Technical Guide

Secure Shell (SSH)

Feature Overview and Configuration Guide

Introduction

This guide describes how the Secure Shell protocol (SSH) is implemented in the AlliedWare Plus™ Operating System (OS).

It covers:

- support for Secure Shell
- configuring your device as a Secure Shell server and client
- using Secure Shell to manage your device
- an SSH server configuration example.

AlliedWare Plus supports the SSH protocol version 2.

Caution: SSH was upgraded in version 5.5.1-1.1, to increase security. The upgrade means that some older SSH clients may no longer connect to AlliedWare Plus devices running 5.5.1-1.1 or later. To resolve this, see "In version 5.5.1-1.1 or later, older SSH clients can't connect to AlliedWare Plus devices" on page 4.

Secure management is important in modern networks, as the ability to easily and effectively manage switches and routers, and the necessity for security, are two almost universal requirements.

Protocols such as Telnet and commands like UNIX's rlogin allow you to manage devices remotely, but can have serious security problems, such as relying on reusable clear text passwords that are vulnerable to wiretapping or password guessing. The Secure Shell protocol is superior to these access methods by providing encrypted and strongly authenticated remote login sessions.

SSH provides sessions between a host running a SSH server and a machine with a SSH client. AlliedWare Plus includes both a SSH server and a SSH client to enable you to securely—with the benefit of cryptographic authentication and encryption—manage your devices over an insecure network.

In summary, SSH:

- replaces Telnet for remote terminal sessions; SSH is strongly authenticated and encrypted.
- includes remote command execution, which allows you to send commands to a device securely and conveniently, without requiring a terminal session on the device.
- allows you to connect to another host from your AlliedWare Plus device.

AlliedWare Plus supports Secure Copy (SCP) and SSH File Transfer Protocol (SFTP). Both these protocols allow you to securely copy files between your device and remote machines. SFTP provides additional features from SCP, such as allowing you to manipulate the remote files, and halt or resume file transfers without closing the session.

Products and software version that apply to this guide

This guide applies to all AlliedWare Plus products, running software version **5.5.5-1.x** or later.

Screenshots in this guide were replicated with the versions below. Please use these Device GUI and software versions together if you wish to replicate this setup.

To access the latest SSH features, we recommend the following versions:

- Device GUI version **2.22.0** or later.
- AlliedWare Plus software version **5.5.5-2.x** or later.
- From version **5.4.7-0.1** onwards, if the SSH service is enabled on a device and that device detects that the host key is missing, the device generates a new host key automatically instead of terminating SSH.
- In version **5.4.9-2.1**, 3DES was removed from the supported cypher set for SSH. Modern clients and servers can continue to interoperate using AES-based cyphers transparently.
- In version **5.5.1-1.1**, support was removed for the ssh-rsa algorithm in OpenSSH and for SSH protocol v1.
- In version **5.5.2-0.1**, the RSA key length range changed to 1024-16384 (default is 2048).
- In version **5.5.2-0.1**, the ECDSA key size options became 256, 384 or 521 bits (default is 384).
- In version **5.5.2-1.1** the SSH server and client have been made VRF-aware.
- In version **5.5.2-2.1** Alongside the existing exec-mode SSH client commands, a new config-mode SSH client command has been added.

For more information, see the following documents:

- The product's Datasheet.
- The product's Command Reference.

These documents are available from the above links on our website at alliedtelesis.com.

Contents

Secure Shell (SSH) on AlliedWare Plus

Secure Shell (SSH) supports the following features:

- Inbound SSH connections (server mode) and outbound SSH connections (client mode).
- File loading to and from remote machines using Secure Copy, using either the SSH client or SSH server mode.
- Public keys:
  - RSA keys with lengths of 1024–16384 bits, and
  - ECDSA keys with key size of 256, 384 or 521 bits (default is 384).
  - Keys are stored in a format compatible with other SSH implementations, and mechanisms are provided to copy keys to and from your device.
- Secure encryption, such as AES.
- Remote non-interactive shell that allows arbitrary commands to be sent securely to your device, possibly automatically.
- Compression of Secure Shell traffic.
- Tunneling of TCP/IP traffic.
- File loading from remote machines using SSH File Transfer Protocol (SFTP).
- A login banner on the SSH server, that displays when SSHv2 clients connect to the server.

Feature support in Secure Mode

Secure Mode enhances security by disabling any algorithms that are not supported under FIPS (Federal Information Processing Standards). Secure Mode is available on a number of Allied Telesis switches.

For step-by-step instructions on enabling Secure Mode, see “How to Enable Secure Mode” in the Getting Started with AlliedWare Plus Feature Overview and Configuration Guide.

In version 5.5.1-1.1 or later, older SSH clients can’t connect to AlliedWare Plus devices

In AlliedWare Plus version 5.5.1-1.1, OpenSSH was upgraded. This means versions 5.5.1-1.1 onwards no longer support the following insecure options:

- the ssh-rsa algorithm in OpenSSH, which is based on SHA1
- SSH protocol version 1.

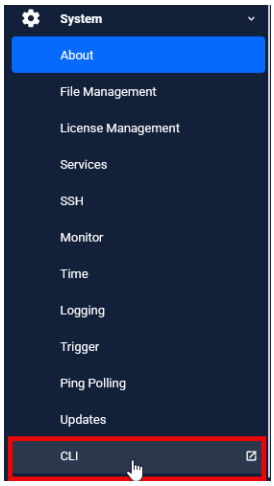
Unfortunately, some older SSH clients and older libraries still expect to use ssh-rsa and may not be able to connect to a device running 5.5.1-1.1 or later.

From version 5.5.1-1.3 onwards, AlliedWare Plus devices automatically create an ECDSA key when the SSH service is enabled, if an ECDSA key doesn't already exist. This makes it possible for many of these older SSH clients to connect to AlliedWare Plus devices securely.

Therefore, if you need to upgrade, we recommend you:

- ensure your SSH client is up to date, and
- if upgrading to 5.5.1-1.1 or 5.5.1-1.2, create an ECDSA key for the server to use, in case the client does not support secure SSH RSA algorithms.

To **create the ECDSA key**, use the following steps:



1. Access the CLI of the AlliedWare Plus device. If you have already upgraded and can no longer use your SSH client, you can access the device through its console port, or through its GUI as shown in this screenshot.
2. Create an ECDSA key using the commands:

```
awplus# configure terminal
awplus(config)# crypto key generate hostkey ecdsa 384
```

3. Either reboot the device, or turn the SSH service off and on again, using the commands:

```
awplus(config)# no service ssh
awplus(config)# service ssh
```

Note that you only need to do this procedure on existing AlliedWare Plus devices. From 5.5.1-1.1 onwards, AlliedWare Plus automatically creates an ECDSA key on factory-new devices and devices that have been returned to a factory state.

Configuring the SSH Server

This section provides instructions on:

From version **Device GUI 2.16.0** onwards, you can configure SSH and allow or deny specific username or hostname patterns for SSH login from the Device GUI.

Note: Versions before 2.16.0 only let you enable or disable SSH. We recommend you update your devices to version 2.16.0 onwards to access extra features.

Creating a host key

The SSH server uses either an RSA or ECDSA host key to authenticate itself with SSH clients. Once created, the host key is stored securely on the device.

When you enable the SSH server, if no host keys exist, the server automatically generates SSHv2 host key pairs using ECDSA with a curve length of 384, and RSA with a 2048-bit key (unless in secure mode, when it only generates the ECDSA key).

If you need different keys, you can create them as follows.

- To generate an **RSA** host key for the SSH server, use the command:

```
awplus(config)#crypto key generate hostkey rsa [<1024-16384>]
```

From version 5.5.2-0.1 onwards, the default RSA key length is 2048. In earlier versions, it is 1024.

- To generate an **ECDSA** host key for the SSH server, use the command:

```
awplus(config)#crypto key generate hostkey ecdsa [256|384|521]
```

From version 5.5.2-0.1 onwards, the default ECDSA key size is 384. In earlier versions, it is 256.

- To **destroy** a host key, use the command:

```
awplus(config)#crypto key destroy hostkey {rsa|ecdsa}
```

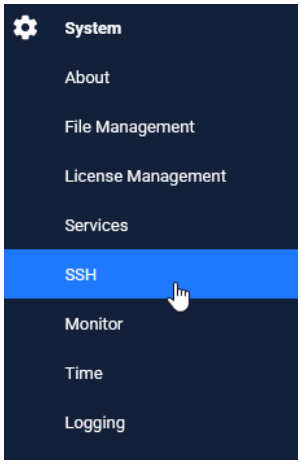
- To **display** the public part of a host key stored on your device, use the command:

```
awplus#show crypto key hostkey [rsa|ecdsa]
```

Enabling the server

You must enable the SSH server before connections from SSH, SCP, and SFTP clients are accepted. When the SSH server is disabled it rejects connections from SSH clients. The SSH server is disabled by default on most AlliedWare Plus devices.

Enabling SSH from the Device GUI



To enable SSH from the Device GUI, click on the **System** tab on the left-hand menu, and select **SSH**. The SSH page opens.

From version 2.5.0 onwards, support for enabling SSH is available, but is accessible from the Services page instead of the System page.

We recommend you update to **2.18.0** or later to access the latest security features from the **System** > **SSH** tab.

To enable the SSH server from the Device GUI, click the **Toggle** on the top right.

To see info about allowing and denying users, see "Using the Device GUI to allow SSH users" on page 11 and "Using the Device GUI to deny SSH users" on page 13.

Enabling SSH from the CLI

When enabled, the SSH server allows SCP and SFTP sessions by default.

- To enable the SSH server, use the command:

```
awplus(config)#service ssh [ip|ipv6]
```

- To disable the SSH server, use the command:

```
awplus(config)#no service ssh [ip|ipv6]
```

- To disable these services, use the commands:

```
awplus(config)#no ssh server scp
awplus(config)#no ssh server sftp
```

This allows you to reject SCP or SFTP file transfer requests, while still allowing Secure Shell connections.

- To re-enable SCP and SFTP services, use the commands:

```
awplus(config)#ssh server scp
awplus(config)#ssh server sftp
```

Modifying the server

- To modify the SSH protocol version that the server supports, use the command:

```
awplus(config)#ssh server
```

From version 5.5.1-1.1 onwards, only SSH protocol version 2 is supported.

- To modify the TCP port that the server listens to for incoming sessions, use the command:

```
awplus(config)#ssh server <1-65535>
```

By default, the server listens on port 22 for incoming sessions.

- To modify the number of unauthenticated connections the server allows, use the command:

```
awplus(config)#ssh server max-startups <1-128>
```

The SSH server only allows only 10 unauthenticated SSH sessions at any point in time, by default.

- To modify session and login timeouts on the SSH server, use the command:

```
awplus(config)#ssh server [session-timeout <0-3600>]
[login-timeout <1-600>]
```

By default, the SSH server waits 60 seconds for a client to authenticate itself. You can alter this waiting time by using the **login-timeout** parameter. If the client is still not authenticated after the timeout, then the SSH server disconnects the session.

Once a client has authenticated, the SSH session does not time out, by default. Use the **session-timeout** parameter to set a maximum time period the server waits before deciding that a session is inactive and terminating it.

For example:

- To set the session timeout to 600 seconds, the login timeout to 30 seconds, and the maximum number of concurrent unauthenticated sessions to 5, use the command:

```
awplus(config)#ssh server session-timeout 600 login-timeout 30
max-startups 5
```

- To remove the configured timeouts and maximum startups, use the command:

```
awplus(config)#no ssh server session-timeout login-timeout max-startups
```

The SSH server and client have been made VRF-aware, and new configuration has been added to enable this functionality.

For example:

- To isolate the SSH server and make it operate within the VRF named ‘red’, use the following command:

```
awplus(config)#ssh server vrf red
```

- To return the SSH server to the global VRF use the following command:

```
awplus(config)#no ssh server vrf
```

By default SSH, and other services, operate within the global VRF.

Validating the server configuration

- To validate the SSH server configuration, use the commands:

```
awplus#show running-config ssh
```

or

```
awplus#show ssh server
```

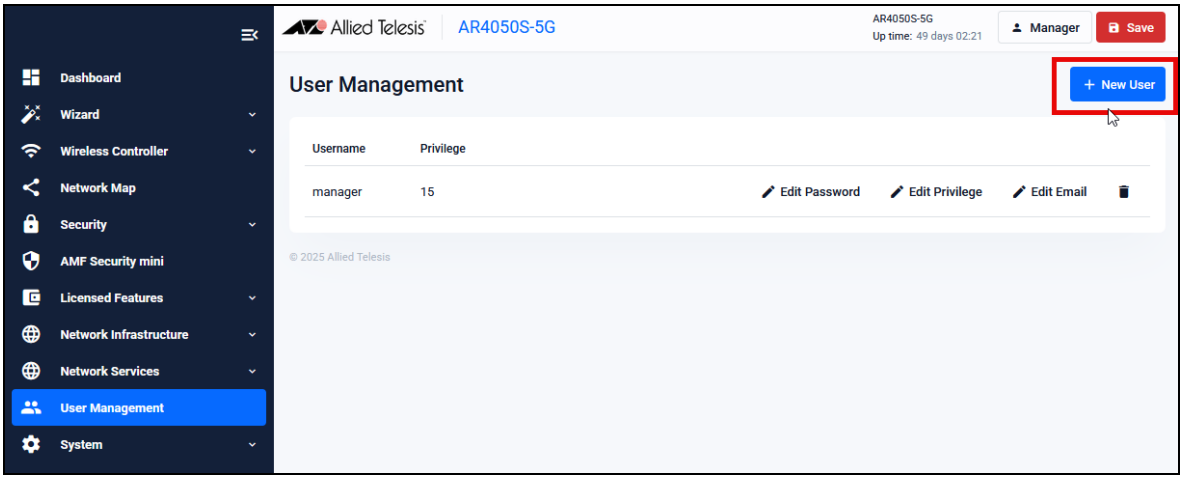
```
awplus#show ssh server
Secure Shell Server Configuration
-----------------------------------------------------------------------------
SSH Server                          : Enabled
Protocol                            : IPv4,IPv6
Port                                : 22
Version                             : 2
VRF                                 : red
Services                            : scp, sftp
User Authentication                 : publickey, password
Resolve Hosts                       : Disabled
Session Timeout                     : 0 (Off)
Login Timeout                       : 60 seconds
Maximum Authentication Tries        : 6
Maximum Startups                    : 10
Debug                               : NONE
Ciphers                             : aes128-cbc,aes128-ctr,aes192-ctr,aes256-ctr
KEX                                 : curve25519-sha256@libssh.org,
                                      ecdh-sha2-nistp256,ecdh-sha2-nistp384,
                                      ecdh-sha2-nistp521,
                                      diffie-hellman-group-exchange-sha256,
                                      diffie-hellman-group-exchange-sha1,
                                      diffie-hellman-group14-sha1
```

Registering SSH Users

The SSH server requires you to register SSH users before you can allow or deny them. Users that are not registered cannot access the SSH server. Ensure first that you have defined the user in the Authorized User Database of your device.

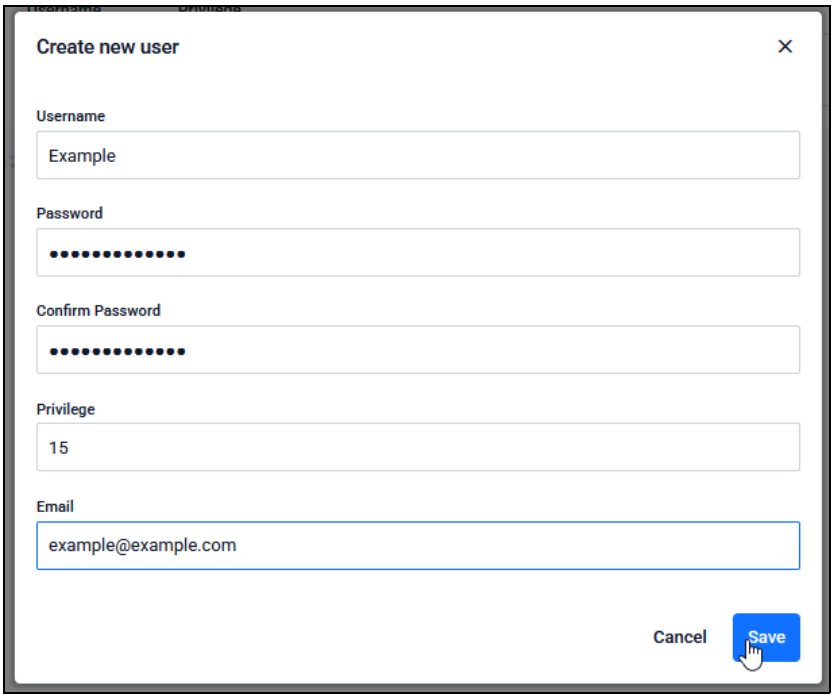
You can add a new user in the Device GUI, or the CLI in the following ways:

- To add a new user in the Device GUI, go to the User Management page and click **+ New User**.



- Enter the user information in the **Create new user** dialog.

Note: Users with a privilege setting of less than 15 will not be allowed access to the device GUI.



- **Click Save** to add the new user to the list.

To add a new user in the CLI, use the command:

```
awplus(config)#username <username> privilege <1-15> password <password>
```

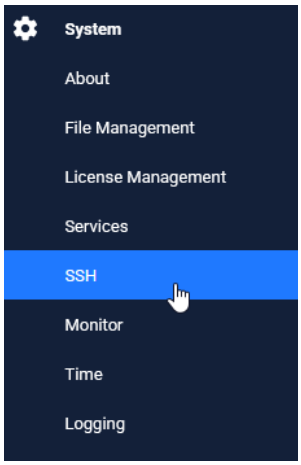
Using the Device GUI to allow SSH users

This section covers allowing users to make SSH connections to a device from the Device GUI or the CLI.

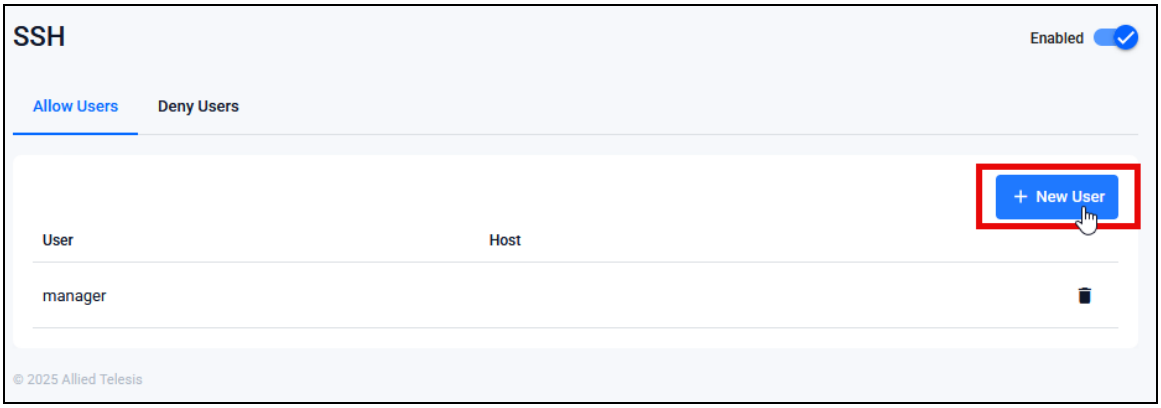
Note that you must first register users from the User Management page in order to allow or deny SSH access. See the following sections for more information:

- To see how to register users, see "Registering SSH Users" on page 10.
- To skip to the CLI information, see "Using the CLI to allow SSH users" on page 12.
- To see how to deny users, see "Using the Device GUI to deny SSH users" on page 13.

1. Use the left-hand menu to navigate to the **System** > **SSH** page.



2. Select the Allow Users tab on the table.



3. Click **+ New User.**
4. In the **New User** popup box, you can enter the username in the **New User** field.
5. You may choose to add an optional host for identification in the **New Host** field.

   The hostname pattern in the New Host field matches against the remote host via SSH. If specified, the server allows the user to connect only from hosts matching the pattern. The hostname may be an IP address or a domain name.

   You can also use an asterisk, which acts as a wildcard character that matches any string of characters. For example:

- to allow all users whose name ends in CompanyName, enter a username of *CompanyName
- to allow all users from the IP range 192.168.1.1 to 192.168.1.255, enter a username of * and a host of 192.168.1.*

6. Click **Apply** to save changes.

Using the CLI to allow SSH users

The SSH server requires you to register SSH users. Users that are not registered cannot access the SSH server. Ensure first that you have defined the user in the Authorized User Database of your device.

- To add a new user, use the command:

```
awplus(config)#username <username> privilege <1-15> password <password>
```

- To allow a user with the SSH server, use the command:

```
awplus(config)#ssh server allow-users <username-pattern> [<hostname-pattern>]
```

Registered entries can contain just the username, or the username with some host details, such as an IP address range. Additionally you can specify a range of users or hostname details by using an asterisk to match any string of characters.

For example:

- To allow any user from the IP range 192.168.1.1 to 192.168.1.255, use the command:

```
awplus(config)#ssh server allow-users * 192.168.1.*
```

- To display the list of allowed users, use the command:

```
awplus#show ssh server allow-users
```

- To delete an entry from the list of allowed users, use the command:

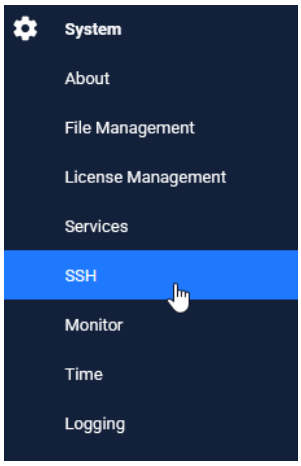
```
awplus(config)#no ssh server allow-users <username-pattern> [<hostname-pattern>]
```

Using the Device GUI to deny SSH users

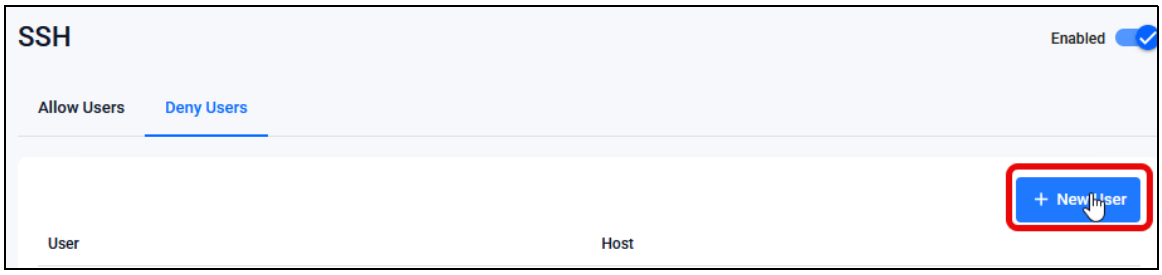
The SSH server also contains a list of **denied** users. The server checks all incoming sessions against this list and denies any matching session, regardless of whether the session matches an entry in the allowed users list. From version 2.16.0 onwards, you can allow or deny users from the Device GUI.

- To see how to register users, see "Registering SSH Users" on page 10.
- To skip to the CLI information, see "Using the CLI to deny SSH users" on page 14.
- To see how to allow users, see "Using the Device GUI to allow SSH users" on page 11.

1. Use the left-hand menu to navigate to the **System** > **SSH** page.



2. To deny a user SSH access, click on the **Deny Users** tab of the table.

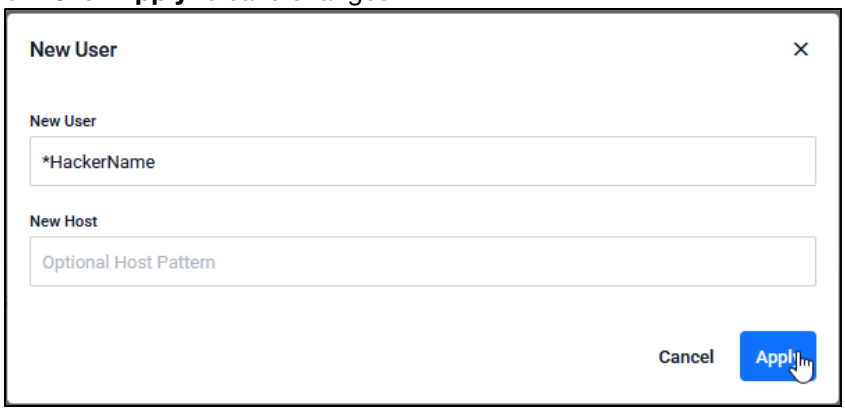


3. Click **+ New User.**
4. In the **New User** popup box, you can enter the username and optional host for identification.

   You may choose to use a wildcard character to specify a range of users or host details. The hostname may be an IP address or a domain name. For example:

   - to deny all users whose name ends in HackerName, enter a username of *HackerName
   - to deny all users from the IP range 192.168.1.1 to 192.168.1.255, enter a username of * and a host of 192.168.1.*

5. Click **Apply** to save changes.



Using the CLI to deny SSH users

- To add an entry to the list of denied users, use the command:

```
awplus(config)#ssh server deny-users <username-pattern> [<hostname-pattern>]
```

This allows you to deny specific users from a range of allowed users.

For example:

- To deny a user with the IP address 192.168.1.12, use the command:

```
awplus(config)#ssh server deny-users * 192.168.1.12
```

- To display the database of denied users, use the command:

```
awplus#show ssh server deny-users
```

- To delete a client from the database of denied users, use the command:

```
awplus(config)#no ssh server deny-users <username-pattern> [<hostname-pattern>]
```

Authenticating SSH users

SSH users can use either their password or public key authentication to authenticate themselves with the SSH server. To use public key authentication, copy the user's public key file from their client device to the SSH server. To associate the key with a user, use the command:

```
awplus(config)#crypto key pubkey-chain userkey <username> [<filename>]
```

For example:

- To associate the file key.pub with the user "langley", use the command:

```
awplus(config)#crypto key pubkey-chain userkey langley key.pub
```

- To add a key as text into the terminal for user "geoff", first enter the command:

```
awplus(config)#crypto key pubkey-chain userkey geoff
```

then paste or type the key in as text. You can add multiple keys for the same user.

- To display the list of public keys associated with a user, use the command:

```
awplus#show crypto key pubkey-chain userkey <username> [<1-65535>]
```

The <1-65535> parameter allows you to display an individual key.

- To delete a key associated with a user from your device, use the command:

```
awplus(config)#no crypto key pubkey-chain userkey <username> <1-65535>
```

Forcing the server to only use secure algorithms

You can increase security by configuring the SSH server to only negotiate algorithms that are seen as current best practice. For example, this will prevent the server from using CBC ciphers, which are now regarded as weak. You can do this for ciphers, key exchange, Message Authentication Code (MAC) algorithms, or all 3 of these. To do this, use the commands:

```
awplus# configure terminal
```

For ciphers:

```
awplus(config)# ssh server secure-ciphers
```

For key exchange:

```
awplus(config)# ssh server secure-kex
```

For MAC:

```
awplus(config)# ssh server secure-mac
```

For all:

```
awplus(config)# ssh server secure-algs
```

Note that these commands are not available in Secure Mode, because Secure Mode already automatically limits the device to using only FIPS-approved algorithms.

Adding a login banner

You can add a login banner to the SSH server for sessions.

The server displays the banner to clients before the login prompt.

- To set the login banner's message, use the command:

```
awplus(config)#banner login
```

then enter your message and use Ctrl+D to finish.

- To view the configured login banner, use the command:

```
awplus#show banner login
```

- To remove the configured message for the login banner, use the command:

```
awplus(config)#no banner login
```

Monitoring the server and managing sessions

- To display the current status of the SSH server, use the command:

```
awplus#show ssh server
```

- To display the current status of SSH sessions on your device, use the command:

```
awplus#show ssh
```

Note that this displays both SSH server and SSH client sessions that your Allied Telesis device is running. Use this command to view the unique identification number assigned to each incoming or outgoing SSH session. You need the ID number when terminating a specific session from your device.

- To terminate a session, or all sessions, use the command:

```
awplus#clear ssh {<1-65535>|all}
```

Creating host keys automatically when replacing devices

You can replace a failed device and copy the old device's configuration onto the replacement device, making it easier to remotely access the replacement device. This is possible because when you enable the SSH server, if no host keys exist, the server automatically generates keys.

So, if you need to replace a device and copy its existing configuration file, use the following steps:

1. Make sure that the new device is in a factory-clean state. If necessary, use the **erase factory-default** command to achieve this
2. Copy the firmware and configuration file from the old device to the Flash file system of the new device.
3. Set the copied files as the boot firmware and configuration files
4. Reboot the new device.

Because the host keys are new on the device, if a remote user tries to connect to the new device with existing SSH credentials, the SSH client will notice that the host keys for the device are different and may give a warning. The warning will include a selection option to replace the old host key, or instructions on how to do this. Follow the client's selection option or instructions.

For example, a Linux client displays the following warning:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle
attack)!
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
55:7d:82:00:7e:6f:ac:ac:de:1c:f1:53:08:51:1c:68.
Please contact your system administrator.
Add correct host key in /Users/fergus/.ssh/known_hosts to get rid of
this message.
Offending RSA key in /Users/fergus/.ssh/known_hosts:12
RSA host key for 192.168.1.1 has changed and you have requested
strict checking.
Host key verification failed.
```

Debugging the server

Information which may be useful for troubleshooting the SSH server is available using the SSH debugging function. You can enable server debugging while the SSH server is functioning.

To enable server debugging, use the command:

```
awplus#debug ssh server [brief|full]
```

To disable SSH server debugging, use the command:

```
awplus#no debug ssh server
```

Configuring the SSH Client

This section provides instructions on:

Modifying the client

You can configure a selection of variables when using the SSH client. Note that the following configuration commands apply only to client sessions initiated after the command. The configured settings are not saved; after you have logged out from the SSH client, the client returns to using the default settings.

Use the command:

```
awplus#ssh client {port <1-65535>|session-timeout <0-3600>|connect-timeout
<1-600>|vrf <vrf-name>}
```

The SSH client uses TCP port 22, by default. You can change the TCP port for the remote SSH server by using the **port** parameter.

The client terminates sessions that are not established after 30 seconds, by default. You can change this time period by using the **session-timeout** parameter.

Once the client has authenticated with a server, the client does not time out the SSH session, by default. Use the **connect-timeout** parameter to set a maximum time period the client waits before deciding that a session is inactive and terminating the session.

- To modify the SSH client so that it uses port 2000 for sessions, use the command:

  ```
  awplus#ssh client port 2000
  ```

- To modify the SSH client so that unestablished sessions time out after 60 seconds, and inactive connection time out after 100 seconds, use the command:

  ```
  awplus#ssh client session-timeout 60 connect-timeout 100
  ```

- To remove the configured port, session timeout, and connection timeout settings, use the command:

  ```
  awplus#no ssh client port session-timeout connect-timeout
  ```

The SSH server and client have been made VRF-aware, and new configuration has been added to enable this functionality.

- To make the client operate within named VRF ‘red’ for the session, use the following command:

```
awplus#ssh client vrf red
```

- To return the client to operating within the default VRF for the session, use the following command:

```
awplus#no ssh client vrf
```

Configuring the SSH client session with a VRF will enable scp and sftp to copy files to and from a remote host residing within said VRF.

Alongside the existing exec-mode SSH client commands, a new config-mode SSH client command has been added.

You can use the following SSH client config-mode command to specify a global VRF for all new login sessions. This will become part of the devices configuration and persist across reboots.

```
awplus(config)##ssh client vrf <vrf-name>
```

The way this config-mode command interacts with the existing exec-mode command is as follows:

- The config-mode command will configure a global SSH Client VRF, that will be used by default for new login sessions.
- Changes to the global SSH Client VRF will only affect new login sessions. The change will not be reflected in existing sessions.
- Individual login sessions may override the global SSH Client VRF through use of the existing exec-mode **ssh client vrf** commands.
- The exec-mode **no ssh client vrf** command will set the session to use the default VRF, instead of reverting back to the global configured one.

Configure the SSH client to use VRF 'red' for the current login session. Subsequent sessions will continue to use the default VRF. Use the command:

```
awplus#ssh client vrf red
```

Configure the SSH client to use VRF 'blue' for all subsequent login sessions, excluding the current one. Use the commands:

```
awplus#configure terminal
awplus(conf)#ssh client vrf blue
```

Configure the SSH client to use the default VRF for the current login session. Subsequent session with continue to use VRF 'blue'. Use the command:

```
awplus#no ssh client vrf
```

Configure the SSH client to use the default VRF for all subsequent login sessions, excluding the current one. Use the commands:

```
awplus#configure terminal
awplus(conf)#no ssh client vrf
```

The output of the **show ssh client** command, showing the VRF being used for this session:

```
awplus#show ssh client
Secure Shell Client Configuration
-----------------------------------------------------------
Port                                : 22
Version                             : 2
Connect Timeout                     : 30 seconds
Session Timeout                     : 0 (Off)
VRF                                 : red
Debug                               : NONE
```

Adding SSH servers

SSH servers identify themselves using a host key (see "Creating a host key" on page 6). Before the SSH client establishes a session with a SSH server, it confirms that the host key sent by the server matches its database entry for the server. If the database does not contains a host key for the server, then the SSH client requires you to confirm that the host key sent from the server is correct.

- To add an SSH server to the client's database, use the commands:

```
awplus(config)#crypto key pubkey-chain knownhosts [ip|ipv6] <hostname>
[rsa|ecdsa]
awplus(config)#crypto key pubkey-chain knownhosts [vrf <vrf-name>] [ip|
ipv6] <hostname> [rsa|ecdsa]
```

- To display the SSH servers in the client's database, use the commands:

```
awplus(config)#show crypto key pubkey-chain knownhosts [<1-65535>]
awplus(config)#show crypto key pubkey-chain knownhosts [vrf <vrf-name>|
global] [<1-65535>]
```

- To remove an entry in the database, use the commands:

```
awplus(config)#no crypto key pubkey-chain knownhosts <1-65535>
awplus(config)#no crypto key pubkey-chain knownhosts [vrf <vrf-name>]
<1-65535>
```

Authenticating with a server

You can authenticate your session with a server by either using a password, or using RSA, DSA or ECDSA public key authentication. To use public key authentication, you must generate a pair of keys, one private and one public, and copy the public key onto the SSH server.

- To generate an **RSA** set of private and public keys for an SSH user, use the command:

```
awplus(config)#crypto key generate userkey <username> rsa
[<1024-16384>]
```

From version 5.5.2-0.1 onwards, the default RSA key length is 2048. In earlier versions, it is 1024.

- To generate an **ECDSA** set of private and public keys for an SSH user, use the command:

```
awplus(config)#crypto key generate userkey <username> ecdsa [256|384|
521]
```

From version 5.5.2-0.1 onwards, the default ECDSA key size is 384. In earlier versions, it is 256.

You can generate one key of each encryption type per user on your client.

- To copy the public key onto the SSH server, you must display the key onscreen. To display the public key associated with a user, use the command:

```
awplus#show crypto key userkey <username> [rsa|ecdsa]
```

To display the public keys set for other users, you must specify their username. Only users with the highest privilege setting can use this command to view the keys of other users.

- To delete a public and private pair of keys associated with a user, use the command:

```
awplus(config)#crypto key destroy userkey <username> {rsa|ecdsa}
```

Connecting using SSH

- To connect to a remote device that is acting as an SSH server, use the command:

```
awplus#ssh <hostname>
```

The <***hostname***> parameter specifies the server and can be either an IP address or a host name.

You can also optionally specify other parameters when connecting, including the VRF instance, to use IPv6, the user or port number to connect on, and a command to execute on the server.

The command is:

```
awplus#ssh [vrf <vrf-name>] [ip|ipv6] [user <username>]|[port
<1-65535>] <hostname> [<command>]
```

For example:

- to connect to the SSH server at 192.168.1.2 as user ‘john’, and execute the command **show sys**, use the command:

```
awplus(config)#ssh user john 192.168.1.2 “show sys”
```

Note that you can only specify one of user or port. To change the default port, use the command **ssh client**.

Copying files to and from the server

You can use either the SCP or SFTP client to transfer files from a remote SSH server.

Use the command:

```
awplus#copy <source-url> <destination-url>
```

For example:

- to use SFTP to load a file from the SSH server 192.168.1.2, onto the Flash memory of your device, use the command:

```
awplus#copy sftp://192.168.1.2/key.pub flash
```

Debugging the client

Information which may be useful for troubleshooting the SSH client is available using the SSH debugging function. You can enable client debugging while the SSH client is functioning, using the command:

```
awplus#debug ssh client [brief|full]
```

- To disable SSH client debugging, use the command:

```
awplus#no debug ssh client
```

SSH Server Configuration Examples

Using public key authentication

This section provides a Secure Shell server configuration example, where:

- the SSH server uses ECDSA encryption
- three SSH users are configured: Manager, John, and Asuka. “manager” can connect from only a defined range of hosts, while “john” and “asuka” can SSH from all hosts
- the SSH users use ECDSA private and public key authentication, using keys generated by the client device.

This example shows how to create RSA encryption keys, configure the Secure Shell server, and register users to make Secure Shell connections to your device.

Step 1: **Login as a highest Privileged User**

To create the keys and add users, you must login as a privileged user.

Step 2: **Create encryption keys**

On new devices, keys will be created automatically when you start the SSH service. Or you can create one, using the commands:

```
awplus#configure terminal
awplus(config)#crypto key generate hostkey ecdsa 384
awplus(config)#exit
```

This creates a key with a size of 384. To verify the key creation, use the command:

```
awplus#show crypto key hostkey
```

Step 3: **Enable the Secure Shell server**

Enable Secure Shell on the device using the commands:

```
awplus#configure terminal
awplus(config)#service ssh
```

Modify the SSH server settings as desired.

For example, to set the login-timeout to 60, and the session-timeout to 3600, use the commands:

```
awplus(config)#ssh server session-timeout 3600 login-timeout 60
```

To verify the server configuration, use the command:

```
awplus#show ssh
```

Step 4: **Create SSH users**

In order to connect and execute commands, you must register users in the SSH user database, and in the User Authentication Database of the device.

To create the users **john** and **asuka** in the User Authentication Database, use the commands:

```
awplus#configure terminal
awplus(config)#username john privilege 15 password secret
awplus(config)#username asuka privilege 15 password very-secret
```

To register **john** and **asuka** as SSH clients, use the commands:

```
awplus(config)#ssh server allow-users john
awplus(config)#ssh server allow-users asuka
```

To register **manager** as an SSH client so that can only connect from the IP address 192.168.1.1, use the command:

```
awplus(config)#ssh server allow-users manager 192.168.1.1
```

Step 5: **Set up authentication**

SSH users cannot connect unless the server can authenticate them. There are two ways to authenticate an SSH session:

- password authentication, and
- private/public key authentication.

When using password authentication, the user must supply their User Authentication Database password.

To use private/public key authentication, copy the public keys for each user onto the device. To copy the files onto Flash from the key directory of an attached TFTP server, use the commands:

```
awplus#copy tftp://key/john.pub flash:/john.pub
awplus#copy tftp://key/asuka.pub flash:/asuka.pub
```

To associate the key file with each user, use the commands:

```
awplus#configure terminal
awplus(config)#crypto key pubkey-chain userkey john john.pub
awplus(config)#crypto key pubkey-chain userkey asuka asuka.pub
awplus(config)#crypto key pubkey-chain userkey manager manager.pub
```

Configure VRF and associate an SSH server with VRF

Use the following commands to configure a VRF:

```
awplus#configure terminal
awplus(conf)#ip vrf red 1
```

Assign a VLAN address with VRF:

```
awplus#configure terminal
awplus(conf)#interface vlan1
awplus(conf-if)#ip vrf forwarding red
awplus(conf-if)#ip address 10.0.0.1/24
```

Use the following commands to configure an SSH server with VRF ‘red’:

```
awplus#configure terminal
awplus(conf)#ssh server vrf red
awplus(conf)#service ssh
```

C613-22051-00 REV K

Allied Telesis NETWORK SMARTER

North America Headquarters | 19800 North Creek Parkway | Suite 100 | Bothell | WA 98011 | USA | T: +1 800 424 4284 | F: +1 425 481 3895
Asia-Pacific Headquarters | 11 Tai Seng Link | Singapore | 534182 | T: +65 6383 3832 | F: +65 6383 3830
EMEA & CSA Operations | Incheonweg 7 | 1437 EK Rozenburg | The Netherlands | T: +31 20 7950020 | F: +31 20 7950021

alliedtelesis.com