# SecuGATE Common Criteria Configuration Guide

SecuSUITE for Government 5.0

Version 0.7

# Contents

# 1. Acronyms

| Term | Definition |
|---|---|
| CLI | Command Line Interface (Local console and SSH access) |
| CRL | Certificate revocation list |
| CSR | Certificate signing request |
| ECC | Elliptic curve cryptography |
| MAC | Message authentication code |
| NDCPP | Network devices collaborative protection profile |
| NTP | Network time protocol |
| RSA | Rivest-Shamir-Adleman cryptosystem |
| RTP | Real-time transport protocol |
| SCA | Secure client authentication |
| SGLVN | SecuGATE LVN |
| SSH | Secure shell |
| TOE | Target of evaluation |
| TSF | Target of evaluation security function |

# 2. References

| Ref. | Document |
|---|---|
| [A] | Blackberry SecuGATE v.5.0 Security Target |
| [B] | Collaborative protection profile for network devices (NDcPP21) |
| [C] | extended package for enterprise session controller (ESC) Version 1.0 |
| [D] | SecuGATE installation guide |

# 3. Document history

| Version | Date | Status | Author | Comments |
|---|---|---|---|---|
| 0.7 | 28 July 2022 | Final | Secusmart | Cleanup for publication |

# 4. Introduction

1

## 4.1 Purpose

The SecuSUITE SecuGATE LVN server common criteria configuration guide describes TOE functions and interfaces, for the purposes of configuring and operating the TOE in the evaluated configuration.

This document has been developed as part of the BlackBerry SecuGATE v5.0 common criteria (CC) documentation suite. The security functions under evaluation are defined in the Security Target.

The Security Target document as well as the document at hand and the assurance reports can be found on the NIAP web page: https://www.niap-ccevs.org/Product/.

## 4.2 Audience

This document is intended for TOE administrators. TOE administrators are assumed to have knowledge of the BlackBerry SecuSUITE solution and the server system.

It is also assumed that the reader has a general understanding of the underlying technologies and concepts, such as server-side network configurations, X509 certificates, and VoIP features.

In addition, the reader needs a general understanding of managing Unix-based systems on a command line.

## 4.3 Operational Environment

TOE supports the following non-TOE components in its operational environment:

**Table 1: Operational environment components**

| Component | Required | Usage/purpose |
|---|---|---|
| Syslog server | No | The TOE sends system and audit logs to a remote syslog server. The syslog server must conform to RFC 5424 |
| CRL distribution point | No | Provides CRLs, which are used to verify X.509 certificates. The CRL distribution point must conform to RFC 3280. The CRL distribution point must publish CRLs in DER format. |
| PBX/SIP server(s) | No | TOE may use additional SIP servers on the network to allow forwarded calls via a SIP trunk. Additional SIP servers must conform to RFC 3261. |
| External CA | No | TOE may use an external CA for certificate management. External CAs must conform to enrollment over secure transport (EST) RFC 7030- |
| NTPv4 | Yes | NTPv4 server that TOE uses for correct time stamps. NTPv4 servers must conform to RFC 5905, and they must support SHA-256-based authentication. |
| ESXi | Yes | TOE is installed on VMWare ESXi hypervisor. Refer to [A] where specific models and ESXi versions that were evaluated are identified. |

| Component | Required | Usage/purpose |
|---|---|---|
| SecuSUITE mobile clients | Yes | SecuSUITE VoIP endpoints that use TOE's SIP and SCA services. The version of the SecuSUITE mobile clients must be 5.0 or compatible. |

## 4.4 Assumptions

The following guidance should be followed to uphold the security objectives for the operational environment:

**Table 2: Environment and Assumptions**

| Objective | Guidance |
|---|---|
| There are no general-purpose computing capabilities (for example, compilers or user applications) available on the server, other than the services that are necessary for the operation, administration, and support of the server. | Do not install other software on the server. |
| Physical security, commensurate with the value of the server and the data it contains, is provided by the environment. | Ensure that the server is hosted in a physically secure environment, such as a locked server room. |
| Server administrators are trusted to follow and apply all administrator guidance in a trustworthy manner. | Ensure that administrators are trustworthy – for example, implement background checks or similar controls. |
| The server does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment. | The SecuSUITE Secure Call client implements end-to-end encryption and should be used in conjunction with the server. |
| The server firmware and software is updated by an administrator on a regular basis, in response to the release of product updates due to known vulnerabilities. | Apply updates regularly. |
| The administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside. | Administrators should take care not to disclose credentials and ensure that SSH private keys are stored securely. |

# 5. Administrative roles



The TOE can be accessed through a command line interface (CLI) on the local console, a CLI available remotely through SSHv2 or an HTTPS protected Web UI. The remote network management communication paths are protected against modification and disclosure using SSHv2 and HTTPS.

The following table lists the different roles, default login accounts (if existing) and the access method:

**Table 3: Administrative Roles**

| Role | User Accounts Type | Description | Default login credentials | Local access method | Remote access method |
|---|---|---|---|---|---|
| Super Admin | Root | Predefined administration account reserved for disaster recovery or special use cases beyond the evaluated configuration. This account can become root and must not be used in the evaluated configuration and must be initialized with a new password and locked away during system setup. | superadmin/ superadmin | Console | SSH |
| Security administrator | Default Security admin | Predefined administration account used for initial system configuration and recovery the CLI access. Used to set-up the initial dedicated user accounts for system administration and monitoring during normal operations. | secuadmin/ secuadmin | Console | SSH |
| | Console user | Dedicated administration account with CLI access. Console users are initially created via the 'secuadmin' account and used for normal system administration and monitoring. | As assigned by security administrator | Console | SSH |
| Web UI admin | Default UI admin | Predefined default admin account used for initial system set-up and recovery. Used to set-up the dedicated user accounts for administration of the TOE via the Web UI | admin/ admin | NA | Web UI |
| | Web UI User | Dedicated administration account created by the default admin account. | Assigned by default admin | NA | Web UI |
| Tenant admin | Web UI user | Dedicated administration account for end user account management. The tenant admin can create, remove, modify, or invite mobile client users and view and access data related to the tenant (including tenant-based call detail records. | As assigned by web root admin | NA | Web UI |

TOE management functions described throughout this document are isolated through user authentication such that only authorized administrators can issue management commands to the TOE. After completion of the login with successful authentication all administrative actions that modify the configuration of the TOE are audited (please see paragraph 9).

**Notes:**

- All default login accounts for the command line interface as well as for the default Web UI login require a password change on the first login before any administrative actions can be performed.
- Unless otherwise specified, the administrative tasks should be executed by individual administrator accounts of the Web UI admin role, or the Security administrator role. The pre-existing default user accounts shall only be used for the initial set-up of the TOE described in '6 Initial setup'.
- During initial set-up the Security Admin shall log into all preexisting accounts using the default credentials and set new passwords.

# 6. Initial setup

3

The following section describes how to set up the TOE into an initial operational state.

## 6.1 Set up the management interface

To initially activate the management network interface `eth0` of the TOE, the security administrator must log in to the TOE as `secuadmin` user on the local terminal CLI. The default password for the setup user is `secuadmin`. The user is prompted to change the default password upon first login.

After logging in, the `secuadmin` user should configure the etho network for management access by executing the following command: `sudo secusetup`.

Depending on the network environment, the security administrator needs to define DHCP or Static IP (including appropriate necessary network information) for the interface.

Should the TOE operate in the "Restricted ciphers" mode (see section 7.14 Restricted ciphers), secuadmin should create a 384-bit SSH host key as described in section 7.14.2 Restricted SSH ciphers.

## 6.2 Set up network interfaces

After the management interface is set up, the TOE provides two access methods to the TOE over the management interface:

- Web UI over HTTPS administrative interface
  - On a web browser, connect to the management interface IP with port 443 (for example: https://172.20.20.20:443). Recommended browsers are Google Chrome, Mozilla Firefox, or Microsoft Edge.
  - The default username and password for the web UI administrator (web root admin) are **admin/admin**.
  - During first login, the user is asked to change the default password. The new password should be recorded and stored in a secure way to be able to use the default account during emergencies
  - To terminate a WebUI session, the user needs to click on the 'power button' ⏻.
- SSH CLI access administrative interface
  - The security administrator connects with an SSH client to the TOE using the secuadmin default user (for example, by using OpenSSH SSH client `ssh secuadmin@172.20.20.20`).
  - The default username and password for the Security administrator are **secuadmin/secuadmin**
  - During first login, the user is asked to change the default password.
  - To terminate a local or remote CLI session, the user needs to execute the `exit` command.

The system offers three network interfaces:

- Etho: The management interface
- Eth1: The external interface that SecuSUITE clients use, allowing the use of SCA and SIP services.
- Eth2: If the TOE is licensed for secure landing, this interface is used as a SIP trunk interface towards connected external PBX systems.

The interfaces are set up by logging in via the web UI.

1. Click **System > Configuration > Network Settings** in the web UI.
2. In the **Network Settings** tab, network interfaces must be set according to the network environment.
3. The **Hostname** should be set to the name that is used to identify the TOE in the connected network.

4. The **Domain** must be set to the DNS name of the eth1 external interface.

The ports 81, 2224, and 5060 are unused in the system, and may appear as "closed open" ports during network scans.

# 6.3 Install licenses

## 6.3.1 Install a BlackBerry service license

To request, create, and install the BlackBerry service license to the TOE, complete the following tasks.

1. Log into the Web UI as a web root admin.
2. Generate a license order: Click **System > License > SecuSUITE Service License.**
    a. Click the **Generate a new license** icon.
    b. Complete the required license order information (for example, number of subscribers, subscription period).
    c. Click **Generate** to create the order file.
3. Send the generated license order to BlackBerry SecuSUITE support for signing.
4. After receiving the signed license file from BlackBerry support, upload the license. Click **System > License SecuSUITE Service License.**
    a. Click the **Upload a license** icon.
    b. Select the received license file and click **Upload**.

# 6.4 Install BlackBerry signed certificates

To activate SCA and push notification services in the TOE, you must install the BlackBerry-signed TLS certificates for the Z\zero SCA server and push notification client to the TOE.

The procedure is described in section 7.2.1 Certificate handling with offline SecuSUITE CA.

# 6.5 Set-up User Accounts

## 6.5.1 Web UI Admins

The default user account for the Web UI interface should only be used for the initial steps described above. For further system setup, maintenance and operations, dedicated user accounts for individual admins shall be used so that administrative actions in the UI can be associated with individuals.

Set-up of additional Security and Tenant admins is described in 8.4.3 Web UI admin management.

**User login**

1. On a web browser, connect to the management interface IP using HTTPS (for example: https://172.20.20.20:443). Recommended browsers are Google Chrome, Mozilla Firefox, or Microsoft Edge.
2. Enter the username and initial password received from the Security Admin
3. During first login, the user is asked to change the initial password.
4. Afterwards the 'Home' tab of the Admin Portal is shown.
5. To terminate a WebUI session, the user needs to click on the 'power button'.

### 6.5.2 Add CLI Admin Users

The Security Administrator 'secuadmin' account should create additional console users for local and remote access to the TOE during the initial setup of the TOE. These additional individual administration accounts shall be used for the further configuration and normal operation of the TOE.

The 'secuadmin' default user account shall be configured with a complex password that should be stored securely. Further administrative tasks described in the following chapters shall be performed via the individual admin accounts only.

Additional user accounts can be created by:

```
sudo manage-secuadmin-users add <username> <password>
```

Creation and deletion of additional Security Administrators is described in 8.4 Admin user management.

**User login:**

1. The CLI Admin User connects with an SSH client to the TOE using the credentials provided by the Security Admin. (for example, by using OpenSSH SSH client: `ssh <username>@172.20.20.20`).
2. During first login, the user is asked to change the initial password.
3. To terminate a local or remote CLI session, the user needs to execute the `exit` command.

# 7. Security configuration

4

The following sections describe how to configure the system in the common criteria evaluated configuration.

Please note that configuration changes in the Web UI need to be **confirmed by the 'Submit' button** at the button of each configuration tab. The only exception is the upload of certificate, key and license files. The Security Administrator shall wait until the UI dialog confirms the successful application of the changes.

## 7.1 FIPS mode

The SecuSUITE SecuGATE server is compliant with FIPS 140-2. To achieve that compliance, OpenSSL is used with FIPS mode enabled for all cryptographic operations.

The FIPS object module of OpenSSL is required to be enabled explicitly during the initialization of all services that use OpenSSL as an underlying cryptographic module.

The FIPS mode is enabled for all services by default during the TOE commissioning process.

## 7.2 Certificates and keys

Certificate and key management is a crucial part of the system administration from security perspective. To get the SecuGATE system up and running, a number of certificates and keys must be imported, created, or exchanged.

System certificates and keys are managed in the web UI by clicking **System > Certificates** for certificate management.

Certificate management provides the following functionalities:

- Exporting, importing, and managing trusted CA bundles
- Generating and managing embedded CA certificates and keys
- Generating certificate signing requests (CSR) for TLS Client and TLS Server certificates that are used by the SecuGATE server. CSRs must be created for zero SCA and push notification service certificates. These CSRs must be signed by BlackBerry, and signed certificates must be uploaded to the TOE. See section 6.4 Install BlackBerry signed certificates for setting up the BlackBerry signed certificates. Other server certificates can be signed directly by the embedded CA or an external CA.
- Uploading externally-signed server certificates
- Signing server certificates with the embedded CA

To facilitate the setting of the certificate's subject and subject alternative name (SAN) properties, the security admin can set the default values for the certificate subject fields in the web UI by clicking **Configuration > PKI**.

These values are used as default values for every certificate signing request (CSR) created by the TOE.

The SecuGATE can be configured to utilize an existing (external) customer CA for certificate signing, or to use a self-signed CA embedded to the SecuGATE server. External CA setup is described in section 7.2.3 External CA, and the management of the embedded CA is covered in section 7.2.2 Embedded SCA CA.

Please notice that PEM format files that are used for certificates and keys shall not include Carriage Return characters (ASCII decimal value 13).

## 7.2.1 Certificate handling with offline SecuSUITE CA

SecuGATE requires two TLS certificates that must be signed by the offline SecuSUITE CA:

- **Zero SCA TLS server certificate:**
  This certificate is used for the first connection of SecuSUITE clients to the SecuGATE instance. The certificate will be provided by the BlackBerry SecuSUITE support team based on an offline CA (and upon request by the customer), and it is used by the SecuSUITE client to authenticate the SecuGATE instance to ensure the application is connecting to a trusted component. The SecuSUITE client has a related public root certificate hardcoded to the application, which allows the TLS certificate sent by the SecuGATE instance to be validated.
- **Push notification service TLS client certificate**:
  If the push notification services from the mobile platforms are utilized, the SecuGATE server must have a TLS client certificate to authenticate towards the BlackBerry push service aggregation server. This aggregation server collects push requests from different SecuGATE instances and forwards them (under a single trusted connection) to the vendor's push services.

These certificates are generated and signed by BlackBerry, based on certificate signing requests created via the SecuGATE certificate management.

For that, BlackBerry operates the following offline CAs and Sub-CAs:

- **SecuSUITE root CA**: This is the self-signed root CA for further SecuSUITE subordinate CA certificates. It signs the SecuSUITE zero SCA CA and the SecuSUITE push CA.
- **SecuSUITE zero SCA CA:** This is an initial trust point for SecuSUITE apps when starting the registration and certificate enrollment with a SecuSUITE system. It is signed by the SecuSUITE root CA.
- **SecuSUITE push CA**: This is signed by the SecuSUITE root CA, and it is used to sign endpoint certificates that enable SecuGATE to connect to the SecuSUITE push notification aggregator.

The related public root certificates are uploaded to the SecuGATE server as part of the trusted CA bundle provided by BlackBerry SecuSUITE support.

### 7.2.1.1 Uploading the trusted CA bundle from BlackBerry SecuSUITE support

BlackBerry SecuSUITE support provides you with a trusted CA bundle file. To deploy the CA bundle to the TOE via the web UI, complete the following steps:

1. Click **Certificates** > **Create Trusted CA Bundle**
2. Click **Upload certificate….**
3. Select the trusted CA bundle file and confirm.
4. To verify the certificate status, click **Show current certificate status** in the middle of the screen.
   Status should be **OK**.

### 7.2.1.2 Creation of zero SCA and push TLS certificates

To generate and set up zero SCA TLS and push notification service client TLS certificates, complete the following steps for both services: .

1. Click **Certificates > Create …** in the line of the service you want to create the certificate for.
2. Define the subject alternative name (SAN): "DNS=[FQDN]" or IP=[IP address of the external network interface you defined in the interfaces tab]."
   **Example:**
   "DNS=sglv.canada.com"
3. Define the **Subject** field: "/C=[two-letter abbreviation for your country]/ST=[state or province (optional)]/L=[City (optional)]/O=[name of your organization or company]/CN=[FQDN ]."
   **Example:**
   "/C=CA/ST=ON/L=Toronto/O=BlackBerry SecuSUITE support/CN=sglv.canada.com"

4. Click **Create CSR…** and save the certificate signing request.
5. Send the CSRs to BlackBerry SecuSUITE support. BlackBerry SecuSUITE support will sign the CSRs and provide you with the signed certificates.
6. After receiving the certificates, choose the corresponding service under **Certificates**, and click **Upload certificate…**.

**Important**: To activate all newly-created or imported certificates, you must reboot the SecuSUITE system (click **Update & Backup** > **Server management:** > **Reboot server**…).

## 7.2.2 Embedded SCA CA

The SecuGATE can be operated as a stand-alone infrastructure relying on an embedded CA and issues all endpoint certificates that are used for authentication and encryption between SecuGATE Server and SecuSUITE clients, as well for the end-2-end encryption between the secure VoIP clients.

The embedded SCA CA is not a part of the CA bundle file that is initially sent by BlackBerry SecuSUITE support. It has to be created via the SecuGATE web UI to be used as a customer-specific CA.

After creation, this CA is added to the internal trusted CA bundle. The CA can then be exported and backed up.

### 7.2.2.1 Create the embedded SCA CA

The embedded SCA CA is created exclusively on your SecuSUITE system.

1. Click **Certificates > Create Embedded SCA CA.**
2. Define the "subject" field according to this example:
   "/C=[*two letter abbreviation for your country*]/ST=[*state or province (optional)*]/O=[*name of your organization or company (optional)*]/CN=[*The common name associated with this self-signed SCA certificate*]".
   **Example:**
   "/C=CA/ST=ON/L=Toronto/O=BlackBerry SecuSUITE support/CN=sglv.canada.com"
3. Click **Create**. The embedded SCA CA is generated and installed in the system.
4. Save the certificate (optional).

The certificates can now be signed for subordinated instances (see next section 7.2.2.3 Create endpoint certificates).

### 7.2.2.2 Import a trusted Root CA as embedded SCA CA

In the case an externally generated trusted Root CA shall be imported to be used as embedded SCA CA, the admin must upload the related keys and certificates as an updated trusted bundle.

Preconditions:

- CAs certificate or certificate chain available in the users /home directory (filename sca1_ca.crt)
- CAs private key available in the user home directory (filename sca1_ca.key)
- Certificate export file is in the user's home directory (from step 2)

Steps to create a bundle that includes the externally created trusted Root:

1. In the Web UI click on System > Certificates
2. Export all certificates and keys (provide a secure password for the export; the same password should be used also for the re-import)
3. Create a temporary working directory and extract the exported tar file
   Example:

```
cd
mkdir trusted-ca-tmp
cd trusted-ca-tmp
tar xvf ~/<exported-cert-bundle>
```

4. Remove embedded SCA p12 container if existing
   Example:
   `rm sca1_ca.p12`
5. Create p12 container including the CAs key and certificate
   Example:
   `openssl pkcs12 -export -inkey ~/sca1_ca.key -in ~/sca1_ca.crt -out ~/trusted-ca-tmp/sca1_ca.p12 -descert`
   Use the same password as used in step 2.
6. Replace or add the CAs root certificate to the trusted bundle tar file
   Example:
   `tar vf trusted-ca.tar --delete ./sca1_ca.crt` (if existing)
   `tar rvf trusted-ca.tar -C ~/ ./sca1_ca.crt`
7. Create the import tar file
   Example:
   `tar cvf .~/import.tar *.p12 secucrtmgr.db trusted-ca.tar`
8. Import the file via the import button (System > Certificate)
   Use the password applied in step 2 and 5.

#### 7.2.2.3 Create endpoint certificates

When the embedded SCA CA is used, the following certificates must be issued by the embedded CA:

**Table 4: Endpoint Certificates**

| SecuGATE certificate type | Description |
|---|---|
| External SIP TLS | Mutually-authenticated TLS between SecuSUITE server and clients |
| SCA TLS | Used for mutually-authenticated TLS between SecuSUITE server and clients, after the initial SCA registration<br>**Note:** The BlackBerry SecuSUITE trust anchor for the VoIP client is only used during the initial SCA registration. After the initial registration, the client only uses the trust anchor provisioned during the registration (using either embedded CA or external CA). |
| When secure landing is used: | |
| Voice encryption S/MIME | Used for encrypted and authenticated voice connections between SecuSUITE clients and the SecuGATE server when forwarding an unencrypted voice call to a PBX. |
| Voice authentication S/MIME | |

The generation of all these certificates follows the same pattern:

1. Click **Create …** in the line of the service you want to create the certificate for.
2. Check **Sign with embedded CA**.
   - If not yet completed, define the following "subject alternative name" (SAN): "DNS=[FQDN] or IP=[IP address of the external network interface you defined in the interfaces tab]."
     **Example**:
     "DNS=sglv.canada.com"

- The **Subject** field: "/C=[*two-letter abbreviation for your country*]/ST=[*state or province (optional)*]/L=[*City (optional)*]/O=[*name of your organization or company*]/CN=[*FQDN* ]."
  **Example:**
  "/C=CA/ST=ON/L=Toronto/O=BlackBerry SecuSUITE support/CN=sglv.canada.com"

3. Click **Create a certificate…**
4. Save the certificate (optional).

**Important:** To activate all newly-created or imported certificates, reboot the SecuSUITE system (click **Update & Backup** > **Server management** > **Reboot server…**).

## 7.2.3 External CA

The handling of the certificates in context with the external CA differs only in a few places from the handling required with a SecuSUITE Embedded CA, aside from the certificate management via the external CA itself (instructions for using the CertAgent tool are included in the SecuSUITE for Government External CA SecuGATE Configuration Guide).

Instead of directly issuing the certificates based on the embedded CA, the certificate-signing requests for each required certificate is submitted to the external CA, and the signed X.509 certificates are imported back to the SecuGATE.

The following certificates issued by the external CA must be imported:

- External CA's root certificate
- EST Client TLS

### 7.2.3.1 Import the external CA's root certificate

For certificate validation, and for submitting the root certificate to the SecuSUITE clients, the SecuGATE needs the public root certificate of the external CA:

1. If necessary, download or export the external CA's root certificate from the external CA.
2. Change the name of the root certificate file (in PEM file format) to "`sca1_ca.crt`".
3. Generate a .tar file of the certificate. For example:
   - Create a directory for copying the `sca1_ca.crt` file on your computer.
     **Example:** `mkdir trusted-ca`
   - Change the working directory to the created directory.
     **Example**: `cd trusted-ca`
   - Copy the *sca1_ca.crt* certificate to the created directory.
     **Example**: `cp /home/sca1_ca.crt`
   - Create a .tar file of the directory content.
     **Example**: `tar cf trusted-ca.tar -C`
4. Upload this newly-created *trusted-ca.tar* file archive by clicking **Certificates > Trusted CA Bundle > Upload certificate…**.

### 7.2.3.2 Create and import the EST client TLS certificate

The EST client TLS certificate is used for the mutual authentication between the SecuGATE and the External CA. The certificate authenticates the SecuGATE as a registration authority, authorized to send certificate-signing requests to the external CA that are excepted and signed by the CA without further verification of the validity of the singing request.

1. Click **Certificates > Create EST Client TLS**.
2. Enter the subject alternative name and common name.
3. Clear the **Sign with embedded CA** check box.

4. Click **create** to create a CSR for the EST client certificate
5. After you received the EST Client certificate issued by the external CA, click **Certificates > EST Client TLS > Upload certificate**.

#### 7.2.3.3 Create remaining certificates

In addition to the external CA's root certificate and the EST Client TLS certificate, you must create and import all of the certificates mentioned in chapter 7.2.2 Embedded SCA CA, including the following:

- External SIP TLS
- SCA TLS
- Voice encryption
- Voice authentication

To generate a certificate, complete the following steps for each certificate:

- Click **Certificates > Create …** in the line of the service you want to create the certificate for.
- Enter the subject alternative name and common name.
- Clear the **Sign with embedded CA** check box.
- Click "**Create CSR…**
- The generated CSR must be **signed** by the external CA.
- After you received the corresponding certificate issued by the external CA, click **Certificates**.
- Click the certificate name, and click **Upload certificate…**.

## 7.2.4 Generating other certificates

In addition to the certificates mentioned above, the TOE may need the following additional certificates:

- **SIP Trunk TLS:** the certificate used for SIP trunk TLS communication between the TOE and an external SIP PBX system
- **Web Portal TLS:** the web server certificate
- **Syslog Client TLS:** the certificate used for TLS communication between the TOE and an external syslog server

The certificate can be signed by the SecuGATE CA (embedded or imported) or a customer CA.

SecuGATE CA-signed certificates must be created as described in sections 7.2.2.3 Create endpoint certificates (for embedded CA certificates) or 7.2.3.3 Create remaining certificates (for external CA certificates).

If a customer-specific CA is used for these certificates, the certificates must be generated using CSRs that are signed by the customer CA, similar to the steps outlined in section 7.2.3.3 Create remaining certificates.

## 7.2.5 General certificate handling

#### 7.2.5.1 Exporting certificates

After you created all the certificates, it's recommended that you export them (and re-export them each time you change or refresh a certificate).

1. In the upper-right corner of the **Certificates** tab, click **Export certificates…**
2. Select the **Include keys** check box.

3. Set a password for the exported certificates. This password is required to import the certificates from the export file.
4. Click **Export**.
5. Save the exported file, titled `exported_certificates.tar`.
6. When you open this file in an extractor program, the single certificates are listed as pk12 containers.

#### 7.2.5.2 Importing certificates

1. In the upper-right corner of the **Certificates** tab, click **Import certificates…**
2. Upload a previously-exported certificate file.
3. Click **Import**.
4. Enter the password you set when you exported the certificates.

#### 7.2.5.3 Managing certificates

Certificates expire after a period of time. You need to check each certificate's validity frequently and start the creation of a new certificate before the current one expires.

**Recommendation:** Create an overview of all certificates with their expiration dates to monitor them and request new certificates when needed.

- **Checking the status of the certificates:**
    1. Click **Certificates**.
    2. Click on a certificate.
    3. Click **Show current certificate status**.
        - If the status reads **OK**, the certificate is valid.
        - If the certificate is not valid, you can see a certificate validation error number.
    4. In the **Validity** section, check the validity dates. The certificate is valid between the **Not before** and **Not after** dates.
- **Deleting a certificate:**
    1. Click **Certificates**.
    2. Click on a certificate.
    3. Click **Delete certificate and key** to delete the certificate and its corresponding private key. Click **Delete certificate** to delete the certificate only.
- **Create a new key/rekey**

    To perform rekeying (for example, to generate a new private key on which the certificates are based), complete the following steps:

    1. Click **Certificates**.
    2. Click on a certificate.
    3. Click **Delete certificate and key** to delete the certificate and its corresponding private key. Click **Delete certificate** to delete the certificate only.

    **Important:** To ensure continuity of service, the new private key and the corresponding certificate are first activated after the new certificate has been uploaded successfully.

- **Importing a custom trusted CA bundle**

In case a customer-specific CA is used for SIP Trunk TLS, web portal TLS, or syslog client TLS certificates, the CA certificate must be uploaded to the system as a trusted CA. The CA files are .tar files, and can be uploaded by clicking **Certificates > Trusted CA Bundle**.

Generate a .tar file of the certificate.

**Example:**

1. Make a directory for copying the CA certificate files on your computer.
   **Example:** `mkdir trusted-ca`
2. Change the working directory to the created directory.
   **Example:** `cd trusted-ca`
3. Copy the certificates (the certificates must be in PEM format with the .crt postfix) to the created directory.
   **Example:** `cp ca1.crt ca2.crt`
4. Create a .tar file of the directory content
   **Example:** `tar cf trusted-ca.tar -C`
5. Upload the generated tar file by clicking **Certificates > Trusted CA Bundle > Upload certificate**.

## 7.2.6 RSA and ECC support for TLS

The TOE supports two types of TLS service configurations that are described in sections above using either RSA or ECC algorithms for authentication while using only ECDHE for key exchanges. The TOE supports NIST curve p-384 by default, but can be configured (as shown below) to use NIST curve p-256 by installing a P-256 EC key pair into the TOE. Depending on that configuration, the TOE accepts TLS connections only from the TLS peers that support the corresponding cipher-suite.

By default, key pairs that are generated using the Web UI **Certificates** options are generated as elliptic curve keys pairs based on Nist curve p-384.

To generate 2048 bit RSA or P-256 EC key pairs, the security admin must first generate the keys using the CLI interfaces. After the keys are generated, the Web UI can be used for further certificate management, as described in section 7.2 Certificates and keys.

1. Log in via the CLI interface as `secuadmin`.
2. Execute the following:

   ```
   sudo secuadmin gen-key -n <key-name> -a <algorithm>
   ```

- The `<key-name>` above identifies the key to be generated. The following keys are available:
  - pns_client_tls: Push notification service client TLS
  - scao_tls: Zero SCA TLS
  - esip_tls: External SIP TLS
  - sca1_tls: SCA TLS
  - voice_enc: Voice encryption S/MIME
  - voice_auth: Voice authentication S/MIME
  - pabx_sip_tls: SIP trunk TLS
  - nginx_tls: Web portal TLS
  - est_tls: EST client TLS
  - syslog_tls: Syslog client TLS
- `<algorithm>` identifies the elliptic curve or RSA algorithms to be used. The following algorithms are available:
  - rsa

- P-256
- P-384

3. In order to generate a self-signed embedded CA, type the following command:
   `sudo secuadmin gen-self-signed -n sca1_ca -s <subject> -a <algorithm>`

   The **-a** parameter is as described above, and the **<subject>** variable is the subject of the self-signed certificate.

**Note:** For the external SIP interface used by the VoIP clients (key-name: esip_tls) the security admin may only create elliptic curve-based keys and certificates for the operation of the TOE in the evaluated configuration.

# 7.3 Remote access login options

Login and password options are configured by logging in to the system locally or via SSH, by using the `secuadmin` commands as described below.

The following configurations are managed by `secuadmin` config command like `sudo secuadmin config --config_item` value. The settings apply to ssh user logins as well as to Web UI logins

## 7.3.1 Inactivity time

**`login_max_idle_time:`** The session expiration time in minutes. This option affects CLI and web UI settings for all system users.

**For example:** `sudo secuadmin config --login_max_idle_time 10`

After you change this setting, you must log out to complete the change.

## 7.3.2 Blocking time after failed logins

**`login_failure_blocktime:`** The time a user is blocked from signing in (in minutes) after the maximum number of login attempts is reached. This option affects SSH and web UI settings for all system users. This setting does not affect users logged in to a local console.

**For example:** `sudo secuadmin config --login_failure_blocktime 10`

After you change this setting, you must log out to complete the change.

**Notes:**

- The change does not affect the blocking time of already-blocked web UI users.
- User blocking does not affect local console users. Hence a user that got blocked for ssh remote access, can still login to the system using a local console access

## 7.3.3 Maximum login retries before blocking

**`login_retries`***:* The maximum number of failed login attempts before the account is blocked. This option affects ssh CLI and web UI settings for all system users.

**For example:** `sudo secuadmin config --login_retries 3`

**Note:**

- The blocking of user after maximum login retries is not applied to local console users

### 7.3.4 Password minimum length

**`password_min_len`:** The minimum number of characters for a user password (between 8 and 32 characters). This option affects CLI and web UI settings for all system users.

**For example:** `sudo secuadmin config --password_min_len 15`

## 7.4 Access banner

To change the access banner displayed to administrative users before they log in using the web UI or CLI, log in to the system as `secuadmin` via SSH or local console.

The banner is defined in the following file: `/etc/session-banner`.

The file can be edited with `vi` editor.

**For example:** `vi /etc/session-banner`

The file can also be copied to the file location via `cp` or `scp` Unix commands.

## 7.5 Remote access SSH options

The TOE authenticates administrative users and handles SSH sessions. The SSH session parameters cannot be changed by the Security Administrator and are defaulted to the values listed in Table 5.

**Table 5: SSH server configuration**

| Parameter | Values | Required values | Description |
|---|---|---|---|
| Protocol | 1 or 2 | **2** | SSH protocol version used |
| ListenAddress | HOST[:PORT] | **IP address of TOE** | Local address to be listened on |
| Port | Port number | **22** | Port number to be listened on |
| Ciphers | Comma-separated list of cipher names | **aes256-gcm@openssh.com,aes256-ctr,aes128-gcm@openssh.com,aes128-ctr** | Ciphers allowed |
| KexAlgorithms | Comma-separated list of key exchange algorithms | **ecdh-sha2-nistp256,ecdh-sha2-nistp384** | Key exchange algorithms allowed |
| MACs | Comma-separated list of MAC algorithms | **hmac-sha2-256** | MAC algorithms allowed |

| Parameter | Values | Required values | Description |
|---|---|---|---|
| HostKey | Path to a file | **/etc/ssh/ssh_host_ecdsa_key** | SSH server private key |
| AuthorizedKeysFile | Relative path to a file | **.ssh/authorized_keys** | File with public keys to be used for user authentication |
| PubKeyAuthentication | yes \| no | **Yes** | Enable or disable public key authentication |
| PasswordAuthentication | yes \| no | **Yes** | Enable or disable password authentication |
| RekeyLimit | MAXBYTES[ MAXTIME] | **64M 30M** | Maximum bytes (and, optionally, maximum time) allowed before the session key is renegotiated.<br>MAXBYTES is specified in bytes, and can have a suffix of 'K', 'M', or 'G' to indicate Kilobytes, Megabytes, or Gigabytes, respectively. The maximum value the administrator can configure is '1G' (one gigabyte). The optional second value, MAXTIME, has the suffix of 'S', 'M', 'H', 'D', or 'W' to indicate seconds, minutes, hours, days, or weeks, respectively. The maximum value the administrator can configure is '1H' (one hour).<br>When either of the configured thresholds is met, the TOE will rekey and reset the limit counter. |
| LogLevel | One of QUIET, FATAL, ERROR, INFO, VERBOSE, DEBUG, DEBUG1, DEBUG2, and DEBUG3 | **INFO** | Verbosity level |

# 7.6 SSH public key access configuration

To use public key authentication for remote SSH logins, a security administrator must put their public key on the TOE. The file must be created if it does not already exist.

To generate and deploy public keys for the TOE using OpenSSH tools, complete the following steps:

1. Generate an ECDSA 384-bit key pair for SSH public key authentication using `ssh-keygen` (not necessarily on the TOE). To generate the key pair, run the following command: `ssh-keygen -t ecdsa -b 384`. Follow the instructions on the screen and keep the default filename for the generated keys.

2. Copy the generated public key to the TOE by running the following command: `ssh-copy-id -i ~/.ssh/id_ecdsa.pub secuadmin@<TOE-management-IP>`. The `~/.ssh/id_ecdsa.pub` is the file name of the generated public key from step 1.

# 7.7 TLS configurations

The TOE can establish TLS connections with external entities.

The following chapters describe the configuration of the various TLS connections with external entities. As a default the TOE uses NIST curve p-384 for all keys created via the Web UI. In case 2048 RSA[1] or p-256 keys need to be used, those need to be created and configured via the CLI by secuadmin user. Please refer to the instructions in ' 7.2.6 RSA and ECC support for TLS'.

For all of the following connections between the TOE and an external component (if those apply), the related TLS configurations are required for the evaluated configuration.

Please note that the TOE automatically re-connects TLS sessions which become disconnected due to network connectivity issues without any administrative actions by the Security Administrator.

## 7.7.1 SIP: TLS client/server mode

If the TOE shall be connected to other SIP server or PBX system via trunk connection, the communication must be TLS protected in the evaluated configuration. The TOE is able to establish those TLS connections in client or server mode.

The TOE can be configured to communicate with 1 trunking peer.

For the mutual authentication between the TOE and the peer a TLS certificate needs to be either created or imported to the TOE by the Security Administrator (please see 7.2.5 General certificate handling).

You can configure this TLS connection in the web UI.

1. Click **System > Configuration > PABX Transport Settings**.
2. Select **Secure Landing Enabled**.
3. Set the **Transport Mode** to **TLS**.
4. Set the **PABX IP** and **PABX Port** to point to the connected PBX.
1. Set the reference identifier for the PABX IP address in the 'Advanced' section following the instructions in '5 Set the **Cipher Suite** to **ECDHE-ECDSA-AES256-GCM-SHA384, ECDHE-ECDSA-AES128-GCM-SHA256, ECDHE-RSA-AES256-GCM-SHA384, ECDHE-RSA-AES128-GCM-SHA256**.
2. Set the **TLS Signature Algorithms** to **ECDSA+SHA384, ECDSA+SHA256, RSA+SHA384, RSA+SHA256.**
3. Set the **TLS Curves** to **secp384r1**.
5. X.509 Certificate verification'
6. Set **TLS Method** to **1.2**.
7. Set the **Cipher Suite** to **ECDHE-ECDSA-AES256-GCM-SHA384, ECDHE-ECDSA-AES128-GCM-SHA256, ECDHE-RSA-AES256-GCM-SHA384,ECDHE-RSA-AES128-GCM-SHA256**.
8. Set the **TLS Signature Algorithms** to **ECDSA+SHA384, ECDSA+SHA256, RSA+SHA384, RSA+SHA256.**
9. Set the **TLS Curves** to **secp384r1**.
10. Select **CRL** option.

[1] RSA support only for TLS authentication, not for key exchange.

The default listening port in the server mode is **5061**.

TLS connections are mutually authenticated.

### 7.7.2 SIP: TLS server mode

The TOE establishes TLS connections in server mode for SecuSUITE clients.

You can't change the TLS parameters.

The default listening port in the server mode is **5061**.

TLS connections are mutually authenticated.

### 7.7.3 SCA: TLS server mode

The TOE establishes TLS connections in server mode for SCA service. You can't change the TLS parameters.

The default listening port in the SCA server mode without mutual authentication is **3978**.

The default listening port in the SCA server mode with mutual authentication is **5062**.

### 7.7.4 Push server TLS client mode

The TOE establishes TLS connections in client mode for push.

You can't change the TLS parameters.

### 7.7.5 Syslog TLS client mode

The TOE establishes TLS connections in client mode to an external syslog server. The connection to the syslog server needs to be protected by TLS in the evaluated configuration.

The CC evaluation compliant configuration is done via web portal by following steps:

1. Click **System > Configuration > Logging > External Syslog Server**.
2. Select **External Syslog enabled**.
3. Set the **Host** and **Port** to point to the connected external syslog server.
4. Set **Transport Mode** to **TLS**.
5. Set **TLS Version** to **1.2**.
6. Set the **Cipher Suite** to **ECDHE-ECDSA-AES256-GCM-SHA384, ECDHE-ECDSA-AES128-GCM-SHA256, ECDHE-RSA-AES256-GCM-SHA384,ECDHE-RSA-AES128-GCM-SHA256**.
7. Set the **TLS Signature Algorithms** to **ECDSA+SHA384, ECDSA+SHA256, RSA+SHA384, RSA+SHA256.**
8. Set the **TLS Curves** to **secp384r1**.

Once configured the TOE will automatically open the connection to the syslog server without further administrative actions by the Security Admin. The TOE will continue writing audit logs to the local storage in parallel.

### 7.7.6 Web server TLS server mode

The TOE establishes TLS connections in server mode for web portal HTTPS connections.

You can't change the TLS parameters.

### 7.7.7 External CA TLS client mode

In case an external CA shall be used to sign the certificates that shall be used for mutual authentication between the VoIP Endpoints and the TOE, the connection between the TOE and the external CA must be TLS protected in the evaluated configuration.

4. The TOE establishes TLS connections in client mode to an external CA server via EST protocol.
5. Configuration is done via web portal.
6. Click **System > Configuration > External CA** and set the **External CA** and **Profiles** sections according the configuration of the external CA to be connected.
7. Set **TLS Version** to **1.2**.
8. Set the **Cipher Suite** to **ECDHE-ECDSA-AES256-GCM-SHA384, ECDHE-ECDSA-AES128-GCM-SHA256, ECDHE-RSA-AES256-GCM-SHA384, ECDHE-RSA-AES128-GCM-SHA256**.
9. Set the **TLS Signature Algorithms** to **ECDSA+SHA384, ECDSA+SHA256, RSA+SHA384, RSA+SHA256.**
10. Set the **TLS Curves** to **secp384r1**.

## 7.8 X.509 Certificate verification

X.509 certificates received during a TLS negotiation are validated according to RFC 5280 certificate validation. Certificate paths must terminate with a trusted CA certificate.

Certificate Revocation Lists (CRLs) are validated as specified in RFC 5280. If a CRL is not reachable, then the certificate is considered revoked.

The following X.509 extensions are verified:

- basicConstraints: CA flag must be true for all CA certificates
- extendedKeyUsage:
  - Server authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) must be present in all TLS server certificates.
  - Client authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) must be present in all TLS client certificates.
- KeyUsage
  - CRLsign must be present in all certificates that sign a CRL received by the TOE

In addition, certificates in TLS connections are verified against expected TLS peer identifiers. A TLS peer identifier is the common name (CN) or the subject alternative name (SAN) in the certificate. The system will only accept an IP address in the SAN.

Checking the identifier against a matching IP address or DNS name is done in three phases:

- If the identifier in the certificate directly matches with the IP address or DNS name of the connection peer, the certificate identifier is accepted.
- The identifier is validated against a mapped IP address or DNS name for SIP trunks, syslog, push server, and external CA. To configure the mapping via the web UI, complete the following steps:
  1. Click **System** > **Configuration** > **Advanced** > **TLS Peer Validation Identifiers.**
  2. Add a **Host** (IP address or DNS name) and **Expected Identifier** (CN or SAN entry) to the connecting TLS service. For example:
     - Host: **172.20.20.20,** Expected Identifier: **DNS:my.syslog.local** (DNS entry in the SAN)
     - Host**: 172.20.20.21,** Expected Identifier: **IP:172.20.20.21** (IP entry in the SAN)

- Host**: my.syslog2.local,** Expected Identifier: **my.syslog3.local** (CN in the Subject)

  If a host has multiple identifiers that the host address must be checked against, each identifier must be added to a new item in the **TLS peer validation identifier** section.

- In the last phase, the identifier is checked against expected SAN identifiers that are allocated to mobile clients during SCA certificate enrolment.

Wildcard certificates are validated according best practices in the Security Target.

# 7.9 Syslog configuration

The TOE can be connected to an external syslog server, and you can forward TOE syslog and audit logs to the external syslog server.

An external syslog server with TLS as transport is configured according section 7.7.5 Syslog TLS client mode.

# 7.10 NTP configuration

The TOE can be connected to NTP servers, which are used for TOE time synchronization. NTP v4 is supported and up to 4 different servers can be configured via the Web UI by the Security Administrator.

The NTP servers must be configured with SHA-256 authentication in the evaluated configuration.

1. Click **System > Configuration > Network Settings > NTP Servers**.
2. Add an NTP server by completing the following information:
   - **Host**: The IP address or the DNS name of the server
   - **Port**: The port of the NTP server (typically `123`).
   - **Key ID**: The key ID of the authentication key.
   - **Key:** The SHA-256-based key itself. A sample of the format of the key is `HEX:EDD291F5E0B6E86F6C228EA9E9F1E800BE43463257DE250A5E8F01A8B48D13C78CA6F81B68402188A0EBDF0954853B5DA333F27BB02ACBC5BFA59EC19CA4FD06`

The keys must match for the key ID on the server and TOE side. The key can be obtained from the server or generated in the TOE. The process for obtaining the key from the server is dependent on the server infrastructure and deployment. The key can be generated in the TOE by running the following command:

```
chronyc keygen keyid SHA256 256
```

**For example:** `$ chronyc keygen 73 SHA256 256`

```
73 SHA256 HEX:CBD5D6CB4F5E40C1A3484A322146B601E9475AB65121B4B88C261EADEC864396
```

**Note:** Please note that broadcast and multicast time updates are not accepted by the TOE and no additional administrative action is required.

# 7.11 Entropy configuration

The `rngd service` running on the TOE reads random data from hardware (via Intel RDRAND) and feeds the kernel entropy pool whenever the kernel entropy level is less than a threshold value. As a result, all applications on the TOE can use `/dev/random` for high-quality entropy needs without blocking.

All TOE services seed OpenSSL DRBG with random bytes from `/dev/random`.Configuration of the `rngd` service (including the threshold value) is defined in the file `/etc/sysconfig/rngd` (see Table 6). The option `--fill-watermark` is used to set the kernel entropy level. When the level is reached, `rngd` feeds the entropy pool. This is set to 3072 in the default TOE configuration.

**Table 6: rngd configuration**

| Parameter | Values | Default value on TOE | Description |
|---|---|---|---|
| EXTRAOPTIONS | rngd arguments:<br>`--fill-watermark=n`: entropy level threshold to stop feeding, 0 <= n <= 4096.<br>(see `rngd --help` for all) | "**`--fill-watermark=3072`**" | Additional start-up options |

Please note: The entropy configuration information has been added for information and for completeness only. The default configuration of the rngd service shall not be changed and the related config file cannot be changed by the Security Administrator role.

# 7.12 System status log configuration

The TOE provides a possibility to forward system status information as described in section 8.12 System status log to the syslog (external syslog and /var/log/messages).

The configuration is done via the CLI interface with the command

```
sudo systemctl restart status-logger.timer
```

The status is sent to the syslog every 60 seconds.

The service can be controlled by `systemctl` command (for example: `status, start, restart, enable`). To ensure that the service is enabled during system boot execute

```
sudo systemctl enable status-logger.timer
```

Please see 10 System Logs for the type and format of system logs created.

# 7.13 Call detail records

To turn on call detail records in the system, click **System > Configuration > Export Archives** and enable **Call Detail Records (CDR)**.

No administrative action is required to ensure that locally stored call detail records are protected from unauthorized deletion and modification.

The Web UI admin role as well as the Tenant Admin role can access and download Call Detail Records in the portal via: **Administration > Export Archives / Select a record: Call Detail Record (CDR)**.

# 7.14 Restricted ciphers

The TOE provides a possibility to restrict SSH and TLS cipher parameters to conform to the CSfC Selections for Transport Layer Security (TLS) Protected Servers. This requires use of both the Restricted TLS and SSH ciphers, as well as configuration of all other interfaces to be in compliance. The required configuration steps are described below and must all be followed.

### 7.14.1 Restricted TLS ciphers

To turn on restricted TLS ciphers in the system for interfaces that are not configurable by the security administrator, click **System > Configuration > Advanced > Restricted ciphers** and enable **Restricted TLS ciphers.**

The setting will set TLS ciphers in TLS interfaces that are not configurable in detail by the security administrator to the following values:

Ciphers: ECDHE-ECDSA-AES256-GCM-SHA384, ECDHE-RSA-AES256-GCM-SHA384
Elliptic curves: secp384r1
Signature algorithms: ECDSA+SHA384, RSA+SHA384

In addition, **SIP: TLS client/server mode** (7.7.1 SIP: TLS client/server mode), **Syslog TLS client mode** (7.7.5 Syslog TLS client mode) and **External CA TLS client mode** (External CA TLS client mode) must be configured as follows:

Ciphers: ECDHE-ECDSA-AES256-GCM-SHA384, ECDHE-RSA-AES256-GCM-SHA384
Elliptic curves: secp384r1
TLS signature algorithms: ECDSA+SHA384, RSA+SHA384

After changing the setting, the system should be rebooted.

### 7.14.2 Restricted SSH ciphers

To turn on restricted SSH ciphers for the system, click **System > Configuration > Advanced > Restricted ciphers** and enable **Restricted SSH ciphers.**

The setting will set SSH ciphers to the following values:

Ciphers: aes256-gcm@openssh.com
Kex algorithms: ecdh-sha2-nistp384
Host key algorithms: ecdsa-sha2-nistp384

MAC algorithm will be negotiated as implicit.

Before enabling the restricted SSH ciphers, the security administrator must create a 384-bit host key on the CLI interface:

```
sudo ssh-keygen -b 384 -t ecdsa -f /etc/ssh/ssh_host_ecdsa_key
```

After changing the setting, the system should be rebooted.

# 8. Security management

5

## 8.1 Software version query

Via Web UI:

The current installed and active software version can be queried in the web portal by clicking the **About** tab.

Via CLI:

The current installed and active software version can be queried using

```
/bin/cat /etc/server-version
```

## 8.2 VoIP endpoint management

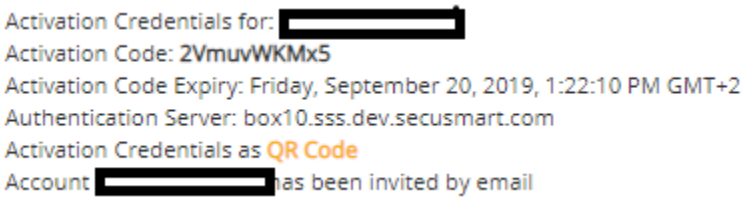
SecuSUITE client VoIP endpoints are managed in the web UI in the User **Accounts** tab. Before an end-user can activate and use the SecuSUITE app, an end-user account (VoIP endpoint account) needs to be created by a tenant admin. After the successful creation of the user account, an activation code can be created and communicated to the user, along with the registration procedure.

The SecuSUITE user is required to download, install, and launch the SecuSUITE app and register with the activation code mentioned above. This process ensures that the VoIP endpoints receive the X.509 certificates that are used to authenticate the endpoints during TLS mutual authentication for regular SIP and SCA connections.

### 8.2.1 Creating and activating a VoIP endpoint

As a tenant admin:

1. Select the tab "User Accounts".
2. Click on the icon to add a user account.
3. All fields are mandatory except "Alternative Phone Number" and "Display" name.
4. After entering the user's data, click on "Create".
5. Activate the user by clicking on the invite icon.
   Depending on your system's settings, this will trigger an email to the user, or just display the activation credentials on top of the "User Accounts" tab, e.g.




Note: If the email is not activated, it's up to you to inform users of activation codes and where to get the app.

**Alternative Phone Number**

The Alternative Phone Number is only required if VoIP endpoints shall be reachable via an PABX internal number. The number is used to route incoming calls (called into the PABX) to the TOE. The Alternative Phone Number is the PABX internal number that is configured to be forwarded to the TOE within the PABX.
Only users with a defined Alternative Phone Number are allowed to receive incoming calls via the PABX Trunk connection.

**Activation Code handling**

The activation code consists of an 8-digit authentication code and 2 additional characters reserved for additional client configuration. Only the first 8 digits are evaluated by the TOE to authenticate a connection request coming from a VoIP endpoint during the registration process.

### 8.2.2 Endpoint Re-invitation

There are different user scenarios where an end-user may need to be re-invited and registered again to the SecuGATE infrastructure. For instance, after device switch or in case the delivered activation code has expired before the user was able to use it with the SecuSUITE application.

1. Navigate to the "User Accounts" tab
2. Click on the "Invite Account" button in the 'Action' column to create a new activation code.
3. A notification asks for confirmation: "Do you really want to re-invite User <last name, first name>?"
4. Click "Yes".

With the newly created activation code, the user can (re-)activate the app.

### 8.2.3 Editing endpoint accounts

To edit user data, go to "User Accounts" tab and click on the "Edit Account" button for the selected user account. All changes made are applied immediately, except for a change in phone number, which requires reactivation:

Important: If a phone number change shall be performed, the user needs to be reactivated by the re-invitation procedure described above.

### 8.2.4 Deleting endpoint accounts

To delete a user account, click on the "Delete Account" icon next to their user entry and confirm. If the same user shall be invited to SecuSUITE for Government again, a new user account needs to be created.

Important: The user will immediately be unable to make secure calls with the SecuSUITE application. The app does not notify a user when their account is deleted.

## 8.3 Admin password change

### 8.3.1 Web UI Users

To change the web UI administrator password, log in to the web UI and click **Administration > Manage Admins > Change Password.** Follow the instructions on the screen to change the password.

**WebUI Password complexity**

The WebUI enforces password complexity. The TOE ensures the following complexity rules:

- Minimum password length (please see chapter 7.3)
- At least 1 digit
- At least 1 lower as well as 1 upper character
- At least 1 special char out of the following set of characters: ['!', '@', '#', '$', '%', '^', '&', '*', '(', ')'];

Apart from this password complexity rules, passwords can be composed of any combination of upper and lower case letters, numbers, and the above mentioned special characters.

**CLI Password complexity**

In addition to complexity rules for WebUI users, CLI enforces additional Red Hat default password complexity rules such as checks for password rotation, user name inclusion in the password, palindromes and case changes only.

### 8.3.2 CLI and local console users

To change the password as a CLI and local console user, log in to the CLI interface and run the following command:

`/bin/passwd`.

The user will be prompted for the current password before the new password can be defined.

**CLI User Password Complexity**

For CLI users the TOE enforces the following complexity rules:

- Minimum password length (please see chapter 7.3)
- At least 1 digit
- At least 1 lower as well as 1 upper character
- At least 1 special char (at least ['!', '@', '#', '$', '%', '^', '&', '*', '(', ')'] are supported)

Apart from this password complexity rules, passwords can be composed of any combination of upper- and lower-case letters, numbers, and the above-mentioned special characters.

## 8.4 Admin user management

### 8.4.1 Add console users

Additional console users with the role 'Security Administrator' shall be created by the default 'secuadmin' account using the `manage-secuadmin-users` script:

1. As a Security Administrator type '`sudo manage-secuadmin-users --add`'
2. Enter the name of the new admin account
3. Enter the initial password of the newly created account

Please note:

- New users created via '`manage-secuadmin-users`' are required to change their password on first login
- Security administrators created via '`manage-secuadmin-users`' are able to create further console users using the same script.
- Alternatively, to the interactive script the security admin can directly type:

  `sudo manage-secuadmin-users add <username> <password>`

### 8.4.2 Delete console users

CLI users with the role 'Security Administrator' must be removed using the `manage-secuadmin-users` script:

1. As a Security Administrator type `'sudo manage-secuadmin-users del'`
2. Enter the name of the security admin account that shall be delete

## 8.4.3 Web UI admin management

Additional web root admins and tenant admins can be created and invited in the web UI by clicking **Administration > Manage Admins**. In the **Manage Admins** tab, you can add, delete, and edit web root admins and tenant admins.

### 8.4.3.1 Creating new admin accounts

1. Click **Administration** > **Manage Admins**.
2. Click the **Create new** icon.
3. Enter the following details for the new administrator:
   - **First Name/Last Name**: Enter the name of the new administrator.
   - **Email**: Enter the email address. This will be the administrator's user ID. If email activation is set in the SecuSUITE system, the new administrator receives an invitation email with the SecuSUITE user administration portal link, and the administrator's initial password.
   - **Phone number**: Enter a phone number.
4. Select the user sub-role:
   - **root-admin for the web UI admin**
   - **tenant-admin for the tenant admin.**
5. For tenant admins, select the related tenant from the drop-down list.
6. Click **Create**.

### 8.4.3.2 Invite an administrator

Before a new web UI administrator is activated, you must invite the administrator to use the TOE.

1. Click **Administration** > **Manage Admins**.
2. Next to the user you want to invite, click the **Invite Account** icon.
3. The password is displayed on the screen or sent to the new administrator via email.
   - If the password is displayed, communicate it to the new administrator in a secure way. You must also provide the link to your administration portal link.
   - If you choose email delivery, the new administrator will receive the link to the portal and their password in two separate emails.
4. After the first login, the new administrator is prompted to change their password immediately.

### 8.4.3.3 Manage admin accounts

- **Edit or delete administration accounts:**
  Click **Administration** > **Manage Admin**. Next to an administrator account, click the **edit** or **delete** icon.
- **Change your own password:**
  Click **Administration** > **Change Password**.
- **Change an admin account password**
  To reset an admin account password, complete the instructions in the 8.4.3.2 Invite an administrator section above.

## 8.5 Unlocking a blocked administrator

In some cases, access to the SecuSUITE SGLVN is blocked (for example, when a user tries to log in with the wrong credentials too many times). Security administrators can unlock a blocked user by logging in as `secuadmin` from the console.

To unblock CLI system administrators, type the following commands on the CLI interface:

```
sudo /sbin/faillock --reset --user <username-to-unblock>
```

To unblock web UI administrators, type the following command:

```
sudo secuadmin unblock-user -u <username-to-unblock>
```

## 8.6 Software update

The SecuSUITE SGLVN server can be updated using the software update packages provided through BlackBerry online services.

An update consists of an update file (in .gzip format), and a SHA-256 checksum of the package for file integrity verification. Update packages are named as `update-package_<update_version_number>.tar.gz`.

Updates are performed manually and can only be done by a security administrator.

Update packages are signed by BlackBerry and the digital signature is verified by the TOE automatically during the update process to ensure integrity. The provided SHA-256 checksum can be used by the Security Admin as a secondary check if desired.

To update the server, in the web UI, complete the following steps:

1. Click **System > Management > Update**.
2. Click the BlackBerry signed update file or drag and drop the file to the indicated area.
3. Click **Update.**

Update status and detailed information is displayed in the **System Update Status** window.

Please note that the SecuGATE won't apply any updates with invalid signatures or updates.

In case the update failed, the Security Administrator shall review the update log in the status window for the exact failure reason and contact the BlackBerry SecuSUITE Support team.

## 8.7 Server selftest

A server software integrity check and FIPS Object Module self-tests are executed during the server start-up.

If the self-test fails, an audit log is generated and all services that depend on the crypto module will be stopped automatically (SIP, SCA, HTTPS, SSH and other TLS connections). As a result, the access to the TOE via SSH and WebUI is blocked. The Security Administrator needs to access the TOE via the local console to resolve the incident. The detailed results of the software integrity check can be investigated in the log file of the Aide service (

Server self-test status can be seen in web UI by completing the following steps:

Click **System** > **System Status** > **Self-test**.

The self-test can be triggered by the Security Administrator by executing:

```
sudo secuadmin selftest
```

# 8.8 Viewing system logs

The following table lists the most relevant log files on the TOE. No administrative action is needed that audit logs are generated and persisted. To enable system logs please see 7.12 System status.

System and audit logs can additionally be written to an external syslog server by following the instructions in 7.7.5.

**Table 7: Log files**

| Log file | Description |
|---|---|
| /var/log/audit/audit.log | This file contains all audit events |
| /var/log/aide/aide.log | This file contains log messages about the last system integrity check |

Log files can be viewed using standard Linux file-viewing tools such as `less`, `more`, or `tail`. To view log files, you must log in as `secuadmin`. Format and contents for the system and audit logs are described in chapter **Error! Reference source not found.** and chapter 10.

The security admin can also view local audit events via the CLI by using the `ausearch` tool.

Complete local system logs including audit logs can be viewed by standard Red Hat tool `journalctl`.

# 8.9 Viewing call detail records

Call detail records are downloaded in CSV format by clicking **Administration > Export Archives,** and selecting **Call Detail Records.**

Downloaded voice call CDR files are identified with a default file name, such as `exportCDR-Primary Tenant-2022-04-29 15_04_37.343180.csv`, where the postfix figure indicates the download date of the file.

The CSV file contains the following comma separated values:

- **Server ID**: unique identifier of the TOE
- **Record ID, call-ID**: unique transaction sequence number
- **Calling party number**: calling party number (the call originator)
- **Caller tenant name**: tenant name of the caller (if applicable)
- **Called party number**: The call recipient or terminating number
- **Callee tenant name**: tenant name of the call recipient (if applicable)
- **Call Created**: time the call was initiated
- **Call Start**: time the call started
- **Call End**: time the call ended
- **Call Duration:** duration of the call
- **Reason Code**: The SIP reason code for the call release cause, and the call disposition information
- **Caller Routing**: call routing into TOE
- **Callee Routing**: call routing out of TOE
- **Timestamp**: timestamp of when the call record was created

- **Time Zone**: timezone of the call.

# 8.10 Date and time management

The current system date can be set manually via the Security admin interface by using the `date` command.

**For example:** `date --set="20220413 21:15:00"` will set the date to 9:14:00pm on 13th April 2022.

The current time in which the server is running can be queried in the web UI by clicking **System > System Status > Server > Server time.**

**Timezone Setting**

The server time zone can be set by the Security Administrator in the web UI by clicking **System > Configuration > Network Settings** and selecting the desired time zone in the 'General' section from the drop-down menu.

# 8.11 System control

System reboot and shutdown are performed in web UI by clicking **System > Management > Power.** Click the **Reboot** or **Shutdown** button.

A system reboot via the Security admin interface can be performed by typing the following command:

**`sudo systemctl reboot.`**

# 8.12 System status log

System status can be viewed by clicking **System > System Status.**

The following status can be viewed:

- **CPU**: The current CPU load information.

  **For example:**

  ```
  Type    Load
  User    8.2%
  System  6.4%
  Other   0%
  ```

- **Disk**: The current disk usage information.

  **For example:**

  ```
  Mount point          Blocks       Used         Available     % used
  /       3997376      62916        3708364      2
  ```

- **Memory**: The current memory usage information.

  **For example:**

  ```
  Total   16266428 MB
  Free    10855648 MB
  Used    1713244 MB (10.53%)
  Buffer/Cache         3697536 MB
  ```

- **NTP**: The NTP server synchronization information.
  NTP can be synchronized or unsynchronized.
- **Ethernet**: The current NIC interface status for all available interfaces.
  The status can be UP, DOWN, or UNKNOWN.

  **For example:**

  ```
  Name    Status
  lo      UNKNOWN
  eth0    UP
  eth1    UP
  eth2    DOWN
  ```

- **Fan:** The FAN RPM information of the fans in the system (if available).

  **For example:**

  ```
  Fan no. RPM
  FAN1    3500
  ```

- **Self-test**: Information about the latest self-test initialization, latest self-test, and the status of the latest self-test.

  **For example:**

  ```
  Last integrity init : 2021-07-23 00:00:43.917678632 +0000
  Last integrity check: 2021-07-24 00:00:22.136044729 +0000:
  Result: FAILED
  ```

- **Users**: A summary status of the SecuSUITE clients in the system.

  **For example:**

  ```
  all: 100
  new: 10
  invited: 2
  activated: 90
  online: 4
  offline: 96
  ```

- **Server**: Includes general server information.

  **For example:**

  ```
  Uptime: 17:34:59 up 2 days, 4:17, 0 users, load average: 0.12, 0.11, 0.13
  Server time: Wed Jul 24 17:34:59 2021
  Version: 5.0.0-210
  Hostname: SecuGATE-LVN
  ```

# 8.13 Querying Real-time Connection Status

The real-time connection status for VoIP endpoints and SIP trunk telecommunication devices can be viewed in the CLI interface by typing the following command:

```
sudo secuadmin call-status
```

The command displays a list of ongoing calls with information for caller number, callee number, call start time and current call duration. Furthermore, it shows the established connections to telecommunication devices such as a connected PBX or another TOE

For example:

```
# secuadmin call-status
{
       "activeCalls": [
            {
                "caller": "+4911111111",
                "caller_address": "127.0.0.1:40000",
                "callee": "+49222222222",
                "callee_address": "127.0.0.1:5060",
                "call_start_time": "2022-04-29 15:08:34.960",
                "call_duration": "00:00:02.010"
            }
        ],
        "sipConnections": {
            "sipTrunkStatus": "active",
            "connections": [
                {
                    "local": "127.0.0.1:5061",
                    "remote": "0.0.0.0:*",
                    "status": "LISTEN",
                    "service": "sip"
                },
                {
                    "local": "127.0.0.1:58907",
                    "remote": "127.0.0.2:5060",
                    "status": "ESTABLISHED",
                    "service": "sip"
                }
            ]
        }
}
```

# 8.14 TSF data erasure

The TOE stores private keys used during TLS and SSH operation and allows administrators to securely wipe these private keys.

- Appropriate private TLS keys:
  - SIP external (to SecuGATE clients),
  - SIP trunk (to external SIP servers),
  - SCA0 TLS,
  - SCA1 TLS,
  - syslog client,
  - web portal,
  - push client,
  - external CA client
- Private embedded CA keys
- Private Voice SMIME keys
- SSH server host key

The TOE allows administrators to perform a secure wipe operation that clears TOE configuration data including these private keys. A secure wipe operation overwrites data to be deleted with three iterations of random data and the fourth iteration with zeros. The `sudo secuwipe –s –o –w` command will cause the TOE to perform a secure wipe operation clearing all TOE configuration data in the root folder which include all of these private keys. The following are the full set of options available with the `secuwipe` command.

```
secuwipe -h -d -l logfile -e -P -o -r -s
   h: help
   d: dry-run
   l: logfile

   s: Service data content will not be wiped
   e: Do not exit from secuwipe
   o: Home data content will not be wiped
   r: Root system data content will not be wiped
   P: Forces complete partition wipe. This may take time
```

After this action, the system will be in a non-operational state and cannot be recovered.

# 9. Audit logs

6

The TOE uses the Linux audit system (auditd) to log the auditing events. The service is using log rotation with 5 log files (audit.log, audit.log.1…audit.log.4). During log rotation (when a new audit.log is created) the oldest file (audit.log.4) is deleted, all numbered files are renamed with incremented numbers and old 'audit.log' is moved to audit.log.1. The log rotation is triggered when the current /var/log/audit/audit.log exceeds 6 Mbyte. Hence the disk space available for logging is at least 30 Mbyte.

The TOE reserves enough disk space for the /var/log sub-tree so that there is enough disk space available for logging at any time.

The security admin can view local audit events via the CLI by using the `ausearch` tool.

**For example:** *`sudo ausearch -i | grep -B1 <pattern>`*

Each audit record has the following format:

**type**=UNKNOWN[1166] *msg*=audit(08/16/2021 12:40:30.232:3700) : pid=2746 uid=root auid=unset ses=unset subj=system_u:system_r:unconfined_service_t:so **msg**='"SSM-AUDIT: "Admin: Login: success; user: admin; local: 127.0.0.1:8000; remote: 127.0.0.1 exe=/usr/sbin/uaudit hostname=? addr=? terminal=? *res*=success'

Based on this example, the following information is common:

- `msg=audit(08/16/2021 12:40:30.232:3700)`: identifies the date and time of the event.
- `res=success`: identifies the outcome of the event (`success` or `failed`).
- `hostname=?`: the character '?' should not be read as a regular expression or placeholder is used as shown in case the information is unavailable or not applicable

The event type and the identifier of the subject responsible for the activity information depends on the audit message (examples below). Type (**type**) and message (**msg**) fields can be found in the audit event messages (marked in bold in the example above).

In addition, the examples below may have the following place holders:

- **<remote_ip>**: the remote IP in the event (if applicable). The port number (if applicable) is random in the following examples.
- **<local_ip>**: the local IP in the event (if applicable).
- **<username>:** the username associated with the event (if applicable).
- **<reason_for_failure>**: the reason for failure of the event (if applicable).
- **<hostname>**: the local system hostname or domain

Depending on the dedicated audit log, not all the supporting fields might be included to the logs and described fields might be included to other fields.

**FCS_HTTPS_EXT.1**

Failure to establish a HTTPS session.

**msg**='AUDIT; TLS_TUNNEL; event: TLS connection; clientMode: 0; local: <local_ip>:443; remote: <remote_ip>; reason: err = <reason_for_failure>; res: failure exe=/usr/sbin/uaudit hostname=? addr=? terminal=? res=failed'

Successfully establish a HTTPS session.

**msg**='AUDIT; TLS_TUNNEL; event: TLS connection; clientMode: 0; local: <local_ip>:443; remote: <remote_ip>; reason: err = 0; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**FCS_SSHS_EXT.1**

Establishment of an SSH session includes the following audit, with <RESULT> being either 'failed' or 'success'

**type**=CRYPTO_SESSION

**msg**='op=<reason_for_failure> direction=? cipher=? ksize=? rport=53772 laddr=<local_ip> lport=22 exe=/usr/sbin/sshd hostname=? addr=<remote_ip> terminal=? res=<RESULT>'

followed by:

**type**=USER_LOGIN

**msg**='op=login acct="(unknown)" exe="/usr/sbin/sshd" hostname=? addr=<remote_ip> terminal=ssh res=failed'

Reasons for failure of the SSH session establishment are listed in the table below:

**Table 8: Failure reasons for SSH session establishment as a server**

| | Failure Text | Description |
|---|---|---|
| 1 | "unsupported-cipher" | Cipher requested by the SSH client is not supported |
| 2 | "no matching MAC found" | No matching MAC (HMAC-SHA2-256) found based on the offer of the SSH client. |
| 3 | "no matching Host key type found" | No matching host key type (i.e., ecdsa_sha2-nistp256) found based on the offer of the SSH client |
| 4 | "no matching key exchange method found" | No matching key exchange algorithm (e.g. ECDH-SHA2-NISTP256) found based on the offer of the SSH client. |

**FCS_TLSC_EXT.2, FIA_UAU.2/TC**

A failure to establish a TLS session as a **TLS client** will result into one of the following Audit message types:

**msg**='DOS-DET; TLS; event: TLS connection; local: <local_ip>; remote: <remote_ip>; reason: <reason_for_failure>; res: failure exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=failed'

**msg**='AUDIT; TLS_TUNNEL; event: TLS connection; clientMode: 1; local: <local_ip>; remote: <remote_ip>; reason: <reason_for_failure>; res: failure exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=failed'

**msg**='AUDIT; EST; event: TLS connection; clientMode: 1; local: <local_ip>; remote: <remote_ip>, reason: <reason_for_failure>; res: failure

The reasons for failure of TLS session establishment are listed in Table 9.

**Table 9: Failure reasons for TLS session establishment**

| Failure Text | Description |
|---|---|
| "handshake failure" | Unable to negotiate an acceptable set of security parameters given the options available. |
| "digest check failed" | Digest check of the server finished handshake message failed. |
| "certificate verify failed" | Verification of the certificate provided by the server failed. |
| "wrong certificate type" | Public key of provided server certificate does not match selected cipher suite in Server Hello message. |
| "internal error" | Crypto lib cannot proceed with encryption due to an internal error. |
| "wrong curve" | Failure of ECDHE key agreement due to wrong elliptic curve being used by the TLS Server. |
| "unknown cipher returned" or "no shared cipher" | Cipher suite returned by the TLS Server is unknown/not matching to the proposed list. |
| "unsupported protocol" or "Wrong TLS version" | TLS Server configured to an unsupported TLS protocol version (TLS 1.2 supported only) |
| "bad signature" | RSA signature validation check failed during the handshake. |

Successful TLS session as SIP client will result in one of the following Audit message types in the logs:

**msg**='AUDIT; TLS; event: TLS connection; local: <local_ip>; remote: <remote_ip>; reason: connect success; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**=AUDIT; TLS_TUNNEL; event: TLS connection; clientMode: 1; local: <local_ip>; remote: <remote_ip>; reason: err = 0; res: success

**msg**='AUDIT; EST; event: TLS connection; clientMode: 1; local: <local_ip>; remote: <remote_ip>; res: success

**FCS_TLSS_EXT (SecuGATE as TLS Server)**

A failure to establish a TLS session as a **TLS server** will result into one of the following Audit message types:

**msg**='DOS-DET; TLS; event: TLS connection; local: 5061; remote: <remote_ip>; reason: <reason_for_failure>; res: failure exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=failed'

**msg**='AUDIT; TLS_TUNNEL; event: TLS connection; clientMode: 0; local: <local_ip>; remote: <remote_ip>; reason: err = <reason_for_failure>; res: failure exe=/usr/sbin/uaudit hostname=? addr=? terminal=? res=failed'

The reasons for failure of TLS session establishment are listed in Table 9.

Successful TLS session as TLS server will result into one of the following Audit messages:

**msg**='AUDIT; TLS; event: TLS connection; local: <local_ip>:5061; remote: <remote_ip>; reason: accept success; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**=AUDIT; TLS_TUNNEL; event: TLS connection; clientMode: 0; local: <local_ip>; remote: <remote_ip>; reason: err = 0; res: success

**FIA_AFL.1: Web**

The limit for unsuccessful login attempts was reached or exceeded in the web interface.

**msg**='AUDIT; WEB_ADMIN; event: Login-Blocked; reason: too many failed login attempts; user: admin; local: 127.0.0.1:8000; remote: <remote_ip>; res: failure exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=failed'

**FIA_AFL.1: SSH password**

The limit for unsuccessful login attempts was reached or exceeded in the SSH password login interface.

**type**=RESP_ACCT_LOCK msg=audit(1656077990.943:487): pid=3292 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 **msg**='pam_faillock uid=502  exe="/usr/sbin/sshd" hostname=? addr=? terminal=? res=success'

**FIA_UAU.2/VvoIp**

Successful or failed registration of endpoint

SIP successful registration:

**msg**='AUDIT; SIP; event: Authentication; user: <sips:<username>@<hostname>>; local: <local_ip>:5061; remote: <remote_ip>; reason: 200 OK; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

SIP failed registration:

**msg**='AUDIT; SIP; event: Authentication; user: <sips:<username>@<hostname>>; local: <local_ip>:5061; remote: <remote_ip>; reason: <reason_for_failure>; res: failure exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=failed'

SCA successful registration:

**msg**='AUDIT; SECUSERV; event: Authentication; remote: <remote_ip>; user: <username>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

SCA failed registration:

**msg**='DOS-DET; SECUSERV; event: Authentication, remote: <remote_ip>; user: <username>; reason: <reason_for_failure>; res: failure exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=failed'

**FIA_UAU_EXT.2, FIA_UIA_EXT.1, FTA_SSL.3, FTA_SSL.4, FTA_SSL_EXT.1: Web**

All use of identification and authentication mechanisms in the web interface

All use of identification and authentication mechanisms in web password interface

Termination of remote session by the locking mechanism

Termination of interactive session

Any attempts at unlocking of interactive session

Login and authentication:

**msg**='AUDIT; WEB_ADMIN; event: Login-Login; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

Logout:

**msg**='AUDIT; WEB_ADMIN; event: Login-Logout; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

Session expiration:

**msg**='AUDIT; CLI_ADMIN; event: Login-Expired; user: <username>; local: None; remote: None; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

Login failure:

**msg**='AUDIT; WEB_ADMIN; event: Login-Login; reason: Unable to log in with provided credentials.; user: <username>; local: 127.0.0.1:8000; remote: <remote_ip>; data: {'username': '<username>', 'password': '***'}; res: failure exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=failed'

Password change:

**msg**='AUDIT; WEB_ADMIN; event: Admin-Password changed; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Admin-Password not changed; reason: authentication failed; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; data: {'current_password': '***', 'new_password': '***'}; res: failure exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=failed'

**FIA_UAU_EXT.2, FIA_UIA_EXT.1, FTA_SSL.3, FTA_SSL.4, FTA_SSL_EXT.1: SSH password**

All use of identification and authentication mechanisms in the SSH password interface

Termination of remote session by the locking mechanism

Termination of interactive session

Any attempts at unlocking of interactive session

Termination of local session by the locking mechanism

Login:

**type**=USER_START, **msg**='op=login id=<username>exe=/usr/sbin/sshd hostname=<remote_ip> addr=<remote_ip> terminal=/dev/pts/3 res=success'

Logout:

**msg**='AUDIT; sshd; event: Disconnection; data: Disconnected from user <username> <remote_ip> port <remote_port>; res: success

**type**=USER_END, **msg**='op=PAM:session_close grantors=pam_selinux,pam_loginuid,pam_selinux,pam_namespace,pam_keyinit,pam_keyinit,pam_limits,pam_systemd,pam_unix,pam_lastlog acct=<username>exe=/usr/sbin/sshd hostname=<remote_ip>addr=<remote_ip> terminal=ssh res=success'

Session expiration:

**msg**='AUDIT; User <username> timedout; res: success

Authentication succeeded:

**type**=USER_AUTH, **msg**='op=success acct=<username> exe=/usr/sbin/sshd hostname=? addr=<remote_ip> terminal=ssh res=success'

Authentication failed:

**type**=USER_AUTH, **msg**='op=password acct=<username>exe=/usr/sbin/sshd hostname=? addr=<remote_ip> terminal=ssh res=failed'

Password change:

**msg**='op=PAM:chauthtok grantors=pam_pwquality,pam_unix acct=<username> exe=/usr/bin/passwd hostname=? addr=? terminal=pts/1 res=success'

**FIA_UAU_EXT.2, FIA_UIA_EXT.1, FTA_SSL.3, FTA_SSL.4, FTA_SSL_EXT.1: SSH public key**

Authentication:

**type**=USER_AUTH **msg**='op=pubkey_auth rport=54712 acct=<username> exe="/usr/sbin/sshd" hostname=? addr=<remote_ip> terminal=? res=success'

Login:

**type**=USER_START **msg**='op=PAM:session_open grantors=pam_selinux,pam_loginuid,pam_selinux,pam_namespace,pam_keyinit,pam_keyinit,pam_limits,pam_systemd,pam_unix,pam_lastlog acct=<username> exe="/usr/sbin/sshd" hostname=<remote_ip> addr=<remote_ip> terminal=ssh res=success'

Logout:

**type**=USER_END, **msg**='op=PAM:session_close grantors=pam_selinux,pam_loginuid,pam_selinux,pam_namespace,pam_keyinit,pam_keyinit,pam_limits,pam_systemd,pam_unix,pam_lastlog acct=<username> exe="/usr/sbin/sshd" hostname=<remote_ip> addr=<remote_ip> terminal=ssh res=success'

Upload of public key via ssh-copy-id

Login:

**type**=USER_LOGIN, **msg**='op=login id=<username> exe=/usr/sbin/sshd hostname=<remote_ip> addr=<remote_ip> terminal=ssh res=success'

Configuration change of ~/.ssh/authorized_keys:

**type**=SYSCALL msg=audit(10/29/2021 00:48:45.713:562) : arch=x86_64 syscall=open success=yes exit=3 a0=0x1727000 a1=O_WRONLY|O_CREAT|O_APPEND a2=0666 a3=0x0 items=2 ppid=4232 pid=4252 auid=secuadmin **uid**=<username> gid=secuadmin euid=secuadmin suid=secuadmin fsuid=secuadmin egid=secuadmin sgid=secuadmin fsgid=secuadmin tty=(none) ses=14 comm=sh exe=/usr/bin/bash subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 **key**=authorized_keys

**FIA_UAU_EXT.2, FIA_UIA_EXT.1, FTA_SSL.3, FTA_SSL.4, FTA_SSL_EXT.1: Local**

All use of identification and authentication mechanisms in the local console interface

All use of identification and authentication mechanisms in the SSH password interface

Termination of remote session by the locking mechanism

Termination of interactive session

Any attempts at unlocking of interactive session

Termination of local session by the locking mechanism

Login:

**type**=USER_START, **msg**='op=PAM:session_open grantors=pam_selinux,pam_loginuid,pam_selinux,pam_namespace,pam_keyinit,pam_keyinit,pam_limits,pam_systemd,pam_unix,pam_lastlog acct=<username> exe=/usr/bin/login hostname=<hostname> addr=? terminal=tty1 res=success'

Logout:

**type**=USER_END, **msg**='op=PAM:session_close grantors=pam_selinux,pam_loginuid,pam_selinux,pam_namespace,pam_keyinit,pam_keyinit,pam_limits,pam_systemd,pam_unix,pam_lastlog acct=<username> exe=/usr/bin/login hostname=<hostname> addr=? terminal=tty1 res=success'

Session expiration:

**type**=USER_END, **msg**='op=PAM:session_close grantors=pam_selinux,pam_loginuid,pam_selinux,pam_namespace,pam_keyinit,pam_keyinit,pam_limits,pam_systemd,pam_unix,pam_lastlog acct=<username> exe=/usr/bin/login hostname=<hostname> addr=? terminal=tty1 res=success'

Authentication:

**type**=USER_AUTH, **msg**='op=PAM:authentication grantors=? acct=<username> exe=/usr/bin/login hostname=<hostname> addr=? terminal=tty1 res=failed'

The value in the "res" field for the audit event above can be either 'failed' or 'success'.

**FCS_SSHS_EXT.1.8**

SSH rekeying produces following kind of audit logs.

**type**=CRYPTO_KEY_USER msg=audit(1657299222.297:5378): pid=26373 uid=0 auid=502 ses=275 subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 **msg**='op=destroy kind=session fp=? direction=both spid=26382 suid=502 rport=54628 laddr=<local_ip> lport=22  exe="/usr/sbin/sshd" hostname=? addr=<remote_ip> terminal=? res=success'

**FIA_x509_EXT.1/Rev**

Unsuccessful attempt to validate a certificate

**msg**= 'AUDIT; X509_VERIFY; event: Certificate check; remote: <remote_ip>; subject: <certificate_subject>; reason: <reason_for_failure>; res: failure exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=failed'

| Reason code | Description |
|---|---|
| 2 | Unable to get issuer certificate |
| 3 | Unable to get CRL |
| 7 | Certificate signature failure |
| 9 | Certificate not yet valid |
| 10 | Certificate has expired |
| 11 | CRL not yet valid |
| 12 | CRL has expired |
| 19 | Self-signed certificate in certificate chain (hence the certificate does not chain to a trusted root) |
| 20 | Unable to get local issuer certificate |
| 23 | Certificate revoked |
| 24 | Invalid CA certificate |
| 26 | Invalid purpose |
| 35 | Key usage does not include CRL Sign |
| 62 | Hostname mismatch |
| 63 | Email mismatch |
| 64 | IP address mismatch |
| 70 | Issuer look-up error |
| 94 | EC key explicit parameters |

**FDP_RIP.1**

Completed factory reset is logged into a newly created 'audit.log' file after the reset procedure has been completed:

**msg**='AUDIT; CLI_ADMIN; event: System wipe started; user: <username>; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

This is logged in the beginning of reset.

**FMT_MOF.1, FPT_TUD_EXT.1**

Any attempt to initiate a manual update

Initiation of update:

**msg**='AUDIT; WEB_ADMIN; event: System-Update started; user: <username>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

Successful update:

**msg**='AUDIT; WEB_ADMIN; event: System-Update; details: Server updated from version 5.0.30 to 6.6.6; user: <username>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

Update failure:

**msg**='AUDIT; WEB_ADMIN; event: System-Update; reason: Signature verification failed; user: <username>; res: failure exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=failed'

**FMT_MTD.1/CoreData: Web**

All management activities of TSF data via the web UI.

The res of the audit events below can be **failed** or **success**.

**msg**='AUDIT; WEB_ADMIN; event: System-Config updated; details: <changed_configuration_items>; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

Mobile client management:

**msg**='AUDIT; WEB_ADMIN; event: Subscribers listed; user: admin; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Subscriber <subscriber_name> created; user: <username>; userLevel: 300; tenantUid: <tenant_id>; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Subscriber <subscriber_name> invited; user: <username>; userLevel: 300; tenantUid: <tenant_id>; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Subscriber <subscriber_name> updated; user: <username>; userLevel: 300; tenantUid: <tenant_id>; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=?

res=success'

**msg**='AUDIT; WEB_ADMIN; event: Subscriber <subscriber_name> deleted; user: <username>; userLevel: 300; tenantUid: <tenant_id>; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

Tenant management:

**msg**='AUDIT; WEB_ADMIN; event: Tenants listed; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Tenant <tenant_name> created; user: <username>; userLevel: 300; tenantUid: <tenant_id>; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Tenant <tenant_name> updated; user: <username>; userLevel: 300; tenantUid: <tenant_id>; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Tenant <tenant_name> deleted; user: <username>; userLevel: 300; tenantUid: <tenant_id>; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

Web UI and Tenant Admin management:

**msg**='AUDIT; WEB_ADMIN; event: Admins listed; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Admin <admin_name> created; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Admin <admin_name> invited; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Admin <admin_name> updated; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Admin <admin_name> deleted; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

Certificate management:

**msg**='AUDIT; WEB_ADMIN; event: Certificates-Certificates imported; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; data: {'certs': 'Li8AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...', 'password': '***'}; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Certificates-Certificates not imported; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; data: {'certs': 'Li8AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...', 'password': '***'}; res: failure exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=failed'

**msg**='AUDIT; WEB_ADMIN; event: Certificates-Certificates exported; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; data: {'names': [], 'include_keys': True, 'password': '***'}; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Certificates-Certificates key generated; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; data: {'certname': <cert_name>}; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Certificates-Certificates CSR created; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; data: {'name': <cert_name>, 'subject': <cert_subject>, 'san': <cert_san>, 'service':<cert_service>, 'sign_with_local': False}; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Certificates-Certificates CSR created; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; data: {'name': <cert_name>, 'subject': <cert_subject>, 'san': <cert_san>, 'service': <cert_service>, 'sign_with_local': True}; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Certificates-Status listed; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; data: {'name': <cert_name>}; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Certificates-Certificates <cert_name> set; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; data: {'name': <cert_name>, 'certificate': 'LAAAAAAAAAAAAAAAAAAAA...', 'services': [<cert_service>]}; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: Certificates-Certificates <cert_name> not set; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; data: {'name': <cert_name>, 'certificate': 'LSAAAAAAAAAAAAAAAAAAAAAA...', 'services': [<cert_service>]}; res: failure exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=failed'

**msg**='AUDIT; WEB_ADMIN; event: Certificates-Certificates <cert_name> deleted; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; data: {'name': <cert_name>, 'delete_key': True}; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

Adding/removing NTP server:

**msg**='AUDIT; WEB_ADMIN; event: System-Config updated; details: ntpServers; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: System-NTP created; details: <server_ip>:<server_port>; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; WEB_ADMIN; event: System-NTP deleted; details: <server_ip>:<server_port>; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**FMT_MTD.1/CoreData: CLI**

All management activities of TSF data via the CLI interface.

secuadmin activities

All secuadmin activites are logged in the following format:

**msg**='AUDIT; CLI_ADMIN; event: <command>; user: <username>; details: ['/bin/secuadmin', <command>,<parameter_list>]; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

The commands are as follows:

| Command | Description |
|---|---|
| **config** | Change system configurations items |
| **gen-key** | Create private public key pair |
| **gen-selfsigned** | Generate self-signed CA |
| **gen-csr** | Generate a certificate signing request |
| **set-ca** | Set and activate embedded CA |
| **sign-local-csr** | Signs a local CSR |
| **sign-csr** | Signs an externally-generated CSR |
| **set-service** | Sets and activates a local certificate to a service |
| **delete-ca** | Deletes the embedded CA |
| **delete-cert** | Deletes a certificate |
| **delete-trusted** | Deletes trusted certificates |
| **upload-trusted** | Uploads a trusted CA .tar file bundle |
| **upload-cert** | Uploads a certificate |
| **revoke-cert** | Revokes a certificate |
| **gen-crl** | Generates a certificate revocation list (CRL) |
| **export-certs** | Exports certificates in the system |
| **import-certs** | Imports certificates to the system |
| **revoke-user** | Set a user to a revoked state |
| **revoke-list** | Set a list of users to a revoked state |
| **delete-keys** | Deletes all existing certificates |
| **unblock-user** | Unblock a web UI admin user |
| **change-password** | Change the password for web UI admin user |
| **server-update** | Update the server |
| **selftest** | Perform a server self-test |

| Command | Description |
|---|---|
| **status** | Get the system status |
| **user-status** | Get the list of a SIP user's status |
| **license** | Get the current license info's status |
| **rhlicense** | Red Hat license upload |
| **call-status** | Real-time connection status of srtp endpoints and telecommunication devices. |
| **bootstrap** | Bootstrap is not used by system administrators.<br>The bootstrap command is called by an internal component during the first boot of the system after installation. |
| **update-ca-trust** | Update the trusted certificate bundle after adding new certs that needs to be trusted. |
| **restart-cert-services** | Restart of the cert service after updating the certificates. |
| **self-signed-nginx** | Creation of a self-signed certificate for the Web Portal UI (nginx web server) |
| **keep-sshd-alive** | Sets the sshd alive mode on (which keeps the ssh connection alive on aide/fips failures) |
| **wipe-certs** | Wipe all the certificates |

The parameter list is command-dependent.

Execute the secusetup command for management interface setup

type=USER_CMD; msg='cwd="/home/secuadmin" cmd="/bin/secusetup" terminal=tty1 res=success'

Add a console user:

```
type=UNKNOWN[1166] msg='SSM-AUDIT: Add secuadmin: success; Account '<account_username>'
created by user '<username>' with secuadmin rights exe="/usr/sbin/uaudit" hostname=?
addr=? terminal=? res=success'
```

Delete a console user:

```
type=UNKNOWN[1166] msg='SSM-AUDIT: Delete secuadmin: success; Account
'<account_username>' deleted by user '<username>' exe="/usr/sbin/uaudit" hostname=?
addr=? terminal=? res=success'
```

Unblock a console user:

**msg**='cwd=/home/secuadmin cmd=/sbin/faillock --user <username> --reset terminal=pts/0 res=success'

**FMT_SMF.1**

Modification of the TOE call details records (CDR) (TD0419 applied)

Enabling/disabling VVoIP endpoint/device features

Not applicable

**FPT_STM_EXT.1**

**Discontinuous changes in time by admin date command.**

The time change for discontinuous changes by the date command is reported by a series of audit logs that indicate the start and success of the change. Time change start is reported by:

**key**=time_change

where the old (current) time is indicated by the log time-stamp. The execution success is reported by:

**key**=time_newThe 'time_new' log entry still shows the old time and the new-time is than visible as the following log entry

**Discontinuous changes in time by NTP:**

msg='AUDIT; NTP; event: Time change; oldTime: <old_time>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**Server time zone change**

The time zone change is logged by

type=CONFIG_CHANGE msg=audit(10/08/2021 15:39:07.688:4677) : auid=unset ses=unset op=updated_rules path=/etc/localtime **key**=time_local list=exit res=yes

type=UNKNOWN[1166] msg=audit(10/08/2021 15:39:31.560:4748) : pid=2565 uid=root auid=unset ses=unset subj=system_u:system_r:unconfined_service_t:so

**msg**='AUDIT; WEB_ADMIN; event: System-Config updated; details: timezone; user: <username>; userLevel: 300; local: 127.0.0.1:8000; remote: <remote_ip>; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**FTP_TRP.1/Admin**

Initiation of the trusted path.

Termination of the trusted path.

Failure of the trusted path functions.

Web, SSH and local administrative audit logs are described above.

**FPT_FLS.1.1/Self-test success and failure**

The execution of the success and failure of the self-tests are logged by:

**msg**='AUDIT; event: System-Selftest Performing file system integrity check;; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; event: System-Selftest Performing FIPS mode check;; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success' exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

**msg**='AUDIT; CLI_ADMIN; event: selftest; user: <username>; details: ['/bin/secuadmin', 'selftest']; res: success exe="/usr/sbin/uaudit" hostname=? addr=? terminal=? res=success'

# 9.1 Audit entries for starting and stopping auditing services

**FAU_GEN.1**

Red Hat 8 auditing service starts automatically in the system startup. Starting the service and related daemon produces the following audit log:

type=**SERVICE_START** msg=audit(07/07/2022 15:13:11.893:8) : pid=1 uid=root auid=unset ses=unset subj=system_u:system_r:init_t:so **msg='unit=uaudit** comm=systemd exe=/usr/lib/systemd/systemd hostname=? addr=? terminal=? res=success'

Stopping the uaudit service 'uaudit' results in the following audit log:

type=**SERVICE_STOP** msg=audit(07/15/2022 10:36:49.053:1289) : pid=1 uid=root auid=unset ses=unset subj=system_u:system_r:init_t:so msg='unit=uaudit comm=systemd exe=/usr/lib/systemd/systemd hostname=? addr=? terminal=? res=success'

Stopping the audit daemon 'auditd' results in following audit log:

type=**DAEMON_END** msg=audit(07/15/2022 10:27:06.365:4740) : op=**terminate** auid=root pid=5272 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 res=success

Stopping the uaudit service 'uaudit' results in the following audit log:

During a normal system shut-down both audit logs (SERVICE_STOP AND DAEMON_END) would be logged to the local TOE audit logs only. Because the syslog connection has already been terminated, these audits appear only in the local TOE audit files.

# 10. System Logs

Every time a system status is written to the syslog, the following status are generated:

**NTP Status log**

NTP status can be synchronized or unsynchronized.

Format:

{"status_ntp": "synchronised"}

**NIC Status log**

The NIC status for all interfaces can be UP, DOWN or UNKNOWN

Format:

{"status_nic": {"lo": "UNKNOWN", "etho": "UP", "eth1": "UP", "eth2": "UP"}}

**CPU Status log**

Format:

{"status_cpu": {"us": 1.8, "sy": 3.7, "ni": 0.0, "id": 94.5, "wa": 0.0, "hi": 0.0, "si": 0.0, "st": 0.0},

| Indicator | Description |
|---|---|
| us | percentage of time running un-niced user processes |
| sy | percentage of time running kernel processes |
| ni | percentage of time running niced user processes |
| id | percentage of time in idle |
| wa | percentage of time waiting for I/O |
| hi | percentage of time serving hardware interrupts |
| si | percentage of time serving software interrupts |
| st | percentage of time hypervisor actions |

**Memory Status log**

**Format:**

"status_mem": {"total": 16266284.0, "free": 12506124.0, "used": 1736868.0, "buff/cache": 2023292.0, "use%": 10.68}}

| Indicator | Description |
|---|---|
| total | Total memory in kilobytes |

| | |
|---|---|
| free | Free memory in kilobytes |
| used: | Used memory in kilobytes |
| buff/cache | Memory used by kernel buffers and page cache |
| use% | Percentage of used memory |

**File System Status log**

For each file mounted file system:

{"status_fs": {"/dev": {"blocks": 8101852, "used": 0, "available": 8101852, "use%": 0}, "/dev/shm": {"blocks": 8133140, "used": 0, "available": 8133140, "use%": 0...}}}

| Indicator | Description |
|---|---|
| blocks | Number of 1024 byte blocks allocated for the file system |
| used | Number of used blocks |
| available | Number of available blocks |
| use% | Percentage of used blocks |

**FAN Status log**

Format:

{"status_fan": {}}

Showing the RPM of each fan. The fan status is not available in the evaluated configuration using ESXi.

# 10.1 Audit entries for system logs

Starting and stopping the system status logging produces audit logs in the format below:

**type**=SERVICE_START msg=audit(1656082957.155:1170): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:so **msg**='unit=status-logger comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'

**type**=SERVICE_STOP msg=audit(1656082957.155:1171): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:so **msg**='unit=status-logger comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'

# 11. Annex

## 11.1 Web UI Structure

| Main | Sub-Section | Subsub-Section | Description | SFR relevant |
|---|---|---|---|---|
| User Accounts | | | Used to set-up VoIP endpoint accounts (from the primary tenant). Endpoint accounts are required to register VoIP endpoints to the SecuGATE. | No |
| Tenants | | | Allows management of tenant accounts. Tenant account are used to offer independent management of end user accounts from e.g. different company departments. Tenant admin can manage (add, invite, delete) their own user group as well as a couple of tenant specific settings. | No |
| Administration | Manage Administrators | | Allows creation and management of additional Security Admins | Yes |
| | Change Password | | Change password of the current user. | Yes |
| | Export Archives | | Create and download CDRs. | Yes |
| | Monitoring | | View and download system and audit logs. | Yes |
| System | Management | Update | Upload and initiate trusted system updates. | Yes |
| | | Power | Allows reboot and system shutdown. | No |
| | | Backup&Restore | Backup and Restore system databases and configuration files. | No |
| | Configuration | Network Settings | Set-up of system network interfaces.<br>Configuration of NTP Servers. | Yes |
| | | RTP Settings | RTP Port range configuration.<br>Audio codecs<br>SRTP for Breakout | No |
| | | Advanced | TLS Peer Validation Identifiers configuration. | Yes |
| | | Restricted ciphers | Restricted TLS ciphers<br>Restricted SSH ciphers | Yes |

| | | | | |
|---|---|---|---|---|
| | | Message Storage Settings | Retention time for message records. | No |
| | | Export Archives | Enabling CDR and message recording features. | Yes |
| | | Client Configuration | Push service and Push Server Configuration. | No |
| | | | RTP Timeout | Yes |
| | | Invitation | Configuration of delivery method for admin passwords and activation code for VoIP endpoint registration (delivered via email or shown on the screen) | No |
| | | Email Server Settings | Settings for the email server for outgoing emails. | No |
| | | High Availability | Configuration of high availability set-up.<br>**Please note** that HA cluster set-up is not evaluated and must not be configured in the evaluated configuration. | No |
| | | External CA | Configuration of external CA settings | No |
| | | | Configuration of TLS settings for external CA | Yes |
| | | PKI | Definition of default values for subject and subject alternative name fields that should be used for certificate signing requests created by the SecuGATE for certificate enrollment. | No |
| | | Auto Provisioning | Settings for LDAP based external system configuration provisioning. | |
| | | Logging | External Syslog Server configuration | Yes |
| | | | SIP traces configuration | No |
| | | | SIP PCAP Configuration | No |
| | | PABX Transport | Configuration of a connected PABX including TLS | Yes |
| | | SIP Trunk Settings | Allows the differentiation of calls to PABX-internal numbers (that can still be perceived as trusted calls) and numbers (or number blocks) that will be forwarded the PSTN by the PABX.<br>The different connection types will be indicated in the call screen and can be notified additionally via voice tags. | No |
| | | Voice Tags | Allows the upload of different voice tags per language. SecuGATE can be configured in the "SIP Trunk | No |

| | | | | |
|---|---|---|---|---|
| | | | Settings" to play a voice tag into the voice stream for breakout into untrusted networks. | |
| | Certificates | | Certificate management (please see chapter 7.2) | Yes |
| | Status | | Visualization of System Status | Yes |
| About | | | About screen indicates the TOEs release version number. | Yes |

# 12. Legal Notice

7