RADIUS Login Authentication With FreeRADIUS and Google Authenticator

For **Moxa EDS-(G)4000 Series** and **NPort -G2 Series**

Moxa Technical Support Team

support@moxa.com

Contents

 Released on Jun 23, 2026

About Moxa

Moxa is a leading provider of edge connectivity, industrial computing, and network infrastructure solutions for enabling connectivity for the Industrial Internet of Things. With over 35 years of industry experience, Moxa has connected more than 111 million devices worldwide and has a distribution and service network that reaches customers in more than 91 countries. Moxa delivers lasting business value by empowering industry with reliable networks and sincere service for industrial communications infrastructures. Information about Moxa's solutions is available at www.moxa.com.

How to Contact Moxa

Tel: 1-714-528-6777
Fax: 1-714-528-6778

1 Purpose and Scope

This step-by-step guide describes how to configure a FreeRADIUS server on Ubuntu 24.04 LTS Server for RADIUS login authentication with two-factor authentication (2FA) using Google Authenticator. The guide covers the Linux and FreeRADIUS configuration, Google Authenticator setup, and the RADIUS login configuration for the Moxa EDS-(G)4000 Series switches and the NPort 6150-G2-T.

2 Test Bench Environment

The following environment was used to validate the configuration.

Table 1: Validated test bench

Component	Version/Build
EDS-G4008	Firmware Version: v4.2 Build 2026_0209_1745
NPort 6150-G2-T	Firmware Version: v1.2.0 Build 26042302
Ubuntu Server	Ubuntu 24.04.4 LTS; Linux kernel 6.8.0-110-generic
FreeRADIUS	Version 3.2.5

3 Prerequisites

Before starting, make sure that:

- The Ubuntu server can reach the Moxa device management network.
- The Moxa device can reach the FreeRADIUS server through the management VLAN.
- You have sudo access on the Ubuntu server.
- The Google Authenticator app is available on the administrator's mobile phone.
- Time synchronization is enabled on the Ubuntu server. OTP validation is time-sensitive.

IMPORTANT

This deployment uses RADIUS PAP because PAM and Google Authenticator require access to the cleartext password string in the form password+OTP. PAP is not recommended on untrusted networks. Use a dedicated management VLAN, ACLs/firewall rules, and optionally VPN access before reaching the management VLAN.

IMPORTANT

The backslash character at the end of a line—Linux command or configuration—shows that the command continues on the next line.

4 Install and Prepare the FreeRADIUS Server

4.1 Install Google Authenticator on the Phone

Install Google Authenticator from Google Play or the Apple App Store.

Figure 1: Google Authenticator in Google Play

Figure 2: Google Authenticator in the Apple App Store

4.2 Set the Local Time Zone

Set the server's time zone. The example below uses Europe/Berlin.

```
$ sudo timedatectl set-timezone Europe/Berlin
```

Verify the current time zone and synchronization status:

```
$ timedatectl
```

Example output:

```
Local time: Mon 2026-04-20 10:54:09 CEST
Universal time: Mon 2026-04-20 08:54:09 UTC
RTC time: Mon 2026-04-20 08:54:09
Time zone: Europe/Berlin (CEST, +0200)
System clock synchronized: no
NTP service: active
RTC in local TZ: no
```

4.3 Install Required Packages

```
$ sudo apt install freeradius freeradius-utils \
libpam-google-authenticator
```

4.4 Create a Linux User

Create the Linux user that will authenticate through RADIUS. In this example, the username is moxa_user, and the password is moxa.

```
$ sudo adduser moxa_user
```

Example output:

```
info: Adding user ‘moxa_user’ ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group ‘moxa_user’ (1002) ...
info: Adding new user ‘moxa_user’ (1002) with group ‘moxa_user
(1002)’ ...
info: Creating home directory ‘/home/moxa_user’ ...
info: Copying files from ‘/etc/skel’ ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for moxa_user
Enter the new value, or press ENTER for the default
Full Name []: Moxa User
Room Number []:
Work Phone []:
Home Phone []:
Other []:
Is the information correct? [Y/n] y
info: Adding new user ‘moxa_user’ to supplemental / extra groups
‘users’ ...
info: Adding user ‘moxa_user’ to group ‘users’ ...
```

Note Example test credentials used in this guide: username moxa_user; password moxa. Replace these values in production.

4.5 Generate the Google Authenticator Secret

Log in via SSH as the target user and run google-authenticator to generate the user secret file.

```
ssh moxa_user@<server-ip-address>
$ google-authenticator
```

```
Do you want authentication tokens to be time-based (y/n) y
Warning: pasting the following URL into your browser exposes the OTP secret to Google:
  https://www.google.com/chart?chs=200x200&chld=M|0&cht=qr&chl=otpauth://totp/moxa_user@moxa%3Fsecret%3DTW3NQYX3ZGEINAQQBDZ4B7B6UY%26issuer%3Dmoxa
```

```
Your new secret key is: TW3NQYX3ZGEINAQQBDZ4B7B6UY
Enter code from app (-1 to skip): 793213
Code confirmed
Your emergency scratch codes are:
  22808415
  45040174
  86118705
  81372278
  15811330

Do you want me to update your "/home/moxa_user/.google_authenticator" file? (y/n) y

Do you want to disallow multiple uses of the same authentication
token? This restricts you to one login about every 30s, but it increases
your chances to notice or even prevent man-in-the-middle attacks (y/n) y

By default, a new token is generated every 30 seconds by the mobile app.
In order to compensate for possible time-skew between the client and the server,
we allow an extra token before and after the current time. This allows for a
time skew of up to 30 seconds between authentication server and client. If you
experience problems with poor time synchronization, you can increase the window
from its default size of 3 permitted codes (one previous code, the current
code, the next code) to 17 permitted codes (the 8 previous codes, the current
code, and the 8 next codes). This will permit for a time skew of up to 4 minutes
between client and server.
Do you want to do so? (y/n) n

If the computer that you are logging into isn't hardened against brute-force
login attempts, you can enable rate-limiting for the authentication module.
By default, this limits attackers to no more than 3 login attempts every 30s.
Do you want to enable rate-limiting? (y/n) y
```

Figure 3: Generating the Google Authenticator secret

Launch Google Authenticator on the phone, scan the QR code, enter the code from the app, and complete the prompts shown by the tool. After the setup is complete, exit the SSH session and continue as moxa sudo user.

```
$ exit
```

5 Configure PAM for FreeRADIUS

Edit the PAM configuration used by FreeRADIUS.

```
$ sudo nano /etc/pam.d/radiusd
```

For password plus OTP authentication, the file should contain only the following configuration:

```
#
# /etc/pam.d/radiusd - PAM configuration for FreeRADIUS #
# We  fall  back to the system default in /etc/pam.d/common-*  #

auth required pam_google_authenticator.so \
secret=/etc/freeradius/otp/${USER} \
user=freerad no_strict_owner \
allowed_perm=0600 forward_pass debug
auth required pam_unix.so use_first_pass
account required pam_unix.so
```

6 Configure FreeRADIUS

6.1 Enable the PAM Module

Edit the PAM module file.

```
$ sudo nano /etc/freeradius/3.0/mods-enabled/pam
```

It should contain:

```
$ sudo -i
# cd /etc/freeradius/3.0/mods-enabled/
```

Switch to root and change into the FreeRADIUS module directory:

```
$ sudo -i
# cd /etc/freeradius/3.0/mods-enabled/
```

Set the owner and permissions for the PAM module file:

```
# chown freerad:freerad pam
# chmod 640 pam
```

6.2 Enable PAM Authentication in the Default Site

Edit the default site configuration.

```
# nano /etc/freeradius/3.0/sites-enabled/default
```

In the authenticate section, add:

```
Auth-Type PAM {
     pam
}
```

6.3 Force the User to Use PAM

Edit the FreeRADIUS users authorization file.

```
# nano /etc/freeradius/3.0/mods-config/files/authorize
```

For the EDS-G4008 example, add:

```
moxa_user  Auth-Type :=  PAM
           Service-Type = 6,
           Reply-Message := "Admin access"
```

6.4 Enable Time Synchronization

Time synchronization is critical for OTP validation.

```
# timedatectl set-ntp true
```

6.5 Create a Shared OTP Directory

Create a directory that FreeRADIUS can use to read the Google Authenticator secret file.

```
$ sudo mkdir -p /etc/freeradius/otp
$ sudo chown freerad:freerad /etc/freeradius/otp
$ sudo chmod 700 /etc/freeradius/otp
```

Move the user's Google Authenticator secret file into the shared OTP directory:

```
$ sudo cp /home/moxa_user/.google_authenticator \
/etc/freeradius/otp/moxa_user
$ sudo chown freerad:freerad /etc/freeradius/otp/moxa_user
$ sudo chmod 600 /etc/freeradius/otp/moxa_user
```

6.6 Add the RADIUS Client Network

Edit /etc/freeradius/3.0/clients.conf.

```
# nano /etc/freeradius/3.0/clients.conf
```

Add the IPv4 client definition after the IPv6 content:

```
# IPv4 Client
client MOXA_NET {
     ipaddr = 192.168.127.0/24
     secret = testing1234
     require_message_authenticator = no
     nas_type = other
}
```

IMPORTANT

Use a strong shared secret in production. The sample values testing123 and testing1234 are for lab validation only.

6.7 Start FreeRADIUS in Debug Mode

Stop the FreeRADIUS service and start it in debug mode for validation.

```
# systemctl stop freeradius
# freeradius -X
```

7 Validate RADIUS Authentication on Linux

Test authentication with the Linux password followed immediately by the current Google Authenticator OTP.

```
radtest moxa_user 'moxa<CURRENT_OTP>' localhost 0 testing123
```

When testing on the FreeRADIUS server itself, make sure the client localhost section in clients.conf uses the same shared secret:

```
client localhost {
     ...
     secret = testing123
     ...
}
```

8 Configure the EDS-G4008 Switch

8.1 Set Login Authentication

In the Web UI, open **Security > Authentication > Login Authentication** and set the login authentication method to **RADIUS, Local**.

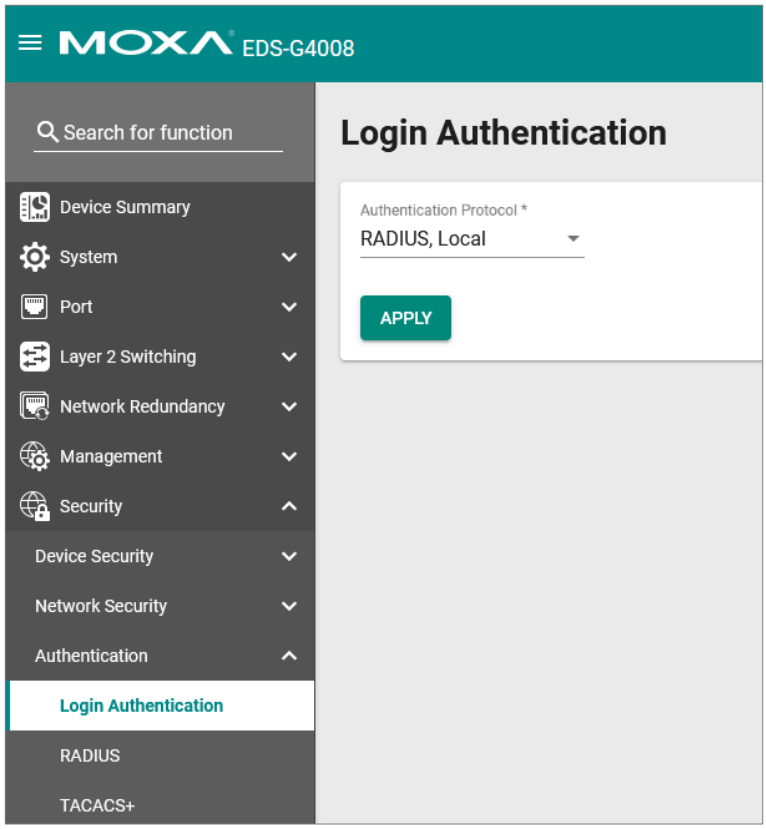

Figure 4: EDS-G4008 login authentication settings

8.2 Configure the RADIUS Server

Open **Security > Authentication > RADIUS setting** and configure the RADIUS server parameters.

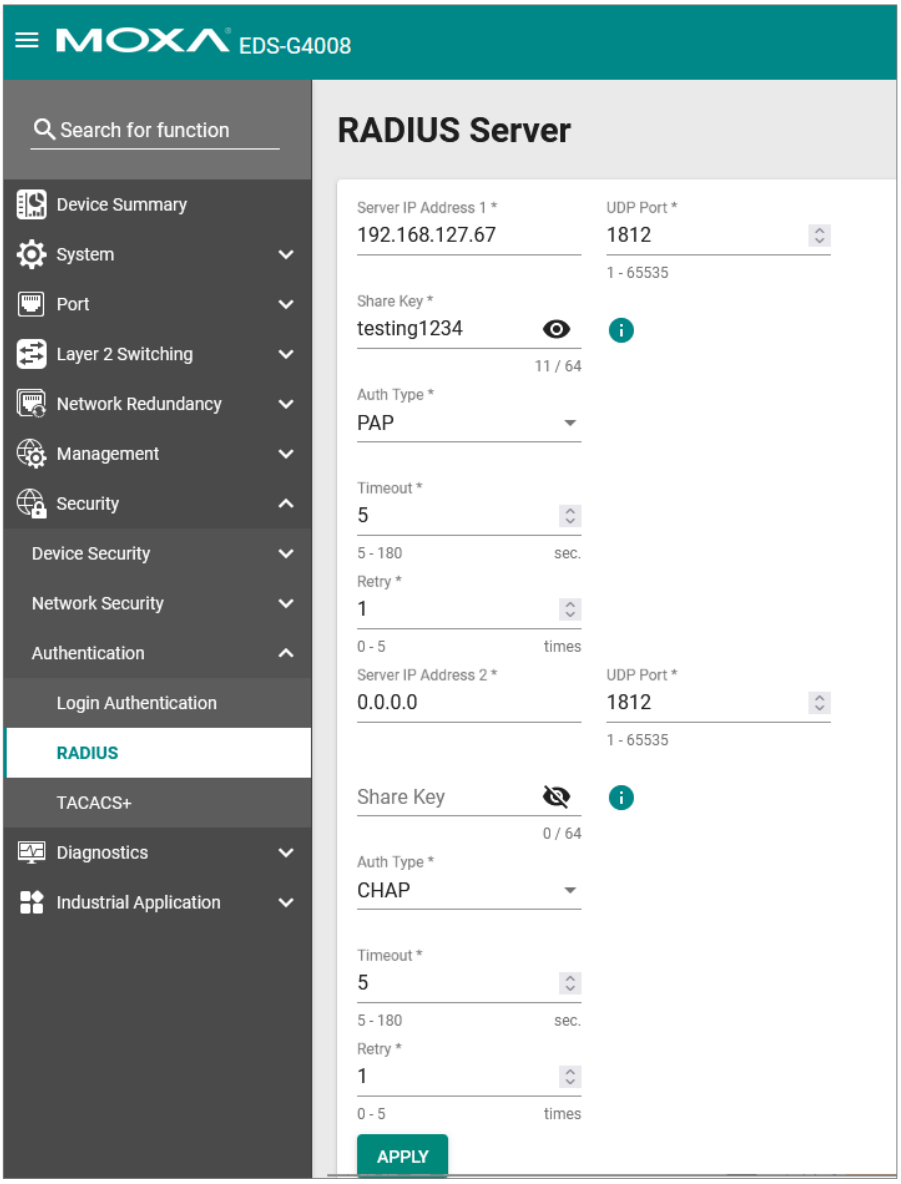

Figure 5: EDS-G4008 RADIUS server settings

IMPORTANT

Authentication Type must be set to **PAP**. If the client only supports MS-CHAPv2 or EAP-PEAP/MSCHAPv2, simple password+OTP concatenation with PAM will not work directly because PAM requires a cleartext user-password. PAP is required when the server needs the cleartext user password.

8.3 Log In to the Device

For Web UI login on the EDS-(G)4000 or NPort-G2 series, use:

- Username: moxa_user
- Password: moxa<OTP>

For example, if the current OTP is 246071, the password is:

```
moxa246071
```

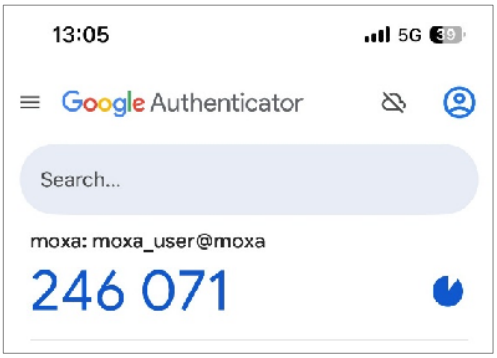

Figure 6: Example OTP shown in Google Authenticator

9 Useful Operational Information

9.1 PAM Authentication Debugging

The debug option in /etc/pam.d/radiusd enables more verbose syslog logging from pam_google_authenticator. This helps identify whether the OTP is wrong or whether there is still a file access problem.

Relevant PAM line:

```
auth required pam_google_authenticator.so \
secret=/etc/freeradius/otp/${USER} user=freerad no_strict_owner \
allowed_perm=0600 forward_pass debug
```

Watch logs with:

```
# journalctl -f | grep radiusd
```

9.2 NPort 6150-G2-T Authorization Attributes

For the NPort 6150-G2-T, use the following entry in /etc/freeradius/3.0/mods-config/files/authorize:

```
moxa_user   Auth-Type := PAM
            Service-Type = Login,
            Filter-Id := "Administrator",
            Reply-Message := "Admin access"
```

10 Configure the NPort 6150-G2-T

Open the NPort Web UI and configure the RADIUS server under **Account Management > Authentication Server > CREATE**.

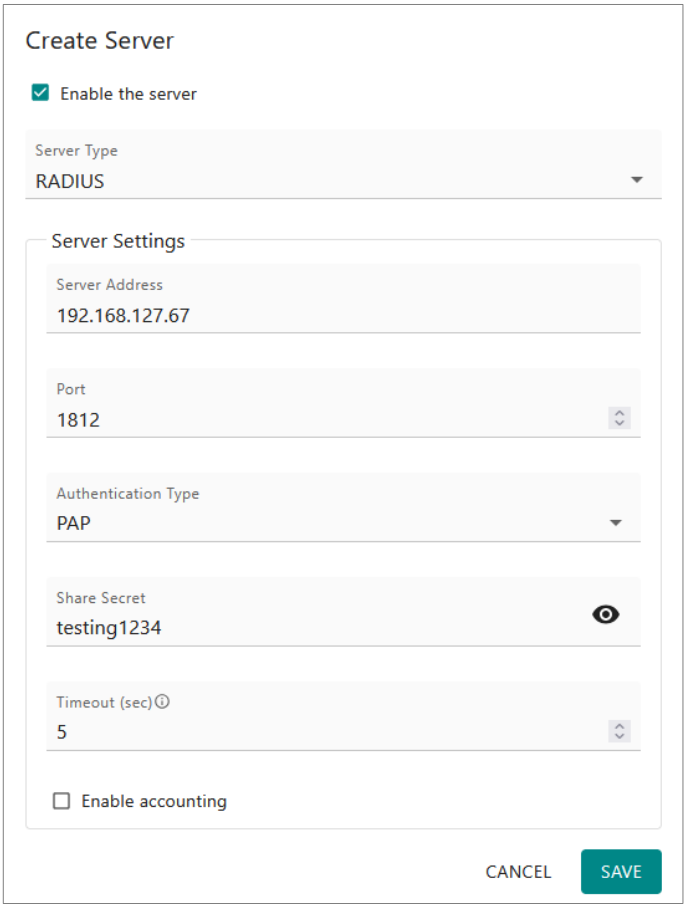

Figure 7: NPort 6150-G2-T RADIUS authentication server configuration

11 Alternative PAM Authentication Modes

11.1 Google Authenticator Only

If the user should be authenticated only by Google Authenticator, edit /etc/pam.d/radiusd and use:

```
$ sudo nano /etc/pam.d/radiusd

auth required pam_google_authenticator.so \
  secret=/etc/freeradius/otp/${USER} user=freerad no_strict_owner \
  allowed_perm=0600 debug
```

11.2 Linux Username and Password Only

If the user should be authenticated only by Linux account username and password, edit /etc/pam.d/radiusd and use:

```
$ sudo nano /etc/pam.d/radiusd

auth required pam_unix.so
account required pam_unix.so
```

12 Why PAP Is Used

PAP is used in this deployment because it allows FreeRADIUS and PAM to process:

- The real user password;
- Password+OTP concatenation;
- PAM integration with Google Authenticator.

This workflow cannot be implemented in the same way with MS-CHAPv2 because PAM does not receive the cleartext password string.

12.1 Recommended Architecture

Because PAP is not encrypted end-to-end, protect the RADIUS path using a secure management design:

- Place the switch and RADIUS server in a dedicated management VLAN.
- Restrict access to the VLAN with firewall rules or ACLs.
- Allow only administrators to access the management network.
- Optionally require VPN access before accessing the management VLAN.

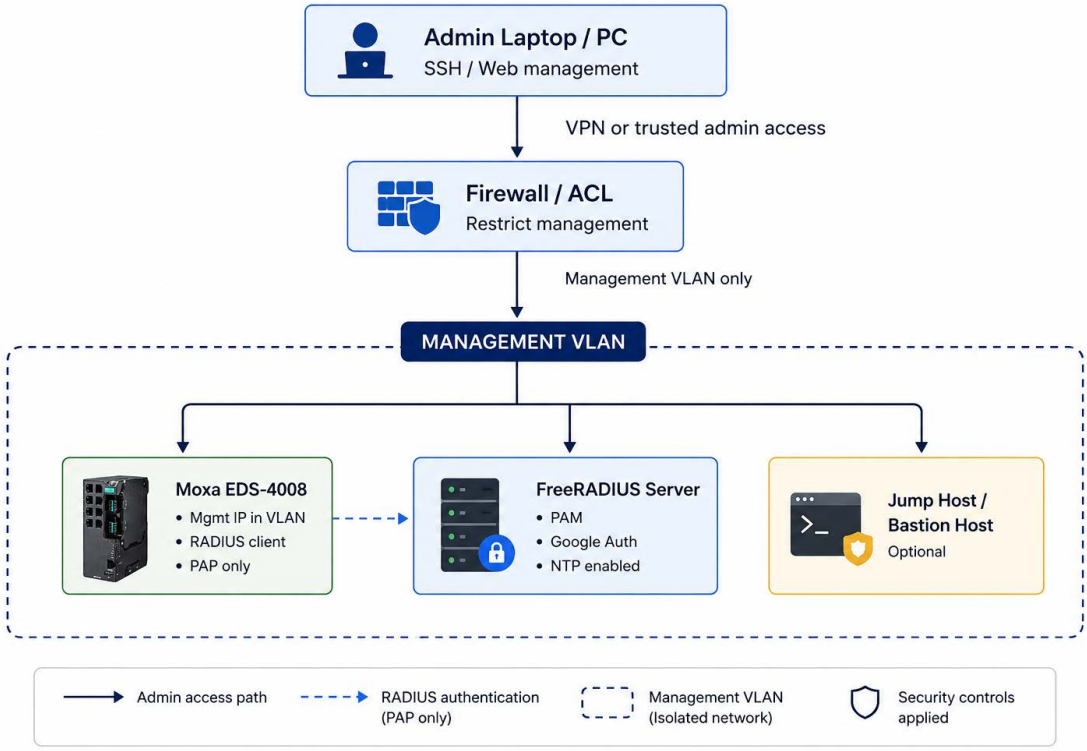

Figure 8: Recommended management VLAN architecture

12.2 Authentication Flow

1. The administrator connects to the switch via SSH or HTTPS.
2. The switch sends a RADIUS request to FreeRADIUS over the management VLAN.
3. The switch authentication method is PAP.
4. FreeRADIUS passes the request to PAM.
5. PAM checks the Linux password and the Google Authenticator OTP.
6. FreeRADIUS returns Access-Accept and role attributes such as Service-Type and Filter-Id.

13 PAP Security Behavior

When a switch uses PAP, it sends credentials in a RADIUS Access-Request. The attribute used is User-Password. RADIUS does not send the password as plain ASCII directly, but it is reversibly encrypted using the shared secret. This is considered obfuscation rather than strong encryption.

13.1 Wireshark Behavior

If traffic is captured without the RADIUS shared secret, the username is readable and the password appears as an encrypted or hashed blob. If Wireshark is configured with the RADIUS shared secret, it can decode the packet and show the user password.

To configure the shared secret in Wireshark, go to:

```
Edit -> Preferences -> Protocols -> RADIUS
```

Then add the shared secret. Wireshark can then show values such as:

```
User-Name = "admin"
User-Password = "mypassword"
```

IMPORTANT

With PAP, anyone who can capture the traffic and knows or guesses the shared secret can recover passwords. Even without the shared secret, PAP is considered weak. Restrict and protect the management network accordingly.

14 Deployment Checklist

Use this checklist before handing over the deployment.

- FreeRADIUS packages and the Google Authenticator PAM module are installed.
- Time synchronization is enabled and verified.
- Linux user account exists and has a Google Authenticator secret.
- The secret file is copied to /etc/freeradius/otp/ with owner freerad:freerad and mode 600.
- PAM configuration points to /etc/freeradius/otp/${USER}.
- FreeRADIUS PAM module is enabled and owned by freerad.
- The target user is forced to use Auth-Type:= PAM.
- The Moxa device network is listed in clients.conf.
- EDS or NPort RADIUS authentication type is set to PAP.
- Login has been tested using password+OTP.
- Management VLAN, firewall rules, and ACLs are in place.