelockd-install.sh -h
Drivelock self extract installer
extracting archive...
usage: ./drivelockd-install.sh [options]

options:
-h|--help                print this help message
-c|--custom-part         create a custom partition package
-i|--install <PATH>     install into path
-s|--server <SRV>       server
-t|--tenant <TENANT>    tenant
test@debian10:~$ sudo ./drivelockd-install.sh -t kav -s https://192.168.8.207:6067

```

You can specify the following installation parameters:

- `-h`: Displays help for the installation parameters
- `-c`: This parameter only applies to IGEL clients. Here you enter the Custom Partition Package you want to use.
- `-i`: Enter the path to the DriveLock installation directory. The default is the current working directory, but you can also specify a different path.
- `-s`: Enter the server in the format 'https://<server>:<port>' here. See figure above.
- `-t`: Enter the tenant, the default is 'root'.

3.3 Installing the DriveLock Agent on IGEL clients

Follow these steps to install the DriveLock Linux Agent on your IGEL clients.

1. Copy and extract the **tar -xzf drivelock.tgz** file on your Linux clients. It is included on the DriveLock ISO image.
2. The tar file contains the **drivelockd-install.sh** installation script. Run this script with the parameter `-c` (see figure).

```

test@testub:~/igel_custom_partition$ ./drivelockd-install.sh -c
Drivelock self extract installer
extracting archive...
install to path [suggest: '/home/test/igel_custom_partition']:
drivelock server url [format: http(s)://<server>:<port>]: https://192.168.8.207:6067
drivelock tenant [default: root]:
installing drivelock linux agent to: '/home/test/igel_custom_partition'
setting server to: 'https://192.168.8.207:6067'
setting tenant to: 'root'
path to save custom partition package [default: '/home/test/igel_custom_partition']:
custom partition package name [default: 'drivelock']:

```

See [Installation parameters](#) for more information.

3. Enter the following:

- Installation path: The default is the current working directory, but you can also specify a different path (in the figure `/home/test/igel_custom_partition`).
- DES and port: Enter the server URL in the format `'https://<Server>:<Port>'` here.
- Tenant: The default is `root`, but you can also specify a different tenant.
- Path and name for the user-defined IGEL OS partition files. By default, these files are created in the current working directory.



Note: You do not need root rights for this process.

4. Once the script is finished, the IGEL OS partition files `drivelock.inf` und `drivelock.tar.bz2` are generated and located in the path specified in the above step.

```

test@testub:~/igel_custom_partition$ ls -al
total 42224
drwxr-xr-x  3 test test    4096 Feb 19 10:02 .
drwxr-xr-x 15 test test    4096 Feb 19 10:00 ..
drwxr-xr-x  2 test test    4096 Feb 14 16:45 bin
-rwxr-xr-x  1 test test   1032 Feb  4 18:09 dl_getinfo
-rw-r--r--  1 test test  36864 Feb 19 10:02 DLSettings.db3
-rw-r--r--  1 test test  36864 Feb 19 10:02 DLSettings.db3-ini
-rwxr-xr-x  1 test test   3723 Feb  4 18:09 drivelock-ctl
-rwxr-xr-x  1 test test 14694959 Feb 14 16:45 drivelockd-install.sh
-rwxr-xr-x  1 test test    213 Jan  7 13:55 drivelockd.service
-rw-r--r--  1 test test    72 Feb 19 10:02 drivelock.inf
-rw-r--r--  1 test test 13974612 Feb 19 10:02 drivelock.tar.bz2
-rwxr-xr-x  1 test test 14451584 Feb 19 10:01 drivelock.tgz
-rwxr-xr-x  1 test test    127 Jan  7 13:55 run

```

5. Next, configure the [UMS server](#).

3.3.1 Configuring the UMS server

Please do the following:

1. Upload the **drivelock.inf** and **drivelock.tar.bz2** files to the UMS server.
2. Open the UMS Console.
3. In the UMS Console, navigate to **Files** -> **New File** -> **Upload local file to UMS server**.
4. Set **Root** as **Owner** (see figure).

Edit file

Source URL:

Classification:

Devices file location:

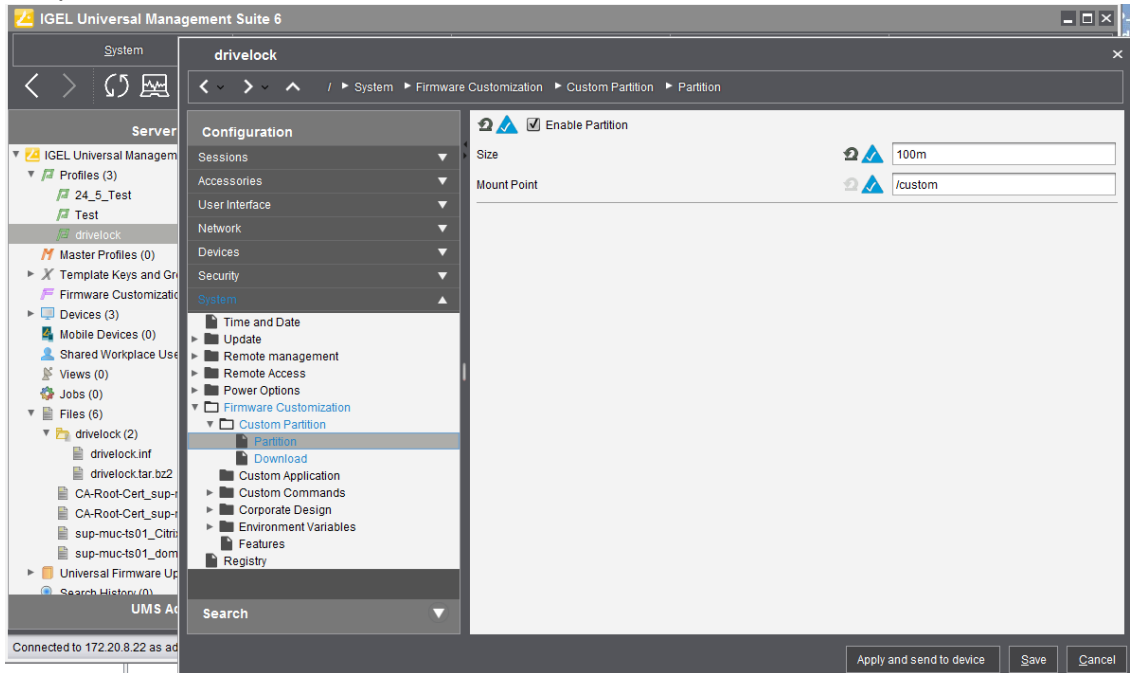
Access rights

| | Read | Write | Execute |
|--------|-------------------------------------|-------------------------------------|-------------------------------------|
| Owner | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Others | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Owner:

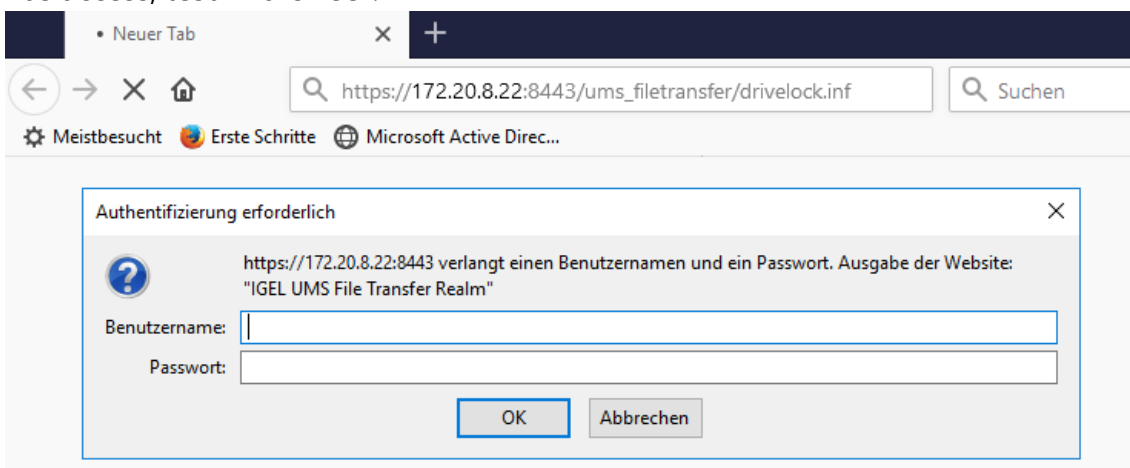
5. Repeat the same for the **drivelock.tar.bz2** file.
6. In the UMS system, create a new profile, e.g. drivelock.
7. In the UMS Console, navigate to **Profiles** -> **New Profiles** -> **Profile Name**.
8. Edit the created profile and activate the Custom Partition as follows (see figure):
 1. Navigate to **System** -> **Firmware Customization** -> **Custom Partition** -> **Partition**
 2. Unlock **Enable Partition**
 3. Check **Enable Partition**
 4. Set size of the partition to 150 or 200 MB

5. Keep /custom as **Mount Point**.



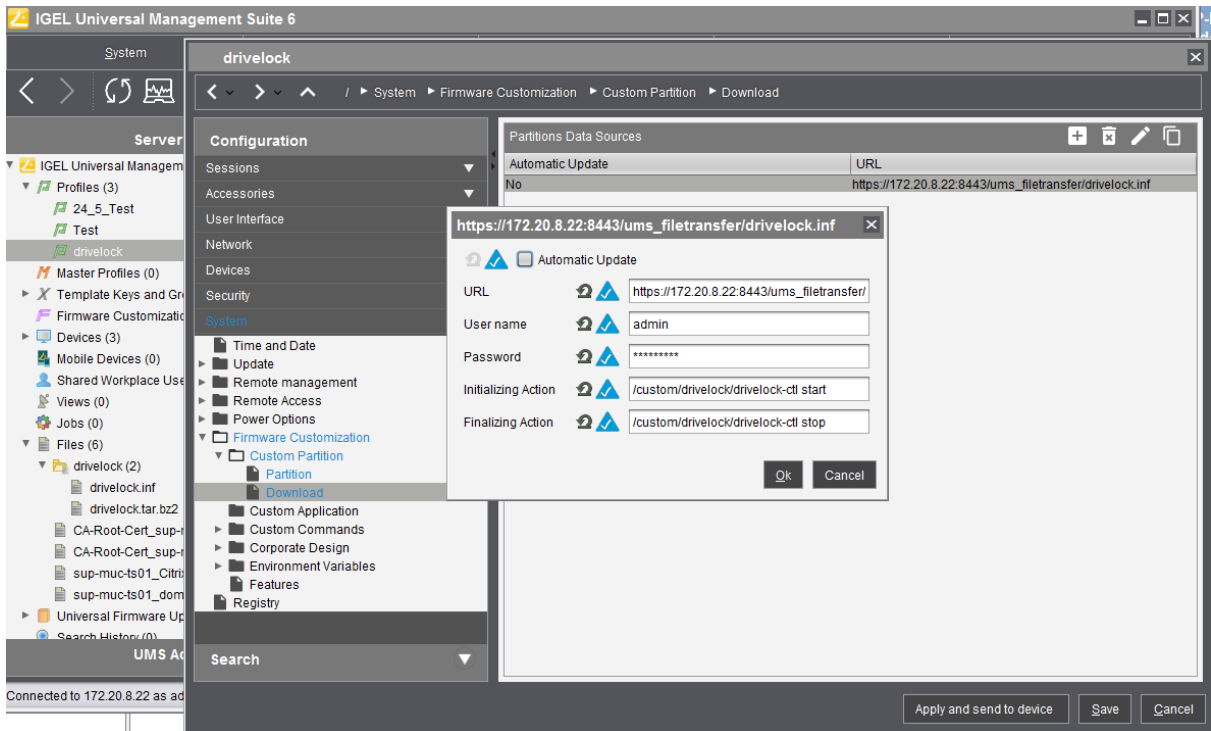
9. Specify the download source.


1. Navigate to **System -> Firmware Customization -> Custom Partition -> Download**
2. Click [+] to add a **Partition Download Source**.
3. Add the download URL **http(s)://<server>:8443/ums_file-transfer/drivelock.inf**
4. Enter the **user name** and **password** to download the file. To confirm the user has access, test in browser.



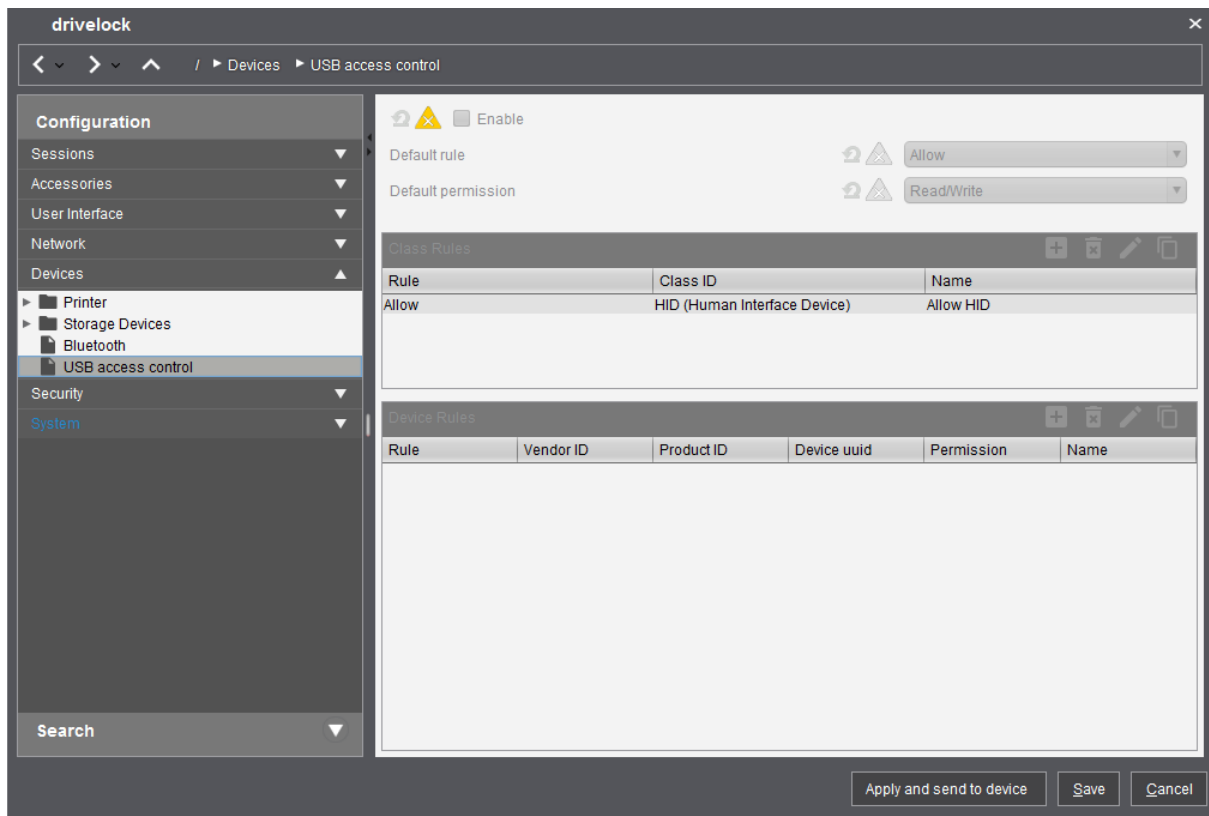
10. In the next step, enter the following (see figure):

- Set **Initializing Action** to /custom/drivelock/drivelock-ctl start.
- Set **Finalizing Action** to /custom/drivelock/drivelock-ctl stop.



 Note: Please note that the Mount Point matches the mount point configured in step 8.

11. Disable **USB access control** on Thin Clients.
 Navigate to **Devices** -> **USB access control** -> uncheck **Enable**.



12. Assign the DriveLock profile to the Thin Clients.

1. Navigate to **Devices** -> **Client**. Drag and drop the DriveLock profile icon to the Thin Client.
2. As per requirement, select **Now** or **By next reboot** to activate the changes.

4 Configuration settings

4.1 Recommended procedure

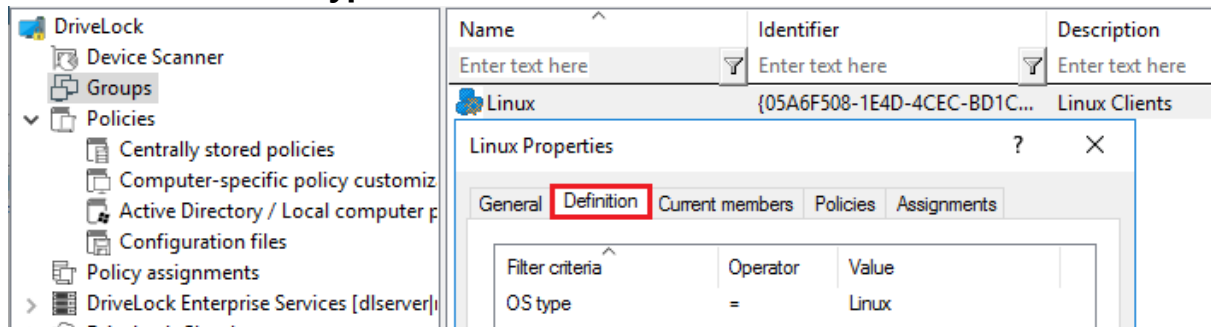
To configure the DriveLock Linux Agent, we recommend following the procedure below:

1. Start by creating a DriveLock group (static or dynamic) that includes your Linux agents.

This makes it easier to assign the policy you configure for your Linux agents later.

Select the filter criteria **OS type Linux** as group definition.

The figure below shows the dynamic **Linux** group with description **All Linux clients** and filter criterion **OS type = Linux**.

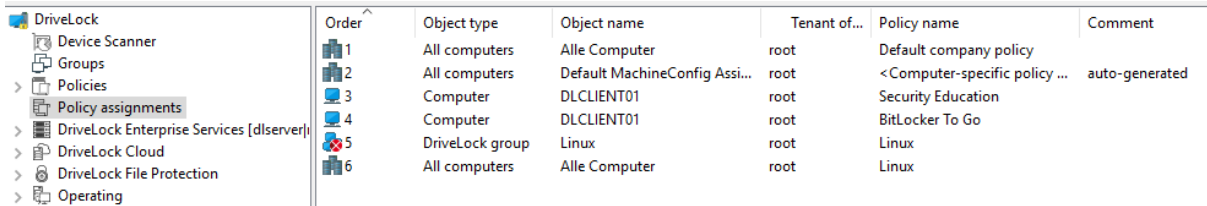


For more information on DriveLock groups, see the Administration Guide at [DriveLock Online Help](#).

2. To use a different tenant for your DriveLock Linux agents, select another one. For more information on using tenants, please also refer to the Administration Guide.
3. Create a new centrally stored policy for your Linux clients, name it accordingly (e.g. 'Linux policy') and start with [Global settings](#).
4. Depending on whether you want to control the use of [devices](#), [drives](#) or [applications](#), set the appropriate settings.

- Assign the 'Linux policy' to your DriveLock group. You can also assign to All Computers if you do not want to use a group.

The figure below shows the 'Linux' policy assigned to the Linux group and to All Computers.



| Order | Object type | Object name | Tenant of... | Policy name | Comment |
|-------|-----------------|-------------------------------|--------------|-------------------------------|----------------|
| 1 | All computers | Alle Computer | root | Default company policy | |
| 2 | All computers | Default MachineConfig Assi... | root | <Computer-specific policy ... | auto-generated |
| 3 | Computer | DLCLIENT01 | root | Security Education | |
| 4 | Computer | DLCLIENT01 | root | BitLocker To Go | |
| 5 | DriveLock group | Linux | root | Linux | |
| 6 | All computers | Alle Computer | root | Linux | |

4.2 Policy settings for DriveLock Linux Agents

Use the following settings to configure the policies you want to assign to DriveLock Linux Agents:

- **Global configuration:** Settings, Server connections, Trusted certificates
- **EDR:** Events (General Agent events, Device and Drive events), Event filter definitions
- **Drives:** Removable drive locking, Drive whitelist rules
- **Devices:** Device class locking, Device whitelist rules, Device collections
- **Applications:** Scan and Block mode setting, Special rule and File properties rule (both for Linux)

 Warning: Please note that the settings for drives and devices for DriveLock Linux agents are limited to controlling the USB interface.

The configuration of your 'Linux policy' depends on the specific requirements for your DriveLock Linux Agents.

Here are two scenarios for device settings (applicable to all users of the Linux clients):

- You want to allow the usage of Human Interface Devices, e.g. keyboards, but want to lock specific keyboards: create a device rule where you only list the devices you want to lock (blacklist mode).
- You want to block the usage of USB drives, e.g. USB flash drives, but want to allow specific USB flash drives: create a drive rule where you specify the allowed USB flash drives (whitelist mode).



Warning: The [device and drive classes](#) in Windows and Linux do not always match. DriveLock currently uses the hardware ID of the device or drive that will be locked (or allowed) on the DriveLock Linux Agent as match criteria.

4.2.1 Global configuration

1. Open the **Settings** section to configure the following:
 - **Remote control settings and permissions:** On the **Permissions** tab you can add the users that are allowed to take action on the Linux agent, such as changing the configuration.
 - **Event message transfer settings:** Make sure to check the **Enable event forwarding to the DriveLock Enterprise Service** option on the **Server** tab. The second option, **Report agent status to server**, allows you to specify the intervals for sending agent alive messages to the DES.
 - **Advanced DriveLock Agent settings:** On the **Intervals** tab you can set the intervals for loading the configuration from the server.
2. In the **Server connections** section you can add a new server, if required.
3. In the **Trusted certificates** section you select the certificates for the secure communication between the DriveLock Management Console and/or the DriveLock Linux Agents and the DES. Please refer to the DriveLock Administration Guide at [drive-lock.help](#) for further information on certificates.

4.2.2 EDR

With EDR (Event Detection & Response), you get an optimized display of individual events combined with filter options.

For DriveLock Linux agents, the following event categories are important: **Application control**, **General agent events** and **Device** and **Drive** events. See [Events](#) for a detailed list.

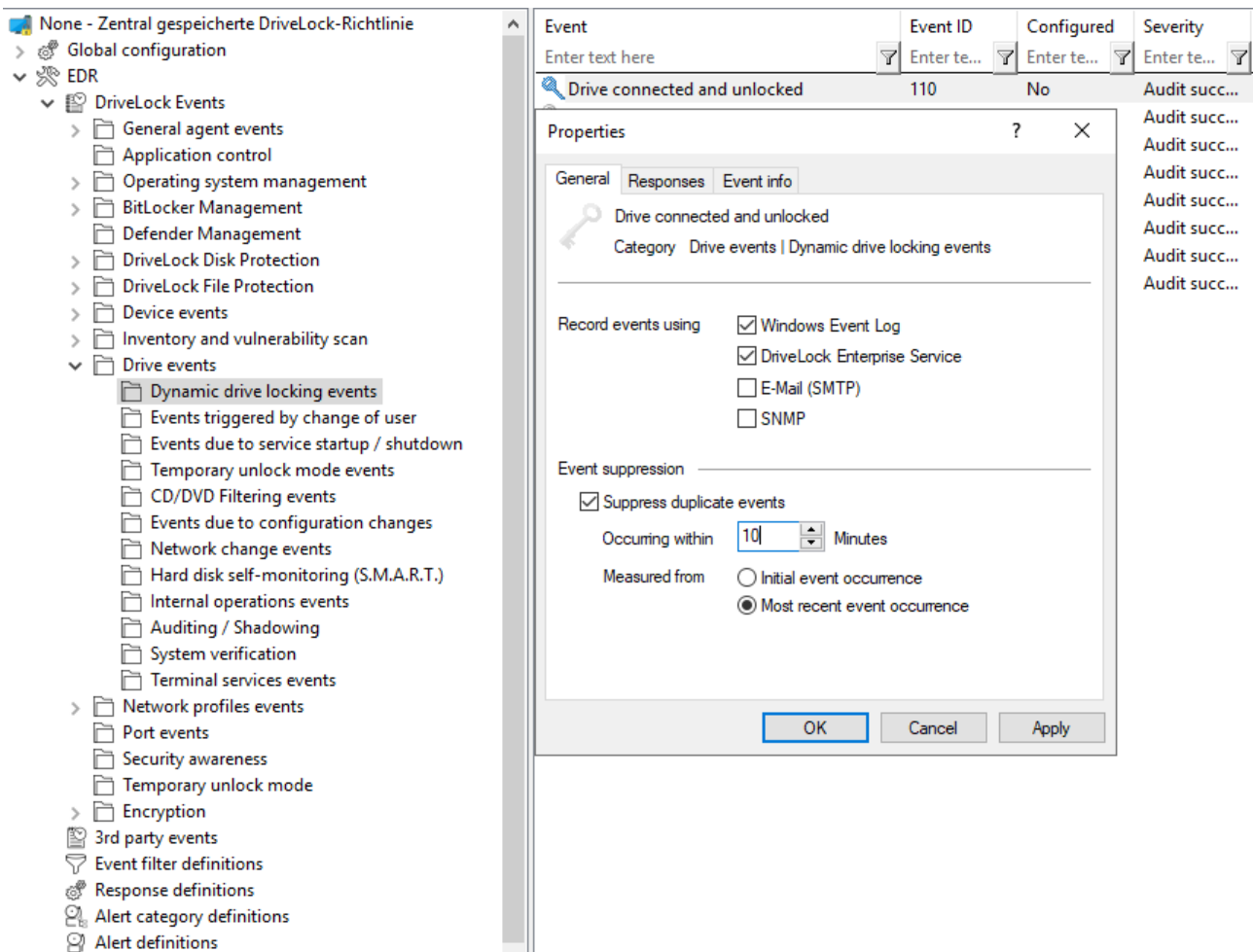
You can log events in the Windows Event Log or on the DriveLock Enterprise Service, but not in SNMP or SMTP.

The following [settings](#) are currently available for Linux agents.

4.2.2.1 EDR: Event settings

Example of how to configure drive event 110, which indicates that a drive is connected to the DriveLock Linux Agent and that it is not locked.

1. In the **EDR** node, open the **Events** subnode. Doubleclick the event in the **Drive events** section. Currently only the settings on the **General** tab are available for Linux agents (see figure).
2. The System Event Log (**Windows Event Log**) option is the default, but you can also select **DriveLock Enterprise Service** to save the events in the event log on the DES.
3. If required, you can also check the **Suppress duplicate events** option.



4.2.2.2 Event filter definitions

On Linux agents it is possible to apply event filter definitions to the events available for Linux.

You can filter

- by filter criteria,
- by computers (with computer names or Drivelock groups)
- and by times.

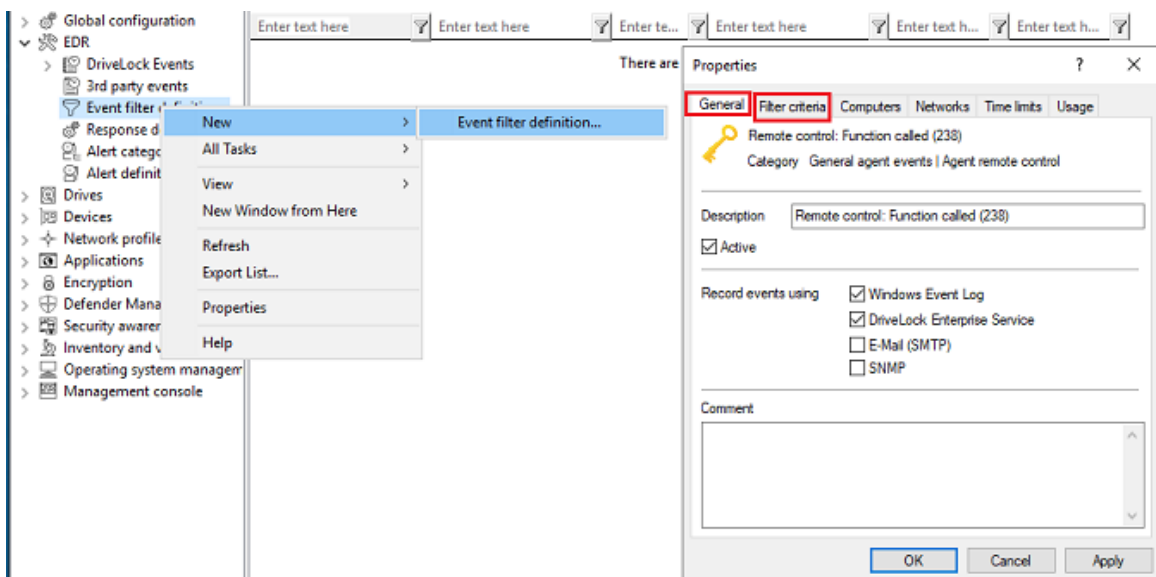
Event filter definitions can be used to reduce the number of events in the DOC event view, making it easier to find relevant events.

4.2.2.2.1 Create event filter definitions

Example: Event 238 (remote control access) - generates a large number of events during a session. To reduce the number and restrict only to certain ones, specify filter criteria with certain parameters.

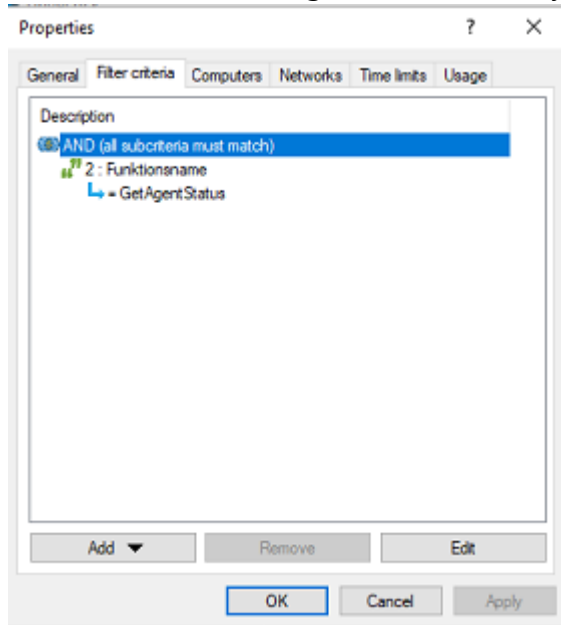
Please do the following:

1. Right-click the **Event filter definitions** subnode in the **EDR** node and select **New...** from the menu. A list of available events is displayed. Select the event 238.
2. On the **General** tab, check the **Windows Event Log** and **DriveLock Enterprise Service** options.



3. On the **Filter criteria** tab, select the parameters to filter by. By clicking the **Add** button you can select the appropriate criteria and the operators. In the example above, one criterion would be the **function name** GetAgentStatus.

Then the DriveLock Agent will send only the relevant events.




4.2.3 Drives

4.2.3.1 Drive settings

In the **Drives** node, select **Removable drive locking** and then doubleclick the **USB bus connected drives** option.

The Removable drive locking section provides two choices for your Linux policy:

 Note: Note that only the settings on the **General** tab apply to Linux policies.

1. Select the default option **Deny (lock) for all users (default)**:
This setting blocks the use of all drives connected via the USB interface for all users. You will need to define a whitelist rule that allows specific drives to be used.
2. Select **Allow** (for all users):
This option allows users to connect all drives over the USB interface. You will need to specify the drives you want to block in your drive rule.

4.2.3.2 Drive whitelist rules

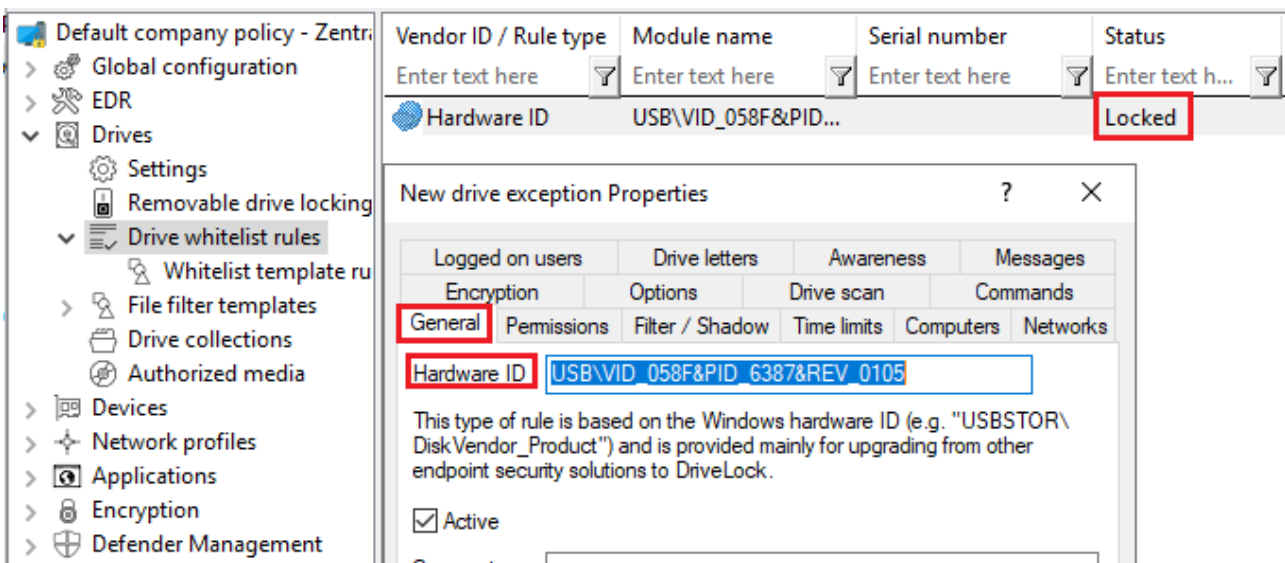
To configure a drive rule (as whitelist or blacklist), please proceed as follows:

1. In the **Drives** node, select **Drive whitelist rule**. Open the context menu, select **New** and then **Hardware ID rule**.

2. On the **General** tab, please enter the drive's hardware ID. This ID consists of the vendor ID (VID), product ID (PID) and revision number (REV).
3. On the **Permissions** tab, specify whether to deny (lock) or allow the drive (depending on your removable drive settings).

Warning: Please note that you cannot use the option 'Deny (lock) but allow access for defined users and groups' on Linux agents.

The figure below shows that the USB drive with the hardware ID USB\VID_058F&PID_6387&REV_0105 is blocked and cannot be used.



4.2.4 Devices

4.2.4.1 Supported device classes for Linux agents

The following DriveLock device classes are currently supported for Linux:

- **Devices:**

- Debugging and software protection devices (WinUSB, ADB) -> corresponds to Linux "Diagnostic Device class" (DC)
- Printers -> corresponds to Linux "Printers class" (07)
- Human Interface Devices (HID) -> corresponds to Linux "Human Interface Devices class" (03)
- Modems, network adapters -> corresponds to Linux "Communications & CDC control class" (02)
- Scanners and cameras -> corresponds to Linux "Image class" (06)
- Smartcard readers -> corresponds to Linux "Smart Card class" (0B)
- Sound, video and game controllers -> corresponds to Linux "Audio/Video/Audio&Video classes" (01|0e|10)

- **Controllers and Ports:**

- Bluetooth transmitters -> corresponds to Linux "Wireless Controller Class" (e0)
- USB controllers -> corresponds to Linux "Hub class" (09)

4.2.4.2 Device settings

In the **Devices** node, select **Device class locking**.

This section provides two choices for your Linux policy:

1. Open the **Controllers and Ports** section and doubleclick **USB controllers**. This setting lets you block or allow the complete USB interface of the Linux Agent.

The following options are available:

- a. Leave the setting as it is.

You do not check the **Enable controlling devices of this device class** option.


This is the default setting: **Not configured (not locked)**.

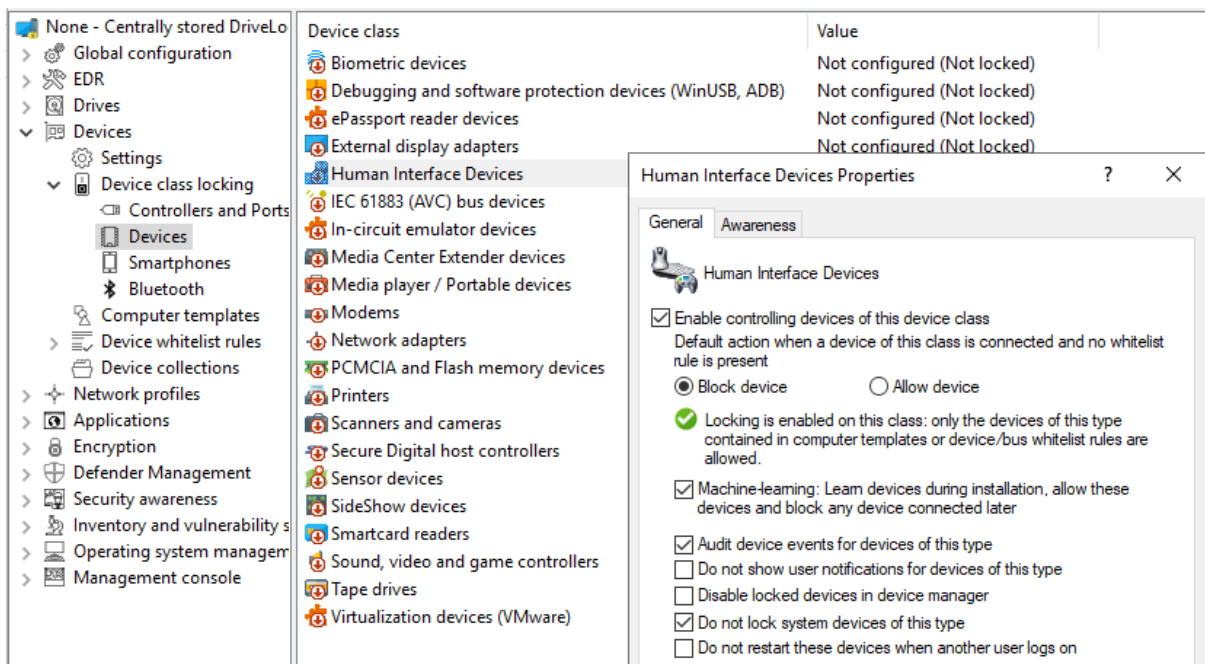
- b. Lock the USB interface.

Check the **Enable controlling devices of this device class** option and then select **Block device**. This means that you will need to configure appropriate whitelist rules for the devices you want to allow.

- c. Allow the USB interface.
Check the **Enable controlling devices of this device class** option and then select **Allow device**. This means that you will need to configure appropriate rules (blacklist) for the devices you want to block.
- d. If you select the **Machine Learning** option, all devices that are connected to the Linux Agent during installation are entered into a local whitelist and thereby allowed. Note here that the devices must also remain connected when the Linux agents are started. All other devices that are connected later are blocked.

2. Open the **Devices** section and doubleclick **Human Interface Devices**.

 Note: Please note that only some of the [device classes](#) available for Windows policies have a counterpart on the Linux side.



The same dialog is displayed as described above:

- a. Check the **Enable controlling devices of this device class** option and then select **Block device**.
All HID devices connected to the USB interface are blocked after the policy is assigned to the DriveLock Linux Agent. You must configure an appropriate whitelist rule for the devices you want to allow.
- b. Check the **Enable controlling devices of this device class** option and then select **Allow device**.


All HID devices are allowed. This means that you will need to configure appropriate rules (blacklist) for the devices you want to block.

- c. You can also select the **Machine Learning** option.
- d. Keep the default options checked. None of the other options are relevant for Linux agents.

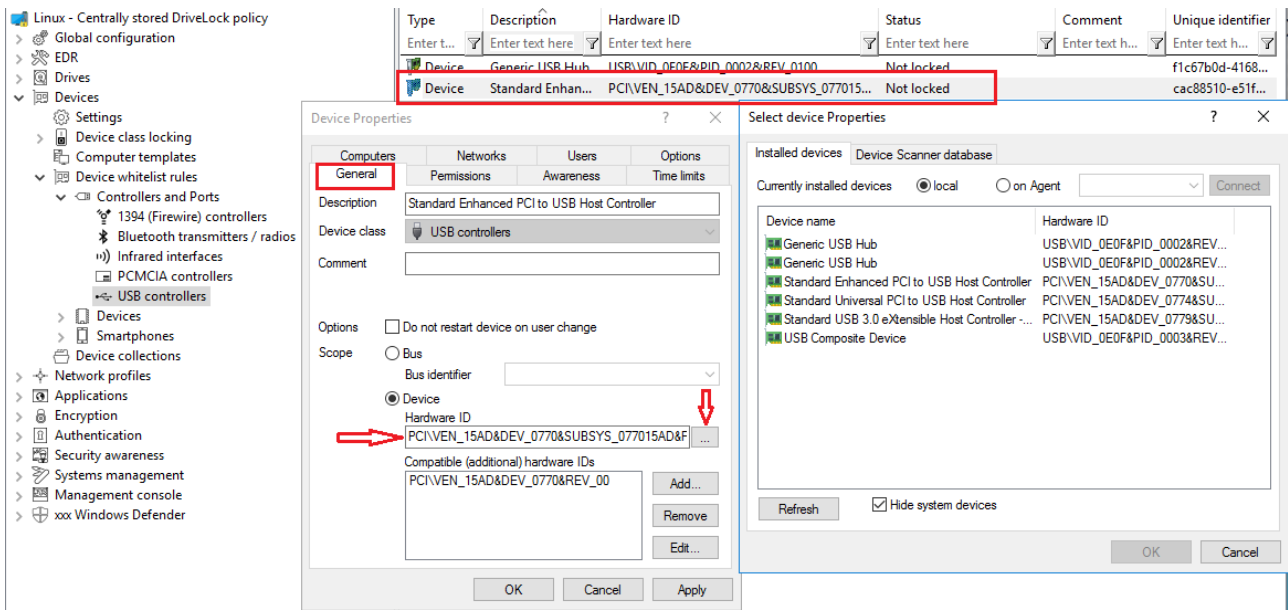
4.2.4.2.1 Device whitelist rules (for USB controllers)

To configure a device rule (as whitelist or blacklist) for USB controllers, please proceed as follows:

1. In the **Devices** node, open the **Device whitelist rules** subnode; select **Controllers and Ports** and then **USB controllers** (see figure).
2. Open the context menu, select **New** and then **Device or bus...**
None of the other options are relevant for Linux agents.
3. On the **General** tab, select the **Device** radio button and find the device you want to lock or allow (depending on whitelist or blacklist mode).
4. In the **Select devices** dialog you can display the devices that are installed **locally** or the devices that are currently connected to the DriveLock Linux Agent (**on Agent**). Note that the DriveLock Linux Agent must be online if you choose the 'on Agent' option.
5. On the **Permissions** tab, specify the appropriate **Device locking behavior**.

 Warning: Please note that you cannot use the option 'Deny (lock) but allow access for defined users and groups' on Linux agents.

In the figure below the USB controller with the ID **PCI\VEN_15AD&DEV_0770&SUBSYS_077015AD&REV_00** is allowed and has the status **not locked**.

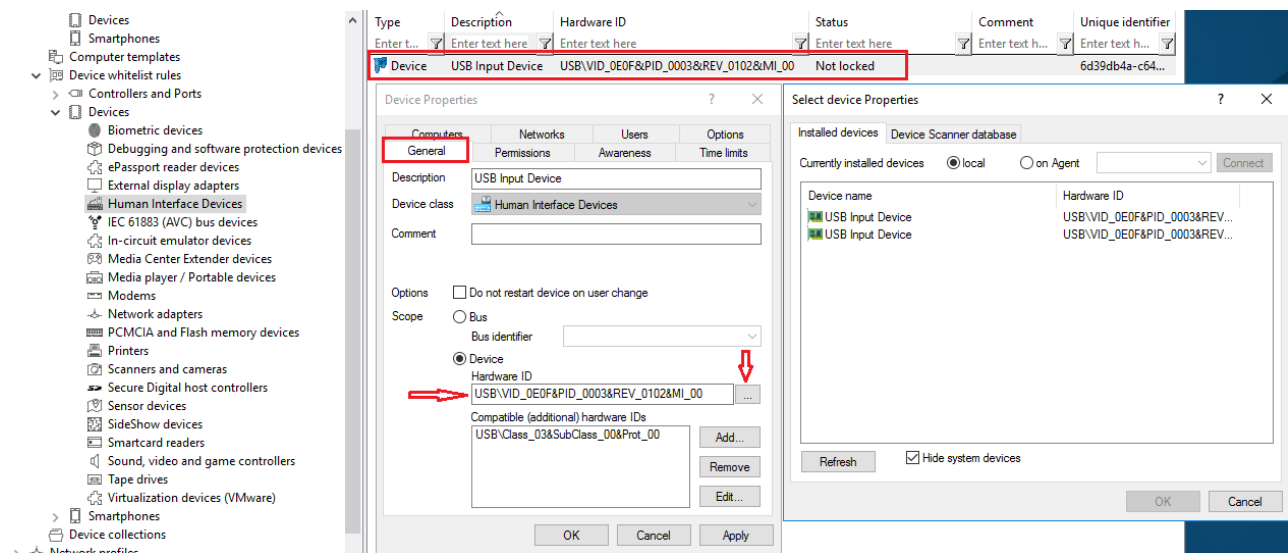


4.2.4.2.2 Device whitelist rules (for devices)

To configure a whitelist rule for devices, proceed as explained in [Device whitelist rules \(for USB controllers\)](#) except that you select **Input Devices (HID)** in the **Device whitelist rules** sub-node.

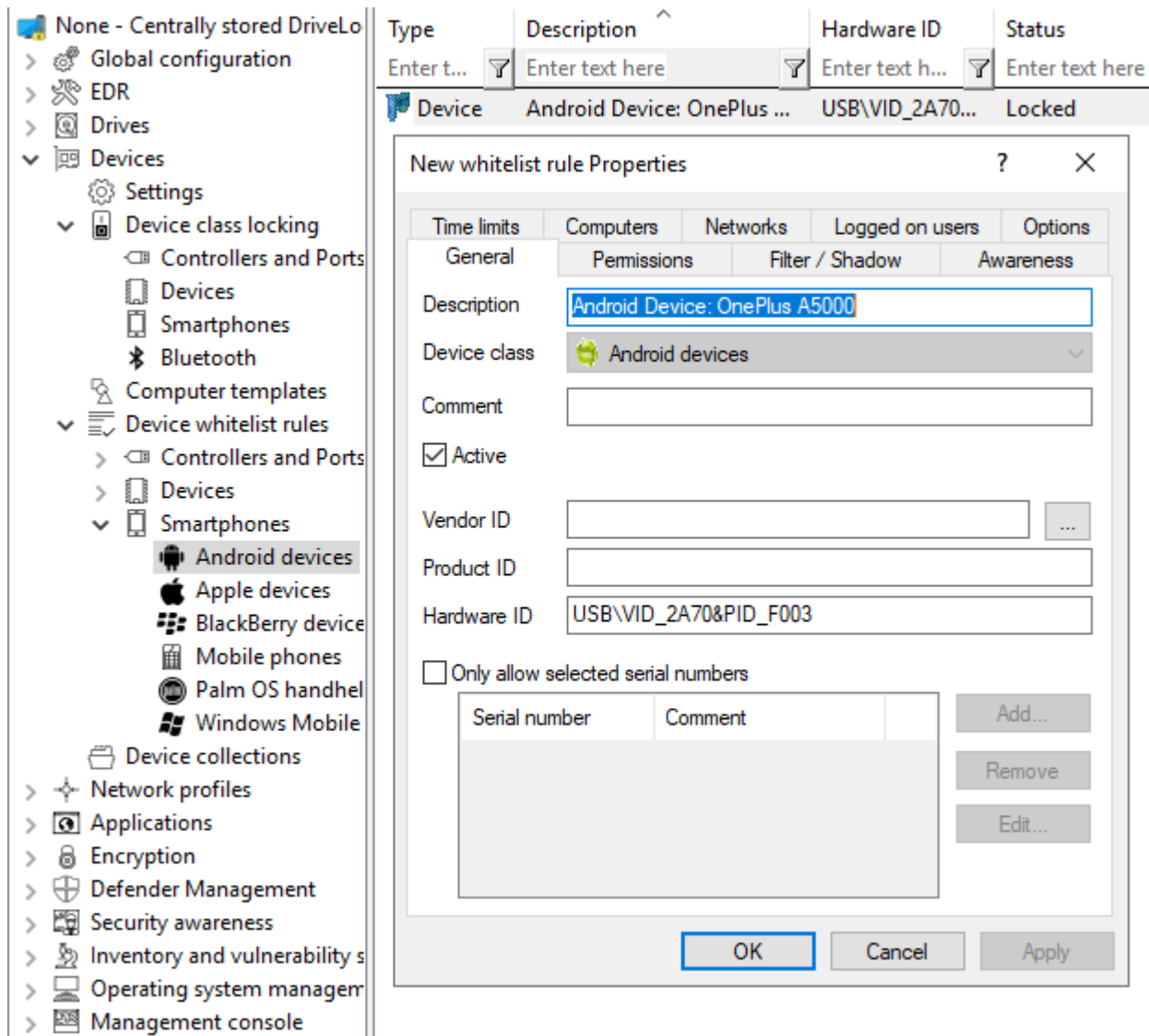
All other steps are identical.

In the figure below, the USB device with the hardware ID **USB\VID_0E0F&PID_0003&REV_0102&MI_00** has the status **Not locked**.




4.2.4.2.3 Android and Apple devices

Creating rules for Android and Apple devices is also supported, see the figure below. Similar to other device categories, you need the hardware ID or serial number of the device to do so. On the **Permissions** tab, you can set the appropriate blocking settings.



The agent identifies a device as an Android or Apple device if it appears in the list of devices that is installed with the Drivelock Agent. The list contains the product and vendor IDs (or serial numbers); when connecting the respective device, the IDs are compared.

This list is located in the system in the /etc/udev/rules.d/ directory in the **51-drivelock-apple.rules** and **51-drivelock-android.rules** files.

 Note: The list can be extended. If you need assistance with this, please contact our support.

4.2.4.2.4 Devices collections

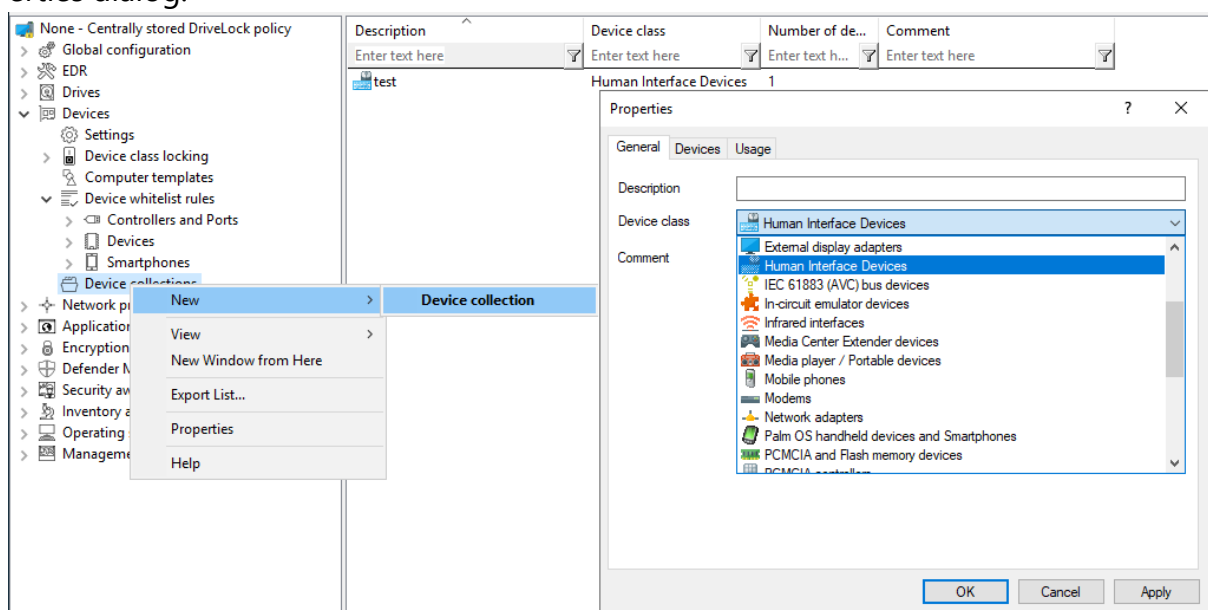
You can use device collections on Linux agents. They simplify managing devices of the same type when the same settings apply to them, while reducing the number of whitelist rules needed. Device collections may contain several similar devices and can be used in whitelist rules.

Note that only some [device classes](#) are supported on Linux agents. By specifying the corresponding hardware ID, the device match process will ignore the device class if possible.

4.2.4.2.4.1 Create device collections

How to create a device collection:

1. In the **Devices** node, go to the **Device collections** subnode and then click **New** from the context menu.
2. Select the required device class on the **General** tab in the device collection's properties dialog.



3. You can then select the devices on the **Devices** tab by clicking the **Add** button.
4. In the following dialog, select the corresponding **hardware ID** of the device. You can also connect to the Linux agent and select devices directly.
5. Once you have created a device collection, you can use it in [device collection rules](#).

4.2.5 Applications

DriveLock includes some Application Control options for Linux agents.

1. The [Scanning and blocking mode](#) setting activates the Application Control functionality.
2. On Linux, you can use two application rules: [File properties rule](#) and [Special rule](#) (both for Linux).

To be able to use Application Control for Linux, specific [requirements](#) regarding the Linux kernel must be met.


You can find additional information about Application Control in the corresponding documentation at [DriveLock Online Help](#)..


4.2.5.1 Prerequisites for Application Control on Linux Agents

To support the full functionality of Application Control with whitelisting, the following requirements must be met:

- The fanotify API must be active in the Linux kernel
- The Linux kernel must be greater than 5.0.
In kernel versions smaller than 5.0, only the fanotify flag FAN_OPEN_PERM is available and only blacklisting is possible.
- The file system must support fanotify events.
Current list of supported file systems:
 - bfs
 - btrfs
 - cifs
 - ecryptfs
 - ext2
 - ext3
 - ext4
 - fuseblk
 - fuse.vmhgfs-fuse
 - iso9660
 - jfs

- minix
- msdos
- nfs
- nfs4
- nssvol
- ncpfs
- overlay
- overlayfs
- ramfs
- reiserfs
- smbfs
- squashfs
- tmpfs
- udf
- vfat
- xfs
- zfs

 Note: We do not support running Application Control on Linux alongside other fanotify-based security solutions. This can cause unexpected behavior, including suspending the operating system.

 Note: It is not possible to use Application Control inside containers due to the limitations of fanotify.

4.2.5.2 Scanning and blocking mode

Use this setting to select the mode DriveLock uses to scan applications on the Linux agent and/or to initiate appropriate actions.

Please do the following:

Select **Set to fixed value**, and then select one of the following options from the list:

- **Audit only:** events are generated only; you can analyze them later
- **Whitelist:** applications may only be executed if a corresponding whitelist rule exists. All other applications will be blocked.
- **Blacklist:** applications are blocked only if there is a corresponding blacklist rule. All other applications are allowed.
- **including DLLs:** this addition also checks the shared libraries
- **(simulate):** this addition means that the effects of your rules are tested in advance and corresponding events are generated.

4.2.5.3 File properties rule (for Linux)

This rule allows you to specify different file properties to filter by. This rule can be created as a whitelist or blacklist rule.

Please do the following:

Go to the **Applications** node, select **Application rules** and then select **File properties rule (for Linux)...** from the context menu.

1. On the **General** tab, the first thing you do is set the rule type.

Then you have the following choices:

- **Path:** Specify a path in Linux format (e.g. /home/test/) if you want to allow (or block) applications from a specific path. Wildcards are allowed.
- **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.
- **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.

A combination of the options is possible.

2. On the **Time limits** tab you can specify the times when you want the rule to be active.
3. On the **Computers** tab you can specify the computers where the rule will be active.

4.2.5.4 Special rule (for Linux)

The special rule can be used only as a whitelist rule.

Please do the following:

1. Go to the **Applications** node, select **Application rules** and then select **Special rule (for Linux)...** from the context menu.
2. On the **General** tab you have three options to choose from:
 - **Program file is part of the operating system:**

This option automatically allows operating system programs from the following system directories:

 - /bin, /sbin, /lib, /lib64, /usr, /etc
 - Ubuntu: /snap
 - Suse: /.snapshots
 - **Program file is part of DriveLock**


Here binaries are allowed in the Drivelock installation folder and the "bin" folder below it.

The custom installer drivelockd-install.sh is not included, the user must add a rule to run the script in case of upgrades.
 - **Any program is started:**

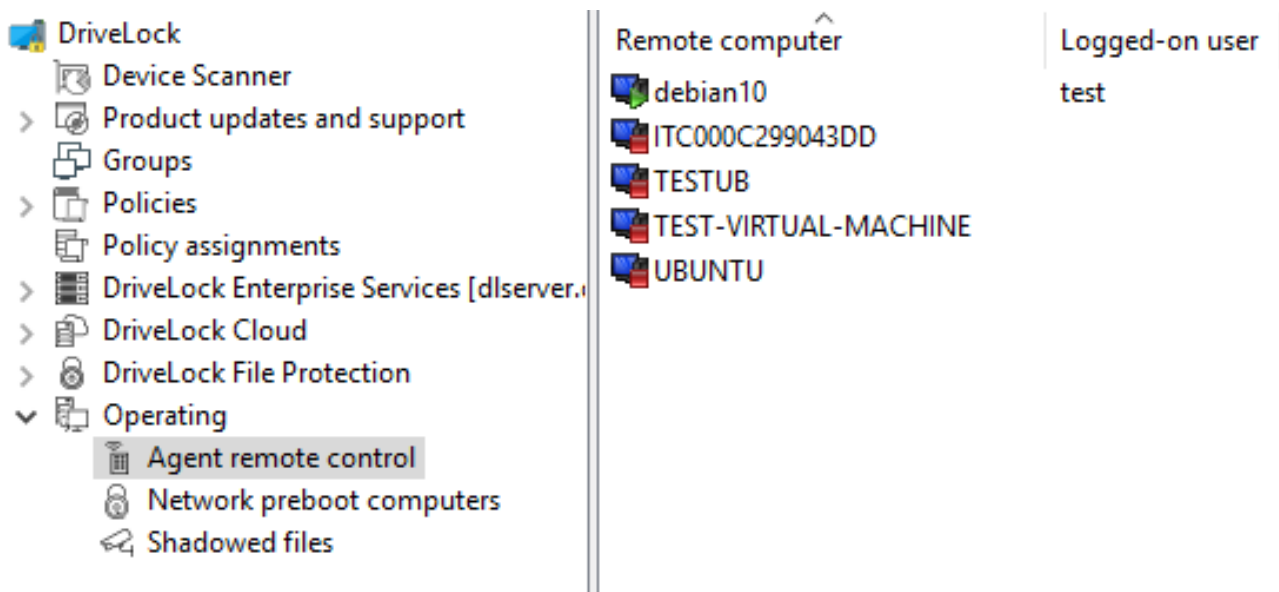
All started applications are allowed here, regardless of the directory.
3. On the **Time limits** tab you can specify the times when you want the rule to be active.
4. On the **Computers** tab you can specify the computers where the rule will be active.

4.3 Agent remote control

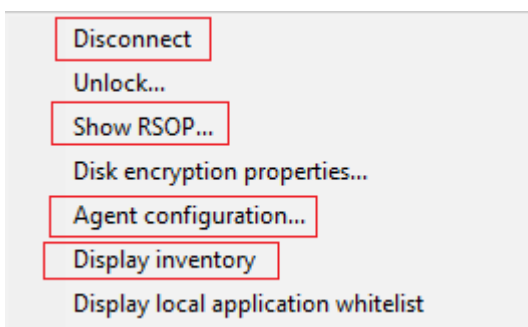
Open the **Operating** node in the DriveLock Management Console and select **Agent remote control**. You see a list of client computers where the DriveLock Agent is installed (see figure).

 Note: Please refer to the DriveLock Administration Guide at drivelock.help for further information on agent remote control.

Open the context menu of the Linux client you selected and click **Connect**.



The following agent remote control actions are relevant for Linux agents:

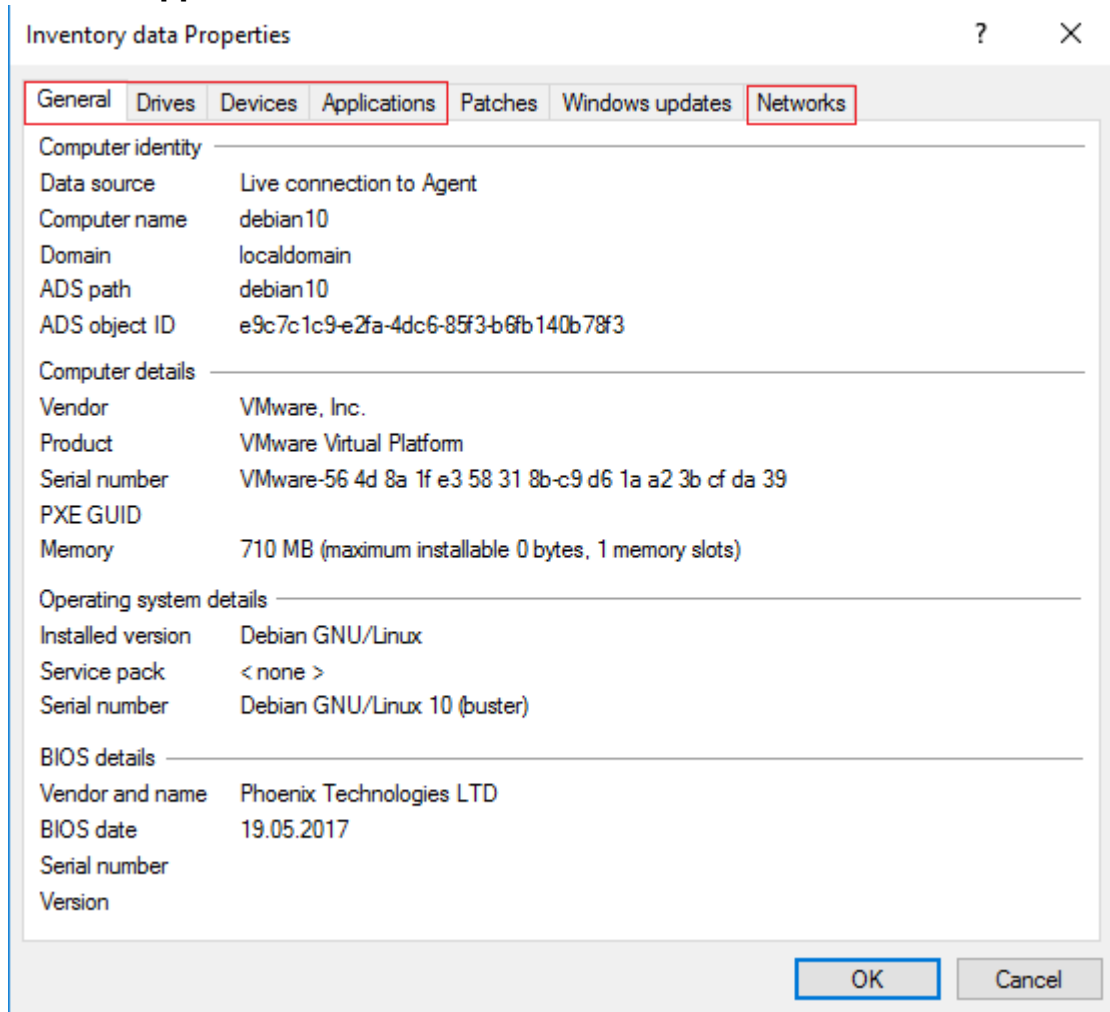


1. **Disconnect** the Linux agent.
2. **Show RSOP...**
Click this option to view a summary of the policy (Resultant Set of Policy) assigned to the Linux agent. You can not change any settings here.
3. **Agent configuration...**

Click this option to open a dialog with information on the agent's configuration. It shows you the server your Linux agent receives the centrally stored policy from and, if necessary, you can add another server or enter another tenant on the **Options** tab.

4. Display inventory

Click here to get inventory information on your Linux agent (on the **General**, **Drives**, **Devices**, **Applications** and **Networks** tabs).



5 Linux agents in the DCC

DriveLock Linux Agents are displayed in the DriveLock Control Center (DCC) like other DriveLock Agents.



Note: Refer to the DriveLock Control Center documentation for a detailed description of the DCC at drivelock.help.

The following views and features are important for Linux agents:

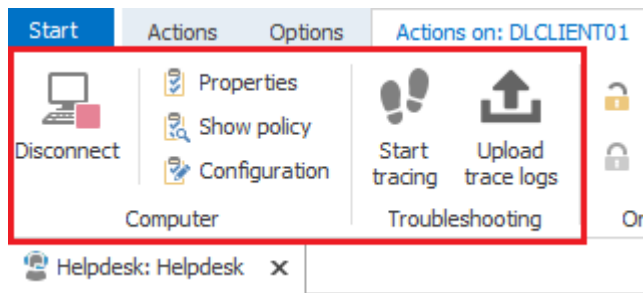
- **HelpDesk:**
The HelpDesk view provides status and other information about your Linux agents. Find a description of the actions [here](#).
- **Statistic report:**
Agent alive: Here you can see the Linux agents that recently reported to DES.
- **Event report:**
Shows all events the Linux agent sends to the DES. Refer to the list of events [here](#).
- **Inventory:**
Computer: Here you get an overview of your Linux agents with information about the respective Linux computer, operating system and DriveLock Linux Agent.
- **Open DOC:**
Open the [DriveLock Operations Center \(DOC\)](#) in your browser to check the status of the DriveLock Linux Agents.

5.1 DCC: Help Desk actions

On the **Actions** tab, you can use the **Connect** button for DriveLock Linux Agents.

This action starts the agent remote control. You can also start remote control from the [DriveLock Management Console](#).


1. Connect: Select the Linux agent on the list and click **Connect** or enter the name of the Linux client in the text box below the button.
2. Once the connection is set up, a new tab will appear, **Actions on: [Name of the Linux client]**.
Here you can choose the following actions (see figure):




3. Click **Properties** for detailed information about the status of the Linux agent. The **General** tab provides an overview. By clicking the **Refresh policy...** button, you start the policy update on the agent.
4. Click **Show policy** to display the Resultant Set of Policy (RSOP) of the Linux agent.
5. Clicking **Configuration** opens a dialog with information about the configuration of the Linux agent. You can add another server or select another tenant here, for example.
6. Please contact DriveLock Technical Support if you want to enable **tracing** or debugging for your Linux agents.

6 Linux agents in the DOC

DriveLock Linux Agents are displayed in the DriveLock Operations Center (DOC) like other DriveLock Agents.

 Note: Refer to the DriveLock Control Center documentation for an introduction to the DOC at drivelock.help.

The following DOC views are relevant for Linux agents:

- **Computer:** Filter by **OS Type** ( icon), for example, to group your Linux agents by their OS type. Select any Linux agent to check details.
- **Groups:** If you have defined a DriveLock group for your Linux agents, it is displayed here with information about the respective members and the assigned policies.
- **Events:** This view lists the events that a Linux agent sends to the DES.
- **EDR:** The Endpoint Detection & Response view provides continuous monitoring and allows you to configure your response to security alerts.
- **Accounts:** This view provides a list of all user accounts that are allowed to access the DOC. It also shows information on status and roles along with name and logon details.

7 List of events

The table contains all events related to Linux as displayed in the DriveLock Control Center or the DriveLock Operations Center (DOC). All events below are triggered by DriveLock.

You can find a list of all events that are important in connection with DriveLock in the Events documentation at [DriveLock Online Help](#).

The DriveLock Linux Agent sends the following events to the DES:

| Event ID | Event level (Information, Warning, Error) | Event text | Description |
|----------|---|------------------------------|--|
| 105 | Information | Service started | The [name] service was started. |
| 108 | Information | Service stopped | The service [name] was stopped. |
| 110 | Audit | Drive connected and unlocked | The drive [name] ([category]) was added to the system. It is a [type] bus device. The drive is [locked/unlocked] for this event's user account. Device Id: [ID] [ID] (Rev. [rev]) (Serial number [number]) Applied whitelist rule: [rule] Screen state (keyboard [Win]-[L]): [state] |
| 111 | Audit | Drive connected and locked | The drive [name] ([cat- |

| Event ID | Event level (Information, Warning, Error) | Event text | Description |
|----------|---|---------------------------------|---|
| | | | <p>egory]) was added to the system. It is controlled by {Product} because of company policy. As an ACL was applied to the drive, some users may no longer be able to access it. It is a [type] bus device. The drive is [locked/unlocked] for this event's user account. Device Id: [ID] [ID] (Rev. [rev]) (Serial number [number]) Applied whitelist rule: [rule] Screen state (keyboard [Win]-[L]): [state]</p> |
| 129 | Audit | Device connected and locked | <p>The device [name] was connected to the computer. It was locked due to company policy. Device type: [type] Hardware ID: [ID] Class ID: [ID] Applied whitelist rule: [rule] Screen state (keyboard [Win]-[L]): [state]</p> |
| 130 | Audit | Device connected and not locked | <p>The device [name] was connected to the computer. Device type: [type] Hardware ID: [ID] Class ID: [ID] Applied whitelist rule: [rule]</p> |

| Event ID | Event level (Information, Warning, Error) | Event text | Description |
|----------|---|-------------------------------------|--|
| | | | Screen state (keyboard [Win]-[L]): [state] |
| 152 | Warning | Policy storage extraction failed | The policy storage container [name] cannot be unpacked to the local computer. Some functions relying on files stored in this container may fail. |
| 153 | Warning | Configuration file applied | The configuration file [name] was successfully applied. |
| 154 | Error | Configuration file download error | The configuration file [name] could not be downloaded. Error code: [code] Error: [error] |
| 158 | Error | Configuration file error | The configuration file [name] could not be read. Error code: [code] Error: [error] |
| 191 | Warning | {Pre-fixEnterpriseService} selected | The {Pre-fixEnterpriseService} [name] was selected by {Product}. Connection ID: [ID] Used for: [Invent- |

| Event ID | Event level (Information, Warning, Error) | Event text | Description |
|----------|---|--|---|
| | | | ory/Recovery/Events] |
| 192 | Warning | {Pre-fixEnterpriseService} not available | No {Pre-fixEnterpriseService} is available because no valid server connection is configured. |
| 235 | Error | SSL: Cannot set up | The encrypted communications layer (SSL) could not be set up. Error: [error] |
| 236 | Error | Remote control: Cannot set up server | The remote control server component could not be set up. Agent remote control will be unavailable. Error: [error] |
| 237 | Error | Remote control: Internal error | Agent remote control: An internal SOAP communications error occurred. Error: [error] |
| 238 | SuccessAudit | Remote control: Function called | An Agent remote control function was called. Calling IP address: [IP address] Called function: [function] |

| Event ID | Event level (Information, Warning, Error) | Event text | Description |
|----------|---|---|--|
| 243 | Error | Cannot open database | A database could not be opened. Database file: [name] Error code: [code] Error: [error] |
| 246 | Error | Cannot store configuration status | The Agent cannot store the configuration status used by other {Product} components. Error code: [code] Error: [error] |
| 247 | Error | Cannot initialize configuration store | {Product} Agent cannot initialize the configuration database stores. |
| 249 | Error | Configuration file: Fall-back configuration applied | A configuration using configuration files was detected but no settings could be retrieved from a configuration database. {Product} will fall-back to a configuration where all removable drives are blocked. |
| 250 | Warning | Configuration file: Using cached copy | The configuration file [name] could not be loaded from its original location. A locally cached copy |

| Event ID | Event level (Information, Warning, Error) | Event text | Description |
|----------|---|---|--|
| | | | was used. |
| 251 | Error | Configuration file: Cannot extract | A {Product} configuration file could not be extracted.%rSettings from this file will not be applied. Database file: [name] Error code: [code] Error: [error] |
| 264 | Error | Cannot merge configuration database with RSoP | Cannot merge the configuration database [name] into the resulting set of policy. |
| 287 | Error | No server defined for inventory | No server is defined for uploading collected inventory data. |
| 288 | Information | Inventory collection successful | Hard- and software inventory data was successfully collected and uploaded. DES server: [server name] Connection ID: [ID] |
| 289 | Information | Inventory collection failed | An error occurred while collecting hard- and software inventory data. DES server: [server name] Connection ID: [ID] Error: [error] |

| Event ID | Event level (Information, Warning, Error) | Event text | Description |
|----------|---|--|--|
| 294 | Error | Cannot download centrally stored policy | The centrally stored policy [name] could not be downloaded. Server: [name] Error: [error] |
| 295 | Error | Centrally stored policy: Cannot extract | A centrally stored policy could no be extracted. Settings from this file will not be applied. Configuration ID: [ID] Error code: [code] Error: [error] |
| 297 | Error | Centrally stored policy: Fall-back configuration applied | A configuration using centrally stored policies was detected but no settings could be retrieved from a server. {Product} will fall-back to a configuration where all removable drives are blocked. |
| 299 | Information | Centrally stored policy downloaded | The centrally stored policy [name] was successfully downloaded. Configuration ID: [ID] Version: [version] |
| 443 | Error | Component start error | A {Product} system component could not be started on this computer. Error |

| Event ID | Event level (Information, Warning, Error) | Event text | Description |
|----------|---|-----------------|---|
| | | | code: [code] Error: [error] Component ID: [ID] |
| 473 | Audit | Process blocked | The execution of a process was blocked by company policy. Process: [ProcessName] File Hash: [ProcessHash] Applied rule: [ObjectID] Rule type: [WType] File owner (user name): [UserName] File owner (user sid): [SID] File version: [FileVersion] Certificate issuer: [CertIssuer] Certificate subject: [CertSubject] Certificate serial: [CertSerNo] Certificate thumb print: [CertThumbprint] Description: [VerDescription] Product: [VerProduct] Command line: [CmdLine] Parent Process: [ProcessName] ([ProcessGuid]) |
| 474 | Audit | Process started | A process was started. Process: [ProcessName] File Hash: [ProcessHash] Applied rule: [ObjectID] |

| Event ID | Event level (Information, Warning, Error) | Event text | Description |
|----------|---|----------------------------------|---|
| | | | Rule type: [WType] File owner (user name): [UserName] File owner (user sid): [SID] File version: [FileVersion] Certificate issuer: [CertIssuer] Certificate subject: [CertSubject] Certificate serial: [CertSerNo] Certificate thumb print: [CertThumbprint] Description: [VerDescription] Product: [VerProduct] Unique Process ID: [ProcessGuid] Command line: [CmdLine] Parent Process: [ProcessName] ([ProcessGuid]) |
| 520 | Error | All {PrefixES} not reachable | Cannot load company policy. All configured {PrefixEnterpriseService}s are not reachable. |
| 521 | Error | Cannot determine computer token | Cannot determine the computer token. Error code: [code] Error: [error] |
| 522 | Error | Error loading policy assignments | An error occurred while loading policy assignments from server [name]. Error: |

| Event ID | Event level (Information, Warning, Error) | Event text | Description |
|----------|---|-------------------------------|---|
| | | | [error] |
| 523 | Error | Policy integrity check failed | The integrity of an assigned policy could not be verified.%rPolicy ID: [ID] Policy name: [name] Actual hash: [value] Expected hash: [value] |
| 533 | Warning | No policy - wiped | No valid policy available - the company policy was wiped because the computer was offline for a long period of time. |
| 584 | Information | Inventory started | Inventory generation was triggered by DES. |
| 639 | Error | Server certificate error | Server certificate error detected. Certificate: [name]. Error message: [text] |
| 648 | Audit | DLL blocked | The loading of a DLL was blocked by company policy. Process: [ProcessName] ([ProcessGuid]) Applied rule: [ObjectID] |

| Event ID | Event level (Information, Warning, Error) | Event text | Description |
|----------|---|------------|--|
| | | | Rule type: [WType] DLL File Name: [ProcessName] DLL File Hash: [ProcessHash] File owner (user name): [UserName] File owner (user sid): [SID] File version: [FileVersion] Certificate issuer: [CertIssuer] Certificate subject: [CertSubject] Certificate serial: [CertSerNo] Certificate thumb print: [CertThumbprint] Description: [VerDescription] Product: [VerProduct] |
| 649 | Audit | DLL loaded | A DLL was loaded. Process: [ProcessName] ([ProcessGuid]) Applied rule: [ObjectID] Rule type: [WType] DLL File Name: [ProcessName] DLL File Hash: [ProcessHash] File owner (user name): [UserName] File owner (user sid): [SID] File version: [FileVersion] Certificate issuer: [CertIssuer] Certificate subject: [CertSubject] Certificate serial: [CertSerNo] Cer- |

| Event ID | Event level (Information, Warning, Error) | Event text | Description |
|----------|---|------------|--|
| | | | tificate thumb print: [CertThumbprint] Description: [VerDescription] Product: [VerProduct] |

8 Command line tool

Use this command line tool to change the local configuration of a Linux Agent or to display the current configuration. You will find the **drivelock-ctl** tool in the installation directory of the DriveLock Linux Agent.

The following commands are available (see figure):

```
test@debian10:~$ /opt/drivelock/drivelock-ctl -h
-----
Drivelock Linux Agent- Command line tool
-----
DriveLock, 19.2.5.27684

Usage: drivelock-ctl [Option]

Options:
  -enabletracing           Enable service logging
  -disabletracing         Disable service logging
  -updateconfig            Trigger a configuration update
  -showstatus              Show drivelock configuration status
  -settenant <tenantname> Set tenant name
  -setserver [http(s)://<server>:<port>] Set one or more server(DES) URLs,
                                         URLs should be delimited by ;
```

- **enabletracing**: Enables tracing to the **Drivelock.log** file residing in the installation directory in the **log** child directory.
- **disabletracing**: Disables tracing
- **updateconfig**: Updates your configuration, e.g. if you have made changes to your policies. The Linux agent then immediately connects to the DES and loads the changes
- **settenant**: Specifies the tenant for your Linux agent
- **setserver**: Specifies the DES that communicates with the Linux agent
- **showstatus**: Shows the current status of the Linux agent and provides information such as the last time the DES was contacted and the policies assigned (see figure below)

```
test@debian10:~$ /opt/drivelock/drivelock-ctl -showstatus
```

Agent Identity:

```
-----  
Agent version: 19.2.5.27684  
Computer Name: debian10  
Computer GUID: e9c7c1c9-e2fa-4dc6-85f3-b6fb140b78f3  
Domain Name: localdomain  
OS Name: Debian GNU/Linux  
OS Version: 10 (buster)
```

Agent Configuration & Status:

```
-----  
Tenant : kav  
Server URL(s) : https://192.168.8.207:6067  
Last server contact at : 10.02.2020 15:34:34  
Last inventory at : unknown
```

Assigned Policies:

```
-----  
1  CSP ID: 55f8de53-9444-4151-979b-8895c2cdc6da  
   ConfigName: Linux Tenant Test  
   Version: 7  
   Target: LinuxGroup  
  
2  CSP ID: aad3f718-228f-4737-871b-e16e13fffc7a  
   ConfigName: TestEvtNotCfg  
   Version: 2  
   Target: LinuxGroup
```


Copyright

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user.

© 2022 DriveLock SE. All rights reserved.

DriveLock and others are either registered trademarks or trademarks of or its subsidiaries in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.