

# Practical Assignment #2

João Neto – 2023234004

Vasco Alves – 2022228207

24 de abril de 2026

## Índice

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Criação de certificados</b>	<b>2</b>
<b>3</b>	<b>Configuração da <i>Gateway</i> VPN</b>	<b>3</b>
3.1	Configurar TOTP . . . . .	3
3.2	Aceder ao código . . . . .	3
3.3	Encaminhamento e Firewall . . . . .	3
<b>4</b>	<b>Configuração do Cliente (Road Warrior)</b>	<b>3</b>
<b>5</b>	<b>Servidor Apache e OCSP</b>	<b>4</b>
5.1	Revocation e OCSP . . . . .	5
<b>6</b>	<b>Conclusão</b>	<b>5</b>

## 1 Introdução

Este projecto tem como âmbito implementar uma rede virtual privada (VPN) em um cenário de road-warrior, ou seja, onde o administrador de acesso da rede é o cliente ou tem acesso a ele.

Para tal, foi implementado um servidor e um cliente OpenVPN, certificados por uma autoridade central (CA) que em si é self-signed. Para além disto, foi implementado um sistema de autenticação de dois factores através do plugin *google-authenticator* para o OpenVPN.

Existe ainda um servidor Apache e um servidor de OpenSSL OCSP. Para simplificar, a elaboração do projecto foram colocados na mesma maquina virtual, mas por razoes de segurança poderia querer ter estes serviços separados.

Temos então três máquinas virtuais:

Nome	Script	Rede
Road Warrior	VM_ROAD_WARRIOR.sh	Rede Externa 193.168.0.0/24
VPN Gateway	VM_OPENVPN_GATEWAY.sh	Router
OpenSSL / Apache	VM_OPENSSL_APACHE.sh	Rede Interna 10.60.0.0/24

## 2 Criação de certificados

Criar chaves com 2048 bits.

Todos os certificados são criados de uma so vez e são depois copiados para as respetivas máquinas virtuais.

```
1 cert_ca="/C=PT/ST=Coimbra/L=Coimbra/O=UC/CN=CoimbraVPN"
2 cert_vpn="/C=PT/ST=Coimbra/L=Coimbra/O=UC/CN=gateway"
3 cert_user="/C=PT/ST=Coimbra/L=Coimbra/O=UC/CN=warrior"
4 cert_apache="/C=PT/ST=Coimbra/L=Coimbra/O=UC/CN=apache.coimbra"
5
6 openssl genrsa -out "ca.key" 2048
7 openssl req -x509 -nodes -days 365 -key "ca.key" -out "ca.crt" -subj "$cert_ca"
8 openssl genrsa -out "vpn.key" 2048
9 openssl req -new -key "vpn.key" -out "vpn.csr" -subj "$cert_vpn"
10 openssl ca -batch -in "vpn.csr" -cert "ca.crt" -keyfile "ca.key" -out "vpn.crt"
    -config cheese.cfg
11 openssl dhparam -out "dh2048.pem" 2048
12 openvpn --genkey secret "ta.key"
13 openssl genrsa -out user.key
14 openssl req -new -key user.key -out user.csr -subj "$cert_user"
15 openssl ca -batch -in "user.csr" -cert "ca.crt" -keyfile "ca.key" -out "user.
    crt" -config cheese.cfg
16 openssl genrsa -out apache.key
17 openssl req -new -key apache.key -out apache.csr -subj "$cert_apache" -addext "
    subjectAltName = IP:10.60.0.1,DNS:apache"
18 openssl ca -batch -in "apache.csr" -cert "ca.crt" -keyfile "ca.key" -out "
    apache.crt" -config cheese.cfg
```

## 3 Configuração da Gateway VPN

### 3.1 Configurar TOTP

Foi criado o ficheiro `totp` com a configuração de autenticação a ser utilizada pelo plugin de PAM para o `openvpn`.

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

Adicionalmente, devido às restrições de segurança do `systemd`, foi necessário desativar o `ProtectHome` no serviço do OpenVPN para que o plugin PAM consiga ler os ficheiros de segredo do Google Authenticator localizados nas diretorias *home* dos utilizadores.

```
1 [Service]
2 ProtectHome=false
```

Primeiro, na gateway, entramos como o utilizador desejado e obtemos a chave do gerador de palavras passas temporárias. Ao inserir a chave no `google authenticator` podemos obter um código QR, a nossa primeira chave de 6 dígitos.

```
1 su john
2 google-authenticator
```

### 3.2 Encaminhamento e Firewall

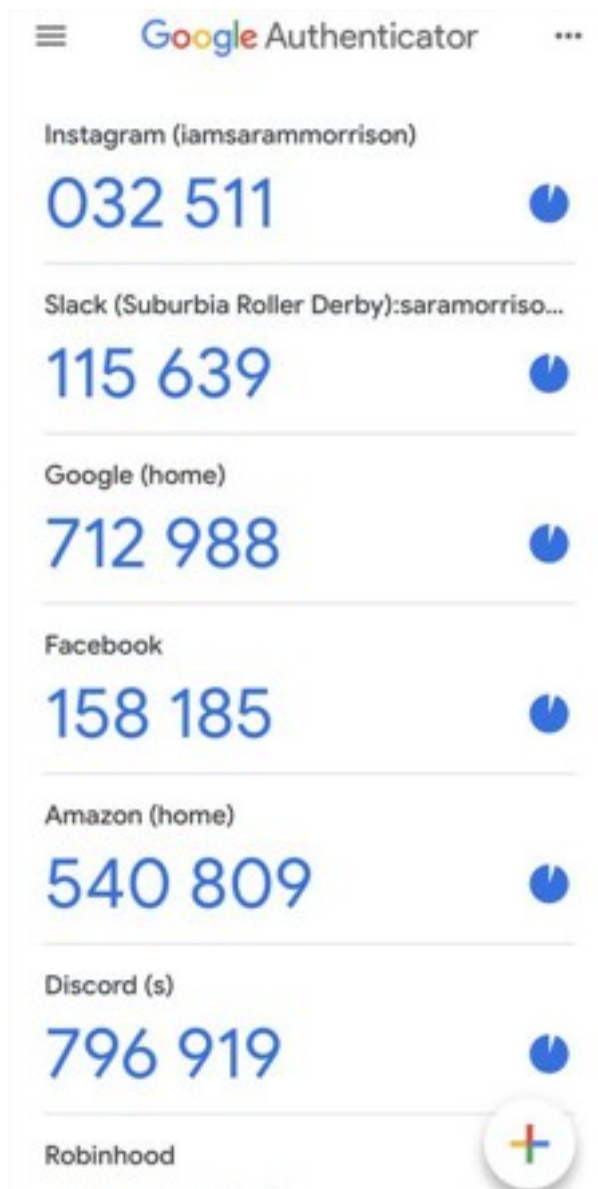
Para que a gateway funcione como router entre a rede externa e a rede interna, foi necessário ativar o *IP forwarding* no kernel e configurar as regras de *iptables* para permitir o tráfego da VPN e realizar o mascaramento de IP (NAT).

```
1 # Ativar encaminhamento
2 echo "net.ipv4.ip_forward = 1" >> /etc/sysctl.conf
3 sysctl -p /etc/sysctl.conf
4
5 # Regras de Firewall
6 iptables -I INPUT 1 -p udp --dport 1194 -j ACCEPT
7 iptables -I FORWARD 1 -i tun0 -o enp0s9 -j ACCEPT
8 iptables -I FORWARD 1 -i enp0s9 -o tun0 -j ACCEPT
9 iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o enp0s8 -j MASQUERADE
```

## 4 Configuração do Cliente (Road Warrior)

O cliente encontra-se na rede externa (193.136.212.10) e liga-se à VPN gateway na porta 1194. Para garantir a segurança, utiliza-mos autenticação mútua (os certificados X.509) e um *two factor authentication* (2FA) como palavras-passe temporárias, geradas através do *Google Authenticator*.

```
1 client
2 dev tun
3 proto udp
4 remote 193.136.212.1 1194
5 ca ca.crt
6 cert user.crt
7 key user.key
```



```
8 auth-user-pass
9 cipher AES-256-GCM
10 auth SHA256
```

## 5 Servidor Apache e OCSP

O servidor interno (10.60.0.1) alberga o serviço Apache e o responder OCSP da autoridade de certificação.

### 5.1 Revocation e OCSP

1. Estabelecer a ligação VPN e verificar a conectividade à rede interna.
2. No diretório da autoridade de certificação (máquina *host*), revogar o certificado do utilizador:

```
1 openssl ca -revoke user.crt -config cheese.cfg -keyfile ca.key -cert ca.crt
2
```

3. Atualizar o ficheiro `index.txt` no servidor OSCP e reiniciar o serviço para carregar o novo estado de revogação.
4. Tentar estabelecer uma nova ligação VPN e verificar que a autenticação falha devido à resposta `revoked` do responder OSCP.

## 6 Conclusão

A implementação deste projeto permitiu consolidar conhecimentos sobre redes privadas virtuais e segurança em comunicações. A combinação de certificados digitais com autenticação de dois fatores (TOTP) garante uma robustez significativa contra ataques de interceção e roubo de credenciais.

A integração do protocolo OSCP permite uma gestão dinâmica da confiança, possibilitando a revogação imediata de acesso a clientes comprometidos sem necessidade de redistribuição de listas de revogação (CRLs) volumosas. Em suma, o sistema cumpre os requisitos de confidencialidade, integridade e disponibilidade propostos.