



SWIFT ALLIANCE ACCESS AND OPENOTP

The specifications and information in this document are subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. This document may not be copied or distributed by any means, in whole or in part, for any reason, without the express written permission of RCDevs.

Copyright (c) 2010-2017 RCDevs SA. All rights reserved.

<http://www.rcdevs.com>

WebADM and OpenOTP are trademarks of RCDevs. All further trademarks are the property of their respective owners.

Limited Warranty

No guarantee is given for the correctness of the information contained in this document. Please send any comments or corrections to info@rcdevs.com.

1. Overview

In this documentation, we will demonstrate how to integrate OpenOTP with Swift Alliance Access 7.2 (AA). LDAP and Radius protocols can be used to integrate AA with OpenOTP. Here, we will demonstrate the Radius integration. This guide has been written with the help of the official Swift Alliance Access 7.2 Administrator Guide. So here, we will use RADIUS one-time passwords authentication method and not the embedded two-factor authentication module implemented in AA. WebADM and OpenOTP server(s) should be already configured with Radius Bridge component(s).

2. Swift Alliance Access Configuration

2.1. Authentication Server Groups

First, we will configure the Radius servers at the AA level. Log into the AA Web management page with the LSO (Left Security Officer) account and configure a new authentication server group.

Once you are under Radius Authentication Servers Group, you are able to configure the required information to communicate with Radius Bridge.

Configure the Primary Server in the Future configuration section.

> **192.168.3.54** : Is my Radius Bridge, WebADM & OpenOTP Server IP.

> **Key Left** : This should be a value of 16 characters.

> **Key Right** : This should be a value of 16 characters.

These two keys (bilateral key) will be used as Radius secret in the Radius Bridge clients definition.

> **Port Number** : This is the port used by the Radius Bridge service.

Once this configuration is done, you can click on the save button. If you have a WebADM/OpenOTP cluster then configure the secondary server the same way. These changes should be approved by the RSO (Right Security Officer) account. Once the RSO has approved the new configuration, the Radius server configuration on Alliance Access is done.

⚠ Important note from Swift

The usage of one-time passwords is set per operator. To activate the use of one-time passwords, in the Operator Details for each security officer, the Authentication Type must be set to RADIUS one-time password and the Authentication Server Group must be selected. Each change must be approved by RSO and LSO account.

3. Radius Bridge Configuration

3.1 Clients Configuration

To allow Swift Alliance Access to communicate over Radius protocol, we have to configure the AA Radius client in Radius Bridge configuration. To configure the client edit `/opt/radiusd/conf/clients.conf` file. At the end of this file, you will find the clients definition.

```
[root@webadm ~]# vi /opt/radiusd/conf/clients.conf
```

Add a new client for allowing Swift AA:

```
client Swift_AA {
    ipaddr = 192.168.3.56
    secret = Left_key_1234567Right_key_123456
}
```

- > `192.168.3.56`: Is the Swift AA IP who will contact Radius Bridge.
- > `Left_key_1234567Right_key_123456`: Is the concatenation of left and right keys defined in Alliance Access configuration.

Once the Swift Alliance Access client is configured in `clients.conf` file, you will have to restart Radius Bridge service:

```
[root@webadm ~]# /opt/radiusd.bin/radiusd restart
```

These changes must be done on each Radius Bridge if you are working with a WebADM/OpenOTP cluster.

3.2 Radiusd Advanced Configuration (Optional)

To map the User IP information in `WebADM WebSrv logs`, you will have to configure the attribute used by Swift which contains the User IP in Radius Bridge configuration in the `source_attribute` setting.

```
[root@webadm ~]# vi /opt/radiusd/conf/radiusd.conf
```

```
# Source attribute
# This is the RADIUS attribute in which the RADIUS client can pass the end user source
IP address to
# OpenOTP. Attribute must be of type IPAddr.
# By default the source attribute is set to Calling-Station-Id & PaloAlto-Client-
Source-IP.
source_attribute = "Swift_user_ip_attribute"
```

Restart radius bridge service after modifying this file.

4. OpenOTP Client Policy Configuration

We will now configure a client policy for Swift authentications. Login on the WebADM Administrator GUI > **Admin** tab > **Client Policies** > **Add Client**.

Name the client policy object which will be created, on my side **Swift** and optionally add a description.

Click on **Proceed** button and then **Create Object**.

You are now in the Swift Client Policy configuration menu. The first setting you will have to configure is the **Client Name Aliases** where you will configure the AA IP which will contact the OpenOTP. On my side **192.168.3.56**.

Next step is to configure the authentication policy to require MFA on Swift AA. You will have to edit the **Forced Application Policies** under the client policy menu. Please, activate **Application Settings** and then click on **Edit**.

Now, activate **Login Mode: LDAPOTP**, **OTP Type: TOKEN**, **Challenge Mode Supported: No** and **Challenge Password Retry: No**. Finally, click on **Apply**.

You should have this result like below:

Click on the **Apply** button to save your changes.

⚠ Important Note

OpenOTP.ChallengeMode=No is mandatory with Swift AA because Swift didn't implement the Radius challenge in their product. So even with Radius, you will have only 2 fields on the AA login page, one for the Username and the other for the password. In the password field, you will have to put LDAP password and OTP password in concatenated mode.

Your client policy for Swift is now configured. You can test a login on AA with OpenOTP.

⚠ Push Login Authentication method

Swift can work with OpenOTP and Push login but if the Push Notification is not received on your phone, or if you are not able to Approve the login with your phone, the authentication will fail because no fallback method is available in that scenario. You will have to retry the authentication and the failed login counter will be increased at Swift level. After x login failure for the same account, the account will be blocked at the Swift level.

5. LDAP User and Swift user mapping

If your Swift users already have an account in your Directory then it's possible to do mapping at the LDAP level between the LDAP account and the Swift local account. This mapping is done by adding the Swift login name value in an LDAP attribute. This attribute must be configured in `/opt/webadm/conf/webadm.conf` file in `uid_attrs` setting. By default with the Active Directory template, the following ones are available.

```
uid_attrs          "cn", "samAccountName", "userPrincipalName"
```

If one of these attributes is not used then you can use one of them and configure the Swift username on that attribute. If the default attributes are already used in your organization, then you can use another one. For example `uid`, in that case, you have to add the `uid` attribute in `uid_attrs` setting in `webadm.conf` like below:

```
uid_attrs          "cn", "samAccountName", "userPrincipalName", "uid"
```

When you will perform a login from Swift with your Swift account, then the Swift username will be sent to WebADM/OpenOTP and will match with the corresponding LDAP account.

6. Authentication Logs

After performing authentication on Swift Alliance Access, you are able to check logs on the WebADM side. Through the WebADM Admin GUI > `Databases` > `WebADM Server Log Files` you should have something like this below:

```

[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] New openotpSimpleLogin SOAP
request
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] > Username: Administrateur
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] > Password: xxxxxxxxxxxxxxxx
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] > Options: RADIUS,-U2F
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Enforcing client policy: Swift
(matched server IP)
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Registered openotpSimpleLogin
request
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Resolved LDAP user:
CN=Administrateur,CN=Users,DC=yorcdevs,DC=com
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Resolved LDAP groups:
master,propriétaires créateurs de la stratégie de groupe,admins du
domaine,administrateurs de l'entreprise,administrateurs du schéma,utilisateurs du
bureau à distance,administrateurs,groupe de réplication dont le mot de passe rodc est
refusé
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Started transaction lock for
user
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Found user fullname:
administrateur
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Found user language: EN
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Found 1 user mobiles:
xxxxxxxxxxxx
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Found 1 user emails:
support@rcdevs.com
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Found 1 user certificates
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Found 43 user settings:
LoginMode=LDAPOTP,ExpireNotify=MAIL,OTPTType=TOKEN,OTPLength=6,ChallengeMode=No,Challenge1
1:HOTP-SHA1-6:QN06-
TIM,DeviceType=FID02,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,PrefetchExpire=10,
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Found 20 user data:
LoginCount,RejectCount,LastOTP,ListInit,ListState,OTPPrefix,NowaitState,TokenKey
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Found 1 registered OTP token
(TOTP)
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Challenge mode disabled
(checking concatenated passwords)
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Requested login factors: LDAP &
OTP
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] LDAP password Ok
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] TOTP password Ok (token #1)
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Updated user data
[2018-12-11 17:43:07] [192.168.3.54] [OpenOTP:4IT5D3I6] Sent success response

```

And under **Databases** > **WebSrv Logs** you should have something like this below:

As you can see here, we only see the host IP which is the Swift AA IP in the audit logs. To map the User IP information, please refer

to 3.2 chapter.

This manual was prepared with great care. However, RCDevs S.A. and the author cannot assume any legal or other liability for possible errors and their consequences. No responsibility is taken for the details contained in this manual. Subject to alternation without notice. RCDevs S.A. does not enter into any responsibility in this respect. The hardware and software described in this manual is provided on the basis of a license agreement. This manual is protected by copyright law. RCDevs S.A. reserves all rights, especially for translation into foreign languages. No part of this manual may be reproduced in any way (photocopies, microfilm or other methods) or transformed into machine-readable language without the prior written permission of RCDevs S.A. The latter especially applies for data processing systems. RCDevs S.A. also reserves all communication rights (lectures, radio and television). The hardware and software names mentioned in this manual are most often the registered trademarks of the respective manufacturers and as such are subject to the statutory regulations. Product and brand names are the property of RCDevs S.A. © 2021 RCDevs SA, All Rights Reserved