1 2 9 0

UNIVERSIDADE D
COIMBRA

Information Technology Security

Certificates + OpenVPN

Francisco Catarino Mendes - 2019222823
Leonardo Oliveira Pereira - 2020239125
Department of Informatics Engineering
University of Coimbra

Introduction

This assignment was made to learn several concepts, mainly regarding the functioning of certificates and OpenVPN. Secure communications are supported by VPN tunnels established between remote clients (road warriors) and the Coimbra VPN gateway, and between the two VPN gateways (Coimbra and Lisboa). To enable VPN tunnels, OpenVPN will be used. Figure 1 shows the architecture of this work:

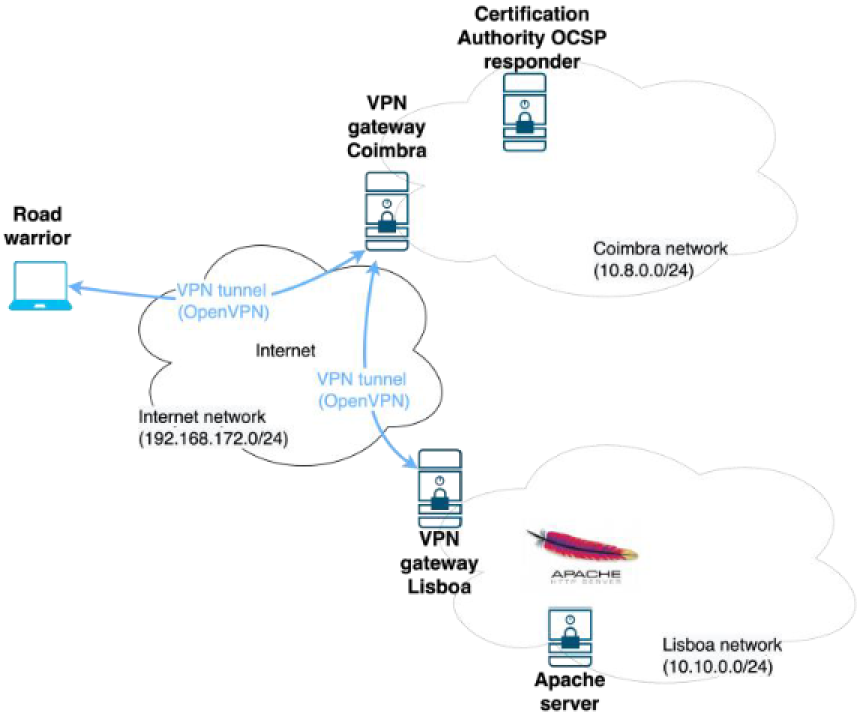

Certification Authority OCSP responder
VPN gateway Coimbra
Road warrior
Coimbra network (10.8.0.0/24)
VPN tunnel (OpenVPN)
Internet
VPN tunnel (OpenVPN)
Internet network (192.168.172.0/24)
VPN gateway Lisboa
APACHE
Lisboa network (10.10.0.0/24)
Apache server

Figure 1: Scenario of the Assignment

The main goals are to configure VPN tunnels in the “network-to-network” and “road warrior” scenarios, enable two-factor user authentication for OpenVPN, manage certification authorities, X.509 certificates and OCSP responders, and provide access to a web service only through the VPN tunnel.

1 - Configurations for the implementation of the Services

Here, it is explained how the CA and the X.509 certificates were created and configured. Moreover, the web server and its configuration are also described.

Before moving on to those topics, it must be said that three virtual machines were used, one corresponding to the road warrior, one to the Coimbra network, and one to the Lisbon network. The idea is the Lisbon network and the road warrior being clients of the server created in the Coimbra network, using different ports. Furthermore, the Coimbra network also allocates an OCSP responder, while the Lisbon network possesses an Apache server (the web server).

1.1 - Private Certification Authority (CA) and X.509 Certificates

The CA was created via Easy-RSA, and put into all the machines, via the following commands:

./easyrsa init-pki - to start the pki folder;

./easyrsa build-ca - to create the CA and all its respective components, like for example the key.

Then, it was time to create the needed X.509 certificates, all signed by this CA. In total, 7 certificates were created:

Lisbon - ./easyrsa build-client-full lisboa - for the connection with Coimbra as a client;

./easyrsa build-server-full www.stiroles.edu - for the Apache server.

Coimbra - ./easyrsa build-server-full coimbra - for the connections with the road warrior and Lisbon clients;

./easyrsa dh - diffie-hellmann key is also needed for the configuration of the Coimbra server.

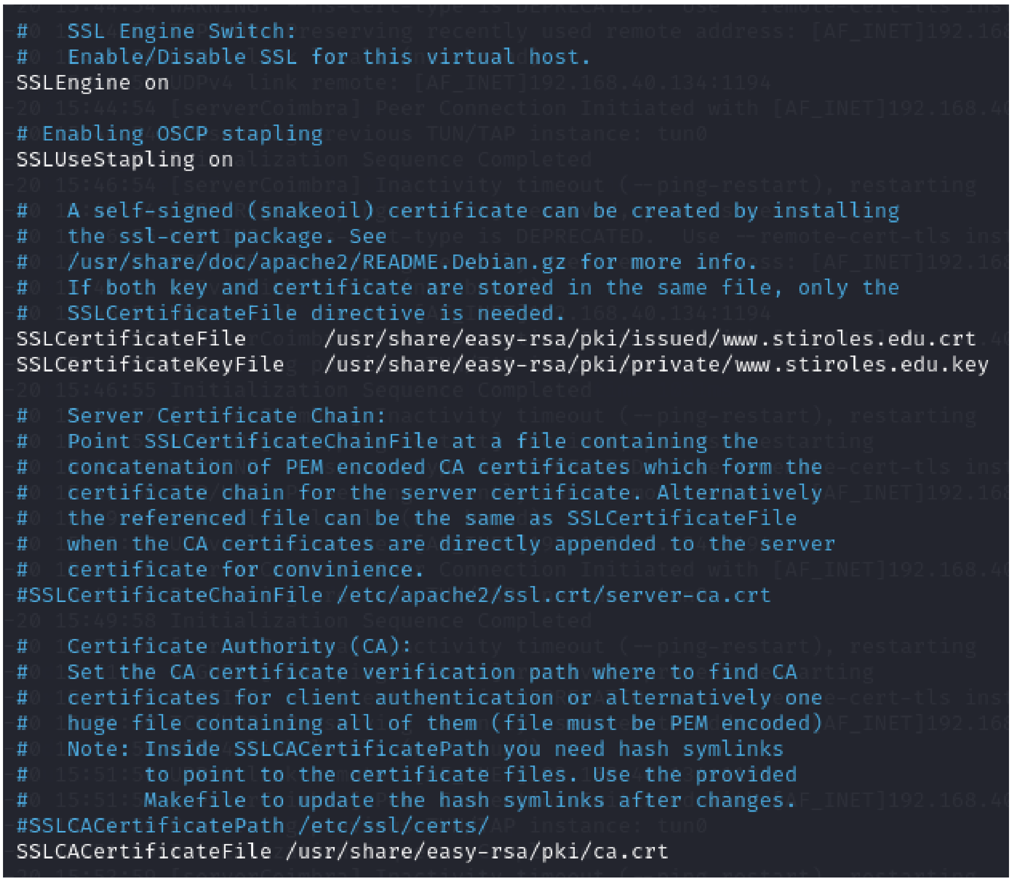
Road Warrior - ./easyrsa build-client-full rw - for the connection with Coimbra as a client;

./easyrsa build-client-full clientapache - certificate for the web client of Apache;

./easyrsa revoke rev - to make a revoked certificate for testing with OCSP.

1.2 - Web Server (Apache)

To configure the Apache server, first, one needs to access the file **/etc/apache2/sites-available/default-ssl.conf**, and put the right paths of the CA, X.509 certificate and key in **SSLCertificateFile**, **SSLCertificateKeyFile**, and **SSLCACertificateFile**, like Figure 2 illustrates:

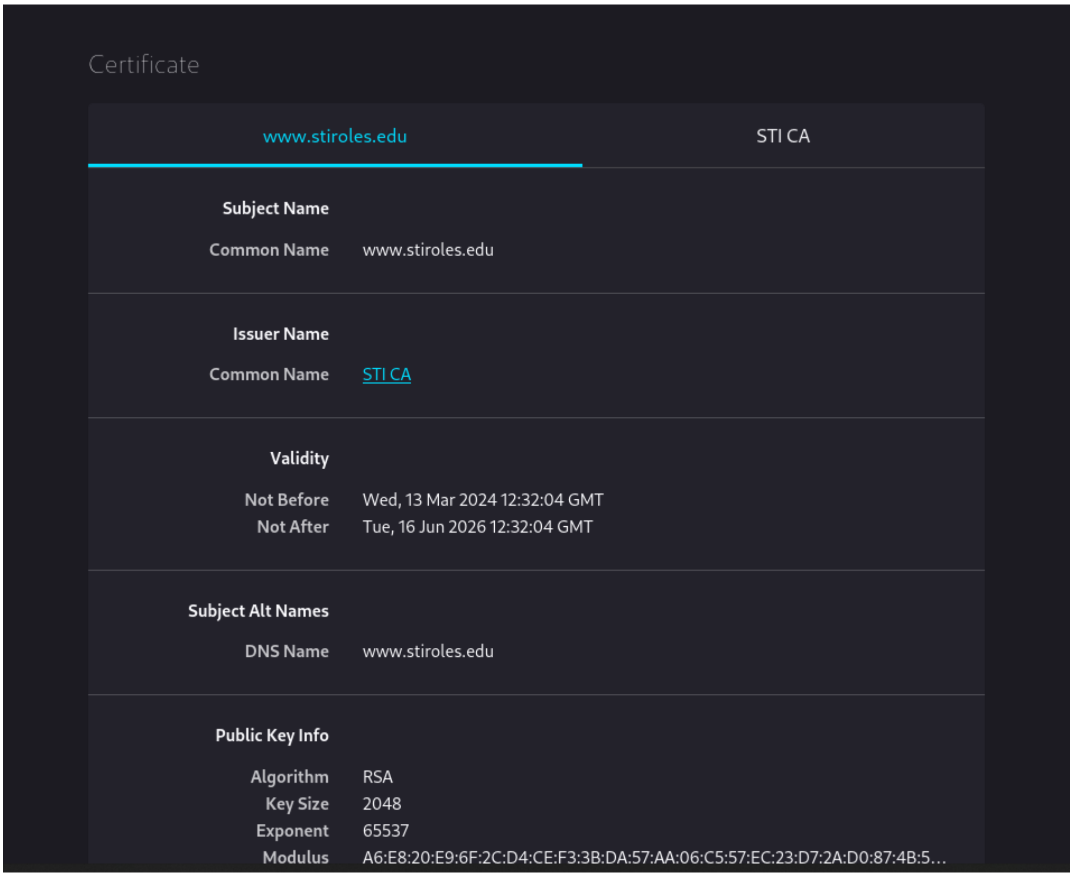


```
#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.
SSLEngine on

# Enabling OSCP stapling
SSLUseStapling on

#   A self-signed (snakeoil) certificate can be created by installing
#   the ssl-cert package. See
#   /usr/share/doc/apache2/README.Debian.gz for more info.
#   If both key and certificate are stored in the same file, only the
#   SSLCertificateFile directive is needed.
SSLCertificateFile      /usr/share/easy-rsa/pki/issued/www.stiroles.edu.crt
SSLCertificateKeyFile   /usr/share/easy-rsa/pki/private/www.stiroles.edu.key

#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
#   concatenation of PEM encoded CA certificates which form the
#   certificate chain for the server certificate. Alternatively
#   the referenced file can be the same as SSLCertificateFile
#   when the CA certificates are directly appended to the server
#   certificate for convinience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

#   Certificate Authority (CA):
#   Set the CA certificate verification path where to find CA
#   certificates for client authentication or alternatively one
#   huge file containing all of them (file must be PEM encoded)
#   Note: Inside SSLCACertificatePath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
#SSLCACertificatePath /etc/ssl/certs/
SSLCACertificateFile /usr/share/easy-rsa/pki/ca.crt
```

Figure 2: Configuration of the default-ssl.conf file

After this is done and applied, and Apache is restarted, **/etc/hosts** must be modified to include the right IP for the link used for the web server.

Then, the certificate has to be imported to the browser, in the PKCS12 format. The same goes for the web client, it has to import the web client certificate to the browser. Figure 3 shows the result of the import process, when one can see the certificate and the CA in the browser:



Certificate

Subject Name

Common Name www.stiroles.edu

Issuer Name

Common Name STI CA

Validity

Not Before Wed, 13 Mar 2024 12:32:04 GMT
Not After Tue, 16 Jun 2026 12:32:04 GMT

Subject Alt Names

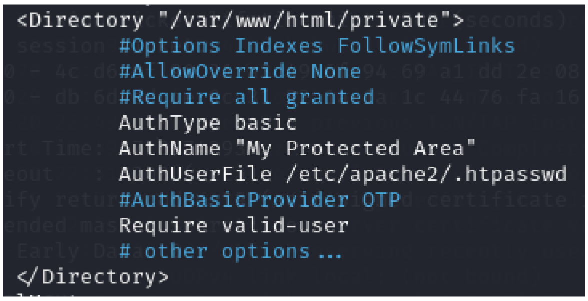
DNS Name www.stiroles.edu

Public Key Info

Algorithm RSA
Key Size 2048
Exponent 65537
Modulus A6:E8:20:E9:6F:2C:D4:CE:F3:3B:DA:57:AA:06:C5:57:EC:23:D7:2A:D0:87:4B:5…

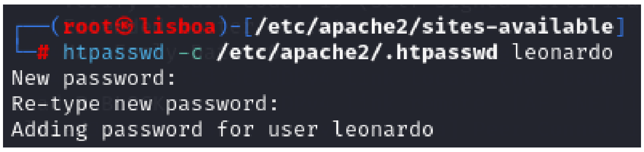
Figure 3: Certificates in the browser

Finally, there is also a resource (directory) that has to be accessible only through One Time User-Password. The configurations for this are shown in Figures 4, 5, 6, and 7:



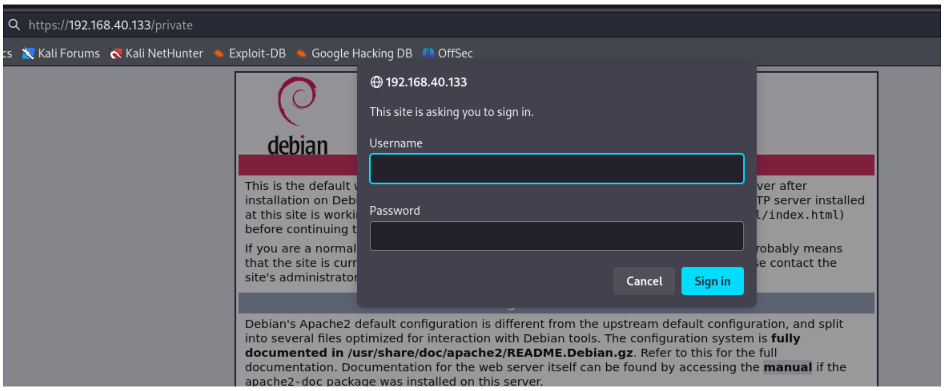
```
<Directory "/var/www/html/private">
        #Options Indexes FollowSymLinks
        #AllowOverride None
        #Require all granted
        AuthType basic
        AuthName "My Protected Area"
        AuthUserFile /etc/apache2/.htpasswd
        #AuthBasicProvider OTP
        Require valid-user
        # other options ...
</Directory>
```

Figure 4: Directory with OTP Configuration (1)



```
┌──(root㉿lisboa)-[/etc/apache2/sites-available]
└─# htpasswd -c /etc/apache2/.htpasswd leonardo
New password:
Re-type new password:
Adding password for user leonardo
```

Figure 5: Directory with OTP Configuration (2)



192.168.40.133

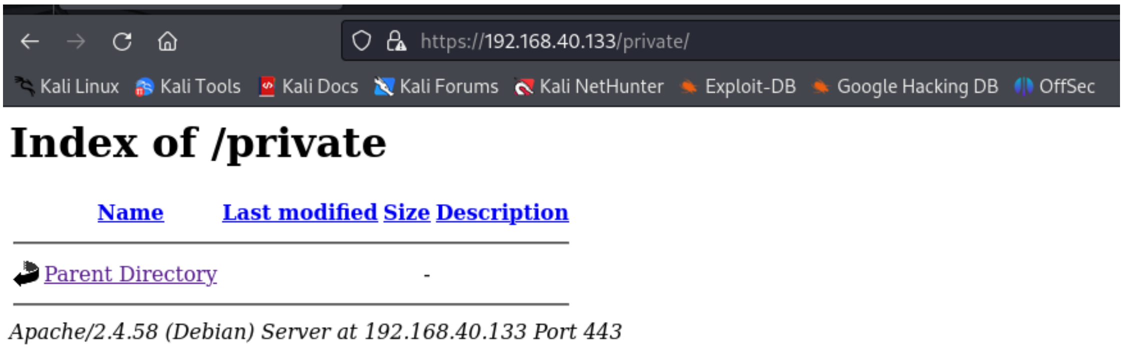
This site is asking you to sign in.

Username

Password

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

Figure 6: Directory with OTP Configuration (3)



Index of /private

Name	Last modified	Size	Description
Parent Directory		-	

Apache/2.4.58 (Debian) Server at 192.168.40.133 Port 443

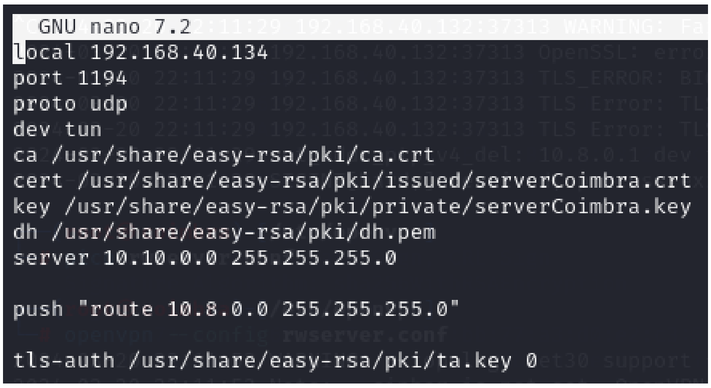
Figure 7: Directory with OTP Configuration (4)

2 - Configurations for the implementation of the VPN tunnels

In this section, a description of the configurations made to implement the VPN tunnels between the networks is correctly is given, as well as an explanation of the OCSP Responder and the implemented two-factor user authentication.

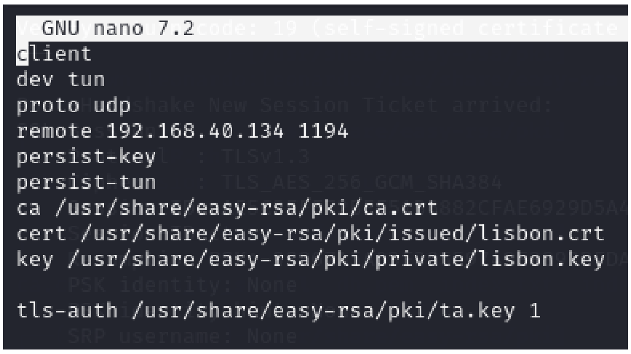
2.1 - VPN “network-to-network” tunnel between Coimbra and Lisboa

The first thing to do in this part is to reveal the .conf files that correspond to the configurations of both the server in Coimbra and the client in Lisbon. Figures 8 and 9 show the content of these files respectively:



```
  GNU nano 7.2
local 192.168.40.134
port 1194
proto udp
dev tun
ca /usr/share/easy-rsa/pki/ca.crt
cert /usr/share/easy-rsa/pki/issued/serverCoimbra.crt
key /usr/share/easy-rsa/pki/private/serverCoimbra.key
dh /usr/share/easy-rsa/pki/dh.pem
server 10.10.0.0 255.255.255.0

push "route 10.8.0.0 255.255.255.0"

tls-auth /usr/share/easy-rsa/pki/ta.key 0
```

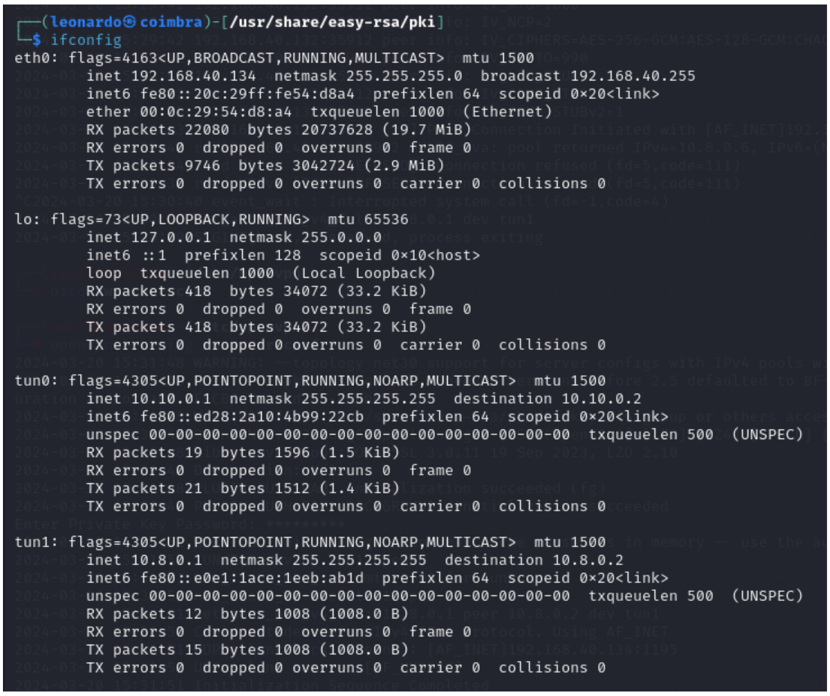
Figure 8: ServerCoimbraLisbon.conf



```
  GNU nano 7.2
client
dev tun
proto udp
remote 192.168.40.134 1194
persist-key
persist-tun
ca /usr/share/easy-rsa/pki/ca.crt
cert /usr/share/easy-rsa/pki/issued/lisbon.crt
key /usr/share/easy-rsa/pki/private/lisbon.key

tls-auth /usr/share/easy-rsa/pki/ta.key 1
```

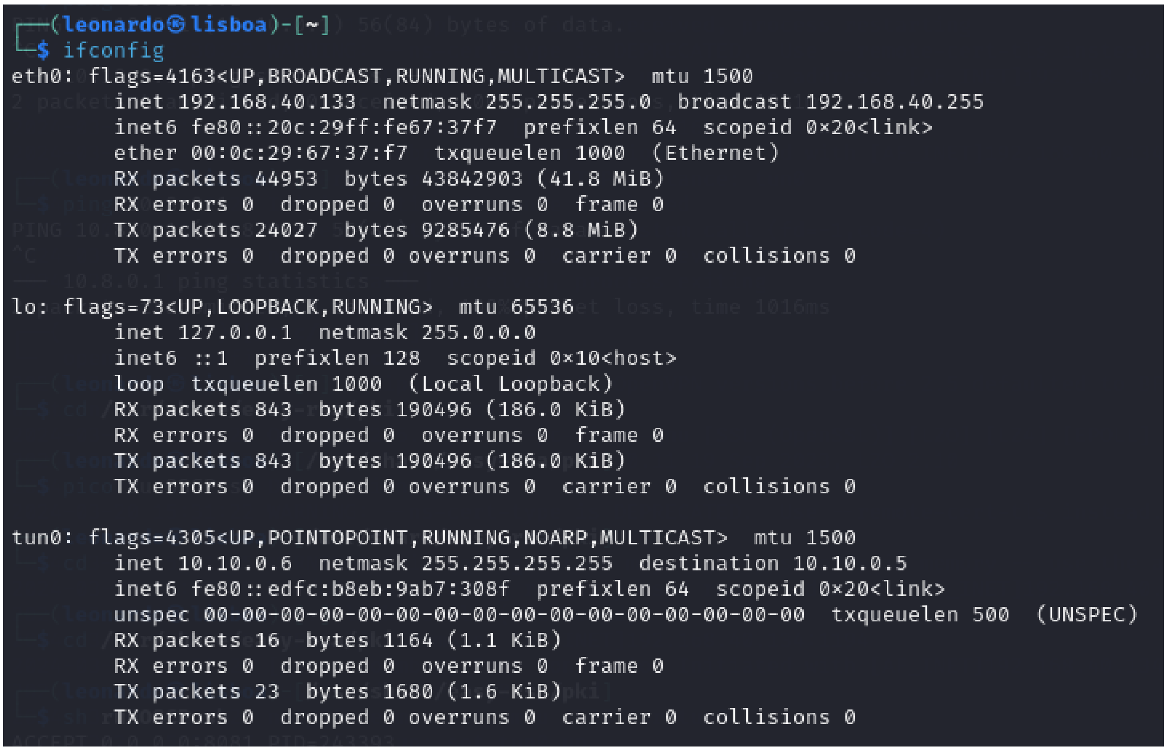
Figure 9: ClientLisbon.conf

To check that the tunnel is indeed working the way intended, Figures 10 and 11 show the interfaces with the tunnels (Coimbra, Lisbon), and Figures 12 and 13 show the pings made to each other (Coimbra, Lisbon):



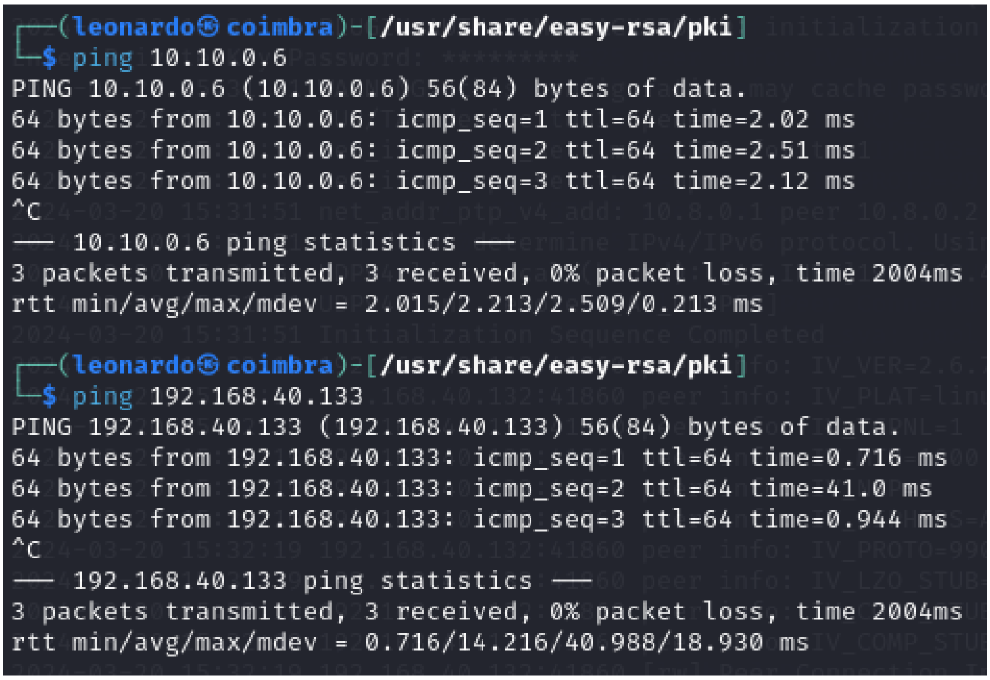
```
┌──(leonardo㉿coimbra)-[/usr/share/easy-rsa/pki]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.40.134  netmask 255.255.255.0  broadcast 192.168.40.255
        inet6 fe80::20c:29ff:fe54:d8a4  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:54:d8:a4  txqueuelen 1000  (Ethernet)
        RX packets 22080  bytes 20737628 (19.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9746  bytes 3042724 (2.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 418  bytes 34072 (33.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 418  bytes 34072 (33.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.10.0.1  netmask 255.255.255.255  destination 10.10.0.2
        inet6 fe80::ed28:2a10:4b99:22cb  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 19  bytes 1596 (1.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 21  bytes 1512 (1.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tun1: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.8.0.1  netmask 255.255.255.255  destination 10.8.0.2
        inet6 fe80::e0e1:1ace:1eeb:ab1d  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 12  bytes 1008 (1008.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 15  bytes 1008 (1008.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 10: Coimbra interfaces



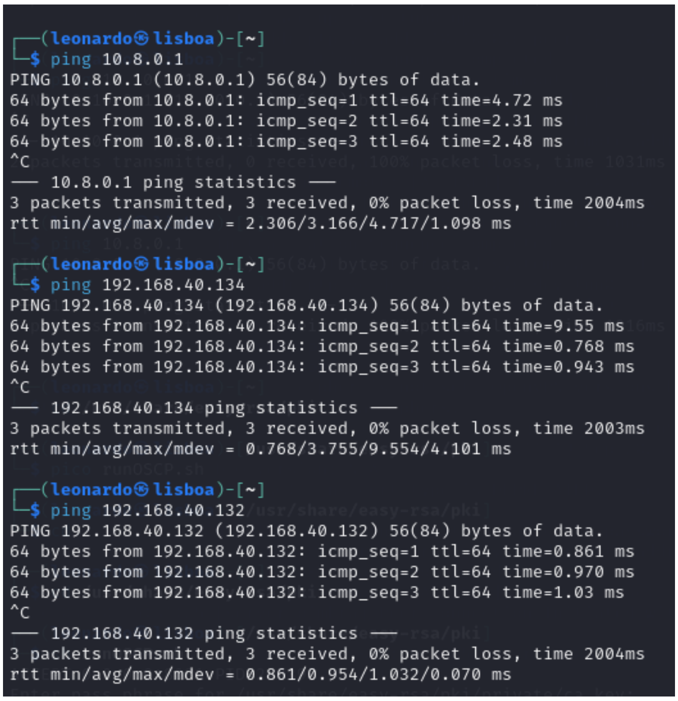
```
┌──(leonardo㉿lisboa)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.40.133  netmask 255.255.255.0  broadcast 192.168.40.255
        inet6 fe80::20c:29ff:fe67:37f7  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:67:37:f7  txqueuelen 1000  (Ethernet)
        RX packets 44953  bytes 43842903 (41.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24027  bytes 9285476 (8.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 843  bytes 190496 (186.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 843  bytes 190496 (186.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.10.0.6  netmask 255.255.255.255  destination 10.10.0.5
        inet6 fe80::edfc:b8eb:9ab7:308f  prefixlen 64  scopeid 0×20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 16  bytes 1164 (1.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 23  bytes 1680 (1.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 11: Lisbon interfaces



```
┌──(leonardo㉿coimbra)-[/usr/share/easy-rsa/pki]
└─$ ping 10.10.0.6
PING 10.10.0.6 (10.10.0.6) 56(84) bytes of data.
64 bytes from 10.10.0.6: icmp_seq=1 ttl=64 time=2.02 ms
64 bytes from 10.10.0.6: icmp_seq=2 ttl=64 time=2.51 ms
64 bytes from 10.10.0.6: icmp_seq=3 ttl=64 time=2.12 ms
^C
── 10.10.0.6 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 2.015/2.213/2.509/0.213 ms

┌──(leonardo㉿coimbra)-[/usr/share/easy-rsa/pki]
└─$ ping 192.168.40.133
PING 192.168.40.133 (192.168.40.133) 56(84) bytes of data.
64 bytes from 192.168.40.133: icmp_seq=1 ttl=64 time=0.716 ms
64 bytes from 192.168.40.133: icmp_seq=2 ttl=64 time=41.0 ms
64 bytes from 192.168.40.133: icmp_seq=3 ttl=64 time=0.944 ms
^C
── 192.168.40.133 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.716/14.216/40.988/18.930 ms
```

Figure 12: Coimbra-Lisbon pings

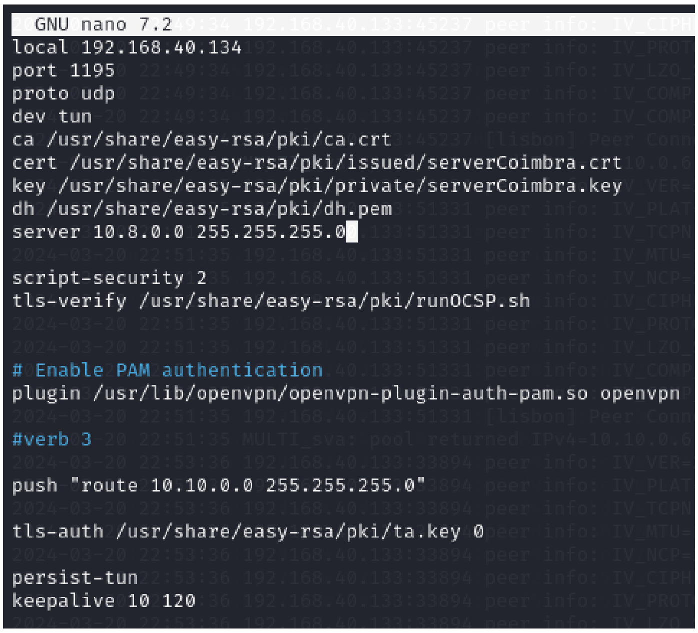


```
┌──(leonardo㉿lisboa)-[~]
└─$ ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=4.72 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=2.31 ms
64 bytes from 10.8.0.1: icmp_seq=3 ttl=64 time=2.48 ms
^C
--- 10.8.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 2.306/3.166/4.717/1.098 ms

┌──(leonardo㉿lisboa)-[~]
└─$ ping 192.168.40.134
PING 192.168.40.134 (192.168.40.134) 56(84) bytes of data.
64 bytes from 192.168.40.134: icmp_seq=1 ttl=64 time=9.55 ms
64 bytes from 192.168.40.134: icmp_seq=2 ttl=64 time=0.768 ms
64 bytes from 192.168.40.134: icmp_seq=3 ttl=64 time=0.943 ms
^C
--- 192.168.40.134 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.768/3.755/9.554/4.101 ms

┌──(leonardo㉿lisboa)-[~]
└─$ ping 192.168.40.132
PING 192.168.40.132 (192.168.40.132) 56(84) bytes of data.
64 bytes from 192.168.40.132: icmp_seq=1 ttl=64 time=0.861 ms
64 bytes from 192.168.40.132: icmp_seq=2 ttl=64 time=0.970 ms
64 bytes from 192.168.40.132: icmp_seq=3 ttl=64 time=1.03 ms
^C
--- 192.168.40.132 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.861/0.954/1.032/0.070 ms
```

Figure 13: Lisbon-Coimbra pings

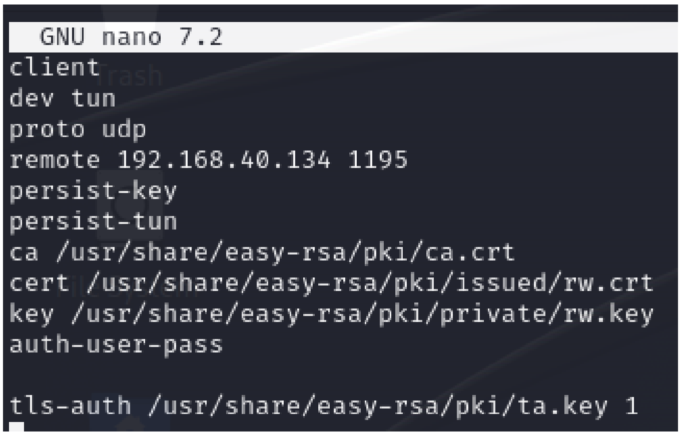
2.2 - Remote VPN client (road warrior)

Now, for the Road Warrior, it also needs to establish a tunnel with the server in Coimbra. Therefore, Figures 14 and 15 show the server configuration of Coimbra, made with a different port and the One Time User-Password plugin, and the client configuration of the Road Warrior, respectively:



```
 GNU nano 7.2
local 192.168.40.134
port 1195
proto udp
dev tun
ca /usr/share/easy-rsa/pki/ca.crt
cert /usr/share/easy-rsa/pki/issued/serverCoimbra.crt
key /usr/share/easy-rsa/pki/private/serverCoimbra.key
dh /usr/share/easy-rsa/pki/dh.pem
server 10.8.0.0 255.255.255.0

script-security 2
tls-verify /usr/share/easy-rsa/pki/runOCSP.sh

# Enable PAM authentication
plugin /usr/lib/openvpn/openvpn-plugin-auth-pam.so openvpn

#verb 3

push "route 10.10.0.0 255.255.255.0"

tls-auth /usr/share/easy-rsa/pki/ta.key 0

persist-tun
keepalive 10 120
```

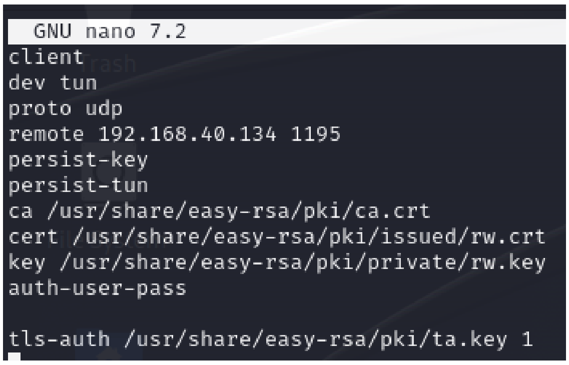
Figure 12: ServerCoimbraRW.conf



```
 GNU nano 7.2
client
dev tun
proto udp
remote 192.168.40.134 1195
persist-key
persist-tun
ca /usr/share/easy-rsa/pki/ca.crt
cert /usr/share/easy-rsa/pki/issued/rw.crt
key /usr/share/easy-rsa/pki/private/rw.key
auth-user-pass

tls-auth /usr/share/easy-rsa/pki/ta.key 1
```

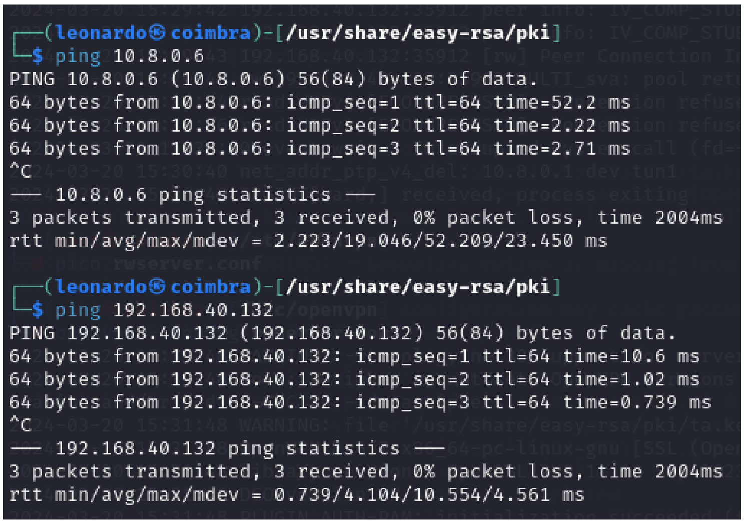
Figure 13: ClientRW.conf

Once again, to check whether the tunnel is working between the two networks, Figures 10 and 14 show the interfaces of Coimbra and the Road Warrior, while Figures 15 and 16 show the pings from Coimbra to the Road Warrior, and the other way around:



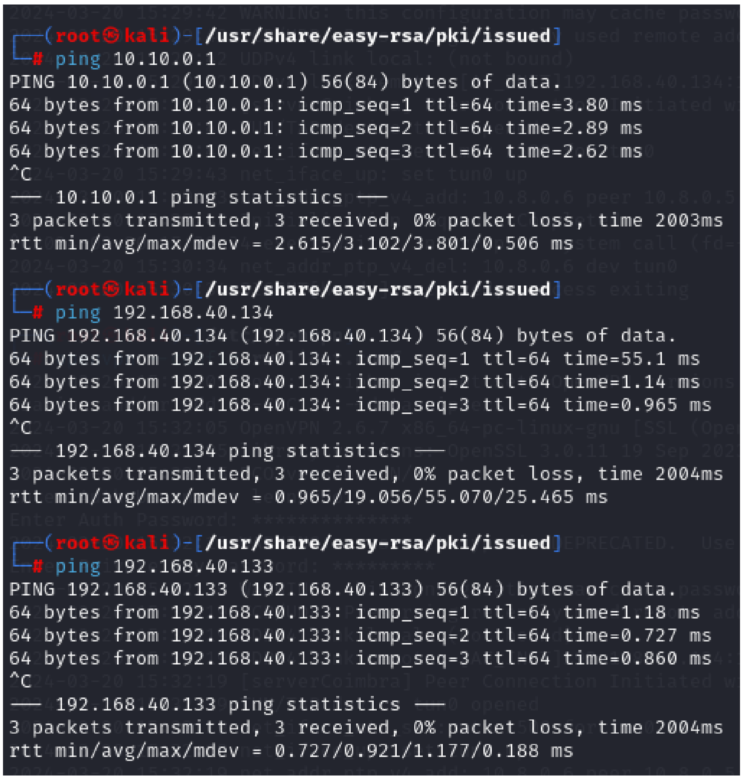
```
 GNU nano 7.2
client
dev tun
proto udp
remote 192.168.40.134 1195
persist-key
persist-tun
ca /usr/share/easy-rsa/pki/ca.crt
cert /usr/share/easy-rsa/pki/issued/rw.crt
key /usr/share/easy-rsa/pki/private/rw.key
auth-user-pass

tls-auth /usr/share/easy-rsa/pki/ta.key 1
```

Figure 14: Road Warrior interfaces



```
┌──(leonardo㉿coimbra)-[/usr/share/easy-rsa/pki]
└─$ ping 10.8.0.6
PING 10.8.0.6 (10.8.0.6) 56(84) bytes of data.
64 bytes from 10.8.0.6: icmp_seq=1 ttl=64 time=52.2 ms
64 bytes from 10.8.0.6: icmp_seq=2 ttl=64 time=2.22 ms
64 bytes from 10.8.0.6: icmp_seq=3 ttl=64 time=2.71 ms
^C
── 10.8.0.6 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 2.223/19.046/52.209/23.450 ms

┌──(leonardo㉿coimbra)-[/usr/share/easy-rsa/pki]
└─$ ping 192.168.40.132
PING 192.168.40.132 (192.168.40.132) 56(84) bytes of data.
64 bytes from 192.168.40.132: icmp_seq=1 ttl=64 time=10.6 ms
64 bytes from 192.168.40.132: icmp_seq=2 ttl=64 time=1.02 ms
64 bytes from 192.168.40.132: icmp_seq=3 ttl=64 time=0.739 ms
^C
── 192.168.40.132 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.739/4.104/10.554/4.561 ms
```

Figure 15: Coimbra-RW pings

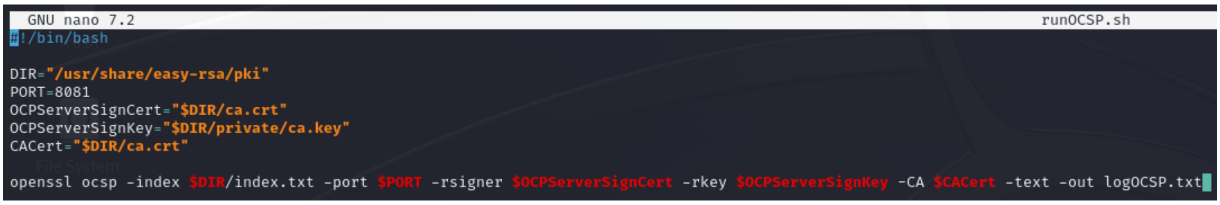


```
┌──(root㉿kali)-[/usr/share/easy-rsa/pki/issued]
└─# ping 10.10.0.1
PING 10.10.0.1 (10.10.0.1) 56(84) bytes of data.
64 bytes from 10.10.0.1: icmp_seq=1 ttl=64 time=3.80 ms
64 bytes from 10.10.0.1: icmp_seq=2 ttl=64 time=2.89 ms
64 bytes from 10.10.0.1: icmp_seq=3 ttl=64 time=2.62 ms
^C
─── 10.10.0.1 ping statistics ───
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 2.615/3.102/3.801/0.506 ms

┌──(root㉿kali)-[/usr/share/easy-rsa/pki/issued]
└─# ping 192.168.40.134
PING 192.168.40.134 (192.168.40.134) 56(84) bytes of data.
64 bytes from 192.168.40.134: icmp_seq=1 ttl=64 time=55.1 ms
64 bytes from 192.168.40.134: icmp_seq=2 ttl=64 time=1.14 ms
64 bytes from 192.168.40.134: icmp_seq=3 ttl=64 time=0.965 ms
^C
─── 192.168.40.134 ping statistics ───
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.965/19.056/55.070/25.465 ms

┌──(root㉿kali)-[/usr/share/easy-rsa/pki/issued]
└─# ping 192.168.40.133
PING 192.168.40.133 (192.168.40.133) 56(84) bytes of data.
64 bytes from 192.168.40.133: icmp_seq=1 ttl=64 time=1.18 ms
64 bytes from 192.168.40.133: icmp_seq=2 ttl=64 time=0.727 ms
64 bytes from 192.168.40.133: icmp_seq=3 ttl=64 time=0.860 ms
^C
─── 192.168.40.133 ping statistics ───
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.727/0.921/1.177/0.188 ms
```

Figure 16: RW-Coimbra pings

2.3 - OCSP Responder

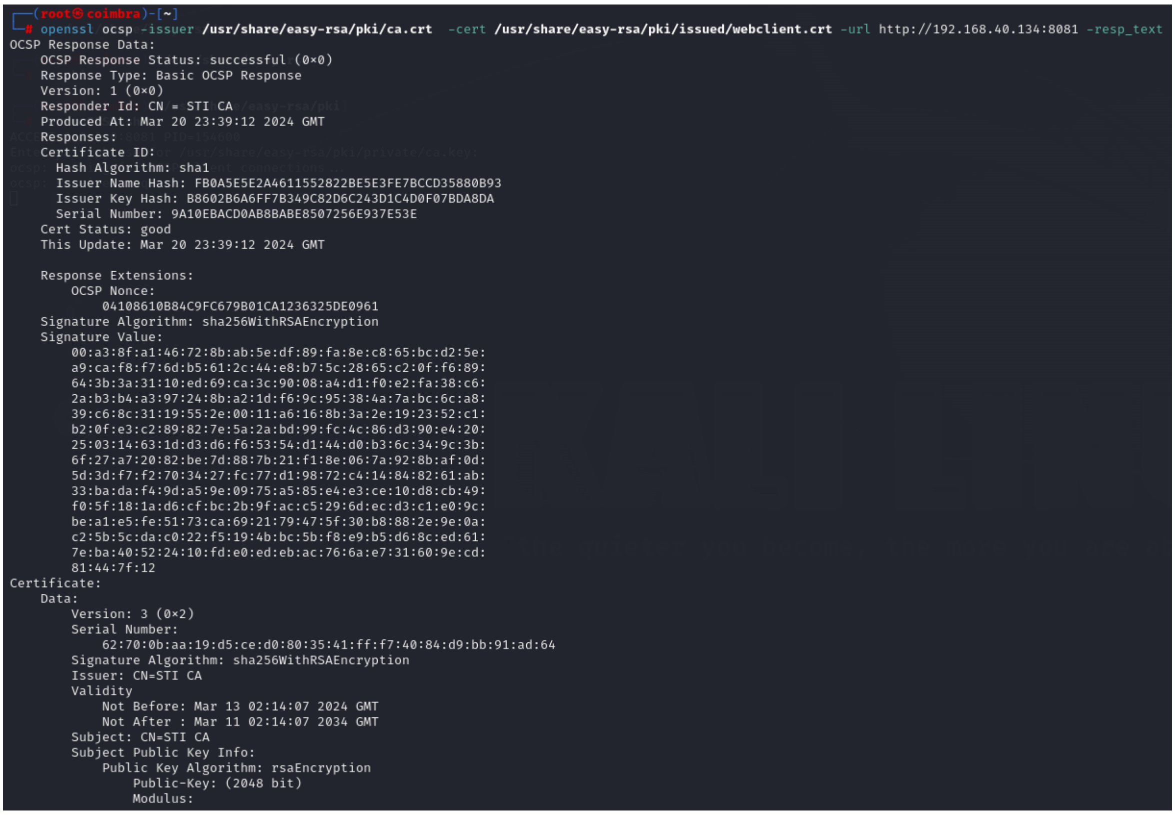
Concerning the OCSP service, installed in the Coimbra network via a different port, its configuration file appears in Figure 17:



```
  GNU nano 7.2                                                                                   runOCSP.sh
#!/bin/bash

DIR="/usr/share/easy-rsa/pki"
PORT=8081
OCPServerSignCert="$DIR/ca.crt"
OCPServerSignKey="$DIR/private/ca.key"
CACert="$DIR/ca.crt"

openssl ocsp -index $DIR/index.txt -port $PORT -rsigner $OCPServerSignCert -rkey $OCPServerSignKey -CA $CACert -text -out logOCSP.txt
```

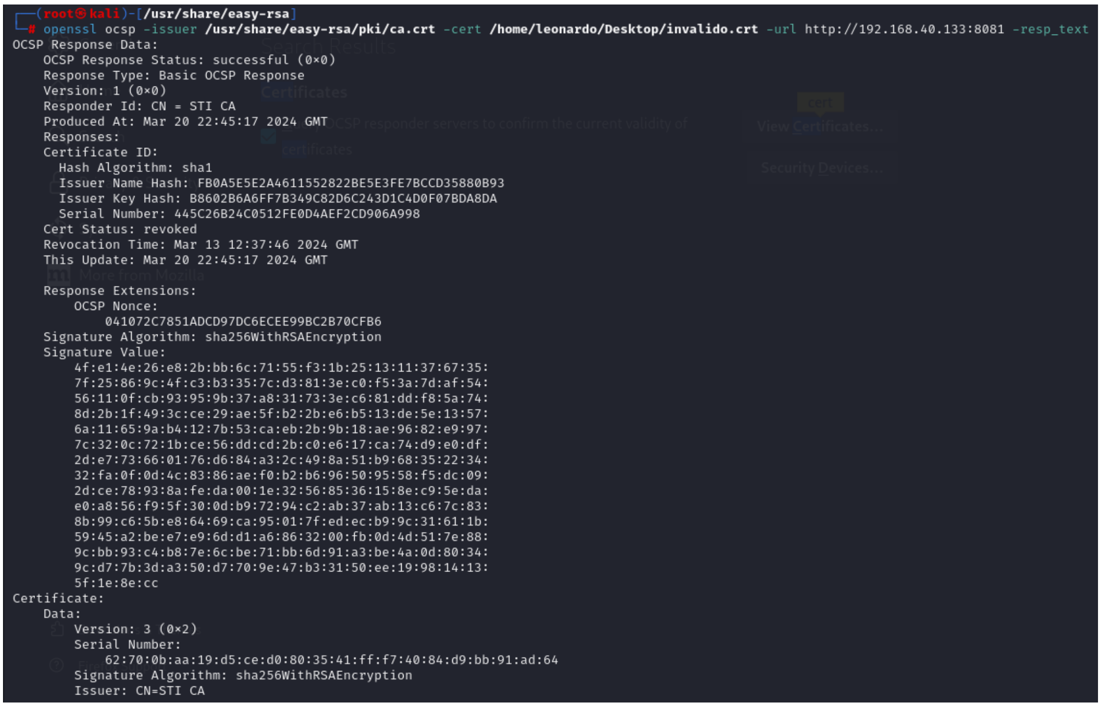
Figure 17: runOCSP.sh file

To test it, from the Road Warrior, a validation request of a certificate is made. Then, the OCSP service will make its acknowledgment, and return Figure 18 if the certificate is in a good state, or Figure 19 if the certificate was revoked:



```
┌──(root㉿coimbra)-[~]
└─# openssl ocsp -issuer /usr/share/easy-rsa/pki/ca.crt  -cert /usr/share/easy-rsa/pki/issued/webclient.crt -url http://192.168.40.134:8081 -resp_text
OCSP Response Data:
    OCSP Response Status: successful (0×0)
    Response Type: Basic OCSP Response
    Version: 1 (0×0)
    Responder Id: CN = STI CA
    Produced At: Mar 20 23:39:12 2024 GMT
    Responses:
    Certificate ID:
      Hash Algorithm: sha1
      Issuer Name Hash: FB0A5E5E2A4611552822BE5E3FE7BCCD35880B93
      Issuer Key Hash: B8602B6A6FF7B349C82D6C243D1C4D0F07BDA8DA
      Serial Number: 9A10EBACD0AB8BABE8507256E937E53E
    Cert Status: good
    This Update: Mar 20 23:39:12 2024 GMT

    Response Extensions:
        OCSP Nonce:
            04108610B84C9FC679B01CA1236325DE0961
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        00:a3:8f:a1:46:72:8b:ab:5e:df:89:fa:8e:c8:65:bc:d2:5e:
        a9:ca:f8:f7:6d:b5:61:2c:44:e8:b7:5c:28:65:c2:0f:f6:89:
        64:3b:3a:31:10:ed:69:ca:3c:90:08:a4:d1:f0:e2:fa:38:c6:
        2a:b3:b4:a3:97:24:8b:a2:1d:f6:9c:95:38:4a:7a:bc:6c:a8:
        39:c6:8c:31:19:55:2e:00:11:a6:16:8b:3a:2e:19:23:52:c1:
        b2:0f:e3:c2:89:82:7e:5a:2a:bd:99:fc:4c:86:d3:90:e4:20:
        25:03:14:63:1d:d3:d6:f6:53:54:d1:44:d0:b3:6c:34:9c:3b:
        6f:27:a7:20:82:be:7d:88:7b:21:f1:8e:06:7a:92:8b:af:0d:
        5d:3d:f7:f2:70:34:27:fc:77:d1:98:72:c4:14:84:82:61:ab:
        33:ba:da:f4:9d:a5:9e:09:75:a5:85:e4:e3:ce:10:d8:cb:49:
        f0:5f:18:1a:d6:cf:bc:2b:9f:ac:c5:29:6d:ec:d3:c1:e0:9c:
        be:a1:e5:fe:51:73:ca:69:21:79:47:5f:30:b8:88:2e:9e:0a:
        c2:5b:5c:da:c0:22:f5:19:4b:bc:5b:f8:e9:b5:d6:8c:ed:61:
        7e:ba:40:52:24:10:fd:e0:ed:eb:ac:76:6a:e7:31:60:9e:cd:
        81:44:7f:12
Certificate:
    Data:
        Version: 3 (0×2)
        Serial Number:
            62:70:0b:aa:19:d5:ce:d0:80:35:41:ff:f7:40:84:d9:bb:91:ad:64
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=STI CA
        Validity
            Not Before: Mar 13 02:14:07 2024 GMT
            Not After : Mar 11 02:14:07 2034 GMT
        Subject: CN=STI CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
```

Figure 18: A good certificate



```
┌──(root㉿kali)-[/usr/share/easy-rsa]
└─# openssl ocsp -issuer /usr/share/easy-rsa/pki/ca.crt -cert /home/leonardo/Desktop/invalido.crt -url http://192.168.40.133:8081 -resp_text
OCSP Response Data:
    OCSP Response Status: successful (0×0)
    Response Type: Basic OCSP Response
    Version: 1 (0×0)
    Responder Id: CN = STI CA
    Produced At: Mar 20 22:45:17 2024 GMT
    Responses:
    Certificate ID:
      Hash Algorithm: sha1
      Issuer Name Hash: FB0A5E5E2A4611552822BE5E3FE7BCCD35880B93
      Issuer Key Hash: B8602B6A6FF7B349C82D6C243D1C4D0F07BDA8DA
      Serial Number: 445C26B24C0512FE0D4AEF2CD906A998
    Cert Status: revoked
    Revocation Time: Mar 13 12:37:46 2024 GMT
    This Update: Mar 20 22:45:17 2024 GMT

    Response Extensions:
        OCSP Nonce:
            041072C7851ADCD97DC6ECEE99BC2B70CFB6
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        4f:e1:4e:26:e8:2b:bb:6c:71:55:f3:1b:25:13:11:37:67:35:
        7f:25:86:9c:4f:c3:b3:35:7c:d3:81:3e:c0:f5:3a:7d:af:54:
        56:11:0f:cb:93:95:9b:37:a8:31:73:3e:c6:81:dd:f8:5a:74:
        8d:2b:1f:49:3c:ce:29:ae:5f:b2:2b:e6:b5:13:de:5e:13:57:
        6a:11:65:9a:b4:12:7b:53:ca:eb:2b:9b:18:ae:96:82:e9:97:
        7c:32:0c:72:1b:ce:56:dd:cd:2b:c0:e6:17:ca:74:d9:e0:df:
        2d:e7:73:66:01:76:d6:84:a3:2c:49:8a:51:b9:68:35:22:34:
        32:fa:0f:0d:4c:83:86:ae:f0:b2:b6:96:50:95:58:f5:dc:09:
        2d:ce:78:93:8a:fe:da:00:1e:32:56:85:36:15:8e:c9:5e:da:
        e0:a8:56:f9:5f:30:0d:b9:72:94:c2:ab:37:ab:13:c6:7c:83:
        8b:99:c6:5b:e8:64:69:ca:95:01:7f:ed:ec:b9:9c:31:61:1b:
        59:45:a2:be:e7:e9:6d:d1:a6:86:32:00:fb:0d:4d:51:7e:88:
        9c:bb:93:c4:b8:7e:6c:be:71:bb:6d:91:a3:be:4a:0d:80:34:
        9c:d7:7b:3d:a3:50:d7:70:9e:47:b3:31:50:ee:19:98:14:13:
        5f:1e:8e:cc
Certificate:
    Data:
        Version: 3 (0×2)
        Serial Number:
            62:70:0b:aa:19:d5:ce:d0:80:35:41:ff:f7:40:84:d9:bb:91:ad:64
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=STI CA
```

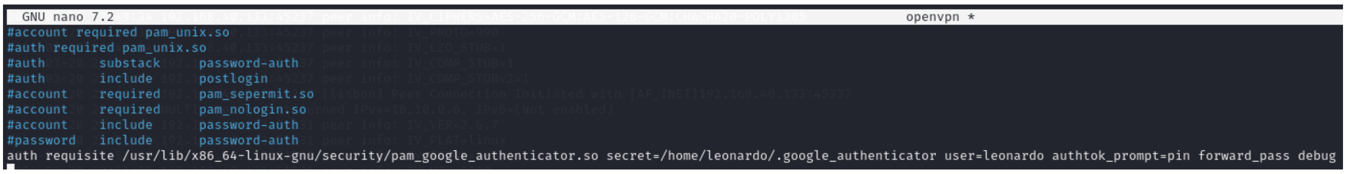
Figure 19: A revoked certificate

2.4 - Two-factor user authentication

Finally, the Road Warrior connecting to the Coimbra VPN gateway must be authenticated by a One-Time Password (authentication token) generated by an appropriate application.

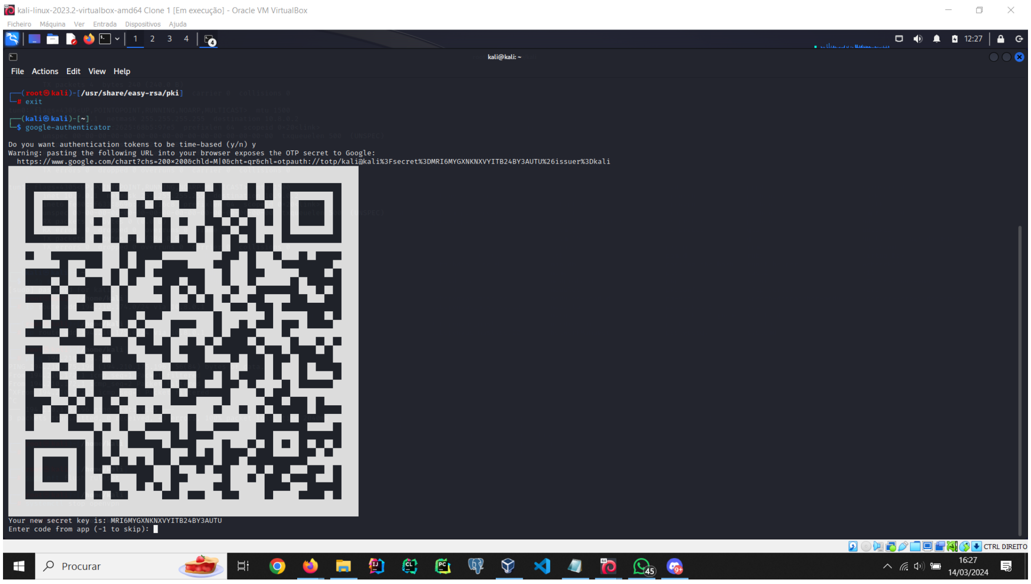
First and foremost, the library for the google-authenticator module was installed, via **apt-get install libpam-google-authenticator**. Then, in the configuration file that corresponds to the server of Coimbra that communicates with the Road Warrior, the plugin **/usr/lib/openvpn/openvpn-plugin-auth-pam.so openvpn** was incorporated, as Figure 12 shows.

Also, a new file must be created in **/etc/pam.d**, where the command Figure 20 shows has to exist. Then, a request for a code is made via the google authenticator mechanism, shown in Figure 21, and from there on, the Road Warrior must connect to the Coimbra server by authenticating with its username and password concatenated with the code received from google authenticator, a process illustrated by Figure 22, giving life to two-factor authentication.



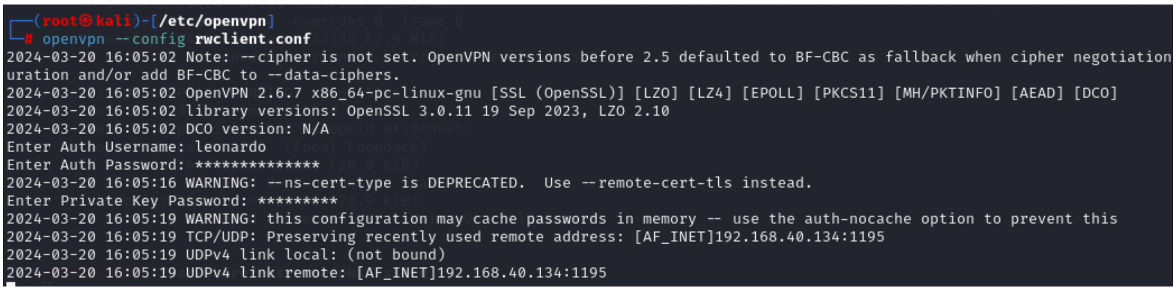
```
#account required pam_unix.so
#auth required pam_unix.so
#auth       substack     password-auth
#auth       include      postlogin
#account    required     pam_sepermit.so
#account    required     pam_nologin.so
#account    include      password-auth
#password   include      password-auth
auth requisite /usr/lib/x86_64-linux-gnu/security/pam_google_authenticator.so secret=/home/leonardo/.google_authenticator user=leonardo authtok_prompt=pin forward_pass debug
```

Figure 20: Openvpn Google Authenticator conf file



```
┌──(root㉿kali)-[/usr/share/easy-rsa/pki]
└─# exit

┌──(kali㉿kali)-[~]
└─$ google-authenticator

Do you want authentication tokens to be time-based (y/n) y
Warning: pasting the following URL into your browser exposes the OTP secret to Google:
  https://www.google.com/chart?chs=200x200&chld=M|0&cht=qr&chl=otpauth://totp/kali@kali%3Fsecret%3DMRI6MYGXNKNXVYITB24BY3AUTU%26issuer%3Dkali
Your new secret key is: MRI6MYGXNKNXVYITB24BY3AUTU
Enter code from app (-1 to skip): 
```

Figure 21: Request for a code to Google Authenticator

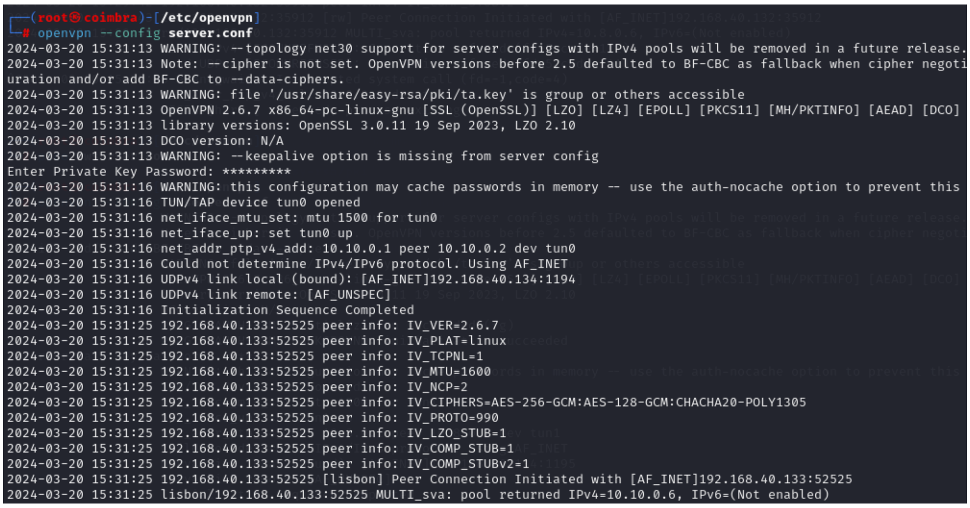


```
┌──(root㉿kali)-[/etc/openvpn]
└─# openvpn --config rwclient.conf
2024-03-20 16:05:02 Note: --cipher is not set. OpenVPN versions before 2.5 defaulted to BF-CBC as fallback when cipher negotiation
uration and/or add BF-CBC to --data-ciphers.
2024-03-20 16:05:02 OpenVPN 2.6.7 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] [DCO]
2024-03-20 16:05:02 library versions: OpenSSL 3.0.11 19 Sep 2023, LZO 2.10
2024-03-20 16:05:02 DCO version: N/A
Enter Auth Username: leonardo
Enter Auth Password: **************
2024-03-20 16:05:16 WARNING: --ns-cert-type is DEPRECATED.  Use --remote-cert-tls instead.
Enter Private Key Password: *********
2024-03-20 16:05:19 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
2024-03-20 16:05:19 TCP/UDP: Preserving recently used remote address: [AF_INET]192.168.40.134:1195
2024-03-20 16:05:19 UDPv4 link local: (not bound)
2024-03-20 16:05:19 UDPv4 link remote: [AF_INET]192.168.40.134:1195
```

Figure 22: Proof of the request for 2FA in ClientRW

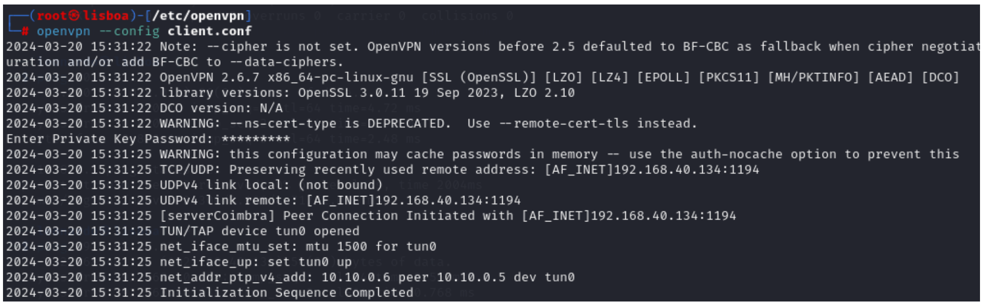
3 - Remaining Tests performed to validate the functionalities implemented

To end this report, here are the images that prove the servers and clients (with the exception of ServerRW, shown in Figure 22) run smoothly according to the configurations told in this report, shown by Figures 23, 24, 25 below:



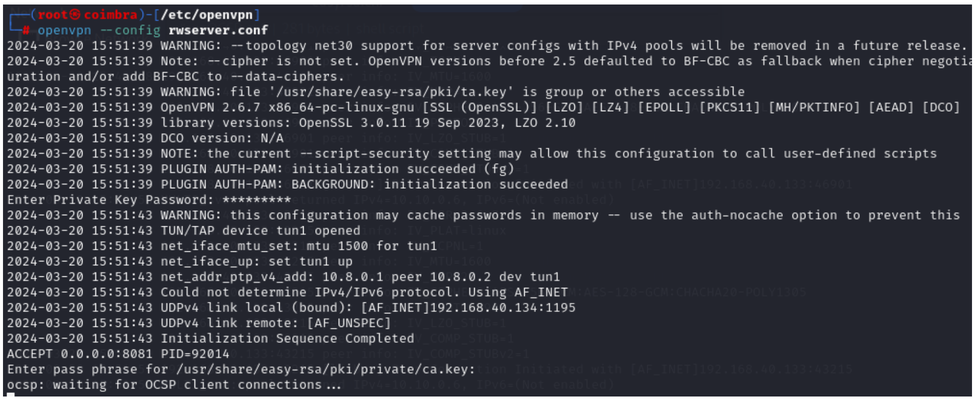
```
┌──(root㉿coimbra)-[/etc/openvpn]
└─# openvpn --config server.conf
2024-03-20 15:31:13 WARNING: --topology net30 support for server configs with IPv4 pools will be removed in a future release.
2024-03-20 15:31:13 Note: --cipher is not set. OpenVPN versions before 2.5 defaulted to BF-CBC as fallback when cipher negoti
uration and/or add BF-CBC to --data-ciphers.
2024-03-20 15:31:13 WARNING: file '/usr/share/easy-rsa/pki/ta.key' is group or others accessible
2024-03-20 15:31:13 OpenVPN 2.6.7 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] [DCO]
2024-03-20 15:31:13 library versions: OpenSSL 3.0.11 19 Sep 2023, LZO 2.10
2024-03-20 15:31:13 DCO version: N/A
2024-03-20 15:31:13 WARNING: --keepalive option is missing from server config
Enter Private Key Password: *********
2024-03-20 15:31:16 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
2024-03-20 15:31:16 TUN/TAP device tun0 opened
2024-03-20 15:31:16 net_iface_mtu_set: mtu 1500 for tun0
2024-03-20 15:31:16 net_iface_up: set tun0 up
2024-03-20 15:31:16 net_addr_ptp_v4_add: 10.10.0.1 peer 10.10.0.2 dev tun0
2024-03-20 15:31:16 Could not determine IPv4/IPv6 protocol. Using AF_INET
2024-03-20 15:31:16 UDPv4 link local (bound): [AF_INET]192.168.40.134:1194
2024-03-20 15:31:16 UDPv4 link remote: [AF_UNSPEC]
2024-03-20 15:31:16 Initialization Sequence Completed
2024-03-20 15:31:25 192.168.40.133:52525 peer info: IV_VER=2.6.7
2024-03-20 15:31:25 192.168.40.133:52525 peer info: IV_PLAT=linux
2024-03-20 15:31:25 192.168.40.133:52525 peer info: IV_TCPNL=1
2024-03-20 15:31:25 192.168.40.133:52525 peer info: IV_MTU=1600
2024-03-20 15:31:25 192.168.40.133:52525 peer info: IV_NCP=2
2024-03-20 15:31:25 192.168.40.133:52525 peer info: IV_CIPHERS=AES-256-GCM:AES-128-GCM:CHACHA20-POLY1305
2024-03-20 15:31:25 192.168.40.133:52525 peer info: IV_PROTO=990
2024-03-20 15:31:25 192.168.40.133:52525 peer info: IV_LZO_STUB=1
2024-03-20 15:31:25 192.168.40.133:52525 peer info: IV_COMP_STUB=1
2024-03-20 15:31:25 192.168.40.133:52525 peer info: IV_COMP_STUBv2=1
2024-03-20 15:31:25 192.168.40.133:52525 [lisbon] Peer Connection Initiated with [AF_INET]192.168.40.133:52525
2024-03-20 15:31:25 lisbon/192.168.40.133:52525 MULTI_sva: pool returned IPv4=10.10.0.6, IPv6=(Not enabled)
```

Figure 23: ServerCoimbraLisbon running



```
┌──(root㉿lisboa)-[/etc/openvpn]
└─# openvpn --config client.conf
2024-03-20 15:31:22 Note: --cipher is not set. OpenVPN versions before 2.5 defaulted to BF-CBC as fallback when cipher negotiat
uration and/or add BF-CBC to --data-ciphers.
2024-03-20 15:31:22 OpenVPN 2.6.7 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] [DCO]
2024-03-20 15:31:22 library versions: OpenSSL 3.0.11 19 Sep 2023, LZO 2.10
2024-03-20 15:31:22 DCO version: N/A
2024-03-20 15:31:22 WARNING: --ns-cert-type is DEPRECATED.  Use --remote-cert-tls instead.
Enter Private Key Password: *********
2024-03-20 15:31:25 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
2024-03-20 15:31:25 TCP/UDP: Preserving recently used remote address: [AF_INET]192.168.40.134:1194
2024-03-20 15:31:25 UDPv4 link local: (not bound)
2024-03-20 15:31:25 UDPv4 link remote: [AF_INET]192.168.40.134:1194
2024-03-20 15:31:25 [serverCoimbra] Peer Connection Initiated with [AF_INET]192.168.40.134:1194
2024-03-20 15:31:25 TUN/TAP device tun0 opened
2024-03-20 15:31:25 net_iface_mtu_set: mtu 1500 for tun0
2024-03-20 15:31:25 net_iface_up: set tun0 up
2024-03-20 15:31:25 net_addr_ptp_v4_add: 10.10.0.6 peer 10.10.0.5 dev tun0
2024-03-20 15:31:25 Initialization Sequence Completed
```

Figure 24: ClientLisbon running



```
┌──(root㉿coimbra)-[/etc/openvpn]
└─# openvpn --config rwserver.conf
2024-03-20 15:51:39 WARNING: --topology net30 support for server configs with IPv4 pools will be removed in a future release.
2024-03-20 15:51:39 Note: --cipher is not set. OpenVPN versions before 2.5 defaulted to BF-CBC as fallback when cipher negoti
uration and/or add BF-CBC to --data-ciphers.
2024-03-20 15:51:39 WARNING: file '/usr/share/easy-rsa/pki/ta.key' is group or others accessible
2024-03-20 15:51:39 OpenVPN 2.6.7 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] [DCO]
2024-03-20 15:51:39 library versions: OpenSSL 3.0.11 19 Sep 2023, LZO 2.10
2024-03-20 15:51:39 DCO version: N/A
2024-03-20 15:51:39 NOTE: the current --script-security setting may allow this configuration to call user-defined scripts
2024-03-20 15:51:39 PLUGIN AUTH-PAM: initialization succeeded (fg)
2024-03-20 15:51:39 PLUGIN AUTH-PAM: BACKGROUND: initialization succeeded
Enter Private Key Password: *********
2024-03-20 15:51:43 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
2024-03-20 15:51:43 TUN/TAP device tun1 opened
2024-03-20 15:51:43 net_iface_mtu_set: mtu 1500 for tun1
2024-03-20 15:51:43 net_iface_up: set tun1 up
2024-03-20 15:51:43 net_addr_ptp_v4_add: 10.8.0.1 peer 10.8.0.2 dev tun1
2024-03-20 15:51:43 Could not determine IPv4/IPv6 protocol. Using AF_INET
2024-03-20 15:51:43 UDPv4 link local (bound): [AF_INET]192.168.40.134:1195
2024-03-20 15:51:43 UDPv4 link remote: [AF_UNSPEC]
2024-03-20 15:51:43 Initialization Sequence Completed
ACCEPT 0.0.0.0:8081 PID=92014
Enter pass phrase for /usr/share/easy-rsa/pki/private/ca.key:
ocsp: waiting for OCSP client connections ...
```

Figure 25: ServerRW running

Conclusion

To conclude, in this assignment, a lot of knowledge was gathered about Certificates, OpenVPN, and Apache. Also, other concepts were worked on, like running an OCSP service and implementing Two Factor Authentication with One Time Passwords.

It also must be noted that the delivery of this report with the respective configuration files is signed with a PGP key and encrypted with the PGP key of the Professor.