RCDEVS VPN SERVER (MFAVPN)

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.

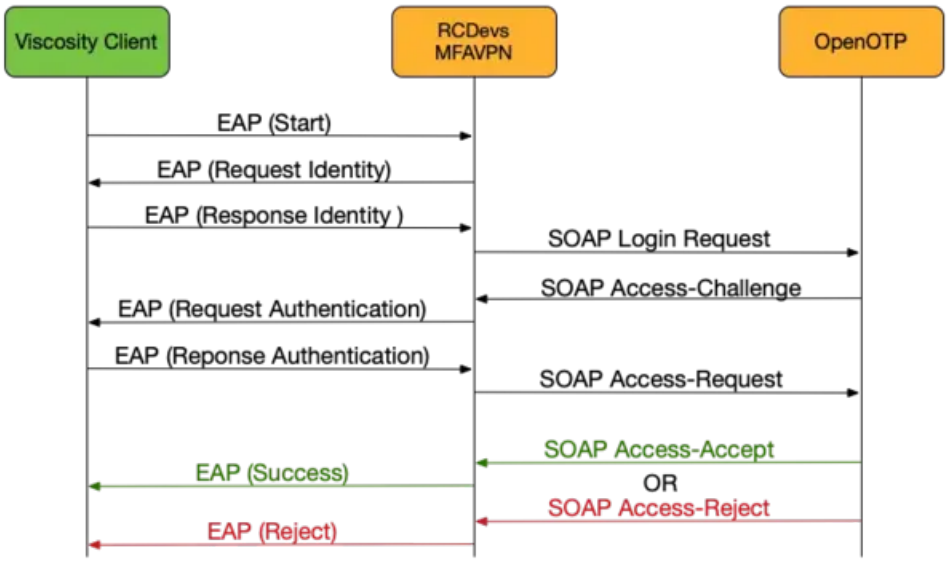
WebADM and OpenOTP are trademarks of RCDevs. All further trademarks are the property of their respective owners.

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

RCDevs VPN Server (MFAVPN)

OpenVPN MFAVPN

1. Overview



This document is an installation guide for the MFA VPN provided by RCDevs. Hence, the installation or configuration of WebADM, including token registration is not covered in this guide. For installation and usage guides of WebADM and OpenOTP, please refer to the RCDevs WebADM Installation Guide and the RCDevs WebADM Administrator Guide available through the RCDevs online documentation Website.

2. Installation of MFA VPN

On a RedHat, CentOS or Fedora system, you can use our repository, which simplifies updates. Add the repository:

```
yum install https://repos.rcdevs.com/redhat/base/rcdevs_release-1.1.1-1.noarch.rpm
```

You are now able to install RCDevs packages on your system:

```
yum install mfavpn
```

On a Debian and Ubuntu system, you can use our repository, which simplifies updates. Add the repository:

```
wget https://repos.rcdevs.com/debian/base/rcdevs-release_1.1.1-1_all.deb
apt-get install ./rcdevs-release_1.1.1-1_all.deb
```

Update apt cache:

```
apt-get update
```

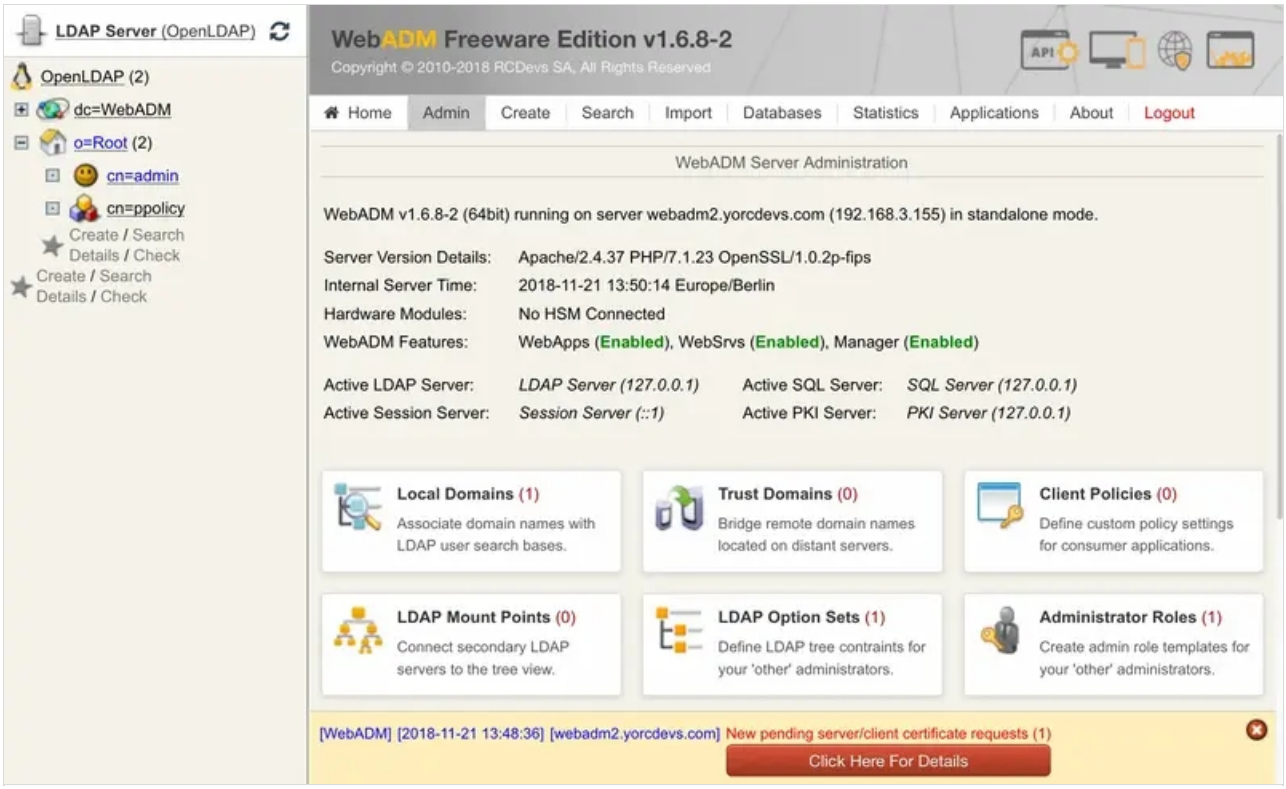
You are now able to install RCDevs packages on your system:

```
apt-get install mfavpn
```

Setup script is running:

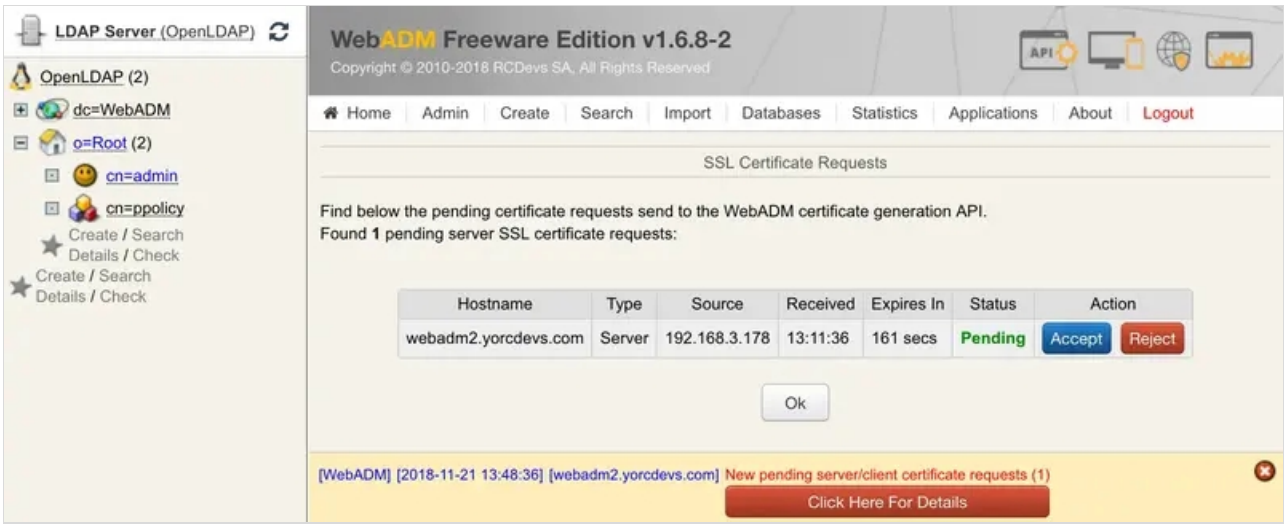
```
[root@webadm1 conf]# /opt/mfavpn/bin/setup
The fully qualified domain name of this server is 'mfavpn.yorcdevs.com'.
Please press [ENTER] to confirm or type another one:
Enter one of your running WebADM node IP or hostname: 192.168.3.54
Registered hostname is 'webadm1.yorcdevs.com'. Would you like to use it as client id (y/n)? [N]: y
Do you want to register MFA VPN Server logrotate script (y/n)? [Y]: y
Do you want MFA VPN Server to be automatically started at boot (y/n)? [Y]: y

    Primary OpenOTP service URL is: 'https://192.168.3.54:8443/openotp/'
    Secondary OpenOTP service URL is: 'https://192.168.3.55:8443/openotp/'
    Use 'webadm1.yorcdevs.com' as client id: 'YES'
    Register MFA VPN Server logrotate script: 'YES'
    MFA VPN Server must be automatically started at boot: 'YES'

Do you confirm (y/n)?: y

Applying MFA VPN Server settings from default configuration files... Ok
Generating diffie-hellman key file for daemon 'openvpn'...
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time.....................................
.....................................................................................Ok
Generating ta.key (tls-auth) file for daemon 'openvpn'... Ok
Retrieving WebADM CA certificate... Ok
The setup needs now to request a signed 'openvpn' certificate.
This request should show up as pending in your WebADM interface and an administrator must accept it.
Waiting for approbation...
```

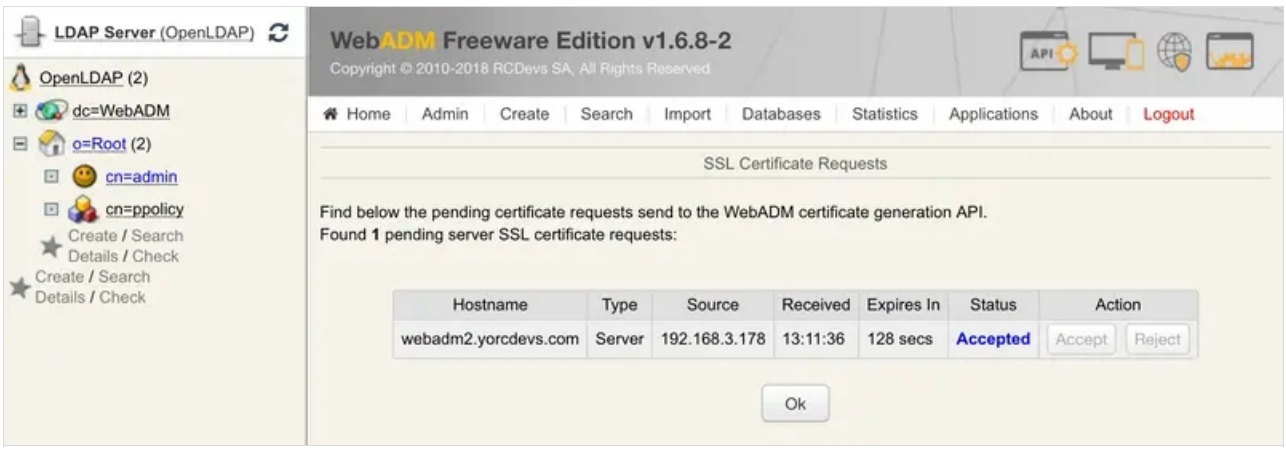
A certificate should be generated for OpenVPN. So at this step, you have to go on the WebADM Administrator GUI to approve the certificate request.



After logging on the WebADM GUI, you will show a red button at the end of the page. Please, click on it.

You will have a certificate request pending…





Click on the `Accept` button to generate the certificate, and the setup script will continue.

```
Waiting for approbation... Ok
Updating entry 'client_id' in file '/opt/mfavpn/conf/ovpnauthd.conf'... Ok
Registering MFA VPN Server service...Registering MFA VPN Server service... Ok
Adding logrotate script... Ok

MFA VPN Server has successfully been setup.
```

The installation is now complete. Installation folder is `/opt/mfavpn/`.

3. RCDevs MFA VPN Configuration Files

Every configuration files are located in `/opt/mfavpn/conf/`.

3.1 ovpnauthd.conf

In this file, you can reconfigure WebADM URLs, enable/disable U2F support or set a client ID to point to a client policy. Other settings can be kept by default.

```
#-#-#-#
#
#  ovpnauthd's configuration file.
#
      #-#-#-#
      #
      #  A location where to store the daemon's log file.
      #  Note that at the very early stage (when the daemon started but did not read yet this configuration
file),
      #  logs are sent to the standard output. Anyway, since the launcher script uses a redirection, you
won't even see them.
      #  Don't forget to adapt also file '/opt/mfavpn/lib/logrotate' if for any reason you decide to change
the default path...
      #
            log_file /opt/mfavpn/logs/ovpnauthd.log
      #
      #
      #-#-#-#


      #-#-#-#
      #
      #  A location where to store the daemon's process ID file.
      #
            pid_file /opt/mfavpn/temp/ovpnauthd.pid
      #
      #
```

```
#-#-#-#


#-#-#-#
#
#  A CA file is required in order to trust OTP servers the daemon will send requests to.
#
      ca_file  /opt/mfavpn/conf/ca.crt
#
#
#-#-#-#


#-#-#-#
#
#  A list containing the address of your WebADM servers.
#  It must contain one or items to let the daemon know where to send authentication requests.
#
      server_urls {
            url1 https://192.168.3.155:8443/openotp/
   }
#
#
#  How ovpnauthd will relay a request to the WebADM backend.
#   - "balanced" means the request will be balanced between server 1 and server 2 in a round-robin
fashion.
#   - "ordered" means server 2 is kept as a hot spare in case the primary server stops answering
requests properly.
#
      #server_policy       Ordered
#
#
# When two servers are configured, ovpnauthd can check the server statuses at regular
# intervals by trying TCP socket connections. The status_cache is the polling interval
# between 10 and 600 seconds.
      status_cache        30
#
#
#-#-#-#


#-#-#-#
#
#  The default domain name passed to OpenOTP backends when the client entered a username only.
#  This prevents WebADM server to apply any default domain configured on its own side.
#
      #default_domain_name Default
#
#
```

```
	#  Tells ovpnauthd how to extract 'domain' and 'username' from the username string entered
	#  in order to send them both separately to OpenOTP backends.
	#
		#domain_separator    \
	#
	#
	#-#-#-#


	#-#-#-#
	#
	#  When U2F login is enabled and the user login mode is set to LDAPU2F or LDAPMFA,
	#  a U2F authentication challenge is used (overriding OTP methods if present).
	#  U2F for OpenVPN is supported by the Viscosity client only!
	#
		u2f_support        Yes
	#
	#
	#-#-#-#


	#-#-#-#
	#
	#  The client identifier to be sent to OpenOTP servers along authentication requests.
	#  This allows applying per client contextual policies on the WebADM server while running an
authentication workflow.
	#
		client_id          MFAVPN
	#
	#
	#-#-#-#

	#-#-#-#
	#
	#  The SOAP request TCP timeout is by default 30.
	#  Just keep it as it unless you really understand all the possible consequences a change could have.
	#
		#soap_timeout       30
	#
	#
	#-#-#-#
#
#
#-#-#-#
```

3.2 openvpn.conf

This is the OpenVPN configuration file. You have to configure it to start your OpenVPN server. For more information about the

configuration of OpenVPN, please refer to the OpenVPN official documentation.

My OpenVPN configuration is very simple:

- I have defined a dev tun interface to create a routed IP tunnel.
- I have configured 2 routes who will be pushed to the client to access my different networks.
- I have set my DNS server and domain who will be pushed to the client for the names resolution and the domain resolution for Windows client.
- Other settings are kept by default.

```
############################################################
# Sample OpenVPN 2.0 config file for                       #
# multi-client server.                                     #
#                                                          #
# This file is for the server side                         #
# of a many-clients <-> one-server                         #
# OpenVPN configuration.                                   #
#                                                          #
# OpenVPN also supports                                    #
# single-machine <-> single-machine                        #
# configurations (See the Examples page                    #
# on the web site for more info).                          #
#                                                          #
# This config should work on Windows                       #
# or Linux/BSD systems.  Remember on                       #
# Windows to quote pathnames and use                       #
# double backslashes, e.g.:                                #
# "C:\\Program Files\\OpenVPN\\config\\foo.key"            #
#                                                          #
# Comments are preceded with '#' or ';'                    #
############################################################

# Which local IP address should OpenVPN
# listen on? (optional)
;local a.b.c.d

# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
# on the same machine, use a different port
# number for each one.  You will need to
# open up this port on your firewall.
port 1194

# TCP or UDP server?
;proto tcp
proto udp
```

```
# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
# Use "dev tap0" if you are ethernet bridging
# and have pre-created a tap0 virtual interface
# and bridged it with your ethernet interface.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the TUN/TAP interface.
# On non-Windows systems, you can give
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Network topology
# Should be subnet (addressing via IP)
# unless Windows clients v2.0.9 and lower have to
# be supported (then net30, i.e. a /30 per client)
# Defaults to net30 (not recommended)
topology subnet

# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.8.0.1. Comment this line out if you are
# ethernet bridging. See the man page for more info.
server 10.8.0.0 255.255.255.0

# Push routes to the client to allow it
# to reach other private subnets behind
# the server.  Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
push "route 192.168.2.0 255.255.255.0"
push "route 172.16.8.0 255.255.255.0"
;push "route 192.168.20.0 255.255.255.0"

# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
```

```
# or bridge the TUN/TAP interface to the internet
# in order for this to work properly).
;push "redirect-gateway def1 bypass-dhcp"

# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
# The addresses below refer to the public
# DNS servers provided by opendns.com.
push "dhcp-option DNS 192.168.3.50"
;push "dhcp-option DNS 208.67.220.220"
;push "dhcp-option WINS 208.67.220.223"
push "dhcp-option DOMAIN yorcdevs.com"

# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
;client-to-client

# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120

# Enable compression on the VPN link and push the
# option to the client (2.4+ only, for earlier
# versions see below)
;compress lz4-v2
;push "compress lz4-v2"

# For compression compatible with older clients use comp-lzo
# If you enable it here, you must also
# enable it in the client config file.
;comp-lzo

# The maximum number of concurrently connected
# clients we want to allow.
;max-clients 100

# The persist options will try to avoid
# accessing certain resources on restart
```

```
# that may no longer be accessible because
# of the privilege downgrade.
persist-key
persist-tun

# Set the appropriate level of log
# file verbosity.
#
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
verb 3

# Silence repeating messages.  At most 20
# sequential messages of the same message
# category will be output to the log.
;mute 20

# Notify the client that when the server restarts so it
# can automatically reconnect.
explicit-exit-notify 1

#
reneg-sec 0
```

After editing both files, please restart RCDevs MFA VPN services:

```
/opt/mfavpn/bin/mfavpn restart
```

Your MFA VPN is now ready to work!

4. Generate an End-user Package

4.1 Creation of the Package

A script is available to generate an end user package who will contain OpenVPN configuration file, Certificate authority file and the key.

To run this script, please execute the following command:

```
/opt/mfavpn/bin/clientpkg user

  adding: user/ (stored 0%)
  adding: user/ca.crt (deflated 25%)
  adding: user/ta.key (deflated 40%)
  adding: user/user.ovpn (deflated 54%)
```

An output file named `user.zip` will be created. Give this folder to your user.

4.2 Modification to enable FIDO authentication

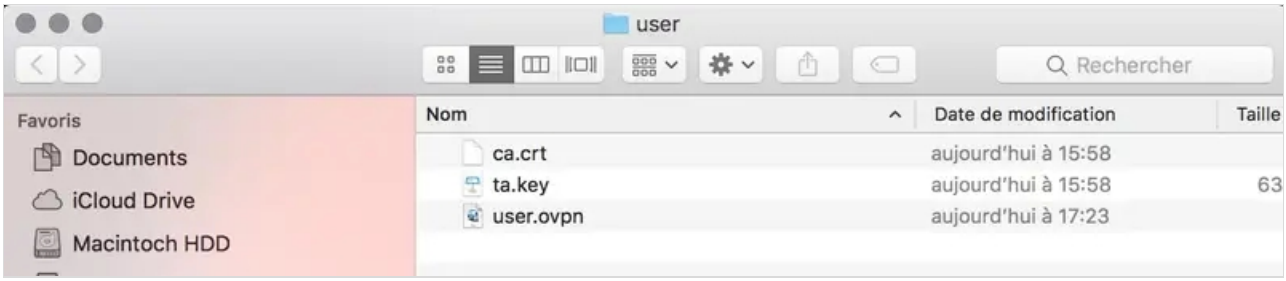
The modification in this part have to be done only to be able to authenticate with FIDO keys.

You will have to edit the file `user.ovpn` generated in the package previously. There, you will have to add the 2 following viscosity lines at the beginning :

```
##############################################
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server.     #
#                                            #
# This configuration can be used by multiple #
# clients, however each client should have   #
# its own cert and key files.                #
#                                            #
# On Windows, you might want to rename this  #
# file so it has a .ovpn extension           #
##############################################

#viscosity FIDOServerOrigin rcdevs.com
#viscosity U2FURIScheme none

# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client
```

5. Import VPN Configuration File in Viscosity or OpenVPN Client

After copying the zip file on your client machine, you can extract it.

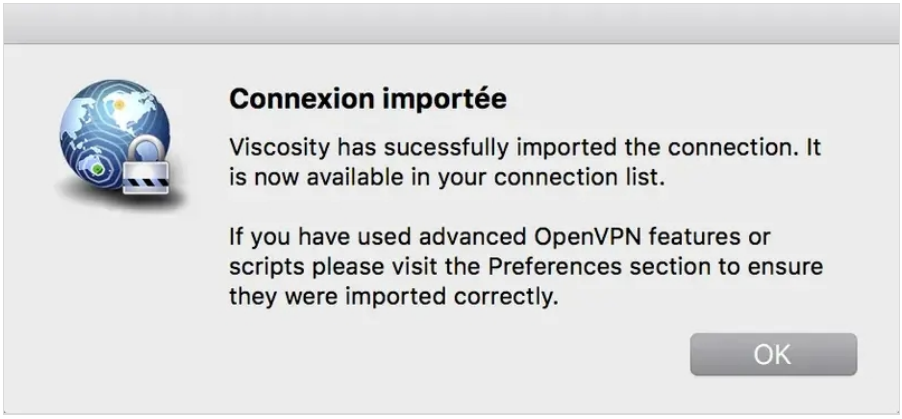


On my side, I use Viscosity as a VPN client. Viscosity is fully compatible with RCDevs MFA VPN server and OpenOTP for the U2F authentication.

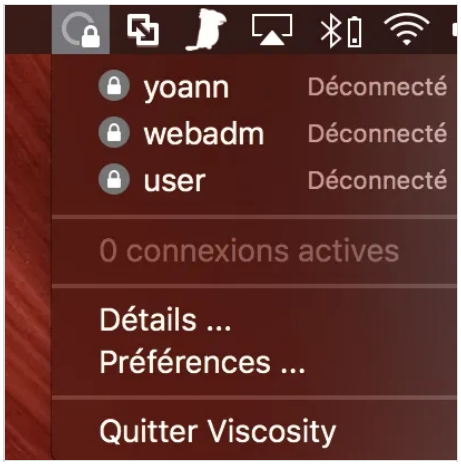
Note

Currently, Viscosity client is the only VPN client able to manage U2F authentication.

To import the OpenVPN client configuation, you just have to double-click on the `.ovpn` file.

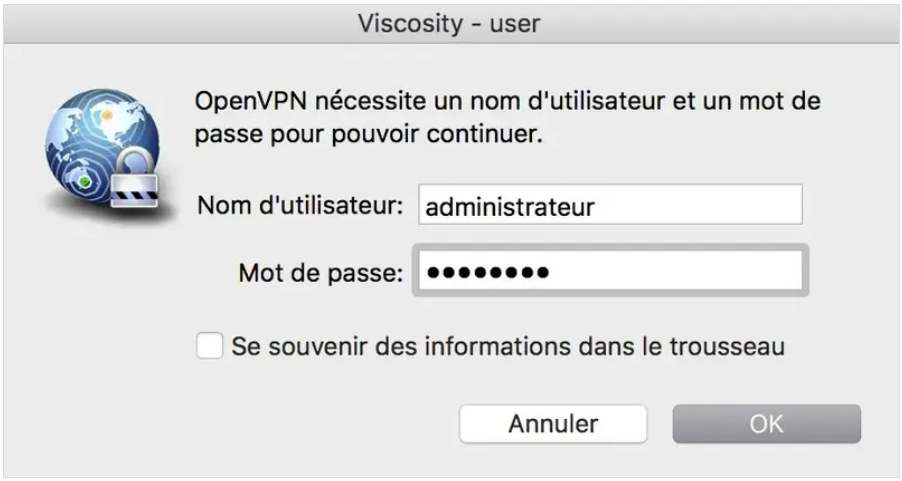


My new VPN configuration is now imported in Viscocity client.

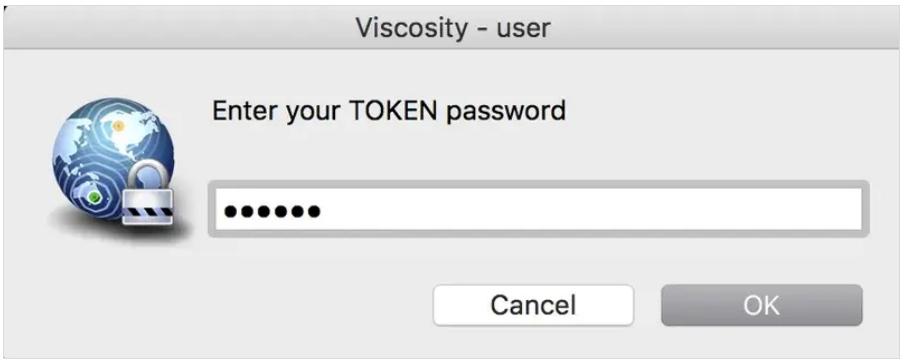


5.1 Login Test with an OTP

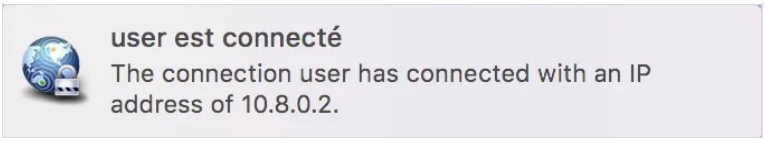
I click on the user connection to initiate the VPN connection, and I'm prompted to enter my LDAP credentials:

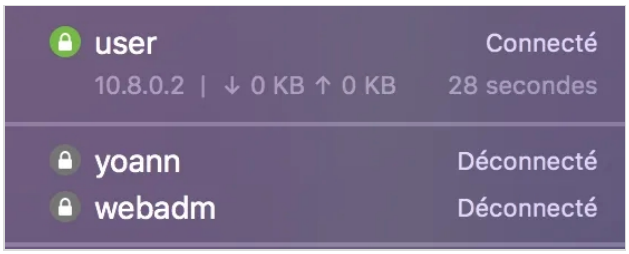


I press `OK` and on the next screen, I'm prompted for the OTP:



I enter my OTP password and press `OK`. I'm now successfully logged through RCDevs MFA VPN with an OTP.



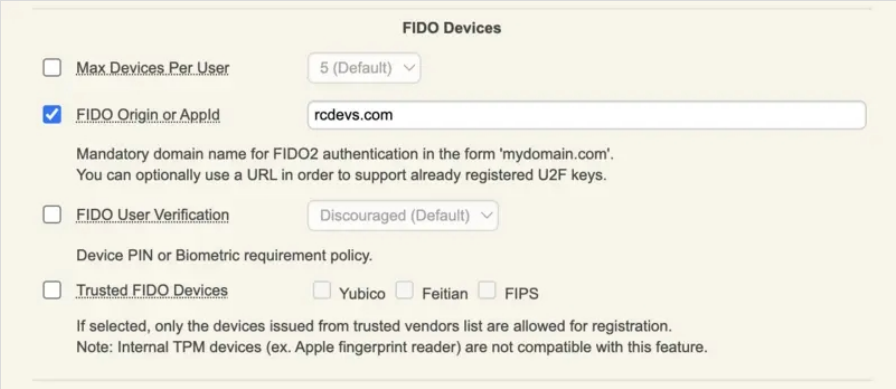


5.2 Login Test with FIDO Key

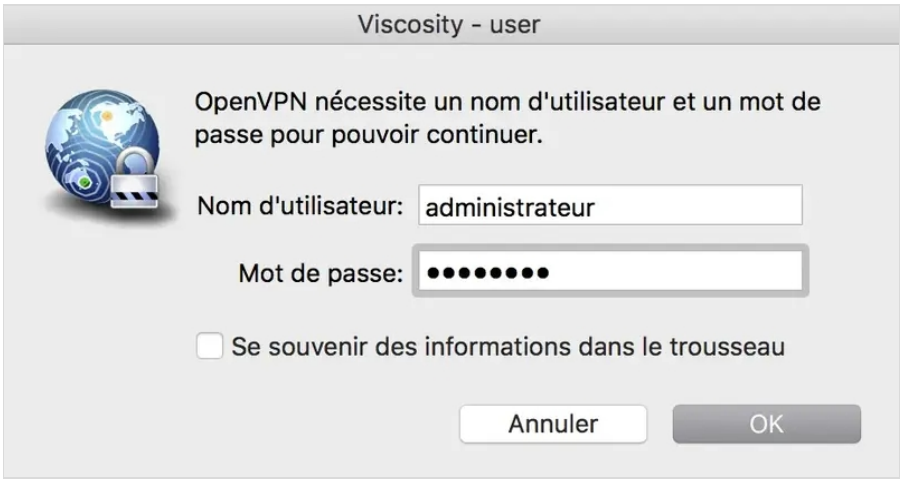
To use RCDevs MFA VPN with a FIDO key, the setting `support_u2f` should be set to `yes` in `/opt/mfavpn/conf/ovpnauthd.conf` file. By default, this setting is set to `yes`.

You also have to configure the login mode = LDAPMFA or LDAPU2F in OpenOTP and have the following proper FIDO configuration:

› FIDO Origin or AppId : rcdevs.com (must match with the parameter #viscosity `FIDOServerOrigin` previsouly set)



When FIDO configuration and FIDO key enrollment are done, I click on the user connection in Viscosity to initiate the VPN connection, and I'm prompted to enter my LDAP credentials:



I press `Ok` and on the next screen, I'm prompted to press my FIDO key:

I press my FIDO key already plugged on my laptop, and I'm now connected through RCDevs MFA VPN.

6. Authentications Logs

6.1 OTP Login Logs

[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] > Username: administrateur
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] > Password: xxxxxxxx
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] > Client ID: MFAVPN
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] > Source IP: 192.168.3.254
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] Registered openotpSimpleLogin request
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] Resolved LDAP user:
CN=Administrateur,CN=Users,DC=yorcdevs,DC=com (cached)
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] Resolved LDAP groups: propri\xc3\xa9taires
cr\xc3\xa9ateurs de la strat\xc3\xa9gie de groupe,admins du domaine,administrateurs de
l\xe2\x80\x99entreprise,administrateurs du sch\xc3\xa9ma,administrateurs,utilisateurs du bureau
\xc3\xa0 distance,groupe de r\xc3\xa9plication dont le mot de passe rodc est refus\xc3\xa9
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] Started transaction lock for user
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] Found user language: EN
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] Found 1 user mobiles: +33xxxxxxxx
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] Found 1 user emails: support@rcdevs.com
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] Found 4 user certificates
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] Found 37 user settings:
LoginMode=LDAPOTP,OTPType=TOKEN,OTPLength=6,ChallengeMode=Yes,ChallengeTimeout=90,MobileTim
1:HOTP-SHA1-6:QN06-
T1M,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,LastOTPTime=300,ListChallengeMode=

[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] Found 12 user data:
LoginCount,RejectCount,LastOTP,ListInit,ListState,TokenType,TokenKey,TokenState,TokenID,Device1Name,D

[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] Last OTP present (valid until 2018-03-12
10:54:57)
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] OTP List present (2/25 passwords used)
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] Found 1 registered OTP token (TOTP)
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] Requested login factors: LDAP & OTP
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] LDAP password Ok
[2018-03-12 10:50:51] [192.168.3.55] [OpenOTP:1GXC35AG] Challenge required
[2018-03-12 10:50:52] [192.168.3.55] [OpenOTP:1GXC35AG] Sent push notification for token #1
[2018-03-12 10:50:52] [192.168.3.55] [OpenOTP:1GXC35AG] Started OTP challenge session of ID
0Dy12B8P32ElaO5p valid for 90 seconds
[2018-03-12 10:50:52] [192.168.3.55] [OpenOTP:1GXC35AG] Sent challenge response
[2018-03-12 10:51:09] [192.168.3.55] [OpenOTP:1GXC35AG] New openotpChallenge SOAP request
[2018-03-12 10:51:09] [192.168.3.55] [OpenOTP:1GXC35AG] > Username: administrateur
[2018-03-12 10:51:09] [192.168.3.55] [OpenOTP:1GXC35AG] > Session: 0Dy12B8P32ElaO5p
[2018-03-12 10:51:09] [192.168.3.55] [OpenOTP:1GXC35AG] > OTP Password: xxxxxx
[2018-03-12 10:51:09] [192.168.3.55] [OpenOTP:1GXC35AG] Registered openotpChallenge request
[2018-03-12 10:51:09] [192.168.3.55] [OpenOTP:1GXC35AG] Found challenge session started 2018-03-12
10:50:51
[2018-03-12 10:51:09] [192.168.3.55] [OpenOTP:1GXC35AG] Started transaction lock for user
[2018-03-12 10:51:09] [192.168.3.55] [OpenOTP:1GXC35AG] PUSH password Ok (token #1)
[2018-03-12 10:51:09] [192.168.3.55] [OpenOTP:1GXC35AG] Sent stop notification for token #1
[2018-03-12 10:51:09] [192.168.3.55] [OpenOTP:1GXC35AG] Updated user data
[2018-03-12 10:51:09] [192.168.3.55] [OpenOTP:1GXC35AG] Sent success response

6.2 FIDO Login Logs

```
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] New openotpSimpleLogin SOAP request
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] > Username: administrateur
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] > Password: xxxxxxxx
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] > Client ID: MFAVPN
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] > Source IP: 192.168.3.156
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] Registered openotpSimpleLogin request
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] Resolved LDAP user:
CN=Administrateur,CN=Users,DC=yorcdevs,DC=com (cached)
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] Resolved LDAP groups: propriétaires
créateurs de la stratégie de groupe,admins du domaine,administrateurs de l'entreprise,administrateurs
du schéma,administrateurs,utilisateurs du bureau à distance,groupe de réplication dont le mot de passe
rodc est refusé
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] Started transaction lock for user
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] Found user language: EN
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] Found 1 user mobiles: +33658506140
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] Found 1 user emails: support@rcdevs.com
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] Found 4 user certificates
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] Found 38 user settings:
LoginMode=LDAPMFA,OTPType=TOKEN,OTPLength=6,ChallengeMode=Yes,ChallengeTimeout=90,MobileTim
1:HOTP-SHA1-6:QN06-
T1M,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,LastOTPTime=300,ListChallengeMode=

[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] Found 12 user data:
LoginCount,RejectCount,ListInit,ListState,TokenType,TokenKey,TokenState,TokenID,TokenSerial,Device1Nam

[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] OTP List present (2/25 passwords used)
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] Token #1 (TOTP) is disabled
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] User has no OTP token registered
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] Found 1 registered U2F device
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] Requested login factors: LDAP & U2F
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] LDAP password Ok
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] Challenge required
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] Started U2F challenge session of ID
f0FKOqM20x2XxqkF valid for 90 seconds
[2018-03-12 10:40:58] [192.168.3.55] [OpenOTP:IDPCMSSZ] Sent challenge response
[2018-03-12 10:41:13] [192.168.3.55] [OpenOTP:IDPCMSSZ] New openotpChallenge SOAP request
[2018-03-12 10:41:13] [192.168.3.55] [OpenOTP:IDPCMSSZ] > Username: administrateur
[2018-03-12 10:41:13] [192.168.3.55] [OpenOTP:IDPCMSSZ] > Session: f0FKOqM20x2XxqkF
[2018-03-12 10:41:13] [192.168.3.55] [OpenOTP:IDPCMSSZ] > U2F Response: 417 Bytes
[2018-03-12 10:41:13] [192.168.3.55] [OpenOTP:IDPCMSSZ] Registered openotpChallenge request
[2018-03-12 10:41:13] [192.168.3.55] [OpenOTP:IDPCMSSZ] Found challenge session started 2018-03-12
10:40:58
[2018-03-12 10:41:13] [192.168.3.55] [OpenOTP:IDPCMSSZ] Started transaction lock for user
[2018-03-12 10:41:13] [192.168.3.55] [OpenOTP:IDPCMSSZ] U2F response Ok (device #1)
[2018-03-12 10:41:13] [192.168.3.55] [OpenOTP:IDPCMSSZ] Updated user data
[2018-03-12 10:41:13] [192.168.3.55] [OpenOTP:IDPCMSSZ] Sent success response
```

7. Multiple Server Setup for High Availability

If you need load balancing or high availability, you can install multiple MFAVPN servers. After installing the first server, use the backup script to create a backup package.

```
[root@mfavpn1 ~]# /opt/mfavpn/bin/backup mfavpn_backup
Are you sure you want to backup MFA VPN Server (y/n)? y
Starting backup procedure...
Adding required file 'conf/ca.crt'... Ok
Adding required file 'conf/openvpn.conf'... Ok
Adding required file 'conf/openvpn.crt'... Ok
Adding required file 'conf/openvpn.key'... Ok
Adding required file 'conf/ovpnauthd.conf'... Ok
Adding optional file 'temp/'... Ok
Adding optional file 'logs/'... Ok
Compressing backup file... Ok

MFA VPN Server backup created in 'mfavpn_backup.gz'.
```

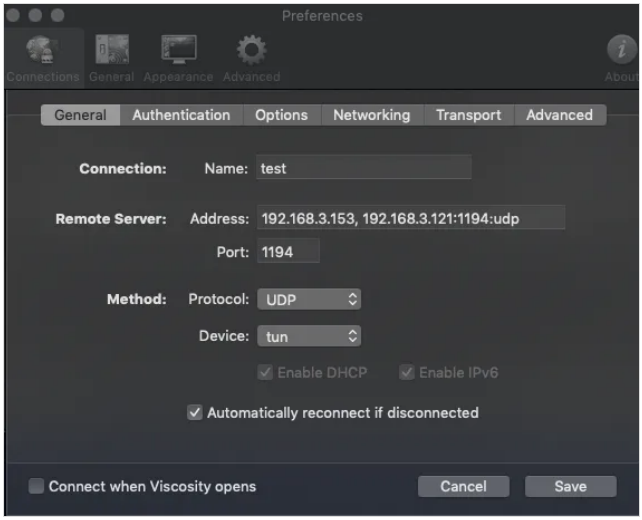
Install the second server the same way as the first one, copy the backup file from the first server and use the restore script to the settings:

```
[root@mfavpn2 ~]# /opt/mfavpn/bin/restore mfavpn_backup.gz
Are you sure you want to restore MFA VPN Server (y/n)? y
Starting restore procedure...
Unpacking backup files... Ok
Silent mode requires one WebADM node's hostname or IP.

Usage: /opt/mfavpn/bin/setup <silent HOSTNAME|IP | reset | restore>
  Silent setup: setup without user any prompt.
  Reset: Reset MFA VPN Server to its original state.


MFA VPN Server backup restored from file 'mfavpn_backup.gz'.

[root@mfavpn2 ~]# /opt/mfavpn/bin/mfavpn start
Starting MFA VPN Server...
Starting daemon 'openvpn'... Ok
Starting daemon 'ovpnauthd'... Ok
```

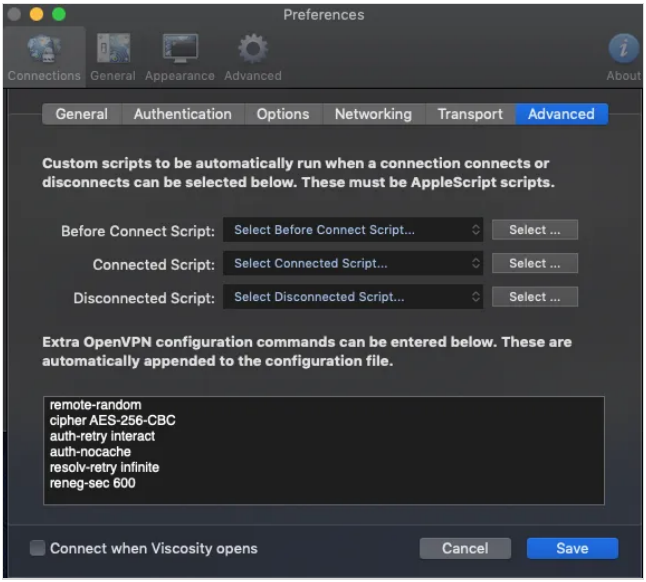
You need to configure your VPN clients to use both servers, which can be done either by modifying the .ovpn file in the end-user package or in case of Viscosity VPN client in the GUI:

```
# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote my-server-1 1194
remote my-server-2 1194

# Choose a random host from the remote
# list for load-balancing.  Otherwise
# try hosts in the order specified.
remote-random
```

First, add the addresses of both servers in the General settings:



Then configure the “remote-random” option in the Advanced settings:



8. Video Tutorial