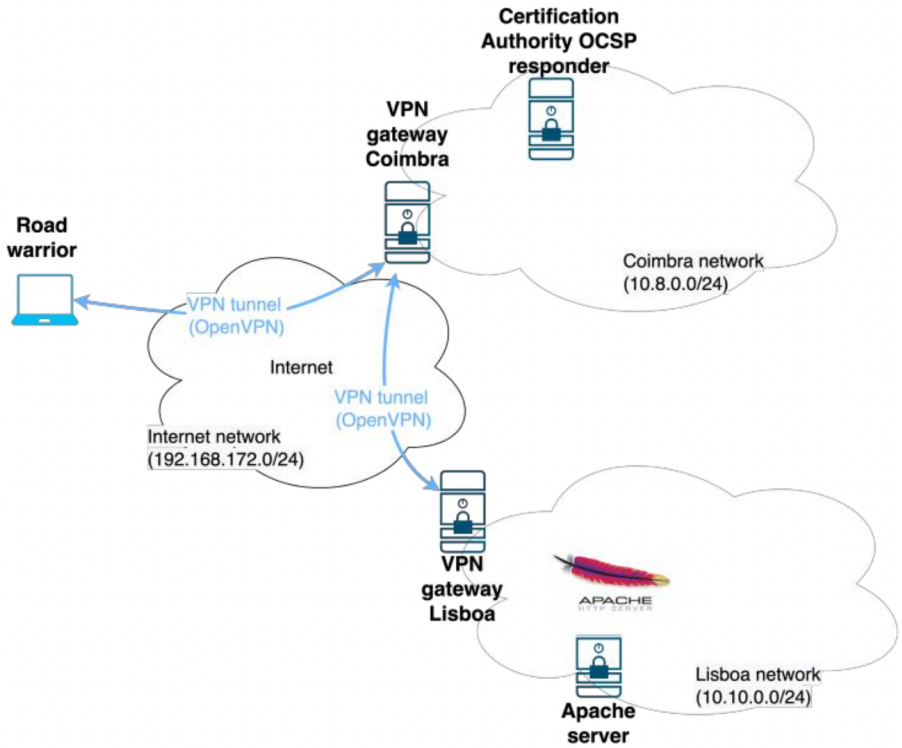
Cenários de VPN com Maquinas Virtuais

Trabalho Prático 1

Bruno Faria e Dylan Perdigão
2018295474, 2018233092
{brunofaria, dgp}@student.dei.uc.pt
Departamento de Engenharia Informática, Universidade de Coimbra

Março 2022



Conteúdo

1 Introdução

O objetivo do seguinte trabalho prático é configurar duas redes internas, uma em Coimbra e outra em Lisboa, simuladas através de Maquinas Virtuais (VMs)e criar túneis vpn de forma a haver comunicação entre estas. A rede de Lisboa é constituída por um servidor Apache, pelo que um cliente que se ligue através de um túnel OpenVPN ao servidor de Coimbra deve ser redireccionado por outro túnel OpenVPN para o servidor de Lisboa de forma a aceder ao servidor apache. Para além disto foi implementado autenticação com Google Authenticator assim como gestão de certificados com OCSP.

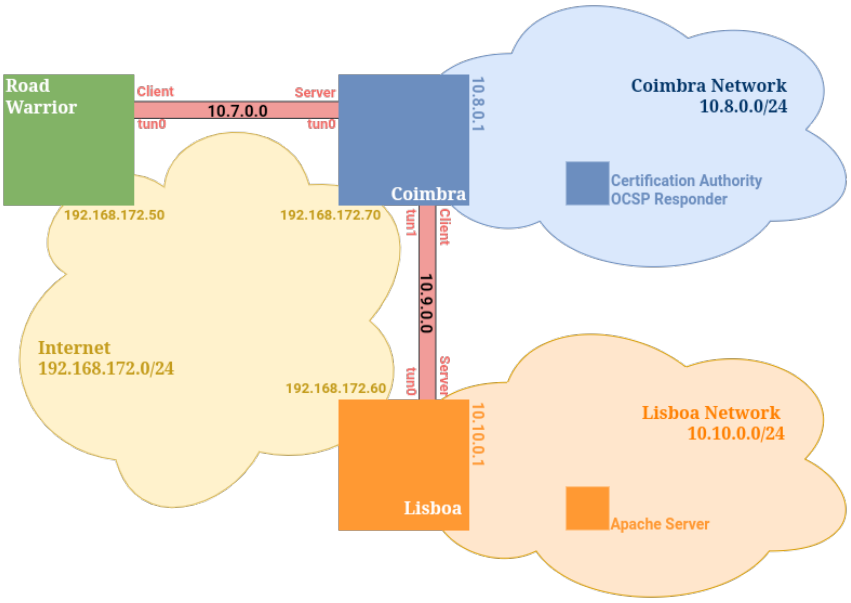


Figura 1: Esquema geral implementado

O cenário experimental é composto por 3 máquinas virtuais (Cliente, Coimbra, Lisboa) cujo os endereços IPv4 das redes global e local podem ser verificados na figura anterior. Nela conseguimos identificar também os ips das redes assim como dos túneis vpn.

2 Configuração Inicial das Maquinas Virtuais

As Maquinas virtuais foram configuradas no *VMWare* com vários *Networks Adapaters* de forma a suportarem ligações: (i) por SSH com o computador *host*, (ii) para rede Internet do cenário, (iii) para as rede internas (caso de Coimbra e Lisboa).

	Global	Local
Coimbra	192.168.172.70/24	10.8.0.1/24
Lisboa	192.168.172.60/24	10.10.0.1/24
Road Warrior	192.168.172.50/24	-

Tabela 1: Endereços IP das máquinas virtuais

Nas configurações de rede do Debian na VM, é possivél criar perfis de rede definindo manualmente na secção "*IPv4*"os endereços IP da rede local e global para cada maquina como mostra a figura 2. Foi então efetuada estas configurações ficando com os ips da tabela 1.

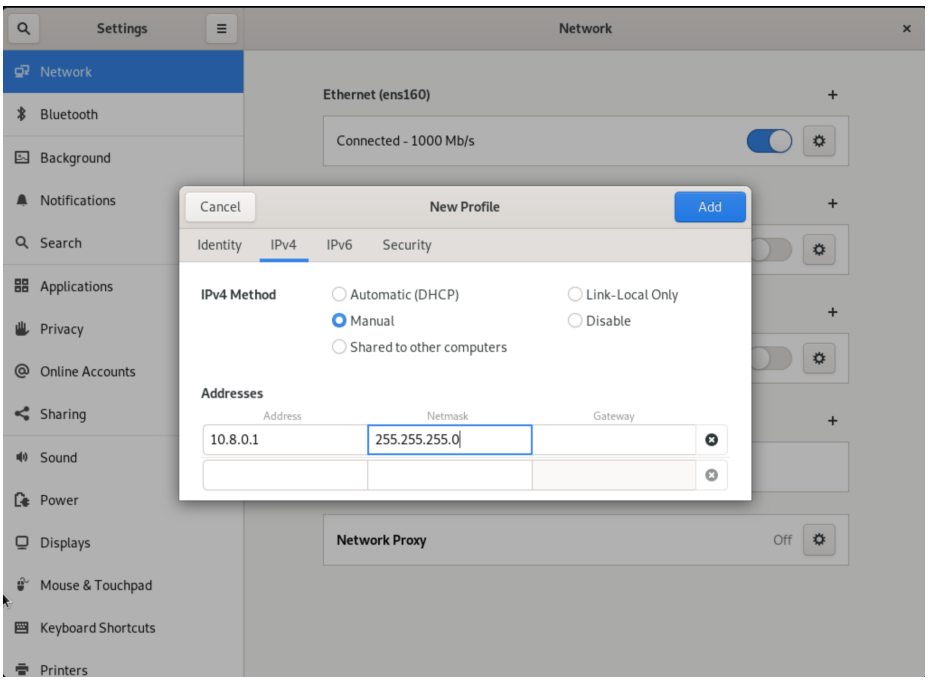


Figura 2: Configuração da Rede Local de Coimbra

3 Autoridade de Certificação

A Autoridade de Certificação (CA) permite emitir e assinar certificados para as diferentes entidades do presente trabalho prático (Gateways do OpenVPN, OCSP e Apache). Recordamos que a CA é sediada na VM de Lisboa.

Antes de recorrer aos comandos do OpenSSL, é necessário criar as diretorias e ficheiros que permitem armazenar as chaves e certificados das entidades. Edita-se também o ficheiro */etc/ssl/openssl.cnf* de forma a atualizar os caminhos das diretorias consoante o que foi préviamente criado (código 1).

```
...
    ####################################################################
    [ CA_default ]
    dir             = /etc/pki/CA
    certs           = $dir/certs            # Where the issued certs are kept
    crl_dir         = $dir/crl              # Where the issued crl are kept
    database        = $dir/index.txt        # database index file.
    #unique_subject = no                    # Set to 'no' to allow creation of
                                            # several certs with same subject.
    new_certs_dir   = $dir/newcerts         # default place for new certs.
    certificate     = $dir/certs/ca.crt     # The CA certificate
    serial          = $dir/serial           # The current serial number
    crlnumber       = $dir/crlnumber        # the current crl number
                                            # must be commented out to leave a V1 CRL
    crl             = $dir/crl/ca.crl       # The current CRL
    private_key     = $dir/private/ca.key   # The private key
    x509_extensions = usr_cert              # The extensions to add to the cert
...
```

Código 1: Configuração do OpenSSL para a CA

Os comandos do OpenSSL do código 2[2] permitem gerar o certificado da CA. Mais especificamente:

- Na linha 14, gera-se a chave RSA da CA protegida por uma password;
- Na linha 16, cria-se o CSR (*Certificate Signing Request*) em que se define os dados da CA como: (i) o país, (ii) o estado/distrito, (iii) a cidade, (iv) a organização, (v) a unidade da organização, (vi) o nome da entidade e (vii) o endereço e-mail;
- Na linha 20, o CSR é autoassinado pela chave RSA da CA, o que dá origem ao certificado final válido no nosso caso durante 3650 dias ($\approx$ 10 anos).

```
1  # Creates directories and files
2  mkdir /etc/pki/CA
3  cd /etc/pki/CA/
4  mkdir private
5  mkdir ca
6  mkdir certs
7  mkdir newcerts
8  mkdir crl
9  touch index.txt
10 echo 01 > serial
11 echo 01 > crlnumber
12 # Key
13 openssl genrsa -des3 -out private/ca.key  # pass: sti2022
14 # CSR
15 openssl req -new -key private/ca.key -out ca/ca.csr -subj \
16     /C=PT/ST=Coimbra/L=Coimbra/O=UC/OU=DEI \
17     /CN=CA-Coimbra/emailAddress=ca-coimbra@gmail.com
18 # Certificate
19 openssl x509 -req -days 3650 -in ca/ca.csr -out certs/ca.crt -signkey private/ca.key
```

Código 2: Criação da CA

4 Servidor OCSP

4.1 Configuração no OpenSSL

O protocolo OCSP permite validar certificados, isto é propagar a informação do estado de revogação dos certificados de forma automática. Para tal, é necessário ao criar certificados incluir a informação OCSP nos futuros certificados criados [3].

```
1 ...
2     [ usr_cert ]
3 ...
4     authorityInfoAccess          = OCSP;URI:http://127.0.0.1:81
5     [ v3_OCSP ]
6     basicConstraints = CA:FALSE
7     keyUsage = nonRepudiation, digitalSignature, keyEncipherment
8     extendedKeyUsage = OCSPSigning
9 ...
```

Código 3: Configuração do OpenSSL para o OCSP

O código 3 permite incluir a informação OCSP nos certificados, bem como configurar o endereço IPv4 (http://127.0.0.1) e o porte (81) onde o servidor irá correr.

4.2 Criação do Certificado

À semelhança da criação do certificado da CA é necessário gerar uma chave RSA que srá usada na criação do CSR, sendo este necessário para a obtenção do certificado (código 4). A diferença encontra-se no facto do certificado ser assinado desta vez pela CA (ou seja não é auto-assinado) e é construído com a *flag* **-extentions v3_OCSP** correspondendo à seçcão homónima do código 3.

```
cd /etc/pki/CA/
mkdir ocsp
# Key
openssl genrsa -des3 -out private/ocsp.key # pass: sti2022
# CSR
openssl req -new -key private/ocsp.key -out ocsp/ocsp.csr -subj \
    /C=PT/ST=Coimbra/L=Coimbra/O=UC/OU=DEI/CN=OCSP/emailAddress=ocsp@gmail.com
# Certificate
openssl ca -in ocsp/ocsp.csr -cert certs/ca.crt -keyfile private/ca.key -out \
    certs/ocsp.crt -extensions v3_OCSP
```

Código 4: Criação do Certificado do OCSP

4.3 Correr o Servidor

A ativação do serviço OCSP recorre ao código 5. É necessário indicar o ficheiro **index.txt** para saber o estado dos certificados geridos pela AC, bem como o certificado e a chave privada da mesma para assinar digitalmente os certificados emitidos. Finalmente o ficheiro **log.txt** guarda os pedidos de validação enviados ao servidor OCSP.

Com a execução do comando, o servidor OCSP escutar as ligações de clientes pelo porte 81 no endereço "127.0.0.1".

```
cd /etc/pki/CA/
touch log.txt
openssl ocsp -index index.txt -port 81 -rsigner certs/ocsp.crt -rkey private/ocsp.key
     -CA certs/ca.crt -text -out log.txt
```

Código 5: Criação do Certificado do OCSP

5 VPNs

Como mostra a figura 1, as redes de Lisboa e Coimbra, assim como o *Road Warrior* e a rede de Coimbra, estão interconectadas por um túnel vpn. Estes dois túneis foram criados com o uso da ferramenta OpenVPN seguindo uma configuração cliente - servidor. Formamos assim as ligações Coimbra (servidor/tun0) - *Road Warrior* (cliente/tun0) com os ips em 10.7.0.0/24 e Lisboa (servidor/tun0) - Coimbra (cliente/tun1) com os ips em 10.9.0.0/24.

5.1 Criação de Certificados

Primeiramente foram criados, pela CA em Coimbra, keys e certificados para cada ponta dos túneis, código 6, assim como keys para *tls-auth* (ta.key) e *diffie-hellmann* (dh2048.pem) necessárias para a configuração dos túneis.

```
cd /etc/pki/CA/
mkdir openvpn
# diffie-hellmann
openssl dhparam -out openvpn/dh2048.pem 2048
# tls-auth
sudo openvpn --genkey secret private/ta.key
####    Cert TUN0-Client    ####
# Key
openssl genrsa -des3 -out private/tun0-client.key # pass: sti2022
# CSR
openssl req -new -key private/tun0-client.key -out openvpn/tun0-client.csr -subj \
    /C=PT/ST=Coimbra/L=Coimbra/O=UC/OU=DEI\
    /CN=TUN0-Client/emailAddress=tun0-client@gmail.com
# Certificate
openssl ca -in openvpn/tun0-client.csr -cert certs/ca.crt -keyfile private/ca.key \
    -out certs/tun0-client.crt
####    Cert TUN0-Coimbra    #### ...
####    Cert TUN1-Coimbra    #### ...
####    Cert TUN1-Lisboa     #### ...
```

Código 6: Criação das keys e certificados necessários, feito em Coimbra

5.2 Configurações

Para configurar os servidores e clientes dos túneis foram criados os ficheiros client.conf e server.conf partindo dos exemplos dados pelo openvpn, alterando as paths dos certificados e keys para as das previamente criadas. O conteúdo destes ficheiros podem ser verificados nos códigos 7, 8, 9 e 10.

```
local       192.168.172.60
port        1194 # DIFFERENT FROM TUN1
proto       udp
dev         tun
ca          /etc/pki/CA/certs/ca.crt
cert        /etc/pki/CA/certs/tun1-lisboa.crt
key         /etc/pki/CA/private/tun1-lisboa.key
dh          /etc/pki/CA/openvpn/dh2048.pem
server      10.9.0.0 255.255.255.0
ifconfig-pool-persist /var/log/openvpn/ipp.txt
push        "route 10.7.0.0 255.255.255.0"
push        "route 10.8.0.0 255.255.255.0"
push        "route 10.10.0.0 255.255.255.0"
keepalive   10 120
tls-auth    /etc/pki/CA/private/ta.key 0
cipher      AES-256-CBC
persist-key
persist-tun
status      /var/log/openvpn/openvpn-status.log
verb        3
explicit-exit-notify 1
```

Código 7: Configuração do tun0 de Lisboa (server.conf)

```
client
dev          tun
proto        udp
remote       192.168.172.60 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca           /etc/pki/CA/certs/ca.crt
cert         /etc/pki/CA/certs/tun1-coimbra.crt
key          /etc/pki/CA/private/tun1-coimbra.key
tls-auth     /etc/pki/CA/private/ta.key 1
cipher       AES-256-CBC
verb         3
```

Código 8: Configuração do tun1 de Coimbra (client.conf)

```
plugin       /usr/lib/openvpn/openvpn-plugin-auth-pam.so \"login login USERNAME
    password PASSWORD pin OTP\"
local        192.168.172.70
port         1195 # DIFFERENT FROM TUN1
proto        udp
dev          tun
ca           /etc/pki/CA/certs/ca.crt
cert         /etc/pki/CA/certs/tun0-coimbra.crt
key          /etc/pki/CA/private/tun0-coimbra.key
dh           /etc/pki/CA/openvpn/dh2048.pem
server       10.7.0.0 255.255.255.0
ifconfig-pool-persist /var/log/openvpn/ipp.txt
push         \"route 10.8.0.0 255.255.255.0\"
push         \"route 10.9.0.0 255.255.255.0\"
push         \"route 10.10.0.0 255.255.255.0\"
keepalive    10 120
tls-auth     /etc/pki/CA/private/ta.key 0
cipher       AES-256-CBC
persist-key
persist-tun
status       /var/log/openvpn/openvpn-status.log
script-security 2
tls-verify /etc/pki/OCSP_check.sh
verb         3
explicit-exit-notify 1
```

Código 9: Configuração do tun0 de Coimbra (server.conf)

```
auth-user-pass
client
dev          tun
proto        udp
remote       192.168.172.70 1195
resolv-retry infinite
nobind
persist-key
persist-tun
ca           /etc/pki/CA/certs/ca.crt
cert         /etc/pki/CA/certs/tun0-client.crt
key          /etc/pki/CA/private/tun0-client.key
tls-auth     /etc/pki/CA/private/ta.key 1
cipher       AES-256-CBC
verb         3
```

Código 10: Configuração do tun0 do Road Warrior (client.conf)

Aqui, as maiores diferenças são em relação à autenticação e ao OCSP que foram faladas nas secções 7 e 4, respectivamente. Relativamente ao OCSP, o OpenVPN disponibilizou no seu Github [4] um script para efetuar a verificação dos estados dos certificados quando está à escuta, pelo que o servidor em Coimbra precisa na configuração da flag *tls-verify*.

5.3 Routing

De forma a que o *Road Warrior* consiga comunicar com a rede de Lisboa, de forma a aceder ao servidor apache foi necessário adicionadar os comandos *push-route* para fazer a ligação entre as redes. Além disso, foi preciso ativar o *ip forward* da máquina de Coimbra (que corre o servidor e cliente dos túneis). Isto foi feito usando os comandos:

- sudo sysctl -w net.ipv4.ip_forward=1
- sudo iptables -t nat -A POSTROUTING -s 10.7.0.0/24 -o tun0 -j MASQUERADE

6 Servidor apache

Numa primeira fase é necessário gerar um certificado para o servidor apache como visto anteriormente. Após efetuar o download do servidor Apache pela linha de comando do Debian e correr o comando "*a2ensite default-ssl*", altera-se o ficheiro *default-ssl.conf* como no código 11 para associar os certificados da CA e do servidor Apache. Finalmente, initializa-se o serviço *apache2*.

```
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html
        ErrorLog \${APACHE_LOG_DIR}/error.log
        CustomLog \${APACHE_LOG_DIR}/access.log combined
        SSLEngine on
        SSLCertificateFile      /etc/pki/CA/certs/apache.crt
        SSLCertificateKeyFile   /etc/pki/CA/private/apache.key
        SSLCACertificateFile    /etc/pki/CA/certs/ca.crt
        <FilesMatch \"\.(cgi|shtml|phtml|php)\$\">
            SSLOptions +StdEnvVars
        </FilesMatch>
        <Directory /usr/lib/cgi-bin>
            SSLOptions +StdEnvVars
        </Directory>
    </VirtualHost>
</IfModule>
```

Código 11: Configuração do Ficheiro *default-ssl.conf*

No entanto é necessário instalar o certificado da CA nos *browsers* das três Maquinas Virtuais para testar ligar por HTTPS a partir de cada uma das localizações (ver secção 8). No *FireFox* importa-se o certificado indo para ***Settings*** > ***Privacy and Security*** > ***Certificates*** > ***View Certificate*** > ***Authorities***. Depois carrega-se no botão ***Import*** e escolhe-se o certificado da CA (no nosso caso *ca.crt*).

De notar que para ter uma ligação HTTPS é preciso mudar o nome do host do servidor apache para o mesmo nome que o seu certificado associado, ou seja o certificado é *apache.crt* logo no ficheiro "*/etc/hosts*"adiciona-se a linha:

- 10.10.0.1 apache

Sendo o endereço 10.10.0.1 o endereço onde se localiza o servidor Apache.

7 Authenticação com Google Authenticator

Entre o cliente e Coimbra, usa-se na VPN uma autenticação por OTP (*One-Time Password*) que recorre ao uso do *Google Autenticator*. A aplicação móvel permite gerar um codigo de 6 dígitos válida num período máximo de 4 minutos como mostra a figura 3. Esse código de 6 dígitos é usado

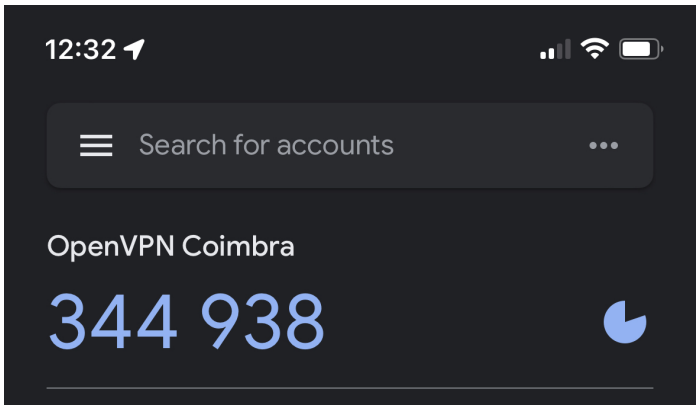


Figura 3: OTP no Google Autenticator

como meio de autenticação junto do nome de utilizador e da password do utilizador na Maquina Virtual. Ao ligar o cliente e quando for pedida a password é necessário concatenar a password do utilizador e a OTP que aparece na aplicação da Google. Por exemplo se a passord for "**sti**"e a OTP for "**344938**", então a password resultante é "**sti344938**".

A configuração do lado do cliente é simples, basta adicionar no ficheiro */etc/openvpn/client.conf* a linha:

- auth-user-pass

que permite pedir as credenciais para serem validadas no servidor de Coimbra.

Do lado da Cidade dos Estudantes, é preciso adicionar ao */etc/openvpn/server.conf* a linha que permite ativar o plugin:

- plugin /usr/lib/openvpn/openvpn-plugin-auth-pam.so "login login USERNAME password PASSWORD pin OTP"

Finalmente e como mostra o código 12 [1] cria-se um utilizador para o *Google Authenticator* e executa-se o comando da linha 9 para gerar o QRCode para scanear pela aplicação. Uma vez o código lido, a aplicação gera aleatoriamente OTPs associados ao servidor de Coimbra a cada 30 segundos (válidos numa janela de 4 minutos).

```
1 # creates gauth user
2 addgroup gauth
3 useradd -g gauth gauth
4 sudo google-authenticator
5 mkdir google-authenticator
6 chown gauth:gauth google-authenticator
7 chmod 0700 google-authenticator
8 # geenerates QR Code and saves emergency scratch codes
9 su -c "google-authenticator -t -d -r3 -R30 -f -l 'OpenVPN Server' -s /etc/openvpn/
      google-authenticator/{USER}" - gauth
```

Código 12: Geração do QR Code para Introduzir na Aplicação

8 Testes

8.1 Ligações entre VMs

Os primeiros testes efetuados consistiram na verificação de Ligações entre maquinas virtuais. Através da tabela 2 é possivel observar que as ligações uniderecionais no sentido *Road Warrior*-Coimbra e Coimbra-Lisboa estão funcionais. Após esta verificação foi então testada a ligação *Road Warrior*-Lisboa, que também foi sucedida. Para tal recorreu-se ao comando *ping* para ver se os pacotes estão a ser encaminhados para as redes 10.8.0.1 e 10.10.0.1, ou seja respetivamente as redes internas de Coimbra e Lisboa. De notar que para testar a ligação com origem no cliente é fundamental ter a autenticação com o *Google Autenticator* (OTP) a funcionar.

Origem\Destino	**Road Warrior**	**Coimbra**	**Lisboa**
Road Warrior	N/A	Sim	Sim
Coimbra	-	N/A	Sim
Lisboa	-	-	N/A

Tabela 2: Pings entre Maquinas Virtuais

8.2 Acesso ao Servidor Apache de Lisboa

O acesso ao servidor Apache é testado através do *browser* introduzindo os seguintes endereços na barra de pesquisa:

- `http://apache/`
- `https://apache/`

Pelo que a tabela 3 mostra que foi possível aceder com sucesso às paginas tanto a partir de Lisboa onde está hospedada, bem como a partir de Coimbra e do cliente.

	HTTP	**HTTPS**
Road Warrior	Sim	Sim
Coimbra	Sim	Sim
Lisboa	Sim	Sim

Tabela 3: Acesso ao Servidor Apache

8.3 Revogação de Certificados

A revogação de certificados foi testada revogando o certificado do apache. Primeiramente foi testada a ligação por *https* ao apache pelo *Road Warrior*, verificando que estava funcional. Após isso, foi revogado o certificado do apache com o seguinte comando:
openssl ca -revoke certs/apache.crt -keyfile private/ca.key -cert certs/ca.crt
Seguida da revogação, foi testada novamente a ligação por *https* ao apache pelo *Road Warrior*, sendo que, desta vez, a ligação não foi possível dando aviso de certificado revogado.

8.4 Autenticação

A autenticação foi testada tentando fazer *login* com credenciais erradas, o que não permitia ligação ao túnel vpn para Coimbra.

9 Conclusão

O presente trabalho prático proporcionou uma aprendizagem mais aprofundada sobre a criação e uso de certificados, bem como de redes virtuais privadas com OpenVPN e servidores www seguros com o Apache. Implicitamente o uso de chaves RSA e Certificados fez apelo aos nosso conhecimentos sobre encriptação simétrica e assimétrica.

Como trabalho futuro, seria possível fazer com que as maquinas virtuais estejam ligadas bidireccionalmente e acrescentar outros mecanismos de segurança para tornar as ligações ainda mas seguras.

Referências

[1] *Extending a Debian openvpn server with Multi Factor Authentication via google authenticator.* URL: `https://ulimit.nl/2019/08/27/extending-a-debian-openvpn-server-with-multi-factor-authentication-via-google-authenticator/`.

[2] Jorge Granjal. *Gestão de Sistemas e Redes em Linux, 3ª Edição Actualizada e Aumentada.* 3ª ed. Vol. 1. FCA, Editora de Informática, set. de 2013. ISBN: 978-972-722-784-6.

[3] Jorge Granjal. *Segurança Prática em Sistemas e Redes com Linux.* 1ª ed. Vol. 1. FCA Informática, fev. de 2017. ISBN: 978-972-722-865-2.

[4] OpenVPN. *Github - OpenVPN/openvpn.* Fev. de 2016. URL: `https://github.com/OpenVPN/openvpn/blob/master/contrib/OCSP_check/OCSP_check.sh`.