



Extreme 9920 Software Security Configuration Guide, 21.1.0.0

Supporting Extreme 9920

9037106-00 Rev AA
June 2021



Copyright © 2021 Extreme Networks, Inc. All rights reserved.

Legal Notice

Extreme Networks, Inc. reserves the right to make changes in specifications and other information contained in this document and its website without prior notice. The reader should in all cases consult representatives of Extreme Networks to determine whether any such changes have been made.

The hardware, firmware, software or any specifications described or referred to in this document are subject to change without notice.

Trademarks

Extreme Networks and the Extreme Networks logo are trademarks or registered trademarks of Extreme Networks, Inc. in the United States and/or other countries.

All other names (including any product names) mentioned in this document are the property of their respective owners and may be trademarks or registered trademarks of their respective companies/owners.

For additional information on Extreme Networks trademarks, see: www.extremenetworks.com/company/legal/trademarks

Open Source Declarations

Some software files have been licensed under certain open source or third-party licenses. End-user license agreements and open source declarations can be found at: <https://www.extremenetworks.com/support/policies/open-source-declaration/>



Table of Contents

Preface	5
Text Conventions.....	5
Documentation and Training.....	6
Getting Help.....	7
Subscribe to Product Announcements.....	7
Providing Feedback.....	7
What's New in This Document	9
Secure Shell	10
User Accounts and Passwords	11
Default Account Credentials.....	11
Predefined Accounts and Roles.....	11
User Account and Role Commands	12
Account Guidelines and Limitations.....	12
Basic Account Management.....	13
Create a New Admin-Role User Account.....	13
Create a New User-Role User Account.....	13
Remote Server Authentication	15
Remote Server Authentication Overview.....	15
Login Authentication Mode.....	15
Conditions for Conformance.....	16
Set and Verify the Login Authentication Mode.....	16
Reset the Login Authentication Mode to the Default.....	17
TACACS+ Server Authentication	18
TACACS+ Authentication and Accounting.....	18
Supported TACACS+ Packages and Protocols.....	18
TACACS+ Configuration Components.....	19
Client Configuration Parameters for TACACS+ Support.....	19
Configure the Client to Use TACACS+ for Login Authentication.....	20
Add a TACACS+ Server to the Client Server List.....	20
Modify the Client-Side TACACS+ Server Configuration.....	21
Remove a TACACS+ Server Key from the Client.....	22
Remove Client-Side TACACS+ Server Configuration.....	23
HTTPS Certificates	24
Import or Replace an HTTPS Certificate.....	24
Remove an Imported HTTPS Certificate.....	25
Token-Based Authentication	26
Token-Based Authentication for gRPC Requests.....	26
Token-Based Authentication Guidelines and Limitations.....	26

Token-Based Authentication Flow.....	27
Remote System Logging.....	28
Set Up a Remote Logging Server.....	28
Configure Remote Logging Server Storage.....	29
Configure Remote Logging to Use UDP.....	29
Configure Remote Logging to Use TCP.....	30
Install Certificates Required for TLS Encryption.....	31
Configure Remote Logging to Use TCP with TLS Encryption.....	31



Preface

Read the following topics to learn about:

- The meanings of text formats used in this document.
- Where you can find additional information and help.
- How to reach us with questions and comments.

Text Conventions

Unless otherwise noted, information in this document applies to all supported environments for the products in question. Exceptions, like command keywords associated with a specific software version, are identified in the text.

When a feature, function, or operation pertains to a specific hardware product, the product name is used. When features, functions, and operations are the same across an entire product family, such as ExtremeSwitching switches or SLX routers, the product is referred to as *the switch* or *the router*.

Table 1: Notes and warnings






Icon	Notice type	Alerts you to...
	Tip	Helpful tips and notices for using the product
	Note	Useful information or instructions
	Important	Important features or instructions
	Caution	Risk of personal injury, system damage, or loss of data
	Warning	Risk of severe personal injury

Table 2: Text

Convention	Description
screen displays	This typeface indicates command syntax, or represents information as it is displayed on the screen.
The words <i>enter</i> and <i>type</i>	When you see the word <i>enter</i> in this guide, you must type something, and then press the Return or Enter key. Do not press the Return or Enter key when an instruction simply says <i>type</i> .
Key names	Key names are written in boldface, for example Ctrl or Esc . If you must press two or more keys simultaneously, the key names are linked with a plus sign (+). Example: Press Ctrl+Alt+Del
Words in italicized type	Italics emphasize a point or denote new terms at the place where they are defined in the text. Italics are also used when referring to publication titles.
NEW!	New information. In a PDF, this is searchable text.

Table 3: Command syntax

Convention	Description
bold text	Bold text indicates command names, keywords, and command options.
<i>italic</i> text	Italic text indicates variable content.
[]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ x y z }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
x y	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, such as passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, <i>member</i> [<i>member</i> . . .].
\	In command examples, the backslash indicates a “soft” line break. When a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Documentation and Training

Find Extreme Networks product information at the following locations:

[Current Product Documentation](#)

[Release Notes](#)

[Hardware and software compatibility](#) for Extreme Networks products

[Extreme Optics Compatibility](#)

[Other resources](#) such as white papers, data sheets, and case studies

Extreme Networks offers product training courses, both online and in person, as well as specialized certifications. For details, visit www.extremenetworks.com/education/.

Getting Help

If you require assistance, contact Extreme Networks using one of the following methods:

Extreme Portal

Search the GTAC (Global Technical Assistance Center) knowledge base; manage support cases and service contracts; download software; and obtain product licensing, training, and certifications.

The Hub

A forum for Extreme Networks customers to connect with one another, answer questions, and share ideas and feedback. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.

Call GTAC

For immediate support: (800) 998 2408 (toll-free in U.S. and Canada) or 1 (408) 579 2826. For the support phone number in your country, visit: www.extremenetworks.com/support/contact

Before contacting Extreme Networks for technical support, have the following information ready:

- Your Extreme Networks service contract number, or serial numbers for all involved Extreme Networks products
- A description of the failure
- A description of any actions already taken to resolve the problem
- A description of your network environment (such as layout, cable type, other relevant environmental information)
- Network load at the time of trouble (if known)
- The device history (for example, if you have returned the device before, or if this is a recurring problem)
- Any related RMA (Return Material Authorization) numbers

Subscribe to Product Announcements

You can subscribe to email notifications for product and software release announcements, Field Notices, and Vulnerability Notices.

1. Go to [The Hub](#).
2. In the list of categories, expand the **Product Announcements** list.
3. Select a product for which you would like to receive notifications.
4. Select **Subscribe**.
5. To select additional products, return to the **Product Announcements** list and repeat steps 3 and 4.

You can modify your product selections or unsubscribe at any time.

Providing Feedback

The Information Development team at Extreme Networks has made every effort to ensure the accuracy and completeness of this document. We are always striving to improve our documentation and help you work better, so we want to hear from you. We welcome all feedback, but we especially want to know about:

- Content errors, or confusing or conflicting information.

- Improvements that would help you find relevant information in the document.
- Broken links or usability issues.

If you would like to provide feedback, you can do so in three ways:

- In a web browser, select the feedback icon and complete the online feedback form.
- Access the feedback form at <https://www.extremenetworks.com/documentation-feedback/>.
- Email us at documentation@extremenetworks.com.

Provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.



What's New in This Document

This document is new for the release of the Extreme 9920 software with the NPB application.

For more information about this release, see the [Extreme 9920 Software Release Notes, 21.1.0.0](#).



Secure Shell

Learn how SSH is supported for accessing the Extreme 9920.

Secure Shell (SSH) is a protocol that encrypts remote access connections to network devices. Using encrypted shared keys (RSA/ECDSA), SSH authenticates clients or servers, ensuring that the devices accessing your network are authentic. SSH Authentication is supported using basic authentication (username-password based) only.

Secure Shell (SSH) is a protocol that encrypts remote access connections to network devices. Using encrypted shared keys SSH authenticates clients or servers, ensuring that the devices accessing your network are authentic.

Up to 32 SSH logins are permitted.

SSH Server Support

- Support for SSH server is available and the device ensures RSA and ECDSA host key pairs are always available for use during SSH operation.
- Remote user authentication and locally stored usernames and passwords are supported for SSH.

Authentication Support (Local + Remote)

- SSH Authentication is supported using basic authentication (username-password) only.



User Accounts and Passwords

- [Default Account Credentials on page 11](#)
- [Predefined Accounts and Roles on page 11](#)
- [User Account and Role Commands on page 12](#)
- [Account Guidelines and Limitations on page 12](#)
- [Basic Account Management on page 13](#)
- [Create a New Admin-Role User Account on page 13](#)
- [Create a New User-Role User Account on page 13](#)

The NPB application uses role-based access control (RBAC) as the authorization mechanism for access to resources. A *role* is assigned to a user account and is a container for rules that specify which commands can be executed and with which permissions. When you create a user account you need to specify a role for that account. In general, *user* (as opposed to *user-level*) refers to any account—to which any role can be assigned—user or admin.

The following topics describe accounts and roles and how to configure and manage them.

Default Account Credentials

The NPB application ships with two default user accounts.

When you install the NPB application on Extreme 9920, two default user accounts are provided—**admin** and **user**—with the following case-sensitive default passwords:

- admin account password: **rocks**
- user account password: **password**

As a best practice, log on as the administrator and change the default passwords immediately after the NPB application is installed.

Predefined Accounts and Roles

Learn about predefined accounts and roles for the NPB application

The NPB application ships with two predefined accounts—admin and user. The maximum number of user accounts that you can configure is 64, including the predefined accounts.

- **admin**—Accounts with admin role access can execute all commands supported on the device.

- **user**—Accounts with user-level access have read-only permissions. User-level accounts can run the following operational CLI commands.

Table 4: User-level operational commands

Command	Action
dir	List flash files
end	End current mode and change to enable mode
exit	Exit current mode and revert to previous mode
list	Print command list
ping	Ping
quit	Exit current mode and revert to previous mode
show	Show values
terminal	Set terminal timeout parameters
traceroute	Run traceroute

User Account and Role Commands

See a quick overview of the commands that set and display user account and role information.

For complete information for using these commands, see the [Extreme 9920 Software Command Reference, 21.1.0.0](#)

Table 5: User account and password commands

Command	Description
username <username> role <role> password <password> [encryption- level <0 10>]	Sets the role, password, and encryption level for the specified username.
show role	Displays all role information available in the system.

Account Guidelines and Limitations

Learn about the guidelines and limitations for creating accounts.

- Extreme recommends that access to the CLI for every user is through a unique account: After logging in as admin, create a unique account for yourself, specifying **role admin**.
- You cannot modify rules for the admin or the user predefined accounts.
- You cannot modify rules for the admin or the user default roles.
- By default, all account information is stored in the device-local user database.
- By default, user authentication and tracking of logins to the device is local.
- The maximum number of accounts—including the predefined accounts—is 64. If you need more than 64 user accounts, configure remote server authentication (for more information, see the [Remote Server Authentication](#) on page 15 section).

- The maximum number of TACACS+ servers is 5.
- The maximum number of simultaneous active SSH sessions is 32.

Basic Account Management

Learn how to create and manage basic admin and user accounts.

You can create user- and admin- role accounts for using the the application.

Create a New Admin-Role User Account

Create a new user account associated with the admin role that can run all supported CLI commands.

Before You Begin

You must have the admin role to create a new user account that has the admin-role.

About This Task

Be sure to observe the following guidelines as you create the user account:

Table 6: User-account guidelines

User-defined variables	Values
<i>username</i>	1-40 alphanumeric characters, including underscore and dot. Underscore as first character is not allowed.
<i>rolename</i>	Pre-defined role to be assigned to the user ('admin' and 'user' are the only supported roles).
<i>password</i>	8-40 characters for plain-text password 8-128 characters for hashed passwords
<i>encryption-level</i>	0 = clear-text (default) 10 = encrypted

Procedure

1. Run the **configure terminal** command to access Config mode.

The command line changes to configuration mode.

```
device(config)#
```

2. Create the admin-role account using the following commands.

```
username username role rolename password password
device(config)# username jsmith role admin password "uber#p@ssW0^b" encryption-level 10
```

Create a New User-Role User Account

Create a new user account and associate with a user role that can run all show and other basic CLI commands.

Before You Begin

You must have the admin role to create a new user account that has the user role.

About This Task

Be sure to observe the following guidelines as you create the user account:

Table 7: User-account guidelines

User-defined variables	Values
<i>username</i>	1-40 alphanumeric characters, including underscore and dot. Underscore as first character is not allowed.
<i>rolename</i>	Pre-defined role to be assigned to the user ('admin' and 'user' are the only supported roles).
<i>password</i>	Clear-text: 8-40 characters for plain-text password. Encrypted: 8-128 for hashed passwords.
<i>encryption-level</i>	0 = clear-text (default). 10 = encrypted.

Procedure

1. Run the **configure terminal** command to access Config mode.

The command line changes to configuration mode.

```
device(config)#
```

2. Create the user-role account using the following commands.

```
username username role rolename password password
device(config)# username jdoe role user password iKt1Sas*p
```



Remote Server Authentication

[Remote Server Authentication Overview](#) on page 15

[Login Authentication Mode](#) on page 15

[Conditions for Conformance](#) on page 16

[Set and Verify the Login Authentication Mode](#) on page 16

[Reset the Login Authentication Mode to the Default](#) on page 17

The NPB application supports two authentication sources to provide external Authentication and Accounting (AA) services for devices. Supported authentication sources are local and terminal access controller access-control system plus (TACACS+).

The following topics describe how to configure and manage remote-server authentication.

Remote Server Authentication Overview

Learn about remote server authentication for the application.

The NPB application supports external authentication and accounting (AA) services for accessing the 9920 device. Supported authentication is TACACS+.

Extreme recommends that you configure at least two remote AA servers to provide redundancy in the event of failure. For TACACS+, you can configure up to five external servers on the device. Each device maintains its own server configuration.

Login Authentication Mode

The login-authentication mode is defined as the order in which AA services are used on the device for user authentication during the login process. If AA login is not configured, authentication mode defaults to local authentication mode.

The NPB application supports two sources of authentication: primary and secondary. The secondary source of authentication is used in the event of primary source failure. AA login configuration is supported, with TACACS+ as primary and local-auth-fallback as secondary.

You can configure two possible sources for authentication to access the 9920 device:

- TACACS+ — Use an external TACACS+ server as the primary
- local-auth-fallback — Use the fallback local server as the secondary

If login fails through the primary source because of none of the configured servers not responding or because of login rejected by a server, failover occurs and authentication is done again through the secondary source (local).

By default, external AA services are disabled, and AA services default to the device-local user database. An environment requiring more than 64 users, including default and admin users, should adopt AA servers for user management.

Conditions for Conformance

Make sure your authentication server setup conforms to guidelines.

Consider the following conditions for remote server authentication:

- By default, the application authenticates with an internal local database.
- The source of authentication and the corresponding server type configuration are dependent on each other. Therefore, at least one server should be configured before that server type can be specified as a source.

Set and Verify the Login Authentication Mode

Learn how to set and verify the login authentication mode.

Before You Begin

- You must have an admin role to perform this task.
- The TACACS+ host must be configured on the 9920 device.

About This Task

The following procedure configures TACACS+ as the primary source of authentication and the local-auth-fallback as the secondary source. (For additional information, see [Client Configuration Parameters for TACACS+ Support](#) on page 19.) For complete information on login authentication mode, refer to the `aaa authentication` command in the *Extreme 9920 Software Command Reference, 21.1.0.0*.

Procedure

1. Run the `configure terminal` command to access Config mode.

The command line changes to configuration mode.

```
device(config)#
```

2. Run the `aaa authentication` command with the following parameters.

```
device# configure terminal
device(config)# aaa authentication login tacacs+ local-auth-fallback
device(config)# aaa accounting commands default start-stop tacacs+
device(config)# tacacs-server host 1.2.3.4
device(config-tacacs-config)# plain-key testing123
```

Authentication is attempted first with the TACACS+ server. If that fails, authentication is attempted with the local database.

3. Run the `show running-config aaa` command to display the configuration.

```
device(config-tacacs-config)# do show run
username testuser2 role user password $6$salt$cevuzTZ/QBjzuZG0/ebEeedmcTnhyM8ITUu8K032
  ➔ Cp2XvIibq7voqYagm18bwpLBqrg/1/16YxTmKKibJz5r10
tacacs-server host 1.2.3.4
  encrypted-key QjQkJLQUF3ncI1ooQCOaoEsBn5epVI3GsQwFD6i_BW
aaa authentication login tacacs+ local-auth-fallback
aaa accounting commands default start-stop tacacs+
interface ethernet 1/2
  shutdown
```



```
interface ethernet 2/2
 shutdown
```

4. Log in to the device using an account with TACACS+-only credentials to verify that TACACS+ is being used to authenticate the user.

Reset the Login Authentication Mode to the Default

Learn how to reset the login authentication mode to default.

About This Task

The following procedure resets the login configuration mode to the default value using the `no aaa authentication login` command.

Procedure

1. Run the **configure terminal** command to access Config mode.

The command line changes to configuration mode.

```
device(config)#
```

2. Run the following command to remove the configured authentication sequence and to restore the default value (local-only).

```
device(config)# no aaa authentication login tacacs+ local-auth-fallback
```

3. Log in to the device using an account with TACACS+-only credentials. The login should fail with an "access denied" error.
4. Log in to the device using an account with local-only credentials. The login should succeed.



TACACS+ Server Authentication

- [TACACS+ Authentication and Accounting on page 18](#)
- [Supported TACACS+ Packages and Protocols on page 18](#)
- [TACACS+ Configuration Components on page 19](#)
- [Client Configuration Parameters for TACACS+ Support on page 19](#)
- [Configure the Client to Use TACACS+ for Login Authentication on page 20](#)
- [Add a TACACS+ Server to the Client Server List on page 20](#)
- [Modify the Client-Side TACACS+ Server Configuration on page 21](#)
- [Remove a TACACS+ Server Key from the Client on page 22](#)
- [Remove Client-Side TACACS+ Server Configuration on page 23](#)

Terminal Access Controller Access-Control System Plus (TACACS+) is an AAA server protocol that uses a centralized authentication server and multiple network access servers or clients. With TACACS+ support, management of devices seamlessly integrates into network fabric environments. After a device is configured to use TACACS+, it becomes a TACACS+ client.

The following topics describe how to configure and manage TACACS+ server authentication.

TACACS+ Authentication and Accounting

The TACACS+ server is used for authentication and accounting on the application. You can access the device via SSH. The device goes through the same TACACS+ authentication process with either access method.



Note

For more information about configuring remote server authentication, see [Remote Server Authentication](#) on page 15. For complete information on login authentication mode, refer to the `aaa authentication` command in the *Extreme 9920 Software Command Reference, 21.1.0.0*.

Supported TACACS+ Packages and Protocols

The NPB application supports the following TACACS+ packages for running the TACACS+ daemon on remote AAA servers:

- Free TACACS+ daemon. You can download the latest package from www.shrubbery.net/tac_plus.
- ACS 5.3
- ACS 4.2

The TACACS+ protocol v1.78 is used for AAA services between the device client and the TACACS+ server.

Challenge Handshake Authentication Protocol (CHAP) authentication protocol is supported for user authentication.

TACACS+ Configuration Components

Configuring TACACS+ requires configuring TACACS+ support on the client and configuring TACACS+ on the server.

Client Configuration Parameters for TACACS+ Support

You must individually configure each client device to use TACACS+ servers. To configure the server IP address and key, use the **tacacs-server** command. You can configure a maximum of five TACACS+ servers on a device for AAA service.

The parameters in the following table are associated with a TACACS+ server that is configured on the device.

Table 8: TACACS+ server parameters

Parameter	Description
host	IP address (IPv4) or domain name or host name of the TACACS+ server. Host name requires prior DNS configuration. The maximum supported length for the host name is 40 characters.
port	The TCP port used to connect the TACACS+ server for authentication. The port range is 1 through 65535; the default port is 49 and is not configurable. Default value used.
protocol	The authentication protocol to be used and is not configurable. CHAP is used.
key	Specifies the configurable text string that is used as the shared secret between the device and the TACACS+ server to make the message exchange secure. The plain-text key must be between 1 and 40 characters in length and the encrypted key length must be less than or equal to 128 characters. Note: The value of key must match the value configured in the TACACS+ configuration file; otherwise, the communication between the server and the device fails.
retries	The number of attempts permitted to connect to a TACACS+ server. The range is 0 through 100, and the default value is 5. Not configurable. Default value is used.
timeout	The maximum amount of time to wait for a server to respond. Options are from 1 through 60 seconds, and the default value is 5 seconds. Not configurable. Default value is used.

Configure the Client to Use TACACS+ for Login Authentication

View the parameters to set the authentication mode so TACACS+ is primary.

After you configure the client-side TACACS+ server list, you must set the authentication mode so that TACACS+ is used as the primary source of authentication.

Full Syntax	[no] aaa authentication login tacacs+ local-auth- fallback
Parameter descriptions	<p>keyword no: Negate the command</p> <p>keyword aaa : Configure preferred order of types of AAA server (only TACACS+ is supported)</p> <p>keyword authentication: Configure preferred order for authentication</p> <p>Keyword login: Order of sources for login (default='local')</p> <p>Keyword tacacs+ : Use TACACS+ servers</p> <p>Keyword local-auth-fallback: Use local switch database if TACACS+ authentication methods are not active or authorization fails.</p>
Command modes	Configuration mode
Permissions & Validations	<ul style="list-style-type: none"> This command is allowed in configuration mode only. This command is available only to users with admin role.
Behavior description	By default, the local database is used for authentication. You can configure the application to authenticate users with the TACACS+ server as the primary method, with the local database as the fallback if TACACS+ is unavailable or authentication fails.

Add a TACACS+ Server to the Client Server List

Learn how to add a TACACS+ server to the client server list.

Before You Begin

You must have the admin role to perform this task.

About This Task

You add a TACACS+ server with an IPv4 address.



Note

When a list of servers is configured, failover from one server to another server happens only when a TACACS+ server fails to respond; it does not happen when user authentication fails.

The following procedure adds a TACACS+ server host in IPv4 format.

Procedure

1. Run the **configure terminal** command to access Config mode.

The command line changes to configuration mode.

```
device(config)#
```

2. Run the **tacacs-server host** command and specify the server IP address.

```
device(config)# tacacs-server host 10.2.3.5
```

After running the command, you are in TACACS server configuration mode, where you can configure the shared secret key.

3. Run the key command and type the shared secret string surrounded by quotation marks (either plain-text or encrypted string).

```
device(config-tacacs-config)# plain-key "new#hercules*secret*"
```

4. Type the end command to return to Exec mode and run the following command to verify the configuration.

```
device(config-tacacs-config)# end
```

5. Verify the configuration.

```
device# show running-config tacacs-server
tacacs-server host 10.2.3.5
  encrypted-key jahasjikjdoaskjuihuhiaoljsiaknkaiua=
```

Modify the Client-Side TACACS+ Server Configuration

Learn how to modify the client-side TACACS+ server configuration.

Before You Begin

You must have the admin role to perform this task.

Procedure

1. Display the configured server IP addresses.

```
device# show running-config tacacs-server
tacacs-server host 10.2.3.5
  encrypted-key "jahasjikjdoaskjuihuhiaoljsiaknkaiua="

tacacs-server host 1.2.3.4
  encrypted-key JMeYDVdBN4Vb-wx35d7HnXIE8BL9KLUCecePFwMNGo
```

2. Run the **configure terminal** command to access Config mode.

The command line changes to configuration mode.

```
device(config)#
```

3. Enter TACACS+ server configuration mode.

```
device(config)# tacacs-server host 10.2.3.5
device(config-tacacs-config)#
```

After running the command, you are in TACACS server configuration mode.

4. Specify the parameters that you want to modify. This example shows how to modify the shared secret key.

```
device(config-tacacs-config)# plain-key "changedsec"
```

5. Return to privileged EXEC mode.

```
device(config-tacacs-config)# end
```

6. Run the show running-config tacacs-server command to verify the configuration.

```
device# show running-config tacacs-server
tacacs-server host 10.2.3.5
  encrypted-key "jahasjikjdoaskjuihuhiaoljsiaknkaiua="
```

Remove a TACACS+ Server Key from the Client

Learn how to remove a configured TACACS+ server key from the client.

Before You Begin

You must have an admin role to perform this task.

Procedure

1. Display the configured server IP addresses and keys.

```
device# show running-config tacacs-server
tacacs-server host 10.2.3.5
    encrypted-key "jahasjikjdoaskjuihuhiaoljsiaknkaiua="

tacacs-server host 1.2.3.4
    encrypted-key JMeYDVdBN4Vb-wx35d7HnXIE8BL9KLUcEcePFwMNGo
```

2. Run the **configure terminal** command to access Config mode.

The command line changes to configuration mode.

```
device(config)#
```

3. Enter TACACS+ server configuration mode for the selected TACACS+ server.

```
device(config)# tacacs-server host ip-address
device(config-tacacs-config)#
```

After running the command, you are in TACACS server configuration mode.

4. Run the **no encrypted key** command to remove the key from the server.

```
device(config)# tacacs-server host ip-address
device(config-tacacs-config)# no encrypted-key
```

5. Return to privileged EXEC mode with the **end** command.

```
device(config-tacacs-config)# end
```

6. Run the **show running-config tacacs-server** command to verify the configuration.

```
device# show running-config tacacs-server
tacacs-server host host-address
```

Example

The following example removes the key from TACACS+ server on 10.2.3.5 and then verifies that a key is not configured on the specified server by running the **show running-config tacacs-server** command.

```
device# configure terminal
device(config)# tacacs-server host 10.2.3.5
device(config-tacacs-config)# no encrypted-key
device(config-tacacs-config)# end

device# show running-config tacacs-server
tacacs-server host 10.2.3.5

tacacs-server host 1.2.3.4
    encrypted-key JMeYDVdBN4Vb-wx35d7HnXIE8BL9KLUcEcePFwMNGo
```

Remove Client-Side TACACS+ Server Configuration

Learn how to remove TACACS+ server configuration from the client.

Before You Begin

You must have the admin role to perform this task.

Procedure

1. Display the configured server IP addresses and keys.

```
device# show running-config tacacs-server
tacacs-server host 10.2.3.5
    encrypted-key "jahasjikjdoaskjuihuhiaoljsiaknkaiua="

tacacs-server host 1.2.3.4
    encrypted-key JMeYDVdBN4Vb-wx35d7HnXIE8BL9KLUcEcePFwMNGo
```

2. Run the **configure terminal** command to access Config mode.

The command line changes to configuration mode.

```
device(config)#
```

3. Run the `no tacacs-server host` command to remove the TACACS+ configuration from the server.

```
device(config)# no tacacs-server host 10.2.3.5
```

4. Return to privileged EXEC mode with the **end** command.

```
device(config-tacacs-config)# end
```

5. Run the `show running-config tacacs-server` command to verify that the specified server does not appear in the list of TACACS+ servers configuration.

```
device# show running-config tacacs-server
tacacs-server host 1.2.3.4
    encrypted-key JMeYDVdBN4Vb-wx35d7HnXIE8BL9KLUcEcePFwMNGo
```

Example

The following example removes the TACACS+ configuration from the server on 10.2.3.5 and then verifies that it is no longer in the list of server hosts with the **show** command.

```
device# configure terminal
device(config)# no tacacs-server host 10.2.3.5
device(config)# end

device# show running-config tacacs-server
tacacs-server host 1.2.3.4
    encrypted-key JMeYDVdBN4Vb-wx35d7HnXIE8BL9KLUcEcePFwMNGo
```



HTTPS Certificates

[Import or Replace an HTTPS Certificate on page 24](#)

[Remove an Imported HTTPS Certificate on page 25](#)

Extreme 9920 software uses a TLS connection for incoming requests, using a default certificate. The following topics discuss HTTPS certificate management on the 9920 when a default certificate is not used.

Import or Replace an HTTPS Certificate

Learn how to import or replace an HTTPS certificate on the 9920 device.

Before You Begin

- You must have the admin role to perform this procedure.
- The HTTPS certificate file must be in PEM or PKCS format.

About This Task

You can use the following procedure to import or replace an HTTPS certificate on the ingress controller. Applications communicating with the 9920 device are secured with TLS. For additional security, a third-party certificate can replace the default certificates. The third-party certificate can be shared with client applications to validate the server. The IP address of the 9920 device should present in the SAN and the common name of third-party server certificates.



Note

If an IP address mismatch occurs between the 9920 device and the server certificate SAN IP, authentication will fail during TLS connection.

Procedure

Run the command `crypto import type https`.

```
device# crypto import type https protocol scp host <host address> certificate cert.pem  
key key.pem user <username> password <password>
```

```
Installing https certificate will result in a momentary delay and may affect active CLI  
connections - please be patient.
```

```
Successfully imported file: cert.pem
```

```
Successfully imported file: key.pem
```


Remove an Imported HTTPS Certificate

Learn how to remove the imported HTTPS certificate so the ingress controller reverts to using the self-signed certificate.

Before You Begin

- You must have the admin role to perform this procedure.

About This Task

Removes the imported certificate.

Now ingress controller should use the self signed certificate

You can shut down the HTTPS service without disabling HTTPS certificates. When the Apache web server boots, it enables the HTTPS service only in the presence of HTTPS crypto certificates.



Note

HTTPS certificates must be configured and enabled for web service to function on the device.

Procedure

- Delete the device certificate with the `no crypto import` command.

```
device# no crypto import type https
Deleting https certificate!
```

- Run the command `show crypto certificates` to verify HTTPS certificates are removed..

```
device# show crypto certificates
```



Token-Based Authentication

[Token-Based Authentication for gRPC Requests](#) on page 26

[Token-Based Authentication Guidelines and Limitations](#) on page 26

[Token-Based Authentication Flow](#) on page 27

the application supports token-based authentication, where a user provides credentials in the form of a username and password and receives a generated token that facilitates authentication for future access.

The following topics discuss token-based authentication and guidelines and limitations for use with the application.

Token-Based Authentication for gRPC Requests

Learn about using JWT bearer-token authentication for gRPC requests.

the application supports JSON Web token (JWT) token authentication for gRPC requests. The client accesses the RSA key-pair-signed token by presenting the credentials to an authentication API. When the token is stored on the client, it can send additional gRPC/HTTPS requests, with `Authorization: <type> <credentials>`, where the authorization type is Bearer followed by your JWT access token credentials, similar to the following example.

```
headers: {  
  Authorization: "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzd...G8p-_cD0"  
}
```

The authenticate request/response includes a long-lived refresh token, which can be used to get a new access-token when the previous access-token expires, as shown in the following code snippet.

```
service Auth {  
  rpc Authenticate (AuthenticateRequest) returns (AuthenticateResponse);  
  rpc GetAccessToken (RefreshToken) returns (AccessToken);  
}
```

For more information about implementing JWT token-based authentication, see the *the application YANG Reference Guide*.

Token-Based Authentication Guidelines and Limitations

Learn about how to implement JWT token-based authentication and its limitations.

When implementing token-based authentication, keep in mind the following guidelines and limitations.

- The access token lifetime is 24 hours. When it expires, a refresh token is used to fetch a new access token.

- The refresh token has a 30-day lifetime. When it expires, the user must reauthenticate and obtain a new access token and then a refresh token.
- The existing tokens become invalid in the following scenarios, and a user must reauthenticate and obtain a new access token.
 - Token expired.
 - Login-authentication method changed.
 - User account associated with the token deleted or blocked (local users only).
 - Changed user password (local users only).
 - Changed user role (local users only).

Token-Based Authentication Flow

The following steps describe the NPB application token-authentication process.

1. The client requests an access token from AuthServer, using the Authenticate() API method (from AuthClient) to pass user credentials.
2. AuthService issues the token in response, using the following process:
 - a. User credentials are validated with the AAA login mechanism configured on the device.
 - b. The JWT token is generated and includes role, expiry, and other relevant information.
 - c. AuthService signs the token with its private key and sends it as the response to Authenticate() API.
3. The client stores the response token, sending it with every gNMI/gNOI request with the token type and token credential in the Authorization header.
4. AuthService validates the token by performing the following steps:
 - a. Validates the signature with the public key from the cert store.
 - b. Validates the claims (expiry check, role validation, and any others).
 - c. Checks whether the role in the token has permission to access the requested resource.
5. If 4a, 4b, or 4c fails, the authentication fails, and the request response is an error message.



Remote System Logging

[Set Up a Remote Logging Server on page 28](#)

[Configure Remote Logging Server Storage on page 29](#)

[Configure Remote Logging to Use UDP on page 29](#)

[Configure Remote Logging to Use TCP on page 30](#)

[Install Certificates Required for TLS Encryption on page 31](#)

[Configure Remote Logging to Use TCP with TLS Encryption on page 31](#)

You can configure any Linux server that has the Rsyslog utility installed to accept system logs (syslogs) from Extreme 9920. For more information about installing Rsyslog, see https://www.rsyslog.com/doc/master/installation/install_from_source.html. For all supported system logging commands, see the *Extreme 9920 Software Command Reference, 21.1.0.0*

Extreme 9920 software supports the following transport protocols for remote system logs:

- UDP
- TCP
- TCP/TLS

Keep in mind the following limitations when configuring the application remote system logging:

- Log filtering is not supported. All syslog types are forwarded to the configured remote server.
- No log-level mapping.

The following topics discuss how to configure and maintain remote-system logging.

Set Up a Remote Logging Server

Learn how to verify or add a rule to the rsyslog.conf file to set up a remote logging server.

Before You Begin

The remote server must have the Rsyslog utility installed.

About This Task

You can configure any Linux server that has the Rsyslog utility installed to accept syslogs from the application.

Procedure

1. Navigate to the rsyslog.conf file and open it in your preferred text editor.

```
$ /etc/rsyslog.conf
```

2. Verify that the following rule is included in this file, or add it if it is missing.

```
$IncludeConfig /etc/rsyslog.d/*.conf
```

3. Save and close the rsyslog.conf file.

Configure Remote Logging Server Storage

Learn how to configure the remote logging server to store client log files in separate directories.

Before You Begin

The remote server must have the Rsyslog utility installed.

About This Task

By default, system logs are stored in the /var/log directory. But when receiving system logs from other machines, it is a best practice to store the syslogs from each client in separate directories.

Procedure

1. Create the following conf file.

```
$ /etc/rsyslog.d/directives.conf
```

2. Open the file directives.conf in your preferred text editor and add the following content.

```
$template RemoteLogs, "/var/log/%HOSTNAME%/%PROGRAMNAME%.log"  
*. * ?RemoteLogs  
& ~
```

The directives.conf file does the following:

- Creates the template RemoteLogs and applies it to all logs
 - Creates a log directory for each client with the local server's host name and stores log files with the syslog's service name from each sending device to the named directory.
 - Creates a directory with the local server host name and stores local syslogs to this location.
 - Appends logs to the files that already exist.
3. Save and close the directives.conf file.
 4. Run the following command to restart the rsyslog service and begin logging according to directives.conf.

```
$ sudo systemctl restart rsyslog
```

5. Run the following command to verify the rsyslog service status.

```
$ sudo systemctl status rsyslog
```

Configure Remote Logging to Use UDP

Learn how to configure the remote server for logging via UDP.

Before You Begin

The NPB application supports remote logging on Linux, Mac, or Windows operating systems, and the following commands are Linux-specific. Refer to the documentation for the Rsyslog utility for your operating system, as needed.

About This Task

You create a UDP-specific configuration file to enable UDP transport of syslog.

Procedure

1. At the command prompt, create and open the following file in your preferred text editor.

```
$ /etc/rsyslog.d/udp.conf
```

2. Copy and paste the following text into the udp.conf file, replacing the port number if needed with one you choose.

```
# load UDP listener
module(load="imudp")
# start listener at port 514
input(type="imudp" port="514")
```

3. Save and close udp.conf.
4. Run the following command to restart the rsyslog service.

```
$ sudo systemctl restart rsyslog
```

5. Run the following command to verify the rsyslog service status.

```
$ sudo systemctl status rsyslog
```

Configure Remote Logging to Use TCP

Learn how to configure the remote server for logging via TCP.

Before You Begin

The NPB application supports remote logging on Linux, Mac, or Windows operating systems, and the following commands are Linux-specific. Refer to the documentation for the Rsyslog utility for your operating system, as needed.

About This Task

You create a TCP-specific configuration file to enable UDP transport of syslog.

Procedure

1. At the command prompt, create and open the following file in your preferred text editor.

```
$ /etc/rsyslog.d/tcp.conf
```

2. Copy and paste the following text into the tcp.conf file, replacing the port number if needed with one you choose.

```
# load TCP listener
module(load="imtcp")
# start listener at port 514
input(type="imtcp" port="514")
```

3. Save and close tcp.conf.
4. Run the following command to restart the rsyslog service.

```
$ sudo systemctl restart rsyslog
```

5. Run the following command to verify the rsyslog service status.

```
$ sudo systemctl status rsyslog
```

Install Certificates Required for TLS Encryption

Learn how to install the three certificates required for using TLS encryption for remote logging.

Before You Begin

The NPB application supports remote logging on Linux, Mac, or Windows operating systems, and the following commands are Linux-specific. Refer to the documentation for the Rsyslog utility for your operating system, as needed.

About This Task

To optionally enable TLS encryption over TCP, you must generate and install three certificates on the remote logging server to enable TLS encryption over TCP. All three certificates are in PEM format:

- CA certificate
- Machine key certificate
- Machine key



Note

the application, the Rsyslog client that sends syslogs to the remote logging server, needs only the CA certificate that is in current use on the device.

Procedure

1. Generate the three required certificates, using the instructions provided at the following Rsyslog locations.
 - https://www.rsyslog.com/doc/v8-stable/tutorials/tls_cert_ca.html
 - https://www.rsyslog.com/doc/v8-stable/tutorials/tls_cert_machine.html
2. Use the **copy** command to copy the certificates to the preferred directory (default is `/etc/ssl/certs`).



Note

Note the filepath for each certificate, which is used to configure the remote logging server to use TLS encryption.

3. Run the **chmod** command to set file permissions to 0644 on each certificate.

Configure Remote Logging to Use TCP with TLS Encryption

Learn how to configure the remote server for logging via TCP using TLS encryption.

Before You Begin

Generate the certificates required to use TLS encryption and import them to the remote server, making sure they have the proper read permissions (0644). Make sure you have noted the filepaths to each certificate.

About This Task

You install an rsyslog utilities package and add content to `tcp.conf` on the remote server to enable TLS encryption over TCP.

Procedure

1. If not already installed, run the following command on the remote server to install the package `rsyslog-gnutls`.

```
$ sudo apt-get install rsyslog-gnutls
```

2. At the command prompt, create and open the following file in your preferred text editor.

```
$ /etc/rsyslog.d/tcp.conf
```

3. Copy and paste the following text into the `tcp.conf` file, making sure the certificate filepaths are correct and replacing the port number if needed with one you choose.

```
global (
  DefaultNetstreamDriver="gtls"
  DefaultNetstreamDriverCAFile="/path/to/ca-certificate/ca.pem"
  DefaultNetstreamDriverCertFile="/path/to/server-certificate/server-cert.pem"
  DefaultNetstreamDriverKeyFile="/path/to/server-key/server-key.pem"
)

# load TCP listener
module(
  load="imtcp"
  StreamDriver.Name="gtls"
  StreamDriver.Mode="1"
  StreamDriver.Authmode="anon"
)

# start up listener at port 514
input (
  type="imtcp"
  port="514"
```

4. Save and close `tcp.conf`.
5. Run the following command to restart the `rsyslog` service.

```
$ sudo systemctl restart rsyslog
```

6. Run the following command to verify the `rsyslog` service status.

```
$ sudo systemctl status rsyslog
```