DATASUNRISE
DATABASE SECURITY SUITE 3.7.7
ADMINISTRATION GUIDE (LINUX, UNIX)
## CONTENTS

### Convention .................................................................................................................................................. 5

### General Information .................................................................................................................................. 6
1.1 Product Description ........................................................................................................................................ 6
1.2 Supported databases ...................................................................................................................................... 6
1.3 DataSunrise Operation Modes ........................................................................................................................ 7
1.3.1 Sniffer Mode ................................................................................................................................................ 7
1.3.2 Proxy Mode ................................................................................................................................................ 8
1.4 System requirements ....................................................................................................................................... 8
1.5 Useful resources ............................................................................................................................................. 8

### Deployment topologies ................................................................................................................................. 10
2.1 Installing DataSunrise on a database server ................................................................................................. 10
2.1.1 Proxy Mode ................................................................................................................................................ 10
2.1.2 Sniffer Mode .............................................................................................................................................. 11
2.2 Installing DataSunrise on a separate server ............................................................................................... 11
2.2.1 Proxy mode ................................................................................................................................................ 11
2.2.2 Sniffer mode .............................................................................................................................................. 11

### DataSunrise installation and removal ........................................................................................................ 13
3.1 Required components ..................................................................................................................................... 13
3.2 Program installation ....................................................................................................................................... 15
3.3 Program removal .......................................................................................................................................... 16
3.4 DataSunrise installation folder ..................................................................................................................... 16
3.5 Updating DataSunrise ................................................................................................................................... 17
3.6 Migrating DataSunrise to Another server .................................................................................................... 18
3.7 Rollback ....................................................................................................................................................... 19

### Starting DataSunrise for the first time ........................................................................................................ 20
4.1 Starting DataSunrise ....................................................................................................................................... 20
4.2 Connecting to DataSunrise web interface ................................................................................................. 20
4.2.1 Changing administrator’s password ........................................................................................................ 21
4.2.2 Creating a certificate for the UI ............................................................................................................... 21
4.3 Product registration ..................................................................................................................................... 21
4.4 Creating a database profile on startup (optional) ....................................................................................... 21
4.5 Preparing your database and creating database users required for getting database metadata ............ 23
4.5.1 Creating an Oracle database user .......................................................................................................... 23
4.5.2 Creating a PostgreSQL user ................................................................................................................... 24
4.5.3 Creating a Netezza database user ........................................................................................................... 25
4.5.4 Creating a MySQL/MariaDB database user .......................................................................................... 25
4.5.5 Creating a Greenplum user .................................................................................................................... 25
4.5.6 Creating a Teradata database user ........................................................................................................ 26
4.5.7 Creating a SAP HANA database user .................................................................................................... 26
4.5.8 Creating a Redshift database user .......................................................................................................... 26
4.5.9 Creating a Vertica database user .......................................................................................................... 26
4.5.10 Granting necessary privileges to a DB2 user .................................................................................... 27
4.5.11 Configuring MS SQL Server connection ............................................................................................ 27
4.5.12 Troubleshooting Connection Failure .................................................................................................. 31
4.6 Additional proxy configuration .................................................................................................................. 31
4.6.1 Changing PostgreSQL port number .................................................................................................. 31
## Convention

The following table describes text conventions.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bold text</strong></td>
<td>UI interface active elements (buttons, drop-down lists, links etc)</td>
<td>Click <strong>Save</strong> button</td>
</tr>
<tr>
<td><strong>Italic text</strong></td>
<td>Variables and file names, names of text fields etc</td>
<td>Replace the <em>Appfirewall.pem</em> certificate</td>
</tr>
<tr>
<td>→</td>
<td>An arrow which shows sequence of actions</td>
<td>Go to the <strong>System settings —&gt; About</strong> subsection</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>Useful information</td>
<td><strong>Note:</strong> you can access the user guide by clicking <strong>Help</strong> link</td>
</tr>
<tr>
<td><strong>Important:</strong></td>
<td>Important information</td>
<td><strong>Important:</strong> use &quot;admin&quot; user name on first startup</td>
</tr>
<tr>
<td><strong>code block</strong></td>
<td>Various types of code (SQL code, CLI commands etc.)</td>
<td><code>connect -host localhost -port -54321 -password 123456 -login admin</code></td>
</tr>
<tr>
<td><strong>Link</strong></td>
<td>Links (X-references or hyper links)</td>
<td><strong>Refer to Convention</strong></td>
</tr>
</tbody>
</table>

**Note:** feel free to provide feedback, suggestions and comments so we can improve the documentation. Contact us at [info@datasunrise.com](mailto:info@datasunrise.com)
1.1 PRODUCT DESCRIPTION

DataSunrise Database Security is an application firewall purpose-built to protect relational databases against hacker attacks and insider-driven threats. DataSunrise is compatible with Windows and Linux operating systems, runs fast and independently of any applications and doesn't inflict any unnecessary load on database server.

DataSunrise can complete the following tasks:

- **Data Auditing.** DataSunrise logs all user actions, SQL queries and query results. DataSunrise's Data Audit saves information on database users, user sessions, query code etc. Data auditing results can be exported to an external system, such as SIEM.

- **Data Protection.** DataSunrise's database firewall analyzes database traffic, detects and prevents execution of unauthorized queries and SQL injections on-the-fly. Alerts and reports on detected threats can be sent to network administrators via email or instant messengers.

- **Data Masking.** DataSunrise prevents sensitive data exposure due to its dynamic masking capability. DataSunrise obfuscates output of sensitive data from a database by replacing it with random data or real-looking data. DataSunrise's masking includes both Dynamic data masking and Static data masking. A variety of masking algorithms for any possible scenario is offered.

- **Data Discovery.** DataSunrise scans databases, locates personally identifiable information (PII), electronic protected health information (ePHI) and creates a security or a masking rule for the columns with sensitive data.

1.2 SUPPORTED DATABASES

DataSunrise is compatible with the following databases:

- Oracle Database 9.2-12.1 working on Windows, Linux, Solaris (SPARC) or IBM AIX servers
- PostgreSQL 7.4-10
- Netezza 6.0-7.2.1
- Greenplum 4.2-4.3
- IBM DB2 9.7-11.1
- MS SQL Server 2005-2016
- Amazon Aurora
- Amazon Redshift
- MariaDB 5.1-10.2
- MySQL 5.0-5.7
- Teradata 13-15
- Hive 1.0-2.1
- Vertica 7.0-8.1
- SAP HANA Database 1.0-2.0
1.3 DATASUNRISE OPERATION MODES

DataSunrise can be deployed in one of the following configurations: **Sniffer mode** or **Proxy mode**.

### 1.3.1 SNIFER MODE

When deployed in the Sniffer mode, DataSunrise is connected to a SPAN port of a network switch. Thus it acts as a traffic analyzer capable to capture copy of the database traffic from network switch’s “mirrored” port.

![Sniffer mode operation scheme.](image)

In this configuration, DataSunrise can be used only for “passive security” ("active security" means such as database firewall or masking are not supported in this mode). When deployed in the Sniffer mode, DataSunrise is capable to perform database activity monitoring (Data Audit) only, because it can’t modify database traffic. Running DataSunrise in the Sniffer mode does not require any additional reconfiguring of databases or client applications. Sniffer mode can be used for data auditing purpose or for running DataSunrise in the Learning mode (see subs. 6.1.2 of the DataSunrise User Guide).

**Important:** database traffic should not be encrypted. Check your database settings as some databases encrypt traffic by default. If you’re operating SQL Server database, do not use ephemeral ciphers. DataSunrise deployed in the sniffer mode does not support connections redirecting to a random port (like Oracle). All network interfaces (the main and the one database is redirecting to) should be added to DataSunrise.
1.3.2 Proxy Mode

When deployed in this configuration, DataSunrise works as an intermediary between a database server and its client applications. Thus it is able to process all incoming queries before redirecting them to a database server.

![Proxy mode operation scheme.](image)

Proxy mode is for "active protection". DataSunrise intercepts SQL queries sent to a protected database by database users, checks if they comply with existing security policies, and audits, blocks or modifies the incoming queries or query results if necessary. When running in the Proxy mode, DataSunrise sports its full functionality: database activity monitoring, database firewall, both dynamic and static data masking are available.

**Important:** We recommend to use DataSunrise in the proxy mode. It provides full protection and in this mode, DataSunrise supports processing of encrypted traffic and redirect connections (it is essential for Hana, Oracle, Vertica, MS SQL).

1.4 System Requirements

Before installing DataSunrise, make sure your server meets the following requirements:

- Operating system: 64-bit Linux or UNIX
- CPU: 8 cores
- RAM: 8-16 Gb
- Available disk space: 3GB. 100 GB for storing audit records if necessary

1.5 Useful Resources

- DataSunrise official web site: [https://www.datasunrise.com/](https://www.datasunrise.com/)
- DataSunrise latest version download page: [https://www.datasunrise.com/download](https://www.datasunrise.com/download)
- DataSunrise Administration Guide for Linux ([DataSunrise_Database_Security_Suite_Admin_Guide_Linux.pdf](https://www.datasunrise.com/download) file located in the doc subfolder of the program installation directory). Describes installation and post-installation procedures, deployment schemes, includes troubleshooting subsection.
- DataSunrise User Guide ([DataSunrise_Database_Security_Suite_User_Guide.pdf](https://www.datasunrise.com/download) file located in the doc subfolder). Describes GUI structure, program managing, etc.
• Command Line Interface Guide (CLI_guide.pdf file in the doc subfolder). Contains CLI commands description, use cases, etc.
• Release Notes (Release_notes.pdf file in the doc subfolder). Describes changes and enhancements made in the latest DataSunrise version, known bugs and version history.
• EULA (DataSunrise_EULA.pdf file in the doc subfolder). Contains End User License Agreement.
Deployment topologies

DataSunrise can be installed either on the database server or on a separate server. In both cases, the software can be used both in the Sniffer mode and in the Proxy mode.

2.1 INSTALLING DATASUNRISE ON A DATABASE SERVER

Figure 3: Deployment on DB server

2.1.1 PROXY MODE

To deploy DataSunrise in the Proxy mode, use one of the following methods:

A) TWEAKING OF DATABASE SETTINGS

- Configure DataSunrise to use the port which the database uses to connect to the client applications
- Change the database's port number (because its old port is occupied by DataSunrise now)
- Configure connection between DataSunrise and the database considering changes made at the previous steps.

All aforementioned steps are not relevant to Teradata and Vertica. Vertica and Teradata use a default port that cannot be changed. If you are going to use DataSunrise proxy for a single database, this would work. But if another Vertica or Teradata database exists, the default port cannot be used again because it is already busy. A proxy to another database should be opened on another port and database clients should be reconfigured.

Tip: You can use the installation option described above during firewall testing, but some DB clients would still retain direct access to the DB. Use a system firewall (Windows Firewall or Iptables for Linux for example) to block direct access to the DB.

Important: Many operating systems reserve port numbers less than 1024 for privileged system processes. That’s why it’s preferable to use port numbers higher than 1024 to establish proxy-connection.

B) RECONFIGURING OF CLIENT APPLICATIONS

- Make sure that DataSunrise uses the same port number that the database does
- Configure all client applications to connect with DataSunrise instead of the database
2.1.2 SNIFER MODE
It is not required to tweak any client applications or database settings.

2.2 INSTALLING DATASUNRISE ON A SEPARATE SERVER

2.2.1 PROXY MODE

Figure 4: Proxy mode deployment scheme

To deploy DataSunrise in the Proxy mode, perform the following:

• Configure connection between DataSunrise and the database.
• Configure all the client applications to connect to the DataSunrise’s proxy instead of the database.

Important: Many operating systems reserve port numbers less than 1024 for privileged system processes, so it’s preferable to use port numbers higher than 1024.

2.2.2 SNIFER MODE

Figure 5: Sniffer mode deployment scheme
To deploy DataSunrise in the Sniffer mode, configure your network switch for transferring mirrored traffic to DataSunrise (refer to your network switch's user guide for description of port mirroring procedure).
Note: Before you begin the DataSunrise installation process, select appropriate deployment option (subsections 2.1 and 2.2) and perform all required preparations. Also make sure that the machine where you want to install DataSunrise meets system requirements listed in subsection 1.4.

3.1 REQUIRED COMPONENTS

Depending on DataSunrise deployment variant and RDBMS used, it could be necessary to install some additional components.

1. **Important:** All environment variables should be added to:
   - `/etc/datasunrise.conf`
   
   It is not recommended to change:
   - `/opt/datasunrise/start_firewall.sh`

   Install unixODBC:
   
   To install it from repository, execute the following command (on Ubuntu):
   ```bash
   sudo apt-get install unixODBC
   ```
   
   or the following command (on Red Hat):
   ```bash
   sudo yum install unixODBC
   ```
   

2. To run DataSunrise with MySQL and PostgreSQL databases, install ODBC driver. You can download it here: [http://www.postgresql.org/ftp/odbc/versions/](http://www.postgresql.org/ftp/odbc/versions/)


   **Note:** Having installed the Oracle Instant Client, add its home directory path to the $ORACLE_HOME environment variable and to the $PATH variable. Or you can add the required path to the `/etc/datasunrise.conf` file. Example:
   ```bash
   bash -c "echo \"ORACLE_HOME=/opt/instantclient_12_1/\" >> /etc/datasunrise.conf"
   ```
   
   Then create a symbolic link for the required `libclntsh.so` library (libclntsh.so.12.1 or libclntsh.so.11.3 for example — it depends on Oracle version):
   ```bash
   cd /opt/instantclient_12_1
   sudo ln -s libclntsh.so.12.1 libclntsh.so
   ```

4. To run DataSunrise with Netezza database, install dedicated ODBC driver. Download it from IBM Fix Central website:
Note: Your IBM ID should be associated with your IBM customer ID with active support and maintenance contract for Netezza appliance.

Refer to the following page for more details: https://www-304.ibm.com/support/knowledgecenter/SSULQD_7.0.3/com.ibm.nz.adm.doc/c_sysadm_client_software_packages.html

5. To run DataSunrise with DB2 databases, install ODBC driver: https://www-304.ibm.com/support/docview.wss?uid=swg21418043
7. To run DataSunrise with Hive, install Hortonworks ODBC driver: https://hortonworks.com/downloads/
8. To run DataSunrise with HP Vertica databases, install ODBC client drivers: https://my.vertica.com/download/vertica/client-drivers/

Note: You will have to log in to your Vertica community account or create it if you don’t have one. Having downloaded the required drivers, log in to the system as root and perform the following:

- Create a /opt/vertica/ folder

  mkdir -p /opt/vertica/  # create the folder /opt/vertica/

- Copy an archive with drivers to the folder

  cp vertica_x.x..xx_odbc_64_linux.tar.gz /opt/vertica/

- Change the directory to /opt/vertica/

  cd /opt/vertica/$

- Uncompress the file

  tar vzxvf vertica_x.x..xx_odbc_64_linux.tar.gz

- Open the file that contains ODBC settings

  /etc/odbcinst.ini

- In the open *.ini file, change the directory of the Vertica ODBC driver as follows:

  [Vertica]
  Description = Vertica_ODBC_Driver
  Driver64 = /opt/vertica/opt/vertica/lib64/libverticaodbc.so
  ErrorMessagesPath=/opt/vertica

- If you want to set some specific settings for your Vertica client, create /etc/vertica.ini file and paste the following code into it:

  [Driver]
  DriverManagerEncoding=UTF-16
  ErrorMessagesPath=/opt/vertica
  LogLevel=4
  LogPath=/tmp


10. To connect to IBM DB2 databases, install IBM Data Server Client Package.
• Download drivers from the official website. You will have to log in to your IBM account or create a new one.
  http://www-01.ibm.com/support/docview.wss?rs=4020&uid=swg27016878
• Unzip the downloaded file.
  
  ```
  tar -xfv ibm_data_server_driver_for_odbc_cli_linuxx64_v11.1.tar.gz
  ```
• Open the idbcinst.ini file with the text editor and add the directory to the libdb2o.so file as shown in the example below.
  
  ```
  [IBM DB2 ODBC DRIVER]
  Description=DB2 Driver
  Driver=/home/user/сlidriver/lib/libdb2o.so
  FileUsage=1
  DontDLClose=1
  ```

### 3.2 Program installation

To install DataSunrise, perform the following:

1. Use the following command to give execution permission for the DataSunrise installation file:

  ```
  sudo chmod +x DataSunrise_Suite_XXX.linux.64bit.run
  ```

2. Start installation by executing the following command:

  ```
  sudo ./DataSunrise_Suite_XXX.linux.64bit.run install
  ```

**Note:** on some Linux distributives (CentOS 6.8 at least), you might need to specify a temporary folder for unzipping the installation archive:

  ```
  sudo DataSunrise_Suite_XXX.linux.64bit.run --target ~/ds
  ```

You can use some additional parameters to customize the installation process:

<table>
<thead>
<tr>
<th>CLI parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f</td>
<td>Perform quick installation using default settings.</td>
</tr>
<tr>
<td>--no-password</td>
<td>Don’t generate a password for GUI at the end of the installation process (set the password at GUI after installation).</td>
</tr>
<tr>
<td>--extract-only</td>
<td>Extract DataSunrise distributive into specified folder without installation (in this case start DataSunrise manually from the installation folder).</td>
</tr>
</tbody>
</table>

For example:

  ```
  sudo ./DataSunrise_Suite_XXX.linux.64bit.run install -f --no-password
  ```

3. Specify program installation folder in the *Target directory* line if necessary.

**Note:** DataSunrise is installed into the *opt/datasunrise* folder by default.
4. If necessary, replace DataSunrise's SSL certificate with a new one (refer to 4.2.2).

3.3 PROGRAM REMOVAL

To uninstall DataSunrise, perform the following:

Initiate program removal by executing the following command via the Linux CLI:

```
sudo ./DataSunrise_Suite_X_XX.linux.64bit.run remove
```

3.4 DATASUNRISE INSTALLATION FOLDER

This subsection describes DataSunrise files and structure of the installation folder.

Figure 7: DataSunrise files and folders

1. DataSunrise folders:
### DataSunrise installation and removal

<table>
<thead>
<tr>
<th>Folder name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmdline</td>
<td>Contains DataSunrise Command Line Interface (CLI) files</td>
</tr>
<tr>
<td>gwt</td>
<td>Contains the GUI files</td>
</tr>
<tr>
<td>logs</td>
<td>Log files (Back end, Core, GUI logs)</td>
</tr>
</tbody>
</table>

### 2. DataSunrise files:

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppBackendService</td>
<td>System process required for operation of the GUI and for control of AppFirewallCore.exe</td>
</tr>
<tr>
<td>appfirewall.pem</td>
<td>SSL certificate for the GUI</td>
</tr>
<tr>
<td>AppFirewallCore</td>
<td>Program’s Core</td>
</tr>
<tr>
<td>audit.db</td>
<td>SQLite database file to store audit data (the Audit Storage)</td>
</tr>
<tr>
<td>cacert.pem</td>
<td>SSL certificate required for online update</td>
</tr>
<tr>
<td>dictionary.db</td>
<td>Contains program settings, DataSunrise-specific objects such as database profiles, user profiles, rules, etc.</td>
</tr>
<tr>
<td>event.db</td>
<td>System events logs</td>
</tr>
<tr>
<td>libcrypto.so.10</td>
<td>OpenSSL library</td>
</tr>
<tr>
<td>libssl.so.10</td>
<td>OpenSSL library</td>
</tr>
<tr>
<td>proxy.pem</td>
<td>OpenSSL keys and certificates used for proxies by default</td>
</tr>
<tr>
<td>standart_application_queries.db</td>
<td>Contains queries used by Oracle SQL Developer (refer to Query Groups subsection for more information)</td>
</tr>
<tr>
<td>start_firewall.sh</td>
<td>Script that starts datasunrise system service</td>
</tr>
<tr>
<td>stop_firewall.sh</td>
<td>Script that stops datasunrise system service</td>
</tr>
</tbody>
</table>

### 3.5 UPDATING DATASUNRISE

To update DataSunrise, perform the following:

1. Go to System Settings → About.
2. Click Update.
3. Wait for the update to complete and reload the GUI page.

**Note:** You can also update the program in another way. Download the latest version of DataSunrise from the official web site and run the installation file using the `update` command:

```
chmod +x ./DataSunrise_Suite_X_X_X.linux.64bit.run
sudo ./DataSunrise_Suite_X_X_X.linux.64bit.run update
```

**Note:** During the update process, the installer creates a backup folder (`/opt/datasunrise/backup/`) where all important files are saved.

In case the update procedure fails and you need to return to the previous state, all the backed up files should be manually copied to the main DataSunrise directory. It will roll back to a previous version.

The update procedure would take between 5 to 7 seconds. During that period the DataSunrise service can be unavailable.

### 3.6 MIGRATING DATASUNRISE TO ANOTHER SERVER

To export DataSunrise settings to other instance installed on other server, perform the following:

1. Stop DataSunrise service (refer to **Subs. 4.1**).
2. Copy the `dictionary.db`, `event.db` and `audit.db` files from the source DataSunrise installation folder.
3. Install a new DataSunrise instance on another server. Stop DataSunrise system service.
4. Paste the `dictionary.db`, `event.db` and `audit.db` files into the new DataSunrise instance installation folder.
5. Start the DataSunrise system service.
6. Check imported settings.
3.7 ROLLBACK

At every DataSunrise update the installer creates a backup folder with all important files from the previous version. To restore previous version of DataSunrise, perform the following:

1. Run DataSunrise installer file with the *rollback* parameter:

```
    sudo ./DataSunrise.run rollback
```

   You will get a list of all available backups.

2. Select number of the required backup. Note that the higher number, the newer the backup.
4.1 STARTING DATASUNRISE

DataSunrise needs the datasunrise service running to operate. This service starts DataSunrise’s back end and core automatically.

1. You can start the DataSunrise service manually by executing the following command via Linux CLI:

   ```
   sudo service datasunrise start
   ```

   **Note:** To stop the DataSunrise service, execute the following command:

   ```
   sudo service datasunrise stop
   ```

2. Enter the DataSunrise web interface (refer to subs. 4.2).

4.2 CONNECTING TO DATASUNRISE WEB INTERFACE

DataSunrise features a comprehensive web-based interface used to control all the program’s actions.

1. To enter the web interface, perform the following:
   - To connect to the GUI using HTTPS protocol (by default), open the following address via your web browser:

     ```
     https://<DataSunrise_ip_address>:11000
     ```

   **Note:** `<DataSunrise_ip_address>` is DataSunrise’s IP address or host name, **11000** is the DataSunrise’s port number. For example, if your DataSunrise is installed on your local PC, the address would be the following:

     ```
     https://localhost:11000
     ```

   **Important:** If you’ve configured DataSunrise to use Active Directory users (refer to subs. 6.4), use port 11000. To log in as a default admin user, use port 12000.

   - If you want to connect to DataSunrise using HTTP protocol, you should activate HTTP in system settings (System Settings → General → Ports). Then open the following address via your web browser:

     ```
     http://DataSunrise_ip_address:11000
     ```

   **Note:** `<DataSunrise_ip_address>` is an IP address or a name of the host where DataSunrise is installed, **11000** is the DataSunrise’s port number.

2. You browser will display "Unsecure connection" prompt because of untrusted SSL certificate. Follow your browser’s prompts to confirm security exception for DataSunrise’s GUI.

3. Enter your credentials and click **Login** to enter the web interface.
**Important:** On first startup, use `admin` as a login and the password you’ve received at the end of the program installation.

### 4.2.1 Changing Administrator’s Password

You cannot restore DataSunrise administrator password if you’ve lost it, but you can set a new one. To change `admin` user’s password, perform the following:

1. Start the Linux CLI.
2. Use `cd` command to go to the DataSunrise installation folder (`opt/datasunrise` by default).
3. Run the `AppBackendService` file with the `set_admin_password` parameter. Specify a new password as the parameter’s value:

   ```bash
   sudo ./AppBackendService set_admin_password=new_password
   ```

4. Restart DataSunrise system service for the changes to take effect.

### 4.2.2 Creating a Certificate for the UI

On the first startup, a web browser used to access DataSunrise GUI warns about unsecure connection and prompts to add a security exception for the GUI. This issue is caused by DataSunrise’s self-signed SSL certificate. To avoid this, you can use a signed SSL certificate from a certain certification authority. For example, you can get such certificate for free from Let’s Encrypt service.

1. Refer to the following link for a guide on obtaining a certificate from Let’s Encrypt: [https://www.datasunrise.com/blog/getting-an-ssl-certificate-with Lets-Encrypt/](https://www.datasunrise.com/blog/getting-an-ssl-certificate-with Lets-Encrypt/)
2. Paste the private key and the certificate you got from Let’s Encrypt into the `appfirewall.pem` file and move the file to the DataSunrise installation folder.

### 4.3 Product Registration

The first time you start DataSunrise, you will be prompted to register it.

1. Paste a license key you got from the DataSunrise developers into the dedicated text field.
2. Click **Save**.

### 4.4 Creating a Database Profile on Startup (Optional)

At first startup, you are proposed to create a target database profile. You can skip this step to perform it later. Before establishing protection of a certain database, you should specify this database in DataSunrise settings. To do this, you need to create a target database profile. The profile includes connection details which enable DataSunrise to get database metadata. To create a new database profile, perform the following:

1. Enter information about the target database in the **New Database Configuration** subsection:

<table>
<thead>
<tr>
<th><strong>UI element</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logical Name</strong> text field</td>
<td>Logical name of the database instance</td>
</tr>
<tr>
<td><strong>Database Type</strong> drop-down list</td>
<td>Target DB type</td>
</tr>
<tr>
<td><strong>Hostname/IP</strong> text field</td>
<td>Target DB address (host name or IP address)</td>
</tr>
</tbody>
</table>
Starting DataSunrise for the first time

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port text field</td>
<td>DB port number</td>
</tr>
<tr>
<td>User Type drop-down list</td>
<td>User authentication type (regular login/password or Active Directory user authentication)</td>
</tr>
<tr>
<td>IP Version drop-down list</td>
<td>IP protocol version to use for database connection:</td>
</tr>
<tr>
<td></td>
<td>• Auto: define automatically</td>
</tr>
<tr>
<td></td>
<td>• IPv 4</td>
</tr>
<tr>
<td></td>
<td>• IPv 6</td>
</tr>
<tr>
<td>Instance text field (for Oracle database only)</td>
<td>Oracle service name or SID</td>
</tr>
<tr>
<td>Database text field (for all DB types except Oracle and MySQL)</td>
<td>Name of target database. Required to get database metadata.</td>
</tr>
<tr>
<td>Default Login text field</td>
<td>DB user name which DataSunrise should use to connect to the database</td>
</tr>
<tr>
<td>Password text field</td>
<td>DB user password that DataSunrise should use to connect to the database</td>
</tr>
<tr>
<td>Important: DataSunrise uses credentials only to get database metadata.</td>
<td></td>
</tr>
<tr>
<td>Encryption drop-down list (for Oracle only)</td>
<td>Encryption method:</td>
</tr>
<tr>
<td></td>
<td>• No: no encryption</td>
</tr>
<tr>
<td></td>
<td>• SSL</td>
</tr>
<tr>
<td>Instance Type drop-down list (for Oracle only)</td>
<td>Method that DataSunrise should use to connect to the database:</td>
</tr>
<tr>
<td></td>
<td>• SID: using SID</td>
</tr>
<tr>
<td></td>
<td>• Service Name: using the Oracle service name</td>
</tr>
<tr>
<td>Save the Password check box</td>
<td>Save database user password in DataSunrise</td>
</tr>
<tr>
<td>Key Group drop-down list</td>
<td>SSL key group (SSL certificates)</td>
</tr>
<tr>
<td>Use a Custom Connection String check box</td>
<td>Enable Custom Connection String (see below)</td>
</tr>
<tr>
<td>Custom Connection String field</td>
<td>Specify a custom connection string to connect to a database</td>
</tr>
</tbody>
</table>

**Important:** ODBC connection string should be used for all databases except Oracle, MySQL-based (MySQL, MariaDB, Aurora) and PostgreSQL-based (PostgreSQL, Greenplum, Redshift) databases. For PG-based databases LibPQ driver is used by default, and for MySQL-based MySQL Connector is used by default. To switch drivers for MySQL and PostgreSQL to ODBC, disable the `MySQLConnectorEnable` and `LibPQEnable` options in the Additional Settings (refer to subs. 4.1.3 of the DataSunrise User Guide. For examples of connection strings refer to the following web site: [https://www.connectionstrings.com/](https://www.connectionstrings.com/)

2. Click the **Test** button to check connection between the database and the DataSunrise server.
3. Specify the method of interaction between DataSunrise and the target DB in the **Open Proxy or Sniffer for Database** subsection (either a proxy or a sniffer):

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server</strong> drop-down list</td>
<td>Select DataSunrise server (DataSunrise Instance) on which to open a proxy or a sniffer (refer to subs. 4.4)</td>
</tr>
</tbody>
</table>
| **Action** drop-down list | Select operating mode DataSunrise should employ to process requests to a target DB (refer to subs. 1.3):  
  • Proxy  
  • Sniffer |
| **Network Adapter** drop-down list (for Sniffer mode only) | Select a network controller which DataSunrise should use to connect to the target DB |
| **Host** drop-down list (for Proxy mode only) | IP address of the proxy |
| **Port** text field (for Proxy mode only) | Number of a network port DataSunrise should be listening to |

4. Click **Save** to save the target DB profile.

---

**4.5 PREPARING YOUR DATABASE AND CREATING DATABASE USERS REQUIRED FOR GETTING DATABASE METADATA**

DataSunrise interacts with a target database and receives its metadata required for operation by connecting through a user account of a target database (the account, username and password of which are specified in the target database profile). It is possible to use any user account with sufficient privileges.

This section includes instructions for establishing connection between DataSunrise and various databases and instruction for creating database users with privileges to get database metadata.

**4.5.1 CREATING AN ORACLE DATABASE USER**

1. Connect to the Oracle target database using the SYS user account.
2. To create a new user, perform the following.

   • For Oracle 11 g Release 2 or earlier, execute the following command:

     ```sql
     CREATE USER <User name> IDENTIFIED BY <Password>;
     ```

   • For Oracle 12 c, create a global user (for all Oracle containers). Connect to `CDB$ROOT` and execute the following command:

     ```sql
     CREATE USER c##<User name> IDENTIFIED BY <Password>;
     ```

     You can also create a local user (for one container). To do this, execute the following commands:

     ```sql
     ALTER SESSION SET CONTAINER = pdborcl;
     CREATE USER <User name> IDENTIFIED BY <Password>;
     ```

   **Warning:** In most cases, it is preferable to use a global user for connecting to target databases, because if you use a local user (created for one container), DataSunrise would not be able to work with other containers.
3. Grant all required privileges to the new user if necessary. To do this, execute the following commands:

- For Oracle Database or Amazon RDS Oracle.
  
  On the source database:
  
  ```
  GRANT SELECT_CATALOG_ROLE TO <User name>;
  ```
  
  On the target database:
  
  ```
  GRANT CREATE ANY TABLE TO <User name>;
  ```
  
  **Important:** You can also use the following method to grant DataSunrise access to certain tables and views.

- For Oracle 11g Release 2 or earlier:
  
  ```
  GRANT CONNECT TO <User name>;
  GRANT SELECT on "SYS"."DBA_OBJECTS" TO <User name>;
  GRANT SELECT on "SYS"."DBA_TAB_COLUMNS" TO <User name>;
  GRANT SELECT on "SYS"."DBA_SYNONYMS" TO <User name>;
  GRANT SELECT on "SYS"."DBA_NESTED_TABLES" TO <User name>;
  GRANT SELECT on "SYS"."V_$SERVICES" TO <User name>;
  GRANT SELECT on "SYS"."V_$INSTANCE" TO <User name>;
  GRANT SELECT on "SYS"."DBA_USERS" TO <User name>;
  GRANT SELECT on "SYS"."DBA_PROCEDURES" TO <User name>;
  ```

- For Oracle 12c. For a local user:
  
  ```
  GRANT SELECT on "SYS"."V_$SERVICES" to <User name>;
  GRANT SELECT on "SYS"."CDB_USERS" to <User name>;
  GRANT SELECT on "SYS"."CDB_OBJECTS" to <User name>;
  GRANT SELECT on "SYS"."CDB_TAB_COLUMNS" to <User name>;
  GRANT SELECT on "SYS"."CDB_SYNONYMS" to <User name>;
  GRANT SELECT on "SYS"."CDB_NESTED_TABLES" to <User name>;
  GRANT SELECT on "SYS"."V_$INSTANCE" to <User name>;
  GRANT SELECT on "SYS"."CDB_PROCEDURES" to <User name>;
  GRANT CREATE TABLE to <User name>;
  ```

  **Tip:** You can create a required table manually instead of giving the CREATE TABLE privilege to a new user:

  ```
  create global temporary table DAF_OBJECTS ON COMMIT DELETE ROWS as select * from CDB_OBJECTS where 1 != 1;
  ```

  To grant required privileges to a global user, execute the following commands:

  ```
  GRANT CONNECT to c##<User name> CONTAINER=ALL;
  GRANT SYSDBA to c##<User name>;
  ```

4.5.2 CREATING A POSTGRESQL USER

To create a PostgreSQL user, execute the following command:

```
CREATE USER <User name> WITH PASSWORD '<Password>';
```
• pg_class
• pg_catalog
• pg_attribute
• pg_user
• pg_settings
• pg_db_role_setting

To do this, execute the following query:

```sql
GRANT SELECT ON pg_catalog.pg_database, pg_catalog.pg_namespace, pg_catalog.pg_class, pg_catalog.pg_attribute, pg_catalog.pg_user, pg_catalog.pg_settings, pg_catalog.pg_db_role_setting TO dsuser
```

4.5.3 CREATING A NETEZZA DATABASE USER

To create a new Netezza user, execute the following command:

```sql
CREATE USER <User name> WITH PASSWORD '<Password>';
```

**Note:** Grant all required privileges to the new user. Connect to the SYSTEM database and send it an appropriate SQL query:

- For Netezza 6.X:

  ```sql
  GRANT LIST ON AGGREGATE, DATABASE, EXTERNAL TABLE, FUNCTION, GROUP, MANAGEMENT TABLE, MANAGEMENT VIEW, PROCEDURE, SEQUENCE, SYNONYM, SYSTEM TABLE, SYSTEM VIEW, TABLE, USER, VIEW to <User_name>;
  ```

- For Netezza 7.X:

  ```sql
  GRANT LIST ON AGGREGATE, DATABASE, EXTERNAL TABLE, FUNCTION, GROUP, MANAGEMENT TABLE, MANAGEMENT VIEW, PROCEDURE, SCHEMA, SEQUENCE, SYNONYM, SYSTEM TABLE, SYSTEM VIEW, TABLE, USER, VIEW to <User_name>;
  ```

4.5.4 CREATING A MYSQL/MARIADB DATABASE USER

To create a new MySQL/MariaDB user, execute the following command:

```sql
CREATE USER <User name> IDENTIFIED BY '<Password>';
```

**Note:** Grant required privileges to the new user by executing the following query:

```sql
GRANT SELECT ON *.* TO '<User name>'; FLUSH PRIVILEGES;
```

4.5.5 CREATING A GREENPLUM USER

To create a new Greenplum user, execute the following command:

```sql
CREATE USER <User name> WITH PASSWORD '<Password>';
```

Execute the following query to provide the user with necessary privileges:
4.5.6 CREATING A TERADATA DATABASE USER

To create a new Teradata user, run the following command:

```sql
CREATE USER "<User Name>"
AS
PERM = 0
PASSWORD = "<Password>";
```

**Note:** Grant required privileges to the user by executing the following query:

```sql
GRANT SELECT ON ALL TABLES IN SCHEMA <Schema name> TO <User name>;
```

4.5.7 CREATING A SAP HANA DATABASE USER

To create a new SAP HANA user, execute the following command:

```sql
CREATE USER <User name> PASSWORD '<Password>'; 
```

**Note:** Providing required privileges includes two stages: first a role should receive privileges to access schema’s objects, and then the role is assigned to a user. To grant required privileges, execute the following queries:

```sql
GRANT SELECT ON SCHEMA <Schema name> TO <Role name>;
GRANT <Role name> TO <User name>;
```

4.5.8 CREATING A REDSHIFT DATABASE USER

To create a new Redshift user, execute the following command:

```sql
CREATE USER <User name> PASSWORD '<Password>'; 
```

**Note:** Grant required privileges to the new user by executing the following query:

```sql
alter default privileges in schema <Schema name> grant select on tables to <User name>;
```

4.5.9 CREATING A VERTICA DATABASE USER

1. To create a new Vertica user, execute the following command:

```sql
CREATE USER <User name> IDENTIFIED BY '<Password>'; 
```

2. Define an authentication type you will use:

```sql
GRANT AUTHENTICATION <Authentication method name> TO <User name>;
```
Starting DataSunrise for the first time

3. Grant USAGE for all schemas:
   
   ```
   GRANT USAGE ON SCHEMA <Schema name> TO <User name>;
   ```

4. Grant SELECT for all schemas:
   
   ```
   GRANT SELECT ON ALL TABLES IN SCHEMA <Schema name> TO <User name>;
   ```

**Note:** You can execute the following queries that generate queries from GRANTS for all required schemas:

```sql
SELECT 'GRANT USAGE ON SCHEMA ' || Schema name || ' to <User name>;' from v_catalog.schemata WHERE is_system_schema = false;

SELECT 'GRANT SELECT ON ALL TABLES IN SCHEMA ' || Schema name || ' to <User name>;' from v_catalog.schemata WHERE is_system_schema = false;
```

### 4.5.10 Granting Necessary Privileges to a DB2 User

To make DataSunrise work correctly with a DB2 database, it is necessary to provide a user with privileges to select data from the following system views:

- `syscat.schemata`
- `syscat.procedures`
- `syscat.functions`
- `syscat.tables`
- `syscat.columns`
- `syscat.sequences`
- `syscat.packages`

To grant necessary user privileges, run the following script. This query returns a statement for each schema table (DB2 can’t provide rights for accessing a whole schema). Copy this code to DB2’s SQL Editor and run as a single script.

```sql
SELECT DISTINCT
  'GRANT Select ON TABLE ' || rtrim (tabschema) || ' ' || rtrim (tabname) || ' TO USER <User name>;' FROM syscat.tables
WHERE tabschema = '<Schema name>'
UNION
SELECT    'GRANT Select ON VIEW ' || rtrim (VIEWSCHEMA) || ' ' || rtrim (viewname) || ' TO USER <User name>;' FROM SYSCAT.VIEWS
WHERE VIEWSCHEMA = '<Schema name>';  
```

### 4.5.11 Configuring MS SQL Server Connection

To establish connection between DataSunrise and a SQL Server database, perform the following:

1. Run **SQL Server configuration manager** utility (it is included in SQL Server pack). Open **SQL Server Network Configuration → Protocols for (DB instance name)**
2. Right-click on **TCP/IP** protocol name and select **Properties** in the context menu
Starting DataSunrise for the first time

3. In the **TCP/IP Properties** window, **Protocol** tab, set **Yes** value for **Enabled** parameter. Then open **IP-addresses** tab, **IPA11** subsection and set **TCP-port** parameter value to **1433**. Click **OK** to close the window.

4. Open **SQL Server Services** subsection, right-click on **SQL Server (DB instance name)** parameter to open its context menu, and click **Restart**.

5. If you're using a firewall application (including Windows Firewall) of some kind, you should allow the following inbound connections: TCP/IP, port 1433 and UDP, port 1434.

6. When configuring is done, it is recommended to restart your PC.

7. Connect to the database server with SQL Server Management Studio (SSMS).

**Important:** SSMS’s **Encrypt connection** option forces encryption and server certificate check on client’s side (except SSMS 2016 and higher). Thus when this option is enabled, the client would not be able to connect to DataSunrise **proxy** if the certificate included into **proxy.pem** or **dictionary.db** does not include proxy’s host name. In the case when encryption is enabled (it is disabled by default), it is necessary to have a properly signed SSL certificate. Otherwise, disable **Encrypt connection**.

**Important:** use SQL Server authentication instead of Windows authentication. When configuring database connection, specify the database server’s host name or IP address instead of SPN.

4.5.11.1 **GRANTING NECESSARY PRIVILEGES TO A SQL SERVER USER**

To make DataSunrise work correctly with a SQL Server database, you should create a simple script to grant necessary user privileges. There are two options:

1. **Option 1.** Create a LOGIN on the server and grant it maximum privileges:

```
DECLARE @LOGIN NVARCHAR(MAX)
DECLARE @PWD NVARCHAR(MAX)
SET@LOGIN = 'bsa'$SET@PWD = '1234'

USE [master]
IF NOT EXISTS(SELECT loginname FROM [dbo].[syslogins] WHERE name = @LOGIN)
    EXEC('CREATE LOGIN [' + @LOGIN + '] WITH PASSWORD = ''' + @PWD + '''')
EXEC sp_addsrvrolemember @LOGIN, 'sysadmin'
GO
```

2. **Option 2 (safer).** Create a LOGIN on the server, USER in each database and grant them the following privileges:

```
DECLARE @DB SYSNAME
DECLARE @UPDATEABILITY NVARCHAR(128)
DECLARE @LOGIN NVARCHAR(MAX)
DECLARE @SID NVARCHAR(MAX)
DECLARE @PWD NVARCHAR(MAX)
DECLARE @USER NVARCHAR(MAX)
DECLARE ALLDB CURSOR FORSELECT name, CONVERT(NVARCHAR(128), DATABASEPROPERTYEX(name, 'Updateability')) FROM [master].[dbo].[sysdatabases]

SET@LOGIN = 'bsa'$SET@SID = ''SET@PWD = '1234'SET@USER = 'Backend User'
-- create
loginIF LEN(@SID) > 0
    EXEC('USE [master] IF NOT EXISTS(SELECT loginname FROM [dbo].[syslogins]
WHERE name = '' + @LOGIN + ''' CREATE LOGIN [' + @LOGIN + ''] WITH PASSWORD = '' + @PWD + ''', SID = '' + @SID)
ELSE
    EXEC('USE [master] IF NOT EXISTS(SELECT loginname FROM [dbo].[syslogins]
WHERE name = '' + @LOGIN + ''' CREATE LOGIN [' + @LOGIN + ''] WITH PASSWORD = '' + @PWD + '''

-- server permissions
EXEC('USE [master] GRANT VIEW ANY DATABASE TO [' + @LOGIN + ']
EXEC('USE [master] GRANT VIEW ANY DEFINITION TO [' + @LOGIN + ']
```
OPEN ALLDB
LOOP:
   FETCH NEXT FROM ALLDB INTO @DB, @UPDATEABILITY
   IF @@FETCH_STATUS = 0
   BEGIN
   -- updateability check
   IF @UPDATEABILITY = NULL OR @UPDATEABILITY = 'READ_ONLY'
   BEGIN
   PRINT 'The database ''' + @DB + '''' is have not in a updatability state.'
   PRINT 'Perhaps it is available for management from another replica (in the case of AlwaysOn, for example).'</print>
   PRINT 'In this case, make sure that the primary and secondary replica''s SID of login matched.'
   GOTO LOOP
   END-- create user
   EXEC('USE [' + @DB + '] IF NOT EXISTS(SELECT * FROM [sys].[database_principals] WHERE [name] = ''' + @USER + ''') CREATE USER [' + @USER + ']
   FOR LOGIN [' + @LOGIN + '] WITH DEFAULT_SCHEMA = dbo')
   -- map user to login IF CHARINDEX('Microsoft SQL Server 2005',@@VERSION) != 0
   EXEC('USE [' + @DB + '] EXEC sp_change_users_login ''Update_One'',
   ''' + @USER + ''', ''' + @LOGIN + '''')
   ELSE
   EXEC('USE [' + @DB + '] ALTER USER [' + @USER + '] WITH LOGIN = [' + @LOGIN + ']
   -- master permissions
   IF @DB = 'master'
   BEGIN
   EXEC('USE [' + @DB + '] GRANT SELECT ON OBJECT::[sys].[databases] TO [' + @USER + ']
   EXEC('USE [' + @DB + '] GRANT SELECT ON OBJECT::[sys].[server_principals] TO [' + @USER + ']
   -- other permissions
   EXEC('USE [' + @DB + '] GRANT SELECT ON OBJECT::[sys].[database_principals] TO [' + @USER + ']
   EXEC('USE [' + @DB + '] GRANT SELECT ON OBJECT::[sys].[database_permissions] TO [' + @USER + ']
   EXEC('USE [' + @DB + '] GRANT SELECT ON OBJECT::[sys].[all_columns] TO [' + @USER + ']
   EXEC('USE [' + @DB + '] GRANT SELECT ON OBJECT::[sys].[all_views] TO [' + @USER + ']
   EXEC('USE [' + @DB + '] GRANT SELECT ON OBJECT::[sys].[all_objects] TO [' + @USER + ']
   EXEC('USE [' + @DB + '] GRANT SELECT ON OBJECT::[sys].[schemas] TO [' + @USER + ']
   EXEC('USE [' + @DB + '] GRANT SELECT ON OBJECT::[sys].[types] TO [' + @USER + ']
   GOTO LOOP
   END
   CLOSE ALLDB
   DEALLOCATE ALLDB
   EXEC('USE [master] SELECT name, sid FROM sys.server_principals WHERE name = ''' + @LOGIN + '''
   GO
   • Delete the USER (if exists) and LOGIN from all databases

   DECLARE @DB SYSNAME
   DECLARE @UPDATEABILITY NVARCHAR(128)
   DECLARE @USER NVARCHAR(MAX)
   DECLARE @LOGIN NVARCHAR(MAX)
   DECLARE ALLDB CURSOR FOR SELECT name, CONVERT(NVARCHAR(128),
   DATABASEPROPERTYEX(name, 'Updateability')) FROM [master].[dbo].[sysdatabases]

GO
SET @USER = 'Backend User'
SET @LOGIN = 'bsa'

OPEN ALLDB
LOOP:
    FETCH NEXT FROM ALLDB INTO @DB, @UPDATEABILITY
    IF @@FETCH_STATUS = 0
    BEGIN
        -- updateability check
        IF @UPDATEABILITY = NULL OR @UPDATEABILITY = 'READ_ONLY'
        BEGIN
            PRINT 'The database '' + @DB + '' is have not in a updatability state, so we can not delete the user '' + @USER + '' from it.'
            GOTO LOOP
        END
        -- delete user
        EXEC('USE [' + @DB + ']
            IF EXISTS(SELECT * FROM [sys].[database_principals] WHERE [name] = ''' + @USER + ''') DROP USER [' + @USER + ']
        )
    END
    GOTO LOOP
END
CLOSE ALLDB

-- delete login
EXEC('USE [master]
    IF EXISTS(SELECT loginname FROM [dbo].[syslogins] WHERE name = ''' + @LOGIN + ')
    DROP LOGIN [' + @LOGIN + ']
)
DEALLOCATE ALLDB
GO

4.5.11.2 ENABLING "REGEXP REPLACE" DATA MASKING IN SQL SERVER DATABASE

SQL Server database does not support regular expressions but provides a possibility to use external addons by plugging them in the server with DLLs. "Regexp replace" masking function is built as an external addon as well. The key point here is that an addon should be plugged in a specific database, thus it could be used only inside a specific database and schema. Thus, there are two ways you can use "Regexp replace" masking in MS SQL Server:

1. Plug the addon into each database when installing DataSunrise, and use default schema (DBO). It allows to skip database when calling the masking function.

   ```sql
   SELECT [DBO].[RegexReplace]('9731246ab456cde', '[a-z]{2}', '__') AS "T2"
   ```

2. Plug the addon into the database and schema by default (MASTER.DBO).

   ```sql
   SELECT [MASTER].[DBO].[RegexReplace]('9731246ab456cde', '[a-z]{2}', '__') AS "T2"
   ```

   **Important:** In either case it would be necessary to grant the DB user a privilege to run RegexpReplace. You can do it with the following query.

   ```sql
   GRANT EXECUTE ON [MASTER].[DBO].[RegexReplace] to [name of user to obtain the privilege]
   ```

You can install the following function to enable RegExp data masking.

```csharp
using System;
using Microsoft.SqlServer.Server;
using System.Text.RegularExpressions;
```
```csharp
public partial class RegExBase{
    [SqlFunction(IsDeterministic = true, IsPrecise = true)]
    public static string RegexReplace(string input, string pattern, string replacement){
        return Regex.Replace(input, pattern, replacement);
    }
}
```

You can download the full script here: [https://www.datasunrise.com/support-files/ms_sql_regexp_replace.sql](https://www.datasunrise.com/support-files/ms_sql_regexp_replace.sql)

**Note:** More on user-defined functions here: [https://msdn.microsoft.com/en-us/library/w2kae45k(v=vs.80).aspx](https://msdn.microsoft.com/en-us/library/w2kae45k(v=vs.80).aspx)

### 4.5.12 Troubleshooting Connection Failure

In case the connection between DataSunrise and required database fails, perform the following:

1. Check the state of proxies using the web UI.
   a) Open DataSunrise GUI and go to **Configurations → Databases**
   b) Click **Edit**
   c) Click **Test Connection**
   d) Enter the password and click **Test All**
   e) If the status of all ports is **OK**, go to the next step.
2. Scan the host with the **telnet** tool.
   a) Install **telnet**.

        `sudo apt-get install telnet`

   b) Connect to the database by using telnet command and providing database’s IP address and port number. Example:

        `telnet 192.168.101 3306`

   If **Telnet** client cannot connect to the host, the issue is caused by your computer or network, not by DataSunrise software. If provided hostnames and port numbers are correct, check your network firewall or any other kind of conflicting security software that can block network traffic.

### 4.6 Additional Proxy Configuration

#### 4.6.1 Changing PostgreSQL Port Number

When configuring DataSunrise proxy, it would be necessary to change database’s port number. It is necessary if DataSunrise proxy is configured to use the port number assigned to the original database. To do this, perform the following:

1. Open the **postgresql.conf** file which is located in the **data** subfolder of the PostgreSQL’s installation folder.
2. In the **CONNECTIONS AND AUTHENTICATION** section, change the **port** parameter value (5432 by default) to a new port number.
3. Restart PostgreSQL for the changes to take effect.

#### 4.6.2 Configuring Authorization of Local Users in PostgreSQL

If DataSunrise proxy is deployed on the same host as the database, remote users connecting to the database through proxy would be treated by the database as local users. Thus, they can have password-free or simplified
authorization. We advise to disable password-free authorization for local users in the database settings, if it is enabled. To do this, perform the following:

1. Open the pg.hba file which is located in the data subfolder of PostgreSQL installation folder.
2. Edit the pg.hba file in the following way:

   ```
   # TYPE DATABASE USER ADDRESS METHOD
   # IPv4 local connections:
   host all all 127.0.0.1/32 md5
   host all all all md5
   # IPv6 local connections:
   host all all ::1/128 md5
   ```

3. As a result, MD5 or other password authentication method should be assigned for all database connections.

### 4.7 PROCESSING ENCRYPTED TRAFFIC

This subsection describes how to configure processing of encrypted traffic.

#### 4.7.1 GENERATING AN SSL CERTIFICATE

DataSunrise has a self-signed SSL certificate by default. But if you want more protection for the connection between client applications and the firewall server, you can use OpenSSL utility to create your own SSL private key.

1. Use the Linux CLI to execute the following command:

   ```
   openssl req -nodes -new -x509 -keyout server.key -out server.cert
   ```

2. Ensure that OpenSSL has generated a private key and a certificate signing request.

3. Paste the private key and the certificate into the proxy.pem file (create it if necessary) and move this file to DataSunrise installation folder:

   ```
   cat server.key > /opt/datasunrise/appfirewall.pem
   cat server.cert >> /opt/datasunrise/appfirewall.pem
   chmod 600 /opt/datasunrise/appfirewall.pem
   chown datasunrise:datasunrise /opt/datasunrise/appfirewall.pem
   ```

#### 4.7.2 CONFIGURING SSL ENCRYPTION FOR DB2

To configure DataSunrise to process SSL-encrypted traffic, perform the following:

1. Prepare your DB2 server for working with SSL. You need to get a certificate the server delivers to client during SSL connection (hereafter db2_server.crt). Refer to the following page for an example: [http://www.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/t0025241.html](http://www.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/t0025241.html)


3. Specify full path to the certificate storages in Db2KeyStoragePath and Db2KeyStashPath parameters (System Settings → Additional → Parameters).

MULTI-SERVER CONFIGURATION

Along with running single instance of DataSunrise, you can configure multiple servers to implement failover and scalability. This feature enables running multiple DataSunrise instances on separate servers sharing one configuration (Dictionary). If some of the servers go offline, other servers keep working, guaranteeing consistent traffic processing without an impact on system availability.

**Note:** for PostgreSQL users: remote configuration of DataSunrise is available only for PostgreSQL 8.2 or higher versions.

5.1 CONFIGURING A DATASUNRISE SERVER (MULTI-SERVERS)

DataSunrise server can be configured during DataSunrise installation.

1. Start the program installation with the `--remote-config` parameter.
   
   ```
   sudo ./DataSunrise.run install --remote-config
   ```

2. At the end of the installation process, specify the database to store DataSunrise configuration (the Dictionary). All servers configured to use this database, share common configuration (including common credentials to access the web UI).

   ![Figure 8: Dictionary details](image)

   **Figure 8: Dictionary details**

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database type</td>
<td>Type of database to store DataSunrise configuration (MySQL or PostgreSQL).</td>
</tr>
<tr>
<td>Host</td>
<td>IP address or name of the host, the Dictionary database is installed on.</td>
</tr>
<tr>
<td>Port</td>
<td>Port number of the Dictionary database.</td>
</tr>
<tr>
<td>Database Name</td>
<td>Name of the Dictionary database.</td>
</tr>
<tr>
<td>Login</td>
<td>User name to connect to the database.</td>
</tr>
<tr>
<td>Password</td>
<td>Password to connect to the database.</td>
</tr>
</tbody>
</table>

3. Specify details of the current DataSunrise server.
Multi-server configuration

Figure 9: Server details

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Name</td>
<td>Logical name of DataSunrise instance.</td>
</tr>
<tr>
<td>Server Hostname</td>
<td>IP address or name of the host the current DataSunrise instance is installed on.</td>
</tr>
<tr>
<td>Server Port</td>
<td>Port number of the instance’s web UI (11000 by default).</td>
</tr>
</tbody>
</table>

4. Specify the database to store audit data (Audit Storage) similarly as the Dictionary database.
5. After the configuring is completed, you can see all available DataSunrise servers (instances installed on separate servers and sharing common configuration) at the System Settings → Servers subsection.

5.2 Editing servers

All existing DataSunrise servers are equal, so you can access all server settings from the web UI of any DataSunrise instance:
1. Go to Settings → Servers. Select a required server from the list and click Edit>> to access the server settings
2. Reconfigure a server, if necessary:
### Interface Element | Description
--- | ---
Main Settings | Logical name of the DataSunrise server (instance)
Logical Name | Logical address of the server, the Instance is installed on
Host | DataSunrise backend port number (used to access the web UI)
Backend Port | Use HTTPS protocol to access the web UI
Backend HTTPS | DataSunrise Core port number
Core Port | Use HTTPS protocol to access the Core
Core HTTPS | Restart the Core process
Start, Stop, Restart the Core | Start the Core process (if stopped)
Restart | Stop running Core process
Start | Stop
Stop | System Info
Version | Program version
License Type | Type of license activated
The License is expired | License expiration date
### Multi-server configuration

<table>
<thead>
<tr>
<th>Interface Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core State</td>
<td>State of the Core process</td>
</tr>
<tr>
<td></td>
<td>• Active</td>
</tr>
<tr>
<td></td>
<td>• Inactive</td>
</tr>
<tr>
<td>Core Up Time</td>
<td>The Core run duration</td>
</tr>
<tr>
<td>Backend Up Time</td>
<td>The Backend run duration</td>
</tr>
<tr>
<td>Server Time</td>
<td>Current server time</td>
</tr>
<tr>
<td>OS Type</td>
<td>Type of the server's operating system</td>
</tr>
<tr>
<td>OS Version</td>
<td>Version of the server's operating system</td>
</tr>
<tr>
<td>Machine</td>
<td>Server hardware info</td>
</tr>
<tr>
<td>Node Name</td>
<td>Server name (PC name)</td>
</tr>
<tr>
<td>Encoding</td>
<td>Encoding used on the server</td>
</tr>
</tbody>
</table>

### 5.3 CREATING A DATABASE USER REQUIRED FOR OPERATION OF A COMMON DICTIONARY/AUDIT STORAGE

When deploying DataSunrise in multi-server configuration, a PostgreSQL or MySQL database is used to store the common Dictionary and Audit data. First you should use a database user with sufficient privileges (admin for example) to create a database or schema to store that data and then create a user that could be used for access to that data. In general, such a user should have read/write access to your "Dictionary" schema or database.

#### 5.3.1 CREATING A POSTGRESQL USER

1. Create a new user by executing the following command:
   
   ```sql
   CREATE USER <User name> WITH PASSWORD <Password>;
   ```

2. Grant the required privileges to the user:
   
   ```sql
   GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA <Dictionary schema name> TO <User name>;
   ```

#### 5.3.2 CREATING A MYSQL USER

1. Create a new user by executing the following command:
   
   ```sql
   CREATE USER <User Name> IDENTIFIED BY 'Password';
   ```

2. Grant the required privileges to the user:
   
   ```sql
   GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER, DROP, INDEX, REFERENCES ON <Dictionary database name> TO <User name>;
   ```

3. Create two databases: Audit Storage and Config Storage:
   
   ```sql
   CREATE DATABASE <Audit Storage database name> character set UTF8 collate utf8_bin;
   ```
CREATE DATABASE <Config Storage database name> character set UTF8 collate utf8_bin;
6.1. DATA SUNRISE AUTHENTICATION PROXY OVERVIEW

To maintain secure connection to databases or to the web UI, DataSunrise can be used as an authentication proxy. Once user mapping is configured, users will be able to connect to databases through DataSunrise proxy using Active Directory user credentials. DataSunrise maintains the organizational authentication policies of Microsoft Active Directory, Kerberos and LDAP protocols. You can also configure Active Directory authentication for users of DataSunrise GUI to enhance security and easily manage roles for groups of AD users.

Figure 12: Active Directory users can be mapped to one database user or each AD user can be mapped to a separate database user, as shown in the figure. When a client connects to a database, DataSunrise connects to AD services and ascertains rights of the user to connect to the database.

_DataSunrise Authentication Proxy configuration Scheme:_

1. Creating an AD user and assigning principal names with encrypted keys on the domain controller machine.
2. Installing LDAP or Kerberos client.
3. Configuring DataSunrise to map AD users to DB users.

_Guide map:_

- **LDAP authentication for DB connections via DataSunrise proxy** (Subs 6.3.1).
- **Kerberos authentication for DB connections via DataSunrise proxy** (Subs 6.3.2).
- **Active Directory authentication for users of DataSunrise GUI** (Subs 6.4).

For now, DataSunrise supports the following encryption algorithms:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Redshift</td>
<td>MD5</td>
</tr>
<tr>
<td>Greenplum</td>
<td>MD5</td>
</tr>
<tr>
<td>MySQL</td>
<td>SHA-1, SHA-256</td>
</tr>
<tr>
<td>Netezza</td>
<td>MD5, SHA-256, crypto</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>MD5</td>
</tr>
</tbody>
</table>
6.2 Configuring Active Directory Users for DataSunrise Authentication Proxy

6.2.1 Creating an Active Directory User

To configure DataSunrise Proxy Authentication, we need to create an AD user (an existing one can be used as well) and create a keytab file containing pairs of Kerberos principals and encrypted keys. Follow the steps given below to create a new AD user.

1. Log into the domain controller server, click Start → Administrative Tools, and launch Active Directory Users and Computers.
2. If it is not already selected, click the node for your domain (domain.com).
3. Right-click Users, point to New, and then click User.
4. In the New Object → User dialog box specify the parameters of the new user. It could be a regular user, it is not required to provide the user with some additional privileges. User account should be active (Account is disabled check box unchecked), and the password for the account should be perpetual (Password never expires check box checked).

6.2.2 Creating a Keytab File

To configure DataSunrise Proxy Authentication we need to create an AD user (or use an existing one) and create a keytab file containing pairs of Kerberos principals and encrypted keys. A keytab file is used to authenticate to various remote systems using Kerberos without entering a password. Follow the steps given below to configure a keytab file.

Note: it is necessary to have one user for each keytab entry.

1. Create a keytab with the first entry using ktpass tool:

   ```
   ktpass /princ user1_backend@DOMAIN.COM /mapuser user1_backend /pass <PASSWORD> /crypto_all /ptype KRB5_NT_PRINCIPAL /out C:\Users\user1\Desktop datasunrise.keytab
   ```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/princ</td>
<td>Service principal name (SPN) in the following format:</td>
</tr>
<tr>
<td></td>
<td>&lt;user_account&gt;@&lt;REALM&gt;</td>
</tr>
<tr>
<td>/mapuser</td>
<td>Maps the name of the Kerberos principal, which is specified by the princ</td>
</tr>
<tr>
<td></td>
<td>parameter, to the specified domain account.</td>
</tr>
<tr>
<td>/pass</td>
<td>Specifies a password for the principal user name.</td>
</tr>
<tr>
<td>/ptype</td>
<td>Specifies the principal type. Use KRB5_NT_PRINCIPAL.</td>
</tr>
<tr>
<td>/crypto</td>
<td>Specifies the keys that are generated in the keytab file.</td>
</tr>
</tbody>
</table>
Parameter | Description
--- | ---
-setupn | Doesn't set the user principal name along with the service principal name.

The entry created in this step should be the first in the keytab file, it should be created every time you configure AD user connection to the GUI or an AD user connects a database when creating/updating the configuration. This user will connect to the database when getting metadata from it (creating a configuration), so it is necessary to create a database user for getting metadata.

2. Create a second entry in the keytab file for AD authentication in the GUI or for connecting the database through DataSunrise proxy using AD.

The example is given for creating keytab entries for connecting to HP vertica databases via DataSunrise proxy using AD. For other databases or GUI authentication, perform the same command with corresponding service name in the /princ parameter.

```
ktpass /out ./datasunrise.keytab /princ vertica/user1.domain.com@DOMAIN.COM /mapuser user1 /mapop set /pass <PASSWORD> /ptype KRB5_NT_PRINCIPAL /crypto RC4-HMAC-NT
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/out &lt;filename&gt;</td>
<td>Output *.keytab file path.</td>
</tr>
<tr>
<td>/in &lt;filename&gt;</td>
<td>Input *.keytab file path. Use it if you need to add new entries to an existing keytab file</td>
</tr>
</tbody>
</table>
| /princ | Service principal name in the following format: 

```
 service/<FQDN> [:<port>]@<REALM>
```

**Note:** <port> is needed for SQL Server only. It is the port on which DataSunrise should wait for client connections.

- Use HTTP as a service name for AD authentication in GUI.
- Use postgres as a service name for connecting to Amazon Redshift, Greenplum, PostgreSQL, Netezza databases through DataSunrise authentication proxy.
- vertica for HP Vertica databases.
- oracle for Oracle databases.
- netezza for IBM Netezza databases.
- MSSQLSvc for SQL Server databases.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| /mapop {add | set} | Specifies how the mapping attribute is set.

- **Add** adds the value of the specified local user name. This is the default.
- **Set** sets the value for Data Encryption Standard (DES)-only encryption for the specified local user name. |
| /pass | Specifies a password for the principal user name. |
| /ptype | Specifies the principal type. Use KRB5_NT_PRINCIPAL. |
| /crypto | Specifies the keys that are generated in the keytab file. All is the most universal choice. |

3. As the result you will get the datasunrise.keytab file containing the entries as shown below. There is no need to add all database services and HTTP, use only the required ones.

```
klist -k datasunrise.keytab
```
Keytab name: FILE:datasunrise.keytab
KVNO Principal
23 user1_backend@DOMAIN.COM
23 user1_backend@DOMAIN.COM
23 user1_backend@DOMAIN.COM
23 user1_backend@DOMAIN.COM
23 user1_backend@DOMAIN.COM
24 vertica/host1.domain.com@DOMAIN.COM
24 vertica/host1.domain.com@DOMAIN.COM
24 vertica/host1.domain.com@DOMAIN.COM
24 vertica/host1.domain.com@DOMAIN.COM
24 vertica/host1.domain.com@DOMAIN.COM
24 HTTP/host1.domain.com@DOMAIN.COM
24 HTTP/host1.domain.com@DOMAIN.COM
24 HTTP/host1.domain.com@DOMAIN.COM
24 HTTP/host1.domain.com@DOMAIN.COM
24 HTTP/host1.domain.com@DOMAIN.COM

**Note:** Using this keytab by DataSunrise enables you to connect to the GUI through Kerberos and to connect to HP Vertica databases through proxy (Delegation for each proxy should be configured. Refer to sub.6.2.3)

4. Transfer the keytab file to the Linux machine where DataSunrise Security Suite is installed.
   To make DataSunrise use this keytab, add the following lines to `/etc/datasunrise.conf`.

   ```
   KRB5CCNAME=/tmp/krb5cc_datasunrise
   KRB5_KTNAME=/etc/datasunrise.keytab
   KRB5_CLIENT_KTNAME=/etc/datasunrise.keytab
   ```

   You should also change the keytab owner with the following command:

   ```
   sudo chown datasunrise datasunrise.keytab
   ```

   For safety reasons, it is recommended to limit write/save privileges to the owner only by using the following command:

   ```
   sudo chmod 600 /etc/datasunrise.keytab
   ```

**6.2.3 Configuring Active Directory Delegation**

1. On the domain controller machine, go to **Active Directory Users and Computers**, locate the account of the machine DataSunrise is installed on.

2. In the **Properties** section, go to **Delegation** tab and select **Trust this computer for delegation to specified services only** and click **Add**.

3. In the **Users and Computers** window, specify the user account that has been used to launch the database or the name of the server where the RDBMS is installed.

4. Optionally, you can use **Check names** to check if a specified user or computer exists and click **OK**, then select a required service and click **OK**.
   As an alternative, you can add SPN bases to the **AllowedToDelegateTo** parameter in user's attributes (Attribute Editor tab or use any LDAP client).
6.3 Configuring DataSunrise Authentication Proxy for Database Connections

To maintain secure connection to databases, DataSunrise can be used as an authentication proxy. Once user mapping is configured, users will be able to connect to databases through DataSunrise proxy using Active Directory user credentials. Authentication can be made using Kerberos or LDAP protocols.

Kerberos protocol works based on "tickets" and provides mutual authentication — both the user and the server verify each other’s identity. Kerberos protocol messages are protected against eavesdropping and replay attacks.

LDAP is an application protocol used as a central repository for user information and as an authentication service. Compared to Kerberos, LDAP provides one-way authentication. Hence it is not a single sign-on technology, users have to log in to every service.

Important: For now, DataSunrise authentication proxy feature is available for Amazon Redshift, Greenplum, PostgreSQL, Netezza, and Vertica databases.

6.3.1 LDAP Authentication for Database Connections

To configure DataSunrise to map Active Directory users using LDAP, perform the following.

1. To configure DataSunrise Proxy Authentication, we need to create an AD user (an existing one can be used as well) (refer to subs.6.2.1).
2. Install the LDAP client. Example of an installation command (CentOS):

   ```
   sudo yum -y install openldap-clients nss-pam-ldap
   ```
3. Specify the host and port of the LDAP server.

   - To set the parameters via the DataSunrise's GUI, use your web browser to open the following address: https://<IP address>:11000. Replace <IP adress> with the hostname of the machine DataSunrise is installed on, and enter your credentials. Then, go to System Settings → General and set the LDAP parameters given above.

   Note: before connecting to the DataSunrise's GUI on another server, you should open port 11000 on the Linux machine with `firewalld`.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP Server Host</td>
<td>Hostname of the LDAP server.</td>
</tr>
<tr>
<td>LDAP Server Port</td>
<td>&lt;636</td>
</tr>
<tr>
<td>use SSL checkbox</td>
<td>Enable/disable authentication with SSL.</td>
</tr>
<tr>
<td></td>
<td>• If you’ve specified the port 636, select the checkbox. Note that the LDAP server must support SSL connections.</td>
</tr>
<tr>
<td></td>
<td>• For port 389, uncheck the checkbox.</td>
</tr>
<tr>
<td>LDAP Domain</td>
<td>Specify the Active Directory domain name. Example: If the AD domain name is DOMAIN.COM, set the value DOMAIN.</td>
</tr>
</tbody>
</table>
• You can also change the parameters via the DataSunrise's CLI. To do this, switch to `/opt/datasunrise/cmdline` directory and execute the following commands.

```
./executecommand.sh connect -host <DataSunrise_backend_host> -port 11000 -login admin -password <DS_user_password>
./executecommand.sh changeParameter -name LDAPServersHost -value <LDAP server hostname>
./executecommand.sh changeParameter -name LDAPServersPort -value 636
```

4. Enable user mapping and set the required parameters for LDAP and SSL: go to System Settings → Additional and set the parameters given above.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserMappingType</td>
<td>Enable user mapping. Set the value 1.</td>
</tr>
<tr>
<td>LdapBaseDN</td>
<td>Specify the Active Directory domain name (Base DN). Example: If the AD domain name is DOMAIN.COM, assign the following: cn=users,dc=domain, dc=com</td>
</tr>
<tr>
<td>LdapMappingRequireClientSideSSL</td>
<td>&lt;1</td>
</tr>
<tr>
<td></td>
<td>• 1 – SSL connections must be enabled on the client side.</td>
</tr>
<tr>
<td></td>
<td>• 0 – Non-SSL connections are allowed on the client side.</td>
</tr>
<tr>
<td>AcceptOnlyMappedUsers</td>
<td>&lt;1</td>
</tr>
<tr>
<td></td>
<td>• 1 – Connection of database users via DataSunrise proxy will be restricted. Only mapped AD users will be able to connect to the DB via DS proxy.</td>
</tr>
<tr>
<td></td>
<td>• 0 is set by default. DB users and mapped AD users are able to connect to the database via DS proxy.</td>
</tr>
</tbody>
</table>

To change the parameters via the CLI, perform the following:

```
./executecommand.sh changeParameter -name UserMappingType -value 1
./executecommand.sh changeParameter -name LdapSSLMode -value 1
./executecommand.sh changeParameter -name LdapBaseDN -value 'cn=users,dc=domain,dc=com'
./executecommand.sh changeParameter -name LdapMappingRequireClientSideSSL -value 1
./executecommand.sh changeParameter -name UsersDomainName -value DOMAIN
```

5. Follow the mapping configuration instructions in subs. 6.3.3.

Important for MySQL users: there are two available methods of password transferring:

1. `sha256 password`: recommended method of password transferring. Make sure that the `MySQLUseSHA256PasswordMethodForMapping` parameter is enabled in the System Settings → Additional subsection.
2. `mysql_clear_password`: implement this method if you are using non-enterprise version of MySQL which doesn’t support encrypted transfer of passwords. Hashing or encryption cannot be done for authentication schemes that require the server to receive the password as entered on the client side. In this case perform the following.

• Enable the Cleartext Authentication Plugin:

```
mysql -enable-cleartext-plugin -h <DataSunrise_hostname> --port=3307 -u <AD_user> --password=<password>
```

• Go to System Settings → Additional and disable the `MySQLUseSHA256PasswordMethodForMapping` parameter.
**Important:** Note that when the Cleartext Authentication Plugin is used, the passwords will be sent unencrypted, that is not safe.

### 6.3.2 Kerberos Authentication for Database Connections

1. To configure DataSunrise Proxy Authentication, we need to create an AD user (an existing one can be used as well) and create a keytab file containing pairs of Kerberos principals and encrypted keys. Refer to subs.6.2

2. Install Kerberos client on your machine and configure the `/etc/kerb5.conf` file as follows.

```ini
[libdefaults]
    default_realm = DOMAIN.COM       # full domain name
    clockslew = 300
    ticket_lifetime = 1d
    forwardable = true
    proxiable = true
dns_lookup_realm = true
dns_lookup_kdc = true

[realms]
    DOMAIN.COM = {
        kdc = pcname.domain.com     # domain controller name
        admin_server = pcname.domain.com     # domain controller name
        default_domain = DOMAIN.COM     # full domain name
    }

[domain_realm]
    .domain.com = DOMAIN.COM       # domain name for dns names
domain.com = DOMAIN.COM         # domain name for dns names

[appdefaults]
    pam = {
        ticket_lifetime = 1d
        renew_lifetime = 1d
        forwardable = true
        proxiable = false
        retain_after_close = false
        minimum_uid = 0
        debug = false
    }
```

3. Follow the mapping configuration instructions in subs. 6.3.3.

**Important for MySQL users:** There are two available methods of token transferring:

1. auth_windows: To use this method, make sure that the `MySQLUseAuthGSSAPIMethodForMapping` parameter is disabled in the Settings → Additional subsection.

2. GSSAPIAuth: To use this method, go to Settings → Additional and enable the `MySQLUseAuthGSSAPIMethodForMapping` parameter.

### 6.3.3 Configuring User Mapping

1. Launch the DataSunrise command line interface located in the `cmdline` folder and connect to the DataSunrise's web UI.

   ```bash
   cd /opt/datasunrise/cmdline
   ./executecommand.sh connect -host <DataSunrise_backend_host> -login admin -password <PASSWORD>
   ```

2. Create a database instance via the DataSunrise CLI.
./executeCommand.sh addInstancePlus -dbType vertica -database <DB_name> -dbHost 10.0.14.190 -dbPort 5433 -proxyHost pcname.domain.com -proxyPort 5434 -name vertica -login <DB_user_login> -password <DB_user_password> -kerberosName vertica

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-dbType &lt;arg&gt;</td>
<td>Specify a database type (redshift</td>
</tr>
<tr>
<td>-database &lt;arg&gt;</td>
<td>Database name.</td>
</tr>
<tr>
<td>-dbHost &lt;arg&gt;</td>
<td>Host where the RDBMS is installed.</td>
</tr>
<tr>
<td>-dbPort &lt;arg&gt;</td>
<td>Database port.</td>
</tr>
<tr>
<td>-proxyHost &lt;arg&gt;</td>
<td>Use a fully qualified domain name (FQDN) of the machine as a hostname for the port.</td>
</tr>
<tr>
<td>-proxyPort &lt;arg&gt;</td>
<td>Specify the port number. It should be different from port of the interface or other proxies on the same interface.</td>
</tr>
<tr>
<td>-name &lt;arg&gt;</td>
<td>Assign a name for the instance.</td>
</tr>
<tr>
<td>-database &lt;arg&gt;</td>
<td>Database name.</td>
</tr>
<tr>
<td>-login &lt;arg&gt;</td>
<td>Database user login.</td>
</tr>
<tr>
<td>-password &lt;arg&gt;</td>
<td>Database user password.</td>
</tr>
<tr>
<td>-kerberosName &lt;arg&gt;</td>
<td>Kerberos service name.</td>
</tr>
<tr>
<td></td>
<td>• Use vertica for Vertica database</td>
</tr>
<tr>
<td></td>
<td>• postgres for Redshift, Greenplum, PostgreSQL.</td>
</tr>
<tr>
<td></td>
<td>• netezza for IBM Netezza.</td>
</tr>
</tbody>
</table>

You can create an instance via GUI as well, for which you should go to Configurations>Databases subsection, click the Database + button and specify the parameters given above.

3. Enable the user mapping function for the specified instance using the enableDbUserMapping command with the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-mapType &lt;arg&gt;</td>
<td>Mapping type (none</td>
</tr>
<tr>
<td>-database &lt;arg&gt;</td>
<td>Database name (only for -mapType db)</td>
</tr>
<tr>
<td>-dbHost &lt;arg&gt;</td>
<td>Database host (only for -mapType db)</td>
</tr>
<tr>
<td>-dbType &lt;arg&gt;</td>
<td>(greenplum</td>
</tr>
<tr>
<td>-file &lt;arg&gt;</td>
<td>File Name (only for -mapType file)</td>
</tr>
<tr>
<td>-name &lt;arg&gt;</td>
<td>Assign a name for the instance.</td>
</tr>
<tr>
<td>-login &lt;arg&gt;</td>
<td>Database user login (only for -mapType db)</td>
</tr>
<tr>
<td>-password &lt;arg&gt;</td>
<td>Database user password (only for -mapType db)</td>
</tr>
<tr>
<td>-instance &lt;arg&gt;</td>
<td>Database instance name</td>
</tr>
</tbody>
</table>

The -mapType <arg> parameter has 3 options on how to store the list of mapped users.

• -mapType config

The list of mapped users will be stored in DataSunrise configurations.
DataSunrise Authentication Proxy

46

DataSunrise Authentication Proxy

|./executecommand.sh enableDbUserMapping -instance vertica -mapType config |

- `mapType file`

  The list of mapped users will be stored in text format. `-file <arg>` is used to assign a name for the .txt file. There is no need to create a file.

  ./executecommand.sh enableDbUserMapping -instance vertica -mapType file -file mapped_users.txt

4. Map the Active Directory user to a database user. For now, SHA-512 is available for Vertica DB only, SHA-256, crypto is available for Netezza DB only, SHA-1 is available for MySQL.

  **Important:** For MySQL, hashing or encryption cannot be done for authentication schemes that require the server to receive the password as entered on the client side. That’s why we need to enable the Cleartext Authentication Plugin as follows:

  mysql -enable-cleartext-plugin -h <DataSunrise_hostname> --port=3306 -u <AD_user> --password=<password>

- To use MD5 encryption algorithm, perform the following:

  ./executecommand.sh addDbUserMapping -instance vertica -adLogin <AD_user> -dbLogin <DB_user> -dbPassword <DB_password> -hashType MD5

- To use SHA-* encryption algorithm, perform the following:

  ./executecommand.sh addDbUserMapping -instance vertica -adLogin <AD_user> -dbLogin <DB_user_1> -dbPassword <DB_password> -hashType SHA-512 -adminLogin <DB_user_2> -adminPassword <DB_user_password_2>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-instance &lt;arg&gt;</td>
<td>Instance name</td>
</tr>
<tr>
<td>-adLogin &lt;arg&gt;</td>
<td>Active Directory user login</td>
</tr>
<tr>
<td>-dbLogin &lt;arg&gt;</td>
<td>Database user login</td>
</tr>
<tr>
<td>-dbPassword &lt;arg&gt;</td>
<td>Database user password</td>
</tr>
<tr>
<td>-hashType &lt;arg&gt;</td>
<td>HashType (MD5</td>
</tr>
<tr>
<td>-adminLogin &lt;arg&gt;</td>
<td>User login to access the database where salt for the user is stored (only for SHA-***).</td>
</tr>
<tr>
<td>-adminPassword &lt;arg&gt;</td>
<td>User password to access the database where salt for the user is stored (only for SHA-***).</td>
</tr>
</tbody>
</table>

5. As a result, Active Directory user specified in step 4 will be able to connect to the database through the DataSunrise authentication proxy using Active Directory credentials.

  To delete mapping configurations, use the `delDbUserMapping` command of DataSunrise’s CLI.

  To view mapping configurations, use the `showAdDbUserMapping` command.
6.3.4 MAPPING A GROUP OF AD USERS

To map a group of AD users to a database user, perform the following:

1. Specify an Active Directory user that has access to AD groups. To do it, change the following parameters via DataSunrise’s CLI, specifying the LdapUser parameter in the format of `<domain_name>\<AD_username>`.

   ```
   ./executecommand.sh changeParameter -name LdapUser -value "DOMAIN\<AD_username>"
   ./executecommand.sh changeParameter -name LdapPassword -value <AD_user_password>
   ./executecommand.sh changeParameter -name LDAPServerHost -value <LDAP server hostname>
   ./executecommand.sh changeParameter -name LDAPServerPort -value 389
   ./executecommand.sh changeParameter -name LdapSSLMode -value 0
   ./executecommand.sh changeParameter -name UsersDomainName -value DOMAIN
   ./executecommand.sh changeParameter -name LdapBaseDN -value "cn=users,dc=domain,dc=com"
   ```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP Server Host</td>
<td>Hostname of the LDAP server.</td>
</tr>
<tr>
<td>LDAP Server Port</td>
<td>`&lt;636</td>
</tr>
<tr>
<td></td>
<td>• Set 636 to configure DataSunrise to use LDAP port with SSL.</td>
</tr>
<tr>
<td></td>
<td>• Set 389 to configure DataSunrise to use LDAP port without SSL.</td>
</tr>
<tr>
<td>UserMappingType</td>
<td>Enable user mapping. Set the value 1.</td>
</tr>
<tr>
<td>LdapSSLMode</td>
<td>`&lt;1</td>
</tr>
<tr>
<td></td>
<td>• If you’ve specified the port 636, set the value 1. Note that the LDAP server must support SSL connections.</td>
</tr>
<tr>
<td></td>
<td>• For port 389, set the value 0.</td>
</tr>
<tr>
<td>LdapBaseDN</td>
<td>Specify the Active Directory domain name (Base DN). Example: If the AD domain name is <code>DOMAIN.COM</code>, assign the following: <code>cn=users, dc=domain, dc=com</code></td>
</tr>
<tr>
<td>UsersDomainName</td>
<td>Specify the Active Directory domain name. Example: If the AD domain name is <code>DOMAIN.COM</code> set the value <code>DOMAIN</code>.</td>
</tr>
</tbody>
</table>

To change the parameters via the DataSunrise’s GUI, go to **System Settings > General** and change the following parameters:

- Specify the LDAP server’s host and port.
- Check the SSL check-box if you have chosen the port 636, uncheck for the port 389.
- Specify the AD username and password in the corresponding text fields.

Go to **System Settings > Additional** and set the rest of the parameters given above.

2. Perform all the steps from the previous section. All the actions are the same except for the adding of AD user mapping configuration (subs subs 6.3.3).

Instead of an AD username (`-adLogin` parameter) use the name of the AD group (`-adGroup`).

   ```
   ./executecommand.sh addDbUserMapping -instance vertica -adGroup <AD_group_name> -adLogin <DB_user> -dbPassword <DB_password> -hashType MD5
   ```
6.4. CONFIGURING ACTIVE DIRECTORY AUTHENTICATION IN DATASUNRISE’S GUI

DataSunrise’s web interface can also be accessed using Active Directory user credentials. It enhances security and helps to easily manage AD user roles. You can assign privileges for specified actions and sections of DataSunrise’s web interface for a group of AD users at once.

**Important:** A machine you want to configure, must be a part of the Active Directory domain. Refer to the instructions of your operating system provider for integrating a Linux machine into an AD domain.

1. Create a keytab file for a new or existing AD user using HTTP as a service name. Refer to **subs.6.2**.
2. Copy the created keytab file into the DataSunrise installation folder.
3. Configure `krb5.conf` (add keytab string) as follows:

   ```
   [libdefaults]
   default_realm                = DOMAINE.COM
   clockskew                    = 300
   ticket_lifetime              = 1d
   forwardable                  = true
   proxiable                    = true
   dns_lookup_realm             = true
   dns_lookup_kdc               = true
   default_keytab_name          = FILE:/opt/datasunrise/backend.keytab  # created keytab file
   default_ccache_name          = FILE:/tmp/krb5cc_datasunrise

   [realms]
   DOMAINE.COM = {
      kdc       = hostname.domain.com  # Domain controller name
      admin_server = hostname.domain.com  # Domain controller name
      default_domain = DOMAINE.COM  # Full domain name
   }

   [domain_realm]
   .domain.com = DOMAINE.COM
   domain.com = DOMAINE.COM

   [appdefaults]
   pam = {
      ticket_lifetime       = 1d
      renew_lifetime         = 1d
      forwardable            = true
      proxiable              = false
      retain_after_close     = false
      minimum_uid            = 0
      debug                  = false
   }
   ```

   • If non-system `krb5.conf` is used, add the following string to `start_firewall.sh` (before starting the backend):

     ```
     export KRB5_CONFIG=<path to krb5>
     ```

4. Connect to the DataSunrise’s GUI via a web browser.
5. In the **System Settings → General** select **Kerberos** from the **Type of Authentication to DataSunrise UI** drop-down list.
6. In the **System Settings → Users** subsection click **User +** to create a new DataSunrise user in the following format:
7. Check the **Active Directory Auth** checkbox and click **Save**.
8. Restart DataSunrise server process.

   ```
   sudo service datasunrise restart
   ```
9. Connect to the web UI using a fully qualified domain name (FQDN) or the IP-address as a hostname:

   ```
   https://pcname.domain.com:11000
   ```
10. Enter the credentials of the AD user specified in step 6.

### 6.5 Configuring mapping of AD users to database users via the GUI

1. Navigate to your target database profile, click **Auth Proxy Settings**. You will be redirected to the **AD to DB User Mappings** page.
2. **Note**: You can use **Config** option if information about user mapping is stored in the database or **File** option if information about mapping is stored in an external file.

   Enable user mapping for your database. Click **Enable** and in the **Enable User Mapping** window select **Database**. Specify the connection details of your target database and click **Enable**
3. Click **Mapping+** to create a new User Mapping.
4. Fill out the required fields

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AD Type</strong> drop-down list</td>
<td>Select Login for a single AD user and Group for a group of AD users</td>
</tr>
<tr>
<td><strong>AD Login</strong> field</td>
<td>Active Directory user’s name</td>
</tr>
<tr>
<td><strong>DB Login</strong> field</td>
<td>Name of database user you want to map the AD user to</td>
</tr>
<tr>
<td><strong>DB Password field</strong></td>
<td>Password of the database user</td>
</tr>
<tr>
<td><strong>Hash Type</strong> field</td>
<td>Hash type (MD5 or SHA-512)</td>
</tr>
</tbody>
</table>

5. Click **Save**.
This subsection describes how to configure most common client applications to accept proxy connections from DataSunrise. For a description of DataSunrise proxies refer to subs. 4.4

### 7.1 PGADMIN (POSTGRESQL CLIENT)

To enable PGAdmin to connect to a target database through DataSunrise proxy, perform the following:

1. Open PGAdmin III. Note **Object browser** tab displaying server connections. On default, PGAdmin displays one or two available connections (PostgreSQL 9.4 in this case).

![Figure 13: Object browser tab](image)

2. Click **File → Add Server** to add a new connection.

![Figure 14: Adding a connection](image)

3. Specify connection details for an existing DataSunrise proxy.
Figure 15: Configuring a connection

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Logical name of the connection (any name)</td>
</tr>
<tr>
<td>Host</td>
<td>IP address or name of the DataSunrise proxy</td>
</tr>
<tr>
<td>Port</td>
<td>Port number of the DataSunrise proxy</td>
</tr>
</tbody>
</table>

4. After the configuring is completed, a new connection will appear.
7.2 SQL SERVER MANAGEMENT STUDIO (MS SQL SERVER CLIENT)

To enable SSMS to connect to a target database through DataSunrise proxy, perform the following:

1. First you should enable TCP/IP protocol for the database Instance. Start SQL Server Configuration Manager utility. The .exe file can be located inside Windows/ win32 folder.

2. Select SQL Server Network Configuration → Protocols for <MS SQL Instance> → TCP/IP. Right click and select Enable in the context menu. Disable Named Pipes in similar way.

![Figure 17: Enabling TCP/IP](image)


![Figure 18: Starting Connect Object Explorer](image)

4. **Note:** Use login and password you've specified in the database profile.

In the Connect Object Explorer window, enter DataSunrise proxy details.
Configuring client applications

**Figure 19: Connect Object Explorer window**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server name</td>
<td>IP address and port of DataSunrise proxy, separated by a comma</td>
</tr>
<tr>
<td>Authentication</td>
<td>Select SQL Server Authentication, not Windows authentication</td>
</tr>
<tr>
<td>Login</td>
<td>Database user name required for database connection</td>
</tr>
<tr>
<td>Password</td>
<td>Password required for database connection</td>
</tr>
</tbody>
</table>

5. You can also use tcp: prefix before the IP address, to enable TCP/IP for the connection.

**Figure 20: Alternative method of enabling TCP/IP**

6. Click **Connect** to connect to the proxy.

### 7.3 MySQL Workbench (MySQL Client)

To enable Workbench to connect to a target database through DataSunrise proxy, perform the following:

1. Open Workbench. In the top left corner of the screen, click **Plus** to create a new connection.
2. Enter connection details.

![Figure 22: Connection details](image)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Name</td>
<td>Logical name of the connection (any name)</td>
</tr>
<tr>
<td>Connection Method</td>
<td>Use Standard method (TCP/IP)</td>
</tr>
<tr>
<td>Hostname</td>
<td>Specify DataSunrise proxy’s IP address</td>
</tr>
<tr>
<td>Port</td>
<td>Port number of DataSunrise’s proxy</td>
</tr>
<tr>
<td>Username</td>
<td>Name of database user to use for authentication</td>
</tr>
</tbody>
</table>

3. Click **Test Connection** to check if you’ve configured everything properly and click **OK**. A new connection will appear.
4. Click on the connection icon to connect to the database.
## ROTATION OF AUDIT.DB FILES

In case your Audit Storage has grown too large, you can create a new audit.db file and leave the possibility to view the contents of old audit.db files. You can configure automatic rotation of audit.db files or do it manually.

### Figure 24: Rotated audit.db files

#### 8.1 CONFIGURING AUTOMATIC ROTATION OF AUDIT.DB FILES

To schedule automatic creation of new audit.db files, perform the following:

1. Go to **System Settings → Additional**.
2. Configure automatic rotation:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuditRotationAgeThreshold</td>
<td>How long to store a current audit.db file before creating a new one</td>
</tr>
<tr>
<td>AuditRotationMaxCount</td>
<td>Maximum number of audit.db files to store</td>
</tr>
<tr>
<td>AuditRotationSizeThreshold</td>
<td>Maximum size a current audit.db file can reach before creating a new audit.db file</td>
</tr>
</tbody>
</table>

3. Click **Save** to save each parameter.

#### 8.2 MANUAL ROTATION OF AUDIT.DB FILES

To rotate audit.db files manually, perform the following:

1. Go to **System Settings → Audit Storage**
2. In the **Rotated Files** subsection, click **Enable** button to convert audit.db to new format (split it to two files)
3. Click **Manually** to create new audit.db file to write data to. All existing audit.db files are displayed in the table:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Counting number</td>
</tr>
<tr>
<td>Current</td>
<td>Is the audit.db actual or no</td>
</tr>
<tr>
<td>Reason</td>
<td>Why the audit.db file was created</td>
</tr>
<tr>
<td>Column</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>Rotate Time</td>
<td>Time at which the audit.db file was created</td>
</tr>
<tr>
<td>End Time</td>
<td>Time at which the audit.db file became not actual</td>
</tr>
</tbody>
</table>

4. **Note:** you can use an audit file during a current DataSunrise user session only. When the session has been closed (logout occurred), DataSunrise automatically switches active file to a newest available file.

Select a required audit.db in the table and click **Switch to Selected** to enable the selected audit.db.

### 8.3 Disabling Auto Rotation of Audit.db Files

To disable auto rotation of audit.db files, perform the following:

1. Stop datasunrise system service:
   ```
   sudo service datasunrise stop
   ```

2. Log in as datasunrise user
   ```
   sudo su datasunrise
   ```

3. Go to the DataSunrise installation directory (/opt/datasunrise/)

4. Move all audit files (audit*.db) to the folder located outside the DataSunrise installation directory. No audit files shall be left in the installation directory

5. Switch off the rotation of audit.db files:
   ```
   ./AppBackendService CHANGE_SETTINGS AuditRotationEnabled=0
   AuditActiveRotationID=1
   ```

6. Log datasunrise user out and start the datasunrise system service:
   ```
   sudo service datasunrise start
   ```
9.1 Working with Always On Availability Group of SQL Server

This subsection describes the basic principles of working with SQL Server’s Always on availability:

1. A client connects and authorizes in a SQL Server database through DataSunrise proxy.
2. SQL Server sends the client a command to reconnect to a secondary node.
3. DataSunrise intercepts a packet.
   - If there is a proxy associated with the secondary node, the connection address is substituted with the address of a required proxy.
   - If there is no proxy associated with the secondary node, a new proxy is created and the client receives the new proxy’s address
   - A modified packet is sent to the client.
4. Having received the reconnection command, the client connects to the required proxy.

9.2 Configuring of the Firewall Inside Azure Cloud for Maintenance of SaaS SQL Azure

Azure SQL and Always-On cluster use the same mechanism for client redirecting. When connecting to SaaS SQL Azure, if a client is inside Azure subnet, then SQL Azure can redirect a client to service (dynamic) servers for balancing of network load. In such a case a server right after authorization will return to the client an address and port number of the service the client is to be reconnected to.

To be able to control such reconnections, at the moment when a server sends a query for reconnection, DataSunrise replaces address of the service server with address of a proxy that services it.

In such a case, a client is reconnected to the DataSunrise’s proxy and not to the Azure service server. And for each unique reconnection will be created an entry like this in the Event Monitor:

```
Rewrite route: cd164f04fd1f.tr27.westus1-a.worker.database.windows.net:11082 -> 10.1.0.6:14033
```

where cd164f04fd1f.tr27.westus1-a.worker.database.windows.net:11082 is service server’s address and 10.1.0.6:14033 is address of DataSunrise proxy that maintains this service server.

If DataSunrise would not be able to find a proxy in a current instance, then 2 scenarios are available:

- If MsSqlRedirectsDisable option is disabled (by default), a proxy will be created automatically in a current instance (and an interface if required)
- An entry will be added to the Event Monitor:

```
Redirect: cd164f04fd1f.tr27.westus1-a.worker.database.windows.net:11082
```

For SaaS SQL Azure it means that a client should add proxy on this host manually (or it is done already) to make client connections controlled by DataSunrise. Otherwise, DataSunrise will loose control over client connection.

For a cluster with AlwaysOn enabled it is possible to configure redirection to Readonly-replicas, that’s why if redirecting host is already configured on DataSunrise, we will see this host in redirection notification. In both cases, a notification about redirection can be used when administering DS/AlwaysOn for diagnostics.

To add a proxy to an instance, perform the following:
• Add an interface of the target server (its cd164f04fd1f.tr27.westus1-a.worker.database.windows.net:11082 in our case);
• Add a proxy on this interface. When using standard templates for host name (0.0.0.0 or 0:0:0:0:0:0:0:0) of such a proxy, DataSunrise will return the client an address of an available nonlocal interface from DataSunrise’s host as an address for redirect.

9.3 RESTORING CONFIGURATION IF LOCAL_SETTINGS.DB IS LOST

When using HA configuration, if one of the DataSunrise servers needs to be transferred to another computer or local_settings.db is lost, DataSunrise’s configuration would be changed to the default one. To avoid this, perform the following:

1. Navigate to System Settings → Servers. Select a server you want to restore access to. Note the “id” in the web browser’s address bar. The id’s number corresponds to <SERVER_ID> parameter’s value you will be using at the following step. For example “id1” means Server number 1.
2. Execute the following command:
   
   `sudo su datasunrise -s /bin/bash`

3. Navigate to the DataSunrise installation directory:
   
   `cd /opt/datasunrise/`

4. Run the following script using the details of your database to specify the location of the database where DataSunrise Dictionary is stored:
   
   `. /AppBackendService DICTIONARY_TYPE=<type of the Dictionary database. For example "postgresql"> DICTIONARY_HOST=<Dictionary IP address or host name. For example "dstestpg.cpauhz8lxyzp.us-east-1.rds.amazonaws.com"> DICTIONARY_PORT=<Dictionary port number. For example 5432> DICTIONARY_DB_NAME=<Dictionary database name. For example "dict_db"> DICTIONARY_LOGIN=<Dictionary database user name. For example "dsuser"> DICTIONARY_PASS=<Dictionary database password> RESTORE_LOCAL_SETTINGS=<SERVER_ID>`

Note: replace `<SERVER_ID>` with the actual ID of a server you want to restore access to (see step 1). Note that all parameter’s values should be written without quotes.
KNOWLEDGE ISSUES AND TROUBLESHOOTING

This section describes the most common issues DataSunrise users face.

1. I’m trying to add a new Oracle database via Configuration menu, but connection is failing because of a “Couldn’t load oci.dll” error.
   • Probably you installed 32-bit version of Oracle Database Instant Client or did not set system variables correctly. You need to install 64-bit version of Oracle Database Instant Client and add its home directory path to %ORACLE_HOME% system variable. Then you need to add the same directory path to %PATH% system variable.

2. I’m trying to run PostgreSQL, but database connection is failed: “[unixODBC] Missing server name, port, or database name in call to CC_connect.” (error code 201).
   • Check ODBC driver availability by executing the following command:
     ```
     odbcinst -q -d
     ```
     Locate ODBC.ini file and configure it in the following way:
     ```ini
     [postgres_i]
     Description = Postgres Database
     Driver = PostgreSQL
     Database = postgres
     Servername = 127.0.0.1
     Port = 5432
     ```
     Check PostgreSQL connection by executing the following command:
     ```
     isql postgres_i username password
     ```

3. I’m trying to run DataSunrise but getting an error message: “Data source name not found and no default driver specified”.
   • Basically, the data source you are attempting to connect to does not exist on your machine. On Linux and UNIX, SYSTEM data sources are typically defined in /etc/odbc.ini. USER data sources are defined in ~/.odbc.ini.
     You should grant read access to the .ini file that contains the data source. You may need to set ODBCSYSINI, ODBCINSTINI or ODBCINI environment variables to pinpoint odbc.ini and odbcinst.ini files location if it hasn’t been done before.

4. I am not able to create a new Oracle instance on Ubuntu.
   • Most likely Oracle can’t find a missing libaio.so.1 file. Run the following command to install it on Ubuntu:
     ```
     sudo apt-get install libaio1
     ```

5. I’m trying to enter the web interface after program update, but it displays “Internal System Error” message.
   • Most likely, you kept web interface tab opened in your browser while updating the firewall. Log out the web interface if necessary and press Ctrl + F5 to reload the page.

6. I’m trying to establish connection between DataSunrise and a MySQL database, but it fails because of missing ODBC MySQL driver.
   • Certain Linux-type operating systems don’t add MySQL driver parameters into odbcinst.ini file. You should do it manually.
     If necessary, install MySQL ODBC driver by running the following commands:
     • For Debian and Ubuntu:
       ```
       sudo apt-get install libmyodbc libodbcl
       ```
• For CentOS, Red-Hat and Fedora:
  
  ```bash
  sudo yum install mysql-connector-odbc
  ```

• Edit odbcinst.ini file. Run the following command:

  ```bash
  sudo nano /etc/odbcinst.ini
  ```

  Paste the following code into odbcinst.ini and save the file:

  ```ini
  [MySQL]
  Description = ODBC for MySQL
  Driver = /usr/lib/x86_64-linux-gnu/odbc/libmyodbc.so
  Setup = /usr/lib/x86_64-linux-gnu/odbc/libodbcmyS.so
  FileUsage = 1
  ```

  Update configuration files that control ODBC access to database servers by running the following command:

  ```bash
  sudo odbcinst -I -d -f /etc/odbcinst.ini
  ```

7. I’m getting “Could not find libodbc.so.2 (unixODBC is required)” error while trying to install DataSunrise on Ubuntu 14.04. UnixODBC is installed.

• Continue program installation.
• Check if libcrypto.so.10 and libssl.so.10 files are available in the program installation folder (/opt/datasunrise/ on default) by executing the following command:

  ```bash
  ll /opt/datasunrise/
  ```

• Define odbc.so.1 file location by running the following command

  ```bash
  locate libodbc.so.1
  ```

• If libcrypto.so.10 and libssl.so.10 are available in DataSunrise installation folder, execute the following command:

  ```bash
  ln /usr/lib/x86_64-linux-gnu/libodbc.so.1 /opt/datasunrise/libodbc.so.2
  ```

  **Note:** In this case, `/usr/lib/x86_64-linux-gnu/` is Linux system folder where `libodbc.so.1` is located and `/opt/datasunrise/` is DataSunrise installation folder.

If libcrypto.so.10 and libssl.so.10 files are not available in DataSunrise installation folder, execute the following command:

```bash
sudo ln -s /usr/lib/x86_64-linux-gnu/libodbc.so.1 /usr/lib/x86_64-linux-gnu/libodbc.so.2
```

**Note:** In this case, `/usr/lib/x86_64-linux-gnu/` is Linux system folder where `libodbc.so.1` and `libodbc.so.2` are located.

8. I’m getting “Could not find ‘setcap’” error while trying to install DataSunrise on OpenSUSE 42.1.

• Install libcap-progs. To do this, run the following command:

  ```bash
  sudo zypper install libcap-progs
  ```

9. When I’m trying to run DataSunrise in the sniffer mode, it displays a message: “Can’t to parsing SSL connection in sniffer mode”. 
• To run the firewall in sniffer mode, you should disable SSL support in your client application settings (SSL Mode -> Disable). You can also switch application’s SSL Mode to “Allow” or “Prefer”, but disable SSL support in database server settings first.

10. I can’t update my DataSunrise. I run the newer version of DataSunrise installer, but installation wizard is not able to locate old DataSunrise installation folder.
  • Run DataSunrise installer in Repair mode. It removes the previous installation and updates your DataSunrise to newer version.

11. When connecting to Aurora DB, MySQL ODBC driver stops responding.
  • Most probably, you’re using ODBC driver version 5.3.6, which is known to cause freezes from time to time. Install MySQL ODBC driver version 5.3.4.

12. DataSunrise installation is aborted with Permission denied error:

```bash
$ ./DataSunrise_Database_Security_Suite_XXX.linux.64bit.run
Verifying archive integrity... 100% All good.
Uncompressing SFX installer 100%
./DataSunrise_Database_Security_Suite_XXX.linux.64bit.run: line 495: ./install.sh: Permission denied
```
  • This error occurs because on certain OSs the installer cannot unpack the installation archive into temp folder. Execute the following command:
    ```bash
    sudo ./DataSunrise_Database_Security_Suite_XXX.linux.64bit.run --target ./temp install
    ```
  This command creates a temporary folder in the current folder and unpacks the archive into it. After the installation is finished, delete this temp folder manually.

13. I forgot the password to the GUI.
  • You can set new administrator password. Use the Terminal to run DataSunrise’s AppBackendService file with set_admin_password parameter. For example: >sudo ./AppBackendService set_admin_password=new_password. To apply new password, restart DATASUNRISE system service (refer to subs. 4.1).

14. I’m using MySQL database installed on the same PC DataSunrise is installed on. I’m querying the database, but data audit doesn’t capture the events.
  • Most probably, DataSunrise proxy is not intercepting the traffic. It can occur if you’ve configured your database connection (in your DB profile) to access the database at “localhost” instead of “127.0.0.1”. In this case MySQL can use UNIX socket for connection instead of TCP. Specify the full IP address of the database and ensure that your client application uses TCP connection. Refer to the next pages, if necessary: [http://serverfault.com/questions/337818/how-to-force-mysql-to-connect-by-tcp-instead-of-a-unix-socket](http://serverfault.com/questions/337818/how-to-force-mysql-to-connect-by-tcp-instead-of-a-unix-socket), [http://www.computerhope.com/issues/ch001079.htm](http://www.computerhope.com/issues/ch001079.htm)

15. I’m using MS SQL Server database. I’m creating a target database profile, but can’t properly configure the database connection.
  • In the DB connection details, specify the credentials (Default login and Password fields) used for SQL Server authentication and not for Windows authentication. To specify the database server’s host (Host field), use actual DB server’s IP address or host name instead of server’s SPN.
**Glossary**

**Audit Storage**
A relational database (SQLite or an external DB) used to store data collected by DataSunrise's Data Audit functionality (audit data).

**CLI**
Abbreviation for "Command Line Interface" (DataSunrise's CLI). A dedicated console-based DataSunrise management tool supplied with the software. Can be used to automate operations with scripting.

**Data Audit**
DataSunrise's component used to perform Database Activity Monitoring (DAM). Data Audit enables logging database user activity and exporting audit data to an external SIEM application with Syslog messages.

**Sensitive Data Discovery**
DataSunrise's component used to search for supposed sensitive data across a target database. Data Discovery includes search filters for various types of sensitive data (financial, personal etc.)

**DB**
Abbreviation for "database".

**Dictionary**
By default, a SQLite database file (dictionary.db) used to store DataSunrise settings, Rules, database profiles etc. DataSunrise settings can be also saved in an external database when deploying the software in multi-server configuration. In this case the servers use a common Dictionary.

**Dynamic Data Masking**
DataSunrise's component used to perform dynamic data obfuscation. Dynamic Masking enables on-the-fly obfuscation of data stored in a target database. Dynamic masking modifies only data returned by a database (i.e. query results) and doesn't change the actual database contents.
GUI
Abbreviation for "Graphical User Interface" (DataSunrise's GUI, the web UI). A dedicated web-based DataSunrise management tool supplied with the software.

PROXY MODE
DataSunrise deployment option. Proxy mode enables the complete DataSunrise's functionality. In this configuration DataSunrise is deployed as a proxy between a target database and its client applications.

RDBMS
Abbreviation for "Relational Database Management System".

SNIFFER MODE
DataSunrise deployment option. Sniffer mode is used to perform data auditing only. In this configuration DataSunrise is connected to a SPAN port of a network switch to get “mirrored” traffic.

SSMS
Abbreviation for "SQL Server Management Studio". A dedicated database management tool supplied with Microsoft SQL Server databases.

STATIC DATA MASKING
DataSunrise's component used to create and maintain a solid testing or development environment while preventing accidental data leaks to third parties. Static masking enables creating of a full-functional copy of a production database with obfuscated data inside.

TARGET DATABASE
A relational database protected by DataSunrise. To enable interaction between DataSunrise and a target DB, a database profile should be created in DataSunrise's web UI.