Table of Contents

Related Documents .................................................. 1
1. iDENprotect\textsuperscript{server} Software Suite .......................... 2
2. Introduction to iDENprotect\textsuperscript{server} ......................... 3
   2.1. iDENprotect\textsuperscript{server} Components .......................... 3
   2.2. iDENprotect Enabled Devices and iDENprotect\textsuperscript{plus} Tokens .......................... 4
   2.3. Application Protection ........................................... 5
3. Operating iDENprotect\textsuperscript{server} .............................. 7
   3.1. iDENprotect\textsuperscript{server} Startup Workaround .................... 7
   3.2. iDENprotect\textsuperscript{server} Shutdown Workaround .................... 7
   3.3. Starting and Stopping other Services ............................. 8
   3.4. Activating iDENprotect\textsuperscript{server} Licence ................... 8
4. iDENprotect\textsuperscript{server} Admin Console ....................... 10
   4.1. Management Console Views ....................................... 11
      4.1.1. Dashboard Tab ................................................. 11
      4.1.2. Devices Tab .................................................. 11
      4.1.3. Users Tab .................................................... 12
      4.1.4. Logs Tab ..................................................... 12
      4.1.5. Reports Tab .................................................. 13
      4.1.6. Site Tab ...................................................... 13
      4.1.7. Config Tab ................................................... 14
5. iDENprotect Provisioning ............................................. 15
   5.1. Registering Devices .............................................. 16
   5.2. Assigning Devices ............................................... 16
   5.3. Enrolling Devices ................................................ 18
6. Configuring iDENprotect\textsuperscript{server} .......................... 19
   6.1. Configuration Parameters ........................................ 20
      6.1.1. CA Parameters ............................................... 20
      6.1.2. Database Parameters ........................................ 21
      6.1.3. Email Parameters ............................................ 21
      6.1.4. LDAP Parameters ............................................ 22
      6.1.5. Logging Parameters ........................................ 23
      6.1.6. Web Server Parameters ..................................... 24
      6.1.7. User Enrolment Properties .................................. 25
   6.2. Configuring Email Messages ...................................... 26
7. Managing Network Ports on iDENprotect\textsuperscript{server} .......... 27
   7.1. Configuring Firewall ............................................. 27
8. Managing Certificate Authorities (CAs) ............................ 28
9. Managing iDENprotect Users ......................................... 29
   9.1. Manual User Enrolment (with or without LDAP) ................ 29
   9.2. LDAP Auto-enrolment ............................................ 29
10. Configuring LDAP ................................................... 30
    10.1. Connecting to LDAP server ..................................... 30
    10.2. Configuring LDAP Server ....................................... 30
11. Configuring iDENprotect server Database ......................................................... 33
11.1. Configuring MariaDB .................................................................................. 33
11.2. Configuring Oracle Database .................................................................... 33
12. Logging .......................................................................................................... 35
12.1. iDENprotect server Log Events ................................................................. 35
12.2. Log Search Examples .............................................................................. 36
13. Managing Web Server .................................................................................. 38
13.1. Installing a Trusted SSL Certificate ......................................................... 38
  13.1.1. Generating Private Key and CSR ........................................................... 38
  13.1.2. Signing the CSR .................................................................................. 38
  13.1.3. Installing the Certificates .................................................................... 39
  13.1.4. Testing the Certificate ........................................................................ 39
13.2. Configuring nginx ...................................................................................... 40
Appendix A: Using Built-in EJBCA Certificate Authority ................................. 41
  A.1. Installing EJBCA Administrator Certificate ................................................ 41
  A.2. Using EJBCA Web Console ...................................................................... 41
    A.2.1. Creating New iDENprotect Certificate Profile ................................. 42
    A.2.2. Creating End Entity Profile ................................................................. 43
  A.3. Configuring EJBCA Startup ..................................................................... 45
Appendix B: Cipher Suites in iDENprotect server Admin Console ................. 46
Appendix C: iDENprotect Device Keystore ....................................................... 47
Appendix D: Database Schema .......................................................................... 47
This guide describes how to manage the iDENprotect\textsuperscript{server}, including how to start and stop the server, how to change network ports, configure logs and reports, and how to troubleshoot iDENprotect\textsuperscript{server} and Certificate Authorities.

This guide is intended for iDENprotect\textsuperscript{server} administrators.

**Related Documents**

For more information, see the following documents in the documentation set:

- iDENprotect\textsuperscript{server} Installation Guide version 2.0.0
- iDENprotect for iOS Developer Guide 1.0
- iDENprotect Security Architecture
1. iDENprotect\textsuperscript{server} Software Suite

iDENprotect provides multi-factor authentication using PKI technology and standards based authentication protocols. iDENprotect consists of a number of feature rich components ranging from application development tools, integration solutions, identity management, mobile application, user & device provisioning and user management console.

iDENprotect is the name of iDENprotect’s multi-layered software-based identity management and authentication security software, scalable to any number of users. It is used to manage and secure digital identities of users.

iDENprotect Enabled Device is a mobile device, such as a mobile phone, a tablet computer or laptop that has been enabled to work as an, or with an, authentication device in the iDENprotect framework using an iDENprotect App. iDENprotect Apps are available for most device platforms.

iDENprotect\textsuperscript{plus} is an optional hardware authentication component of the iDENprotect solution. It is a physical, credit card form factor device which allows easy multi-factor authentication.

iDENprotect\textsuperscript{server} is the backend management server, central to the iDENprotect solution. The iDENprotect\textsuperscript{server} is flexible due to its ability to connect to external components such as Databases, Certification Authorities and directory services as well as having the ability to create and use internal components such as internal Databases, internal Certificate Authorities. iDENprotect\textsuperscript{server} manages the full lifecycle of the user centrally.

In this document, both iDENprotect Enabled Devices and iDENprotect\textsuperscript{plus}s are sometimes collectively referred to as Devices.
2. Introduction to iDENprotect<sup>server</sup>

iDENprotect<sup>server</sup> is the management server for any organisation that uses iDENprotect and iDENprotect<sup>plus</sup>. The main purpose of iDENprotect<sup>server</sup> is to allow a safe and secure enrolment to the service, to maintain central records of authorised users with their authentication devices, and to revoke and remove access when necessary. iDENprotect uses Public Key Infrastructure (PKI), the most secure and trusted way to manage digital identities and to authenticate users onto a variety of platforms natively. iDENprotect removes the complexities associated with PKI deployments due to its intelligent automation, ensuring a simple user experience for the user but also the administrator who maintains and supports the platform.

iDENprotect<sup>server</sup> is managed through a simple and clear web console that provides functions such as user management, device management, reports, logs and general configuration settings to ensure the solution fits within your business and operational processes. Depending on the deployment, there are other functions such as to view certificates within the internal CA component and also internal database attributes.

![Diagram of iDENprotect server components](image)

*Figure 1. iDENprotect<sup>server</sup> Typical Enterprise Deployment*

Sensitive communications to and from the iDENprotect<sup>server</sup> are secured to prevent eavesdropping attacks. iDENprotect<sup>server</sup> listens to incoming authentication requests from enroled Devices in HTTPS port 443. When requests arrive, iDENprotect<sup>server</sup> validates them based on the information contained within pre-registered digital certificates and only processes authenticated requests. All persistent data in the iDENprotect<sup>server</sup> is stored securely within databases.

When an iDENprotect-secured transaction is started, the iDENprotect App or iDENprotect<sup>server</sup> forms a connection to the iDENprotect<sup>server</sup> to authenticate the user. After successful authentication, the user can access the secured content.

2.1. iDENprotect<sup>server</sup> Components

iDENprotect<sup>server</sup> is a modular system. At the core of the system is always the iDENprotect<sup>server</sup> application, which has the following subcomponents:
The standalone iDENprotect server installer package includes MariaDB, nginx and the iDENprotect internal CA component. For organisations who are using BlackBerry software - in particular BlackBerry Dynamics - iDENprotect server contains a separate PKI connector application that relays authentication information securely between iDENprotect and BlackBerry Dynamics runtime.

iDENprotect server is a Java application that has an embedded Tomcat application server. The iDENprotect server is able to perform its purpose either as a single application server or operate behind other application servers, such as nginx or reverse proxy servers. If the environment where iDENprotect server is deployed has multiple zones for security purposes, its modular design allows for other web facing components to be placed in front of the iDENprotect server in order to inspect the traffic or for other security needs.

2.2. iDENprotect Enabled Devices and iDENprotectplus Tokens

A mobile device secured with iDENprotect technology is called an iDENprotect Enabled Device. In iDENprotect Enabled Devices, iDENprotect security operations are secured by a private key in the Secure Enclave on an iOS device, or an equivalent secure location on other platforms.

iDENprotect architecture is used when a mobile device function or resource requires authentication. At that point, the mobile device connects to iDENprotect server over a TLS-encrypted Ethernet connection (only TLS 1.2 supported by default). The user is prompted to open an iDENprotect App (preinstalled on the mobile device) and to authenticate biometrically (with Touch ID, for example) or PIN code.

An example of an iDENprotect App is iDENprotect for BlackBerry, which integrates the BB Dynamics Mobile Application Management (MAM) framework with iDENprotect authentication. iDENprotect for BlackBerry is available from App Store.

An iDENprotectplus token offers another layer of security. It is a separate hardware module that’s used to identify its owner when prompted by an iDENprotect App.

The iDENprotectplus can only communicate over Bluetooth, so it is unable to connect to iDENprotect server by itself. Connection to iDENprotect server is provided by the iDENprotect App. The iDENprotectplus forms a Bluetooth pairing with the iDENprotect App, which connects to the iDENprotect server over a TLS-encrypted Ethernet connection.

After forming the connection successfully, the user is prompted for identification via Touch ID or PIN code in the iDENprotect App.

iDENprotect server securely manages the whole user life cycle for all provisioned iDENprotect Devices. A
Device is always assigned to a single user - either automatically via LDAP integration or manually in the iDENprotect server Admin Console - and only Devices that are registered in the iDENprotect server database can be assigned to users.

Before the Device is ready for use, the user must enrol the Device on an iDENprotect App that sends a secure enrolment request to the iDENprotect server. Only a Device that has been enroled is able to authenticate the user.

### 2.3. Application Protection

iDENprotect can defend itself against application attacks and detect when an attack is being attempted. The mechanisms behind these controls are provided by Arxan, a leader in application protection.

With the application controls enabled, iDENprotect secures its integrity by being resistant to application tampering, malware insertion, and other types of attacks. iDENprotect applies Arxan solutions by inserting tiny code guards into the application binary to lockdown the application against attacks. These guards employ with obfuscation, encryption, cryptographic key transformation, and other techniques.

The iDENprotect application uses these Arxan Guards to secure the iDENprotect application sourcecode and the iDENprotect engine framework using Arxan GuardSpec.

The configuration of the application protection mechanisms cannot be configured, amended or altered by the administrator. The purpose of this section is to highlight the protection in place and the defined policies:-

<table>
<thead>
<tr>
<th>Application Policy</th>
<th>Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checksum (Code)</td>
<td>Yes</td>
</tr>
<tr>
<td>Checksum (Data)</td>
<td>Yes</td>
</tr>
<tr>
<td>Custom</td>
<td>Yes</td>
</tr>
<tr>
<td>Damage (Code)</td>
<td>Yes</td>
</tr>
<tr>
<td>Damage (Data)</td>
<td>Yes</td>
</tr>
<tr>
<td>Debugger Detection</td>
<td>Yes</td>
</tr>
<tr>
<td>External Symbol Renaming</td>
<td>Yes</td>
</tr>
<tr>
<td>Hook Detection</td>
<td>Yes</td>
</tr>
<tr>
<td>Jailbreak Detection</td>
<td>Yes</td>
</tr>
<tr>
<td>Obfuscation</td>
<td>Yes</td>
</tr>
<tr>
<td>Objective-C Renaming</td>
<td>Yes</td>
</tr>
<tr>
<td>Repair (Code)</td>
<td>Yes</td>
</tr>
<tr>
<td>Repair (Data)</td>
<td>Yes</td>
</tr>
<tr>
<td>Application Policy</td>
<td>Enabled</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Resource Verification</td>
<td>Yes</td>
</tr>
<tr>
<td>Root Detection</td>
<td>Yes</td>
</tr>
<tr>
<td>String Encryption</td>
<td>Yes</td>
</tr>
<tr>
<td>Swizzle Detection</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Application protection is always on.
3. Operating iDENprotect\textsuperscript{server}

iDENprotect\textsuperscript{server} is a Java application that runs as a service named idenprotect. Most of the additional components - such as the nginx web server - are also run as services.

The idenprotect is operated with the service manager using the following commands:

- on RHEL 6, service idenprotect start / stop / restart
- on RHEL 7, servicectl start / stop / restart idenprotect

To check the state of the idenprotect service, enter the following command:

- on RHEL 6, service idenprotect status
- on RHEL 7, servicectl status idenprotect

When iDENprotect\textsuperscript{server} is running normally, the status check returns Running (<process ID>)

3.1. iDENprotect\textsuperscript{server} Startup Workaround

If the service fails, iDENprotect\textsuperscript{server} can be launched with the ispa.sh startup script, which is located in the installation directory /opt/idenprotect/. It is recommended to run the script using the identear user account that was created during iDENprotect\textsuperscript{server} installation:

1. Open a terminal window
2. If not already identear, change user to identear account:

```
   su identear
```

3. Enter password for identear
4. Run the startup script:

```
   cd /opt/idenprotect
   sh ispa.sh
```

The startup script starts launching the iDENprotect\textsuperscript{server} as a background process. All server startup messages are logged in the main Logging Properties. To check the server startup status, read the last lines of the file with `tail /var/log/idenprotect.log` command.

When the server has started correctly, there is a line StartupInfoLogger - Started WebLauncher in the log file.

iDENprotect\textsuperscript{server} home directory /opt/idenprotect/ contains a script file restart.sh to quickly restart the server. Run it with the `sh restart.sh` command using the identear user account.

3.2. iDENprotect\textsuperscript{server} Shutdown Workaround

To stop iDENprotect\textsuperscript{server} after it has been started with ispa.sh script, identify which process ID it is using and then kill the process:
1. Find the iDENprotect server process ID with `ps` and `grep` commands

```
ps -ef | grep idenprotect`
```

2. The output should contain a long Java command. Identify the process ID number (bolded in the following example):

```
identear  *27575*  27574  0 Aug24 ?        00:29:55 -Xms384m -Xmx512m
-XX:MaxPermSize=256m -jar idenprotect-server-1.3.0.jar
```

3. Kill the process

```
kill -1 <process_id>
```

### 3.3. Starting and Stopping other Services

During iDENprotect server installation, the following services are also installed:

- nginx web server - service name nginx
- internal mariDB database - service name mariadb
- (optional) EJBCA CA server - service name ejbca

These services are started and stopped with the service manager using the following commands:

- on RHEL 6, `service <servicename> start / stop / restart`
- on RHEL 7, `servicectl start / stop / restart <servicename>`

For example, to restart nginx on RHEL 7, type `servicectl restart nginx`

### 3.4. Activating iDENprotect server Licence

iDENprotect server licence determines the amount of concurrent iDENprotect devices that iDENprotect server supports. Without a valid licence, iDENprotect server will not accept any iDENprotect enrolments or authentication attempts.

The licence file is supplied from your vendor, or from iDENprotect. The licence file is named `licence.xml`. Contact us at `support <at> idenprotect.com` if you have issues with the licence file.
The licence is activated with the following steps:

1. Log in to iDENprotect server Admin Console
2. In the Dashboard Tab, click on the Licences text
3. Upload your licence file

The licence file is validated and you should immediately see the licenced device pool under Device Licensing chart reflect the amount of available licences.
4. iDENprotect server Admin Console

iDENprotect server Admin Console is the web interface for managing and allocating iDENprotect Enabled Devices and iDENprotect plus to users. The Management Console runs on an embedded Tomcat application server and it is started with the same /opt/identear/ispa.sh startup script as the rest of the iDENprotect server components.

The following web browsers are supported to connect to the management console:

- Internet Explorer 10, 11
- Google Chrome 61, 62, 63
- Mozilla Firefox 55, 56, 57
- Microsoft Edge 12, 13, 14

The Management Console only accepts HTTPS connections secured by protocols specified in the Tomcat properties file. Edit the configuration file to align with your organisation's security policy. We recommend using TLS 1.2 for security reasons.

In normal iDENprotect server installation, the Management Console URL is set as the domain name of the iDENprotect server, which is usually in the format https://iden.<mydomain>.com. The default administrator user account is:

- User name: ADMIN
- Password: 1detearAdm1n

![Figure 3. iDENprotect server login screen](image)

Change the administrator login credentials after the initial iDENprotect server installation.

Using the Management Console, it is possible to:

- Add new iDENprotect plus manually to the device pool
- Assign iDENprotect Enabled Device and iDENprotect plus to users
- Remove Devices from active use
- Create and remove new Management Console user accounts
- View iDENprotect server logs
• Run iDENprotect\textsuperscript{server} reports

iDENprotect Enabled Device and iDENprotect\textsuperscript{plus} are identified in the iDENprotect\textsuperscript{server} Admin Console by their serial numbers

4.1. Management Console Views

The iDENprotect\textsuperscript{server} Admin Console interface has intuitive screens for most common usage scenarios. User and Device management is handled in quickly loaded indexed lists, and editing users is simple.

4.1.1. Dashboard Tab

The Dashboard provides an overview of the server status. It lists currently allocated Devices and login events in one view.

4.1.2. Devices Tab

The Devices tab lists all iDENprotect Devices on the iDENprotect\textsuperscript{server}. Each Device is an iDENprotect authentication entity (mobile device with iDENprotect App or physical iDENprotect\textsuperscript{plus} Token) that is registered on the server.
The Devices can be sorted and filtered based on their Users’ email addresses, device states or other attributes.

Manage firmwares and Register device pages are for registering iDENprotect plus Tokens via a DRA (Device Rights and Attributes) file upload, and updating or synchronising their firmware images.

### 4.1.3. Users Tab

The Users tab lists all users who assigned one or more iDENprotect Devices.

If the iDENprotect server has been Configuring LDAP, you can map Users and their permissions with the LDAP directory using the Sync button.

As with Devices, Users can also be sorted and filtered based on different attributes.

### 4.1.4. Logs Tab

The Logs tab lists the most recent activity on the server, read from the main iDENprotect server log file idenprotect.log. The log file location and logging parameters are defined in [Logging Properties]. By
default the file is located in /var/log/idnprotect

4.1.5. Reports Tab

The Reports tab contains a list of common printable report types for iDENprotect server. By default, reports are printed from the last 30 days, but start and end dates for the reports can be changed in the From Date and To Date fields.

The following reports are available:

- **Certificate signings** - All signed iDENprotect certificates. Certificates are signed during enrolment, and during authentication.
- **Web user management** - All events (create, edit, delete, password reset) performed for

### Certificate signings

<table>
<thead>
<tr>
<th>EVENT TYPE</th>
<th>EVENT NAME</th>
<th>TIME</th>
<th>DEVICE SERIAL NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate Management</td>
<td>Certificate Signed</td>
<td>21/02/2018 15:34:46</td>
<td>AMP002-GpUqG-017394</td>
</tr>
<tr>
<td>Certificate Management</td>
<td>Certificate Signed</td>
<td>21/02/2018 15:30:00</td>
<td>AMP002-GpUqG-017394</td>
</tr>
<tr>
<td>Certificate Management</td>
<td>Certificate Signed</td>
<td>21/02/2018 09:45:49</td>
<td>AMP002-GpUqG-317300</td>
</tr>
</tbody>
</table>

4.1.6. Site Tab

The Site tab contains management tools for the Management Console itself:

- **Manage users** - Displays and edits existing Management Console users
- **Add users** - Adds a new user
- **View certificate** - Displays the nginx web server certificate
- **About** - Displays information about iDENprotect server installation
Managing Users

The following user roles are available:

- **Administrator** - access to all features
- **Helpdesk** - same as Administrator, but can’t create new Management Console users or add new iDENprotect\textsuperscript{plus} Tokens
- **Readonly** - same as Helpdesk, but can’t change anything

Depending on the LDAP configuration, Management Console accounts might be only possible to create for users found in the LDAP directory. For more information, see **LDAP Admin Sync** in **LDAP Parameters**.

4.1.7. Config Tab

The **Config** tab contains configurable iDENprotect\textsuperscript{server} parameters in the following categories:

- **config** - General configuration parameters
- **ca** - Certificate Authority configuration
- **database** - Database connection parameters
- **email** - Email server configuration and automated message contents
- **ldap** - LDAP settings (optional)
- **logging** - Log file locations and logging intervals
- **userenrollment** - Settings for enrolment activity

All listed parameters can be configured in the web browser. The parameters map to equivalent parameters in `/etc/idenprotect/*.properties` configuration files on the iDENprotect\textsuperscript{server} file system. For descriptions of the parameters, see [Configuring Server Parameters].

Changing parameters requires restarting the iDENprotect\textsuperscript{server}. For instructions, see [Restarting iDENprotect\textsuperscript{server}].
5. iDENprotect Provisioning

iDENprotect Devices (iDENprotect Enabled Devices or iDENprotect⊕ tokens) are taken into use in a process called **provisioning**. During provisioning, the Devices are:

1. **Registered** on an organisation's iDENprotect server
2. **Assigned** to a user on the iDENprotect server
3. **Enroled** into active use by its user

Only Devices registered to an iDENprotect server and enroled by their users can be used to authenticate the user in applications that have been integrated with that iDENprotect server.

During an authentication request, iDENprotect server verifies that the enroled iDENprotect Enabled Device or iDENprotect⊕ is in possession of its assigned user. This is enforced through a biometric scan or by entering the user’s PIN code.

To get iDENprotect Enabled Devices and iDENprotect⊕ tokens enroled correctly, set up iDENprotect server so that it accepts incoming connections to HTTPS port 443. In addition, iDENprotect users need an iDENprotect App such as *iDENprotect for BlackBerry* to enrol their Devices.

The status of a Device can be tracked in the iDENprotect server Admin Console.

It’s possible to integrate iDENprotect Device user mapping with LDAP directory. With LDAP, it’s possible to restrict user registration to LDAP directory members and to automatically enrol Devices with recognized owners. Provisioning a new iDENprotect Device using LDAP auto-enrolment looks like this:

![Enrolment Process – iDENprotect SDK](image)

*Figure 8. iDENprotect Provisioning Sequence*
5.1. Registering Devices

iDENprotect Devices can be registered on the iDENprotect server by using an iDENprotect App to start the registration process. In the registration process, the iDENprotect App relays identification information from an iDENprotect Enabled Device or iDENprotect plus token to the iDENprotect server. This information is stored in a DRA (Digital Rights and Attributes) file. The contents of the DRA file are different for iDENprotect Enabled Devices and iDENprotect plus:

- In iDENprotect Enabled Devices, the DRA file for registering the iDENprotect Enabled Device is derived from the mobile device UUID (Universally Unique Identifier)
- In iDENprotect plus, the DRA is derived from the serial number that is programmed into the Token during manufacturing

Users can register both iDENprotect plus and the iDENprotect Enabled Devices with the iDENprotect App. This kind of registration is initiated by the user. When launching the iDENprotect App, the user is prompted for (the user’s own) email address and the iDENprotect server URL to connect to.

Figure 9. Starting Registration

The users must know the iDENprotect server URL before launching the iDENprotect App. It is the iDENprotect administrator’s responsibility to provide it to them.

After a successful registration, the registered Device is available in the iDENprotect server and its initial status is REGISTERED. Each Device has a unique identifier, listed in the **Serial Number** column in iDENprotect server Admin Console.

All Devices that are in REGISTERED state can be **assigned** to users.

When using [LDAP auto-enrolment], Devices are automatically assigned to their users and moved to **ASSIGNED** status. Proceed directly to Enroling Devices after this step.

5.2. Assigning Devices

If **LDAP auto-enrolment** is used, skip this step.

iDENprotect Devices and iDENprotect plus that have been registered in the iDENprotect server database can be assigned to users in the **Users** view in the iDENprotect server Admin Console.

The **Users** view lists all Devices and whether they have been assigned or enroled yet. Any Devices that
have not been assigned yet to an user, have an Assign button visible next to them.

Figure 10. Listing Users

If iDENprotect server has been integrated with Active Directory or other LDAP-based directory services, the user information can be mapped from the LDAP service. If no LDAP integration is used, the user details for assigned Devices are stored as parameters of the Device in the iDENprotect server database. Each Device can only be assigned to a single user. The internal iDENprotect server user table is not indexed, and does not keep track if the same user has multiple Device assigned to them.

To assign an Device to a user:

1. Open the Users view in iDENprotect server Admin Console
2. Find the Device to assign
3. Click Enroll button on the same row as the Device
4. Fill in and verify the user’s details
Whether using LDAP integration or an internal iDENprotect server user table, the user email field must always be populated when assigning Devices. This address is used to deliver the Activation Code or QR Code required to complete the enrolment. The Device entries are dynamically loaded and autocompleted from the database.

After finishing this step, the Device is in ASSIGNED state and is ready to be delivered to the user.

5.3. Enroling Devices

The same iDENprotect App used for registering is also used for finishing the provisioning process by allowing the user to enrol the iDENprotect App or iDENprotect plus.

Enrolment can only be done after the Device has been assigned to a user, and only the user for whom the Device has been assigned to can enrol the Device. When the user runs the App, the App requests an Activation/QR Code that has been sent to the user’s email address when the Device was assigned to the user. When using LDAP auto-enrolment, this step is done automatically.

After successfully entering the Activation Code or scanning the QR Code and confirming the Device serial number, the iDENprotect App prompts the user to set a PIN code for the iDENprotect Enabled Device or iDENprotect plus. This PIN code is used during authentication requests if biometric identification such as Touch ID is not available.

After successful enrolment, the Device is in ENROLLED state and ready to use for user authentication.
6. Configuring iDENprotect\textsuperscript{server}

There are two ways to change iDENprotect\textsuperscript{server} configuration: in iDENprotect\textsuperscript{server} Admin Console Config Tab, or on the server file system by editing .properties files in /etc/idenprotect directory. The parameters in the iDENprotect\textsuperscript{server} Admin Console Config Tab map to parameters in the .properties files. When editing a parameter in one place, the changes are reflected in the other. 

Some changes require restarting iDENprotect\textsuperscript{server}, which must be done in command line. Restart iDENprotect\textsuperscript{server} based on instructions in [Operating iDENprotect\textsuperscript{server} Services]

To edit iDENprotect\textsuperscript{server} configuration parameters in iDENprotect\textsuperscript{server} Admin Console Config Tab, open a configuration section and press Edit on the parameter you want to change. After entering a new value for the parameter, press Commit at the bottom of the configuration section.

![Figure 12. Editing Email Title](image)

If the parameter change requires iDENprotect\textsuperscript{server} restart, the text “Restart required after change” is displayed in the Config tab.

![Figure 13. Restart Required After Change](image)

Configuration Categories

<table>
<thead>
<tr>
<th>iDENprotect\textsuperscript{server} Admin Console configuration section</th>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ca</td>
<td>ca.properties</td>
<td>Certificate Authority (CA) configuration, profiles and keystores</td>
</tr>
</tbody>
</table>
### 6.1. Configuration Parameters

#### 6.1.1. CA Parameters

CA configuration parameters are defined in the `ca` configuration section in the iDENprotect server Admin Console, or in `/etc/idenprotect/ca.properties`. Some of the parameters refer to the optional EJBCA Certificate Authority. If EJBCA is not used, those parameters are ignored.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ca.name</code></td>
<td>iDENtear CA</td>
<td>Name of the CA.</td>
</tr>
<tr>
<td><code>ca.backend</code></td>
<td>ejbca</td>
<td>CA provider type. Supported types are internal and ejbca.</td>
</tr>
<tr>
<td><code>ca.backend.ejbca.trustStore</code></td>
<td>opt/ejbca/p12/truststore.jks</td>
<td>Location of the EJBCA TrustStore, which contains the CA root certificates. Password for this is originally set during iDENprotect server installation</td>
</tr>
<tr>
<td><code>ca.backend.ejbca.keyStore</code></td>
<td>/opt/ejbca/p12/superadmin.p12</td>
<td>Location of the EJBCA KeyStore, which contains the EJBCA private key &amp; certificate chain. The superadmin.p12 file is required to gain access to EJBCA Administration Console.</td>
</tr>
<tr>
<td><code>ca.backend.ejbca.keyStoreType</code></td>
<td>PKCS12</td>
<td>KeyStore type. Supported types are JKS and PKCS12.</td>
</tr>
<tr>
<td><code>ca.backend.ejbca.ws.url</code></td>
<td>https://&lt;servername&gt;:8443/ejbca/ejbcaews/ejbcaws?wsdl</td>
<td>EJBCA web services connection URL. The port number has to match the value in EJBCA web.properties file</td>
</tr>
<tr>
<td><code>ca.backend.ejbca.plugin.url</code></td>
<td>https://&lt;servername&gt;:8443/ejbca/identear</td>
<td>EJBCA network traffic URL. The port number has to match the value in EJBCA web.properties file</td>
</tr>
<tr>
<td><code>ca.backend.ejbca.end.entity.profile</code></td>
<td>iDENtear FOB</td>
<td>End Entity Profile name in EJBCA. This has to match the name set in EJBCA Administration Console.</td>
</tr>
</tbody>
</table>

---

**Table 2. ca.properties**
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ca.backend.ejbca.certificate.profile</td>
<td>ENDUSER</td>
<td>Certificate Profile name in EJBCA. This has to match the name set in EJBCA Administration Console.</td>
</tr>
<tr>
<td>ca.certificate.user.name.fmt</td>
<td>iDENtear %s</td>
<td>CA Certificate user name format definition.</td>
</tr>
<tr>
<td>ca.certificate.cn.fmt</td>
<td>CN=iDENtear</td>
<td>CA Certificate Common Name format definition.</td>
</tr>
</tbody>
</table>

### 6.1.2. Database Parameters

Database connection parameters are defined in database configuration section in iDENprotect Admin Console, or in /etc/idenprotect/database.properties.

**Table 3. database.properties**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>db.username</td>
<td></td>
<td>Database user name. For the built-in MariaDB database, the user name was defined during iDENprotect installation.</td>
</tr>
<tr>
<td>db.password</td>
<td></td>
<td>Password for the database user. For the built-in MariaDB database, the password was defined during iDENprotect installation.</td>
</tr>
<tr>
<td>db.change.log</td>
<td>classpath:db-changelog.xml</td>
<td>Log file name and location</td>
</tr>
<tr>
<td>db.vendor</td>
<td>MYSQL</td>
<td>Database type. Available options are MYSQL and ORACLE.</td>
</tr>
</tbody>
</table>

### 6.1.3. Email Parameters

Email configuration parameters are defined in email configuration section in iDENprotect Admin Console, or in /etc/idenprotect/email.properties.

**Table 4. email.properties**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>email.gateway.username</td>
<td><a href="mailto:idenprotect-admin@example.com">idenprotect-admin@example.com</a></td>
<td>Email account for communicating with registered users of the iDENprotect server.</td>
</tr>
<tr>
<td>email.gateway.password</td>
<td></td>
<td>Password for the email account</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>--------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>email.gateway.host</code></td>
<td><code>mail.example.com</code></td>
<td>URL of the email server</td>
</tr>
<tr>
<td><code>email.gateway.port</code></td>
<td><code>25</code></td>
<td>Outbound email port (25 = unsecured SMTP)</td>
</tr>
<tr>
<td><code>email.gateway.from</code></td>
<td><code>idenprotect-admin@example.com</code></td>
<td>“From” address for emails sent from the iDENprotect server account</td>
</tr>
<tr>
<td><code>email.gateway.displayname</code></td>
<td><code>idenprotect-admin@example.com</code></td>
<td>Human-readable sender name for emails sent from the iDENprotect server account</td>
</tr>
<tr>
<td><code>user.enrollment.email.subject</code></td>
<td>“iDENprotect Enrollment” / “iDENprotect PIN reset”</td>
<td>Email subject field. See Configuring Email Messages</td>
</tr>
<tr>
<td><code>user.enrollment.template</code></td>
<td><code>userEnrollmentEmail.vm</code></td>
<td>Enrollment email template contents in plaintext</td>
</tr>
<tr>
<td><code>device.enrollment.template</code></td>
<td><code>deviceEnrollmentEmail.vm</code></td>
<td>Enrollment email template contents in plaintext</td>
</tr>
<tr>
<td><code>pin.reset.enrollment.template</code></td>
<td><code>pinResetEmail.vm</code></td>
<td>Enrollment email template contents in plaintext</td>
</tr>
<tr>
<td><code>activation.code.enrollment.template</code></td>
<td><code>activationEmail.vm</code></td>
<td>Enrollment email template contents in HTML</td>
</tr>
</tbody>
</table>

### 6.1.4. LDAP Parameters

LDAP configuration parameters are defined in `ldap` configuration section in iDENprotect server Admin Console, or in `/etc/idenprotect/ldap.properties`.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ldap.enabled</code></td>
<td></td>
<td>Enables LDAP integration for iDENprotect users. Possible values are true and false.</td>
</tr>
<tr>
<td><code>ldap.type</code></td>
<td><code>real</code></td>
<td>Sets LDAP directory type for internal testing purposes. The only supported value is real</td>
</tr>
<tr>
<td><code>ldap.server</code></td>
<td><code>ldap://&lt;server-url&gt;:&lt;server-port&gt;</code></td>
<td>URL and port of the LDAP server.</td>
</tr>
<tr>
<td><code>ldap.auth.method</code></td>
<td><code>simple</code></td>
<td>LDAP authentication method. Supported types are simple, sasl and anonymous</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ldap.auth.user</td>
<td></td>
<td>LDAP management account username</td>
</tr>
<tr>
<td>ldap.auth.pass</td>
<td></td>
<td>LDAP management account password</td>
</tr>
<tr>
<td>ldap.search.base</td>
<td></td>
<td>Distinguished Name of the search base Object, which defines the location in the LDAP directory where a search begins</td>
</tr>
<tr>
<td>ldap.search.objectClass</td>
<td>person</td>
<td>The objectClass attribute to search for. Supported values are person, top, use and organizationalPerson</td>
</tr>
<tr>
<td>ldap.field.user/first/last/full.name</td>
<td>uid/givenName/sn/cn</td>
<td>ID / first name / last name / full name fields for the returned LDAP user</td>
</tr>
<tr>
<td>ldap.field.email</td>
<td></td>
<td>Email field for the returned LDAP user. During iDENprotect device registration, the entered email is used to identify the user in the LDAP directory.</td>
</tr>
<tr>
<td>ldap.user.sync</td>
<td>false</td>
<td>Automatically un-enrolls users and marks their accounts for deletion from the iDENprotect server if the user account has been disabled or is no longer found in the correct LDAP group. Possible values are true and false.</td>
</tr>
<tr>
<td>ldap.user.enforce</td>
<td>true</td>
<td>If set to true, deletes user accounts flagged by ldap.user.sync. If set to false, only logs the accounts in /var/log/identear/identear.log.</td>
</tr>
<tr>
<td>ldap.user.attribute name</td>
<td></td>
<td>Additional parameter for storing a description or address of the office where the user is located.</td>
</tr>
<tr>
<td>ldap.autoenroll.enabled</td>
<td>false</td>
<td>Enables automatic enrollment for newly registered users who can be found in the LDAP directory. Possible values are true and false.</td>
</tr>
<tr>
<td>ldap.admin.sync</td>
<td>false</td>
<td>If set to true, automatically updates user accounts that have been linked to one of the three access levels for iDENprotect server Admin Console in the LDAP directory. Possible values are true and false.</td>
</tr>
<tr>
<td>ldap.autoenroll.useRNotFound.forceCreate</td>
<td>false</td>
<td>Attempts to create a user account in the LDAP directory when the user account (as identified by the email address given by the user) has not been found during registration. Possible values are true and false.</td>
</tr>
<tr>
<td>ldap.autoenroll.group</td>
<td>CN=&lt;GroupTitle&gt;,OU=&lt;Groups&gt;,DC=&lt;organisation&gt;</td>
<td>Automatic enrollment is enabled for all LDAP users in this group</td>
</tr>
<tr>
<td>ldap.user.cron/ldap.admin.cron</td>
<td>&lt;cron sync time&gt;</td>
<td>Sets up LDAP synchronisation jobs for user and admin accounts using cron syntax. For example 0 0 * * * ? sets the synchronisation to run every hour.</td>
</tr>
<tr>
<td>ldap.admin.group/ldap.helpdesk.group/ldap.readonly.group</td>
<td>CN=&lt;GroupTitle&gt;,OU=&lt;Groups&gt;,DC=&lt;organisation&gt;</td>
<td>Access level to iDENprotect server Admin Console is granted based on LDAP user's membership in these groups.</td>
</tr>
</tbody>
</table>

### 6.1.5. Logging Parameters

Log file settings are defined in logging configuration section in iDENprotect server Admin Console, or in
Table 6. logging.properties

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>webapp.log</td>
<td>/var/log/idenprotect</td>
<td>Log file directory.</td>
</tr>
<tr>
<td>webapp.log.layout</td>
<td>org.apache.log4j.EnhancedPatternLayout</td>
<td>Log layout pattern class</td>
</tr>
<tr>
<td>webapp.log.layout.ConversionPattern</td>
<td>%d %p [%t] %C{1} - %m %throwable{2}\n</td>
<td>Logging pattern in the log file. See log4j reference for possible values.</td>
</tr>
<tr>
<td>webapp.log.maxSize</td>
<td>50mb</td>
<td>Maximum log file size in MB.</td>
</tr>
<tr>
<td>webapp.log.maxBackupIndex</td>
<td>10</td>
<td>The number of backup log files that are stored. The oldest one is overwritten when this number is exceeded.</td>
</tr>
<tr>
<td>webapp.log.DatePattern</td>
<td>'.dd'</td>
<td>File suffix format for daily logs. '.'dd appends the day of the month to the log file.</td>
</tr>
<tr>
<td>webapp.Rolling</td>
<td>org.apache.log4j.DailyRollingFileAppender</td>
<td>Log4j class for rolling files over</td>
</tr>
<tr>
<td>webapp.Console</td>
<td>org.apache.log4j.ConsoleAppender</td>
<td></td>
</tr>
</tbody>
</table>

log4j.appender.stdout.\* Definitions for appending log events to System.out

log4j.appenderlogfile.\* Definitions for appending log events to the defined log file

log4j.appender.syslog.\* Definitions for appending log events to syslog utility

log4j.logger.co.uk.applymobile.identearer DEBUG, logfile Logging level for events stored in log file

log4j.rootLogger INFO, stdout Logging level for events printed in System.out

6.1.6. Web Server Parameters

Web server configuration parameters are defined in webserver configuration section in iDENprotect server Admin Console, or in /etc/idenprotect/webserver.properties.

Table 7. webserver.properties

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>server.port</td>
<td>8081</td>
<td>Port for internal web server after reverse proxy (iDENprotect server external port is always 443)</td>
</tr>
<tr>
<td>error.whitelabel.enabled</td>
<td>false</td>
<td>Enables Spring Boot whitelabel error page if set to true. nginx default error pages are used if set to false.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------</td>
<td>----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>server.tomcat.basedir</td>
<td>/var/opt/identear</td>
<td>Tomcat server directory. The default is the same as iDENprotect server install directory.</td>
</tr>
<tr>
<td>spring.resources.addMappings</td>
<td>false</td>
<td>Enables Spring Boot default resource handling if set to true.</td>
</tr>
<tr>
<td>document.root</td>
<td>classpath:/static/</td>
<td>Tomcat document root</td>
</tr>
<tr>
<td>frontend.build</td>
<td>classpath:/build/</td>
<td></td>
</tr>
<tr>
<td>encoding.strategy</td>
<td>base64</td>
<td>Conversion method between strings and byte[]. Supported values are hex and base64.</td>
</tr>
<tr>
<td>datetime.source.strategy</td>
<td>dynamic</td>
<td>Sets time handling for TOTP (Time-based One-Time Password) generation. Supported values are dynamic and fixed</td>
</tr>
<tr>
<td>secure.random.generator.strategy</td>
<td>random</td>
<td>Sets random number generator to fixed or truly random. Supported values are fixed and random.</td>
</tr>
<tr>
<td>identity.generator.strategy</td>
<td>random</td>
<td>Sets if identifiers are fixed or truly random. Supported values are fixed and random.</td>
</tr>
<tr>
<td>show.api.docs</td>
<td>false</td>
<td>Displays REST API documentation in the web UI if set to true.</td>
</tr>
<tr>
<td>expiry.warning.days</td>
<td>30</td>
<td>Device certificate expiration warning in days.</td>
</tr>
<tr>
<td>multipart.enabled</td>
<td>true</td>
<td>Enables multipart file transfer to solve file size restrictions</td>
</tr>
<tr>
<td>multipart.location</td>
<td>/tmp</td>
<td>Storage location for temporary files during multipart transfers</td>
</tr>
<tr>
<td>multipart.max-file-size</td>
<td>2MB</td>
<td>Maximum file size for individual multipart files</td>
</tr>
<tr>
<td>multipart.max-request-size</td>
<td>2MB</td>
<td>Maximum request file size for multipart/form-data</td>
</tr>
<tr>
<td>server.dateTime.zone.id</td>
<td></td>
<td>Server time zone as Java TimeZone id</td>
</tr>
</tbody>
</table>

### 6.1.7. User Enrolment Properties

User enrolment parameters are defined in userenrollment configuration section in iDENprotect server Admin Console, or in /etc/idenprotect/userenrollment.properties.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>login.server</td>
<td>iDENprotect server hostname</td>
<td>Hostname or IP address for the server that handles enrollment requests</td>
</tr>
<tr>
<td>qr.code-callback.link</td>
<td>https://&lt;server&gt;/user/api/enrollment/session/ {wuid}?uid={userid}</td>
<td>URL for the generated QR code that is used during enrolment. This URL is included in all enrolment emails containing enrolment</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>time.to.live.minutes</td>
<td>30</td>
<td>Sets how long the QR code URL is active. After the time to live has passed, the URL expires and is no longer accessible.</td>
</tr>
</tbody>
</table>

### 6.2. Configuring Email Messages

In iDENprotect use, iDENprotect server sends email messages to the user during the following events:

**Table 9. Email Messages**

<table>
<thead>
<tr>
<th>Event</th>
<th>Template file</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>iDENprotect activation</td>
<td>ActivationEmail.vm</td>
<td>Email containing Activation Code. Automatically sent to user after successful enrollment.</td>
</tr>
<tr>
<td>iDENprotect enrolment</td>
<td>deviceEnrollmentEmail.vm and userEnrollmentEmail.vm</td>
<td>Email containing instructions on acquiring iDENprotect App and getting started with iDENprotect. Used when an iDENprotect device is set up for a user without user’s input.</td>
</tr>
<tr>
<td>PIN reset</td>
<td>pinResetEmail.vm</td>
<td>Email containing a PIN reset notification and a new Activation Code. PIN reset is initiated by the user.</td>
</tr>
</tbody>
</table>

By default, message template files are located in `/etc/idenprotect/email-templates/`. While the listed `.vm` files are plaintext files, the directory also contains HTML-formatted versions of the files (such as `ActivationEmailHtml.vm`). If a HTML version of a message can be found, iDENprotect server uses it as the email message template. If HTML version is not found, the plaintext file is used instead. Plaintext and HTML file names are configured in Email Parameters.

The email subject fields are entered in `<parameter>.email.subject` fields, which are only available in iDENprotect server Admin Console Config tab.

These emails can contain any text or HTML styling. In addition, the email templates use the following variables:

- `$user.firstName` - First name of the email recipient
- `$user.lastName` - Last name of the email recipient
- `$activationCode` - Generated one-time Activation Code used during enrolment
- `$qrUrl` - Generated enrolment QR Code image embedded within the message. Only works with HTML messages
- `$portalQrUrl` - URL link to the QR Code
7. Managing Network Ports on iDENprotect Server

The iDENprotect Server installation consists of many components that communicate via network ports. Some of these ports use default values, and others are set during installation. Generally, all ports can be changed.

Generally, iDENprotect Server uses well-established default ports for its network traffic, such as 3306 for MySQL database connections. All ports are set during installation, and all ports are configurable afterwards.

The following TCP ports must be open on the server for iDENprotect Server application to function correctly:

Inbound TCP Ports

- 22 (SSH port for terminal connections)
- 443 (HTTPS port for accessing iDENprotect Server Admin Console)
- 8443 (For accessing EJBCA administration console)

Outbound TCP Ports

- 25 (SMTP port for sending emails)
- 53 (DNS service)
- 389 (if using LDAP)
- 1521 (if using external Oracle database server)
- 3306 (if using external MySQL server)

7.1. Configuring Firewall

On RHEL, firewall ports are configured with the iptables or firewalld tools. Both can accept and reject connections to any inbound and outbound ports. For instructions on using iptables or firewalld, see Using the iptables service and Using firewalls, respectively, in RHEL 7 Linux Security Guide.

As an example, to allow inbound TCP connections on the server to port 22 (SSH), enter the following command (as root):

```
iptables -I INPUT 1 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
```
8. Managing Certificate Authorities (CAs)

The Certificate Authority (CA) used by the iDENprotect\textsuperscript{server} establishes trust within the whole iDENprotect solution. It is:

- The trust anchor of iDENprotect, when the CA is run in standalone mode
- Part of the trust chain in an environment where the iDENprotect\textsuperscript{server} CA is subordinated to an existing PKI service.

This iDENprotect CA is the central location where certificates from iDENprotect Apps and iDENprotect\textsuperscript{plus} are validated and signed to establish trust. The CA applies the certificate policy for the certificates issued by iDENprotect\textsuperscript{server} to ensure compliance with the overall PKI architecture.

iDENprotect\textsuperscript{server} can use most commonly available CA implementations, such as Microsoft CA and EJBCA. The iDENprotect\textsuperscript{server} installation package includes an EJBCA (Enterprise JavaBeans Certificate Authority) application. If the EJBCA application was selected to be installed, it is configured as the default CA in iDENprotect\textsuperscript{server}. When using another CA, install iDENprotect\textsuperscript{server} without EJBCA, and configure the CA after installation.

![Warning]

Changing CAs after iDENprotect\textsuperscript{server} is in active use is difficult. We recommend to set up the CA during initial installation without changing it afterwards.
9. Managing iDENprotect Users

An iDENprotect User is someone who has been Registered as the owner / contact person of a specific iDENprotect Device, and who has successfully Enrolled that Device on the iDENprotect server by entering an Activation Code or scanning a QR Code after the device registration has been accepted. During registration, users are mapped to an LDAP directory such as Microsoft Active Directory, or stored as owners of specific Devices in the iDENprotect server database if LDAP is not used.

LDAP integration can be automated to accept all registration attempts from users within a defined LDAP auto-enrolment group. In those cases, a user who has successfully registered a Device and who can be found from the LDAP auto-enrolment group receives the Activation/QR Code immediately. If LDAP auto-enrolment is not used, assigning Devices to Users requires manual intervention by the iDENprotect administrator even when LDAP is otherwise used.

A Device can belong only to a single user, but a user can have multiple iDENprotect Devices assigned and enrolled.

9.1. Manual User Enrolment (with or without LDAP)

In manual enrolment, users register their iDENprotect devices on the iDENprotect server. Each registered Device is stored on the iDENprotect server as a Device in REGISTERED state.

At this point and iDENprotect server administrator must assign the registered device to a user by filling in the user details manually as described in Assigning Devices, or by entering the user's LDAP username. After this step, the user is considered to be the owner of the Device, and the Device state is changed to ASSIGNED.

The user's email address is used as as their contact reference, and it must be correct in order for the user to be able to enrol the device.

When the Device is assigned to the user, the user receives an email containing the Activation/QR Code to the defined email address. By following the instructions in the email, the user can enrol the Device and finish the provisioning process.

9.2. LDAP Auto-enrolment

LDAP auto-enrolment provides an efficient way to register new iDENprotect Devices into the iDENprotect server and automatically send Activation/QR Codes to the users so they can proceed with enrolment.

LDAP auto-enrolment works by connecting users who are registering Devices automatically to a LDAP user group that has permissions to assign users on the iDENprotect server. The user has to provide an email address during registration, which is used to identify the user in the LDAP group. If the email is not found in the group, the Device can be registered but will not be automatically assigned to the user. In this case, a iDENprotect server administrator must intervene to correctly assign the device.

When the user submits the registration, iDENprotect server receives the user's email address and the DRA (Digital Registration Attributes) file of the iDENprotect Enabled Device or iDENprotect plus. If the email account owned by the user exists in LDAP group, iDENprotect server completes the registration by auto-populating the user's personal details and assigns the Device to the user.
10. Configuring LDAP

LDAP directories store user information such as name and email address, which are used to authenticate the users of the LDAP system. When LDAP is enabled in iDENprotect server configuration, iDENprotect server uses the configured LDAP directory to verify the user information submitted during the Device registration process by mapping it with existing LDAP records in a LDAP directory.

iDENprotect server supports the following LDAP systems:

- Microsoft Active Directory 2008 R2
- Microsoft Active Directory 2012
- Microsoft Active Directory 2012 R2
- Microsoft Active Directory 2016

The iDENprotect server LDAP plugin only needs read access to the LDAP directory. Configure the service account allocated for iDENprotect server as read-only. This access allows the lookup of users and users within a specific group that have been approved to be enrolled by the iDENprotect server.

LDAP settings are defined during iDENprotect server installation and stored in the LDAP configuration file LDAP Parameters. When the settings are configured properly, iDENprotect server can perform LDAP tasks such as searching for user information based on first name or surname.

ldap.properties is loaded again every time iDENprotect server is restarted. Therefore all changes to LDAP configuration require a restart.

10.1. Connecting to LDAP server

LDAP server connection is defined in the LDAP configuration file LDAP Parameters. To set up the connection, edit the following values in the file, or in the Config panel in iDENprotect server Admin Console:

- `ldap.enabled` - Set to `true` to enable LDAP
- `ldap.type` - Set to `real`. Other values are only for internal testing purposes
- `ldap.server` - Enter LDAP server IP and port to connect to, in the format `ldap://<server-ip>:<server-port>`
- `ldap.auth.method` - Enter authentication method. Available options are `simple`, `sasl` and `anonymous`. If unsure of the right option, select `simple`.
- `ldap.auth.user` - Enter the LDAP service account username (if using simple or SASL authentication)
- `ldap.auth.pass` - Enter the LDAP service account password (if using simple or SASL authentication)

The connection parameters must match the LDAP server’s settings.

10.2. Configuring LDAP Server

Setting up Search Parameters

Once the connection is established, set the user search parameters for the LDAP query in LDAP Parameters, or in the Config panel in iDENprotect server Admin Console:
• ldap.search.base - Set the Distinguished Name of the search base where the LDAP search begins
• ldap.search.objectClass - Set the objectclass of the search base target
• ldap.field.user.name / first.name / last.name / full.name / email - Set the column names for user information
• ldap.user.attributename - Additional user identification parameter such as physicalDeliveryOfficeName

Setting up Service Account

iDENprotect server needs to have read access to the LDAP server to be able to query the LDAP directory. We recommend creating a dedicated LDAP account for this purpose. The account only requires membership in the Domain Users group.

For example, to create the service account in Active Directory 2016:

1. Open Active Directory Users and Groups
2. Navigate to the Users folder
3. Right-click on the window and select New → User
4. Enter the user account details

Use the same username and password as in LDAP parameters ldap.auth.user and ldap.auth.pass.

Setting up LDAP Auto-enrolment

To streamline the iDENprotect enrollment process, enable auto-enrolment for newly registered Devices.

• ldap.autoenroll.enabled - Set to true to enable auto-enrolment
• ldap.autoenroll.userNotFound.forceCreate - Set to true to automatically create user accounts in the LDAP directory if the email address given during registration is not found in the LDAP directory. For this purpose, the service account must also be a member of Domain Admins group.
• ldap.autoenroll.group - Enter the auto-enrolment Group name in the LDAP directory. All members in this group are automatically enrolled when they register a new Device. The Group syntax is CN=<GroupName>,OU=GroupFolder,DC=<LDAPDirectoryName>.

Setting up LDAP Synchronisation
The system can be set to periodically synchronise the registered users in iDENprotect server and the users in the LDAP directory. This synchronisation removes any user accounts from iDENprotect server that:

- Are removed from the LDAP directory
- Are set to Disabled state in the LDAP directory
- Are removed from the auto-enrolment Group (if auto-enrolment has been enabled)

Any account that matches one or more of these criteria is removed from iDENprotect server. The synchronisation job can be set to run with desired intervals using a cron command. The following settings configure LDAP synchronisation:

- **ldap.user.sync** - Set to true to enable synchronisation
- **ldap.user.enforce** - Set to true to immediately delete the account from iDENprotect server. Setting to false only flags the account in /var/log/idenprotect/identear.log. When an account is flagged, the log contains a line DefaultUserService - <username@domain.com> Not a member of users group for that user.

iDENprotect server Admin Console can be synchronised with access with LDAP directory. When this option is enabled, users can login to iDENprotect server Admin Console with their LDAP account.

- **ldap.admin.sync** - Set to true to automatically update user accounts that have been linked to one of the three access levels in Management Console.
- **ldap.admin/helpdesk/readonly.group** - Group name containing users with Administrator / Helpdesk / Readonly privileges

Both user synchronisation and Management Console synchronisation are set to run on specified intervals using the ldap.user.cron and ldap.admin.cron parameters. Both parameters are entered using **standard cron syntax**.
11. Configuring iDENprotect server Database

The default database system in iDENprotect server is MariaDB, a MySQL-compatible relational database management system. It is possible to use other database systems such as Oracle Database with some additional post-install configuration.

11.1. Configuring MariaDB

During iDENprotect server installation, the installer creates a MariaDB database which contains data for iDENprotect internal use. In addition, it has empty tables for Devices, Users and Certificates, all of which are required for proper iDENprotect authentication use. All Device and certificate data generated during iDENprotect use is stored in these tables.

MariaDB database can be set up on the same server running iDENprotect server application, or on a different server. MariaDB uses very little system resources, so we recommend installing it on the same server as iDENprotect server.

Using MariaDB database requires no further configuration after installation. If a different database has been configured up at some point, change the database connection parameters in /etc/idenprotect/database.properties to start using MariaDB:

1. Set parameter db.jdbc.driver to com.mysql.jdbc.Driver
3. Set parameter db.vendor to MYSQL

11.2. Configuring Oracle Database

iDENprotect server supports Oracle Database version 11g Release 2 (11.2.0.x) Enterprise and Express editions. Due to the high system resource usage of Oracle Database, we don’t recommend running Oracle Database and iDENprotect server on the same server.

Oracle Database is not automatically configured during iDENprotect server installation. To use Oracle Database instead of MariaDB, first change the database connection parameters in /etc/idenprotect/database.properties:

1. Set parameter db.username to the schema / user name of the iDENprotect database
2. Set parameter db.password to the user’s password
3. Set parameter db.jdbc.driver to oracle.jdbc.OracleDriver
4. Set parameter db.jdbc.url to jdbc:<database-SID>:thin:@<server-ip>:<port>/<global-database-name>
5. Set parameter db.vendor to ORACLE

iDENprotect server Oracle Database connector requires that an empty schema for the data is created before launching iDENprotect server successfully. During startup, iDENprotect server creates all tables and required core data when it detects that the schema is empty.

There are multiple ways to create a new user / schema in Oracle Database. For instructions, see
12. Logging

iDENprotect server logs activity in the following log files:

<table>
<thead>
<tr>
<th>Log File</th>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>iDENprotect server log</td>
<td>/var/log/idenprotect.log (location defined in [Logging Properties])</td>
<td>Stores authentication request connections, device enrolments and other connection attempts from devices enrolled on the iDENprotect server. Also contains EJBCA logs (if EJBCA component is in use).</td>
</tr>
<tr>
<td>EJBCA logs (if EJBCA</td>
<td>/var/log/ejbca.log and /var/log/ejbca.err</td>
<td>Stores all processed authentication requests handled by EJBCA, and whether they were successful or not.</td>
</tr>
<tr>
<td>Web server logs</td>
<td>/var/log/nginx/access.log and /var/log/nginx_error.log</td>
<td>Stores information on web server activity, mostly relating to the iDENprotect server Admin Console.</td>
</tr>
<tr>
<td>Install logs</td>
<td>/tmp/ispa_log/*.log</td>
<td>Installation logs for iDENprotect server components.</td>
</tr>
</tbody>
</table>

idenprotect.log and all other log files are plaintext files accessible to all text reader programs. To read and process a file, use Linux utilities such as cat, tail, and grep. For example, to see the contents of the iDENprotect server log file, type cat /var/log/idenprotect.log

12.1. iDENprotect server Log Events

<table>
<thead>
<tr>
<th>Description</th>
<th>Class Name</th>
<th>Contents / Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication event</td>
<td>DefaultCryptoService</td>
<td>Hash data, signature &amp; certificate</td>
</tr>
<tr>
<td>iDENprotect registration</td>
<td>DevicesController</td>
<td>Device registered &lt;serial&gt;</td>
</tr>
<tr>
<td>iDENprotect enrolment - phase 1 (initiation)</td>
<td>DefaultEnrollmentService</td>
<td>Successful enrollment identification for device: Device{id, serial, state, isLocked, hasUser}</td>
</tr>
<tr>
<td>iDENprotect enrolment - phase 2 (certificate signing)</td>
<td>InternalCA</td>
<td>Signing CSR with profile ENDUSER</td>
</tr>
<tr>
<td>iDENprotect enrolment - phase 3 (completion)</td>
<td>EnrollmentController</td>
<td>Successful enrollment completion</td>
</tr>
<tr>
<td>iDENprotect signature verification</td>
<td>DefaultSigningService &amp; SignatureValidationController</td>
<td>Sucessfully verified signature: &lt;serial&gt;</td>
</tr>
<tr>
<td>Registration email</td>
<td>DefaultEmailGatewayService</td>
<td>Sending email send Message to: &lt;address&gt;, subject: &lt;subject&gt;</td>
</tr>
<tr>
<td>LDAP User synchronisation</td>
<td>DefaultLdapService</td>
<td>Sync Job Complete</td>
</tr>
<tr>
<td>iDENprotect server Admin Console login</td>
<td>CsrfAuthenticationSuccessHandler</td>
<td>&lt;user&gt; Successfully Authenticated to Admin Console from &lt;IP address&gt;</td>
</tr>
</tbody>
</table>
12.2. Log Search Examples

Finding Authentication Events

A typical authentication event is identified by DefaultDeviceService and DefaultCryptoService text strings, and the serialNumber of the Device (AMP002-Dw0sOAiY-448495 in the following example):

```
2017-11-01 15:57:24,174 DEBUG [http-nio-8081-exec-4] DefaultDeviceService - Found Device{id=1701, serialNumber=AMP002-Dw0sOAiY-448495, state=ENROLLED, isLocked=false}
2017-11-01 15:57:24,183 DEBUG [http-nio-8081-exec-4] DefaultCryptoService - Verify signature using: input data: 414d503030322d447730734f4169592d343438343935, signature: 3046022100c8a2252a1b2a7f6ccc0a22df891a383fe1ca2bfc8875949d6086ee8dd14b832e022100b2ad347a658cc7516e18baf9a2fa364e11c5e1364d168d20a42ea7edafb0d87, certificate: -----BEGIN CERTIFICATE-----
```

Finding Registration Events

Device registration and enrollment events are logged in /var/log/identear/identear.log. A typical registration event is identified by DefaultRegistrationService text string and the serial number of the Device (AMP002-J6diTlXe-420732 in the following examples):

```
2017-11-02 07:49:13,661 DEBUG [http-nio-8081-exec-10] DefaultRegistrationService - Found ProductLine{id=3, deviceLine=AMP, revision=002, isSoft=true}
2017-11-02 07:49:13,700 DEBUG [http-nio-8081-exec-10] DefaultRegistrationService - Saved Device{id=179, serialNumber=AMP002-J6diTlXe-420732, state=REGISTERED, isLocked=false}
2017-11-02 07:49:13,721 DEBUG [http-nio-8081-exec-10] DefaultRegistrationService - Prepared data for co.uk.applymobile.identear.server.user.User@312bc6f[id=<null>,firstName=,lastName=,middleName=<null>,emailAddress=(OMITTED),radiusUsername=<null>,adAttribute=<null>,activeDirectoryUsername=<null>,device=Device{id=179, serialNumber=AMP002-J6diTlXe-420732, state=REGISTERED, isLocked=false}]
```

Finding Assignment Events

A typical assignment event is identified by DefaultUserService text string and the serial number of the Device:
In the above example, the user is automatically assigned with LDAP auto-enrollment

Finding Enrolment Events

A typical enrolment event is identified by DefaultEnrollmentService. The enrolment process is more complex and includes multiple steps involving TOTP (Time-based One-Time Password) generation, signature verification and CA interaction.
13. Managing Web Server

iDENprotect server uses nginx as the web engine for running iDENprotect server Admin Console and [EJBCA Web Console (optional)]. During installation, nginx is set to only accept connections to ports 443 (for iDENprotect server Admin Console) and 8443 (for EJBCA Web Console) using TLS 1.2 or later protocol.

Generally there is need to change the nginx configuration after it has set up. However, the nginx installation does not include a signed SSL certificate. This chapter contains instructions on creating and signing a trusted SSL certificate, but purchasing and maintaining the certificate is the responsibility of the customer organisation.

13.1. Installing a Trusted SSL Certificate

The terms SSL and TLS are sometimes used interchangeably in networking. Even though the iDENprotect server is configured to only accept TLS connections, the certificates used to secure traffic are by convention referred to as SSL certificates.

In order to handle web connections securely, the iDENprotect server must use a valid SSL certificate. If no valid certificate has been configured, iDENprotect server rejects all web connections.

13.1.1. Generating Private Key and CSR

1. In the iDENprotect server terminal, create a new 2048-bit RSA key using OpenSSL:

   ```bash
   openssl req -new -newkey rsa:2048 -nodes -keyout server.key -out server.csr
   ```

   Change the name of `server.key` and `server.csr` to the hostname of iDENprotect server

   This starts the process of generating 2 files, a private key file for decrypting SSL/TLS traffic and a CSR (Certificate Signing Request) file

   1. Enter the organisational and geographic information for the certificate
   2. When prompted for the Common Name, enter the fully qualified domain name of the iDENprotect server. For example, iden.domain.com.
   3. Enter an email address for contact information regarding the certificate

13.1.2. Signing the CSR

Once the CSR file is created, send it to the CA (Certificate Authority) provider, such as VeriSign. The reply from the CA typically contains the signed certificate chain in a .pem or .crt file. This file is the public key of the iDENprotect server. Store it on the iDENprotect server computer.

   When copy-pasting the CSR file contents, make sure to include all of them. Many CSRs fail because the BEGIN and END lines were not included in the request.
13.1.3. Installing the Certificates

To have a functional SSL setup for iDENprotect server, both keys have to be stored on the server and configured in nginx:

1. Copy the .key file and the CRT or PEM received from the CA in the /var/certs directory
2. Edit /etc/nginx/nginx.conf and add the locations for the keys ssl_certificate and ssl_certificate_key

```plaintext
ssl_certificate: /var/certs/my_domain_name.pem; (or bundle.crt)
ssl_certificate_key: /var/certs/my_domain_name.key;
```

Adjust the file names to match the certificate files.

3. Restart the nginx web server:

`systemctl restart nginx`

13.1.4. Testing the Certificate

Open a web browser and enter the iDENprotect server URL. The browser should not alert to any certificate validation errors.

Depending on the browser, a valid SSL certificate is indicated by a padlock or similar icon in the address bar, which means that iDENprotect server now has a trusted certificate for the web site.

![Certificate Information](image)

*Figure 14. Certificate information*
13.2. Configuring nginx

Configuration settings for nginx are stored in /etc/nginx/nginx.conf. To change the settings, open nginx.conf in a text editor.

The following parameters are important for running iDENprotect server:

Table 12. nginx parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>server_name</td>
<td>localhost</td>
<td>Hostname of the server</td>
</tr>
<tr>
<td>listen</td>
<td>443</td>
<td>Ports that the server listens to</td>
</tr>
<tr>
<td>ssl_protocols</td>
<td>TLSv1, TLSv1.1, TLSv1.2</td>
<td>Allowed SSL/TLS protocols</td>
</tr>
<tr>
<td>ssl_certificate</td>
<td></td>
<td>Location of the web server's public key</td>
</tr>
<tr>
<td>ssl_certificate_key</td>
<td></td>
<td>Location of the web server’s private key</td>
</tr>
<tr>
<td>ssl_ciphers</td>
<td>See Cipher Suites in iDENprotect server Admin Console</td>
<td>List of supported cipher suites</td>
</tr>
</tbody>
</table>
Appendix A: Using Built-in EJBCA Certificate Authority

EJBCA is a Public Key Infrastructure (PKI) Certificate Authority (CA) application developed by PrimeKey Solutions AB.

EJBCA is set as the iDENprotect server Certificate Authority by default during installation. The certificate database and passwords are also set during installation.

When using EJBCA with iDENprotect server, it is not mandatory to perform any additional configuration on the EJBCA application. The iDENprotect solution functions well using the default settings.


A.1. Installing EJBCA Administrator Certificate

Access to the console is secured with the P12 certificate created during installation. The certificate is protected with a password that also was set during installation (parameter \\#17, "administrator P12 keystore").

During iDENprotect server installation, the certificate is generated in the iDENprotect server file system in location opt/identear.ejbca/p12/superadmin.p12. To access the EJBCA web console, import the certificate into the web browser (if using Firefox) or operating system (if using any other browser). See the instructions for Firefox and for Chrome on Linux.

A.2. Using EJBCA Web Console

EJBCA web console main view shows the publicly available resources of EJBCA, such as public CA certificates, and check the status of individual issued certificates.

To access the administration console, click Administration in the left-side menu panel. The administration console contains options to configure iDENprotect server certificate handling on a low level.

The most important parts to configure in the EJBCA administration console are:

- Certificate Authorities, which issue certificates used in iDENprotect
- Certificate Profiles (especially the ENDUSER profile), which define how the iDENprotect Devices are handled in iDENprotect server
- End Entity Profiles, which define the CA and Certificate Profile to use when signing device-specific certificates (called End Entities in EJBCA)
The End Entities listed in EJBCA map to listed Devices in the iDENprotect server Admin Console.

For full information on configuring EJBCA settings with the administration console, see EJBCA Administrator Guide.

**A.2.1. Creating New iDENprotect Certificate Profile**

1. Click **Certificate Profiles**.
2. Enter a name for the new Certificate Profile in the empty row at the bottom of the list, and click **Add**.
3. Click **Edit** on the new profile.
4. Set the following parameters:
   a. Type: Sub CA
   b. Signature algorithm: SHA256withECDSA
   c. Permissions: Allow all, EXCEPT Allow subject DN override by End Entity Information
   d. Extended Key Usage: Use
   e. Subject Alternative Name: Use
   f. Authority Information Access: Use
   g. Available CAs: Select only one
5. Click **Save**
### A.2.2. Creating End Entity Profile

1. Click End Entity Profiles
2. Enter a name for the new End Entity Profile in the empty row at Add Profile section, and click Add
3. Add Subject DN Attributes:
   a. Required: Common Name (CN)
b. Optional: Organisation (O), Organisational Unit (OU), Locality (L), State or Province (ST), Country (C)

c. Set all attributes as Modifiable

4. Set the previously created Certificate Profile as the Default Certificate Profile

5. Select required profiles as Available Certificate Profiles

6. Set iDENtear CA as the Default CA

**Edit End Entity Profile**

End Entity Profile: iDENtearTokenCA

---

**Directives**

Reverse Subject DN and Subject Alt Name Choice (?)

Allow merge DN Web Services (?)

---

**Subject DN Attributes (?)**

Select for Removal

- [ ] Add

Subject DN Attributes: emailAddress, E-mail address in DN

- [ ] CN, Common name
- [ ] O, Organization
- [ ] OU, Organizational Unit
- [ ] L, Locality
- [ ] ST, State or Province
- [ ] C, Country (ISO 3166)

---

**Other subject attributes**

Select for Removal

- [ ] RFC 822 Name (e-mail address)

---

*Figure 18. Setting End Entity Profile*
A.3. Configuring EJBCA Startup

EJBCA (Enterprise JavaBeans Certificate Authority) is installed and configured during iDENprotect server installation. Usually there is no need to change these settings after install, but if the internal listening ports in the iDENprotect server application, corresponding changes in EJBCA configuration must be made as well.

To change the configuration later, edit the `/opt.ejbca/conf/web.properties` configuration file. The following parameters are important:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.trustpassword</td>
<td></td>
<td>Password for the Java trustStore</td>
</tr>
<tr>
<td>superadmin.trustpassword</td>
<td></td>
<td>Password for the internal administrator PKCS12 keystore</td>
</tr>
<tr>
<td>httpsserver.trustpassword</td>
<td></td>
<td>Password for the SSL certificate used by the EJBCA administrator console</td>
</tr>
<tr>
<td>httpsserver.hostname</td>
<td></td>
<td>Full DNS hostname of the EJBCA server that functions as the Certificate Authority for iSPA</td>
</tr>
<tr>
<td>httpsserver.dn</td>
<td>CN=${httpsserver.hostname},O=EJBCA Sample,C=SE</td>
<td>Distinguished Name of the SSL certificate used by the EJBCA administrator console</td>
</tr>
<tr>
<td>httpsserver.privhttps</td>
<td>8443</td>
<td>Port number for the EJBCA server. This has to match the <code>ca.backend.ejbca.ws.url</code> parameter value in <code>ca.properties</code> CA configuration file.</td>
</tr>
</tbody>
</table>

After changing EJBCA settings, re-run the Apache Ant command as ejbca user. The ejbca user account and its password were created during iDENprotect server installation.

1. Change user to ejbca:

   ```
su ejbca
   ```

2. Deploy new EJBCA application to the iDENprotect server:

   ```
ant clean deploy
   ```

3. Run the web configuration script:

   ```
ant web-configure
   ```

4. Restart the EJBCA service:

   ```
service ejbca restart
   ```
Appendix B: Cipher Suites in iDENprotect Server Admin Console

These are the cipher suites supported in the nginx web server running iDENprotect Server Admin Console. The cipher suites can be changed in the `/etc/nginx/nginx.conf` configuration file by editing the `ssl_ciphers` parameter.

- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-RSA-AES128-GCM-SHA256
- DHE-RSA-AES256-GCM-SHA384
- DHE-RSA-AES128-GCM-SHA256
- ECDHE-RSA-AES256-SHA384
- ECDHE-RSA-AES128-SHA256
- ECDHE-RSA-AES256-SHA
- ECDHE-RSA-AES128-SHA
- DHE-RSA-AES256-SHA256
- DHE-RSA-AES128-SHA256
- DHE-RSA-AES256-SHA
- DHE-RSA-AES128-SHA
- ECDHE-RSA-DES-CBC3-SHA
- EDH-RSA-DES-CBC3-SHA
- AES256-GCM-SHA384
- AES128-GCM-SHA256
- AES256-SHA256
- AES128-SHA256
- AES256-SHA
- AES128-SHA
Appendix C: iDENprotect Device Keystore

iDENprotect Enabled Devices and iDENprotect\textsuperscript{plus} have a secure keystore with many security measures designed to protect keys and their usage, and to ensure that keys cannot be tampered with or removed from the device.

iDENprotect Enabled Device have an unlimited amount of slots available for key storage. iDENprotect\textsuperscript{plus} are limited to 11 slots.

Table 14. Keystore Slot Number Table

<table>
<thead>
<tr>
<th>Slot Number</th>
<th>Key</th>
<th>Key Type</th>
<th>Key Usage</th>
<th>Key Generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slot 0</td>
<td>&lt;System reserved&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slot 1</td>
<td>Device Key</td>
<td>ECDH / ECDSA</td>
<td>Encrypting and decrypting Bluetooth communications, and decrypting iDENprotect\textsuperscript{server} asset transfers</td>
<td>Generated during initial startup by device RNG. Signed during enrolment by iDENprotect\textsuperscript{server}.</td>
</tr>
<tr>
<td>Slot 2</td>
<td>iDENprotect\textsuperscript{server} Certificate</td>
<td>ECDH / ECDSA</td>
<td>Downloaded during enrolment</td>
<td></td>
</tr>
<tr>
<td>Slot 3</td>
<td>Signing Key</td>
<td>ECDSA at device’s secure hardware</td>
<td>Local signing key</td>
<td>Created during enrolment</td>
</tr>
<tr>
<td>Slot 4</td>
<td>&lt;Spare&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slot 5</td>
<td>TOTP (Time-based One-Time Password) Key</td>
<td>AES-256</td>
<td>Used in SSO applications</td>
<td>Created during enrolment</td>
</tr>
<tr>
<td>Slot 6</td>
<td>&lt;Spare&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slot 7</td>
<td>MDM (or MAM) Encryption Key from MDM Provider (such as Good Dynamics)</td>
<td>Depends on provider</td>
<td>Management Container encryption</td>
<td>Created at MDM initial setup</td>
</tr>
</tbody>
</table>

Appendix D: Database Schema

Table 15. DEVICES Table

<table>
<thead>
<tr>
<th>Description</th>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Key</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>bigint(20)</td>
<td>NO</td>
<td>PRI</td>
<td>NULL</td>
<td>auto_incr</td>
<td></td>
</tr>
<tr>
<td>SERIAL_NUMBER</td>
<td>varchar(64)</td>
<td>NO</td>
<td>UNI</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACTIVATION_CODE</td>
<td>varchar(16)</td>
<td>NO</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CERTIFICATE</td>
<td>varchar(4096)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Field</td>
<td>Type</td>
<td>Null</td>
<td>Key</td>
<td>Default</td>
<td>Extra</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------------------------------</td>
<td>---------------</td>
<td>------</td>
<td>-----</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>DEVICE_STATE_ID</td>
<td>bigint(20)</td>
<td>NO</td>
<td>MUL</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAST_SUCCESSFUL_OTP_VALUE</td>
<td>bigint(20)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAST_SUCCESSFUL_OTP_DATE</td>
<td>timestamp</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS_LOCKED</td>
<td>bit(1)</td>
<td>NO</td>
<td>b'0'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OTP_KEY</td>
<td>blob</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXPIRY_DATE</td>
<td>timestamp</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOGIN_SUCCESS_COUNT</td>
<td>bigint(20)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOGIN_FAILURE_COUNT</td>
<td>bigint(20)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATTERY_CHANGE</td>
<td>int(11)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USAGE_SECONDS</td>
<td>bigint(20)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIN_RESET_FLAG</td>
<td>bit(1)</td>
<td>NO</td>
<td>b'0'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIN_CHANGE_FLAG</td>
<td>bit(1)</td>
<td>NO</td>
<td>b'0'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OTP_FAILURE_COUNT</td>
<td>bigint(20)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SW_VERSION</td>
<td>varchar(32)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRODUCT_LINE_ID</td>
<td>bigint(20)</td>
<td>NO</td>
<td>MUL</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 16. DEVICES_EXTRA_CERTIFICATES Table*
<table>
<thead>
<tr>
<th>Description</th>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Key</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAG</td>
<td>varchar(32)</td>
<td>NO</td>
<td>MUL</td>
<td>NULL</td>
<td></td>
<td>auto_increment</td>
</tr>
<tr>
<td>CERTIFICATE</td>
<td>varchar(4096)</td>
<td>NO</td>
<td>NULL</td>
<td></td>
<td></td>
<td>auto_increment</td>
</tr>
<tr>
<td>DEVICE_ID</td>
<td>bigint(20)</td>
<td>NO</td>
<td>MUL</td>
<td>NULL</td>
<td></td>
<td>auto_increment</td>
</tr>
<tr>
<td>ID</td>
<td>int(11)</td>
<td>NO</td>
<td>PRI</td>
<td>NULL</td>
<td></td>
<td>auto_increment</td>
</tr>
</tbody>
</table>

Table 17. SERVER_CREDENTIALS Table

<table>
<thead>
<tr>
<th>Description</th>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Key</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>bigint(20)</td>
<td>NO</td>
<td>PRI</td>
<td>NULL</td>
<td></td>
<td>auto_increment</td>
</tr>
<tr>
<td>CREDENTIALS_TYPE</td>
<td>int(11)</td>
<td>NO</td>
<td>UNI</td>
<td>NULL</td>
<td></td>
<td>auto_increment</td>
</tr>
<tr>
<td>CERTIFICATE</td>
<td>blob</td>
<td>NO</td>
<td>NULL</td>
<td></td>
<td></td>
<td>auto_increment</td>
</tr>
<tr>
<td>PRIVATE_KEY</td>
<td>blob</td>
<td>NO</td>
<td>NULL</td>
<td></td>
<td></td>
<td>auto_increment</td>
</tr>
<tr>
<td>COMMON_NAME</td>
<td>varchar(255)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td>auto_increment</td>
</tr>
<tr>
<td>ORGANIZATION</td>
<td>varchar(255)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td>auto_increment</td>
</tr>
<tr>
<td>ORGANIZATION</td>
<td>varchar(255)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td>auto_increment</td>
</tr>
<tr>
<td>LOCALITY</td>
<td>varchar(255)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td>auto_increment</td>
</tr>
<tr>
<td>STATE</td>
<td>varchar(255)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td>auto_increment</td>
</tr>
<tr>
<td>COUNTRY</td>
<td>varchar(255)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td></td>
<td>auto_increment</td>
</tr>
</tbody>
</table>