UNIVERSIDADE D
COIMBRA

Information Tecnology Security

Authors:
Edgar Duarte no. 2019216077
Tatiana Almeida, no. 2019219581

March 12, 2023

Contents

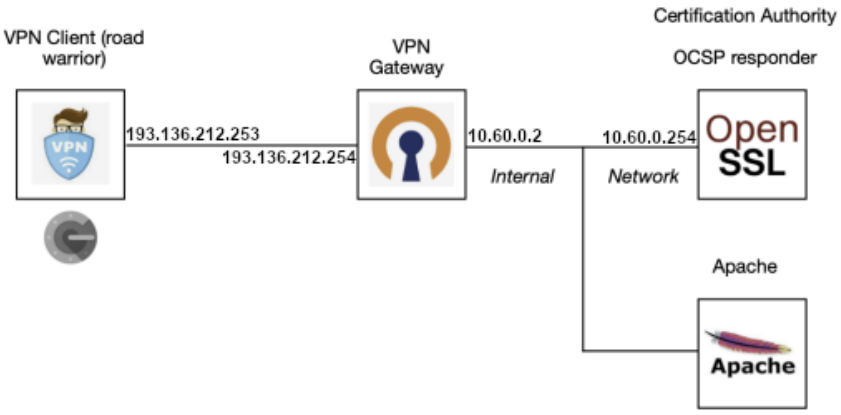
1 Introduction

For this project, a small networking scenario was built with the objective of obtaining knowledge about X509 certificates, VPN tunnels, OCSP, Apache and two factor authentication.

2 Implementation

2.1 Overall Setup

The network consists of one VPN, *VPN Gateway*, one client, *Road Warrior*, and a certificate authority, *OCSP responder and Apache*:



We used a total of three virtual machines for our project. The first virtual machine, which was associated with the Road Warrior, was part of the 193.136.212.0 network. The second virtual machine, which served as the VPN Gateway, belonged to both the external network (193.136.212.0) and the internal network (10.60.0.0). The third and final virtual machine was the Certification Authority, which was located solely on the internal network (10.60.0.0).

2.2 Certificates

All the certificates were created in the OCSP responder and were copied to the VPN and Road Warrior machines. Firstly, we created a self-signed certificate that represents the Certifying Authority. With this certificate we created the remaining certificates.

Certification Authority

```
openssl genrsa -des3 -out private/cakey.pem
openssl req -new -key private/cakey.pem -out ca.crt
    -----
    Country Name (2 letter code) [XX]:PT
    State or Province Name (full name) []:Coimbra
    Locality Name (eg, city) [Default City]:Coimbra
    Organization Name (eg, company) [Default Company Ltd]:UC
    Organizational Unit Name (eg, section) []:DEI
    Common Name (eg, your name or your server's hostname) []:CA
    Email Address []:ca@student.dei.uc.pt

openssl x509 -req -days 365 -in ca.crt -out cacert.pem -signkey
 private/cakey.pem
touch index.txt
echo 01 > serial
```

Apache

```
    openssl genrsa -des3 -out private/apache.key
    openssl req -new -key private/apache.key -out apache.csr
            -----
            Country Name (2 letter code) [XX]:PT
            State or Province Name (full name) []:Coimbra
            Locality Name (eg, city) [Default City]:Coimbra
            Organization Name (eg, company) [Default Company
Ltd]:UC
            Organizational Unit Name (eg, section) []:DEI
            Common Name (eg, your name or your server's
hostname) []: www.stiroles.edu
            Email Address []:apache@student.dei.uc.pt

        openssl ca -in apache.csr -cert cacert.pem -keyfile
private/cakey.pem -out certs/apache.crt
```

OCSP

```
    openssl genrsa -des3 -out private/ocsp.key
    openssl req -new -key private/apache.key -out ocsp.csr
            -----
            Country Name (2 letter code) [XX]:PT
            State or Province Name (full name) []:Coimbra
            Locality Name (eg, city) [Default City]:Coimbra
            Organization Name (eg, company) [Default Company
Ltd]:UC
            Organizational Unit Name (eg, section) []:DEI
            Common Name (eg, your name or your server's
hostname) []: ocsp.dei.uc.pt
            Email Address []:ocsp@student.dei.uc.pt
        openssl ca -in ocsp.csr -cert cacert.pem -keyfile
private/cakey.pem -out certs/ocsp.crt
```

OpenVPN Client

```
    openssl genrsa -des3 -out private/client.key
    openssl req -new -key private/client.key -out client.csr
        -----
        Country Name (2 letter code) [XX]:PT
        State or Province Name (full name) []:Coimbra
        Locality Name (eg, city) [Default City]:Coimbra
        Organization Name (eg, company) [Default Company Ltd]:
UC
        Organizational Unit Name (eg, section) []:DEI
        Common Name (eg, your name or your server's hostname)
[]: client
        Email Address []:client@student.dei.uc.pt

    openssl ca -in client.csr -cert cacert.pem -keyfile private
/cakey.pem -out certs/client.crt
```

Apache Client

```
    openssl genrsa -des3 -out private/client_apache.key
    openssl req -new -key private/client_apache.key -out
client_apache.csr
        -----
        Country Name (2 letter code) [XX]:PT
        State or Province Name (full name) []:Coimbra
        Locality Name (eg, city) [Default City]:Coimbra
        Organization Name (eg, company) [Default Company Ltd]:
UC
        Organizational Unit Name (eg, section) []:DEI
        Common Name (eg, your name or your server's hostname)
[]: client_apache
        Email Address []:client@student.dei.uc.pt

    openssl ca -in client_apache.csr -cert cacert.pem -keyfile
private/cakey.pem -out certs/client_apache.crt
```

VPN Gateway

```
openssl genrsa -des3 -out private/gw-vpn.key
openssl req -new -key private/gw-vpn.key -out gw-vpn.csr
openssl ca -in client.csr -cert cacert.pem -keyfile private/
cakey.pem -out certs/gw-vpn.crt
```

2.3 OpenVPN

The following commands were used to install OpenVPN:

```
yum install epel-release -y
yum install openvpn
```

In order to make the Road Warrior and the VPN gateway agree on a secret key, a file with using Diffie-Hellman's algorithm was generated:

```
openssl dhparam -out dh1024.pem 1024
```

To increase security when authenticating, a TA key was generated:

```
openvpn --genkey --secret ta.key
```

It is to note that the same TA file needs to be both on the client and server's machines.

Now, to have a connection between the machines, the client and server configuration files were altered to the following:

server.conf

```
local 193.136.212.254
port 1197
proto udp
dev tun
ca /etc/pki/CA/cacert.pem
cert /etc/pki/CA/certs/gw-vpn.crt
key /etc/pki/CA/private/gw-vpn.key
dh dh1024.pem
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "route 10.60.0.0 255.255.255.0"
keepalive 10 120
tls-auth /etc/pki/CA/ta.key 0
cipher AES-256-CBC
comp-lzo
```

client.conf

```
client
dev tun
proto udp
remote 193.136.212.254 1197
resolv-retry infinite
nobind
ca /etc/pki/CA/cacert.pem
cert /etc/pki/CA/certs/client.crt
key /etc/pki/CA/private/client.key
tls-auth /etc/pki/CA/ta.key 1
data-ciphers AES-256-CBC
comp-lzo
```

To open the connection, first the firewalls should be turned off and then the configuration files should be activated, as follows:

Server side

```
service firewalld stop
openvpn server.conf
```

Client side

```
service firewalld stop
openvpn client.conf
```

At this point, the machines should be connected via a VPN tunnel. The road warrior is now able to ping the server's 10.60.0.1 address, but it fails to ping any other 10.60.0.0/24 address (for example the Apache machine). This happens because the client's address inside the VPN is an address from 10.8.0.0/24, address not known by the 10.60.0.0/24 machines. There are multiple ways to resolve this, the one we choose was to apply SNAT to the traffic received from the VPN clients to the internal network, by using the following command in the VPN Gateway server:

```
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o ens36 -j SNAT
--to-source 10.60.0.2
```

2.4 Apache

In order to define the location of the Apache certificate and its private key and to allow client authentication, we have to change the following lines in the ssl.conf file:

```
SSLLCertificateFile /etc/pki/CA/certs/apache.crt
SSLCertificateKeyFile /etc/pki/CA/private/apache.key
SSLCACertificateFile /etc/pki/CA/cacert.pem

SSLVerifyClient require
SSLVerifyDepth 10
```

Then it is necessary to restart the httpd service:

```
service httpd restart
```

In order to securely access Apache, it is necessary to convert the client certificate to PKCS12 format and import it into the browser. The CA must also be imported as an authority to establish trust and enable the browser to recognize the secure connection.

```
openssl pkcs12 -export -out clientp1.p12 -inkey private/clientp1
 .key -in certs/clientp1.crt -certfile cacert.pem
```

Additionally, to ensure website access, it's essential to include the line "10.60.0.254 www.stiroles.edu" in the hosts file on the RW side.

```
nano /etc/hosts

(...)
10.60.0.254 www.stiroles.edu
```

2.5 OCSP

OCSP (Online Certificate Status Protocol) is a real-time protocol used to verify the revocation status of digital certificates. When a user accesses a website secured by an SSL/TLS certificate, their browser checks the certificate's revocation status using OCSP to confirm that it has not been revoked or compromised.

To configure the OCSP service in the Apache web server, we specified the address and port of the OCSP server in the Apache configuration file.

```
nano /etc/httpd/conf.d/ssl.conf
(...)
SSLOCSPEnable on
SSLOCSPDefaultResponder http://ocsp.dei.uc.pt:81
SSLOCSPOverrideResponder off
(...)
```

In order to include OCSP server information in the certificates issued by our CA, we needed to modify the way the certificates were generated. We accomplished this by adding relevant details to the OpenSSL configuration file located at /etc/pki/tls/openssl.cnf, which ensured that the necessary information was present in all certificates.

```
nano /etc/pki/tls/openssl.cnf
(...)
[usr_cert]
authorityInfoAccess = OCSP;URI:http://ocsp.dei.uc.pt:81
```

2.6 Two Factor Authentication

Two factor Authentication enables an extra layer of protection to authentication besides the usual username and password setup.

OpenVPN

Firstly, we installed the google-authenticator package (that allows to authenticate using the Google Authenticator), allow network forwarding, then set its permissions and finally allow update of users:

```
yum install google-authenticator

net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
sysctl -p

useradd gauth
mkdir /etc/openvpn/google-authenticator
chown gauth:gauth /etc/openvpn/google-authenticator && chmod
700 /etc/openvpn/google-authenticator

semanage fcontext -a -t openvpn_etc_rw_t -ff '/etc/openvpn/
google-authenticator(/.*)?'
```

Now we can create an user and generate it's QR Code :

```
useradd -s /sbin/nologin client
passwd client

su -c "google-authenticator -t -d -r3 -R30 -W -f -l 'VPN' -s /
etc/openvpn/google-authenticator/client" - gauth
```

It is to note that it is vital to scan the QR code that shows up in the terminal with the Google Authenticator application, so that the client can obtain the code needed when trying to open the VPN.

Now we have to create and configure a PAM file:

```
nano /etc/pam.d/openvpn

    #%PAM-1.0
    auth [user_unknown=ignore success=ok ignore=ignore default=
bad] pam_securetty.so
    auth required pam_google_authenticator.so secret=/etc/
openvpn/google-authenticator/${USER} user=gauth forward_pass
    auth include system-auth
    account include system-auth
    password include system-auth
```

Finally, we need to add the following lines in the client.conf and server.conf files:

client.conf

```
auth-user-pass
```

server.conf

```
plugin /usr/lib64/openvpn/plugins/openvpn-plugin-auth-pam.so
openvpn
```

With these steps completed, if the client now attempts to connect to the VPN, they will be asked a username, the password of the user (which also includes the code shown in the Google Authenticator app) and the password of the client certificate.

Apache

In order to use Google Authenticator with Apache, it was necessary to install a specific module for Apache that supports it.

```
yum install httpd-devel

wget https://storage.googleapis.com/google-code-archive-
downloads/v2/code.google.com/google-authenticator-apache-module
/GoogleAuthApache.src.r10.bz2

tar xvf GoogleAuthApache.src.r10.bz2
cd google-authenticator-apache-module
make install
```

Once the module is installed, it's necessary to modify the Apache configuration file by adding Google Authenticator to it.

```
nano /etc/httpd/conf/httpd.conf

Include conf.modules.d/*.conf
Loadmodule authn_google_module modules/mod_authn_google.so

<Directory />
    Options FollowSymLinks ExecCGI
    AllowOverride All
    Order deny,allow
    Allow from all
    AuthType Basic
    AuthName "Name"
    AuthBasicProvider "google_authenticator"
    Require valid-user
    GoogleAuthUserPath ga_auth
    GoogleAuthCookieLife 3600
    GoogleAuthEntryWindow 2
</Directory>

 <Directory "/var/www">
 (...)
 #Comment all
 </Directory>

<Directory "/var/www/html">
(...)
#Comment all
</Directory>
```

After that, We need to duplicate the user account previously created on the VPN gateway VM to the Apache VM. We also need to create a directory for google-authenticator users and give permissions:

```
[CertificateAuthority] mkdir /etc/httpd/ga_auth
```

```
[Vpn gateway] scp -r  scp -r /etc/openvpn/google-authenticator
192.168.57.135:/etc/httpd/ga_auth
```

```
[CertificateAuthority] chown apache:apache /etc/httpd/ga_auth/
sti
```

3 Tests

To validate the configurations, several tests were executed. The following table shows what tests were conducted and their outcome:

Test Description	Source	Destination	Expected Result	Success
Road Warrior pings Server's external network IP address	RW (193.136.212.253)	VPN Gateway (193.136.212.254)	Ping Successful	✓
Road Warrior pings Server's internal network IP address without a VPN tunnel	RW (193.136.212.253)	VPN Gateway (10.60.0.2)	Ping Failed	✓
Road Warrior pings Server's internal network IP address with a VPN tunnel	RW (193.136.212.253)	VPN Gateway (10.60.0.2)	Ping Successful	✓
Road Warrior pings Certification Authority's IP address without a VPN tunnel	RW (193.136.212.253)	Certification Authority (10.60.0.254)	Ping Failed	✓
Road Warrior pings Certification Authority's IP address with a VPN tunnel	RW (193.136.212.253)	Certification Authority (10.60.0.254)	Ping Successful	✓
Road Warrior tries to connect to Apache's website with valid certificate	RW (193.136.212.253)	www.stiroles.edu	Successful access	✓
Road Warrior tries to connect to Apache's website with revoked certificate	RW (193.136.212.253)	www.stiroles.edu	Blocked access	✓
Client opens OpenVPN using correct Two-Factor Authentication credentials	RW (193.136.212.253)	VPN Gateway (193.136.212.254)	Successful connection	✓
Client opens OpenVPN using wrong Two-Factor Authentication credentials	RW (193.136.212.253)	VPN Gateway (193.136.212.254)	Blocked access	✓
Client connects to Apache's website using correct Two-Factor Authentication credentials	RW (193.136.212.253)	www.stiroles.edu	Successful access	✓
Client connects to Apache's website using correct Two-Factor Authentication credentials	RW (193.136.212.253)	www.stiroles.edu	Blocked access	✓

3.1 Routing

Without OpenVPN

```
[root@localhost openvpn]# ping 10.60.0.2
PING 10.60.0.2 (10.60.0.2) 56(84) bytes of data.
^C
--- 10.60.0.2 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2006ms
```

Figure 1: Road Warrior(193.136.212.254) -> VPN Gateway(10.60.0.2)

```
[root@localhost tatianaalmeida]# ping 10.60.0.254
PING 10.60.0.254 (10.60.0.254) 56(84) bytes of data.
^C
--- 10.60.0.254 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1000ms
```

Figure 2: Road Warrior(193.136.212.254) -> Certification Authority(10.60.0.254)

With OpenVPN

```
[root@localhost tatianaalmeida]# ping 10.60.0.2
PING 10.60.0.2 (10.60.0.2) 56(84) bytes of data.
64 bytes from 10.60.0.2: icmp_seq=1 ttl=64 time=0.959 ms
64 bytes from 10.60.0.2: icmp_seq=2 ttl=64 time=1.12 ms
64 bytes from 10.60.0.2: icmp_seq=3 ttl=64 time=6.66 ms
64 bytes from 10.60.0.2: icmp_seq=4 ttl=64 time=1.11 ms
^C
--- 10.60.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.959/2.467/6.666/2.425 ms
```

Figure 3: Road Warrior(193.136.212.254) -> VPN Gateway(10.60.0.2)

```
[root@localhost tatianaalmeida]# ping 10.60.0.254
PING 10.60.0.254 (10.60.0.254) 56(84) bytes of data.
64 bytes from 10.60.0.254: icmp_seq=1 ttl=63 time=10.1 ms
64 bytes from 10.60.0.254: icmp_seq=2 ttl=63 time=6.78 ms
64 bytes from 10.60.0.254: icmp_seq=3 ttl=63 time=8.39 ms
64 bytes from 10.60.0.254: icmp_seq=4 ttl=63 time=8.46 ms
^C
--- 10.60.0.254 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 6.785/8.443/10.124/1.184 ms
```

Figure 4: Road Warrior(193.136.212.254) -> Certification Authority(10.60.0.254)

3.2 Apache

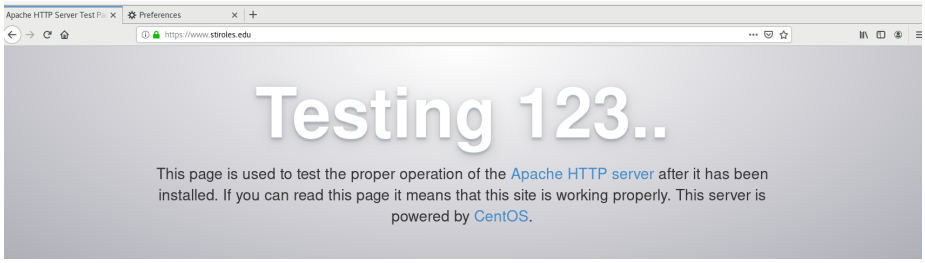


Figure 5: Connection to apache with RW using a valid certificate

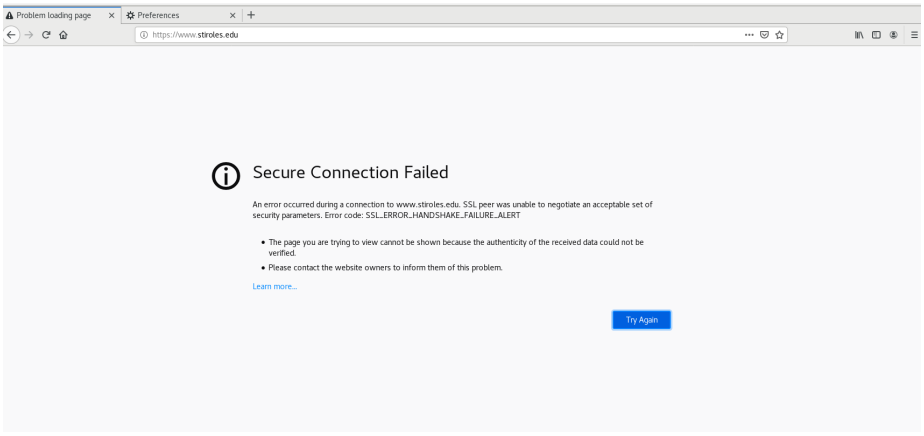


Figure 6: Connection to apache with RW using a revoked certificate

3.3 OCSP

We observed that whenever someone connects to Apache, the log file displays the status of the client that attempted to connect.

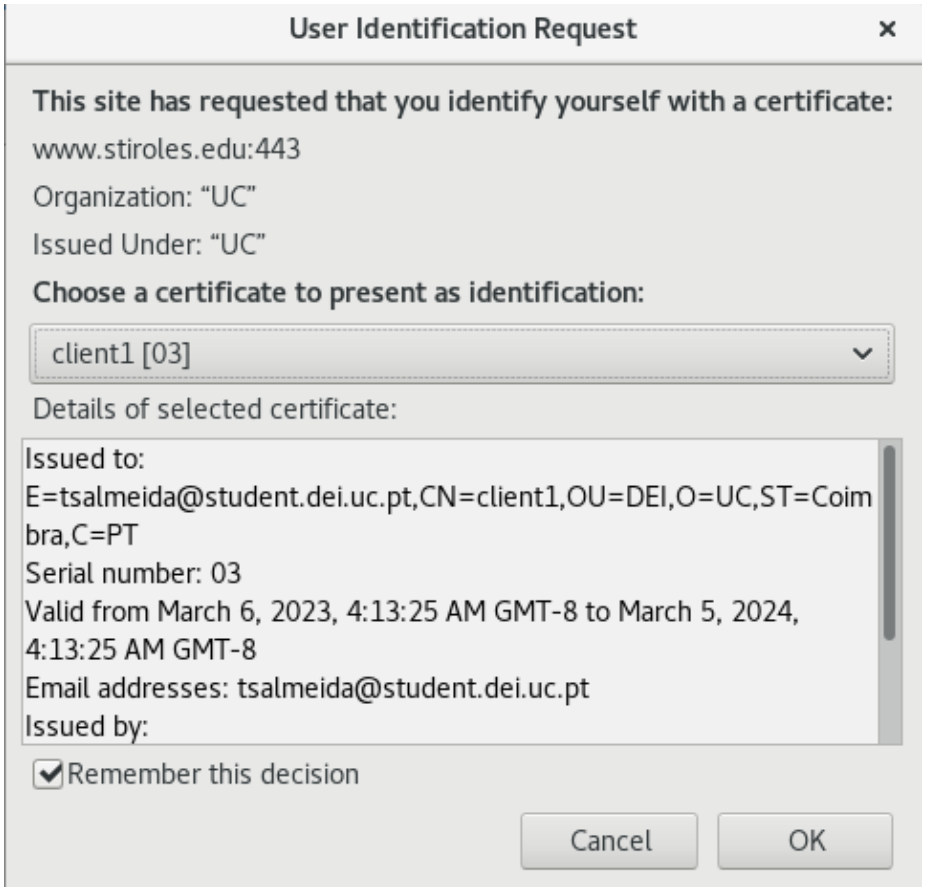


Figure 7: OCSP - certificate validation

```
OCSP Request Data:
    Version: 1 (0x0)
    Requestor List:
        Certificate ID:
          Hash Algorithm: sha1
          Issuer Name Hash: C84A43A7A453052A50FE0109A060D121D401F6F5
          Issuer Key Hash: 7282AF6F45D9EB1AC69AF4F3A67AD470EA393795
          Serial Number: 03
    Request Extensions:
        OCSP Nonce:
            0410323BEE92687A06535D1B40DFAF3BD475
OCSP Response Data:
    OCSP Response Status: successful (0x0)
    Response Type: Basic OCSP Response
    Version: 1 (0x0)
    Responder Id: C = PT, ST = Coimbra, L = Coimbra, O = UC, OU = DEI, CN = ca, emailAddress = tsalmeida@student.dei.uc$
    Produced At: Mar 12 12:06:01 2023 GMT
    Responses:
    Certificate ID:
      Hash Algorithm: sha1
      Issuer Name Hash: C84A43A7A453052A50FE0109A060D121D401F6F5
      Issuer Key Hash: 7282AF6F45D9EB1AC69AF4F3A67AD470EA393795
      Serial Number: 03
    Cert Status: good
    This Update: Mar 12 12:06:01 2023 GMT
```

Figure 8: OCSP - Log File with a valid certificate

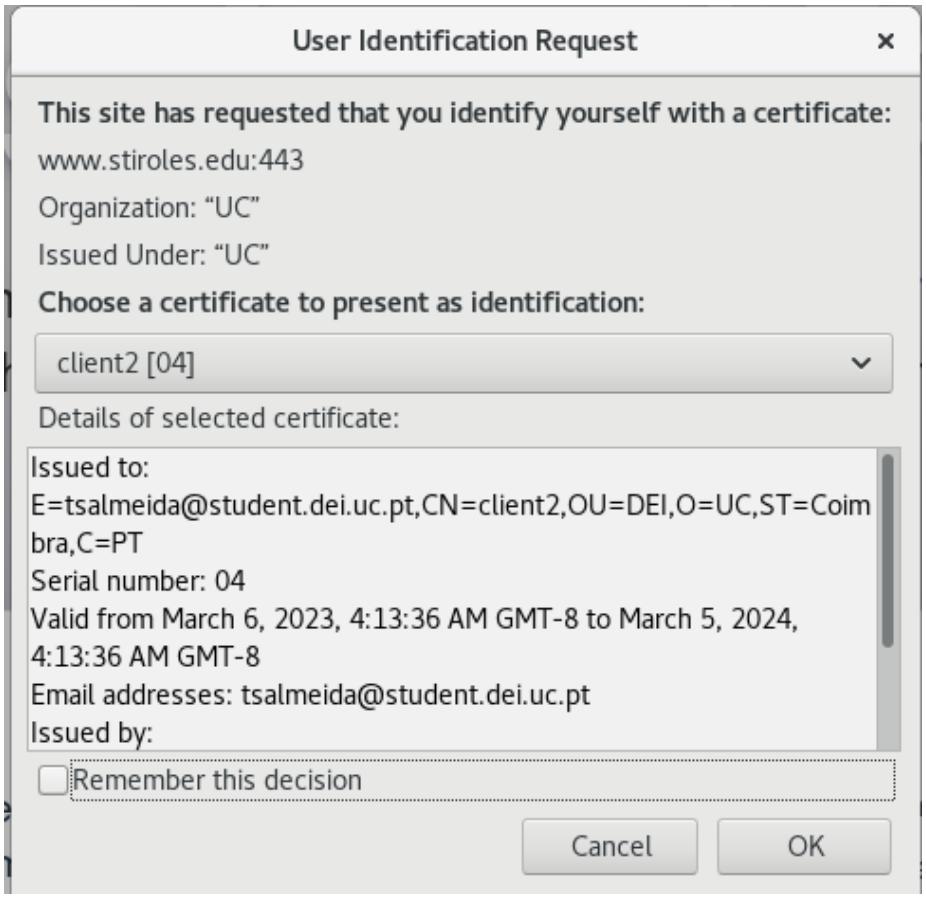


Figure 9: OCSP - certificate validation

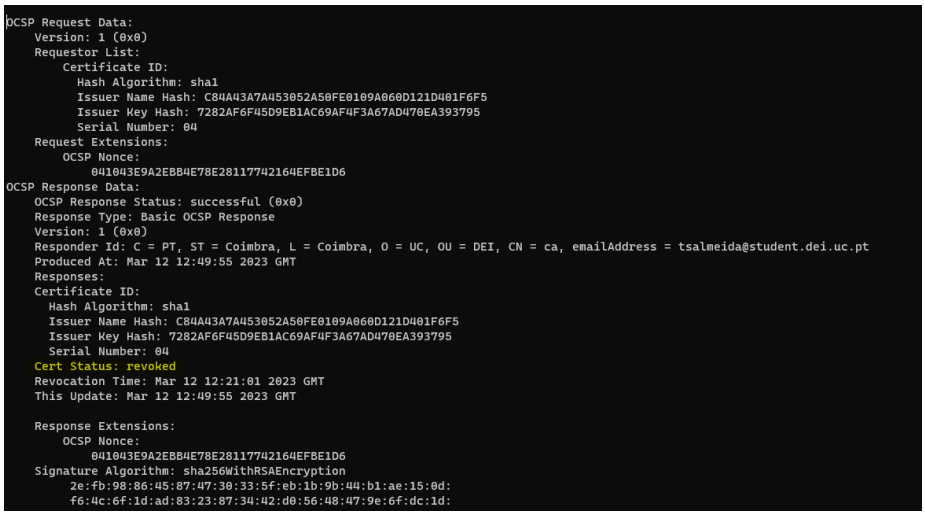


Figure 10: OCSP - Log File with a revoked certificate

3.4 Two-factor Authentication

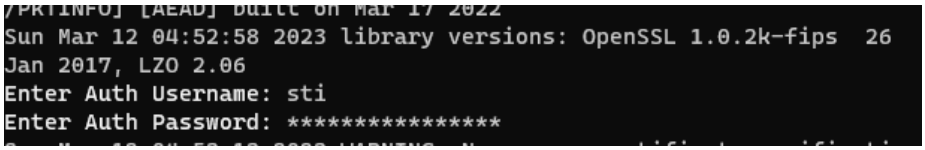


Figure 11: Two-factor Authentication - OpenVPN

Figure 12: Two-factor Authentication - OpenVPN with invalid credentials

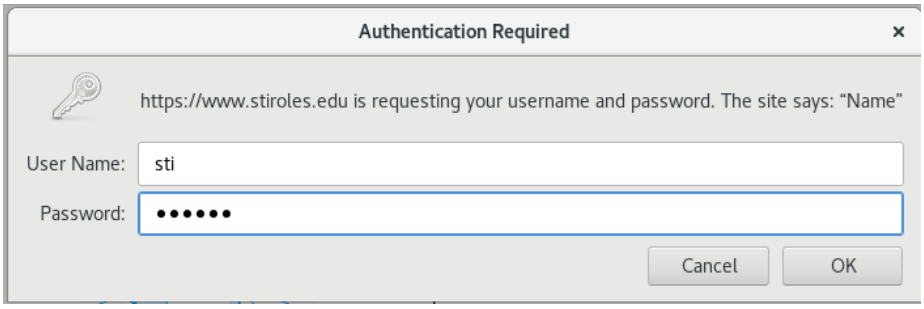


Figure 13: Two-factor Authentication - Apache

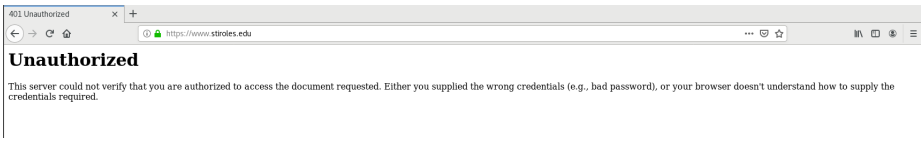


Figure 14: Two-factor Authentication - Apache with invalid credentials

4 Conclusion

In conclusion, with the creation of this small scenario, we learnt to operate with a few very important and relevant technologies (Certificates, OpenVPN, Apache, OCSP and 2FA), that allowed us deepen our knowledge in the field of Internet security. For future work, adding another network with its own could be an interesting challenge.

Bibliography

[1] Apache and google authenticator 1. https://www.techrepublic.com/article/pairing-apache-and-google-authenticator/. Accessed: 10-03-2023.

[2] Apache and google authenticator 2. https://www.blogbyben.com/2012/02/getting-google-authenticator-and-apache.html. Accessed: 10-03-2023.

[3] Apache and google authenticator 3. https://code.google.com/archive/p/google-authenticator-apache-module/wikis/GoogleAuthenticatorApacheModule.wiki. Accessed: 10-03-2023.

[4] Setup an openvpn server with certificate and two-factor authentication on centos 7. https://nethack.ch/2016/12/08/setup-an-openvpn-server-with-certificate-and-two-factor-authentication-on-centos-7/. Accessed: 10-03-2023.

[5] Jorge Granjal. *Segurança Prática em Sistemas e Redes com Linux*. FCA, 2017.