

NFVIS security considerations

Objective

NFVIS Embedded Security

NFVIS security layers and controls

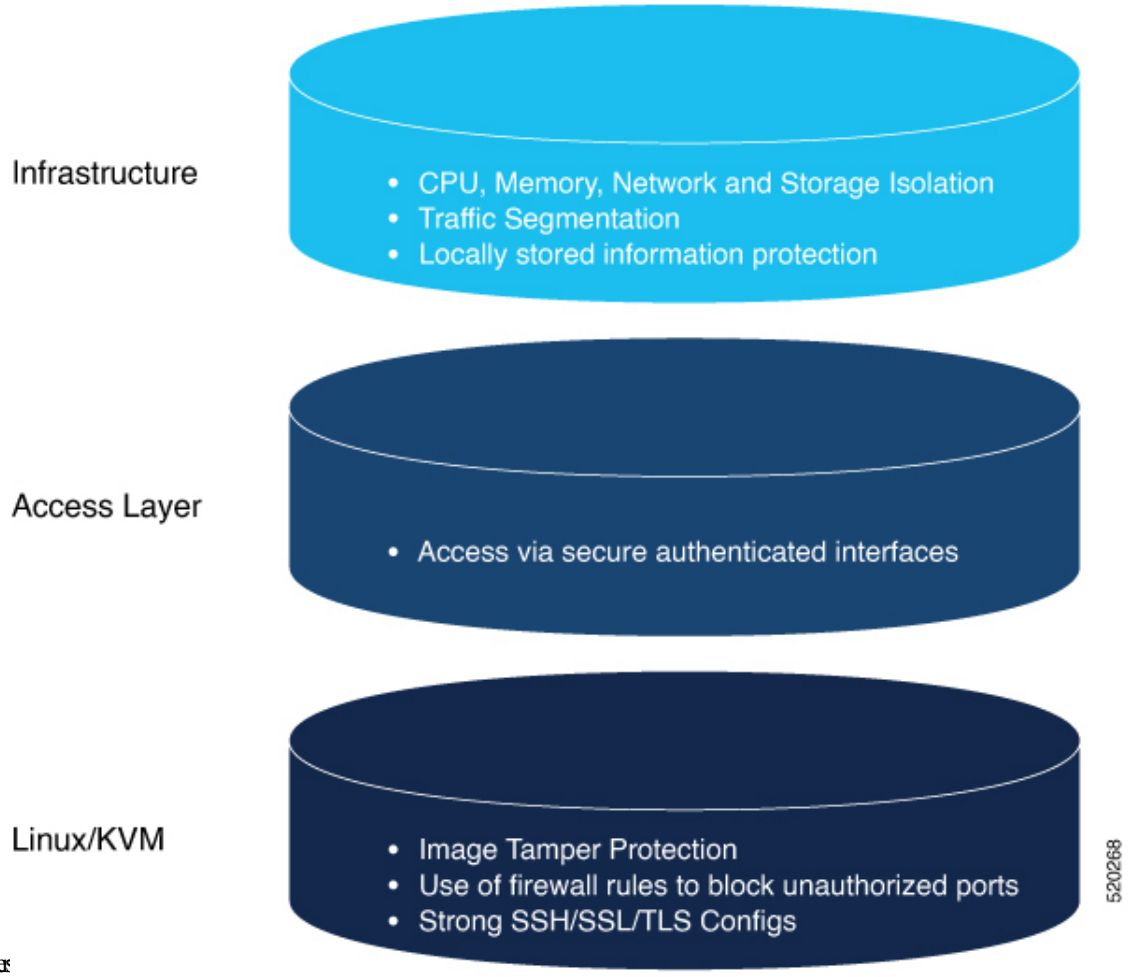
Security considerations are security features and components in NFVIS that

- provide a high-level overview of security related components for planning deployment-specific security strategies
- include recommendations on security best practices for enforcing core network security elements, and
- embed security from installation through all software layers including credential management, integrity and tamper protection, session management, and secure device access.

NFVIS software layers

The NFVIS software has security embedded right from installation through all software layers. The subsequent chapters focus on these out-of-the-box security aspects such as credential management, integrity and tamper protection, session management, secure device access and more.

Figure 1. Software



layer

- Installation, on page 3
- Device access, on page 4
- Secure interfaces, on page 26
- NFVIS portal security, on page 28
- Certificate management, on page 30
- SNMP security features, on page 33
- Banners, on page 35
- Factory default reset, on page 37
- Infrastructure management networks, on page 38
- Protecting sensitive information, on page 40
- Logging, on page 41
- Virtual machine security, on page 42
- Secure development lifecycle, on page 47
- Public sector security certifications, on page 48

Installation

Objective

To ensure that the NFVIS software has not been tampered with, the software image is verified before installation using the following mechanisms.

Image tamper protection

Objective

NFVIS supports RPM signing and signature verification for all RPM packages in the ISO and upgrade images.

RPM signing

RPM signing is a cryptographic security mechanism that

- ensures cryptographic integrity and authenticity of all RPM packages in the Cisco NFVIS ISO and upgrade images
- guarantees that the RPM packages have not been tampered with, and
- verifies that the RPM packages are from NFVIS using private keys created and securely maintained by Cisco.

RPM signature verification

RPM signature verification is a security mechanism that

- verifies the signature of all RPM packages before an installation or upgrade
- aborts the upgrade if signature verification fails, and
- ensures package integrity during NFVIS software operations.

NFVIS behavior during signature verification failure

NFVIS protects software installation and upgrade operations by

- verifying the signature of every RPM package before installation or upgrade, and
- aborting the operation if signature verification fails.

Image integrity verification

Image integrity verification is a security mechanism that

- ensures the integrity of all additional non-RPM files available in the Cisco NFVIS ISO image using published hash values

- provides hash verification for both Cisco NFVIS ISO images and upgrade images, and
- complements RPM signing and signature verification for complete image security.

How hash verification works

Summary

RPM signing and signature verification apply only to RPM packages available in Cisco NFVIS ISO and upgrade images. Cisco publishes a hash with the ISO image and a hash with the upgrade image to verify the integrity of additional non-RPM files.

Workflow

Run **/usr/bin/sha512sum** on the image and compare the result with the hash published by Cisco.

Process details

```
% /usr/bin/sha512sum ImageFile
c2122783efc18b039246ae1bcd4eec4e5e027526967b5b809da5632d462dfa6724a9b20ec318c74548c6bd7e9b8217ce96b5ece93dcdd74fda5e01bb382ad607
  <ImageFile>
```

Device access

Device access is a security framework that

- provides different access mechanisms including console as well as remote access based on protocols such as HTTPS and SSH
- ensures that access is only granted to authenticated users and they can perform just the authorized actions
- restricts device accessibility, user capabilities, and permitted methods of access to prevent unauthorized access to NFVIS.

Device access security controls

Objective

Each access mechanism should be carefully reviewed and configured. Ensure that only the required access mechanisms are enabled and that they are properly secured. The key steps to securing both interactive and management access to NFVIS are to restrict the device accessibility, restrict the capabilities of the permitted users to what is required, and restrict the permitted methods of access. Device access is logged for auditing and NFVIS ensures the confidentiality of locally stored sensitive data.

Device access control recommendations

It is critical to establish the appropriate controls in order to prevent unauthorized access to NFVIS.

Enforced password change at first login

Enforced password change at first login is a security feature that

- prevents security incidents by requiring users to change default credentials after initial login
- forces NFVIS users to update their password when first accessing the system with default credentials (username: admin and password Admin123#), and
- protects systems from attacks that exploit unchanged default login credentials.

Login vulnerability restrictions

You can prevent the vulnerability to dictionary and Denial of Service (DoS) attacks by using the following features.

Enforcement of strong password

Objective

An authentication mechanism is only as strong as its credentials. For this reason, it is important to ensure users have strong passwords. NFVIS checks that a strong password is configured as per these rules.

Password composition requirements

Password must contain:

- At least one uppercase character
- At least one lowercase character
- At least one number
- At least one of these special characters: hash (#), underscore (_), hyphen (-), asterisk (*), or question mark (?)
- Seven characters or more
- The password length should be between 7 and 128 characters.

Configure minimum length for passwords

Configuring a minimum password length strengthens security by making brute-force attacks more difficult. Password length significantly affects the search space for attackers attempting to guess user passwords.

The admin user can set the minimum password length for all users. The value must be from 7 through 128 characters. The default minimum is 7 characters.

Procedure

Set the minimum password length using the CLI or API.

Set the value to an integer from 7 through 128. The following CLI command sets it to 9 characters:

```
nfvis(config)# rbac authentication min-pwd-length 9
```

To configure the value using the API, use this endpoint:

```
/api/config/rbac/authentication/min-pwd-length
```

The minimum password length is configured for all user accounts, enhancing system security by requiring stronger passwords.

What to do next

To set how long users can keep their passwords before changing them, see Configure password lifetime.

Configure password lifetime

The password lifetime determines how long a password can be used before the user is required to change it. The admin user can configure minimum and maximum lifetime values for passwords for all users and enforce a rule to check these values. The default minimum lifetime value is set to 1 day and the default maximum lifetime value is set to 60 days.

When a minimum lifetime value is configured, the user cannot change the password until the specified number of days have passed. Similarly, when a maximum lifetime value is configured, a user must change the password before the specified number of days pass. If a user does not change the password and the specified number of days have passed, a notification is sent to the user.

Note The minimum and maximum lifetime values and the rule to check for these values are not applied to the admin user.

Procedure

Step 1 Configure password lifetime using CLI.

Example:

configure terminal

rbac authentication password-lifetime enforce true min-days *2* **max-days** *30*

commit

Step 2 Configure password lifetime using API.

Example:

```
/api/config/rbac/authentication/password-lifetime/
```

Password lifetime settings are configured with the specified minimum and maximum values, and the enforcement rule is applied to all users except the admin user.

Previous password reuse limitation

Previous password reuse limitation is a security mechanism that

- prevents users from reusing recently used passwords
- makes password expiry meaningful by blocking immediate reversion to old passwords, and
- maintains password history to enforce security policies.

NFVIS password reuse prevention

Objective

NFVIS checks that the new password is not the same as one of the 5 previously used passwords. One exception to this rule is that the admin user can change the password to the default password even if it was one of the 5 previously used passwords.

Login attempt frequency limits

Restrict the frequency of login attempts to prevent brute force attacks and avoid denial of service conditions.

If a remote peer is allowed to login an unlimited number of times, it may eventually be able to guess the login credentials by brute force. Since passphrases are often easy to guess, this is a common attack. By limiting the rate at which the peer can attempt logins, we prevent this attack. We also avoid spending the system resources on unnecessarily authenticating these brute-force login attempts which could create a Denial of Service attack.

NFVIS enforces a 5 minute user lockdown after 10 failed login attempts.

Disable inactive user accounts

Monitoring user activity and disabling unused or stale user accounts helps to secure the system from insider attacks. The unused accounts should eventually be removed.

The admin user can enforce a rule to mark unused user accounts as inactive and configure the number of days after which an unused user account is marked as inactive. Once marked as inactive, that user cannot login to the system. To allow the user to login to the system, the admin user can activate the user account.

Note The inactivity period and the rule to check the inactivity period are not applied to the admin user.

Procedure

Configure account inactivity enforcement using the CLI or API.

The following CLI example enables enforcement and sets the inactivity period to 30 days:

```
configure terminal
rbac authentication account-inactivity enforce true inactivity-days 30
commit
```

To configure account inactivity through the API, use this endpoint:

```
/api/config/rbac/authentication/account-inactivity/
```

User accounts that remain inactive for the specified number of days are automatically marked as inactive and cannot login to the system.

What to do next

To allow an inactive user to log in again, see Activate an inactive user account.

Activate an inactive user account

This task allows the admin user to reactivate a user account that has been deactivated or is in an inactive state.

User accounts may become inactive due to various reasons such as security policies or administrative actions. When a user account needs to be reactivated, the admin user can perform this task using either CLI commands or API calls.

Choose one of the following methods to activate the inactive user account:

Procedure

Activate the user account using the CLI or API.

To activate the example user account `guest_user` using the CLI, enter:

```
configure terminal
rbac authentication users user guest_user activate
commit
```

To activate a user account using the API, use this endpoint:

```
/api/operations/rbac/authentication/users/user/username/activate
```

The previously inactive user account is now activated and the user can access the system with their credentials.

Integration with external AAA servers

Integration with external AAA servers is a security framework that

- authenticates users through password credentials when they login to NFVIS via ssh or the Web UI
- authorizes users to perform specific operations based on their access requirements, and
- supports RADIUS and TACACS protocols to mediate network access through centralized AAA servers.

AAA server deployment recommendations

Objective

It is recommended that a centralized AAA server be deployed to enforce per-user, AAA-based login authentication for NFVIS access. On the AAA server, only minimum access privileges should be granted to authenticated users according to their specific access requirements. This reduces the exposure to both malicious and unintentional security incidents.

Role based access control

Role-based access control (RBAC) is a method of restricting network access that

- limits access based on the roles of individual users within an enterprise
- lets users access just the information they need
- prevents them from accessing information that does not pertain to them, and
- uses an employee's role in the enterprise to determine the permissions granted.

NFVIS user roles and privileges

Objective

Limiting network access is important to organizations that have many employees, employ contractors or permit access to third parties, such as customers and vendors. In such a scenario, it is difficult to monitor network access effectively. Instead, it is better to control what is accessible, in order to secure the sensitive data and critical applications.

NFVIS user roles and privileges

An employee's role in the enterprise should be used to determine the permissions granted, in order to ensure that employees with lower privileges cannot access sensitive information or perform critical tasks.

The user roles and privileges are defined in NFVIS:

User Role	Privilege
Administrators	Can configure all available features and perform all tasks including changing of user roles. The administrator cannot delete basic infrastructure that is fundamental to NFVIS. The Admin user's role cannot be changed; it is always "administrators".
Operators	Can start and stop a VM, and view all information.
Auditors	They are the least privileged users. They have read-only permission and therefore, cannot modify any configuration.

Benefits of using RBAC to restrict unnecessary network access based on people's roles within an organization:

- Improving operational efficiency. Having predefined roles in RBAC makes it easy to include new users with the right privileges or switch roles of existing users. IT also cuts down on the potential for error when user permissions are being assigned.

- Enhancing compliance. Every organization must comply with local, state and federal regulations. Companies generally prefer to implement RBAC systems to meet the regulatory and statutory requirements for confidentiality and privacy because executives and IT departments can more effectively manage how the data is accessed and used. This is particularly important for financial institutions and healthcare companies that manage sensitive data.
- Reducing costs. By not allowing user access to certain processes and applications, companies may conserve or use resources such as network bandwidth, memory and storage in a cost-effective manner.
- Decreasing risk of breaches and data leakage. Implementing RBAC means restricting access to sensitive information, thus reducing the potential for data breaches or data leakage.

Best practices for role-based access control implementations:

- As an administrator, determine the list of users and assign the users to the predefined roles. For example, the user "networkadmin" can be created and added to the user group "administrators".

```
configure terminal
rbac authentication users create-user name networkadmin password Test1_pass
role administrators
commit
```

Note The user groups or roles are created by the system. You cannot create or modify a user group.

To change the password, use the **rbac authentication users user change-password** command in global configuration mode. To change the user role, use the **rbac authentication users user change-role** command in global configuration mode.

- Terminate accounts for users who no longer require access.

```
configure terminal
rbac authentication users delete-user name test1
```

- Periodically conduct audits to evaluate the roles, the employees who are assigned to them and the access that is permitted for each role. If a user is found to have unnecessary access to a certain system, change the user's role.

Roles

Objective

The three roles defined by the system are:

User role permissions

- Administrator: An administrator user has complete access to configure, update or delete a resource.
- Operator: An operator user can only view and operate a resource.
- Auditor: An auditor user can only view a resource and cannot perform any action on it.

Note All three roles have a read-access to all host level configurations, VMs, and images.

Users

A user is an account that

- is created with a role definition that is consistent across all groups
- has read access to NFVIS configurations, filesystems, logs, VMs and images
- can be a member of only one group, and
- may have remote authentication mapping for TACACS and RADIUS based on privilege level.

Admin user characteristics

Objective

The admin user is a special user account with unique properties:

Admin user account properties

- The admin user cannot be deleted or modified.
- The admin user permanently has the administrator role in the default global group.
- The admin user functions as a member of every group and can execute administrative privileges for every group.
- The admin user cannot be assigned to a specific user group.

Groups

A group is a collection of users that

- provides access control to resources based on membership
- defines resource assignment boundaries where a resource can belong to one specific group, and
- ensures all members have access to the resources assigned to that group, with privileges defined by the user's role.

Group types and characteristics

Objective

The global group is a special group assigned to users who are not members of any other group. A user in the global group can access all resources on the system, at the privilege level.

Group types and resource access

Resources can be assigned to the global group or a specific group.

When creating a group, the `resource-access-control` policy should be enabled, to have resource restrictions.

Create groups and assign local users to the groups

Groups organize users and apply policies for role-based access control (RBAC). You can create a group with or without a policy, assign local users to it, and view the users with their roles and group membership.

Procedure

Step 1 Create a group with the resource-access-control policy.

```
aaa groups group rac_group policy resource-access-control
commit
Commit complete.
```

Step 2 Create a local user and assign the user to the group.

```
rbac authentication users create-user name local_admin_3 password Cisco123# role
administrators group rac_group
```

Step 3 View the RBAC users and their role and group information.

```
show running-config rbac authentication users
rbac authentication users user admin role administrators
!
rbac authentication users user local_admin_1 role administrators
!
rbac authentication users user local_admin_2 role administrators
!
rbac authentication users user local_admin_3 role administrators groups group rac_group
!
!
rbac authentication users user local_oper_1 role operators
!
rbac authentication users user local_test_1 role operators
!
```

The group is created and the local user is assigned to it. The running configuration shows the user's role and group.

Assign remote users to the group

Assign remote users to a resource control group so they can manage deployments based on their defined role in the remote server.

NFVIS depends on a remote server for user authentication and authorization. When remote users login to NFVIS, they can only manage the deployment that belongs to its own resource control group, based on their defined role in the remote server. Any remote user that is not mentioned in the resource group, is treated as a global group user and that user can operate the system and manage the deployments based on their defined role.

Procedure

Assign remote users to a resource control group and verify the group membership.

In this example, add three TACACS+ or RADIUS users to the `tac_group` resource control group:

```
aaa groups group tac_group user remote_admin1
user remote_admin2
user remote_operator3
policy resource-access-control
commit
end
```

Verify the group configuration:

```
show running-config aaa groups group tac_group
aaa groups group tac_group
policy resource-access-control
user remote_admin1
user remote_admin2
user remote_operator3
!
!
```

Verify the users:

```
nfvis# show rbac authentication users
NAME
-------
admin
```



Note In this example, `remote_admin1`, `remote_admin2`, and `remote_operator3` are TACACS+ or RADIUS users.

The remote users are assigned to the resource control group and can manage deployments according to their defined roles.

Configure local authentication for a specific group of users

This task enables you to create a group and add specific users to it, allowing these users to bypass the default authentication order and only go through local authentication instead of the default TACACS then local authentication sequence.

Typically, users who need authentication go through the default authentication order. The default order involves external authentication through TACACS as the first step, and local authentication as the second step. The local authentication for a specific group of users feature enables you to create a group and add specific users to it, allowing these users to bypass the default authentication order. The users in this group skip the external TACACS authentication and only go through local authentication. For this group to function as expected, assign the 'local-authentication-only' policy to the group.

Restrictions for local authentication for a specific group of users:

- One user can be assigned to a maximum of one group.
- If a local user assigned to the local authentication group has the same user name as a remote (TACACS+/RADIUS) user, then only the local user's credentials are taken into consideration. The remote user's credentials are considered even if the local user's authentication fails.

Procedure

Step 1 Create a group with the local-authentication-only policy.

Example:

```
nfvis#(config) aaa groups group [ group-name ] policy local-authentication-only
```

Step 2 Assign a user to the group using one of these methods:

Assign an existing user to the group:

```
nfvis# rbac authentication users user username assign-group [ group-name ]
```

Create a new user and assign to the group simultaneously:

```
nfvis# rbac authentication users create-user name username password password role role
group [ group-name ]
```

To remove a user from a group, use:

```
nfvis# rbac authentication users user username remove-group [ group-name ]
```

The group is created with the local-authentication-only policy, and the specified users are assigned to the group. These users bypass external TACACS authentication and use local authentication.

RADIUS

Remote Authentication Dial-In User Service (RADIUS) is a distributed client-server system that

- secures networks against unauthorized access
- uses Cisco routers as RADIUS clients that send authentication requests to a central RADIUS server containing all user authentication and network service access information, and
- operates as a fully open protocol distributed in source code format that can be modified to work with any security system currently available on the market.

RADIUS implementation details

Objective

Cisco supports RADIUS under its AAA security paradigm. RADIUS has been implemented in a variety of network environments that require high levels of security while maintaining network access for remote users.

RADIUS server configuration and failover

Note You can configure upto four RADIUS servers. When multiple RADIUS servers are configured, if the first server is unreachable, NFVIS tries the next server in the order it is configured.

How RADIUS authentication works

Summary

Summary

Workflow

The key components involved in RADIUS authentication are:

Process details

- User: Provides credentials and receives authentication responses
- Access server: Prompts for credentials and relays authentication requests
- RADIUS server: Validates credentials and provides authorization data

Workflow

These stages describe how RADIUS authentication works:

1. The user is prompted to enter the username and PASSWORD.
2. The username and encrypted PASSWORD are sent over the network to the RADIUS server.
3. The user receives one of these responses from the RADIUS server:
   - ACCEPT—The user is authenticated.
   - CHALLENGE—A CHALLENGE is issued by the RADIUS server. The CHALLENGE collects additional data from the user.
   - CHANGE PASSWORD—A request is issued by the RADIUS server, asking the user to select a new PASSWORD.
   - REJECT—The user is not authenticated and is prompted to reenter the username and PASSWORD, or access is denied.

Result

The ACCEPT or REJECT response is bundled with additional data that is used for EXEC or network authorization. You must first complete RADIUS authentication before using RADIUS authorization. The additional data included with the ACCEPT or REJECT packets consists of services that the user can access, including connections such as Telnet, rlogin, or local-area transport (LAT), and services such as PPP, Serial Line Protocol (SLIP), or EXEC services, and connection parameters, including the host or client IP address, access list, and user timeouts.

Configure RADIUS

NFVIS supports RADIUS shared-secret encryption. Configure either a shared secret or an encrypted shared secret at a time. Use an encrypted secret when the secret contains special characters. NFVIS encrypts both shared-secret and encrypted-shared-secret configurations and supports TLS and non-TLS RADIUS communication.

The secret must be 1 through 127 characters and contain only characters from this set: `[-_a-zA-Z0-9 ./\<>%!*$€#{}()+@]*.`

Procedure

Step 1 Configure RADIUS support using one of the following methods:

For non-TLS RADIUS communication, enter:

```
configure terminal
radius-server host 1.2.3.4 shared-secret abc admin-priv 15 oper-priv 11
commit
```

For TLS RADIUS communication, enter:

```
configure terminal
radius-server host 1.2.3.4 shared-secret efkbkuabcwuabvauvvwqbd use-tls true pskidentity
 identity admin-priv 15 oper-priv 11
commit
```

Step 2 Verify the RADIUS configuration.

Use the following command:

```
show running-config radius-server
```

The shared secret is displayed in encrypted form.

```
radius-server host 1.2.3.4
shared-secret $8$lZwgccBvs9x1oQpx6f1s8/JkZ4rLdQ95VMUjRrhD9Z8=
admin-priv    15
oper-priv     11
```

RADIUS authentication is configured with the specified server, secret, and privilege levels.

TACACS+

TACACS+ is a security application that

- provides centralized validation of users attempting to gain access to a router or network access server
- maintains services in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation, and
- requires TACACS+ server configuration before the configured TACACS+ features on your network access server are available.

TACACS+ server configuration requirements

Objective

On the TACACS+ server, ensure you configure Cisco attribute-value (AV) pair privilege level (priv-lvl) for Cisco NFVIS as a service for the minimum privilege level of administrators and operators.

TACACS+ server availability and limits

Note You can configure upto four TACACS+ servers. When multiple TACACS+ servers are configured, if the first server is unreachable, NFVIS tries the next server in the order it is configured.

How TACACS+ authentication works

Summary

Summary

Workflow

The key components involved in TACACS operation are:

Process details

- User: Attempts to log in to NFVIS using credentials
- NFVIS: Sends user credentials to TACACS+ server and processes responses
- TACACS+ server: Authenticates users and provides authorization responses

Workflow

The TACACS operation involves these stages:

1. When the user tries to log in, NFVIS sends user credential to TACACS+ server.
2. NFVIS will eventually receive one of the following responses from the TACACS+ server:
   - ACCEPT—The user is authenticated and service can begin. If NFVIS is configured to require authorization, authorization begins at this time.
   - REJECT—The user is not authenticated. The user can be denied access or is prompted to retry the login sequence, depending on the TACACS+ server.
   - ERROR—An ERROR occurred at some time during authentication with the server or in the network connection between the server and NFVIS. If an ERROR response is received, NFVIS typically tries to use an alternative method for authenticating the user.
   - CONTINUE—The user is prompted for additional authentication information.

   After authentication, NFVIS will send authorization request to TACACS+ server.
3. Based on authorization result, NFVIS will assign user's role.

Configure a TACACS+ server

Configure TACACS+ server authentication to enable centralized user authentication and authorization with customizable privilege levels for different user roles.

TACACS+ secret encryption is supported. You can only configure either secret key or encrypted secret key at a given time. Encrypted secret key can contain special characters but secret key cannot. The following pattern is supported for encrypted-shared-key: [-_a-zA-Z0-9./\\<>%!*$€#{}()+].

NFVIS encrypts both shared-secret and encrypted-shared-secret configurations.

NFVIS supports both TLS and non-TLS client-server communication for TACACS+.

The following constraints apply:

- Secret length must be between 1 and 127 characters.
- Secret must only contain characters from the set: [-_a-zA-Z0-9 ./\<>%!*$€#{}()+@]*

Procedure

Step 1 Configure a TACACS+ server without TLS.

```
configure terminal
tacacs-server host 1.2.3.4 shared-secret asdfghh admin-priv 14 oper-priv 9
commit
```

In this configuration, privilege level 14 is assigned to the administrator role, and privilege level 9 is assigned to the operator role. This means a user with privilge level 14 or higher will have all admin privileges when the user logs into the system, and a user with privilege level 9 or higher will have all privileges of an operator at the time of login.

Step 2 Configure a TACACS+ server with TLS.

```
configure terminal
tacacs-server host 1.2.3.4 shared-secret mfqwudvkdwnbkyuDLndkw use-tls true pskidentity
 identity admin-priv 14 oper-priv 9
commit
```

Step 3 Verify the TACACS+ configuration.

Use the following command:

```
show running-config tacacs-server
```

The shared secret is displayed in encrypted form.

```
tacacs-server host 1.2.3.4
shared-secret $8$JkMZFgA3DkbjAHOrmdBr3U2cLg2qY1FuHAIJiIp7nSw=
admin-priv    14
oper-priv     9
```

The TACACS+ server is configured with the specified host, secret, and privilege levels. Users can authenticate through the server with the corresponding role-based privileges.

Default authentication order

Default authentication order is a security mechanism that

- supports both TACACS+ and RADIUS but allows only one authentication method to be enabled at a time
- requires method lists to be defined for TACACS+ and RADIUS authentication through AAA commands, and
- uses local authentication as fallback when TACACS+ or RADIUS is not accessible.

Authentication configuration

Objective

After you have identified the TACACS+ and RADIUS server and defined an associated TACACS+ and RADIUS authentication key, you must define method lists for TACACS+ and RADIUS authentication. Because TACACS+ and RADIUS authentication is operated through AAA, you need to issue the **AAA authentication** command, specifying TACACS+ or RADIUS as the authentication method.

AAA authentication methods and behavior

```
nfvis(config)# aaa authentication ?
Possible completions:
  radius   Use RADIUS for AAA
  tacacs   Use TACACS+ for AAA
  users    List of local users
```



Note

- Only when TACACS+ or RADIUS is enabled, it can be used for authentication.
- When TACACS+ or RADIUS is not accessible, local authentication is used. It is recommended to use **AAA authentication** TACACS local command to authenticate using local database. Local authentication is disabled if the connection between TACACS+ or RADIUS and NFVIS is restored.
- If the same username is registered for both local authentication and authentication through RADIUS or TACACS+, RADIUS or TACACS+ is chosen as the authentication method.
- It is recommended to configure Syslog so that it is easier to debug if TACACS+ or RADIUS does not work as expected.

All login attempts will be logged in syslogs in the local nfvis_syslog.log, NFVIS-ext-auth.log files and in remote syslog servers.

User specific authentication order

Objective

The system follows this authentication sequence for user-specific authentication:

User-specific authentication sequence

- If the user is part of the local database, local authentication is executed and the user is permitted or denied access.
- If the user is not part of the local database, TACACS+ is used for authentication.
- If the same user is part of both the databases (local and TACACS+), the user can login with either the local password or the TACACS+ password. However, registering the same user in both the databases is not recommended.

Granular role-Based access control

Objective

This feature adds a new resource group policy that manages the VM and VNF and allows you to assign users to a group to control VNF access, during VNF deployment. For more information, see below

Resource group operations and role permissions

Operations/Roles	"admin" Super User	Administrator	Operator	Auditor	VNF Group Administrator	VNF Group Operator	VNF Group Auditor
User Management							
Create, update, delete new logins and assign roles	Y	Y	N	N	N	N	N
List all logins	Y	Y	Y	Y	Y	Y	Y
Ability to reset password on any login	Y	Y	N	N	Y	N	N
Create and delete custom VNF groups and assign users to VNF groups	Y	Y	N	N	N	N	N
VNF Management							
Create	Y	Y	N	N	Y	N	N
Delete or update VNFs	Y	Y	N	N	Specific VNFs	N	N
Update VNF group name	Y	Y	N	N	N	N	N
Start and stop VNFs	Y	Y	Y	N	Specific VNFs	Specific VNFs	N
VNF snapshots	Y	Y	Y	N	Specific VNFs	Specific VNFs	N

Operations/Roles	"admin" Super User	Administrator	Operator	Auditor	VNF Group Administrator	VNF Group Operator	VNF Group Auditor
Restore from snapshot	Y	Y	N	N	Specific VNFs	N	N
List all VNF instances	Y	Y	Y	Y	Specific VNFs	Specific VNFs	Specific VNFs
Access the VNF console	Y	Y	Y	N	Specific VNFs	Specific VNFs	N
Assign VNF to a custom group	Y	Y	N	N	N	N	N
Device Level Operations							
View and monitor device level statistics, including logs	Y	Y	Y	Y	Y	Y	Y
Factory reset	Y	Y	N	N	N	N	N
System-wide device configurations such as DNS	Y	Y	N	N	N	N	N
Device backup and restore	Y	N	N	N	N	N	N
Platform upgrades	Y	Y	N	N	N	N	N
Change BIOS password, change CIMC password, PNIC-breakout, power cycle, host reboot, and hostaction shutdown	Y	Y	N	N	N	N	N
Clear BIOS password, SSH keys, and stats files	Y	Y	N	N	N	N	N
Clear cache, mgmt-dhcp-renew, wan-dhcp-renew, and PnP actions	Y	Y	Y	N	N	N	N
System certificate modifications, ISCSI settings, and file delete operation	Y	Y	N	N	N	N	N
Tech support and logs	Y	Y	Y	Only through portal	Y	Y	Only through portal

Operations/Roles	"admin" Super User	Administrator	Operator	Auditor	VNF Group Administrator	VNF Group Operator	VNF Group Auditor
Create VFs for SR-IOV, and enable DPDK	Y	Y	N	N	N	N	N
Image Repository Changes							
Upload new VNF images	Y	Y	N	N	Y	N	N
Upload new NFVIS upgrade files	Y	Y	N	N	Y	N	N
File transfer to the device (certificates)	Y	Y	N	N	Y	N	N
File transfer from the device	Y	Y	N	N	Y	N	N
File-copy and file-download	Y	Y	N	N	Y	N	N
File-delete	Y	Y	N	N	N	N	N
View contents	Y	Y	Y	Y	Y	Y	Y
Clustering Operations							
Set up a cluster (both management and storage)	Y	Y	N	N	N	N	N

Restrict device accessibility

Objective

Disable unused services to prevent security vulnerabilities from default configurations and passwords that attackers can exploit to gain unauthorized access to system information.

Restricting device access

Users have repeatedly been caught unawares by attacks against features they had not protected because they did not know that those features were enabled. Unused services tend to be left with default configurations which are not always secure. These services may also be using default passwords. Some services can give an attacker easy access to information on what the server is running or how the network is setup.

Essential ports are enabled by default

Only those services which are absolutely necessary to setup and manage NFVIS are available by default. This removes the user effort needed to configure firewalls and deny access to unnecessary services.

Open Port	Service	Description
22/TCP	SSH	Secure Socket Shell for remote command-line access to NFVIS
80/TCP	HTTP	Hypertext Transfer Protocol for the NFVIS portal access. All HTTP traffic received by NFVIS is redirected to port 443 for HTTPS
443/TCP	HTTPS	Hypertext Transfer Protocol Secure for secure NFVIS portal access
830/TCP	NETCONF-SSH	Port opened for the Network Configuration Protocol (NETCONF) over SSH. NETCONF is a protocol used for automated configuration of NFVIS and for receiving asynchronous event notifications from NFVIS.
161/UDP	SNMP	Simple Network Management Protocol (SNMP). Used by NFVIS to communicate with remote network-monitoring applications. For more information see, SNMP.

Authorized network access restrictions

Authorized network access restriction is a security mechanism that

- permits only authorized originators to attempt device management access
- limits access to services they are authorized to use, and
- reduces the risk of unauthorized access and exposure to attacks such as brute force, dictionary, or DoS attacks.

NFVIS access control lists

Objective

NFVIS can be configured such that access is restricted to known, trusted sources and expected management traffic profiles. To protect the NFVIS management interfaces from unnecessary and potentially harmful traffic, an admin user can create Access Control Lists (ACLs) for the network traffic that is received. These ACLs specify the source IP addresses/networks from which the traffic originates, and the type of traffic that is permitted or rejected from these sources. These IP traffic filters are applied to each management interface on NFVIS.

IP receive ACL parameters and examples

These parameters are configured in an IP receive Access Control List (IP-receive-ACL):

Parameter	Value	Description
Source network/Netmask	Network/netmask. For example: 0.0.0.0/0 172.39.162.0/24	This field specifies the IP address/network from which the traffic originates
Service	HTTPS ICMP NETCONF scpd SNMP SSH	Type of traffic from the specified source.
Action	accept drop reject	Action to be taken on the traffic from the source network. With accept, new connection attempts will be granted. With reject, connection attempts will not be accepted. If the rule is for a TCP based service such as HTTPS, NETCONF, SCP, SSH, the source will get a TCP reset (RST) packet. For non-TCP rules such as SNMP and ICMP, the packet will be dropped. With drop, all packets will be dropped immediately, there is no information sent to the source.
Priority	A numeric value	The priority is used to enforce an order on the rules. Rules with a higher numeric value for priority will be added further down in the chain. If you want to make sure that a rule will be added after another one, use a low priority number for the first and a higher priority number for the next.

The sample configurations illustrate some scenarios that can be adapted for specific use-cases.

Configuring the IP Receive ACL

The more restrictive an ACL, the more limited the exposure to unauthorized access attempts. However, a more restrictive ACL can create a management overhead, and can impact accessibility to perform troubleshooting. Consequently, there is a balance to be considered. One compromise is to restrict access to internal corporate IP addresses only. Each customer must evaluate the implementation of ACLs in relation to their own security policy, risks, exposure, and acceptance thereof.

Reject SSH traffic from a subnet:

```
nfvis(config)# system settings ip-receive-acl 171.70.63.0/24 service ssh action
 reject priority 1
```

Removing ACLs:

When an entry is deleted from IP-receive-ACL, all configurations to that source are deleted since the source IP address is the key. To delete just one service, configure other services again.

```
nfvis(config)# no system settings ip-receive-acl 171.70.63.0/24
```

For more details see, Configure the IP receive ACL

Access privileged debug shell

The super-user account on NFVIS is disabled by default to prevent unrestricted, potentially adverse system-wide changes, and NFVIS does not expose the system shell to users. However, for difficult-to-debug issues on NFVIS, the Cisco Technical Assistance Center (TAC) or development team might require shell access to the customer's NFVIS. NFVIS uses a secure unlock infrastructure to restrict privileged debug access on field devices to authorized Cisco employees. A challenge-response authentication mechanism between NFVIS and the Cisco-maintained Interactive Debug server provides access to the Linux shell for this interactive debugging. The admin user's password is also required, in addition to the challenge-response entry, to confirm the customer's consent.

Follow these steps to access the shell for interactive debugging:

Procedure

Step 1 Initiate the shell access procedure using the hidden command.

```
nfvis# system shell-access
```

Step 2 Copy the challenge string displayed on the screen.

```
Challenge String (Copy everything between the asterisk lines exclusively):
********************************************************************************
SPH//wkAAABORlZJU0VOQ1M1NDA4L0s5AQAAABt+dcx+hB0V06r9RkdMMjEzNTgw
RlHq7BxeAAA=
DONE.
********************************************************************************
```

Step 3 Provide the challenge string to the Cisco member for verification.

The Cisco member enters the Challenge string on an Interactive Debug server maintained by Cisco. This server verifies that the Cisco user is authorized to debug NFVIS using the shell and then returns a response string.

Step 4 Enter the response string when prompted.

Input your response when ready:

Step 5 Enter the admin password when prompted.

Step 6 Access the shell if the password is valid.

The development or TAC team uses the shell to proceed with debugging.

Step 7 Type **Exit** to exit shell access when debugging is complete.

You have successfully accessed the privileged debug shell and can perform interactive debugging with Cisco support team assistance.

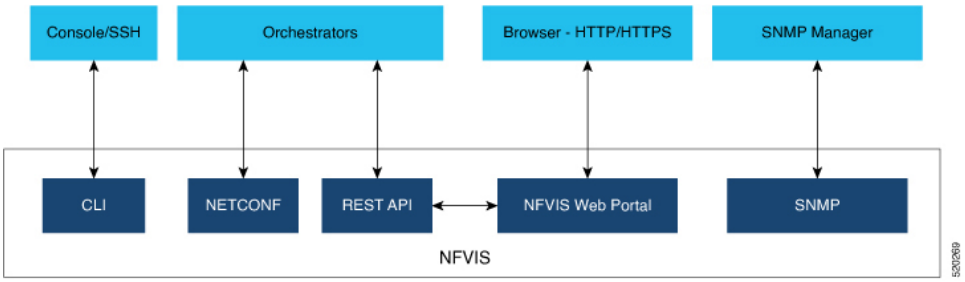
Secure interfaces

Objective

Describes the interfaces available for NFVIS management access and outlines security best practices to implement for each interface to ensure secure system administration.

NFVIS management interfaces

NFVIS management access is allowed using the interfaces shown in the diagram. The following sections describe security best practices for these interfaces to NFVIS.



Console

A console port is an asynchronous serial port that

- allows you to connect to the NFVIS CLI for initial configuration
- can be accessed with either physical access to the NFVIS or remote access through the use of a terminal server, and
- requires access lists on the terminal server to allow access only from the required source addresses when console port access is required via a terminal server.

SSH

SSH is a secure network protocol that

- provides users with secure remote access to the NFVIS CLI
- ensures the integrity and confidentiality of NFVIS management traffic, and
- uses version 2 with strong encryption, hash, and key exchange algorithms recommended by the Security and Trust Organization within Cisco.

NFVIS uses SSH version 2, which is Cisco's and the Internet's de facto standard protocol for interactive logins. The integrity and confidentiality of NFVIS management traffic is essential to the security of the administered network since administration protocols frequently carry information which could be used to penetrate or disrupt the network.

CLI session timeout

A CLI session timeout is a security mechanism that

- automatically logs out users from CLI sessions after 15 minutes of inactivity
- limits the risk of internal attacks, such as one user trying to use another user's session, and
- reduces network security risks when users leave logged-in sessions unattended.

Session security mechanism

Objective

By logging in via SSH, a user establishes a session with NFVIS. While the user is logged in, if the user leaves the logged-in session unattended, this can expose the network to a security risk. Session security limits the risk of internal attacks, such as one user trying to use another user's session.

CLI session timeout behavior

To mitigate this risk, NFVIS times out CLI sessions after 15 minutes of inactivity. When the session timeout is reached, the user is automatically logged out.

NETConf

The Network Configuration Protocol (NETCONF) is a network management protocol that

- is developed and standardized by the IETF for the automated configuration of network devices
- uses an Extensible Markup Language (XML) based data encoding for the configuration data as well as the protocol messages, and
- exchanges protocol messages on top of a secure transport protocol.

NETCONF allows NFVIS to expose an XML-based API that the network operator can use to set and get configuration data and event notifications securely over SSH.

For more information see, NETCONF event notifications.

REST API

A REST API is a configuration interface that

- allows requesting systems to access and manipulate NFVIS configuration using a uniform and predefined set of stateless operations
- operates over HTTPS for secure communication, and

- limits concurrent sessions to 100 to prevent denial of service attacks.

When the user issues a REST API, a session is established with NFVIS. In order to limit risks related to denial of service attacks, NFVIS limits the total number of concurrent REST sessions to 100.

NFVIS portal security

The NFVIS portal is a web-based graphical user interface that

- displays information about NFVIS
- presents users with an easy means to configure and monitor NFVIS over HTTPS, and
- eliminates the need to know the NFVIS CLI and API.

Session management

Objective

The stateless nature of HTTP and HTTPS requires a method of uniquely tracking users through the use of unique session IDs and cookies.

Session encryption, auditing, and timeout

NFVIS encrypts the user's session. The AES-256-CBC cipher is used to encrypt the session contents with an HMAC-SHA-256 authentication tag. A random 128-bit Initialization Vector is generated for each encryption operation.

An Audit record is started when a portal session is created. Session information is deleted when the user logs out or when the session times out.

The default idle timeout for portal sessions is 15 minutes. However, this can be configured for the current session to a value between 5 and 60 minutes on the **Settings** page. Auto-logout will be initiated after this period. Multiple sessions are not permitted in a single browser. The Maximum number of concurrent sessions are set to 30.

The NFVIS portal utilizes cookies to associate data with the user. It uses these cookie properties for enhanced security:

- ephemeral to ensure the cookie expires when the browser is closed
- httpOnly to make the cookie inaccessible from JavaScript
- secureProxy to ensure the cookie can only be sent over SSL.

Even after authentication, attacks such as Cross-Site Request Forgery (CSRF) are possible. In this scenario, an end user might inadvertently execute unwanted actions on a web application in which they are currently authenticated. To prevent this, NFVIS uses CSRF tokens to validate every REST API that is invoked during each session.

URL redirection

Objective

In typical web servers, when a page is not found on the web server, the user gets a 404 message; for pages that exist, they get a login page. The security impact of this is that an attacker can perform a brute force scan and easily detect which pages and folders exist.

HTTP redirection and security headers

To prevent this on NFVIS, all non-existent URLs prefixed with the device IP are redirected to the portal login page with a 301 status response code. This means that irrespective of the URL requested by an attacker, they will always get the login page to authenticate themselves.

All HTTP server requests are redirected to HTTPS and have these headers configured:

- X-Content-Type-Options
- X-XSS-Protection
- Content-Security-Policy
- X-Frame-Options
- Strict-Transport-Security
- Cache-Control

Portal disablement

Objective

The NFVIS portal access is enabled by default. If you are not planning to use the portal, it is recommended to disable portal access using this command:

Disabling portal access

```
Configure terminal
System portal access disabled
commit
```

HTTPS

Objective

All HTTPS data to and from NFVIS uses Transport Layer Security (TLS) to communicate across the network. TLS is the successor to Secure Socket Layer (SSL). The TLS handshake involves authentication during which the client verifies the server's SSL certificate with the certificate authority that issued it. This confirms that the server is who it says it is, and that the client is interacting with the owner of the domain.

HTTPS certificates and TLS security

By default, NFVIS uses a self-signed certificate to prove its identity to its clients. This certificate has a 2048-bit public key to increase the security of the TLS encryption, since the encryption strength is directly related to the key size. NFVIS generates a self-signed SSL certificate when first installed. It is a security best practice to replace this certificate with a valid certificate signed by a compliant Certificate Authority (CA).

Certificate management

Objective

NFVIS uses a server certificate for HTTPS management access for Portal. The default certificate is self-signed; administrators can replace it with a CA-signed certificate using the NFVIS CSR and certificate-install actions. The built-in CSR action accepts subject fields but has no SAN input. NFVIS also has no supported action to import an externally generated private key. The supported flow is to keep the NFVIS-generated key on the appliance and ask the CA to add SANs to the issued certificate, if the CA's policy allows it.

Certificate paths and consumers

Objective

At startup, NFVIS generates a self-signed RSA certificate and key under /etc/pki/self-signed/, then installs copies as /etc/pki/nginx/server.crt and /etc/pki/nginx/server.key. Nginx uses those active files for HTTPS. The VNC console's WebSocket proxy also uses the active Nginx certificate and key.

Certificate paths and consumers

The self-signed certificate is regenerated when the NFVIS hostname changes. The generator sets the common name to that hostname and does not add SAN extensions. Modern browsers check SANs for hostname identity, so the default certificate may show a name error as well as an untrusted-issuer warning.

The **system certificate use-cert** action switches the active certificate between the self-signed pair and the CA-signed pair. Switching to self-signed copies files from /etc/pki/self-signed/. Switching to CA-signed uses /

etc/pki/third-party/server.key and /etc/pki/third-party/server.crt.

Generate and retrieve an NFVIS CSR

The certificate signing request (CSR) command accepts common name, country code, locality, organization, organization unit name, and state fields. The common name is required.

Procedure

Step 1 Generate the CSR on NFVIS.

Replace `NFVIS-HOSTNAME` with the DNS name clients use to reach NFVIS.

```
nfvis# system certificate signing-request common-name NFVIS-HOSTNAME country-code US
organization nfvis-ap
```

NFVIS saves the CSR to `/data/intdatastore/download/nfvis.csr` and generates the matching private key at `/etc/pki/third-party/server.key`. The private key remains on NFVIS and is not included in the CSR.

Step 2 Retrieve only the CSR from a workstation using SCP on port 22222.

Use the `-O` option with OpenSSH clients that fail to work with the NFVIS transfer endpoint because they use SFTP by default:

```
scp -O -P 22222 admin@NFVIS-IP:/data/intdatastore/download/nfvis.csr .
```

The CSR is generated on NFVIS and copied to the workstation. The matching private key remains on NFVIS.

What to do next

To request subject alternative names for the issued certificate, see Add SANs when issuing the certificate.

Private-key support and external CSR limitation

Objective

NFVIS does not provide a supported CLI or API action to import a private key generated elsewhere. The install-cert action installs a certificate file only; it does not import a key. The use-cert cert-type ca-signed action expects the matching key to already exist at /etc/pki/third-party/server.key.

External private key and CSR limitations

The separate system certificate input action is for a PnP root certificate. It does not import a server private key. Manually copying an external key into NFVIS is outside the supported workflow. In the current implementation, activation checks that the key and certificate files exist and that the certificate parses, but it does not verify that the key matches the certificate. Compare their public-key digests before activation.

If the customer must use an externally generated key and CSR, the current NFVIS workflow does not support that request. It would require a product enhancement to securely import the key and generate a SAN-bearing CSR from it.

Add SANs when issuing the certificate

The NFVIS CSR action does not accept subject alternative name (SAN) fields and generates a subject-only request. If SANs must be included in the CSR itself, the current NFVIS command does not support that.

When you submit the CSR to your organization's CA, specify the required DNS names and IP addresses in the CA enrollment form or certificate profile. The CA may issue a certificate with SANs that are not encoded in the CSR if its policy allows it. The issued certificate must retain the public key from the NFVIS CSR so that it matches the private key stored on NFVIS.

Procedure

Step 1 For a local test CA, define the certificate extensions and SAN values.

Replace `NFVIS-HOSTNAME` and `NFVIS-IP` with the hostname and management IP address clients use to reach NFVIS:

```
[server_cert]
basicConstraints = critical,CA:FALSE
keyUsage = critical,digitalSignature,keyEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer

[alt_names]
DNS.1 = NFVIS-HOSTNAME
IP.1 = NFVIS-IP
```

Step 2 Sign the NFVIS CSR with the test CA certificate and private key.

This local CA procedure is for testing. Keep the CA private key off NFVIS. For production, use your organization's approved CA process and profile.

openssl x509 -req -in *nfvis.csr* **-CA** *test-ca.crt* **-CAkey** *test-ca.key* **-CAcreateserial -out** *nfvis-test.crt* **-days** *30* **-sha256 -extfile** *nfvis-san.conf* **-extensions** *server_cert*

The CA can issue a certificate with the requested SANs if its policy allows, and the certificate public key matches the private key generated on NFVIS.

Verify and install the certificate

Objective

Check the issued certificate's subject, issuer, validity period, and SANs:

Certificate verification commands

openssl x509 -in nfvis-test.crt -noout -subject -issuer -dates -ext subjectAltName

To verify the certificate matches the CSR's public key, compare the two SHA-256 digests. They must match:

```
openssl req -in nfvis.csr -pubkey -noout |
openssl pkey -pubin -outform DER | openssl dgst -sha256
openssl x509 -in nfvis-test.crt -pubkey -noout |
openssl pkey -pubin -outform DER | openssl dgst -sha256
```

Upload the issued certificate to NFVIS:

scp -O -P 22222 nfvis-test.crt admin@@@VAR0@@:/data/intdatastore/uploads/

Install and activate it from the NFVIS CLI:

```
nfvis# system certificate install-cert path
file:///data/intdatastore/uploads/nfvis-test.crt
nfvis(config)# system certificate use-cert cert-type ca-signed
```

The install action accepts a certificate path. The CA-signed activation action uses the private key already at /etc/pki/third-party/server.key. Confirm the certificate matches that key before activation; the current action code does not compare them.

HTTPS and browser trust validation

Objective

For a locally issued test certificate, verify the server using the local test CA certificate.

Validate HTTPS and browser trust

```
openssl s_client \
-connect NFVIS-IP:443 \
-servername NFVIS-HOSTNAME \
-CAfile test-ca.crt \
-verify_hostname NFVIS-HOSTNAME \
-brief </dev/null
```

If OpenSSL reports successful verification, the presented chain and hostname validate against `test-ca.crt`. A browser may still show a warning until that test CA certificate is added to the workstation's trusted root store. The browser URL must also match a SAN: use a DNS name listed in the certificate or an IP address listed as an IP SAN.

For production, use a CA already trusted by clients, or distribute the organization's private CA root through its managed trust process. Importing a test CA makes certificates issued by that CA trusted by the workstation, so remove the test CA trust after testing.

SNMP security features

Objective

Three significant versions of SNMP have been developed. NFVIS supports SNMP version 1, version 2c and version 3. SNMP versions 1 and 2 use community strings for authentication, and these are sent in plain-text. So, it is a security best practice to use SNMP v3 instead.

SNMP version security options

SNMPv3 provides secure access to devices by using three aspects: - users, authentication, and encryption. SNMPv3 uses the USM (User-based Security Module) for controlling access to information available via SNMP. The SNMP v3 user is configured with an authentication type, a privacy type as well as a passphrase. All users sharing a group utilize the same SNMP version, however, the specific security level settings (password, encryption type, etc.) are specified per-user.

This table summarizes the security options within SNMP:

Model	Level	Authentication	Encyption	Outcome
v1	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v2c	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v3	noAuthNoPriv	Username	No	Uses a username match for authentication.
v3	authNoPriv	Message Digest 5 (MD5) or Secure Hash Algorithm (SHA)	No	Provides authentication based on the HMAC-MD5-96 or HMAC-SHA-96 algorithms.
v3	authPriv	MD5 or SHA	Data Encryption Standard (DES) or Advanced Encryption Standard (AES)	Provides authentication based on the HMAC-MD5-96 or HMAC-SHA-96 algorithms. Provides DES Cipher algorithm in Cipher Block Chaining Mode (CBC-DES) or AES encryption algorithm used in Cipher FeedBack Mode (CFB), with a 128-bit key size(CFB128-AES-128)

Since its adoption by NIST, AES has become the dominant encryption algorithm throughout the industry. To follow the industry's migration away from MD5 and toward SHA, it is a security best practice to configure the SNMP v3 authentication protocol as SHA and privacy protocol as AES.

SNMP on NFVIS supports V1, V2, and V3 — but only SNMPv3 with authPriv (sha256/AES) operates in secure mode; all other combinations (V1, V2, and V3 with noAuthNoPriv or authNoPriv using md5/SHA/DES) are classified as insecure.

Insecure mode dependency

Objective

Insecure mode dependency.

SNMP requirements in insecure mode

```
nfvis# show system mode
system mode status secure
nfvis(config)# snmp user test user-version 1 user-group test_group
auth-protocol md5 priv-protocol des passphrase qwertyuiop
encryption-passphrase qwertyuiop
nfvis(config-user-test)# commit
Aborted: 'snmp user test user-version': SNMP Version 1 is insecure. Enable insecure mode
to configure this
```

Insecure mode

Objective

Insecure mode.

Insecure mode commands and behavior

```
nfvis# show system mode
system mode status insecure
nfvis(config)# snmp user test user-version 1 user-group test_group
auth-protocol md5 priv-protocol des passphrase qwertyuiop
encryption-passphrase qwertyuiop
nfvis(config-user-test)# commit
Commit complete.
```

Banners

Objective

We recommend that a legal notification banner is present on all interactive sessions to ensure that users are notified of the security policy being enforced and to which they are subject.

Legal notification banner considerations

In some jurisdictions, civil and/or criminal prosecution of an attacker who breaks into a system is easier, or even required, if a legal notification banner is presented, informing unauthorized users that their use is in fact unauthorized. In some jurisdictions, it may also be forbidden to monitor the activity of an unauthorized user unless they have been notified of the intent to do so.

Banner content requirements

Objective

Discuss this issue with your own legal counsel to ensure that the notification banner meets company, local, and international legal requirements.

Legal banner content requirements

- Notification that the system access and use is permitted only by specifically authorized personnel, and perhaps information about who may authorize use.
- Notification that unauthorized access and use of the system is unlawful, and may be subject to civil and/or criminal penalties.
- Notification that access and use of the system may be logged or monitored without further notice, and the resulting logs may be used as evidence in court.
- Additional specific notices required by specific local laws.

Legal notification requirements are complex and vary in each jurisdiction and situation. Even within jurisdictions, legal opinions vary. This is often critical to securing appropriate action in the event of a security breach.

Security considerations for banner content

A legal notification banner should not contain any specific information about the device, such as its name, model, software, location, operator or owner because this kind of information may be useful to an attacker.

Sample banner implementation

The following is a sample legal notification banner which can be displayed before login:

```
UNAUTHORIZED ACCESS TO THIS DEVICE IS PROHIBITED You must have explicit, authorized permission
 to access or configure this device. Unauthorized attempts and actions to access or use
this system may result in civil and/or criminal penalties. All activities performed on this
 device are logged and monitored
```



Note Present a legal notification banner approved by company legal counsel.

NFVIS banner configuration

Objective

We recommend that a login banner is implemented to ensure that a legal notification banner is presented on all the device management access sessions prior to a login prompt being presented.

Banner and MOTD configuration command

NFVIS allows the configuration of a banner and Message of the Day (MOTD). The banner is displayed before the user logs in. Once the user logs in to NFVIS, a system-defined banner provides Copyright information about NFVIS, and the message-of-the-day (MOTD), if configured, will appear, followed by the command line prompt or portal view, depending on the login method. Use this command to configure the banner and MOTD:

```
nfvis(config)# banner-motd banner banner-text motd message-of-the-day-text
```

For more information about the banner command, see Configure your banner and message of the day.

Factory default reset

A factory default reset is a device reset feature that

- removes all customer-specific data that has been added to the device since the time of its shipping
- erases configurations, log files, VM images, connectivity information, and user login credentials, and
- provides one command to reset the device to factory-original settings.

Factory default reset is useful in these scenarios:

- Return Material Authorization (RMA) for a device—If you have to return a device to Cisco for RMA, use Factory Default reset to remove all the customer-specific data.
- Recovering a compromised device— If the key material or credentials stored on a device is compromised, reset the device to factory configuration and then reconfigure the device.
- If the same device needs to be re-used at a different site with a new configuration, perform a Factory Default reset to remove the existing configuration and bring it to a clean state.

NFVIS provides these options within Factory default reset:

Factory Reset Option	Data Erased	Data Retained
all	All configuration, uploaded image files, VMs and logs. Connectivity to the device will be lost.	The admin account is retained and the password will be changed to the factory default password.
all-except-images	All configuration except image configuration, VMs, and uploaded image files. Connectivity to the device will be lost.	Image configuration, registered images and logs The admin account is retained and the password will be changed to the factory default password.

Factory Reset Option	Data Erased	Data Retained
all-except-images-connectivity	All configuration except image, network and connectivity configuration, VMs, and uploaded image files. Connectivity to the device is available.	Images, network and connectivity related configuration, registered images, and logs. The admin account is retained and the previously configured admin password will be preserved.
manufacturing	All configuration except image configuration, VMs, uploaded image files, and logs. Connectivity to the device will be lost.	Image related configuration and registered images The admin account is retained and the password will be changed to the factory default password.

The user must choose the appropriate option carefully based on the purpose of the Factory Default reset.

Infrastructure management networks

An infrastructure management network is a network that

- carries the control and management plane traffic (such as NTP, SSH, SNMP, syslog, etc.) for the infrastructure devices
- provides visibility into and control over the network through device access via console and Ethernet interfaces, and
- enables remote manageability even under high load and high traffic conditions.

Infrastructure management network design considerations

This control and management plane traffic is critical to network operations. Consequently, a well-designed and secure infrastructure management network is critical to the overall security and operations of a network. One of the key recommendations for a secure infrastructure management network is the separation of management and data traffic in order to ensure remote manageability even under high load and high traffic conditions. This can be achieved using a dedicated management interface.

Infrastructure management network implementation approaches are:

Out-of-band management

Out-of-band management (OOB) is a network management approach that

- uses a network which is completely independent and physically disparate from the data network that it helps to manage
- is sometimes referred to as a Data Communications Network (DCN), and
- provides greater control over device management by restricting management packets to designated interfaces.

Connection methods and benefits

Objective

Network devices can connect to the OOB network in different ways:

OOB connection methods and benefits

- NFVIS supports a built-in management interface that can be used to connect to the OOB network. NFVIS allows the configuration of a predefined physical interface, the MGMT port on the ENCS, as a dedicated management interface.
- Network devices can also connect to the OOB network via dedicated data interfaces. In this case, ACLs should be deployed to ensure that management traffic is only handled by the dedicated interfaces.

Benefits include:

- More security for devices by restricting management packets to designated interfaces
- Improved performance for data packets on non-management interfaces
- Support for network scalability
- Need for fewer access control lists (ACLs) to restrict access to a device
- Prevention of management packet floods from reaching the CPU.

Pseudo out-of-band management

A pseudo out-of-band management network is a network configuration that

- uses the same physical infrastructure as the data network but provides logical separation through the virtual separation of traffic, by using VLANs
- enables NFVIS to create VLANs and virtual bridges to help identify different sources of traffic and separate traffic between VMs, and
- isolates the virtual machine network's data traffic and the management network through separate bridges and VLANs, thus providing traffic segmentation between the VMs and the host.

In-band management

An in-band management network is a management approach that uses the same physical and logical paths as the data traffic.

Network design considerations

This network design requires a per-customer analysis of risk versus benefits and costs. Some general considerations include:

- An isolated OOB management network maximizes visibility and control over the network even during disruptive events.

- Transmitting network telemetry over an OOB network minimizes the chance for disruption of the very information which provides critical network visibility.
- In-band management access to network infrastructure, hosts, etc. is vulnerable to complete loss in the event of a network incident, removing all the network visibility and control. Appropriate QoS controls should be put in place to mitigate this occurrence.
- NFVIS features interfaces which are dedicated to device management, including serial console ports and Ethernet management interfaces.
- An OOB management network can typically be deployed at a reasonable cost, since management network traffic does not typically demand high bandwidth nor high performance devices, and only requires sufficient port density to support the connectivity to each infrastructure device.

Protecting sensitive information

Protecting sensitive information is a security mechanism that

- stores passwords and secrets locally on NFVIS as hashes to prevent recovery of original credentials
- maintains passwords through centralized AAA servers while providing local fallback capabilities, and
- follows widely accepted industry norms for password protection.

Local password storage requirements

Objective

NFVIS requires locally-stored passwords for specific cases even when centralized AAA servers are deployed:

Cases requiring locally stored passwords

- Local fallback when AAA servers are not available
- Special-use usernames

File transfer

File transfer is a network operation that

- enables the secure copying of VM image and NFVIS upgrade files to NFVIS devices
- uses Secure Copy (SCP) protocol to ensure security and authentication during transfer, and
- relies on SSH for secure authentication and transport mechanisms.

File transfer implementation

Objective

A secure copy from NFVIS is initiated through the SCP command. The secure copy (SCP) command allows only the admin user to securely copy files from NFVIS to an external system, or from an external system to NFVIS.

SCP syntax and transfer behavior

The syntax for the SCP command is:

```
scp source destination
```

NFVIS uses port 22222 for the SCP server. By default, this port is closed and users cannot secure copy files into NFVIS from an external client. If there is a need to SCP a file from an external client, the user can open the port using:

```
system settings ip-receive-acl (@@VAR0@@)/(@@VAR1@@) service scpd priority
 (@@VAR2@@) action accept
commit
```

To prevent users from accessing system directories, secure copy can be performed only to or from intdatastore:, extdatastore1:, extdatastore2:, usb: and nfs:, if available. Secure copy can also be performed from logs and techsupport.

Logging

Logging is a security mechanism that

- records NFVIS access and configuration changes as audit logs
- provides forensic analysis capabilities for unauthorized access attempts and configuration issues, and
- enables real-time identification of anomalous activities that may indicate attacks.

NFVIS access and configuration changes are logged as audit logs to record this information:

- Who accessed the device
- When did a user log in
- What did a user do in terms of the host configuration and the VM lifecycle
- When did a user log off
- Failed access attempts
- Failed authentication requests
- Failed authorization requests

This information is invaluable for forensic analysis in case of unauthorized attempts or access, as well as for configuration change issues and to help plan group administration changes. It may also be used real time to

identify anomalous activities which may indicate that an attack is taking place. This analysis can be correlated with information from additional external sources, such as IDS and firewall logs.

All the key events on the NFVIS are sent as event notifications to NETCONF subscribers and as syslogs to the configured central logging servers.

Virtual machine security

Objective

This section describes security features related to the registration, deployment and operation of Virtual Machines on NFVIS.

VNC console access protection

VNC console access protection is a security mechanism that

- allows users to create Virtual Network Computing (VNC) sessions to access a deployed VM's remote desktop
- dynamically opens a port for 60 seconds to which users can connect using their web browser, and
- assigns port numbers dynamically to allow only one-time access to the VNC console.

How VNC console access works

Summary

NFVIS dynamically opens a port when a user creates a VNC session. This port is only left open for 60 seconds for an external server to start a session to the VM. If no activity is seen within this time, the port is closed.

Workflow

NFVIS dynamically opens a port when a user creates a VNC session.

VNC console command example

Objective

Pointing your browser to https://<NFVIS ip>:6005/vnc_auto.html will connect to the ROUTER VM's VNC console.

VNC console command example

```
nfvis# vncconsole start deployment-name 1510614035 vm-name ROUTER
vncconsole-url :6005/vnc_auto.html
```

Encrypted VM config data variables

An encrypted VM config data variable is a security mechanism that

- allows users to flag config data variables as sensitive during VM deployment
- encrypts sensitive values using AES-CFB-128 encryption before storage or passing to internal subsystems, and
- prevents passwords and keys from appearing as clear text in log files and internal database records.

During VM deployment, the user provides a day-0 configuration file for the VM. This file can contain sensitive information such as passwords and keys. If this information is passed as clear text, it appears in log files and internal database records in clear text.

Checksum verification for remote image registration

Checksum verification for remote image registration is a security mechanism that

- verifies the integrity of downloaded VNF images from external sources
- ensures files are not corrupted during network transmission or modified by malicious third parties, and
- uses SHA256 or SHA512 algorithms to validate image checksums before installation.

How checksum verification works

Summary

When registering a remotely located VNF image, the user specifies its location for download from external sources such as NFS servers or remote HTTPS servers.

Workflow

NFVIS supports checksum and checksum_algorithm options that allow users to provide the expected checksum and specify the checksum algorithm (SHA256 or SHA512) for verifying the downloaded image. Image creation fails if the checksum does not match the expected value.

Certification validation for remote image registration

Certification validation for remote image registration is a security mechanism that

- verifies SSL certificates when downloading VNF images from remote HTTPS servers
- requires users to specify either the path to the certificate file or PEM format certificate contents, and
- ensures secure downloads during the image registration process in NFVIS.

Certificate specification requirements

Objective

When registering a VNF image located on an HTTPS server, the image must be downloaded from the remote HTTPS server. To securely download this image, NFVIS verifies the SSL certificate of the server. The user needs to specify either the path to the certificate file or the PEM format certificate contents to enable this secure download.

VM isolation and resource provisioning

VM isolation and resource provisioning is a network function virtualization capability that

- partitions physical hardware resources into separate virtual environments for multiple VNFs
- ensures individual VM domains are isolated as separate, distinct, and secure environments that do not contend with each other for shared resources, and
- prevents VMs from using more resources than provisioned to avoid Denial of Service conditions.

NFV architecture components

Objective

The Network Function Virtualization (NFV) architecture consists of:

NFV architecture components and resource protection

- Virtualized network functions (VNFs), which are Virtual Machines running software applications that deliver network functionality such as a router, firewall, load balancer, and so on.
- Network functions virtualization infrastructure, which consists of the infrastructure components—compute, memory, storage, and networking, on a platform that supports the required software and hypervisor.

With NFV, network functions are virtualized so that multiple functions can be run on a single server. As a result, less physical hardware is needed, allowing for resource consolidation. In this environment, it is essential to simulate dedicated resources for multiple VNFs from a single, physical hardware system. Using NFVIS, VMs can be deployed in a controlled manner such that each VM receives the resources it needs. Resources are partitioned as needed from the physical environment to the many virtual environments.

As a result, CPU, memory, network and storage are protected.

CPU isolation

CPU isolation is a resource management mechanism that

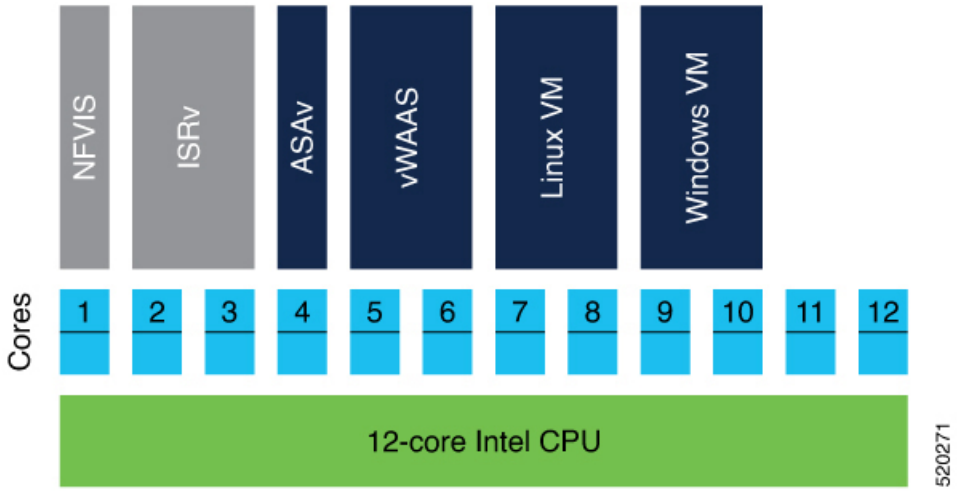
- reserves cores for infrastructure software running on the host
- makes the remaining cores available for VM deployment, and
- guarantees that VM performance does not affect NFVIS host performance.

CPU allocation by VM type

Objective

NFVIS handles CPU allocation differently based on VM latency requirements.

CPU allocation methods by VM type



The system uses two distinct allocation methods:

- Low-latency VMs: NFVIS explicitly assigns dedicated cores to low latency VMs. If the VM requires 2 vCPUs, it is assigned 2 dedicated cores. This prevents sharing and oversubscription of cores and guarantees the performance of the low-latency VMs. If the number of available cores is less than the number of vCPUs requested by another low-latency VM, the deployment is prevented since we do not have sufficient resources.
- Non low-latency VMs: NFVIS assigns sharable CPUs to non low latency VMs. If the VM requires 2 vCPUs, it is assigned 2 CPUs. These 2 CPUs are shareable among other non low latency VMs. If the number of available CPUs is less than the number of vCPUs requested by another non low-latency VM, the deployment is still allowed because this VM will share the CPU with existing non low latency VMs.

Memory allocation

Memory allocation is a resource management mechanism that

- reserves memory for NFVIS Infrastructure and previously deployed VMs before allowing new VM deployment
- ensures sufficient memory availability through validation checks, and
- prevents memory oversubscription for VMs.

How memory is allocated

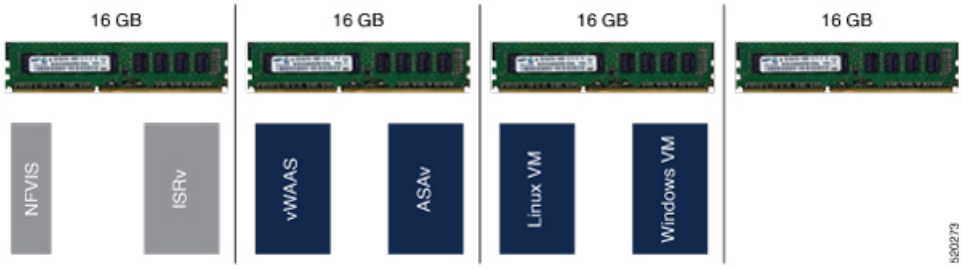
Summary

When a VM is deployed, there is a check to ensure that the memory available after reserving the memory required for the infrastructure and previously deployed VMs, is sufficient for the new VM.

Workflow

VMs are not allowed to directly access the host file system and storage.

Process details



Interface isolation

Interface isolation is a virtualization capability that

- allows the isolation of PCI Express (PCIe) resources such as an Ethernet port using Single Root I/O Virtualization (SR-IOV),
- enables a single Ethernet port to appear as multiple, separate, physical devices known as Virtual Functions, and
- provides data protection between guests on the same physical server as the data is managed and controlled by the hardware.

SR-IOV implementation details

Objective

All Virtual Function devices on an adapter share the same physical network port. A guest can use one or more of these Virtual Functions, with each Virtual Function appearing to the guest as a network card, in the same way as a normal network card would appear to an operating system.

Virtual Function performance and usage

Virtual Functions provide the following performance characteristics:

- Near-native performance
- Better performance than para-virtualized drivers and emulated access

NFVIS VNFs can use SR-IOV networks to connect to WAN and LAN Backplane ports. Each VM owns a virtual interface and its related resources achieving data protection among VMs.

Secure development lifecycle

A secure development lifecycle is a software development process that

- reduces vulnerabilities and enhances the security and resilience of Cisco solutions
- applies industry-leading practices and technology to build trustworthy solutions, and
- results in fewer field-discovered product security incidents.

Attack vector reduction

Attack vector reduction is a security strategy that

- limits software packages to only those essential for NFVIS functionality to reduce potential security vulnerabilities
- registers all third-party software in a central Cisco database for organized security response, and
- ensures periodic patching of software packages for known Common Vulnerabilities and Exposures (CVEs) in every release.

This approach provides multiple security advantages. Any piece of software can potentially contain security vulnerabilities. More software means more avenues for attack. Even if there are no publicly known vulnerabilities at the time of inclusion, vulnerabilities will probably be discovered or disclosed in the future. To avoid such scenarios, only those software packages which are essential for the NFVIS functionality are installed. This helps to limit software vulnerabilities, reduce resource consumption, and reduce extra work when problems are found with those packages.

NFVIS SDL processes

Objective

Every NFVIS release goes through these processes:

NFVIS security development lifecycle

- Following Cisco-internal and market-based Product Security Requirements
- Registering 3rd party software with a central repository at Cisco for vulnerability tracking
- Periodically patching software with known fixes for CVEs
- Designing software with Security in mind
- Following secure coding practices such as using vetted common security modules like CiscoSSL, running Static Analysis and implementing input validation for Preventing command injection
- Using Application Security tools such as IBM AppScan, Nessus, and other Cisco internal tools.

Public sector security certifications

Objective

This reference groups information about Public Sector Security Certifications, including FIPS 140-3.

FIPS 140-3

FIPS mode on NFVIS is a security compliance mechanism that

- implements Federal Information Processing Standards (FIPS) Publication 140-3 for United States federal government and contractor use
- attempts to prevent the use of non-FIPS compatible algorithms on the device, and
- requires manual configuration to ensure only FIPS approved algorithms are used.

How FIPS mode works

Summary

In FIPS mode, certain functions may fail silently if they attempt to use non-compliant algorithms. You must ensure that the device is configured to use only FIPS-approved algorithms. FIPS mode is enabled by default on BExK platforms.

Workflow

In FIPS mode, certain functions may fail silently if they attempt to use non-compliant algorithms.

Disable FIPS mode

To disable FIPS mode, first perform a factory reset. The same sequence applies when you re-enable FIPS mode.

Procedure

Step 1 Reset NFVIS to factory default.

For the reset procedure, see Reset NFVIS to factory default.

Step 2 After the reset, disable FIPS mode and reboot the device.

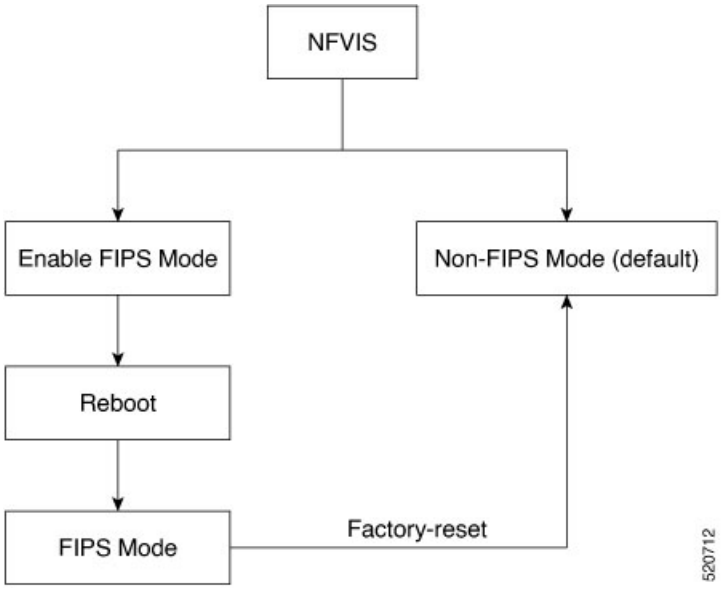
Configure FIPS mode

Before you begin

Before you configure FIPS mode, ensure that SNMP v1, v2, or v3 with the MD5 authentication protocol is not configured. FIPS mode configuration is terminated if any of these are configured.

NFVIS supports FIPS and non-FIPS modes. FIPS mode enables FIPS 140-3-compliant operation for SSH, TLS, and SNMP protocols.

The diagram shows the FIPS mode transition.



Procedure

Step 1 Enable FIPS mode.

Enter the following commands:

```
config terminal
security fips
commit
```

FIPS mode is enabled only after NFVIS reboots. Before reboot, verify that FIPS mode is configured and pending enablement:

```
nfvis# show security
security fips-status CONFIGURED_REBOOT_TO_ENABLE
```

Note FIPS mode can be disabled only with a factory reset. The **no** form of the command cannot disable FIPS mode:

```
config terminal
no security fips
commit
Aborted: This command can only be removed while factory reset
```

Step 2 Reboot NFVIS to enable FIPS mode.

Step 3 Verify the FIPS mode status after reboot.

Use the following command:

```
nfvis# show security
security fips-status ENABLED
```

FIPS mode configuration is terminated when SNMP v1 or v2 is configured, or when SNMP v3 is configured with MD5 authentication.

The following example shows the error returned when SNMP v1 or v2 is configured:

```
config terminal
security fips
commit
Aborted: SNMP version 1 and/or SNMP version 2 is configure. Please unconfigure SNMPv1 and SNMPv2 and
 then try again
```

The following example shows the error returned when SNMP v3 is configured with MD5 authentication:

```
config terminal
security fips
commit
Aborted: SNMP version 3 MD5 auth-protocol configured other secure protocol and try again
```

Note After FIPS mode is enabled, you cannot configure SNMP v1 or v2, or SNMP v3 with MD5 authentication. The following examples show the errors returned for these configurations:

```
snmp group test_v1 snmp 1 noAuthNoPriv read test write test
commit
Aborted: Cannot configure SNMP group-version 1 because fips-status is ENABLED
config terminal
snmp user test_md5_v3
user-version 3
user-group test_v3
auth-protocol md5
auth-key 46:97:c3:b0:ba:45:fd:5e:be:99:44:c5:64:c9:bc:44
commit
Aborted: 'snmp user test_md5_v3_passhd auth-protocol': Cannot configure SNMP user-version
3 with auth-protocol MD5 because fips-status is CONFIGURED_REBOOT_TO_ENABLE
nfvis(config-user-test_md5_v3_passhd)#
```

FIPS mode is enabled, and NFVIS operates in FIPS 140-3-compliant mode for SSH, TLS, and SNMP protocols.

How backup and restore work in FIPS mode

Summary

This topic provides the details on how FIPS mode status is handled during NFVIS backup and restore operations.

Workflow

If you back up NFVIS configurations when FIPS mode is enabled, then upon restore, FIPS mode is configured but needs a manual reboot to enable it.

Process details

Backup configuration

```
nfvis#
nfvis# show running-config security
security fips
nfvis# show security
security fips-status ENABLED
nfvis#
```

After restore

```
nfvis#
nfvis# show running-config security
security fips
nfvis# show security fips-status
security fips-status CONFIGURED_REBOOT_TO_ENABLE
nfvis#
```

After reboot

```
nfvis#
nfvis# show running-config security
security fips
nfvis# show security
security fips-status ENABLED
nfvis#
```

When you backup NFVIS configurations with FIPS mode disabled, but the system where you restore the configurations has FIPS mode enabled, upon restore, the NFVIS configurations disable FIPS mode but the system has to reboot for FIPS mode to be in DISABLED state.

Backup configurations

```
Backup configurations
nfvis# show running-config security fips
% No entries found.
nfvis# show security fips-status
security fips-status DISABLED
nfvis#
```

Restore system configurations

```
Restore system configurations
nfvis#
nfvis# show running-config security
security fips
nfvis# show security
security fips-status ENABLED
nfvis#
```

After restore

```
After restore
nfvis# show running-config security
% No entries found.
nfvis# show security
```

```
security fips-status UNCONFIGURED_REBOOT_TO_DISABLE
nfvis#
```

After reboot

```
After reboot
nfvis# show running-config security fips
% No entries found.
nfvis# show security fips-status
security fips-status DISABLED
nfvis#
```

FIPS operational status

Objective

This reference lists the operational states available when you enable FIPS mode and explains the possible state transitions.

FIPS operational states

These are the operational states when you try to ENABLE FIPS mode:

- DISABLED
- CONFIGURED-REBOOT-TO-ENABLE
- ENABLED
- UNCONFIGURED-REBOOT-TO-DISABLE
- FAILED

From	To	Description
DISABLED	CONFIGURED-REBOOT-TO-ENABLE	If the Oper data of FIPS-state leafs was previously set to DISABLED and if the security FIPS configuration is pushed
DISABLED	FAILED	If there is an error while pushing the security FIPS configuration
CONFIGURED-REBOOT-TO-ENABLE	ENABLED	If the FIPS-mode configuration is successful before and the Oper data was set to CONFIGURED-REBOOT-TO-ENABLE, then after REBOOT set the Oper data to ENABLED
CONFIGURED-REBOOT-TO-ENABLE	FAILED	If the Oper data of FIPS-state leafs was previously set to CONFIGURED-REBOOT-TO-ENABLE and there was an error while removing FIPS-mode configuration

From	To	Description
CONFIGURED-REBOOT-TO-ENABLE	DISABLED	If the FIPS-mode is UNCONFIGURED while restoring from a backup package or factory-reset and the current FIPS-status is CONFIGURED-REBOOT-TO-ENABLE
ENABLED	DISABLED	After factory-reset (of any type)
ENABLED	FAILED	If there is an error while disabling the FIPS mode
ENABLED	UNCONFIGURED-REBOOT-TO-DISABLE	If the FIPS-mode is UNCONFIGURED while restoring from a backup package and the current FIPS-status is ENABLED
FAILED	CONFIGURED-REBOOT-TO-ENABLE	If the Oper date of FIPS-state leafs was previously set to FAILED and now configuring FIPS-mode
FAILED	DISABLED	If the Oper date of FIPS-state leafs was previously set to FAILED and now issued factory-reset
UNCONFIGURED-REBOOT-TO-DISABLE	DISABLED	If the Oper data of FIPS-state leafs was previously set to UNCONFIGURED-REBOOT-TO-DISABLE and then NFVIS is rebooted
UNCONFIGURED-REBOOT-TO-DISABLE	CONFIGURED-REBOOT-TO-ENABLE	If the Oper data of FIPS-state leafs was previously set to UNCONFIGURED-REBOOT-TO-DISABLE and FIPS-mode config is successful
UNCONFIGURED-REBOOT-TO-DISABLE	FAILED	If the Oper data of FIPS-state leafs was previously set to UNCONFIGURED-REBOOT-TO-DISABLE and if the FIPS-mode config was unsuccessful