DATASUNRISE
DATABASE SECURITY SUITE 5.1.0

ADMINISTRATION GUIDE (LINUX)
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-server configuration (High Availability mode)</td>
<td>43</td>
</tr>
<tr>
<td>5.1 Configuring a DataSunrise server (Multi-servers)</td>
<td>43</td>
</tr>
<tr>
<td>5.2 Editing servers</td>
<td>45</td>
</tr>
<tr>
<td>5.3 Creating database user required for operation of common Dictionary/Audit storage</td>
<td>47</td>
</tr>
<tr>
<td>5.3.1 Creating a PostgreSQL user</td>
<td>47</td>
</tr>
<tr>
<td>5.3.2 Creating a MySQL user</td>
<td>47</td>
</tr>
<tr>
<td>DataSunrise Authentication Proxy</td>
<td>49</td>
</tr>
<tr>
<td>6.1. DataSunrise Authentication Proxy Overview</td>
<td>49</td>
</tr>
<tr>
<td>6.2 Configuring Active Directory Users for DataSunrise Authentication Proxy</td>
<td>50</td>
</tr>
<tr>
<td>6.2.1 Creating an Active Directory User</td>
<td>50</td>
</tr>
<tr>
<td>6.2.2 Creating a Keytab File</td>
<td>50</td>
</tr>
<tr>
<td>6.2.3 Configuring Kerberos client</td>
<td>52</td>
</tr>
<tr>
<td>6.2.4 Configuring Active Directory Delegation</td>
<td>53</td>
</tr>
<tr>
<td>6.3 Configuring DataSunrise Authentication Proxy for Database Connections</td>
<td>54</td>
</tr>
<tr>
<td>6.3.1 LDAP Authentication for Database Connections</td>
<td>54</td>
</tr>
<tr>
<td>6.3.2 Kerberos Authentication for Database Connections</td>
<td>56</td>
</tr>
<tr>
<td>6.3.3 Configuring User Mapping</td>
<td>56</td>
</tr>
<tr>
<td>6.3.4 Mapping a Group of AD Users</td>
<td>57</td>
</tr>
<tr>
<td>6.4 Configuring Mapping of AD Users to Database Users via the Web Console</td>
<td>58</td>
</tr>
<tr>
<td>6.4.1 LDAP users cache</td>
<td>59</td>
</tr>
<tr>
<td>Configuring client applications</td>
<td>60</td>
</tr>
<tr>
<td>7.1 PGAdmin (PostgreSQL client)</td>
<td>60</td>
</tr>
<tr>
<td>7.2 SQL Server Management Studio (MS SQL Server client)</td>
<td>62</td>
</tr>
<tr>
<td>7.3 MySQL Workbench (MySQL client)</td>
<td>63</td>
</tr>
<tr>
<td>Rotation of audit.db files</td>
<td>66</td>
</tr>
<tr>
<td>8.1 Configuring automatic rotation of audit.db files</td>
<td>66</td>
</tr>
<tr>
<td>8.2 Manual rotation of audit.db files</td>
<td>66</td>
</tr>
<tr>
<td>8.3 Disabling auto rotation of audit.db files</td>
<td>67</td>
</tr>
<tr>
<td>Always On</td>
<td>68</td>
</tr>
<tr>
<td>9.1 Working with Always On Availability Group of SQL Server</td>
<td>68</td>
</tr>
<tr>
<td>9.2 Configuring Firewall Inside Azure Cloud for Maintenance of SaaS SQL Azure</td>
<td>68</td>
</tr>
<tr>
<td>9.3 Restoring Configuration if local_settings.db is Lost</td>
<td>69</td>
</tr>
<tr>
<td>Integration with CyberArk AIM</td>
<td>70</td>
</tr>
<tr>
<td>10.1 AIM installation</td>
<td>70</td>
</tr>
<tr>
<td>10.2 AIM configuration. Defining the Application ID (APPID) and authentication details</td>
<td>70</td>
</tr>
<tr>
<td>10.3 Provisioning account and settings permission for application access</td>
<td>71</td>
</tr>
<tr>
<td>10.4 DataSunrise installation and configuration</td>
<td>72</td>
</tr>
</tbody>
</table>
Creating a Health Check

AWS Cloud Watch custom metrics

Authentication of AD users to the DataSunrise’s web UI using LDAP

Configuring authentication of users to the web UI using LDAP

Single Sign-On in DataSunrise

Single Sign-On

Configuring SSO authentication based on OpenID (using Okta as the service provider)

Configuring SSO authentication based on SAML (using Okta as the service provider)

Frequently Asked Questions

Glossary
## 3RD-PARTY COMPONENTS USED IN DATASUNRISE

### 1. C/C++ libraries

<table>
<thead>
<tr>
<th>Component</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACE license</td>
<td><a href="http://www.cs.wustl.edu/%7Eschmidt/ACE-copying.html">http://www.cs.wustl.edu/%7Eschmidt/ACE-copying.html</a></td>
</tr>
<tr>
<td>curl license</td>
<td><a href="https://github.com/curl/curl/blob/master/COPYING">https://github.com/curl/curl/blob/master/COPYING</a></td>
</tr>
<tr>
<td>fastlz license</td>
<td><a href="http://fastlz.org/LICENSE">http://fastlz.org/LICENSE</a></td>
</tr>
<tr>
<td>ICU license</td>
<td><a href="http://www.unicode.org/copyright.html#License">http://www.unicode.org/copyright.html#License</a></td>
</tr>
<tr>
<td>krb5 license</td>
<td><a href="https://web.mit.edu/kerberos/krb5-1.12/doc/mitK5license.html">https://web.mit.edu/kerberos/krb5-1.12/doc/mitK5license.html</a></td>
</tr>
<tr>
<td>Documentation components of this software distribution are licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.</td>
<td><a href="http://creativecommons.org/licenses/by-sa/3.0/">http://creativecommons.org/licenses/by-sa/3.0/</a></td>
</tr>
<tr>
<td>libbacktrace license</td>
<td><a href="https://github.com/ianlancetaylor/libbacktrace/blob/master/LICENSE">https://github.com/ianlancetaylor/libbacktrace/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>libpcap license</td>
<td><a href="https://github.com/the-tcpdump-group/libpcap/blob/master/LICENSE">https://github.com/the-tcpdump-group/libpcap/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>libpq (part of PostgreSQL) license</td>
<td><a href="https://www.postgresql.org/about/licence/">https://www.postgresql.org/about/licence/</a></td>
</tr>
<tr>
<td>OpenLDAP license</td>
<td><a href="https://github.com/openldap/openldap/blob/master/LICENSE">https://github.com/openldap/openldap/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>OpenSSL license</td>
<td><a href="https://github.com/openssl/openssl/blob/master/LICENSE">https://github.com/openssl/openssl/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>OTL license</td>
<td><a href="https://sourceforge.net/directory/license:bsd/">https://sourceforge.net/directory/license:bsd/</a></td>
</tr>
<tr>
<td>PCRE license</td>
<td><a href="https://www.pcre.org/licence.txt">https://www.pcre.org/licence.txt</a></td>
</tr>
<tr>
<td>winpcap license</td>
<td><a href="https://www.winpcap.org/misc/copyright.htm">https://www.winpcap.org/misc/copyright.htm</a></td>
</tr>
<tr>
<td>zlib license</td>
<td><a href="https://www.zlib.net/zlib_license.html">https://www.zlib.net/zlib_license.html</a></td>
</tr>
<tr>
<td>httpxx license</td>
<td><a href="https://github.com/AndreLouisCaron/httpxx/blob/master/LICENSE.txt">https://github.com/AndreLouisCaron/httpxx/blob/master/LICENSE.txt</a></td>
</tr>
<tr>
<td>jsonsl license</td>
<td><a href="https://github.com/mnunberg/jsonsl/blob/master/LICENSE">https://github.com/mnunberg/jsonsl/blob/master/LICENSE</a></td>
</tr>
</tbody>
</table>

### 2. OpenSSL, krb5 and OpenLDAP libraries contains implementation of crypto algorithms. They does not subject of EAR because it is "publicity available" software regarding ([https://www.bis.doc.gov/index.php/policy-guidance/encryption/1-encryption-items-not-subject-to-the-ear](https://www.bis.doc.gov/index.php/policy-guidance/encryption/1-encryption-items-not-subject-to-the-ear))

### 3. Java and Javascript libraries used in UI and CLI

<table>
<thead>
<tr>
<th>Component</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>Some of the icons used are from the FAMFAMFAM Silk Icon Collection and are licensed under Creative Commons Attribution 2.5 License</td>
<td><a href="http://www.famfamfam.com/lab/icons/silk/">http://www.famfamfam.com/lab/icons/silk/</a> <a href="https://creativecommons.org/licenses/by/2.5/">https://creativecommons.org/licenses/by/2.5/</a></td>
</tr>
<tr>
<td>guava license</td>
<td><a href="https://github.com/google/guava/blob/master/LICENSE">https://github.com/google/guava/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>Component</td>
<td>License</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>gflot license</td>
<td><a href="https://github.com/nmorel/gflot/blob/master/gflot/LICENSE">https://github.com/nmorel/gflot/blob/master/gflot/LICENSE</a></td>
</tr>
<tr>
<td>jquery license</td>
<td><a href="https://github.com/jquery/jquery/blob/master/LICENSE.txt">https://github.com/jquery/jquery/blob/master/LICENSE.txt</a></td>
</tr>
<tr>
<td>The following icons used</td>
<td><a href="https://creativecommons.org/licenses/by/3.0/">https://creativecommons.org/licenses/by/3.0/</a></td>
</tr>
<tr>
<td>are from the Noun Project</td>
<td></td>
</tr>
<tr>
<td>(thenounproject.com) and</td>
<td></td>
</tr>
<tr>
<td>are licensed under Creative</td>
<td></td>
</tr>
<tr>
<td>Commons Attribution 3.0</td>
<td></td>
</tr>
<tr>
<td>License</td>
<td></td>
</tr>
<tr>
<td>Creative Commons Attribution 3.0 License</td>
<td><a href="https://creativecommons.org/licenses/by/3.0/">https://creativecommons.org/licenses/by/3.0/</a></td>
</tr>
<tr>
<td>crypto-js license</td>
<td><a href="https://github.com/brix/crypto-js/blob/develop/LICENSE">https://github.com/brix/crypto-js/blob/develop/LICENSE</a></td>
</tr>
<tr>
<td>The MIT License (MIT)</td>
<td><a href="http://opensource.org/licenses/MIT">http://opensource.org/licenses/MIT</a></td>
</tr>
<tr>
<td>fontfaceobserver license</td>
<td><a href="https://github.com/bramstein/fontfaceobserver/blob/master/LICENSE">https://github.com/bramstein/fontfaceobserver/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>freezer-js license</td>
<td><a href="https://github.com/isagalaev/highlight.js/blob/master/LICENSE">https://github.com/isagalaev/highlight.js/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>history license</td>
<td><a href="https://github.com/ReactTraining/history/blob/master/LICENSE">https://github.com/ReactTraining/history/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>imagesloaded license</td>
<td><a href="https://desandro.mit-license.org/">https://desandro.mit-license.org/</a></td>
</tr>
<tr>
<td>immutable license</td>
<td><a href="https://github.com/facebook/immutable-js/blob/master/LICENSE">https://github.com/facebook/immutable-js/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>lodash license</td>
<td><a href="https://github.com/lodash/lodash/blob/master/LICENSE">https://github.com/lodash/lodash/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>md5 license</td>
<td><a href="https://github.com/pvorb/node-md5/blob/master/LICENSE">https://github.com/pvorb/node-md5/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>moment license</td>
<td><a href="https://github.com/moment/moment/blob/develop/LICENSE">https://github.com/moment/moment/blob/develop/LICENSE</a></td>
</tr>
<tr>
<td>normalize.css license</td>
<td><a href="https://github.com/necolas/normalize.css/blob/master/LICENSE.md">https://github.com/necolas/normalize.css/blob/master/LICENSE.md</a></td>
</tr>
<tr>
<td>noty license</td>
<td><a href="https://github.com/needim/noty/blob/master/LICENSE.txt">https://github.com/needim/noty/blob/master/LICENSE.txt</a></td>
</tr>
<tr>
<td>prop-types license</td>
<td><a href="https://github.com/facebook/prop-types/blob/master/LICENSE">https://github.com/facebook/prop-types/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>react, react-dom license</td>
<td><a href="https://github.com/facebook/react/blob/master/LICENSE">https://github.com/facebook/react/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>react-day-picker license</td>
<td><a href="https://github.com/gpbl/react-day-picker/blob/master/LICENSE">https://github.com/gpbl/react-day-picker/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>react-docgen license</td>
<td><a href="https://github.com/">https://github.com/</a> reactjs/react-docgen/blob/master/LICENSE</td>
</tr>
<tr>
<td>react-highcharts license</td>
<td><a href="https://github.com/kirjs/react-highcharts/blob/master/LICENSE">https://github.com/kirjs/react-highcharts/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>react-redux license</td>
<td><a href="https://github.com/">https://github.com/</a> reactjs/react-redux/blob/master/LICENSE.md</td>
</tr>
<tr>
<td>react-router license</td>
<td><a href="https://github.com/ReactTraining/react-router/blob/master/LICENSE">https://github.com/ReactTraining/react-router/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>react-router-redux license</td>
<td><a href="https://github.com/">https://github.com/</a> reactjs/react-router-redux/blob/master/LICENSE</td>
</tr>
<tr>
<td>react-sortable-hoc license</td>
<td><a href="https://github.com/clauderic/react-sortable-hoc/blob/master/LICENSE">https://github.com/clauderic/react-sortable-hoc/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>react-styled-select license</td>
<td><a href="https://github.com/agutoli/react-styled-select/blob/master/LICENSE">https://github.com/agutoli/react-styled-select/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>react-test-renderer license</td>
<td><a href="https://github.com/facebook/react/blob/master/LICENSE">https://github.com/facebook/react/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>react-truncate license</td>
<td><a href="https://github.com/One-com/react-truncate/blob/master/LICENSE.md">https://github.com/One-com/react-truncate/blob/master/LICENSE.md</a></td>
</tr>
<tr>
<td>redux license</td>
<td><a href="https://github.com/">https://github.com/</a> reactjs/redux/blob/master/LICENSE.md</td>
</tr>
<tr>
<td>redux-form license</td>
<td><a href="https://github.com/">https://github.com/</a> erikras/redux-form/blob/master/LICENSE</td>
</tr>
<tr>
<td>Component</td>
<td>License</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td>url-search-params-polyfill license</td>
<td><a href="https://github.com/jerrybendy/url-search-params-polyfill/blob/master/LICENSE">https://github.com/jerrybendy/url-search-params-polyfill/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>Open Sans font (regular and semi-bold) license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a></td>
</tr>
<tr>
<td>JasperReports license</td>
<td><a href="https://github.com/TIBCOSoftware/jasperreports/blob/master/jasperreports/license.txt">https://github.com/TIBCOSoftware/jasperreports/blob/master/jasperreports/license.txt</a></td>
</tr>
<tr>
<td>Apache Commons BeanUtils license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0.txt">http://www.apache.org/licenses/LICENSE-2.0.txt</a></td>
</tr>
<tr>
<td>Apache Digester license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0.txt">http://www.apache.org/licenses/LICENSE-2.0.txt</a></td>
</tr>
<tr>
<td>Apache Commons Logging license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0.txt">http://www.apache.org/licenses/LICENSE-2.0.txt</a></td>
</tr>
<tr>
<td>IText 2.1.7 license</td>
<td><a href="https://www.mozilla.org/en-US/MPL/1.1/">https://www.mozilla.org/en-US/MPL/1.1/</a></td>
</tr>
<tr>
<td>Legion of the Bouncy Castle license</td>
<td><a href="http://www.bouncycastle.org/licence.html">http://www.bouncycastle.org/licence.html</a></td>
</tr>
<tr>
<td>JFreeChart, JCommon license</td>
<td><a href="https://github.com/jfree/jfreechart/blob/master/licence-LGPL.txt">https://github.com/jfree/jfreechart/blob/master/licence-LGPL.txt</a></td>
</tr>
<tr>
<td>Codehaus Castor Core, Codehaus Castor XML license</td>
<td><a href="http://castor.exolab.org/license.html">http://castor.exolab.org/license.html</a></td>
</tr>
<tr>
<td>AOP Alliance license</td>
<td><a href="http://aopalliance.sourceforge.net/">http://aopalliance.sourceforge.net/</a></td>
</tr>
<tr>
<td>javax-inject license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a></td>
</tr>
<tr>
<td>FasterXML Jackson Core, FasterXML Jackson Databind and FasterXML Jackson Annotations license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a></td>
</tr>
<tr>
<td>Apache Lucene components license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a></td>
</tr>
<tr>
<td>Apache Jakarta Regexp license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a></td>
</tr>
<tr>
<td>Open Java API for OLAP (olap4j) license</td>
<td><a href="https://github.com/olap4j/olap4j/blob/master/LICENSE">https://github.com/olap4j/olap4j/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>ZXing license</td>
<td><a href="https://github.com/zxing/zxing/blob/master/LICENSE">https://github.com/zxing/zxing/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>icu4j license</td>
<td><a href="http://source.icu-project.org/repos/icu/icu/tags/release-57-1/LICENSE">http://source.icu-project.org/repos/icu/icu/tags/release-57-1/LICENSE</a></td>
</tr>
<tr>
<td>DynamicReports license</td>
<td><a href="http://www.dynamicreports.org/license">http://www.dynamicreports.org/license</a></td>
</tr>
<tr>
<td>Apache Commons Lang license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a></td>
</tr>
<tr>
<td>barcode4j license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a></td>
</tr>
<tr>
<td>Apache Avalon license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a></td>
</tr>
<tr>
<td>xml-apis license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a></td>
</tr>
<tr>
<td>Apache XML Graphics components license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a></td>
</tr>
<tr>
<td>Apache Commons Collections license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a></td>
</tr>
<tr>
<td>JSON Java license</td>
<td><a href="https://github.com/stleary/JSON-java/blob/master/LICENSE">https://github.com/stleary/JSON-java/blob/master/LICENSE</a></td>
</tr>
<tr>
<td>opencsv license</td>
<td><a href="http://opencsv.sourceforge.net/license.html">http://opencsv.sourceforge.net/license.html</a></td>
</tr>
<tr>
<td>JCommander license</td>
<td><a href="https://github.com/cbeust/jcommander/blob/master/license.txt">https://github.com/cbeust/jcommander/blob/master/license.txt</a></td>
</tr>
<tr>
<td>Component</td>
<td>License</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>Apache Commons CLI, Apache Commons Codec and Apache Commons Logging license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a></td>
</tr>
<tr>
<td>Apache Http Client and Apache Http Core license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a></td>
</tr>
<tr>
<td>Apache Http Client and Apache Http Core license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a></td>
</tr>
<tr>
<td>Google JSON.simple license</td>
<td><a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a></td>
</tr>
</tbody>
</table>
## 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, dropdown lists, links etc)</td>
<td>Click <strong>Save</strong></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 the <strong>Help</strong> link</td>
</tr>
<tr>
<td><strong>Important:</strong></td>
<td>Important information</td>
<td><strong>Important:</strong> use the &quot;admin&quot; username on the first startup</td>
</tr>
<tr>
<td>code block</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
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 the Windows and Linux operating systems, runs fast and independently of any applications and doesn’t put any unnecessary load on a database server.

DataSunrise can perform 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 codes etc. Data auditing results can be exported to an external system, such as SIEM.

- **Data Protection.** The 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 the output of sensitive data from a database by replacing it with random or real-looking data. DataSunrise's masking includes both Dynamic and Static data masking. A variety of masking algorithms can be used for any possible scenario.

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

1.2 SUPPORTED DATABASES AND FEATURES

DataSunrise supports the following databases and features:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Database</td>
<td>9.2-18</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>7.4-10.1</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Netezza</td>
<td>6.0-7.2.1</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Greenplum</td>
<td>4.2-4.3</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>IBM DB2</td>
<td>9.7-11.1</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>MS SQL Server</td>
<td>2005-2017</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Amazon Aurora</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Amazon Redshift</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>MariaDB</td>
<td>5.1-10.3</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>MySQL</td>
<td>5.0-8</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Teradata</td>
<td>13-16</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>
### General Information

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache Hive</td>
<td>1.0-2.3</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Vertica</td>
<td>7.0-9.0</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>SAP HANA</td>
<td>1.0-2.0</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>MongoDB</td>
<td>2.6-3.6</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Amazon DynamoDB</td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Percona Server for MySQL</td>
<td>5.1+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Impala</td>
<td>2.x</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cassandra</td>
<td>3.11.1-3.11.2</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>(servers),</td>
<td>3.4.x (CQL)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Athena</td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** proxying both of encrypted and unencrypted traffic is supported for all types of databases

### 1.3 Supported Operating Systems

DataSunrise is compatible with the following operating systems:

- Red Hat Linux 6+
- Debian 7/0+
- CentOS 6+
- Ubuntu 14/04 LTS+
1.4 DATASUNRISE OPERATION MODES

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

1.4.1 SNIFFER 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 able to capture a copy of the database traffic from the network switch’s "mirrored" port.

Figure 1: Sniffer mode operation scheme

In this configuration, DataSunrise can be used only for "passive security" (which means that "active security" features such as Database Firewall or Data Masking are not supported in this mode). When deployed in the Sniffer mode, DataSunrise can only perform database activity monitoring (Data Audit), because it can’t modify database traffic. Running DataSunrise in the Sniffer mode does not require any additional databases or client application reconfiguration. The Sniffer mode can be used for data auditing 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 an SQL Server database, do not use ephemeral ciphers. DataSunrise deployed in the Sniffer mode does not support connection redirection to a random port (like Oracle’s). All the network interfaces (the main one and the one database is redirecting to) shall be added to DataSunrise.
1.4.2 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.

Figure 2: Proxy mode operation scheme

The Proxy mode is for "active protection". DataSunrise intercepts the SQL queries sent to a protected database by database users, checks if they comply with the existing security policies, and audits, blocks or modifies the incoming queries or query results if necessary. When deployed 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 using DataSunrise in the proxy mode. It provides full protection and in this mode DataSunrise supports processing of encrypted traffic and redirects connections (it is essential for SAP HANA, Oracle, Vertica, MS SQL).

1.5 System Requirements

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

- Operating system: 64-bit Linux
- CPU: 8 cores
- RAM: 8-16 Gb
- Available disk space: 3GB. 100 GB for storing audit records is required
- Linux-compatible file system (NFS and SMB systems are not supported).

1.6 Useful Resources

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

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

## 2.1 INSTALLING DATASUNRISE ON A DATABASE SERVER

![Diagram of Deployment on a DB server](image)

**Figure 3: Deployment on a 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 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 in 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 there is one more Vertica or Teradata database, the default port cannot be used again because it is already taken. A proxy to another database should be opened on another port and database clients should be reconfigured.

**Tip:** You can use the installation method described above during firewall testing, but some DB clients will 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 a proxy connection.

#### B) RECONFIGURING CLIENT APPLICATIONS

- Make sure that DataSunrise uses the same port number as the database
- Configure all client applications to connect to DataSunrise, not to the database
2.1.2 SNIFFER 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

To deploy DataSunrise in the Proxy mode, perform the following:
• Configure the 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 the port numbers less than 1024 for privileged system processes, so it’s preferable to use the port numbers greater than 1024.

2.2.2 SNIFFER MODE

**Figure 4: Proxy mode deployment scheme**

**Figure 5: Sniffer mode deployment scheme**
To deploy DataSunrise in the Sniffer mode, configure your network switch to transfer mirrored traffic to DataSunrise (refer to your network switch’s user guide for description of the port mirroring procedure).
**DATASUNRISE INSTALLATION AND REMOVAL**

**Note:** Before you begin the DataSunrise installation process, select an appropriate deployment option (subsections 2.1 and 2.2) and perform all the required preparations. Also make sure that the machine where you want to install DataSunrise meets the system requirements listed in subsection 1.4.

### 3.1 REQUIRED COMPONENTS

Depending on the DataSunrise deployment option and RDBMS used, it might be necessary to install some additional components.

1. **Install unixODBC:**
   To install it from the 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
   ```
   **Important:** All environment variables should be added to:
   ```bash
   /etc/datasunrise.conf
   ```
   It is not recommended to change:
   ```bash
   /opt/datasunrise/start_firewall.sh
   ```

2. **To run DataSunrise with MySQL and PostgreSQL databases, install the ODBC driver. You can download it here:**

3. **To run DataSunrise with Oracle databases, install the OCI driver. You can download it here:**
   **Note:** Having installed 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"
   ```
   After that, restart the DataSunrise's system service:
   ```bash
   sudo service datasunrise restart
   ```
   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 a Netezza database, install the dedicated ODBC driver. Download it from the IBM Fix Central website:

http://www-933.ibm.com/support/fixcentral/

**Note:** Your IBM ID should be associated with your IBM customer ID with active support and maintenance contract for the 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 the ODBC driver:


6. To run DataSunrise with SQL Server, you might need to install the ODBC driver:


7. To run DataSunrise with Hive, install the Hortonworks ODBC driver:

https://hortonworks.com/downloads/

8. To run DataSunrise with Vertica databases, install the 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 vzxf vertica_x.x..xx_odbc_64_linux.tar.gz
  ```

- Open the file that contains the 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/lib64/libverticaodbc.so
  ErrorMessagesPath=/opt/vertica
  ```

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

  ```
  [Driver]
  DriverManagerEncoding=UTF-16
  ErrorMessagesPath=/opt/vertica
  LogLevel=4
  LogPath=/tmp
  ```
9. To run DataSunrise with Amazon Redshift, install the Redshift ODBC driver:
   http://docs.aws.amazon.com/redshift/latest/mgmt/install-odbc-driver-linux.html
10. To connect to IBM DB2 databases, install IBM Data Server Client Package.
    
    - Download the 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_vll.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
      ```
11. To work with SAP Hana databases, install the Hana driver
    
    - Download and install the Hana client. Refer to the following link: https://help.sap.com/viewer/e7e79e15f5284474b965872bf0f0a3d63/2.0.02/en-US/d41dee64bb57101490ff61557863c06.html
    - Run the following command via the Linux CLI:
      ```
      export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/sap/hdbclient/
      ```
    - Add the following lines to the odbc.ini file
      ```
      [HDBODBC]
      Driver64=/usr/sap/hdbclient/libodbcHDB.so
      DriverUnicodeType=1
      ```
    - Add the following lines to the odbcinst.ini file:
      ```
      [HDBODBC]
      Description    = ODBC for SAP HANA
      Driver64    = /usr/sap/hdbclient/libodbcHDB.so
      FileUsage    = 1
      ```
    
    **Important:** on some Linux systems (Red Hat for example), you might need to install the "libaio" package:
    ```
    sudo yum install libaio
    ```
12. To work with Impala databases, install the ODBC driver
    
    - Refer to the following link: https://www.cloudera.com/documentation/other/connectors/impala-odbc/latest/Cloudera-ODBC-Driver-for-Impala-Install-Guide.pdf
13. To use the Report Generator functionality, install Java SE Runtime Environment 8
    JRE 8: http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html
14. To use DataSunrise with a Teradata database, install the Teradata driver:
    
    - Download the driver from here: http://downloads.teradata.com/download/connectivity/odbc-driver/windows
    - Unpack the archive contents:
      ```
      tar -xvzf tdodbc1610__linux_indep.16.10.00.06-1.tar.gz
      ```
    - Install RPMs
3.2 PROGRAM INSTALLATION

To install DataSunrise, perform the following:

1. Use the following command to give the execution permission for the DataSunrise installation file:
   ```
sudo chmod +x DataSunrise_Suite_XXX.linux.64bit.run
   
   ```

2. Start the installation by executing the following command:
   ```
sudo ./DataSunrise_Suite_XXX.linux.64bit.run install
   
   ```

**Note:** on some Linux distributions (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 a quick installation using the default settings</td>
</tr>
<tr>
<td>--no-password</td>
<td>Don’t generate a password for the Web Console at the end of the installation process (set the password for the Web Console after the installation)</td>
</tr>
<tr>
<td>--extract-only</td>
<td>Extract the DataSunrise distribution into the specified folder without installation (in this case you can start DataSunrise manually from the installation folder)</td>
</tr>
<tr>
<td>--no-start</td>
<td>Don’t start the DataSunrise service after the installation</td>
</tr>
<tr>
<td>--remote-config</td>
<td>Configure remote Dictionary</td>
</tr>
<tr>
<td>remove</td>
<td>Uninstall DataSunrise</td>
</tr>
<tr>
<td>repair</td>
<td>Repair the DataSunrise installation</td>
</tr>
<tr>
<td>update</td>
<td>Update DataSunrise (replace the existing binary files, Web Console, CLI and documentation files with the latest counterparts).</td>
</tr>
<tr>
<td>-v</td>
<td>Display errors that could occur during the installation</td>
</tr>
</tbody>
</table>

For example:
3. Specify the DataSunrise installation folder in the Target directory line if necessary.

**Note:** DataSunrise is installed into the `opt/datasunrise` folder by default.

![Installation Log](image)

**Figure 6:** Note that DataSunrise generates a password for the Web Console at the end of the installation process (by default)

4. If necessary, replace DataSunrise's SSL certificate with a new one (refer to subs. 4.2.2).

### 3.3 Program Removal

To uninstall DataSunrise, perform the following:

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

```bash
sudo ./DataSunrise_Suite_X_XX.linux.64bit.run remove
```
3.4 DATA SUNRISE INSTALLATION FOLDER

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

Figure 7: DataSunrise files and folders

1. DataSunrise folders:

<table>
<thead>
<tr>
<th>Folder name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmdline</td>
<td>Contains the DataSunrise Command Line Interface (CLI) files</td>
</tr>
<tr>
<td>gwt</td>
<td>Contains the Web Console files</td>
</tr>
<tr>
<td>logs</td>
<td>Log files (Back end, Core, Web Console 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>The system process required for operation of the Web Console and control of AppFirewallCore</td>
</tr>
<tr>
<td>appfirewall.pem</td>
<td>SSL certificate for the Web Console</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 updates</td>
</tr>
<tr>
<td>dictionary.db</td>
<td>Contains the 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 the queries used by Oracle SQL Developer (refer to the Query Groups subsection for more information)</td>
</tr>
<tr>
<td>start_firewall.sh</td>
<td>The script that starts the datasunrise system service</td>
</tr>
<tr>
<td>stop_firewall.sh</td>
<td>The script that stops the datasunrise system service</td>
</tr>
</tbody>
</table>
### 3.5 Updating DataSunrise

To update DataSunrise, perform the following:

1. Connect to the DataSunrise's web UI (refer to subs. 4.1 Starting DataSunrise on page 27). Navigate to System Settings → About.
2. Click Update. Note that this button appears only if a newer version of DataSunrise available at the official web site.
3. Wait for the update process to complete and refresh the Web Console page (Ctrl+F5) as the initial console session would have been terminated upon the restart of the DataSunrise service.

**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 files required for installation roll back are retained in case of any issues that would result from the upgrade.

In case the update procedure fails and you need to return to the previous state, all the backup files should be manually copied to the main DataSunrise directory.

The update procedure will take between 5 to 7 seconds. During that period the DataSunrise service may become unavailable if you don’t run multiple DataSunrise instances in High Availability (HA) configuration.

### 3.6 Migrating DataSunrise to Another Server

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

1. Stop the DataSunrise system 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 the destination server.
4. Stop the DataSunrise system service on the destination host.
5. Move the `dictionary.db`, `event.db` and `audit.db` files to the new DataSunrise instance’s installation folder.
6. Start the DataSunrise system service on the destination host.
7. Refer to DataSunrise's console to confirm the completeness of the migration of all policies and configurations.

### 3.7 Rollback

During the DataSunrise upgrade process, the installer creates a back up folder where all files required to roll back the installation are retained in case of any issues that would result from the upgrade. To restore the previous version of DataSunrise, perform the following:

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

```
sudo ./DataSunrise.run rollback
```
You will get a list of all available back ups to decide the point to which you would like to reverse the current configuration.

2. Choose the correct back up to restore from (select its number. Note that the higher number is, the newer the back up is.

### 3.8 Encrypting the Dictionary and Audit Storage (PostgreSQL)

DataSunrise enables you to encrypt Dictionary and Audit Storage database fields with pgcrypto (PostgreSQL only). Symmetric encryption is used with password which is stored in the field_crypto.pwd file.

Before enabling encryption, you should create the pgcrypto extension in your target database. Do the following:

1. Connect to your target database as the **postgres** user.
2. Execute the following command:

   ```
   CREATE EXTENSION IF NOT EXISTS pgcrypto;
   ```

   **Note:** you can specify encryption options as described in the **F.25.3.8** subsection of the following guide: [https://www.postgresql.org/docs/9.6/pgcrypto.html](https://www.postgresql.org/docs/9.6/pgcrypto.html)

3. For encrypting the Audit storage (operations.sql_query and operation_data.data fields), activate the **AuditFieldCryptEnabled** parameter in the **System Settings → Additional Parameters** of the DataSunrise's web UI. The **AuditFieldCryptoOptions** parameter enables you to specify encryption options to be used.

4. **Note:** You should enable the Dictionary encryption BEFORE creating your target database profile in the DataSunrise's settings. Once you enable encryption, you will not be able to turn it off.

   For encrypting a Dictionary PostgreSQL database, run the DataSunrise's back end with the **DICTIONARY_FIELD_CRYPTO_ENABLED=1** parameter. For example, you can use the following script during DataSunrise installation process (this script creates new or modifies an existing local_settings.db file):

   ```
   ./.AppBackendService
   DICTIONARY_TYPE="postgresql"
   DICTIONARY_HOST="127.0.0.1"
   DICTIONARY_PORT=5432
   DICTIONARY_DB_NAME="ds1"
   DICTIONARY_LOGIN="ds_user"
   DICTIONARY_PASS="qwerty"
   DICTIONARY_FIELD_CRYPTO_ENABLED=1
   FIREWALL_SERVER_NAME="ds"
   FIREWALL_SERVER_HOST="127.0.0.1"
   FIREWALL_SERVER_BACKEND_PORT="11000"
   FIREWALL_SERVER_CORE_PORT="11001"
   FIREWALL_SERVER_BACKEND_HTTPS=1
   FIREWALL_SERVER_CORE_HTTPS=1
   ```

   To specify Dictionary encryption options, use the **DICTIONARY_FIELD_CRYPTO_OPTIONS** back end parameter.
STARTING DATASUNRISE FOR THE FIRST TIME

4.1 STARTING DATASUNRISE

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

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

   ```bash
   sudo service datasunrise start
   ```

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

   ```bash
   sudo service datasunrise stop
   ```

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

4.2 CONNECTING TO THE DATASUNRISE WEB INTERFACE

DataSunrise features a comprehensive web-based interface used for control of all the program's actions.

1. To enter the web interface, perform the following:
   
   • To connect to the Web Console using the HTTPS protocol (by default), open the following address on your web browser:
     
     ```
     https://<DataSunrise_ip_address>:11000
     ```

   **Note:** `<DataSunrise_ip_address>` is the 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 will be:

   ```
   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 (Kerberos only). You can also log in as an "admin" using port 11000 when LDAP authentication is enabled.

   • If you want to connect to DataSunrise using the HTTP protocol, you should activate HTTP in the system settings (System Settings → General → Ports). Then open the following address on your web browser:
     
     ```
     http://DataSunrise_ip_address:11000
     ```

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

2. Your browser will display the "Unsecure connection" prompt because of an untrusted SSL certificate. Follow your browser's prompts to confirm a security exception for the DataSunrise's Web Console.
3. Enter your credentials and click **Login** to enter the web interface.

   **Important:** On first run, use *admin* as the login and the password you've received at the end of the program installation.

### 4.2.1 Changing the Administrator's Password

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

1. Start the Linux CLI.
2. Use the `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 the DataSunrise system service for the changes to take effect.

### 4.2.2 Creating a Certificate for the Web Console

On first run, the web browser used to access the DataSunrise Web Console warns about the unsecure connection and prompts to add a security exception for the Web Console. This issue is caused by the DataSunrise's self-signed SSL certificate. To avoid this, you can use a signed SSL certificate issued by a certain certification authority. For example, you can get such a certificate for free from the 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 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 proper license key into the dedicated text field.
2. Click **Save**. The key will be saved to the DataSunrise Dictionary.

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

At first startup you are prompted to create a target database profile (if there are no profiles existing). You can skip this step to perform it later. Before establishing protection of a certain database, you should specify this database in DataSunrise's settings. To do this, you need to create a target database profile. The profile includes connection details which enable DataSunrise to get database's 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>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical Name text field</td>
<td>Profile's logical name (it is used by DataSunrise as a reference to the database)</td>
</tr>
<tr>
<td>Database Type drop-down list</td>
<td>Target DB type.</td>
</tr>
<tr>
<td>Hostname/IP text field</td>
<td>Target DB's address (host name or IP address)</td>
</tr>
<tr>
<td>Port text field</td>
<td>Database’s port number</td>
</tr>
<tr>
<td>UI element</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Authentication Method drop-down list</td>
<td>User authentication type (regular login/password or Active Directory user authentication)</td>
</tr>
<tr>
<td>Instance text field (for Oracle database only)</td>
<td>Oracle service name or SID</td>
</tr>
<tr>
<td>Default Login text field</td>
<td>DB user name which DataSunrise should use to connect to the target database</td>
</tr>
<tr>
<td>Save Password drop-down list</td>
<td>Don’t save, Save in DataSunrise, Save in CyberArk (refer to subs. Integration with CyberArk AIM on page 70)</td>
</tr>
<tr>
<td>Password text field</td>
<td>DB user password that DataSunrise should use to connect to a database</td>
</tr>
<tr>
<td><strong>Important:</strong> DataSunrise needs user credentials only to get metadata from the target database</td>
<td></td>
</tr>
<tr>
<td>Database text field (for all DB types except Oracle and MySQL)</td>
<td>Name of the target DB. Required to get metadata from the database</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 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 an Oracle service name</td>
</tr>
<tr>
<td>Kerberos Service Name field</td>
<td>Service name for Kerberos connection</td>
</tr>
<tr>
<td>Use Custom Connection String check box</td>
<td>Enable Custom Connection String field (see below)</td>
</tr>
<tr>
<td>Custom Connection String field</td>
<td>Specify a custom connection string for database connection</td>
</tr>
<tr>
<td>IP Version drop-down list</td>
<td>IP protocol version to use for 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>Database keys drop-down list</td>
<td>SSL Key Group that contains required keys for the 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 **Test** button to check connection between the database and DataSunrise server.

3. Specify method of interaction between DataSunrise and the target DB in *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>
<tr>
<td><strong>Action</strong> drop-down list</td>
<td>Select an operating mode DataSunrise should employ to process requests to the target DB (refer to subs. 1.3):</td>
</tr>
<tr>
<td>• Proxy: proxy mode</td>
<td>• Sniffer: sniffer mode</td>
</tr>
<tr>
<td><strong>Network Adapter</strong> drop-down list (for Sniffer mode only)</td>
<td>Select network controller which DataSunrise should use to connect to the target DB</td>
</tr>
<tr>
<td><strong>IP Address</strong> drop-down list (for Proxy mode only)</td>
<td>IP address of the proxy</td>
</tr>
<tr>
<td><strong>Port</strong> text field (for Proxy mode only)</td>
<td>Number of a network port DataSunrise should be listening to</td>
</tr>
<tr>
<td><strong>Accept Only SSL Connections</strong> check box (for Proxy mode only)</td>
<td>Check to disable unencrypted connections</td>
</tr>
</tbody>
</table>

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 target database user account (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 instructions for creating database users with the 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, do the following.

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

   ```
   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:

   ```
   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:

   ```
   ALTER SESSION SET CONTAINER = pdborcl;
   CREATE USER <User name> IDENTIFIED BY <Password>;
   ```
Starting DataSunrise for the first time

**Warning:** In most cases, it is preferable to use a global user to connect 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 the 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 SELECT_CATALOG_ROLE TO <User name>;
   GRANT CONNECT to <User name> CONTAINER=ALL;
   GRANT RESOURCE to <User name>;
   create global temporary table DAF_OBJECTS ON COMMIT DELETE ROWS as select * from CDB_OBJECTS where 1 != 1;
   ```

   **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"."CDB_PROCEDURES" to <User name>;
   GRANT CREATE TABLE to <User name>;
   GRANT SELECT on cdb_types to <User name>;
   GRANT SELECT on cdb_type_attrs to <User name>;
   GRANT CONNECT to <User name> CONTAINER=ALL;
   GRANT SELECT on cdb_types to <User name>;
   GRANT SELECT on cdb_type_attrs to <User name>;
   ```

   **Tip:** You can create the 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;
   ```
4.5.2 CREATING A POSTGRESQL USER

To create a PostgreSQL user, execute the following command:

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

**Note:** The user should be able to get information about database's structure from the following system tables:

- `pg_database`
- `pg_namespace`
- `pg_class`
- `pg_catalog`
- `pg_attribute`
- `pg_user`
- `pg_settings`
- `pg_db_role_setting`

To do this, execute the following query:

```
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 <User_name>;
```

4.5.3 CREATING A NETEZZA DATABASE USER

1. To create a new Netezza user, execute the following query:

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

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

   - For Netezza 6.X:
     
     ```
     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:
     
     ```
     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

1. **Note:** It is an alternative method to update metadata. Use it if you don't want to grant excessive privileges to the default database user. You should use this method BEFORE creating a database profile in the DataSunrise's web UI. As the result you will get a user for metadata update with minimum possible privileges.

   Connect to your MySQL database as a user with sufficient privileges for SELECT from all tables and rights required to create procedures and functions (the root user for example)
2. Download the required procedure script using the following link: https://www.datasunrise.com/doc/mysql_metadata_procedures.sql

3. **Note:** dsproc is schema's name. You can get the actual name from the mysql_procedures.sql file (see previous step).

Execute the script's contents in a MySQL client. The dsproc schema should be created with the script.

4. Create a new MySQL user and grant this user the following privileges:

   ```
   GRANT EXECUTE ON PROCEDURE dsproc.getColumns TO 'User name';
   GRANT EXECUTE ON PROCEDURE dsproc.getGeneratedProperty TO 'User name';
   GRANT EXECUTE ON PROCEDURE dsproc.getFunctions TO 'User name';
   GRANT EXECUTE ON PROCEDURE dsproc.getSchemas TO 'User name';
   GRANT EXECUTE ON PROCEDURE dsproc.getTables TO 'User name';
   GRANT EXECUTE ON PROCEDURE dsproc.getUsers TO 'User name';
   GRANT EXECUTE ON PROCEDURE dsproc.initProcedures TO 'User name';
   FLUSH PRIVILEGES;
   ```

5. Open the DataSunrise's web UI and navigate to System Settings → Additional. Check the MySQLMetadataUseProcedure check box.

6. Now you can create a new database profile in the DataSunrise's Configuration → Databases subsection. Use the details of the MySQL database you've installed the procedure to earlier. Now you should be able to update the database metadata.

### 4.5.5 Creating a Greenplum User

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

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

2. Execute the following query to provide the user with the necessary privileges:

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

### 4.5.6 Creating a Teradata Database User

1. To create a new Teradata user, execute the following query:

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

2. Grant the required privileges to the user by executing the following query:

   ```
   GRANT SELECT
   ON "<Database name>"
   TO "<User name>";
   ```

### 4.5.7 Creating an SAP HANA Database User

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

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

2. Providing the required privileges includes two stages: first a role should receive the privileges to access schema's objects, and then the role is assigned to a user. To grant the required privileges, execute the following queries:
GRANT SELECT ON SCHEMA <Schema name> TO <Role name>;
GRANT ROLE AP_INTERNAL_HANA_SUPPORT 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 the 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 want to use:

```sql
GRANT AUTHENTICATION <Authentication method name> TO <User name>;
```

3. Grant USAGE for all required schemas:

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

4. Grant SELECT for all required schemas:

```sql
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 ALL 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 the 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 the necessary user privileges, run the following script. This query returns a statement for each schema table (DB2 can’t provide access rights to the whole schema). Copy this code to DB2's SQL Editor and run it 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 Creating a MongoDB User

To create a new MongoDB user with minimum possible privileges, execute the following command:

```javascript
use admin
db.createUser(
  {
    user: "<User name>",
    pwd: "<Password>",
    roles: [{ role: "readAnyDatabase", db: "admin" }]
  }
)
```

### 4.5.12 Configuring MS SQL Server Connection

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

1. Run the SQL Server configuration manager utility (it is included in the SQL Server pack). Open SQL Server Network Configuration → Protocols for (DB instance name)
2. Right-click on the TCP/IP protocol name and select Properties from the context menu
3. In the TCP/IP Properties window, on the Protocol tab, set the Yes value for the Enabled parameter. Then open the IP-addresses tab, the IPA11 subsection and set the TCP-port parameter value to 1433. Click OK to close the window
4. Open the SQL Server Services subsection, right-click on the SQL Server (DB instance name) parameter to open the context menu, and click Restart
5. If you’re using any firewall application (including Windows Firewall), you should allow the following inbound connections: TCP/IP, port 1433 and UDP, port 1434
6. When configuring is finished, it is recommended to restart your PC.
7. Connect to the database server with SQL Server Management Studio (SSMS)

**Important:** the SSMS’s Encrypt connection option starts the encryption and server certificate checks on the client’s side (except for SSMS 2016 and higher). Thus, when this option is enabled, the client will not be able to connect to the DataSunrise proxy if the certificate included in proxy.pem or dictionary.db does not include the 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.
Starting DataSunrise for the first time

Important: use SQL Server authentication instead of Windows authentication. Otherwise, refer to subs. 4.5.3

When configuring a database connection, specify the database server’s host name or IP address instead of SPN.

4.5.12.2 Granting necessary privileges to a SQL Server user

To make DataSunrise work correctly with the 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.

   You can download this script at: https://www.datasunrise.com/doc/mssql_user_2.sql

   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 FOR
   SELECT name, CONVERT(NVARCHAR(128), DATABASEPROPERTYEX(name, 'Updateability')) FROM [master].[dbo].[sysdatabases]

   SET @LOGIN = 'bsa'
   SET @SID = ''
   SET @PWD = '1234'
   SET @USER = 'Backend User'

   -- create login
   IF 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.'
               END
         END
         PRINT 'Perhaps it is available for management from another replica (in the case of AlwaysOn, for example).'
         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')
Starting DataSunrise for the first time

```
-- 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 + ']
END-- 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. You can download this script at: https://www.datasunrise.com/doc/mssql_user_3.sql
```

```
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]
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.'
        END
    END
```
4.5.12.3 Enabling “Regexp replace” data masking for SQL Server

The SQL Server database does not support regular expressions but provides a possibility to use external addons. The "Regexp replace" masking function is built as an 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 with MS SQL Server:

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

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

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

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

Important: in either case it will be necessary to grant the user you use to connect to the database, a privilege to run RegexpReplace. You can do it with the following query:

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

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

```csharp
using System;
using Microsoft.SqlServer.Server;
using System.Text.RegularExpressions;

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/doc/ms_sql_regexp_replace.sql](https://www.datasunrise.com/doc/ms_sql_regexp_replace.sql)

Note: more on user-defined functions is 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)
4.5.13 Troubleshooting Connection Failure

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

1. Check the state of proxies using the DataSunrise web UI.
   - Open the DataSunrise's Web Console and go to the **Configurations → Databases** subsection.
   - Click **Edit**.
   - Click the **Test Connection** button.
   - Enter the password and click the **Test All** button.
   If the status of all ports is **OK**, go to the next step.

2. Scan the DataSunrise host with **Telnet Client**.
   Telnet (Terminal Network) is a network protocol that provides a command line interface to communicate with a device.
   Run the "telnet" command with the required hostname and port number as shown below:

   ```
   telnet <DS server's IP address> <DS's port number>
   ```

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

4.6 Additional Proxy Configuration

4.6.1 Changing PostgreSQL Port Number

When configuring the DataSunrise proxy, it will be necessary to change the database port number. It is necessary if the 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 the DataSunrise proxy is deployed on the same host as the database, remote users connecting to the database through proxy will 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 md5
   # IPv6 local connections:
   host all all ::1/128 md5
   ```

3. As a result, the 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 a connection between the client applications and firewall server, you can use the OpenSSL utility to create your own SSL private key.

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

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

2. Ensure that OpenSSL has generated a private key and 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 the DataSunrise installation folder:

   ```bash
   cat server.key > /opt/datasunrise/proxy.pem
   cat server.cert >> /opt/datasunrise/proxy.pem
   chmod 600 /opt/datasunrise/proxy.pem
   chown datasunrise:datasunrise /opt/datasunrise/proxy.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 to work with SSL you need to get a certificate the server delivers to a client during the 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 the full path to the certificate storages in the `Db2KeyStoragePath` and `Db2KeyStashPath` parameters (System Settings → Additional → Parameters).

4. Configure the client workstation for DB2 traffic to go through DataSunrise. It is required to install a trusted DB2 server certificate on the client’s side. 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/t0053518.html](http://www.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/t0053518.html)

4.7.3 CONFIGURING NETWORK DATA ENCRYPTION AND INTEGRITY FOR ORACLE

For using Oracle encryption, it is necessary to store user password hashes. Oracle calculates a session key using user password hashes (stored in sys.user$) and then combines a session key with an encryption key and uses the result for network packet encryption. DataSunrise needs to store these password hashes to calculate and decrypt session keys on DataSunrise’s side. It is necessary to support Oracle network encryption.

**Important:** DataSunrise supports Password-protected Users only at this moment. Global User Authentication and External Service Authentication are not supported.

User password hashes are stored in the `sys.user$` table and you need certain privileges to access it. To grant the necessary privileges, execute the following query as the SYS user:

Oracle 12 and higher:

```sql
grant select on sys.user$ to <user name> container=all;
```
Oracle lower than version 12:

```
grant select on sys.user$ to <user name>;
```

**Note:** loading of password hashes can be done only if the LoadOracleUsers parameter in DataSunrise's Additional Parameters is enabled (it is disabled by default).

### 4.8 Configuring Active Directory Authentication for the Web Console

Both NTLM and Kerberos authentication methods are available. For Kerberos, an SPN should be created, otherwise NTLM is used.

**1. Prepare your client and server.**

- The client's server and the server the DataSunrise is installed on, should be added to the AD domain
- The service should be started by a Local System user
- DNS should be configured and checked. There should be a connection between the client and the server.
- The DNS name should be used to access the web UI. For example:
  
  ```
  https://<myserver>.<mydomain>:11000
  ```

- For Kerberos authentication, an SPN should be created with the following command (It should be created again if the port number or host name is changed):
  ```
  setspn -S HTTP/<myserver>.<mydomain>:11000 <myserver>
  ```

**2. Configure the backend.**

- In **System Settings —> General** of the web UI, select Kerberos in the **Type of Authentication...** drop-down list:

**3. Create a DataSunrise user.** Name it as follows: `<short domain name>\<user name>`. In other words, name the user similarly to the AD user you are going to log into the Web Console as. For example:

  ```
  DB\Administrator
  ```

**Important:** to log in into the web UI using AD user, use port 11000. Use port 12000 to connect to the Web Console using the default "admin" user.
Along with running a single instance of DataSunrise, you can configure multiple servers to implement Failover and Scalability. This feature enables you to run multiple DataSunrise instances on separate servers sharing a common configuration (Dictionary). If some of the servers go offline, other servers keep working guarantying consistent traffic processing without an impact on system availability.

DataSunrise also includes a built-in load balancer to distribute system load among multiple DataSunrise instances in HA mode. DataSunrise also includes Shared Sessions feature which enables performing authentication to the GUI on one of the DataSunrise servers and all other servers will share queries inside the same session. Once a logout/session timeout is occurred on one instance, the session will be closed on all other DataSunrise instances.

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

DataSunrise needs access to the Dictionary database to be able to load the software configuration. Thus, DataSunrise cannot start without a Dictionary. In case a Dictionary database is disabled AFTER DataSunrise has been started, DataSunrise will continue working because configuration had already been loaded.

In HA mode, DataSunrise can continue working without a connection to a remote Dictionary database. Periodically (by default, in 5 minutes from the moment of the last changes made to the Dictionary), DataSunrise creates a Dictionary backup and stores it in the “AF_CONFIG/systemBackupDictionary” folder. Each DataSunrise server creates its own copy of the shared Dictionary. If DataSunrise uses a Dictionary backup for working, further backing up is not performed. The built-in SQLite database is used to create Dictionary backups.

If a main Dictionary can't be accessed, DataSunrise's Backend and Core connect to a backed up Dictionary to use it. The Backend and Core access Dictionaries independently, so it's possible that they use a main and a reserve Dictionary at the same time.

### 5.1 Configuring a DataSunrise Server (Multi-Servers)

A DataSunrise server can be configured during the DataSunrise installation.

1. Start the program installation with the `--remote-config` parameter.

   ```
   sudo ./DataSunrise<version>.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 the common credentials to access the web UI).

   ![Figure 9: Dictionary details](image)

<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>
</tbody>
</table>
3. Specify the details of the current DataSunrise server.

![Server details](image_url)

**Figure 10: Server details**

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server Name</strong></td>
<td>Logical name of DataSunrise instance.</td>
</tr>
<tr>
<td><strong>Server Hostname</strong></td>
<td>IP address or name of the host the current DataSunrise instance is installed on.</td>
</tr>
<tr>
<td><strong>Server Port</strong></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 to the Dictionary database.

5. After configuring is complete, you can see all the available DataSunrise servers (the instances installed on the separate servers and sharing common configuration) in the **System Settings → Servers** subsection.

6. You can also use the following parameters along with the `-remote-config` parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--dictionary-type &lt;mysql</td>
<td>postgresql&gt;</td>
</tr>
<tr>
<td>--dictionary-host</td>
<td>Dictionary database host name or IP address</td>
</tr>
<tr>
<td>--dictionary-port</td>
<td>Dictionary database port number</td>
</tr>
<tr>
<td>--dictionary-database</td>
<td>Dictionary database name</td>
</tr>
<tr>
<td>--dictionary-login</td>
<td>Database user name to use for connection to the Dictionary</td>
</tr>
<tr>
<td>--dictionary-password</td>
<td>User password to use for connection to the Dictionary</td>
</tr>
<tr>
<td>--audit-type &lt;mysql</td>
<td>postgresql</td>
</tr>
<tr>
<td>--audit-host</td>
<td>Audit Storage database host name or IP address</td>
</tr>
<tr>
<td>--audit-port</td>
<td>Audit Storage port number</td>
</tr>
<tr>
<td>--audit-database</td>
<td>Audit Storage database name</td>
</tr>
<tr>
<td>--audit-login</td>
<td>Database user name to use for connection to the Audit Storage</td>
</tr>
<tr>
<td>--audit-password</td>
<td>Password to use for connection to the Audit Storage</td>
</tr>
</tbody>
</table>
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 System Settings → Servers. Select a required server from the list and click Edit>> to access the server settings.
2. Reconfigure the server, if necessary.
Multi-server configuration (High Availability mode)

Figure 11: DataSunrise Server settings

<table>
<thead>
<tr>
<th>Interface Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main Settings</strong></td>
<td></td>
</tr>
<tr>
<td>Logical Name</td>
<td>Logical name of the DataSunrise server (instance)</td>
</tr>
<tr>
<td>Host</td>
<td>IP address of the server the Instance is installed on</td>
</tr>
<tr>
<td>Backend Port</td>
<td>DataSunrise backend port number (used to access the web UI)</td>
</tr>
<tr>
<td>Backend HTTPS</td>
<td>Use HTTPS protocol to access the web UI</td>
</tr>
<tr>
<td>Core Port</td>
<td>DataSunrise Core port number</td>
</tr>
<tr>
<td>Core HTTPS</td>
<td>Use HTTPS protocol to access the Core</td>
</tr>
<tr>
<td><strong>Start, Stop, Restart the Core</strong></td>
<td>Restart the Core process</td>
</tr>
<tr>
<td>Restart</td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td></td>
</tr>
<tr>
<td>Stop</td>
<td></td>
</tr>
</tbody>
</table>

Server Info

- License Type: Trial for database: Unlimited
- License Expiration Date: 2018-04-24
- Version: 4.0.4, 40597, The latest version is installed
- Core State: Active
- Core Uptime: 00:02:34
- Backend Uptime: 00:02:35
- Server Time: 2018-04-23 11:51:52
- Dictionary: SQLite
- OS Type: Windows
- OS Version: Windows NT 6.2 Build 9200
- Machine: x86_64
- Node Name: twitser
- Encoding: English (United States, 1252)
- Server: Dms1
<table>
<thead>
<tr>
<th>Interface Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>Start the Core process (if stopped)</td>
</tr>
<tr>
<td>Stop</td>
<td>Stop the Core process</td>
</tr>
<tr>
<td>System Info</td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>Program version</td>
</tr>
<tr>
<td>License Type</td>
<td>Type of license activated</td>
</tr>
<tr>
<td>The License is expired</td>
<td>License expiration date</td>
</tr>
<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>Core run time</td>
</tr>
<tr>
<td>Backend Up Time</td>
<td>Backend run time</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 DATABASE USER REQUIRED FOR OPERATION OF COMMON DICTIONARY/AUDIT STORAGE

When deploying DataSunrise in the multi-server configuration, a PostgreSQL or MySQL database is used to store 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 to access that data. In general, such a user should have the 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 two databases: Audit Storage and Config Storage:
CREATE DATABASE <Audit Storage database name> character set UTF8 collate utf8_bin;
CREATE DATABASE <Config Storage database name> character set UTF8 collate utf8_bin;

2. Create a new user by executing the following command:

   CREATE USER <User Name> IDENTIFIED BY 'Password';

3. Grant the required privileges to the user:

   USE <Audit Storage database name>;
   GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, CREATE TEMPORARY TABLES, ALTER,
   DROP, INDEX, REFERENCES ON <Audit Storage database name>* TO <User name>;
   USE <Dictionary database name>;
   GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, CREATE TEMPORARY TABLES, ALTER,
   DROP, INDEX, REFERENCES ON <Dictionary database name>* TO <User name>;
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 the DataSunrise proxy using the Active Directory user credentials. DataSunrise maintains the organizational authentication policies of Microsoft Active Directory, Kerberos and LDAP protocols. You can also configure the Active Directory authentication for users of DataSunrise’s Web Console to enhance security and make AD user groups role management easier.

![Diagram of DataSunrise Authentication Proxy](image)

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. Create an AD user and assign principal names with encrypted keys on the domain controller machine.
2. Install an LDAP or Kerberos client.
3. Configure 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 Web Console** *(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 principal 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 hasn't been selected yet, 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 for the new user. It could be a regular user, if it is not required to grant a user some additional privileges. The user account should be active (Account is disabled check box should be unchecked), and the password for the account should be perpetual (Password never expires check box should be 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. The keytab file is used to authenticate to various remote systems using Kerberos without entering the 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 the ktpass tool:

   ```
   ktpass /princ user1_backend@DOMAIN.COM /mapuser user1_backend /pass <PASSWORD> /crypto all /ptype KRB5_NT_PRINCIPAL /out ~/datasunrise.keytab -setupn
   ```

<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><code>&lt;user_account&gt;@&lt;REALM&gt;</code></td>
</tr>
<tr>
<td>/mapuser &lt;user_account&gt;</td>
<td>Maps the name of the Kerberos principal, which is specified by the princ 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 a keytab file.</td>
</tr>
</tbody>
</table>
The entry created in this step should be first in the keytab file, it should be created every time you configure AD user connection to the Web Console 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 to get metadata.

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

The example is given for creating keytab entries to connect to Vertica databases via the DataSunrise proxy using AD. For other databases or Web Console authentication, perform the same command with the 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
```

3. As a 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 Web Console through Kerberos and connect to Vertica databases through a proxy (Delegation for each proxy should be configured. Refer to subs.6.2.3)

4. Transfer the keytab file to the Linux machine where DataSunrise 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 of the owner only by using the following command:
   ```
   sudo chmod 600 /etc/datasunrise.keytab
   ```

6.2.3 CONFIGURING KERBEROS CLIENT

As the next step, you should install and configure the Kerberos client on your machine.

1. Install the client:

   ```
   sudo apt-get install krb5-user libpam-krb5 libpam-ccreds auth-client-config
   ```

2. Edit the `/etc/krb5.conf` file to add the full domain name, domain controller name and the realm parameter:
   Do not leave any comments tagged with "#" in the config file.

   ```
   [libdefaults]
   default_realm = DOMAIN.COM  # domain specific parameter
   (full domain name)
   
   clockskew = 300
   ticket_lifetime = 1d
   forwardable = true
   ```
proxiable           =           true

dns_lookup.realm    =           true

dns_lookup.kdc      =           true

[realms]

dOMAIN.COM = {
  kdc            =       hostname.domain.com   # domain specific parameter
  admin_server   =       hostname.domain.com   # domain specific parameter
  default_domain =       DOMAIN.COM         # domain specific parameter
}

[domain_realm]

.domain.com = DOMAIN.COM  # domain specific parameter (domain name for
dns names)

domain.com = DOMAIN.COM   # domain specific parameter (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
}

6.2.4 Configuring Active Directory Delegation

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

2. In the Properties section, go to the 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 database 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 the user’s attributes (the 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 done using the **Kerberos** or **LDAP** protocols.

The 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, the 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 on database users using LDAP, perform the following.

1. To configure DataSunrise Proxy Authentication you need to create an AD user (an existing one can be used as well) (refer to [subs.6.2.1](#)).
2. Create an LDAP server. Navigate to **System Settings → LDAP** and click **Add LDAP Server**. Fill out the required fields:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical Name</td>
<td>Logical name of the LDAP server’s profile</td>
</tr>
<tr>
<td>Group Attribute</td>
<td>A search filter used to filter user groups by attribute. Used for mapping of AD user groups.</td>
</tr>
<tr>
<td>Host</td>
<td>LDAP server’s host</td>
</tr>
<tr>
<td>Login Type</td>
<td>Server type</td>
</tr>
<tr>
<td>Port</td>
<td>Server’s port number</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Login Custom Format</td>
<td>If you want to know the format for an LDAP login, you need to replace the dots in a DNS name with a comma. I.e: CN=Test.OU=Europe.O=Novell would become: CN=Test,OU=Europe,O=Novell. If you are not using Novell LDAP it would become: CN=Test,OU=Europe,DC=Novell,DC=com. Depending on the domain (DC) you use to authenticate. DataSunrise supports the following patterns: &lt;name&gt;, &lt;domain&gt;, &lt;basedn&gt;, which are auto replaced. For example: o Active Directory: &lt;domain&gt;&lt;name&gt;; o OpenLDAP: cn=&lt;name&gt;, &lt;basedn&gt;</td>
</tr>
<tr>
<td>SSL check box</td>
<td>Use SSL for connection</td>
</tr>
<tr>
<td>Domain</td>
<td>LDAP server domain name. Used to create an LDAP login.</td>
</tr>
<tr>
<td>Login</td>
<td>LDAP user name. Needed for authentication and execution of queries by a privileged account. Used for mapping groups and AD authentication in the web UI</td>
</tr>
<tr>
<td>Base DN</td>
<td>Distinguished Name (DN) - database to search across. DIT (Directory Information tree) to start data search from</td>
</tr>
<tr>
<td>Save Password</td>
<td>Method of saving the LDAP password: • Save in DataSunrise • Retrieve from CyberArk. In this case you should specify CyberArk’s Safe, Folder and Object (fill in the corresponding fields)</td>
</tr>
<tr>
<td>Password (if an LDAP password is saved in DataSunrise)</td>
<td>LDAP user password. Needed for authentication and execution of queries by a privileged account. Used for mapping groups and AD authentication in the web UI</td>
</tr>
<tr>
<td>Is Default check box</td>
<td>Use this LDAP server as a default one</td>
</tr>
<tr>
<td>User Filter</td>
<td>Expression that defines criteria of selection of catalog objects included into the search area defined by the &quot;scope&quot; parameter. Thus, it is a search filter used to search for user attributes</td>
</tr>
</tbody>
</table>

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

Important for MySQL users: There are two password transfer methods available:

1. **sha256_password**: The recommended method of password transfer. Make sure that the `MySQLUseSHA256PasswordMethodForMapping` parameter is checked in the System Settings → Additional subsection.

2. **mysql_clear_password**: use this method if your client application does not support the sha_256_password method. To enable this method, perform the following.
   - Enable the Cleartext Authentication Plugin on the client side:
     ```
     mysql -enable-cleartext-plugin -h <DataSunrise_hostname> --port=3307 -u <AD_user> --password=<password>
     ```
   - Go to System Settings → Additional and uncheck the `MySQLUseSHA256PasswordMethodForMapping` parameter (set 0 value).
**Important:** when the Cleartext Authentication Plugin is used, the passwords will be sent unencrypted, which is not safe unless you use an SSL-encrypted connection.

**Important:** if the MySQLUseSHA256PasswordMethodForMapping parameter is set to "0" and you get the following error "Authentication with 'mysql_clear_password' method requires SSL encryption to transmit password securely. This requirement can be disabled.", you should enable SSL both on the client side and on the server. Or you can disable the LdapMappingRequireClientSideSSL parameter (set "0" value).

### 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 principal and encrypted keys. Refer to [subs. 6.2](#).
2. Install the Kerberos client on your machine and configure the `/etc/kerb5.conf` file as follows.

```plaintext
[libdefaults]
default_realm = DOMAIN.COM  # full domain name
clockskew = 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:** Two methods to transfer tokens are available:

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. Navigate to Configuration → Databases and select an existing database profile.
2. Click Actions → Authentication Proxy Settings
3. Click Enable to enable authentication proxy for the current database.
4. Select the location of a mapping configuration:
   - **DataSunrise Integrated**: in a DataSunrise config file;
   - **CSV**: in an external CSV file (inside the AF_HOME folder)
   - **External Database**: in an external database. To use an external database to store mapping configuration, you need to create a table there. Here's an example of a query that can be used to create such a table (SQLite database):
     
     ```sql
     CREATE TABLE "active_directory_mapping" ( 
     id INTEGER PRIMARY KEY AUTOINCREMENT, 
     instance_id INTEGER NOT NULL, 
     ad_group VARCHAR(1024), 
     ad_user VARCHAR(1024), 
     db_user VARCHAR(1024) NOT NULL, 
     pass_hash VARCHAR(1024), 
     ldap_server_id INT 
     )
     ```

5. Click **Mapping+** to create a new mapping task.

6. Fill out the required fields:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD Type</td>
<td>Single AD user or a group of AD users</td>
</tr>
<tr>
<td>AD Login</td>
<td>Active Directory user name to map database user on</td>
</tr>
<tr>
<td>DB Login</td>
<td>Database user name to map the AD user on</td>
</tr>
<tr>
<td>DB Password</td>
<td>Database password</td>
</tr>
<tr>
<td>Hash Type</td>
<td>Hash type (see Supported Encryption Algorithms, subs. 1)</td>
</tr>
<tr>
<td>LDAP Server</td>
<td>LDAP server to use for mapping</td>
</tr>
</tbody>
</table>

   Click **Save**.

### 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 this, change the following parameters via the DataSunrise's CLI, specifying the LdapUser parameter in the format of `<domain_name>\<AD_username>`.  

   ```bash
   ./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 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>The hostname of the LDAP server.</td>
</tr>
<tr>
<td>LDAP Server Port</td>
<td>`&lt;636</td>
</tr>
</tbody>
</table>
   - Set 636 to configure DataSunrise to use the LDAP port with SSL.
   - Set 389 to configure DataSunrise to use the LDAP port without SSL.
### Parameter | Description
--- | ---
UserMappingType | Enable user mapping. Set the value 1.
LdapSSLMode | `<1 | 0>` Enable/disable authentication with SSL.
- If you’ve specified the port 636, set the value 1. Note that the LDAP server must support SSL connections.
- For port 389, set the value to 0.
LdapBaseDN | 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`
UsersDomainName | Specify the Active Directory domain name. Example: If the AD domain name is `DOMAIN.COM` set the value as `DOMAIN`.

To change the parameters via the DataSunrise’s Web Console, 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 port 636, uncheck for 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 adding an AD user mapping configuration (subs subs 6.3.3).

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

```
./executecommand.sh addDbUserMapping -instance vertica -adGroup <AD_group_name> -dbLogin <DB_user> -dbPassword <DB_password> -hashType MD5
```

### 6.4 Configuring Mapping of AD Users to Database Users via the Web Console

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 the **Config** option if the information about user mapping is stored in the database or the **File** option if the 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>AD Type drop-down list</td>
<td>Select Login for a single AD user and Group for a group of AD users</td>
</tr>
<tr>
<td>AD Login field</td>
<td>Active Directory user’s name</td>
</tr>
<tr>
<td>DB Login field</td>
<td>Name of the database user you want to map the AD user to</td>
</tr>
<tr>
<td>DB Password field</td>
<td>Password of the database user</td>
</tr>
<tr>
<td>UI element</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Hash Type field</td>
<td>Hash type (MD5 or SHA-512)</td>
</tr>
</tbody>
</table>

5. Click **Save**.

### 6.4.1 LDAP USERS CACHE

Normally, when user mapping configured, DataSunrise establishes connections with the AD Controller to get AD logins required for user mapping. When a large number of such connections established, it can cause performance hit. In such case you can use the Cache of LDAP users feature. This feature enables you to add AD logins to a cache so DataSunrise uses this cache to get AD logins required for user mapping. This cache exists temporarily (it depends on the caching settings, see below). To enable the caching, do the following:

1. Navigate to **System Settings → Additional Parameters**.
2. Locate the `LdapLoginCacheTimeout` parameter (you can use the search feature to do that). This parameter's value means a number of seconds an LDAP user cache exists.
3. Set `LdapLoginCacheTimeout` to 900 for example — it will give you 15 minutes to keep your LDAP logins in DataSunrise’s memory to avoid issues associated with performance hit.
### Configuring Client Applications

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 the DataSunrise proxy, perform the following:

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

![Object browser tab](image)

#### Figure 13: Object browser tab

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

![Adding a connection](image)

#### Figure 14: Adding a connection

3. Specify the connection details for the 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 configuring is completed, a new connection will appear.

Figure 16: List of existing connections/servers
7.2 SQL SERVER MANAGEMENT STUDIO (MS SQL SERVER CLIENT)

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

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


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

In the Connect Object Explorer window, enter the 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 to connect to a database</td>
</tr>
<tr>
<td>Password</td>
<td>Password required to connect to a database</td>
</tr>
</tbody>
</table>

5. You can also use the 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 the 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 the connection details.

![Setup New Connection](image)

**Figure 22: Connection details**

<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 IP address</td>
</tr>
<tr>
<td>Port</td>
<td>Port number of DataSunrise 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.
Figure 23: New connection icon

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 keep open the possibility of viewing the contents of old audit.db files. You can configure automatic rotation of audit.db files or do it manually.

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 the 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 the 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 the **Enable** button to convert audit.db to a new format (split it into two files)
3. Click **Manually** to create a new audit.db file to write data to. All the 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>Sequence number</td>
</tr>
</tbody>
</table>
4. **Note:** you can use an audit file during a current DataSunrise user session only. When the session is closed (a logout occurred), DataSunrise automatically switches the active file to a newest available file.

Select the 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 the *datasunrise* system service:
   ```
   sudo service datasunrise stop
   ```
2. Log in as the *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 the *datasunrise* user out and start the datasunrise system service:
   ```
   sudo service datasunrise start
   ```

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>Is the <em>audit.db</em> file active or not</td>
</tr>
<tr>
<td>Reason</td>
<td>Why the <em>audit.db</em> file was created</td>
</tr>
<tr>
<td>Rotate Time</td>
<td>Time at which the <em>audit.db</em> file was created</td>
</tr>
<tr>
<td>End Time</td>
<td>Time at which the <em>audit.db</em> file became non-active</td>
</tr>
</tbody>
</table>
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 the 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 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 FIREWALL INSIDE AZURE CLOUD FOR MAINTENANCE OF SAAS SQL AZURE

Azure SQL and the Always-On cluster use the same mechanism for client redirecting. When connecting to SaaS SQL Azure, if a client is inside the Azure subnet, then SQL Azure can redirect the client to the service (dynamic) servers to balance the network load. In such a case right after authorization the server will return to the client the 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 a reconnection, DataSunrise replaces the address of the service server with the address of a proxy that services it.

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

```
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 the service server’s address and 10.1.0.6:14033 is the address of DataSunrise proxy that maintains this service server.

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

- If the MsSqlRedirectsDisable option is disabled (by default), a proxy will be created automatically in the current instance (and the 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 the client should add the proxy on this host manually (or it’s been done already) to make DataSunrise control the client connections. Otherwise, DataSunrise will loose control over the client connection. For a cluster with AlwaysOn enabled it is possible to configure redirection to Readonly-replicas, that’s why if the redirecting host is already configured on DataSunrise, we will see this host in the 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 a target server interface (it is cd164f04fd1f.tr27.westus1-a.worker.database.windows.net:11082 in our case);
• Add a proxy on this interface. When using the standard templates for a hostname (0.0.0.0 or 0:0:0:0:0:0:0:0) of such a proxy, DataSunrise will return the client the address of an available non-local interface from DataSunrise’s host as a redirect address.

9.3 RESTORING CONFIGURATION IF LOCAL_SETTINGS.DB IS LOST

When using an HA configuration, if one of the DataSunrise servers needs to be transferred to another computer or local_settings.db is lost, the DataSunrise’s configuration will change 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 number corresponds to the <SERVER_ID> parameter’s value you will be using in the next 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 the server you want to restore access to (see step 1). Note that all the values of a parameter should be written without quotes.
INTEGRATION WITH CYBERARK AIM

DataSunrise can be integrated with CyberArk’s AIM to store database passwords in the CyberArk’s Vault and retrieve them on demand.

10.1 AIM INSTALLATION

Refer to “Credential Provider and ASCP Implementation Guide” for CyberArk Credential Provider.

It is recommended to enable AIM cache as it will increase performance when retrieving passwords from the local cache. It is highly recommended to select a cache refresh interval which is less than 5 minutes. Otherwise, this could lead to inaccurate password cache.

10.2 AIM CONFIGURATION. DEFINING THE APPLICATION ID (APPID) AND AUTHENTICATION DETAILS

To define the application, refer to the following instruction or define it manually via CyberArk’s PVWA (Password Vault Web Access) Interface:

1. Being logged in as a user with the rights to manage applications, on the Applications tab, click Add Application to open the Add Application page.
2. Specify the following details:
   • In the Name field, specify the unique identifier of the application (“DataSunriseDBSecurity”)
   • In Description, write a short description of the application that will help you to identify it.
   • In the Business owner section, specify contact information about the application’s business owner.
   • In the lowest section, specify the Location of the application in the Vault hierarchy. If Location is not specified, the application will be added to the same Location as the user which created this application.
3. Click Add and the application will be added and displayed on the Application Details page.
4. Specify the application’s Authentication details. This information enables the Credential Provider to check certain application characteristics before retrieving the application’s password. DataSunrise recommends to specify the OS user and application path. Refer to “DataSunrise Database Security Suite – Administration Guide (Linux)” and “DataSunrise Database Security Suite – Administration Guide (Windows)” respectively, see “Program installation” section. The default settings for Linux: the OS user name is “datasunrise”, application path is “/opt/datasunrise”; the default parameters for Windows: the OS user name is “Local System” and application path is “C:\Program Files\DataSunrise Database Security Suite”.
   To enable Credential Provider, check the application’s Authentication details:
   • On the “Authentication” tab, click Add; a drop-down list with the authentication characteristics included will be displayed.
   • Select an authentication characteristic to specify.
5. Specify an OS user:
   • Select the OS user; the “Add Operating System User Authentication” window will be displayed.
   • Specify the name of OS user who will run the application, then click Add; the OS user will be listed on the “Authentication” tab.
6. Specify the application path:
   • Select Path; the Add Path Authentication window will be displayed
   • Specify the path to the application.
To indicate that the specified path is a folder, check "Path is Folder".
To allow the internal scripts to retrieve the application password for this application, select "Allow internal scripts to request credentials on behalf of this application ID".
Click Add; the Path will be added as an authentication characteristic with the information that you’ve specified.

7. Specify a hash
- Run the AIMGetAppInfo utility to calculate the application’s unique hash.
- Copy the hash value that is returned by the utility.
- In the PVWA, select Hash; the Add Hash window will be displayed.
- In the Hash field, paste the application’s unique hash value, or specify multiple hash values separated with a semi-colon. You can add comments by using "#" after the hash value. For example, "OE883B7OD5B6E3EE37D37198049C9507C8383DB6 #app2".
- Click Add; the Hash will be added as an authentication characteristic with the information that you’ve specified.

8. Specify the application’s Allowed Machines. This information enables the Credential Provider to ensure that only applications that run from the specified machines can access their passwords.
- On the **Allowed Machines** tab, click Add; the **Add allowed machine** window will be displayed.
- Specify IP/hostname/DNS of the machine where the application will run and will request passwords, then click Add; the IP address is listed on the **Allowed Machines** tab. Make sure the allowed servers include all the mid-tier servers or all the endpoints where the AIM Credential Providers were installed.

### 10.3 Provisioning Account and Settings Permission for Application Access

For the application to perform its functionality or tasks, the application needs to have access to the particular existing accounts, or new accounts to be provisioned in CyberArk Vault (Step 1). Once the accounts are managed by CyberArk, make sure to setup the access to both the application and CyberArk Application Password Providers serving the Application (Step 2).

1. In “Password Safe”, provision the privileged accounts that will be required by the application. You can do this either manually or automatically:
   - Manually - add the accounts manually, one at a time and specify all the account details.
   - Automatically - add multiple accounts using the “Password Upload” feature. Note that for this step, you will require the Add accounts authorization in Password Safe.

   **Note:** for more information about adding and managing privileged accounts, refer to “Privileged Account Security Implementation Guide”.

2. Add Credential Provider and the application users as the members of Password Safes where the application passwords are stored. This can either be done manually on the Safes tab, or by specifying the Safe names in the CSV file for adding multiple applications.

3. Add the Provider user as a “Safe Member” with the following privileges:
   - List accounts
   - Retrieve accounts
   - View Safe Members

4. Add the application (DataSunriseDBSecurity) as a Safe Member with the following authorizations:
   - Retrieve accounts

To enable Credential Provider, check the application’s Authentication details:
• On the “Authentication” tab, click Add; a drop-down list with the authentication characteristics included will be displayed.
• Select an authentication characteristic to specify.

5. If your environment is configured for dual control:

• In the PIM-PSM environments (v7.2 and lower), if Safe is configured to require confirmation from authorized users before passwords can be retrieved, give the Provider user and the application the following permission:
  - Access Safe without Confirmation
• In the Privileged Account Security solutions (v8.0 and higher), when working with dual control, the Provider user has access without confirmation, thus, it is not necessary to set this permission.

6. **Note:** for more information about configuring Safe Members, refer to the “Privileged Account Security Implementation Guide”.

If Safe is configured for object level access, make sure that both the Provider user and application have access to the password(s) to retrieve.

### 10.4 DATASUNRISE INSTALLATION AND CONFIGURATION

1. Configure a DataSunrise database instance in the **Configuration —> Databases** subsection.
   • For your database instance, specify the CyberArk Vault credentials to retrieve a specific password.
   • In the **Save Password** drop-down list, choose the “Retrieve from CyberArk” option and enter CyberArk Safe, Folder and Object parameters for this specific database password to the fields below.

2. Add the Credential Provider and application users as members of the Password Safes where the application passwords are stored. This can either be done manually on the **Safes** tab, or by specifying the Safe names in the CSV file for adding multiple applications.

3. Use the **Save** button to save the vault access credentials in the DataSunrise configuration. When Save is clicked, DataSunrise will perform a test password retrieval for the specified vault parameters. In case of the test failure, DataSunrise will report with the error message “Cannot retrieve password. Please make sure you have entered the correct CyberArk Vault parameters”. Please also make sure you’ve installed AIM properly since DataSunrise depends on the correct AIM Credentials Provider installation.
CREATING A HEALTH CHECK

DataSunrise enables you to create a Health Check for cloud or local services to notify about failing DataSunrise instances. Health checking works in parallel mode which increases check speed and stability when working with multiple proxies and nodes. A separate task is created which is called independently of Health Check queries and serves to save the check results in cache. Then the Health Check uses the cached results.

To create a health check, you can use the following URLs. Paste them in the field that defines the URL to be checked (Ping Path field when creating a health check)

1. This URL enables you to check all proxies for the specified instance:

   \[ \text{https://<datasunrise server name>:11000/healthcheck/instance?inst_name=<instance name>&db_name=<database name>&db_login=<database user>&db_password=<database password>} \]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;datasunrise server name&gt;</td>
<td>IP address or host name of DataSunrise’s server, 11000 is the port number of the DataSunrise’s Web Console</td>
</tr>
<tr>
<td>&lt;instance name&gt;</td>
<td>Name of DataSunrise instance (for example, an AWS instance)</td>
</tr>
<tr>
<td>&lt;database name&gt;</td>
<td>Name of the database configured to be used with DataSunrise (RDS database for example)</td>
</tr>
<tr>
<td>&lt;database user&gt;</td>
<td>Database user name (login) you can use to connect to your database</td>
</tr>
<tr>
<td>&lt;database password&gt;</td>
<td>Database user password you can use to connect to your database</td>
</tr>
</tbody>
</table>

   **Note:** If the login and password are saved for a certain instance, you can skip them in the URL, or you will get error 500 with a corresponding message. The server will return warning 200 if success.

When checking all instances the health checker checks all DataSunrise proxies and if a proxy does not respond, it would return error 500 with a corresponding message.

2. You can use the following URL to check all proxies on all instances.

   /healthcheck/all_instances

   **Note:** if login/password are not saved to the instance’s settings, this particular instance will not be checked.

3. General health check (checks all servers):

   /healthcheck/general
AWS CLOUD WATCH CUSTOM METRICS

DataSunrise can send custom metrics to AWS's Cloud Watch. To do it, perform the following:

1. Create an AWS role that includes the required policies:
   - Navigate to AWS Console → IAM → Policies. Create a new policy. Switch to JSON and paste the following code into the JSON field:

     ```json
     {
     "Version": "2012-10-17",
     "Statement": [
     {
     "Action": [
     "cloudwatch:PutMetricData",
     "ec2:DescribeTags"
     ],
     "Effect": "Allow",
     "Resource": [
     "*"
     ]
     }
     ]
     }
     ```
   - Navigate to Roles. Create a new role → AWS EC2
   - On the Attach permissions policies page, select your policy from the list (use Search). Save the Role

2. Enable EnableAWSMetrics parameter in the DataSunrise's Settings → Additional

3. Navigate to EC2 Console → Cloud Watch → Blowse Metrics → DataSunrise to view the metrics:

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuditProcessingSpeed</td>
<td>Processing of queries by audit journal speed (operations/sec)</td>
</tr>
<tr>
<td>AuditQueueLength</td>
<td>Length of the audit journal queries queue (queries)</td>
</tr>
<tr>
<td>ProxyMessageHandlerQueueLength</td>
<td>Length of the proxy queries queue (queries)</td>
</tr>
<tr>
<td>SnifferMessageHandlerQueueLength</td>
<td>Length of the sniffer queries queue (queries)</td>
</tr>
<tr>
<td>CoreThreadCount</td>
<td>Number of Core threads</td>
</tr>
<tr>
<td>TrafficBufferPoolFreeObjects</td>
<td>Number of free blocks in the traffic buffer</td>
</tr>
<tr>
<td>TrafficBufferPoolBalance</td>
<td>Number of used blocks in the traffic buffer</td>
</tr>
<tr>
<td>Antlr3ParserPoolSize</td>
<td>The overall volume of the Antlr parser (kb)</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>Antlr3ParserPoolUsed</td>
<td>Volume of the Antlr parser used (kb)</td>
</tr>
<tr>
<td>Antlr3TokensPoolSize</td>
<td>Overall volume of the Antlr tokenizer (kb)</td>
</tr>
<tr>
<td>Antlr3TokensPoolUsed</td>
<td>Volume of the Antlr tokenizer used (kb)</td>
</tr>
<tr>
<td>Antlr3StrPoolSize</td>
<td>Overall volume of strings in Antlr parser memory (b)</td>
</tr>
<tr>
<td>Antlr3StrPoolUsed</td>
<td>Used volume of strings in Antlr parser memory (b)</td>
</tr>
<tr>
<td>Antlr3ParserCommentsPoolSize</td>
<td>Overall volume of the Antlr commentary parser (kb)</td>
</tr>
<tr>
<td>Antlr3ParserCommentsPoolUsed</td>
<td>Volume of the Antlr commentary parser used (kb)</td>
</tr>
<tr>
<td>Antlr3TokensCommentsPoolSize</td>
<td>Overall volume of the Antlr commentary tokenizer (kb)</td>
</tr>
<tr>
<td>Antlr3TokensCommentsPoolUsed</td>
<td>Volume of the Antlr commentary tokenizer used (kb)</td>
</tr>
<tr>
<td>Antlr3StrCommentsPoolSize</td>
<td>Overall volume of strings in Antlr commentary parser memory (b)</td>
</tr>
<tr>
<td>Antlr3StrCommentsPoolUsed</td>
<td>Used volume of strings in Antlr commentary parser memory (b)</td>
</tr>
<tr>
<td>CoreVirtualMemoryUsage</td>
<td>Memory volume used by the Core (Mb)</td>
</tr>
<tr>
<td>BackendVirtualMemoryUsage</td>
<td>Memory volume used by the UI (Mb)</td>
</tr>
<tr>
<td>ProxyOperationsSpeed</td>
<td>Number of database operations per second</td>
</tr>
<tr>
<td>ProxyReadTrafficSpeed</td>
<td>Read speed (from DB to client)</td>
</tr>
<tr>
<td>ProxyWriteTrafficSpeed</td>
<td>Write speed (from client to DB)</td>
</tr>
<tr>
<td>AuditDiscFreeSpace</td>
<td>Volume of free space in the audit journal file system (Mb)</td>
</tr>
<tr>
<td>LogsDiscFreeSpace</td>
<td>Volume of free space in the logs file system (Mb)</td>
</tr>
<tr>
<td>ProxyExecutionsSpeed</td>
<td>Number of DB executions per second</td>
</tr>
</tbody>
</table>
AUTHENTICATION OF AD USERS TO THE DATASUNRISE'S WEB UI USING LDAP

CONFIGURING AUTHENTICATION OF USERS TO THE WEB UI USING LDAP

1. Create at least one LDAP server profile in the DataSunrise's settings (System Settings → LDAP). Refer to subs. 4.1.10 of the DataSunrise User Guide for details.

2. Navigate to System Settings → General and in the UI Parameters subsection, select LDAP in the Type of Authentication to DataSunrise UI drop-down list. This enables you to log into the web UI using an LDAP account.

3. Then you can authenticate in the web UI by a user name or group.
   - To authenticate by the user name, navigate to Access Control → Users and create a new DataSunrise user (refer to subs. 4.1.7.1 of the DataSunrise User Guide for details). Check the Active Directory Auth check box in the user’s settings. Then on the login page specify the user name and password saved in LDAP. DataSunrise will check all available LDAP servers and try to connect to them using the specified login and password to authenticate the user.
   - Authentication by group is used when a user not known to the system is trying to login. Note that Group Attribute (System Settings → LDAP) should include a correct attribute name and a role (Access Control → Roles) and should include a Group DN. The backend will try to connect to all available LDAP servers and get an attribute specified in the Group Attribute field. It creates a user and grants it certain privileges according to the Active Directory Path names. Authentication will be performed as “By user name” hereafter, because a user already exists.
SINGLE SIGN-ON IN DATASUNRISE

SINGLE SIGN-ON

DataSunrise features Single sign-on (SSO) feature for accessing the Web Console. For now, DataSunrise supports OpenID and SAML technologies.

CONFIGURING SSO AUTHENTICATION BASED ON OPENID (USING OKTA AS THE SERVICE PROVIDER)

This example describes configuring of SSO authentication provided by Okta. To enable OpenID authentication to the DataSunrise’s web UI, do the following:


![Figure 25: Add Application](image)

2. At the Create a New Application tab, select Web as Platform, and OpenID Connect as Sign on Method.

![Figure 26: Creating a new App](image)

3. At the next tab, set application name (any) and input the following URL:

   ```
   https://<DataSunrise_IP_address>:11000/sso_endpoint
   ```

   For example:

   ```
   https://127.0.0.1:11000/sso_endpoint
   https://localhost:11000/sso_endpoint
   ```
4. Navigate to **Assign Applications** and assign your application to your Okta user

5. Go to the following page: [https://developer.okta.com/docs/api/resources/oidc#request-example-3](https://developer.okta.com/docs/api/resources/oidc#request-example-3). See Request Example. Copy the first part of the query (for example):

   ```
   https://datasunriseantony.okta.com/oath2/${authServerId}/.well-known/openid-configuration
   ```

   And delete the middle part of it:

   ```
   oauth2/${authServerId}
   ```

   The query should look like the following:

   ```
   https://datasunriseantony.okta.com/.well-known/openid-configuration
   ```

   Open this query in your web browser for query results.

   Note that you will need the following values from there:

   - **authorization_endpoint**
   - **token_endpoint**
   - **jwks_uri**

6. Go to Okta's Dashboard and navigate to **Application → Your App → General → Client Credentials**. Note the **Client ID** and **Client secret**. You will need these parameters' values.

![Figure 28: Client Credentials](image)

7. Enter the DataSunrise's web UI. Note that you need to specify the full IP address instead of just a host name. For example:

   ```
   https://127.0.0.1:11000
   ```

   Navigate to **System Settings → SSO**, click **Add SSO Service**.

8. Input a logical name (any), select **OpenID Connect** in the **SSO Service Type**. Input the following values:
### Single Sign-On in DataSunrise

#### Parameter in the Web Console

<table>
<thead>
<tr>
<th>Parameter in the Web Console</th>
<th>Corresponding parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization Token Endpoint URL</td>
<td>authorization_endpoint (see step 5)</td>
</tr>
<tr>
<td>Token Endpoint URL</td>
<td>token_endpoint (see step 5)</td>
</tr>
<tr>
<td>Token Keys Endpoint URL</td>
<td>jwks_uri (see step 5)</td>
</tr>
<tr>
<td>OIDC Client ID</td>
<td>Client ID (see step 6)</td>
</tr>
<tr>
<td>OIDC Client Secret</td>
<td>Client secret (see step 6)</td>
</tr>
</tbody>
</table>

Save the profile.

9. Navigate to **Access Control → Your user** (admin for example) → **Single Sign-On Connections**. In the **Login With** drop-down list, select the SSO Service created in the previous steps and click **Add Connection**.

10. You will be redirected to the logon screen of the web UI. Input OpenID credentials to be logged to the UI.

### Configuring SSO Authentication Based on SAML (Using Okta as the Service Provider)

This example describes configuring of SSO authentication provided by Okta. To enable SAML authentication to the DataSunrise’s web UI, do the following:

1. Register in the Okta service. Navigate to **Dashboard → Add Applications → click Create New App**.

![Figure 29: Adding Application](image)

2. At the **Create a New Application** tab, select **Web as Platform**, and **SAML 2.0 as Sign on Method**

3. At the next tab, set application name (any) and input the following URL into **Single Sign on URL and Audience URI (SP Entity ID)**:

   ```
   https://<DataSunrise_IP_address>:11000/sso_endpoint
   ```

   For example:

   ```
   https://localhost:11000/sso_endpoint
   ```
4. Navigate to Assign Applications and assign your application to your Okta user. A new page will open. Note the Identity Provider Single Sign-On URL. You will need this parameter's value.

5. Enter the DataSunrise's web UI. Navigate to System Settings → SSO, click Add SSO Service.

6. Input a logical name (any), select SAML in the SSO Service Type. Input the "Identity Provider Single Sign-On URL" (see step 4) into the Authorization Token Endpoint URL field. Save the profile.

7. Navigate to Access Control → Your user (admin for example) → Single Sign-On Connections. In the Login With drop-down list, select the SSO Service created in the previous steps and click Add Connection.

8. You will be redirected to the logon screen of the web UI. Input Okta credentials to be logged to the UI.
FREQUENTLY ASKED QUESTIONS

This section describes the most common issues DataSunrise users face.

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

2. I'm trying to run PostgreSQL, but the database connection fails: “[unixODBC] Missing server name, port, or database name in call to CC_connect.” (error code 201).
   - Check the ODBC driver availability by executing the following command:

     odbcinst -q -d

   Locate the ODBC.ini file and configure it in the following way:

     [postgres_i]
     Description = Postgres Database
     Driver = PostgreSQL
     Database = postgres
     Servername = 127.0.0.1
     Port = 5432

   Check the PostgreSQL connection by executing the following command:

     isql postgres_i username password

3. I'm trying to run DataSunrise but getting the 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 the ODBC SYSINI, ODBC INSTINI or ODBCINI environment variables to pinpoint the 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 the 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 the program update, but it displays the “Internal System Error” message.
   - Most likely, you kept the web interface tab opened on your browser while updating the firewall. Log out the web interface if necessary and press Ctrl + F5 to refresh the page.

6. I’m trying to establish the connection between DataSunrise and a MySQL database, but it fails because of the missing ODBC MySQL driver.
   - Certain Linux-type operating systems don’t add the MySQL driver parameters to the odbcinst.ini file. You should do it manually.
   - If necessary, install the MySQL ODBC driver by running the following commands:

     For Debian and Ubuntu:
sudo apt-get install libmyodbc libodbc1

- For CentOS, Red-Hat and Fedora:
  sudo yum install mysql-connector-odbc

- Edit the odbcinst.ini file. Run the following command:
  sudo nano /etc/odbcinst.ini

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

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

sudo odbcinst –I –d –f /etc/odbcinst.ini

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

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

- Define thr odbc.so.1 file location by running the following command
  ```
  locate libodbc.so.1
  ```
- If libcrypt.so.10 and libssl.so.10 are available in the DataSunrise installation folder, execute the following command:
  ```
  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 the Linux system folder where `libodbc.so.1` is located and `/opt/datasunrise/` is the DataSunrise installation folder.

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

```
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 the Linux system folder where `libodbc.so.1` and `libodbc.so.2` are located.

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

- Install libcap-progs. To do this, run the following command:
  ```
  sudo zypper install libcap-progs
  ```
9. When I’m trying to run DataSunrise in the sniffer mode, it displays the message: “Impossible to parse the SSL connection in the sniffer mode”.
   • To run the firewall in the sniffer mode, you should disable SSL support in your client application settings (SSL Mode -> Disable). You can also switch the application’s SSL Mode to “Allow” or “Prefer”, but disable SSL support in the database server settings first.

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

11. When connecting to Aurora DB, MySQL the 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. A DataSunrise installation is aborted with the **Permission denied** error:

    ```
    [test@HVLB001 ~]$ ./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 the temp folder. Execute the following command:

    ```
    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 Web Console.
   • You can set a new administrator password. Use the Terminal to run the DataSunrise’s [AppBackendService](https://example.com) file with the `set_admin_password` parameter. For example: `sudo ./AppBackendService set_admin_password=new_password`. To apply the new password, restart the DATASUNRISE system service (refer to subs. 4.1).

14. I’m using a 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 the UNIX socket for connection instead of TCP. Specify the full IP address of the database and ensure that your client application uses the TCP connection. Refer to the next pages, if necessary: [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), [how to force SQL Server authentication](http://www.computerhope.com/issues/ch001079.htm), [how to force Windows authentication](http://example.com)

15. I’m using a 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 (the 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 the actual DB server’s IP address or host name instead of server’s SPN.

16. I’ve updated DataSunrise via the Web Console and get the following error:

    ```
    (17279|140037881128832) PCAP - pcapOpen: [ERROR] Can not activate pcap object.
    Pcap error: socket: Operation not permitted
    ```

   Also, I’m having a problem with the sniffer:

    ```
    (17279|140037881128832) DS_31012E: The task of the sniffer cannot be initialized.
    ```
• These problems occur because DataSunrise cannot install the required "file capabilities" (cap_net_raw and cap_net_admin=eip to be exact). To enable these capabilities, execute the following command:

```bash
sudo setcap 'cap_net_raw,cap_net_admin=eip' /opt/datasunrise/AppFirewallCore
```

17. I'm getting the following warning: "The free disk space limit for audit has been reached. The current free disc space is XXX MB. The disk space limit is 10240 MB".

• If you want to decrease the disk space threshold for this warning, navigate to System Settings → Additional and change the "LogsDiscFreeSpaceLimit" parameter's value from 10240 to 1024 Mb for example,

18. I'm getting the following notification: "Reached the limit on delayed packets".

• This notification is displayed when the sniffer has captured large amount of traffic in the SSL sessions started before the DataSunrise service was started. By default, the volume of captured traffic should not exceed 10 Mb (the pnMsSqlDelayedPacketsLimit parameter).

• Sometimes this notification can be displayed if there is a huge load on the pcap driver. Thus the sniffer can capture too much of delayed traffic. In this case you need to increase the pnMsSqlDelayedPacketsLimit parameter’s value.

19. I've updated DataSunrise and I get the following error: "PROCEDURE dsproc_<version>.initProcedures does not exist"

• Now DataSunrise uses a new method of getting metadata. Do the following steps again: subs 2.5.4
AUDIT STORAGE

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

CLI

The abbreviation for the "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

A 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

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

DB

The abbreviation for a "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 to an external database when deploying the software in the multi-server configuration. In this case the servers use a common Dictionary.

DYNAMIC DATA MASKING

A 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.
WEB CONSOLE
A dedicated web-based DataSunrise management tool supplied with the software.

PROXY MODE
A DataSunrise deployment option. The proxy mode activates all DataSunrise's functionality. In this configuration DataSunrise is deployed as a proxy between a target database and its client applications.

RDBMS
The abbreviation for "Relational Database Management System".

SNIFFER MODE
A DataSunrise deployment option. The 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
The abbreviation for "SQL Server Management Studio". A dedicated database management tool supplied with Microsoft SQL Server databases.

STATIC DATA MASKING
A 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 the DataSunrise’s web UI.