MIGRATE FROM A THIRD PARTY 2FA SOFTWARE TO OPENOTP

Migrate from a third party 2FA software to OpenOTP

MIGRATION

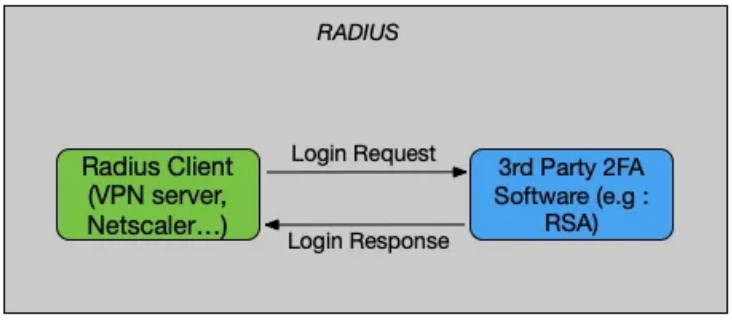
1. Overview

In this how-to, we will demonstrate how to easily migrate from a third party 2FA software to OpenOTP. In this documentation, we assume that you are already running WebADM, OpenOTP and Radius Bridge.

To understand what will be done here, we will describe the steps:

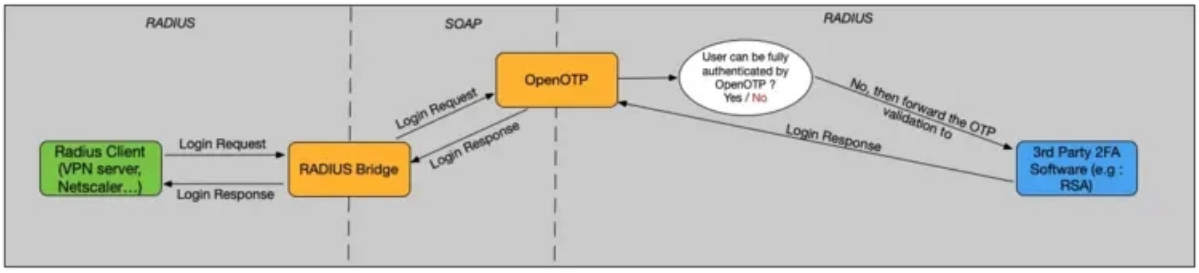
- Have a WebADM, OpenOTP and Radius Bridge installed and configured,
- Activate every user who will require 2FA authentication at the WebADM level,
- Import your third-party hardware Tokens into WebADM.
- Configure a `RADIUS Proxy` under OpenOTP configuration page (that should point to the RADIUS Server available in your current 2FA software),
- Allow OpenOTP as Radius Client in your RADIUS configuration third-party software,
- Configure the OpenOTP fallback authentication method to `PROXY`,
- Configure your Radius Clients in Radius Bridge,
- Redirect your client applications who consume RADIUS protocol from your 3rd party 2FA software to Radius Bridge and OpenOTP.

1.1 Initial setup with you 3rd party 2FA software

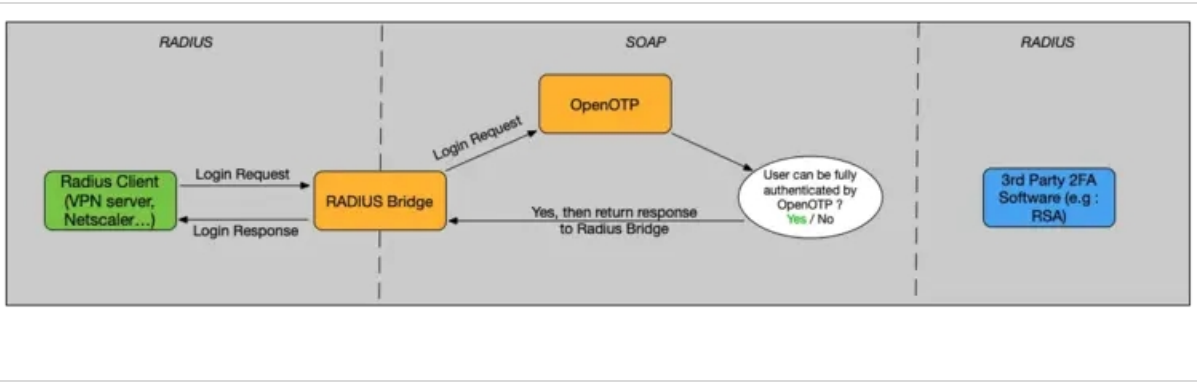


1.2 Required architecture for the migration

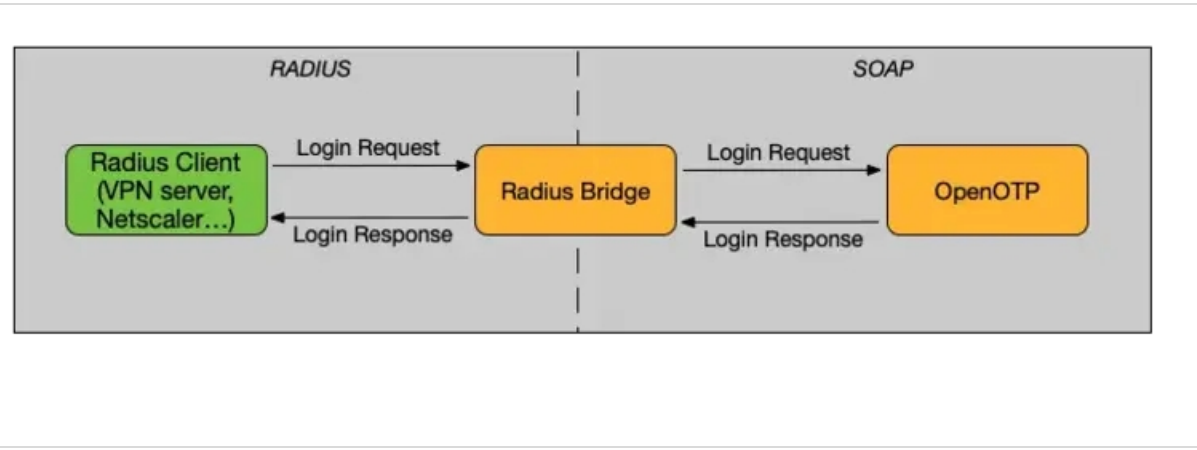
1.2.1 Workflow for a user not already migrated to OpenOTP



1.2.2 Workflow for a user already migrated to OpenOTP



1.3 After every user's migration



2. Users Activation

First, you need to Activate every user you want to migrate from third-party 2FA software to OpenOTP. The activation is mandatory else, OpenOTP will not be able to find the user. There are multiple ways to activate your users, have a look at the

following documentation User Activation for that step.

3. Import third party hardware tokens to use them with OpenOTP

If you already own hardware Tokens from a 3rd party Token vendor, you can maybe use it with WebADM. The requirements are that your Tokens type must respect the OATH standard, and you must have the Token seeds file provided by your token reseller. The token seeds file should be in PSKC format. To be able to use Tokens, we need to import the seeds file under WebADM, but to import your seeds file, you need to convert it first into a format managed by WebADM. To perform this, RCDevs provide a script who will be converting your inventory file in the WebADM format. This script can be found on your WebADM server at the following path :

```
/opt/webadm/websrvs/openotp/bin/pskc2inv
WebADM Inventory converter for OATH PSKC files
Usage: pskc2inv <pskc-file> <inventory-file> [<decryption-key>]
```

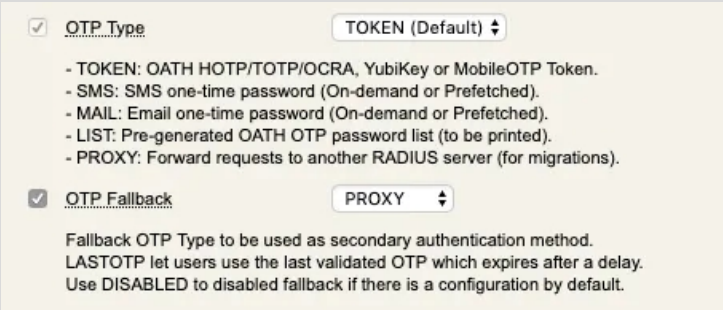
```
/opt/webadm/websrvs/openotp/bin/pskc2inv /tmp/TokenSeed.xml /tmp/webadm_inv.xml
my_decryption_key
```

You can now import the generated seeds file into WebADM. In order to perform this, please have a look at this documentation Hardware Token Import. Once your seeds file is imported in the databases, you are able to assign hardware token to the user account. You have to choose the option `I use a Hardware Token (Inventoried)` during the Token allocation.

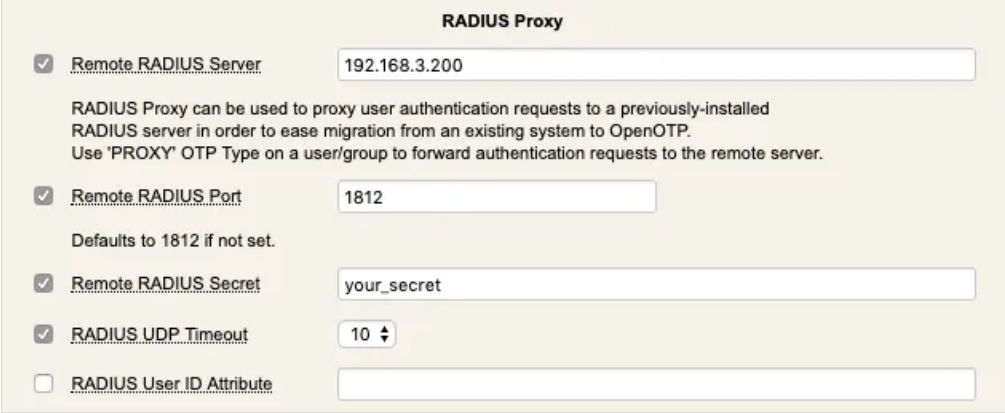
4. Configure a RADIUS Proxy in OpenOTP

In order to migrate progressively each user from your 3rd party 2FA platform to OpenOTP, we need to configure a Radius Proxy into OpenOTP configuration. That means, if OpenOTP is not able to authenticate the user because the user is not yet migrated to OpenOTP then, OpenOTP will forward the authentication request to the Radius proxy. The Radius Proxy will be your 3rd party 2FA software.

To configure your Radius Proxy, you have to log in on the WebADM Admin GUI > `Applications` tab > `Authentication` > `MFA AUthentication Server (OpenOTP)` > `CONFIGURE`. Under the `Authentication Policy` section, you will find the settings `OTP Type` and `OTP fallback` which must be respectively configured to `Token` and `Proxy`.



Scroll down until the `RADIUS Proxy` section and configure your 3rd party 2FA radius server.



You can optionally configure another User ID attribute for the username value sent to the Radius Proxy.

5. Configure OpenOTP as a Radius Client in your 3rd party 2FA software

To allow OpenOTP to communicate with your 3rd party 2FA software, you need to configure OpenOTP as a RADIUS client in your 3rd party 2FA software. Else every authentication requests forwarded by OpenOTP/WebADM will be dropped by your current RADIUS Server (the 3rd party 2FA software).
In order to perform that, please refer to your 3rd party 2FA software documentation.

6. Configure your RADIUS clients in Radius Bridge

Now OpenOTP is allowed to communicate with your 3rd party 2FA software. We have to configure every client who consumes RADIUS protocol on your 3rd party 2FA software into Radius Bridge.

Your RADIUS clients (ex. VPN server) must be registered in the `/opt/radiusd/conf/clients.conf` file to be able to communicate with the Radius Bridge server. A client configuration looks this:

```
client my_vpn {
   ipaddr = 192.168.0.10
   secret = testing123
}
```

You need to set the IP address of your VPN Server and configure the shared RADIUS secret. On the VPN side, you will have to configure the Radius Bridge server IP as RADIUS Server, and you have to set the same secret as configured in the `clients.conf` file.

> **Note**
>
> Always prefer setting no RADIUS retries (retries=0) on the RADIUS configuration of your VPN when you use OpenOTP challenge mode.

Perform this for all of your RADIUS clients.

7. Redirect your RADIUS clients to Radius Bridge

Now Radius Bridge and OpenOTP are ready to perform authentications instead of your 3rd party 2FA software. So now, you can redirect every RADIUS clients usually pointing to your 3rd party 2FA software to Radius Bridge/OpenOTP. A migrated user (user already activated under WebADM which has a Token enrolled on WebADM) will be authenticated by OpenOTP. If the user is not activated at the WebADM level, then it will be a failure for every authentication because WebADM will not be able to find it. If a user is activated but doesn't have any token enrolled then, the authentication request will be delegated from OpenOTP to your 3rd party 2FA software to validate the OTP.

> **Note**
>
> The 'PROXY' OTP fallback method configured under OpenOTP sends only the OTP code and not the LDAP password. The LDAP password will be validated by OpenOTP itself. Be careful to configure your 3rd party 2FA software to check only OTPs for requests coming from OpenOTP.

Until every user are migrated to OpenOTP, both 2FA system must be running concurrently. The 3rd party 2FA software can be shut down when all of your users are migrated to OpenOTP.

```
[2019-02-25 12:03:57] [127.0.0.1] [OpenOTP:0C9Y6C89] New openotpSimpleLogin SOAP request
[2019-02-25 12:03:57] [127.0.0.1] [OpenOTP:0C9Y6C89] > Username: testmigration
[2019-02-25 12:03:57] [127.0.0.1] [OpenOTP:0C9Y6C89] > Password: xxxxxxxx
[2019-02-25 12:03:57] [127.0.0.1] [OpenOTP:0C9Y6C89] > Client ID: RadTest
[2019-02-25 12:03:57] [127.0.0.1] [OpenOTP:0C9Y6C89] > Options: RADIUS,-U2F
[2019-02-25 12:03:57] [127.0.0.1] [OpenOTP:0C9Y6C89] Registered openotpSimpleLogin request
[2019-02-25 12:03:57] [127.0.0.1] [OpenOTP:0C9Y6C89] Resolved LDAP user:
CN=testmigration,CN=Users,DC=yorcdevs,DC=com (cached)
[2019-02-25 12:03:57] [127.0.0.1] [OpenOTP:0C9Y6C89] Started transaction lock for user
[2019-02-25 12:03:57] [127.0.0.1] [OpenOTP:0C9Y6C89] Found user fullname: testmigration
[2019-02-25 12:03:57] [127.0.0.1] [OpenOTP:0C9Y6C89] Found 43 user settings:
LoginMode=LDAPOTP,ExpireNotify=SMS,OTPType=PROXY,OTPLength=6,ChallengeMode=Yes,ChallengeTime
1:HOTP-SHA1-6:QN06-
T1M,DeviceType=FIDO2,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,PrefetchExpire=10,

[2019-02-25 12:03:57] [127.0.0.1] [OpenOTP:0C9Y6C89] Found 1 user data: RejectCount
[2019-02-25 12:03:57] [127.0.0.1] [OpenOTP:0C9Y6C89] Requested login factors: LDAP & OTP
[2019-02-25 12:03:57] [127.0.0.1] [OpenOTP:0C9Y6C89] LDAP password Ok
[2019-02-25 12:03:57] [127.0.0.1] [OpenOTP:0C9Y6C89] Authentication challenge required
[2019-02-25 12:03:57] [127.0.0.1] [OpenOTP:0C9Y6C89] Started OTP authentication session of ID
6R8TE6qftwtFcsrR valid for 90 seconds
[2019-02-25 12:03:57] [127.0.0.1] [OpenOTP:0C9Y6C89] Sent challenge response
[2019-02-25 12:04:04] [127.0.0.1] [OpenOTP:0C9Y6C89] New openotpChallenge SOAP request
[2019-02-25 12:04:04] [127.0.0.1] [OpenOTP:0C9Y6C89] > Username: testmigration
[2019-02-25 12:04:04] [127.0.0.1] [OpenOTP:0C9Y6C89] > Session: 6R8TE6qftwtFcsrR
[2019-02-25 12:04:04] [127.0.0.1] [OpenOTP:0C9Y6C89] > OTP Password: xxxxxx
[2019-02-25 12:04:04] [127.0.0.1] [OpenOTP:0C9Y6C89] Registered openotpChallenge request
[2019-02-25 12:04:04] [127.0.0.1] [OpenOTP:0C9Y6C89] Found authentication session started 2019-02-
25 12:03:57
[2019-02-25 12:04:04] [127.0.0.1] [OpenOTP:0C9Y6C89] Started transaction lock for user
[2019-02-25 12:04:04] [127.0.0.1] [OpenOTP:0C9Y6C89] Forwarding request to RADIUS server
192.168.3.200
[2019-02-25 12:04:04] [127.0.0.1] [OpenOTP:0C9Y6C89] Received RADIUS success
[2019-02-25 12:04:04] [127.0.0.1] [OpenOTP:0C9Y6C89] Proxy password Ok
[2019-02-25 12:04:04] [127.0.0.1] [OpenOTP:0C9Y6C89] Updated user data
[2019-02-25 12:04:04] [127.0.0.1] [OpenOTP:0C9Y6C89] Sent success response
```