System Security

This chapter describes the StarOS security features.

This chapter explores the following topics:

Per-Chassis Key Identifier

A user can set a unique chassis key which will work only for a chassis or for any set of chassis that will share the same configuration information.

The chassis key consists of 1 to 16 alphanumeric ASCII characters. The chassis key plain-text value is never displayed to the user; it is entered interactively and not echoed to the user.

On the ASR5500 the encrypted chassis key is stored in the midplane EEPROM and shared by both MIO/UMIO/MIO2s.

If the chassis key identifier stored in the header comment line of the configuration file does not match the chassis key, an error message is displayed to the user. The user can change the chassis key value simply by entering the chassis key again. The previous chassis key is replaced by a new chassis key. The user is not required to enter a chassis key.

If the user does not configure a chassis key, the system generates a unique value for that chassis.

Important Changing a chassis key may invalidate previously generated configurations. This is because any secret portions of the earlier generated configuration will have used a different encryption key. For this reason the configuration needs to be recreated and restored.

Important To make password configuration easier for administrators, the chassis key should be set during the initial chassis set-up.

The configuration file contains a one-way encrypted value of the chassis key (the chassis key identifier) and the version number in a comment header line. These two pieces of data determine if the encrypted passwords stored within the configuration will be properly decrypted.

While a configuration file is being loaded, the chassis key used to generate the configuration is compared with the stored chassis key. If they do not match the configuration is not loaded.

The user can remove the chassis key identifier value and the version number header from the configuration file. Also, the user may elect to create a configuration file manually. In both of these cases, the system will assume that the same chassis key will be used to encrypt the encrypted passwords. If this is not the case, the passwords will not be decrypted due to resulting non-printable characters or memory size checks. This situation is only recoverable by setting the chassis key back to the previous value, editing the configuration to have the encrypted values which match the current chassis key, or by moving the configuration header line lower in the configuration file.

Beginning with Release 15.0, the chassis ID will be generated from an input chassis key using the SHA2-256 algorithm followed by base36 encoding. The resulting 44-character chassis ID will be stored in the same chassisid file in flash.

Release 14 and Release 15 chassis IDs will be in different formats. Release 15 will recognize a Release 14 chassis ID and consider it as valid. Upgrading from 14.x to 15.0 will not require changing the chassis ID or configuration file

However, if the chassis-key is reset in Release 15 through the setup wizard or **chassis-key** CLI command, a new chassis ID will be generated in Release 15 format (44 instead of 16 characters). Release14 builds will not recognize the 44-character chassis ID. If the chassis is subsequently downgraded to Release 14, a new 16-character chassis ID will be generated. To accommodate the old key format, you must save the configuration file in pre-v12.2 format before the downgrade. If you attempt to load a v15 configuration file on the downgraded chassis, StarOS will not be able to decrypt the password/secrets stored in the configuration file.

MIO Synchronization

On boot up both MIO/UMIO/MIO2s automatically read the chassis key configured on the ASR 5500 midplane.

Protection of Passwords

Users with privilege levels of Inspector and Operator cannot display decrypted passwords in the configuration file via the command line interface (CLI).

Secure Password Encryption

By default for StarOS releases prior to 21.0 the system encrypts passwords using an MD5-based cipher (option A). These passwords also have a random 64-bit (8-byte) salt added to the password. The chassis key is used as the encryption key.

Setting a chassis key supports an encryption method where the decryption requires the knowledge of a "shared secret". Only a chassis with knowledge of this shared secret can access the passwords. To decipher passwords, a hacker who knew the chassis key would still need to identify the location of the 64-bit random salt value within the encryption.

Passwords encrypted with MD-5 will have "+A" prefixes in the configuration file to identify the methodology used for encrypting.

Important For release 21.0 and higher, the default is Algorithm B.

For release 15.0 and higher, another type of encryption algorithm can be specified. The Global Configuration mode **cli-encrypt-algorithm** command allows an operator to configure the password/secret encryption algorithm. The default encryption/password algorithm for releases prior to 21.0 is MD-5 as described above (option A). A second password encryption algorithm (option B) uses AES-CTR-128 for encryption and HMAC-SHA1 for authentication. The encryption key protects the confidentiality of passwords, while the authentication key protects their integrity. For release 21.0 and higher Algorithm B is the default. Passwords encrypted with this key will have "+B" prefixes in the configuration file.

For release 19.2 and higher, a third type of encryption algorithm can be specified (option C). This algorithm specifies the use of the HMAC-SHA512 cipher algorithm for encryption and authentication. Passwords encrypted with this key will have "+C" prefixes in the configuration file.

Also for release 19.2 and higher, the encryption key is hashed from the chassis ID and a 16-byte Initialization Vector (IV) obtained from an internal random number generator. No two passwords are encrypted using the same encryption key/IV pair. The Security Administrator must set a chassis key in order to generate the chassis ID and resulting encryption key. A default chassis key based on a local MAC address is no longer supported.

The syntax for the **cli-encrypt-algorithm** command is:

```
config
   cli-encrypt-algorithm { A | B | C }
```

Support for Non-Current Encryptions and Decryptions

The system supports previously formatted encrypted passwords. The syntax of the encrypted passwords indicates which methodology was used for encryption. If the system does not see a prefix before the encrypted password, the earlier encryption method using a fixed key will be used. If the encrypted password includes the "+A" prefix, the decryption method uses the chassis key and random salt.

If the user saves a new configuration, the generated file will always contain passwords encrypted by the most recent method. The user cannot generate the earlier DES-based encryption values. However, all future StarOS releases will continue to support plain-text password entry for all two-way encryptable passwords

The recommended process for changing the chassis key without causing a "lock-out" state is as follows:

- Load the configuration file of the last good configuration using the previous chassis key.
- Change the chassis key to the new desired value.
- Save the configuration with this new chassis key.

Refer to *Configuring a Chassis Key* in *System Settings* for additional information.

Support for ICSR Configurations

Inter-Chassis Session Recovery (ICSR) is a redundancy configuration that employs two identically configured ASR 5500VPC-SI chassis/instances as a redundant pair.

ICSR pairs share the same chassis key. If the ISCR detects that the two chassis/instances have incompatible chassis keys, an error message is logged but the ICSR system will continue to run. Without the matching chassis key, the standby ICSR peer can recover services if the active peer goes out of service; the standby peer will still have access to the passwords in their decrypted form.

ICSR peers use Service Redundancy Protocol (SRP) to periodically check to see if the redundancy configuration matches with either decrypted passwords or DES-based two-way encryption strings. Since the configuration is generated internally to the software, users are not able to access the configuration used to check ICSR compatibility.

Encrypted SNMP Community Strings

Simple Network Management Protocol (SNMP) uses community strings as passwords for network elements. Although these community strings are sent in clear-text in the SNMP PDUs, the values can be encrypted in the configuration file.

The **snmp community encrypted name** command enables the encryption of SNMP community strings. For additional information, see the *Global Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

Enhanced Password Security

Lawful Intercept Restrictions

This section describes some of the security features associated with the provisioning of Lawful Intercept (LI).

LI Server Addresses

An external authenticating agent (such as RADIUS or Diameter) sends a list of LI server addresses as part of access-accept. For any intercept that was already installed or will be installed for that subscriber, a security check is performed to match the LI server address with any of the LI-addresses that were received from the authenticating agent. Only those addresses that pass this criteria will get the intercepted information for that subscriber.

While configuring a campon trigger, the user will not be required to enter the destination LI server addresses. When a matching call for that campon trigger is detected, a security check is done with the list received from the authentication agent. The LI-related information is only forwarded if a matching address is found.

When an active-only intercept is configured, if a matching call is found, a security check is made for the LI address received from the authentication agent and the intercept configuration will be rejected.

If no information related to LI server addresses is received for that subscriber, LI server addresses will not be restricted.

Important A maximum of five LI server addresses are supported via an authenticating agent.

Important The ability to restrict destination addresses for LI content and event delivery using RADIUS attributes is supported only for PDSN and HA gateways.

Modifying Intercepts

One LI administrator can access and/or modify the intercepts created by another LI administrator. Whenever an intercept is added, removed or modified, an event log is displayed across LI administrators about the change. An SNMP trap is also generated.

Adding, Modifying and Removing Users

It is considered uncommon for a user to be added or removed from the system. Likewise, it is considered uncommon for a user's privileges to modified. However, if the system is compromised, it is common for attackers to add or remove a privileged user, raise their privileges or lower the privileges of others.

As a general rule, lower privileged users should not be allowed to increase their privileges or gain access to sensitive data, such as passwords, which were entered by higher privileged users.

Important The system can only detect changes in users and user attributes, such as privilege level, when these users are configured through the system.

Notification of Users Being Added or Deleted

Users with low level authorization should not be able to create users with high level authorization. However, if a malicious actor were to be able to create a high level authorized user, they could then delete the other high level authorized users, thereby locking them out of the system.

The following SNMP traps notify an administrator when users are added or removed:

- **starLocalUserAdded** – indicates that a new local user account has been added to the system.
- **starLocalUserRemoved** – indicates that a local user account has been removed from the system.

Notification of Changes in Privilege Levels

Whenever a user's privilege level is increased or decreased, an SNMP notification will be sent out. A malicious actor may gain access to more privileged commands by somehow promoting" their privileges. Once this is

done, they could then "demote" the privileges of all the other users, thereby locking the proper administrators out of the system.

The **starLocalUserPrivilegeChanged** trap indicates that a local user's privilege level has been changed.

User Access to Operating System Shell

The **starOsShellAccessed** trap indicates that a user has accessed the operating system shell.

Test-Commands

Users with Security Administrator or Administrator privilege can enable the display of previously hidden test-commands. The CLI test-commands mode displays new command keywords for existing commands, as well as new commands.

Caution CLI test-commands are intended for diagnostic use only. Access to these commands is not required during normal system operation. These commands are intended for use by Cisco TAC personnel only. Some of these commands can slow system performance, drop subscribers, and/or render the system inoperable.

Enabling cli test-commands Mode

To enable access to test-commands, a Security Administrator must log into the Global Configuration mode and enter **cli hidden**.

This command sequence is shown below.

```
[local]host_name# config
[local]host_name(config)# cli hidden
[local]host_name(config)#
```

By default **cli hidden** is disabled.

Important Low-level diagnostic and test commands/keywords will now be visible to a user with Administrator or higher privilege. There is no visual indication on the CLI that the test-commands mode has been enabled.

Enabling Password for Access to CLI-test commands

A Security Administrator can set a plain-text or encrypted password for access to CLI test commands. The *password* value is stored in **/flash** along with the boot configuration information. The **show configuration** and **save configuration** commands will never output this value in plain text.

The Global Configuration mode command **tech-support test-commands [encrypted] password** *new_password* **[old-password** *old_password* **]** sets an encrypted or plain-text password for access to CLI test-commands.

This command sequence is shown below.

```
[local]host_name# config
[local]host_name(config)# tech-support test-commands password new_password [
old-password old_password ]
[local]host_name(config)#
```

If the new password replaces an existing password, you must enter the old password for the change to be accepted.

If the old password is not entered or does not match the existing configured value, the following error message appears: "tech-support password is already configured". A prompt then appears to accept entry of the old password: "Enter old tech-support password:".

Entering **old-password** *old_password* allows you to replace the existing password without being prompted to enter the old password. If you incorrectly enter the old password or do not enter the old password, an error message appears: "Failure: Must enter matching old tech-support password to replace existing password".

The Quick Setup Wizard (Exec mode **setup** command) also prompts for entry of a tech-support test-commands password. If you have forgotten the old tech-support password, you can run **setup** directly from the Console port to enter a new tech-support password.

When a test-commands password is configured, the Global Configuration mode command **cli test-commands [encrypted] password** *password* requires the entry of the password keyword. If the **encrypted** keyword is specified, the *password* argument is interpreted as an encrypted string containing the password value. If the **encrypted** keyword is not specified, the *password* argument is interpreted as the actual plain text value

Important If **tech-support test-commands password** is never configured, StarOS will create a new password. If the **password** keyword is not entered for **cli test-commands**, the user is prompted (no-echo) to enter the password. Also, **cli hidden** must be enabled by an administrator to access the CLI test-commands.

Exec Mode cli test-commands

Exec mode commands are available to a privileged user who enters the command **cli test-commands** from Exec mode.

```
[local]host_name# cli test-commands [encrypted] password password
Warning: Test commands enables internal testing and debugging commands
USE OF THIS MODE MAY CAUSE SIGNIFICANT SERVICE INTERRUPTION
```

Important An SNMP trap (starTestModeEntered) is generated whenever a user enters CLI test-commands mode.

Configuration Mode cli test-commands

Configuration commands which provided access to low-level software parameters are accessible only after a privileged user enters the command **cli test-commands** from Global Configuration mode.

```
[local]host_name# config
[local]host_name(config)# cli test-commands [encrypted] password password
Warning: Test commands enables internal testing and debugging commands
USE OF THIS MODE MAY CAUSE SIGNIFICANT SERVICE INTERRUPTION
```

Important An SNMP trap (starTestModeEntered) is generated whenever a user enters CLI test-commands mode.

Using COTS Hardware for Encryption

StarOS VPC instances perform encryption and tunneling of packets in the software. If, however, your commercial off-the shelf (COTS) server uses the Intel Communications Chipset 89xx and you configure the VPC virtual machines to passthrough this chipset, then the VPC instances automatically utilize this hardware chip for encryption and decryption of packets. The Intel Communications Chipset 89xx is also known as Coleto Creek.

Note All service function (SF) VMs must use the Intel Communications chipset in order for the VPC to use the hardware chipset for encryption and decryption.

To determine if your COTS server uses this chipset, use the **show hardware** command to display information for all slots. This example illustrates sample output from the **show hardware** command for a VPC-SI instance on hardware that uses the Coleto Creek crypto accelerator:

```
[local]swch32# show hardware
System Information:
  Platform                : KVM Guest
  UUID/Serial Number      : 014A4D4F-7644-4CF1-C408-8ABB631B3E34
  CPU Packages            : 1 [#0]
  CPU Nodes               : 1
  CPU Cores/Threads       : 16
  Memory                  : 16384M (qvpc-si-medium)
  Crypto Accelerator      : Coleto Creek A0
Storage Devices:
  Virtual Flash           : Present
    Type                  : 4096M disk
    Model                 : ATA-QEMUHARDDISK
    Serial Number         : QM00001
  Hard Drive 1            : Present
    Type                  : 16384M disk
    Model                 : ATA-QEMUHARDDISK
    Serial Number         : QM00002
  Hard Drive 2            : Not Present
  USB 1                   : Not Present
  USB 2                   : Not Present
  CDROM 1                 : Present
    Type                  : cdrom
    Model                 : QEMU-QEMUDVD-ROM
Network Interfaces:
  loeth0   addr 52:54:00:ae:b7:72 at virtio1, 1af4:0001 (virtio_net)
    RxQ(s)/RINGSZ/COALESCE: 1/256/-NA-
    TxQ(s)/RINGSZ/COALESCE: 1/256/-NA-
    NODE-ID               : -NA-
  port1_10 addr 00:1b:21:87:14:ac at 0000:00:06.0, 8086:10fb (ixgbe)
    RxQ(s)/RINGSZ/COALESCE: 16/4096/500
    TxQ(s)/RINGSZ/COALESCE: 16/4096/0
    NODE-ID               : -NA-
  port1_11 addr 00:1b:21:87:14:ad at 0000:00:07.0, 8086:10fb (ixgbe)
    RxQ(s)/RINGSZ/COALESCE: 16/4096/500
    TxQ(s)/RINGSZ/COALESCE: 16/4096/0
```

```
    NODE-ID               : -NA-
```

This example illustrates sample output from the **show hardware** command for a VPC-SI instance on hardware that does not have a crypto accelerator installed:

```
[local]swch81# show hardware
System Information:
  Platform                : KVM Guest
  UUID/Serial Number      : E0A26495-F822-4AC0-914D-B51332177C4D
  CPU Packages            : 1 [#0]
  CPU Nodes               : 1
  CPU Cores/Threads       : 16
  Memory                  : 32768M (qvpc-si-medium)
  Crypto Accelerator      : None
Storage Devices:
  Virtual Flash           : Present
    Type                  : 4096M disk
    Model                 : ATA-QEMUHARDDISK
    Serial Number         : QM00001
  Hard Drive 1            : Present
    Type                  : 16384M disk
    Model                 : ATA-QEMUHARDDISK
    Serial Number         : QM00002
  Hard Drive 2            : Not Present
  USB 1                   : Not Present
  USB 2                   : Not Present
  CDROM 1                 : Present
    Type                  : cdrom
    Model                 : QEMU-QEMUDVD-ROM
Network Interfaces:
  loeth0   addr 52:54:00:e9:70:05 at virtio1, 1af4:0001 (virtio_net)
    RxQ(s)/RINGSZ/COALESCE: 1/256/-NA-
    TxQ(s)/RINGSZ/COALESCE: 1/256/-NA-
    NODE-ID               : -NA-
  port1_10 addr 52:54:00:22:f7:85 at virtio2, 1af4:0001 (virtio_net)
    RxQ(s)/RINGSZ/COALESCE: 8/256/-NA-
    TxQ(s)/RINGSZ/COALESCE: 8/256/-NA-
    NODE-ID               : -NA-
  port1_11 addr 52:54:00:3e:67:f9 at virtio3, 1af4:0001 (virtio_net)
    RxQ(s)/RINGSZ/COALESCE: 8/256/-NA-
    TxQ(s)/RINGSZ/COALESCE: 8/256/-NA-
    NODE-ID               : -NA-
```

Random Number Generator Support for OS and Platforms

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	All
Applicable Platform(s)	• VPC-DI • VPC-SI

Feature Default	Disabled - Configuration required
Related Changes in This Release	Not applicable
Related Documentation	• *VPC-DI System Administration Guide* • *VPC-SI System Administration Guide*

Revision History

Important Revision history details are not provided for features introduced before releases 21.2 and N5.1.

Revision Details	Release
First introduced.	21.13

Feature Description

A few of the features deployed on the ASR 5500 and VPC platforms require random numbers for performing certain tasks. While it uses the kernel random number generator for these tasks, the numbers generated may or may not be sufficiently random as per the security standards. However, hardware or host-provided random numbers are considered reliable and meet security standards.

The Random Number Generator Support for OS and Platforms feature addresses this security compliance requirement. It enables the system administrator to configure hardware random number generator (HWRNG) on their host machines.

When configured, the system uses the the hardware random number generators.

Note This feature works only when HWRNG support is available on the host.

When HWRNG support is available, add the following configuration to the `libvirt xml` file on the host. This adds `virtio_rng` support to the client (StarOS).

```
<rng model='virtio'>
      <backend model='random'>/dev/random</backend>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
</rng>
```

Note If there is a conflict in using slot number 7 (as shown in the preceding configuration) in the configuration, use the next available slot.

This configuration must be applied on the supported platforms based on the respective deployment configurations.

No configuration changes are required on the client. The client (StarOS) picks up `virtio_rng` automatically if the support is enabled on the host.