WHAT IS WRONG??

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 Security.

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

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

What is Wrong??

Troublehsooting

1. Overview

In this document, we describe how to easily fix some common errors with WebADM, OpenOTP, Web Applications, Radius Bridge, Push login, License services, LDAP permissions etc.

2. WebADM/OpenOTP common issues

The first thing to do when a login failed for an unknown reason is to check the log file `/opt/webadm/log/webadm.log` and find the right log. In addition to the terminal session, you can find the log also in WebADM > Databases > WebADM Server Log files.

Please note that this log is per server, so in clustered environment you will need to check from each server separately.

2.1 User invalid or not found

Logs example :

```
[2017-07-21 09:13:16] [127.0.0.1] [OpenOTP:MKRVHYLX] New openotpSimpleLogin SOAP request
[2017-07-21 09:13:16] [127.0.0.1] [OpenOTP:MKRVHYLX] > Username: test@yorcdevs.eu
[2017-07-21 09:13:16] [127.0.0.1] [OpenOTP:MKRVHYLX] > Password: xxxxxx
[2017-07-21 09:13:16] [127.0.0.1] [OpenOTP:MKRVHYLX] > Client ID: RadTest
[2017-07-21 09:13:16] [127.0.0.1] [OpenOTP:MKRVHYLX] > Options: RADIUS,-U2F
[2017-07-21 09:13:16] [127.0.0.1] [OpenOTP:MKRVHYLX] Registered openotpSimpleLogin request
[2017-07-21 09:13:16] [127.0.0.1] [OpenOTP:MKRVHYLX] User invalid or not found
[2017-07-21 09:13:17] [127.0.0.1] [OpenOTP:MKRVHYLX] Sent failure response
```

Possible reasons/Solutions :

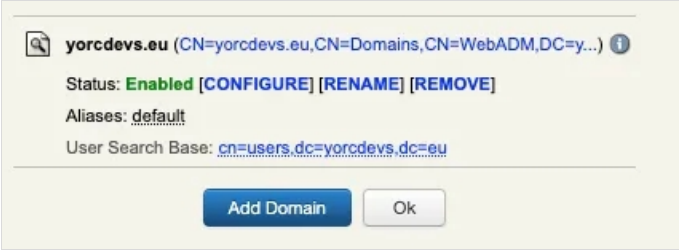
- Account doesn't exist in the LDAP;
- The user account is not activated in WebADM;
- Username used during the authentication (here test@yorcdevs.eu) doesn't match any user in the LDAP (username attributes checked by WebADM can be founded in `/otp/webadm/conf/webadm.conf` setting `uid_attrs`).

For AD, the default username attributes are :

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

- proxy_user configured in `/otp/webadm/conf/webadm.conf` do not have read permission on the user account.
- The user is outside the user search base(s) defined in WebADM > Admin > User Domains.

E.g : WebADM domain configuration



User search base is : `cn=users,dc=yorcdevs,dc=eu` The DN of my User is : `CN=test,CN=Internal,DC=yorcdevs,DC=eu`

As you can see, the user search base configured in my WebADM domain does not include `CN=Internal,DC=yorcdevs,DC=eu` container, thus WebADM is not able to find the user. The solution is to change the User search base configured in your WebADM domain to include the container where your user is located.

E.g : User search base : `dc=yorcdevs,dc=eu`

2.2 No usable login method found / User has no OTP token registered / Account missing required data or MFA enrolment needed

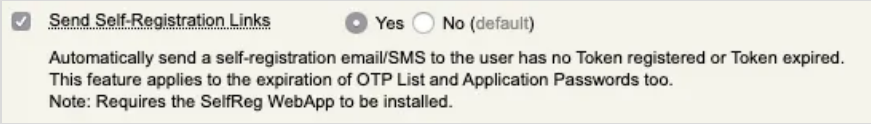
Logs example :

```
[2020-04-14 13:33:14] [192.168.3.64] [OpenOTP:R4NOCMJR] New openotpNormalLogin SOAP request
[2020-04-14 13:33:14] [192.168.3.64] [OpenOTP:R4NOCMJR] > Username: test
[2020-04-14 13:33:14] [192.168.3.64] [OpenOTP:R4NOCMJR] > Domain: yorcdevs.eu
[2020-04-14 13:33:14] [192.168.3.64] [OpenOTP:R4NOCMJR] > LDAP Password: xxxxxxxx
[2020-04-14 13:33:14] [192.168.3.64] [OpenOTP:R4NOCMJR] > Client ID: OpenOTP
[2020-04-14 13:33:14] [192.168.3.64] [OpenOTP:R4NOCMJR] > Source IP: 192.168.3.64
[2020-04-14 13:33:14] [192.168.3.64] [OpenOTP:R4NOCMJR] > Context ID:
7f869f80b588210691b32bb4e09248c5
[2020-04-14 13:33:14] [192.168.3.64] [OpenOTP:R4NOCMJR] Registered openotpNormalLogin request
[2020-04-14 13:33:14] [192.168.3.64] [OpenOTP:R4NOCMJR] Resolved LDAP user:
CN=test,CN=Users,DC=yorcdevs,DC=eu (cached)
[2020-04-14 13:33:15] [192.168.3.64] [OpenOTP:R4NOCMJR] Started transaction lock for user
[2020-04-14 13:33:15] [192.168.3.64] [OpenOTP:R4NOCMJR] Found user fullname: test
[2020-04-14 13:33:15] [192.168.3.64] [OpenOTP:R4NOCMJR] Found 47 user settings:
LoginMode=LDAPOTP,OTPType=TOKEN,PushLogin=Yes,ChallengeMode=Yes,ChallengeTimeout=90,OTPLeng
1:HOTP-SHA1-6:QN06-
T1M,DeviceType=FIDO2,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,PrefetchExpire=10,

[2020-04-14 13:33:15] [192.168.3.64] [OpenOTP:R4NOCMJR] User has no OTP token registered
[2020-04-14 13:33:15] [192.168.3.64] [OpenOTP:R4NOCMJR] No usable login method found
[2020-04-14 13:33:15] [192.168.3.64] [OpenOTP:R4NOCMJR] Sent failure response
```

Possible reasons/Solutions :

- The authentication policy requires a 2nd factor which cannot be provided by the user. In this case the `LoginMode=LDAPOTP` but the user does not have OTP Token registered. The solution here is to register an OTP Token on the user account.

With the User Self-Registration (SelfReg) application, you can automatically send a self-registration request by email or SMS to the user who does not have any Token registered or whose Token has expired. For this, you need to enable the following setting under OpenOTP configuration :



When this setting is enable you can see the next following logs :

```
[2020-04-14 13:56:04] [192.168.3.64] [OpenOTP:92K6RNIA] User has no OTP token registered
[2020-04-14 13:56:04] [192.168.3.64] [OpenOTP:92K6RNIA] No usable login method found
[2020-04-14 13:56:04] [192.168.3.64] [OpenOTP:92K6RNIA] Sent failure response
[2020-04-14 13:56:04] [192.168.3.64] [OpenOTP:92K6RNIA] Resolved LDAP user:
CN=test,CN=Users,DC=yorcdevs,DC=eu (cached)
[2020-04-14 13:56:04] [192.168.3.64] [OpenOTP:92K6RNIA] Using Mail server 'SMTP Server'
[2020-04-14 13:56:04] [192.168.3.64] [OpenOTP:92K6RNIA] Sent self-registration request for OTP
```

A self-registration request has been sent to the end user :

User can now access SelfReg application through the one-time link to register a Token, after which they can retry the login.

2.3 Wrong TOTP/HOTP Password

Logs example :

```
[2020-04-14 14:49:19] [192.168.3.64] [OpenOTP:JOVHS8WV] New openotpStatus SOAP request
[2020-04-14 14:49:19] [192.168.3.64] [OpenOTP:JOVHS8WV] Sent status response (Ok)
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] New openotpNormalLogin SOAP request
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] > Username: test
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] > Domain: yorcdevs.eu
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] > LDAP Password: xxxxxxxx
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] > OTP Password: xxxxxx
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] > Client ID: OpenOTP
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] > Source IP: 192.168.3.64
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] > Context ID:
8054c53df2b28b4e06233591d5651c1b
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] Registered openotpNormalLogin request
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] Resolved LDAP user:
CN=test,CN=Users,DC=yorcdevs,DC=eu (cached)
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] Started transaction lock for user
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] Found user fullname: test
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] Found 1 user emails:
yoann@support.rcdevs.com
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] Found 47 user settings:
LoginMode=LDAPOTP,OTPType=TOKEN,PushLogin=No,ChallengeMode=Yes,ChallengeTimeout=90,OTPLengt
1:HOTP-SHA1-6:QN06-
T1M,DeviceType=FIDO2,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,PrefetchExpire=10,

[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] Found 3 user data:
TokenType,TokenKey,TokenState
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] Found 1 registered OTP token (TOTP)
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] Requested login factors: LDAP & OTP
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] LDAP password Ok
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] Wrong TOTP password (token #1)
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] Updated user data
[2020-04-14 14:49:28] [192.168.3.64] [OpenOTP:D34LZB76] Sent failure response
```

Possible reasons/Solutions :

- Wrong OTP password has been provided during the authentication request
- If the OTP was provided correctly but the login still fails, it probably means the Token is desynchronized. Try to resynchronize it through `WebADM GUI > <USER_ACCOUNT> > MFA Authentication Server > Resynchronize Tokens` and retry to log in.

2.4 Could not modify LDAP object / Could not set user data

Logs example

```
[2018-10-16 11:20:04] [10.10.0.3] [OpenOTP:L9RLQWCV] Could not modify LDAP object
'CN=test,CN=Users,DC=yorcdevs,DC=eu' (Insufficient access)
[2018-10-16 11:20:04] [10.10.0.3] [OpenOTP:L9RLQWCV] Could not set user data for
'CN=test,CN=Users,DC=yorcdevs,DC=eu'
```

Possible reasons/Solutions :

This issue is due to the `proxy_user` account configured in `/otp/webadm/conf/webadm.conf` not having sufficient permissions to write metadata on the user account which is trying to authenticate.

Same error can happens with a `super_admin` account connected to WebADM GUI or using WebADM Manager API which is trying to edit LDAP objects without enough permissions on the AD.

Please refer to the proxy_user or super_admins permissions documentations on MS Active Directory for instructions on how to set the permissions for `proxy_user` and `super_admins` on AD.

2.5 Account is disabled in AD

Logs example

```
[2020-04-14 13:03:31] [192.168.3.64] [OpenOTP:ON5915X4] New openotpNormalLogin SOAP request
[2020-04-14 13:03:31] [192.168.3.64] [OpenOTP:ON5915X4] > Username: test
[2020-04-14 13:03:31] [192.168.3.64] [OpenOTP:ON5915X4] > Domain: yorcdevs.eu
[2020-04-14 13:03:31] [192.168.3.64] [OpenOTP:ON5915X4] > LDAP Password: xxxxxxxx
[2020-04-14 13:03:31] [192.168.3.64] [OpenOTP:ON5915X4] > Client ID: OpenOTP
[2020-04-14 13:03:31] [192.168.3.64] [OpenOTP:ON5915X4] > Source IP: 192.168.3.64
[2020-04-14 13:03:31] [192.168.3.64] [OpenOTP:ON5915X4] > Context ID:
42d3589194785645086d7321af6bcdc6
[2020-04-14 13:03:31] [192.168.3.64] [OpenOTP:ON5915X4] Registered openotpNormalLogin request
[2020-04-14 13:03:31] [192.168.3.64] [OpenOTP:ON5915X4] Resolved LDAP user:
CN=test,CN=Users,DC=yorcdevs,DC=eu (cached)
[2020-04-14 13:03:31] [192.168.3.64] [OpenOTP:ON5915X4] Started transaction lock for user
[2020-04-14 13:03:31] [192.168.3.64] [OpenOTP:ON5915X4] Found user fullname: test
[2020-04-14 13:03:31] [192.168.3.64] [OpenOTP:ON5915X4] Found 47 user settings:
LoginMode=LDAPOTP,OTPType=TOKEN,PushLogin=Yes,ChallengeMode=Yes,ChallengeTimeout=90,OTPLeng
1:HOTP-SHA1-6:QN06-
T1M,DeviceType=FIDO2,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,PrefetchExpire=10,

[2020-04-14 13:03:31] [192.168.3.64] [OpenOTP:ON5915X4] Found 5 user data:
TokenType,TokenKey,TokenState,TokenID,TokenSerial
[2020-04-14 13:03:31] [192.168.3.64] [OpenOTP:ON5915X4] Account is disabled in AD
[2020-04-14 13:03:31] [192.168.3.64] [OpenOTP:ON5915X4] Sent failure response
```

Possible reasons/Solutions :

You can see this error when the user account is disabled in Active Directory. Enable the account on the AD and retry to log in.

2.6 Wrong LDAP password

Logs example

```
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] New openotpNormalLogin SOAP request
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] > Username: test
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] > Domain: yorcdevs.eu
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] > LDAP Password: xxxxxxxx
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] > Client ID: OpenOTP
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] > Source IP: 192.168.3.64
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] > Context ID:
60d8d725717df2ba6883c24d49e08aaa
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] Registered openotpNormalLogin request
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] Resolved LDAP user:
CN=test,CN=Users,DC=yorcdevs,DC=eu (cached)
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] Started transaction lock for user
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] Found user fullname: test
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] Found 1 user emails:
yoann@support.rcdevs.com
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] Found 47 user settings:
LoginMode=LDAPOTP,OTPType=TOKEN,PushLogin=Yes,ChallengeMode=Yes,ChallengeTimeout=90,OTPLeng
1:HOTP-SHA1-6:QN06-
T1M,DeviceType=FIDO2,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,PrefetchExpire=10,

[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] Found 5 user data:
TokenType,TokenKey,TokenState,TokenID,TokenSerial
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] Found 1 registered OTP token (TOTP)
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] Requested login factors: LDAP & OTP
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] Wrong LDAP password
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] Updated user data
[2020-04-14 17:05:18] [192.168.3.64] [OpenOTP:6VO7R9UY] Sent failure response
```

Possible reasons/Solutions :

- Wrong LDAP password has been provided during the authentication process.
- The user account is blocked on the LDAP server. Unblock the account to be able to use it.

2.7 Account has been blocked

Logs example

```
[2020-04-14 17:18:04] [192.168.3.64] [OpenOTP:XTIOD9VG] New openotpNormalLogin SOAP request
[2020-04-14 17:18:04] [192.168.3.64] [OpenOTP:XTIOD9VG] > Username: test
[2020-04-14 17:18:04] [192.168.3.64] [OpenOTP:XTIOD9VG] > Domain: yorcdevs.eu
[2020-04-14 17:18:04] [192.168.3.64] [OpenOTP:XTIOD9VG] > LDAP Password: xxxxxxxxxxx
[2020-04-14 17:18:04] [192.168.3.64] [OpenOTP:XTIOD9VG] > Client ID: OpenOTP
[2020-04-14 17:18:04] [192.168.3.64] [OpenOTP:XTIOD9VG] > Source IP: 192.168.3.64
[2020-04-14 17:18:04] [192.168.3.64] [OpenOTP:XTIOD9VG] > Context ID:
efaf77b42566518314d7b2bcc726eee7
[2020-04-14 17:18:04] [192.168.3.64] [OpenOTP:XTIOD9VG] Registered openotpNormalLogin request
[2020-04-14 17:18:04] [192.168.3.64] [OpenOTP:XTIOD9VG] Resolved LDAP user:
CN=test,CN=Users,DC=yorcdevs,DC=eu (cached)
[2020-04-14 17:18:04] [192.168.3.64] [OpenOTP:XTIOD9VG] Started transaction lock for user
[2020-04-14 17:18:05] [192.168.3.64] [OpenOTP:XTIOD9VG] Found user fullname: test
[2020-04-14 17:18:05] [192.168.3.64] [OpenOTP:XTIOD9VG] Found 1 user emails:
yoann@support.rcdevs.com
[2020-04-14 17:18:05] [192.168.3.64] [OpenOTP:XTIOD9VG] Found 47 user settings:
LoginMode=LDAPOTP,OTPType=TOKEN,PushLogin=Yes,LockTimer=0,MaxTries=3,BlockTime=0,ChallengeMo
1:HOTP-SHA1-6:QN06-
T1M,DeviceType=FIDO2,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,PrefetchExpire=10,

[2020-04-14 17:18:05] [192.168.3.64] [OpenOTP:XTIOD9VG] Found 6 user data:
TriesCount,TokenType,TokenKey,TokenState,TokenID,TokenSerial
[2020-04-14 17:18:05] [192.168.3.64] [OpenOTP:XTIOD9VG] Account has been blocked permanently after
3 tries
[2020-04-14 17:18:05] [192.168.3.64] [OpenOTP:XTIOD9VG] Sent failure response
```

Possible reasons/Solutions :

`Account has been blocked permanently after 3 tries` means that the user account is blocked in OpenOTP because of multiple login failures. This can only happen if a blocking account policy is configured under OpenOTP configuration. If `block Time` setting of OpenOTP is set to 300s, then the user can not perform any login with OpenOTP during 300s from the blocking time. The account will be automatically unblocked after the 300s.

`Max IDLE time` setting can also be the origin of a user blocking. This setting set a number of days after which an account is permanently blocked if it is not used. If this setting is enabled and the configured value is excedeed for a user account, a manual unblock by a WebADM administrator is required. The user can also unblock his account by himself through SelfDesk application.

To unblock the account through the WebADM admin portal, login on `WebADM GUI > <USER_ACCOUNT> > MFA Authentication Server > Unblock Account`.

2.8 Account already under transaction / User under transaction

Logs example

```
[2020-04-15 12:49:01] [192.168.3.64] [OpenOTP:84YBCGAX] New openotpSimpleLogin SOAP request
[2020-04-15 12:49:01] [192.168.3.64] [OpenOTP:84YBCGAX] > Username: administrator
[2020-04-15 12:49:01] [192.168.3.64] [OpenOTP:84YBCGAX] > Password: xxxxxxxx
[2020-04-15 12:49:01] [192.168.3.64] [OpenOTP:84YBCGAX] > Client ID: RadTest
[2020-04-15 12:49:01] [192.168.3.64] [OpenOTP:84YBCGAX] > Options: RADIUS,-U2F
[2020-04-15 12:49:01] [192.168.3.64] [OpenOTP:84YBCGAX] Registered openotpSimpleLogin request
[2020-04-15 12:49:01] [192.168.3.64] [OpenOTP:84YBCGAX] Resolved LDAP user:
CN=Administrator,CN=Users,DC=yorcdevs,DC=eu (cached)
[2020-04-15 12:49:01] [192.168.3.64] [OpenOTP:84YBCGAX] Resolved LDAP groups: group policy creator
owners,domain admins,enterprise admins,schema admins,denied rodc password replication group
[2020-04-15 12:49:01] [192.168.3.64] [OpenOTP:84YBCGAX] User under transaction (retrying user lock in
1 second)
[2020-04-15 12:49:02] [192.168.3.64] [OpenOTP:84YBCGAX] User under transaction (retrying user lock in
1 second)
[2020-04-15 12:49:03] [192.168.3.64] [OpenOTP:84YBCGAX] User under transaction (retrying user lock in
1 second)
[2020-04-15 12:49:04] [192.168.3.64] [OpenOTP:84YBCGAX] User under transaction (retrying user lock in
1 second)
[2020-04-15 12:49:05] [192.168.3.64] [OpenOTP:84YBCGAX] User under transaction (retrying user lock in
1 second)
[2020-04-15 12:49:06] [192.168.3.64] [OpenOTP:84YBCGAX] Account already under transaction
[2020-04-15 12:49:06] [192.168.3.64] [OpenOTP:84YBCGAX] Sent failure response
```

Possible reasons/Solutions :

To prevent OTP replay attacks, the same user account can not initiate concurrent logins with OpenOTP. When a transaction login is started for a user account, you can see the following log in `webadm.log` for the login session :

```
[2020-04-15 12:48:49] [192.168.3.64] [OpenOTP:XXQAGDA5] Started transaction lock for user
```

The transaction lock for the user is enabled until the end of login process.

If the user try to start a concurrent login during the lock period, the new login attempt will be rejected by OpenOTP with the following logs :

```
[2020-04-15 12:49:01] [192.168.3.64] [OpenOTP:84YBCGAX] User under transaction (retrying user lock in
1 second)
```

- This can happen if the user try to log in simultaneously on different systems consuming OpenOTP for authentications,
- A too short `timeout` (e.g: 10s) with a login `retry` configured between OpenOTP clients and OpenOTP server can also be the origin of that kind of issue. The OpenOTP client `timeout` should be set to 30s without push login configured and 40s with push login. 0 or 1 for `retry` setting according to what is allowed by your clients is advised.

2.9 Domain xxx not existing

Logs example

```
[2020-04-15 13:30:38] [192.168.3.62] [OpenOTP:3UFSBKBW] New openotpNormalLogin SOAP request
[2020-04-15 13:30:38] [192.168.3.62] [OpenOTP:3UFSBKBW] > Username: administrator
[2020-04-15 13:30:38] [192.168.3.62] [OpenOTP:3UFSBKBW] > Domain: yorcdevs.eu
[2020-04-15 13:30:38] [192.168.3.62] [OpenOTP:3UFSBKBW] > Client ID: ADFS19
[2020-04-15 13:30:38] [192.168.3.62] [OpenOTP:3UFSBKBW] > Source IP: 192.168.3.62
[2020-04-15 13:30:38] [192.168.3.62] [OpenOTP:3UFSBKBW] > Context ID:
xDRnnYukcsGfPuinwfXmgweA5zUppwYR
[2020-04-15 13:30:38] [192.168.3.62] [OpenOTP:3UFSBKBW] > Settings: LockTimer=0
[2020-04-15 13:30:38] [192.168.3.62] [OpenOTP:3UFSBKBW] > Options: -LDAP
[2020-04-15 13:30:38] [192.168.3.62] [OpenOTP:3UFSBKBW] Enforcing client policy: ADFS19 (matched
client ID)
[2020-04-15 13:30:38] [192.168.3.62] [OpenOTP:3UFSBKBW] Registered openotpNormalLogin request
[2020-04-15 13:30:38] [192.168.3.62] [OpenOTP:3UFSBKBW] Domain 'yorcdevs.eu' not existing
[2020-04-15 13:30:38] [192.168.3.62] [OpenOTP:3UFSBKBW] User invalid or not found
[2020-04-15 13:30:38] [192.168.3.62] [OpenOTP:3UFSBKBW] Sent failure response
```

WebADM domain configuration :

Possible reasons/Solutions :

This error occurs when the domain provided during the authentication request do not match any user domain configured in WebADM. Here the domain provided is `yorcdevs.eu`

To fix it, login on WebADM GUI > Admin > User Domain :

- If an existing domain named `Default` already exist in your configuration and the user who is trying to authenticate is part of that domain, then you can add `yorcdevs.eu` as domain Name Aliases value in your current domain. You can also rename the `Default` domain name by `yorcdevs.eu` and add `Default` as Domain Name Aliases value. Please, take into account that if you choose to rename the domain by your custom value and you don't want to put `Default` as Domain Name Alias of your domain, you need to edit each Web Applications and WebServices configuration to adjust the default domain value to the new one configured.
- If user is not part of existing domain configured, then create a new WebADM domain with a user search base including users who need to authenticate under that domain.

2.10 Domain not provided and no default domain configured

Logs example

```
[2020-04-07 10:32:55] [192.168.3.64] [Admin] Domain not provided and no default domain configured
```

Possible reasons/Solutions :

This error occurs if there is no default domain configured in WebADM and an authentication request does not include domain information. A default domain can be configured in `/opt/webadm/conf/webadm.conf`. This default domain is used for logins on the WebADM admin portal and Web Applications.

```
default_domain "yorcdevs.eu"
```

When this setting is configured, if I log in on WebADM Admin GUI in UID, OTP or U2F mode by not providing any user domain, then the default one configured in webadm.conf will be automatically used.

2.11 Server is unwilling to perform

This error can happen for multiple reasons in WebADM and comes from Active Directory. When you see this error, it means the action attempt can not be performed on the remote Active Directory.

Logs example

```
[2020-04-20 10:03:58] [192.168.3.1] [PwReset:DPIL2WTF] Could not modify LDAP object
'CN=Administrator,CN=Users,DC=yorcdevs,DC=com' (Server is unwilling to perform - 0000001F: SvcErr:
DSID-031A12D2, problem 5003 (WILL_NOT_PERFORM), data 0)
[2020-04-20 10:03:58] [192.168.3.1] [PwReset:DPIL2WTF] Could not set user password for
'CN=Administrator,CN=Users,DC=yorcdevs,DC=com' (Server is unwilling to perform)
```

```
[2020-04-20 09:58:21] [192.168.3.1] [Admin:58SR60O3] Could not modify LDAP object
'CN=RCDevs,CN=Users,DC=yorcdevs,DC=com' (Server is unwilling to perform - 0000001F: SvcErr: DSID-
031A12D2, problem 5003 (WILL_NOT_PERFORM), data 0)
```

Possible reasons/Solutions :

- There is no encryption configured on the LDAP connection between WebADM and Active Directory. Active Directory refuses to change users passwords with unencrypted LDAP. Edit `/opt/webadm/conf/servers.xml` and configure your LDAP servers with SSL or TLS encryption (restart WebADM services to changes takes effect). SSL protocol use 636 port and TLS 389. To enable SSL or TLS between WebADM and Active Directory, AD must be first configured for LDAPS support. To check if LDAPS is enabled on your DCs, you can open powershell and execute the following command :

```
PS C:\Users\Administrateur> netstat -na | Select-String "636"

  TCP    0.0.0.0:636          0.0.0.0:0            LISTENING
  TCP    [::]:636             [::]:0               LISTENING
```

As you can see, LDAPS is enabled on my DC and 636 port is listening.

You can also check from WebADM if WebADM is able to communicate with the DC on port 636. Open an SSH session with your WebADM and use telnet to check :

```
[root@webadm4 ~]# telnet 192.168.3.50 636
Trying 192.168.3.50...
Connected to 192.168.3.50.
Escape character is '^]'.
```

Port is open and reachable from WebADM Server, LDAPS can then be used.

- You tried to change a user password (through WebApps or WebADM Admin GUI), but the new password provided doesn't meet the Active Directory password policy.

2.12 Missing client certificate or subject

Logs example

```
[2020-04-21 10:41:41] [192.168.3.68] [OpenOTP] Missing client certificate or subject
```

Possible reasons/Solutions :

This error can appear if the OpenOTP setting "Require Client Certificate" is enabled and the OpenOTP client doesn't provide certificate during communications with OpenOTP.

- Issue a client certificate through WebADM PKI for your OpenOTP client and configure the certificate on your client.
- Disable the setting "Require Client Certificate" under OpenOTP configuration.

2.13 Checking LDAP proxy user access… ERROR

This error can occur when starting WebADM services.

Logs example

```
[root@webadm1 ~]# /opt/webadm/bin/webadm start
Found Trial Enterprise license (RCDEVSSUPPORT)
Licensed by RCDevs Security SA to RCDevs Support
Licensed product(s): OpenOTP,SpanKey,TiQR

Starting WebADM PKI server... Ok
Starting WebADM Session server... Ok
Starting WebADM Watchd server... Ok
Starting WebADM HTTP server... Ok

Checking server connections...
Connected LDAP server: LDAP Server (192.168.3.60)
Connected SQL server: SQL Server (192.168.3.68)
Connected PKI server: PKI Server (127.0.0.1)
Connected Mail server: SMTP Server (78.141.172.203)
Connected Push server: Push Server (91.134.128.157)
Connected Session server: Session Server (::1)
Connected License server: License Server (91.134.128.157)

Checking LDAP proxy user access... ERROR
Checking SQL database access... Ok
Checking PKI service access... Ok
Checking Mail service access... Ok
Checking Push service access... Ok
Checking License service access... Ok
```

Possible reasons/Solutions :

- Proxy_user account configured doesn't exist in the AD,
- Proxy_user password configured is incorrect,
- Proxy_user DN configured in `webadm.conf` is wrong,

You have to consider the `ldap_treebase` setting in `/otp/webadm/conf/webadm.conf`. If `ldap_treebase` setting is configured, then the treebase of your `proxy_user` must be removed (same for super_admin and WebADM containers definition).

E.g for domain name : `yorcdevs.eu`

```
ldap_treebase "dc=yorcdevs,dc=eu"
proxy_user "cn=proxy_user,cn=Users"
super_admins "cn=super_administrator,cn=Users"
config_container "cn=WebADM"
```

If you configure the `ldap_treebase` setting and keep the `ldap_treebase` value in your proxy_user definition, WebADM

will try to bind the AD with `cn=proxy_user,cn=User,dc=yorcdevs,dc=eu,dc=yorcdevs,dc=eu` as distinguished Name which will result of a failure. Remove your ldap_treebase value of your proxy_user, super_admins and container configurations.

2.14 Calling WebADM CA for certificate request signing… Failed (Could not sign certificate (rsign_sign_csr() failed))

This is a rsignd (PKI) error. Rsignd logs are available in `/opt/webadm/logs/rsignd.log`

When that kind of error happens, double-check the PKI password configured in `/opt/webadm/conf/servers.xml` and the password configured for the client in `/opt/webadm/conf/rsignd.conf`.

Configuration Server/Client example

The secret I want to configure contain specials characters : `secret&!`

`/opt/webadm/conf/servers.xml`

```
<PkiServer name="PKI Server"
      host="192.168.3.64"
      port="5000"
      secret="secret&!"
      ca_file="" />
```

`/opt/webadm/conf/rsignd.conf`

```
client {
      hostname 192.168.3.64
      secret secret&!
}
```

As you can see, the secret is correctly configured on both sides, but I can't still issue SSL certificate from WebADM GUI.

Logs example


```
[2020-04-24 09:50:36] [14342] New connection from 192.168.3.64 port 52698
[2020-04-24 09:50:36] [14342] Access refused
```

Possible reasons/Solutions :

First, specials characters like & or ! can not be used in that format in `/opt/webadm/conf/servers.xml`. That will prevent WebADM to start due to xml parsing with an error like below :

```
root@webadm1 ~]# /opt/webadm/bin/webadm restart
Stopping WebADM Session server... Ok
Stopping WebADM PKI server... Ok
Checking libudev dependency... Ok
Checking system architecture... Ok
Checking server configurations... Failed
Could not parse XML server definitions file '/opt/webadm/conf/servers.xml' (xmlParseEntityRef: no name)
```

To use that secret in servers.xml, if you're running **WebADM version 2.0.*** you can encrypt the password with pwcrypt tool :

```
[root@webadm1 ~]# /opt/webadm/bin/pwcrypt secret$!
This script allows to encrypt some sensitive WebADM configuration settings
like user passwords and encryption keys. You can also replace the cleartext
passwords and keys with encrypted values in webadm.conf and servers.xml.

Encrypted: {wcrypt}hVe0tjVhHnQlRCUDcQMIfw==
```

`{wcrypt}hVe0tjVhHnQlRCUDcQMIfw==` value can be used as secret in both `servers.xml` and `rsignd.conf` files.

If you are running **WebADM version 1.7.***, you have two possibilities :

- **You have an enterprise license** and, in that situation, you can encrypt the password with pwcrypt tool as the above procedure.
- **You don't have any enterprise license**, then you need to encode your password in xml format in `servers.xml` like below :

```
<PkiServer name="PKI Server"
      host="192.168.3.64"
      port="5000"
      secret="secret&amp;!"
      ca_file="" />
```

for `rsignd.conf`, the password must be used as below :

```
client {
      hostname 192.168.3.64
      secret secret&!
}
```

Restart WebADM services to changes takes effect and try to generate a new SSL certificate from `WebADM GUI > Admin > Issue Server or Client SSL Certificate`

You should now be able to issue client/server certificate.

2.15 RCDEvs Cloud Service access issue - Connection failed while calling LOGIN:CR_LOGIN

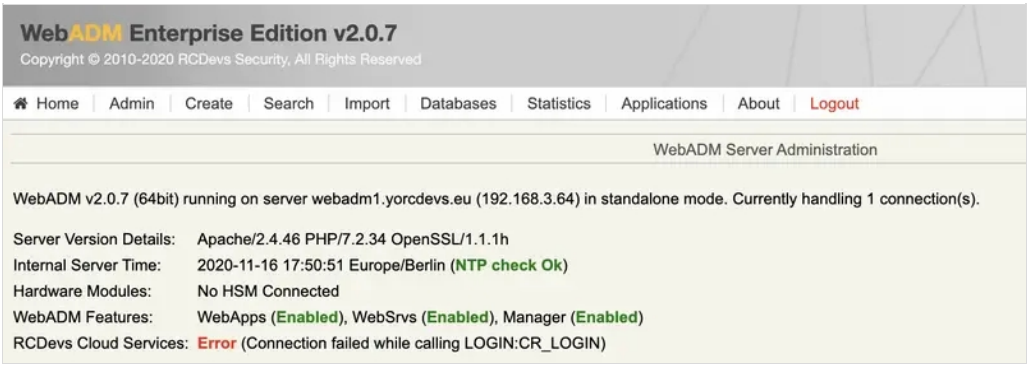
This is an error coming from WebADM (v2.x only) when RCDevs Cloud Services are enabled but can not be reach correctly. The url which must be reachable from your WebADM backends is `https://cloud.rcdevs.com`

Logs example

That error can appear during the WebADM start like below :

```
[root@webadm1 ~]# /opt/webadm/bin/webadm start
Checking libudev dependency... Ok
Checking system architecture... Ok
Checking server configurations... Ok

Found Trial license (RCDEVSSUPPORT)
Licensed by RCDevs Security SA to RCDevs Support
Licensed product(s): OpenOTP,SpanKey,TiQR

Starting WebADM PKI service... Ok
Starting WebADM Session service... Ok
Starting WebADM Watchd service... Ok
Starting WebADM HTTP service... Ok

Checking server connections...
Connected LDAP server: LDAP Server (192.168.3.60)
Connected SQL server: SQL Server (192.168.3.68)
Connected PKI server: PKI Server (127.0.0.1)
Connected Proxy server: HTTP Proxy (192.168.3.1)
Connected Session server: Session Server (::1)

Checking LDAP proxy user access... Ok
Checking SQL database access... Ok
Checking PKI service access... Ok
Checking Cloud service access... ERROR (connection failed while calling LOGIN:CR_LOGIN)
```

Or on the `Admin` tab through WebADM GUI like below :



Possible reasons/Solutions :

- WebADM server(s) can not reach `https://cloud.rcdevs.com` or is/are not able to resolve the DNS name.
- You are using an HTTP proxy between WebADM and `https://cloud.rcdevs.com`. If you are using a proxy, then be sure `https://cloud.rcdevs.com` is reachable from the proxy. The SSL connection between WebADM servers and `https://cloud.rcdevs.com` can not be broken by the proxy for SSL interception mode. Your proxy must work in "Transparent" mode.
- Check that your firewall(s) allow WebADM server(s) to reach `https://cloud.rcdevs.com`
- Your proxy server configuration on WebADM is wrong. Double-check your proxy configuration in `/opt/webadm/conf/servers.xml` file.

Tests :

To test the communications with cloud.rcdevs.com from your WebADM server(s), you can use OpenSSL from WebADM server(s) to test the SSL connectivity with RCDevs Cloud Services :

```
[root@webadm1 ~]# openssl s_client -connect cloud.rcdevs.com:443
CONNECTED(00000003)
depth=2 O = Digital Signature Trust Co., CN = DST Root CA X3
verify return:1
depth=1 C = US, O = Let's Encrypt, CN = Let's Encrypt Authority X3
verify return:1
depth=0 CN = cloud.rcdevs.com
verify return:1
---
Certificate chain
 0 s:CN = cloud.rcdevs.com
   i:C = US, O = Let's Encrypt, CN = Let's Encrypt Authority X3
 1 s:C = US, O = Let's Encrypt, CN = Let's Encrypt Authority X3
   i:O = Digital Signature Trust Co., CN = DST Root CA X3
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIFWjCCBEKgAwIBAgISBAj/h3KAZMgCCOQsXsHCDjWmMA0GCSqGSIb3DQEBCwUA
MEoxCzAJBgNVBAYTAlVTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MSMwIQYDVQQD
ExpMZXQncyBFbmNyeXB0IEF1dGhvcml0eSBYMzAeFw0yMDA5MjUyMTAwNTFaFw0y
```

MDEyMjQyMTAwNTFaMBsxGTAXBgNVBAMTEGNsb3VkLnJjZGV2cy5jb20wggEiMA0G
CSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC+qvuB7ynZ+g1hQPhojOoRPwbSKYvh
JxWCm9WYjt3zEBIPj4XvyecWPmE/unoIfu9O7R65ZKStlGIiODwx0wabLXH/xn/i
dvn5mAs0PoUBRd+AWC99RAG3TFf7Yr+vGeFqSbx+V42BcBF+W+KXIAn8TyNb8H7q
nvRvZkl6OGHJLYi9N+43E3AJoBXrZQxhQPWxAoX/mg+jtKMnqODknWkY+bIZyQno
7AIzwH0w2u26mmelDoFdQ15U/2tvGnm8orDelmDEEPSf2ovsqBoeH5vcQIU5NNCt
KisIupcr08vn9NdkFn/kJwKVGu6N2G5Mn/MIFAsmc5/0GZY3tgBGEi+NAgMBAAGj
ggJnMIICYzAOBgNVHQ8BAf8EBAMCBaAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsG
AQUFBwMCMAwGA1UdEwEB/wQCMAAwHQYDVR0OBBYEFEIZtJGnyZZlry9ZLFLdel5B
Kty0MB8GA1UdIwQYMBaAFKhKamMEfd265tE5t6ZFZe/zqOyhMG8GCCsGAQUFBwEB
BGMwYTAuBggrBgEFBQcwAYYiaHR0cDovL29jc3AuaW50LXgzLmxldHNlbmNyeXB0
Lm9yZzAvBggrBgEFBQcwAoYjaHR0cDovL2NlcnQuaW50LXgzLmxldHNlbmNyeXB0
Lm9yZy8wGwYDVR0RBBQwEoIQY2xvdWQucmNkZXZzLmNvbTBMBgNVHSAERTBDMAgG
BmeBDAECATA3BgsrBgEEAYLfEwEBATAoMCYGCCsGAQUFBwIBFhpodHRwOi8vY3Bz
LmxldHNlbmNyeXB0Lm9yZzCCAQYGCisGAQQB1nkCBAIEgfcEgfQA8gB3AAe3XBvl
fWj/8bDGHSMVx7rmV3xXlLdq7rxhOhpp06IcAAABdMdIgwkAAAQDAEgwRgIhANdk
G9lcmfJPuqG7S3fFD5/YgxJaGa3ejWrogMEDC3hDAiEA2FyrTvPJGYNmvgop2nHj
MYtjBPDzD6oczCLGGYJ2mKoAdwBvU3asMfAxGdiZAKRRFf93FRwR2QLBACkGjbII
mjfZEwAAAXTHSINgAAAEAwBIMEYCIQCw1//Q2lr2DgAJKRsSvfcaLXLsBLKp9B42
NEDwGz56ZwIhAKbIQUYn4oB9L68ECIwZ+F0vSOLXfsPPxa/s0KFfvXIbMA0GCSqG
SIb3DQEBCwUAA4IBAQCROunneQaxr/gY6q8MQ25AU02w2LvxxO0V3uoYOloWFwvr
qBgXeUa0sy8zFSUPugd5AStrkERFHOPquZBr481kz4GtEKx/96TgsUTYAUb+tJN3
VrwhmvfIB2Mw6CTRwElRL++cGUeJu/gjB65ojXDtmOmQdTH61K2IMbwk8mNCGKiI
Cny3+Itn3PtXrwkQLD/eFLVEAX/svZovHkPcfK1qUdmcPk4YgZJpcllrmtf+B1V1
2w02KWiun7vSB2EKw+gAcapwJxnZWIGz7ZrKdOTz8SE3Q4Kl6aVB9kSJYIkbstyA
bjrEnTXtkRtvvuHumR3yyVwiSl5EEiYml6tUgRaO
-----END CERTIFICATE-----
subject=CN = cloud.rcdevs.com

issuer=C = US, O = Let's Encrypt, CN = Let's Encrypt Authority X3

No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits

SSL handshake has read 3113 bytes and written 404 bytes
Verification: OK

New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 0 (ok)


```
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: 8BC21B5E13E2463DDB09CA9640BA76C34180521630CF62B30A27EA605A045A8C
    Session-ID-ctx:
    Resumption PSK:
FC21ADFF2B99BFF5EEF5BC7BF449EF556359A12ECDF7C7B99FB5F46908C8C292D919047D8A8526308D4DD

    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 300 (seconds)
    TLS session ticket:
    0000 - 81 6c 7a 6d a3 42 f2 cc-1e f8 c8 5a 82 71 9d 05   .lzm.B.....Z.q..
    0010 - 94 57 17 53 8c 7b fb 69-ed e8 83 13 c5 54 d7 6b   .W.S.{.i.....T.k
    0020 - 53 25 df 0f 4b 5f ad d1-f9 57 5f 42 3c 7a 47 2d   S%..K_...W_B<zG-
    0030 - eb 3e 98 6c 65 93 c7 90-d1 83 d1 6e dc e2 7d a6   .>.le......n..}.
    0040 - 1c 37 34 87 55 53 6c 9b-67 d4 54 fc e8 a0 30 f6   .74.USl.g.T...0.
    0050 - d2 fc f2 fd 3b 92 99 c0-5c 6f 4e 79 43 5a 9c 5e   ....;...\oNyCZ.^
    0060 - 34 eb 26 69 aa bc 8d 79-ca 5e 4f 2c 2f 5c f1 8f   4.&i...y.^O,/\..
    0070 - 6a 11 9c 10 74 12 65 8c-9c d9 d4 b3 0b 4b cb 4b   j...t.e......K.K
    0080 - 46 5d 0a a8 4b 04 13 e3-7f 91 65 c5 08 a4 ed 6d   F]..K.....e....m
    0090 - cc 4e ec e4 c1 a2 73 1d-92 8f a8 13 aa cd 39 03   .N....s.......9.
    00a0 - 67 53 f7 e7 c0 79 03 ff-bf 29 5a d4 ed 58 03 a5   gS...y...)Z..X..
    00b0 - 2f 34 ec be bd ff a2 ee-e7 73 37 8e 6b 91 db 3e   /4.......s7.k..>
    00c0 - db 82 04 08 68 6e bf da-11 5e 24 69 21 5d 5b 9f   ....hn...^$i!][.
    00d0 - 16 8a 46 09 be 23 d9 54-8f e5 6e dc b7 88 86 6b   ..F..#.T..n....k
    00e0 - 77 c9 ea 76 8a 18 65 81-b9 c5 13 e5 82 a3 d4 56   w..v..e........V
    00f0 - 43 38 72 3e 22 60 b1 e5-5e 01 45 10 4a f8 51 60   C8r>"`..^.E.J.Q`

    Start Time: 1605546287
    Timeout   : 7200 (sec)
    Verify return code: 0 (ok)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
---
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: 50FB63F81369F8E7BDF032FA97C598ACDD8FB70DD64CA29BB2C983B8142EFD73
    Session-ID-ctx:
    Resumption PSK:
706BEB47601374B788E1A1608CC30E9A2FDEC87310349431BC112FBF41E65A1A1446C7089B8F60A605B00

    PSK identity: None
```

```
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 300 (seconds)
    TLS session ticket:
    0000 - 81 6c 7a 6d a3 42 f2 cc-1e f8 c8 5a 82 71 9d 05   .lzm.B.....Z.q..
    0010 - 51 5a 06 2b 79 a0 1a f9-e8 8b ed 8b de 59 cf 44   QZ.+y........Y.D
    0020 - e1 9d 8a c3 a8 30 f1 5d-c3 12 cf 68 d2 91 e2 36   .....0.]...h...6
    0030 - 72 85 f7 16 bc 0c e4 4a-72 0f 7a 42 14 53 6d 4b   r......Jr.zB.SmK
    0040 - 52 90 29 55 0a 07 66 3b-77 e8 d5 07 b2 54 02 3c   R.)U..f;w....T.<
    0050 - c0 35 1b a9 5c 8c ea 32-81 b8 5f 1c e9 dd 82 42   .5..\..2.._....B
    0060 - 20 5d 97 d3 7a 25 28 4a-ab 61 5b fc cf 39 10 5d    ]..z%(J.a[..9.]
    0070 - c5 e1 20 54 f7 b5 41 c6-5e 3e 01 c1 9b 80 a3 90   .. T..A.^>......
    0080 - 02 1a 9e e1 bb c0 18 d8-9e a2 d4 d4 c4 32 c9 73   .............2.s
    0090 - e3 d3 24 53 fb 5f 0f d2-53 6f 14 17 84 20 eb 1c   ..$S._..So... ..
    00a0 - 61 5e 29 7c 5b 7c fd 2c-8f 2d 06 03 5a 8b 14 e4   a^)|[|.,.-..Z...
    00b0 - 7f 26 d0 91 85 7f df 0e-ef fc 4f 03 f6 ff 90 7d   .&........O....}
    00c0 - 4b 7c b1 51 c8 46 43 77-44 a3 69 65 00 03 e5 bc   K|.Q.FCwD.ie....
    00d0 - 13 e3 0c 3b 99 b2 d5 a2-1c 06 14 c3 dd 15 7d ca   ...;..........}.
    00e0 - 0c 6e 0d 81 ce 7b ed ad-b6 16 c7 fe 95 67 a3 d3   .n...{.......g..
    00f0 - 69 99 55 2f f3 f2 2a 1c-81 06 60 81 77 f9 70 8a   i.U/..*...`.w.p.

    Start Time: 1605546287
    Timeout   : 7200 (sec)
    Verify return code: 0 (ok)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
closed
```

Note

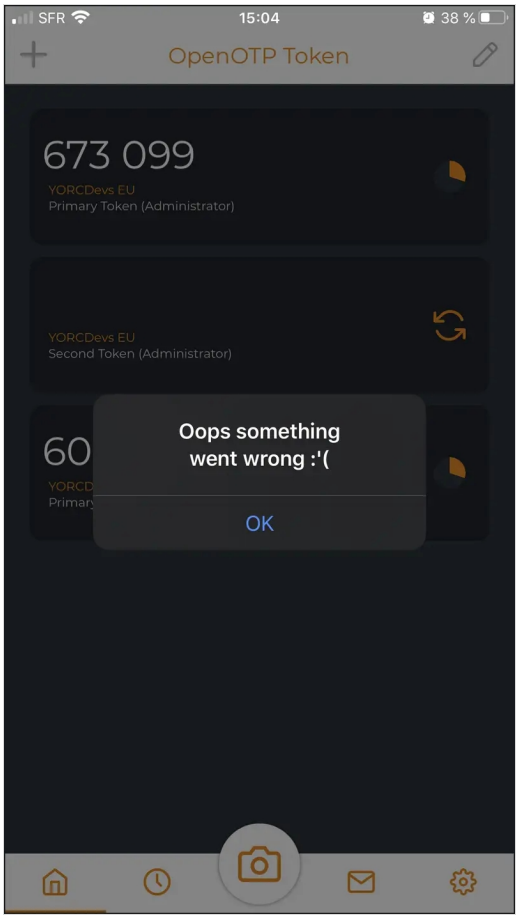
The result of the OpenSSL command may change according to the certificate renewal.

3. Push issues

3.1 Oops, something went wrong (message during push Token registration)

This error appears on OpenOTP Token application when the mobile can not contact the OpenOTP mobile endpoint URL.

Logs example



Possible reasons/Solutions :

- Check your OpenOTP mobile end point URL under OpenOTP graphical configuration.
- Check from the mobile if the mobile endpoint URL is reachable.
- DNS resolution timeout of your mobile endpoint URL can be the origin of that kind of issue if the mobile takes too much time to resolve it.

3.2 Approve/Reject timer is too short when push notification arrives at the phone and prevent you to approve the login request or push notification never arrives to phone

To be compatible with every clients integrations, it's strongly recommended to keep the default "Mobile Response Timeout" value configured to 30 seconds.

In good networks conditions, the push notification should be delivered on the phone with a timeout around 25 seconds. The notification takes between 2-5 seconds to appear on the phone when push login request is sent from OpenOTP. You can easily know when the notification has been sent by OpenOTP server with the timestamp in webadm.log :

```
[2020-04-14 15:55:36] [192.168.3.64] [OpenOTP:PT8SDGLY] Sent push notification for token #1
[2020-04-14 15:55:36] [192.168.3.64] [OpenOTP:PT8SDGLY] Waiting 28 seconds for mobile response
```

At this step of the login process, the push has been sent to the end-user's phone and OpenOTP wait 28s for push response. Without response within 28s, OpenOTP will send back a challenge-response to the client to allow end users to manually enter

their OTP.

```
[2020-04-14 15:56:04] [192.168.3.64] [OpenOTP:PT8SDGLY] Started OTP authentication session of ID b0oMbEJXS3CdcBAh valid for 90 seconds
[2020-04-14 15:56:04] [192.168.3.64] [OpenOTP:PT8SDGLY] Sent login challenge response
```

It's easy to calculate the timeout value who must appear when you receive the notification on the phone.

- Notification Timeout start to 28 secs. If notification takes 5s to arrive at the phone, then the timer should be around 23s.(if the application is not open when receiving the push login request, take into account the time it takes to start the app and popup the login request.) Then the user will have 23s to approve or reject the login before OpenOTP send back a Challenge-response and invalid the simple push request. If the notification take more time to be delivered on the end-user phone, it can comes from Apple/Google push services latencies.
- If the notification takes between 2-5 seconds to appear on the phone, and you see a timer much lower than expected (e.g. 10s), then it probably means the mobile is not well synchronize with an NTP server. Check NTP synchronization on phone and on WebADM. If the NTP desynchronization is too big (more than 28s), then the push with approve/deny will not be prompted at all on the phone because the request is already expired for the OpenOTP Token application.
- If it takes more than 5 seconds to appear on the phone, it comes from network latencies between RCDevs Push services and the mobile.
- If push notification never arrives at the phone, check 3G/4G or Wi-Fi connectivity of your phone.
- If either the phone time or OpenOTP server time is inaccurate, the timer will be incorrectly calculated. Please enable network time on the phone and configure NTP sync on the server.

4.3 Could not send push notification to AND:xxx or IOS:xxx

Logs example

```
[Tue Apr 14 16:09:54.637723 2020] [192.168.3.64] [OpenOTP:H3A5WF97] Could not send push notification to 'AND:eVjRDk5VrZg:APA91bF1v...'
[Tue Apr 14 16:10:18.716921 2020] [192.168.3.64] [OpenOTP:JTIRFVI1] Could not send push notification to 'IOS:76e2f09e06f002b40a027...'
```

Possible reasons/Solutions :

- If this error appear, check if WebADM server is able to communicate with RCDevs push services. RCDevs Push services are configured in `/opt/webadm/conf/servers.xml`
- If RCDevs Push services are reachable from your WebADM servers, then the issue comes from the device ID used to push notification on the mobile. Try to register a new token and perform a login to see if the problem persist.

3.4 Invalid mobile registration response

Logs example

```
[2020-03-20 11:39:29] [195.218.27.170] [OpenOTP:WNXPLREL] Invalid mobile registration response (missing pushid parameter)
```

This error appears when the mobile response for a Push Token registration is invalid. During Push Token registration, after scanning the QRCode with OpenOTP Token, the mobile communicates with the OpenOTP mobile endpoint URL and forwards following information regarding the device/Token :

- Session registration ID : Must match with a session registration currently open in OpenOTP.
- Secret : The final secret is not in the QR code but negotiated between server and token with a key exchange protocol.
- PIN Code: PIN code is sent for validation if enrolment is protected by a PIN
- OS Type : IOS or AND.
- Push ID : Unique identifier used to push notifications on the phone.
- Serial: Metadata stored on user account,
- Phone model: Metadata stored on user account.
- Offset: Only used for HOTP tokens to synchronize the counter between token and server.

Possible reasons/Solutions :

- iPhone or Android device is jail broken or rooted For Android :
- Google services are not running/installed on the Android device
- Google API call to retrieve the mobile Push ID doesn’t return any value for that device. Push login can not be used in these conditions.

3.5 Connected Push server: ERROR (no server available)

This error can occur when starting WebADM services.

Logs example

```
[root@webadm1 ~]# /opt/webadm/bin/webadm restart
Stopping WebADM HTTP server... Ok
Stopping WebADM Watchd server.... Ok
Stopping WebADM Session server... Ok
Stopping WebADM PKI server... Ok
Checking libudev dependency... Ok
Checking system architecture... Ok
Checking server configurations... Ok

Found Trial Enterprise license (RCDEVSSUPPORT)
Licensed by RCDevs Security SA to RCDevs Support
Licensed product(s): OpenOTP,SpanKey,TiQR

Starting WebADM PKI server... Ok
Starting WebADM Session server... Ok
Starting WebADM Watchd server... Ok
Starting WebADM HTTP server... Ok

Checking server connections...
Connected LDAP server: LDAP Server (192.168.3.60)
Connected SQL server: SQL Server (192.168.3.68)
Connected PKI server: PKI Server (127.0.0.1)
Connected Mail server: SMTP Server (78.141.172.203)
Connected Push server: ERROR (no server available)
Connected Session server: Session Server (::1)
Connected License server: License Server (91.134.128.157)

Checking LDAP proxy user access... Ok
Checking SQL database access... Ok
Checking PKI service access... Ok
Checking Mail service access... Ok
Checking License service access... Ok
```

Possible reason/Solution :

- WebADM can not communicate with RCDevs Push services. Check with telnet if push.rcdevs.com and destination port are reachable from WebADM server(s). If not, then check your firewall.

3.6 Checking Push service access… ERROR

This error can occur when starting WebADM services.

Logs example

```
[root@webadm1 ~]# /opt/webadm/bin/webadm restart
Stopping WebADM HTTP server... Ok
Stopping WebADM Watchd server..... Ok
Stopping WebADM Session server... Ok
Stopping WebADM PKI server... Ok
Checking libudev dependency... Ok
Checking system architecture... Ok
Checking server configurations... Ok

Found Trial Enterprise license (RCDEVSSUPPORT)
Licensed by RCDevs Security SA to RCDevs Support
Licensed product(s): OpenOTP,SpanKey,TiQR

Starting WebADM PKI server... Ok
Starting WebADM Session server... Ok
Starting WebADM Watchd server... Ok
Starting WebADM HTTP server... Ok

Checking server connections...
Connected LDAP server: LDAP Server (192.168.3.60)
Connected SQL server: SQL Server (192.168.3.68)
Connected PKI server: PKI Server (127.0.0.1)
Connected Mail server: SMTP Server (78.141.172.203)
Connected Push server: Push Server (91.134.128.157)
Connected Session server: Session Server (::1)
Connected License server: License Server (91.134.128.157)

Checking LDAP proxy user access... Ok
Checking SQL database access... Ok
Checking PKI service access... Ok
Checking Mail service access... Ok
Checking Push service access... ERROR
Checking License service access... Ok
```

Possible reasons/Solutions :

After connections checks during WebADM services start, WebADM will try to perform an authentication to RCDevs push services. The authentication is done by different ways if your are an enterprise or a freeware customer.

Freeware:

Freeware users must have a freeware license to communicate with RCDevs cloud services. Once freeware license is created, you receive an email with a link to download your license.key file which must be located in `/opt/webadm/conf/` folder.

Enterprise:

Your enterprise license is used for authentication and nothing more needs to be configured. You should have a configuration like

below :

```
<PushServer name="Push Server"
      host="push.rcdevs.com"
      port="7000"
      user=""
      password=""
      ca_file="" />
```

New licenses are pushed every hour on RCDevs licenses servers. Your license needs to be pushed on RCDevs licenses servers first in order to be used with RCDevs Push services. You can encountered this issue if your license has been generated by RCDevs Sales team at 3:10 pm for example and your tried to use it before 4:00 pm. If the problem persists after 1 hour, contact RCDevs support.

4. License issues

Please, have a look on the License Server documentation for any known issues.

5. OpenOTP Token issues

RCDevs provided an update of OpenOTP Token application in December 2021 for Android and iOS. We reached 100% of the rollout for Android which mean the update is available for all Android devices.

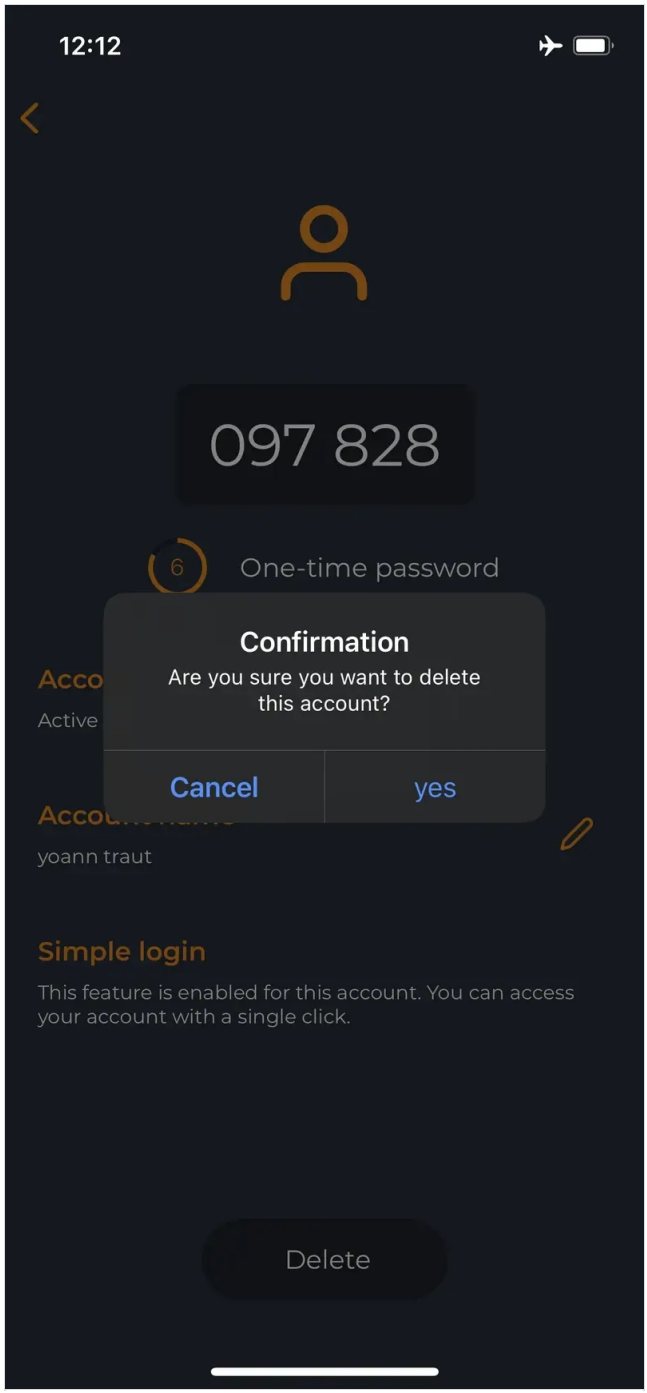
For iOS only 5% of the devices has been updated and the rollout has been stopped until bugs we encountered be fixed by RCDevs.

That update is a major migration from Titanium Framework to React and unfortunately, some issues can be encountered post migration. The issues and possible solutions are listed below.

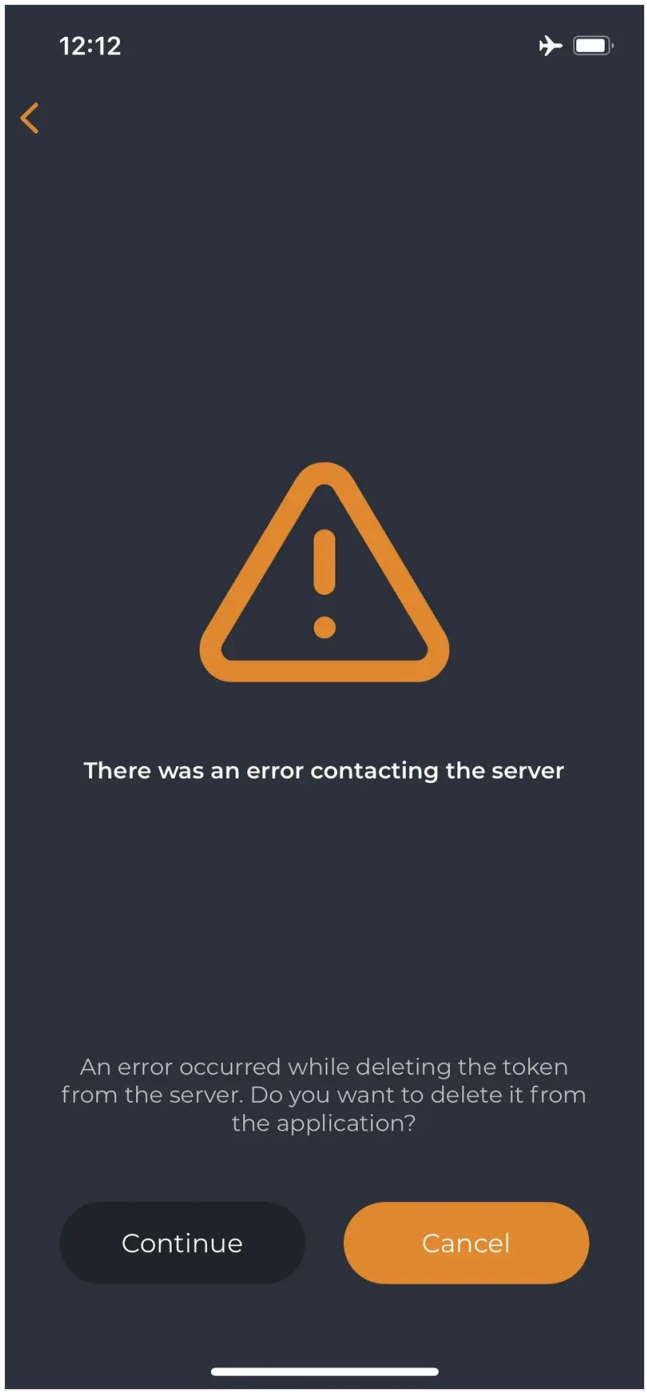
5.1 Duplicate entries for same Token

It appears on some devices, after the update of the application, Tokens are duplicated. This issue prevent push login to work properly.

If you entered that scenario, and you are using Push login to authenticate on systems, the solution to that problem is to disable all network connections (4G, Wifi) and remove the duplicated token entry. It doesn't matter which entry of the duplicated token you remove. Click on one duplicated Token and then press `Delete` button. A confirmation screen appears, press `Yes` :



When you remove Push Tokens with no network connection, a warning like below will appear. Click `Continue` button.



Repeat the operation for all duplicated tokens. Once the duplicated entries are removed, the Push functionality should be back for the corresponding token(s). You can re-enable network on your phone and try to log in with Push notification.

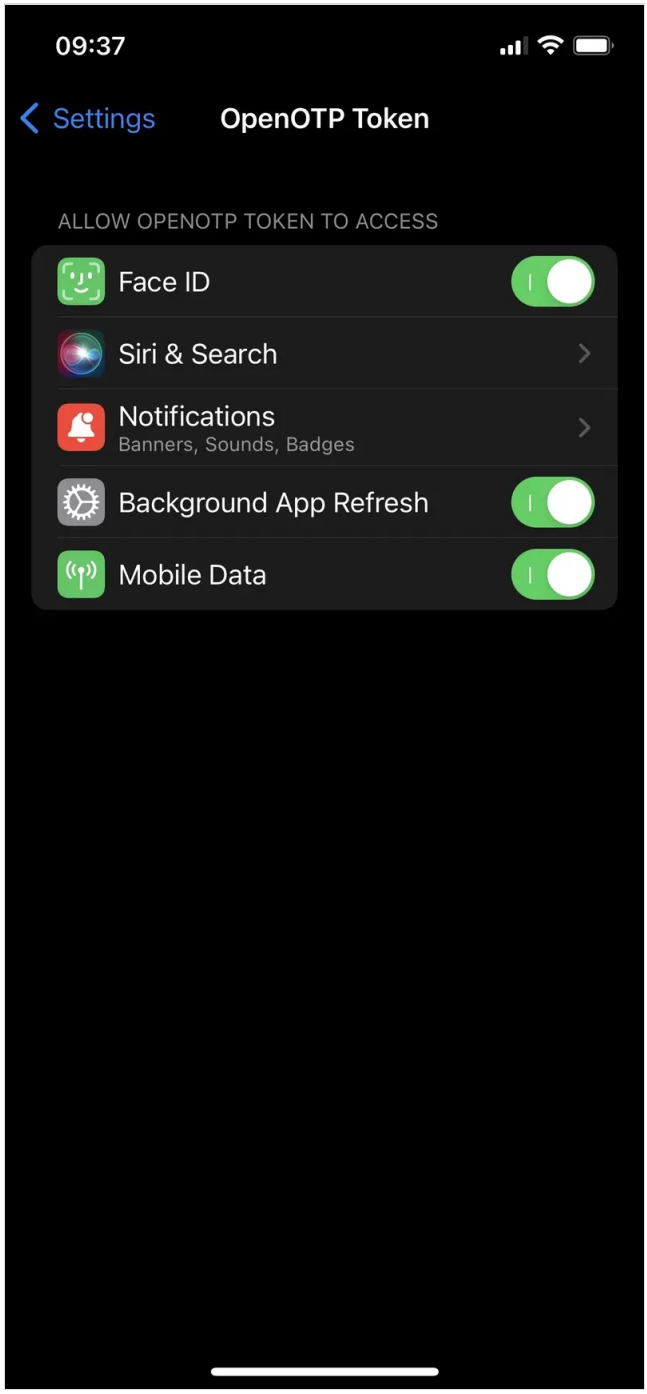
5.2 Prompted for PIN but user do not remember the PIN previously configured

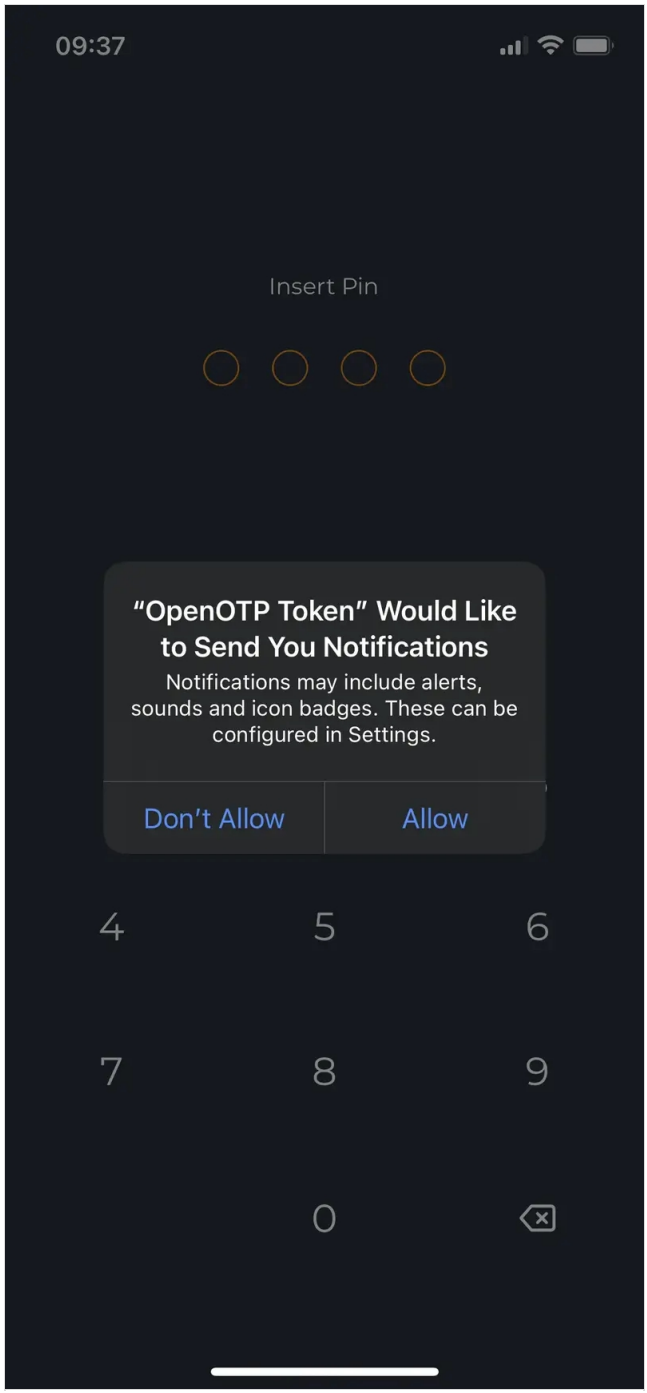
The migration from Titanium to React involves the application protection enabled even if the user disabled it in the past. That setting value could not be kept by the migration process due to technical limitations. It appears that even after uninstall and reinstall the application, the old PIN is still configured. RCDevs is working on that problem.

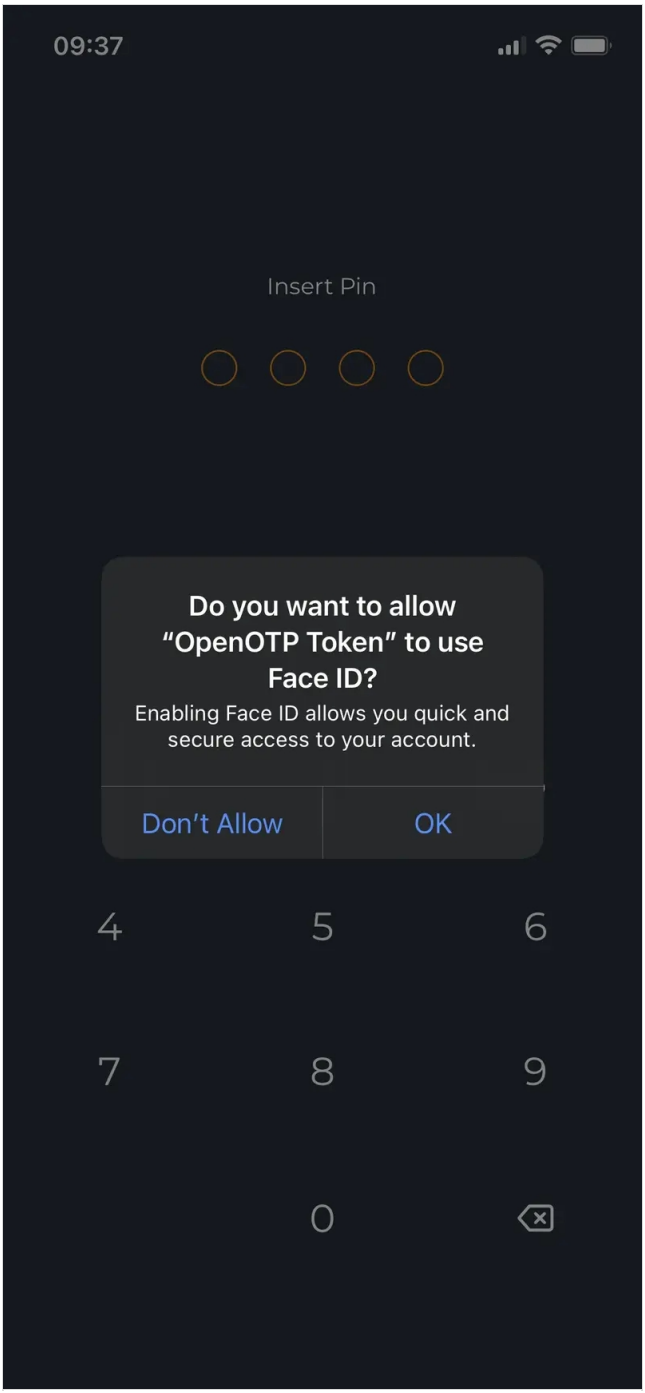
There is 2 scenarios for that issue and to solve it:

- The first scenario is, for devices which support biometric protection, enable the biometric feature of your phone and unlock the application with biometric check. In that case, PIN code should be overriden by biometric check. To check that the

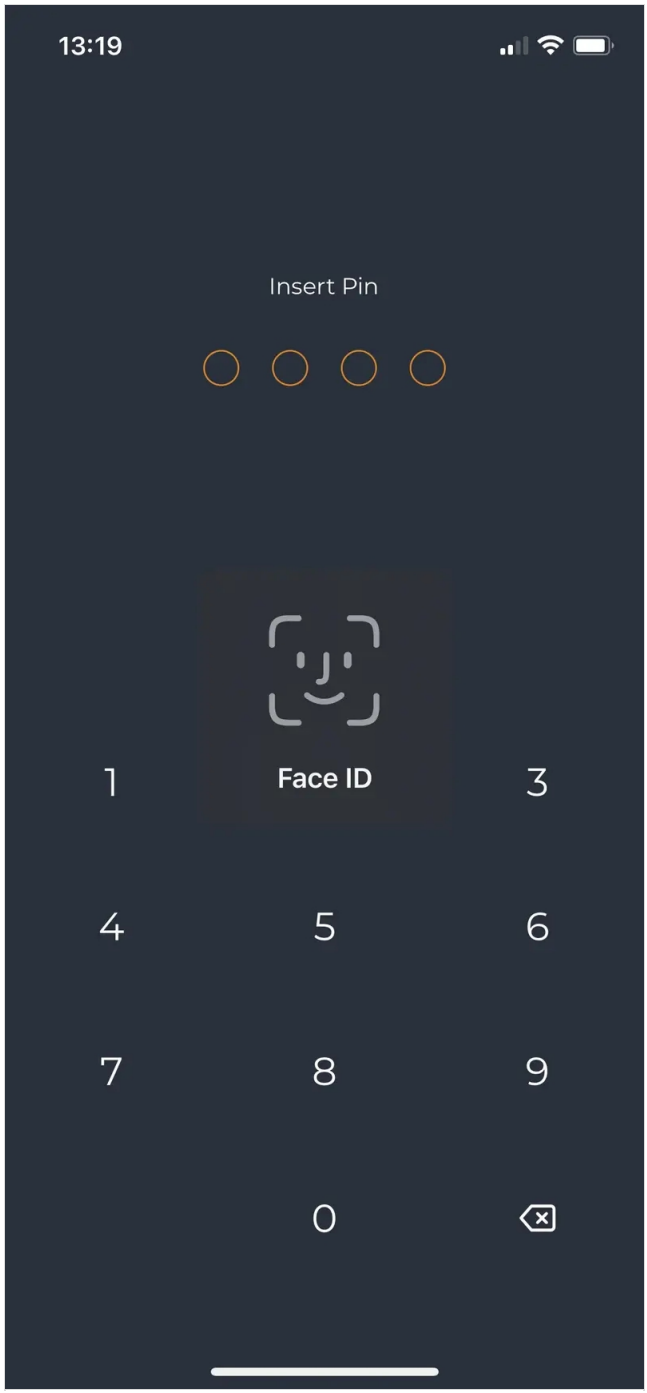
biometric feature is allowed to be used by OpenOTP token, go on `Settings` > `OpenOTP Token` and you will see biometric authorization (here Face ID). Take care that feature is enabled. Kill the application and start it again. You should be prompted for biometric unlock. For iOS users which do not have Face ID but have Touch ID, it seems the behavior is a bit different and there is no authorization to allow the app to use the Touch ID. It is allowed by default. Same procedure can be used on Android but Android devices should not be impacted by this.




09:37
Insert Pin
"OpenOTP Token" Would Like to Send You Notifications
Notifications may include alerts, sounds and icon badges. These can be configured in Settings.
Don't Allow
Allow
4
5
6
7
8
9
0



The biometric unlock is prompted.



- For Android users which are experiencing that issue and which have biometric support on their phone, if biometric has been enabled for the app, then they should be able to open the app and unlock it via the biometric feature of your phone. If biometric feature is enabled on your phone but has not been specifically enabled for that OpenOTP Token application, enable it for OpenOTP Token application. If user do not have biometric feature on their phone, then they should remove all data and cache information of OpenOTP Token application, uninstall the application, reinstall the application.

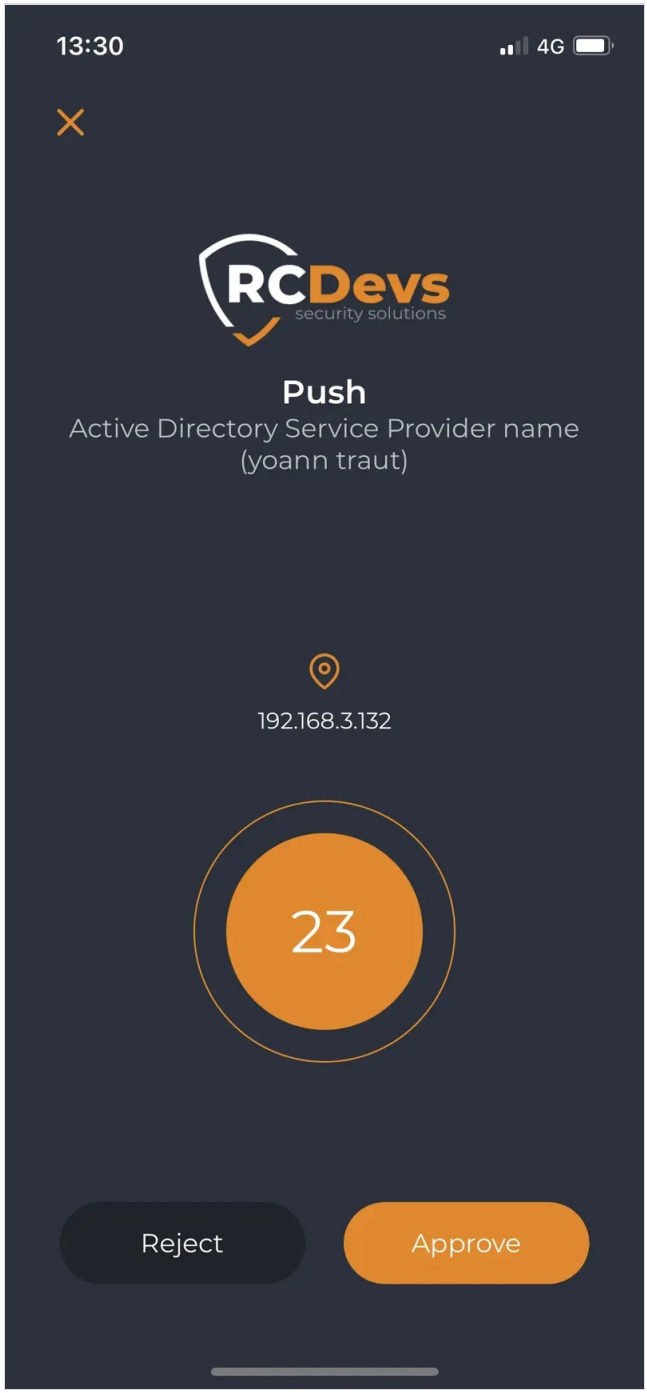
If you removed and re-installed the application (version 1.5.3 build 62479804), you should not be prompted for the PIN at the first start.

5.3 Push responses not working anymore

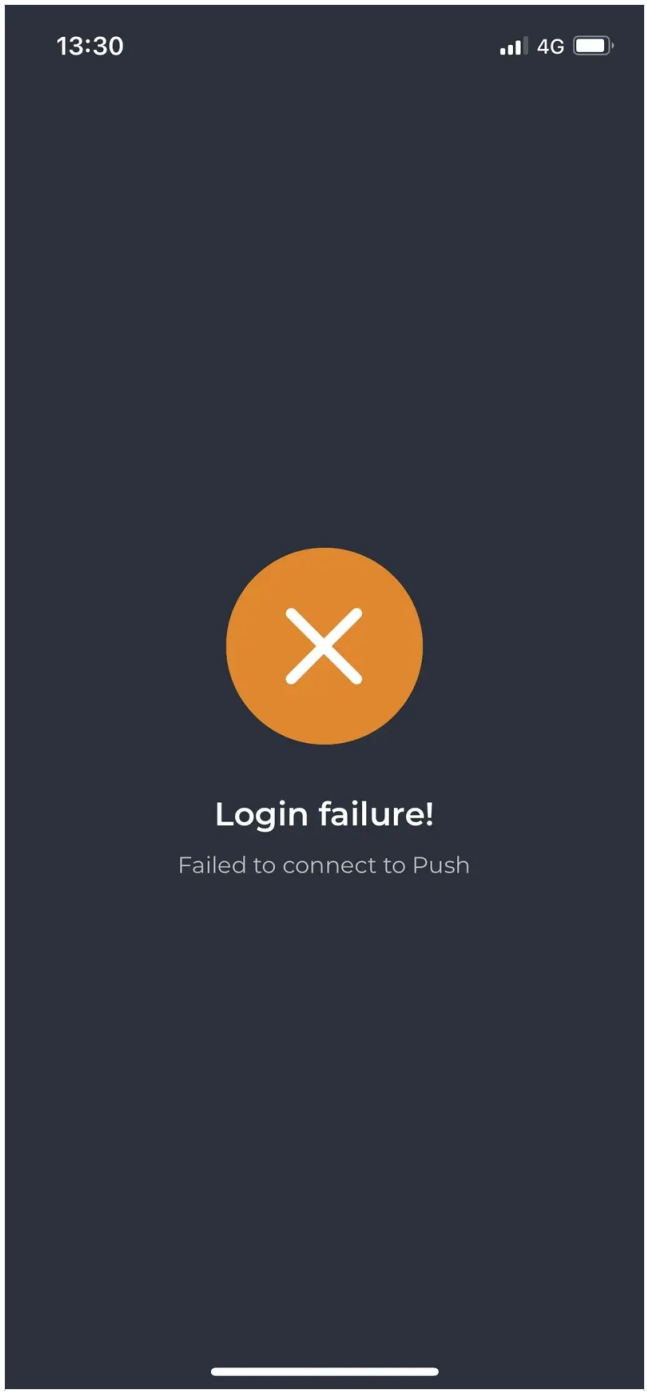
5.3.1 Wrong Reverse Proxy/Web Application Firewall configuration

This issue can happen for customers which are not using WebADM Publishing Proxy (WAProxy) and are using another reverse proxy/WAF to forward mobile push responses to the OpenOTP mobile endpoint URL.

If user correctly receive the push notifications as below:



but when they approve the login or signature request, they have a failure like below:



It is probably due to your Web Application Firewall or reverse proxy which is filtering push responses and which do not accept the new parameters with last version of the application.

Found on the following documentation, all parameters which can be forwarded in Push responses and user agent used by iOS and Android for the last version.

Extra note:

For OPNSense appliances, bot protection is dropping mobile Push responses for Android. Have a look on the following github article. Disable the bot protection feature has restored Push responses forwarding.

If the frontend fails to forward the response to WebADM/OpenOTP servers for any reason, it should respond to the mobile with an HTTP code of either 400 (Bad Request) or 403 (Forbidden). A successful termination of the request is communicated to the

mobile using HTTP 200 by WebADM/WAProxy.

5.3.2 Push response, token enrollment and badging operations not working from iOS devices only

If you are not using a publicly trusted certificate for the mobile endpoint URL, SSL connections with iOS devices may be rejected.

To address this issue, you have two options:

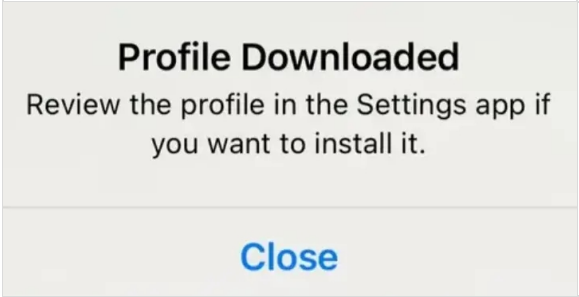
- Change the certificate presented by the mobile endpoint URL to a publicly trusted certificate.
- Alternatively, install the Certificate Authority (CA) file responsible for issuing the certificate used for the mobile endpoint URL on iOS devices.

If the certificate has been issued by WebADM PKI service (Rsignd) then you can download the WebADM CA Certificate from `https://webadm_server_address/cacert`.

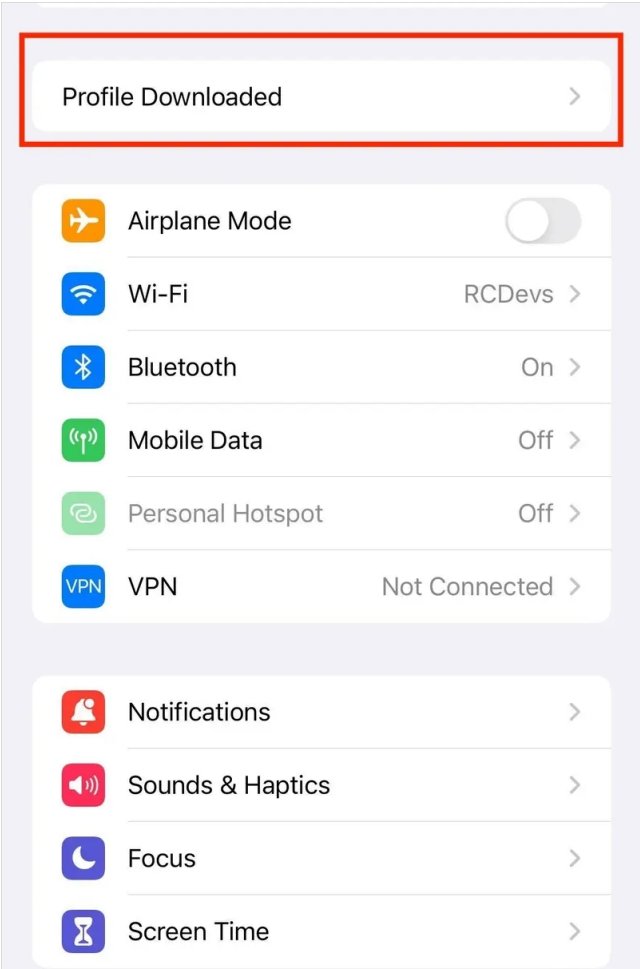
Deploy it on iOS devices through an MDM if your devices are managed this way. Else, copy the CA certificate on the iOS devices and follow the next section.

Installing the Profile

1. Click on the CA Certificate copied on your iOS device then you will have the following message:



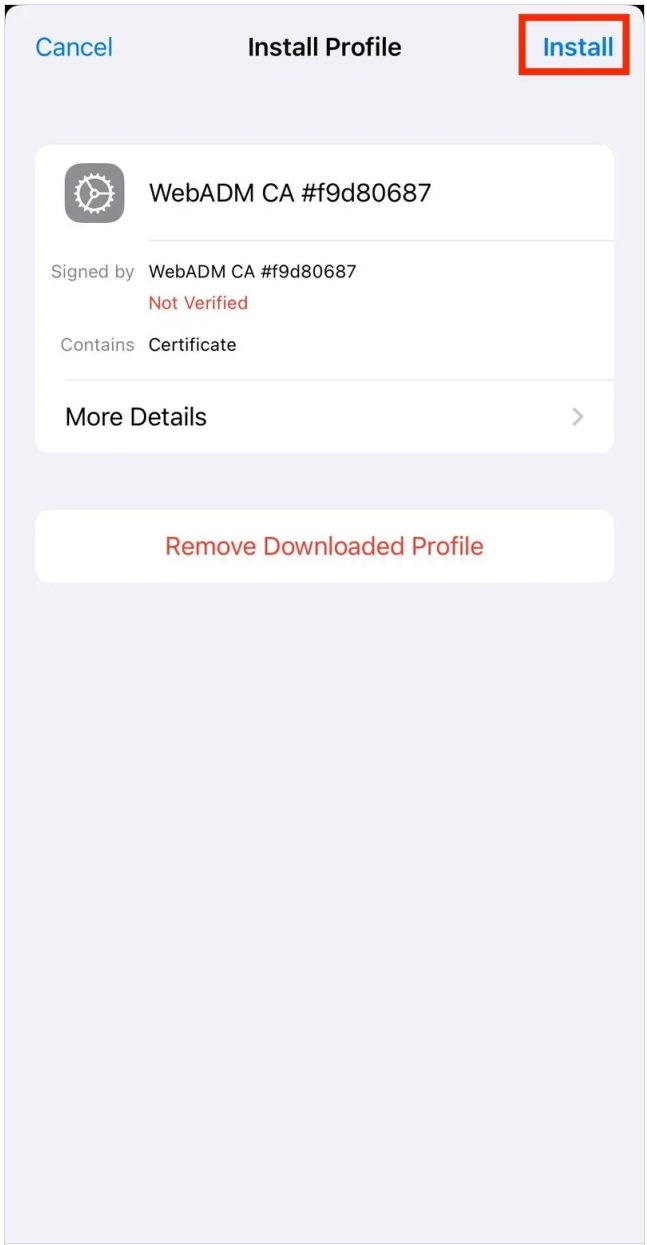
2. Now, you can open Settings menu of you iOS device and you will see something like below:



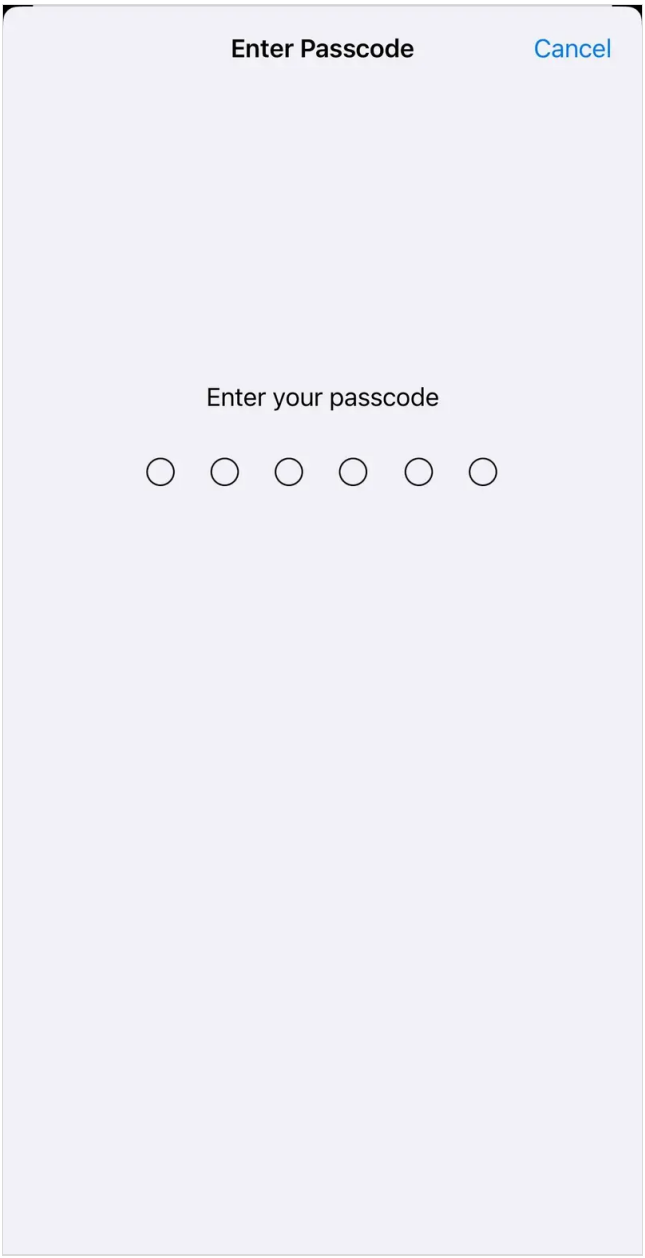
3. Underneath the User details for the device, tap Profile Downloaded.

The Install `Profile` screen will be displayed.

4. In the top right corner, tap `Install`.

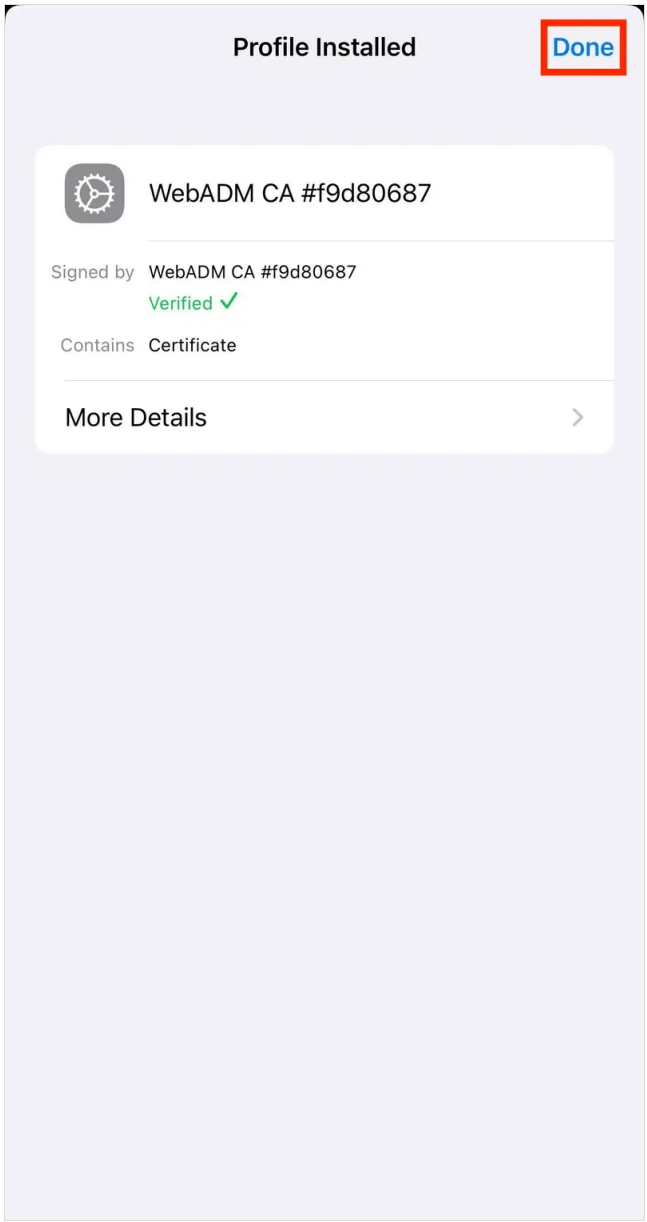


If the iOS device has a passcode set, the device will prompt you to enter it. Enter the passcode.



5. A certificate warning will be displayed. Tap Install. If a second prompt is displayed, tap Install again.

After the Certificate Warning screen, Tap `Done` button.

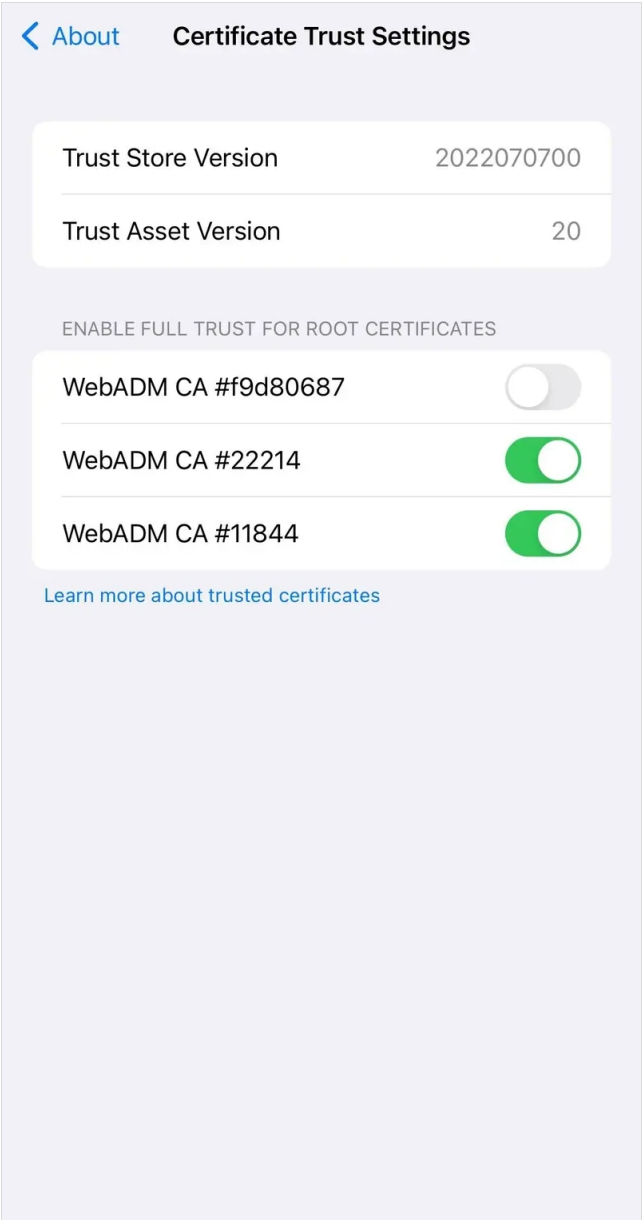


Right now, the certificate is instelled but not trusted yet. Before the certificate can be used as intended, it must be trusted by the iOS device.

On the device, go to Settings > General > About > Certificate Trust Settings.

The installed Root Certificates will be displayed in a section entitled “Enable Full Trust for Root Certificates.”

There is a slide button next to each certificate.



Tap the slide button next to the certificate you just installed.

A confirmation box will be displayed. Tap `Continue` The certificate is trusted. On some devices, it can take few minutes to be fully operationnal and a reboot may be required.

6. ___(nothing)

If you have nothing in the webadm.log, it means the request does not arrive in OpenOTP.

Verify that the OpenOTP webservice port 8443 is reachable from the authentication clients, you can check if the client traffic arrives with *tcpdump* command: `tcpdump -i any port 8080 or port 8443` during the authentication.

The next step is to continue troubleshooting in the adapter component (*radiusd* / *ldproxy*) or client (*Windows Credential Provider* , *ADFS Plugin* etc.).

For a RADIUS authentication with *radiusd*, see the troubleshooting chapter in Radius Bridge Manual

For a ldap authentication with *ldproxy*, we can verify with `tcpdump -i any port 389 or port 636`.

If the result is empty, we check that the port is open from the client. We can use `telnet <webadm server> <port number>`.

If it doesn't respond you have to check firewall rules (don't forget the local firewall on the server) and routing.