Practical Assignment #1

Segurança em Tecnologias da Informação

Faculdade de Ciências e Tecnologias da Universidade de Coimbra

Dário Félix, 2018275530, dario@student.dei.uc.pt
Maria Dias, 2018274188, mddias@student.dei.uc.pt

Coimbra, 12 de março de 2023

1 Introdução

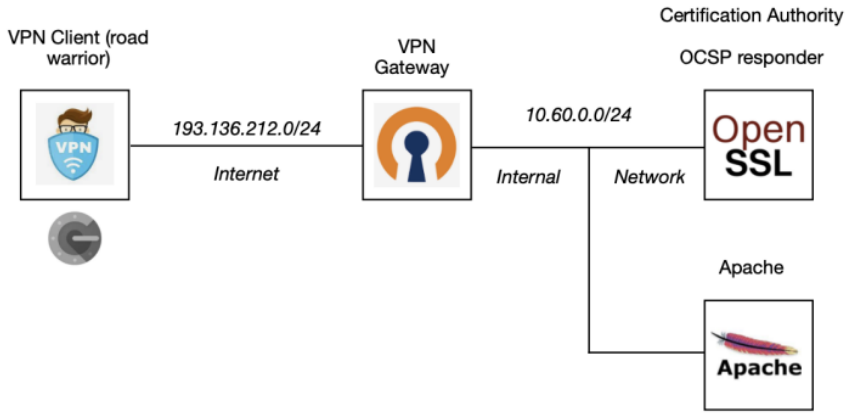

Figura 1: Configuração de rede proposta para o trabalho prático

Neste trabalho prático, foi-nos proposta a configuração de um cenário semelhante ao apresentado na Figura 1.

O primeiro passo deste trabalho prático passou pela criação de uma Autoridade de Certificação (CA) e criação de certificados assinados pela CA criada. A seguir, foi configurado o OpenVPN para permitir a comunicação do cliente com os serviços presentes na zona *Internal*.

O passo seguinte passou pela configuração dos serviços presentes no cenário apresentado (Apache e OCSP) e pela implementação de autenticação de dois fatores em ambos.

A etapa final deste trabalho prático consistiu em testes do trabalho desenvolvido.

2 Especificação dos Sistemas

Todo o trabalho foi realizado utilizando as versões dos *softwares* assinalados na Tabela 1.

Software	Versão
CentOS	CentOS Linux release 7.9.2009
Apache	Apache/2.4.6 (CentOS)
Mozilla Firefox	Mozilla Firefox 91.13.0esr
Thunderbird	Thunderbird 102.8.0
OpenSLL	OpenSSL 1.0.2k-fips
OpenVPN	OpenVPN 2.4.12-1.el7
Google Authenticator	Google Authenticator 1.04-1
GNU Privacy Guard	gpg (GnuPG) 2.0.22

Tabela 1: Versões dos *softwares* utilizados.

Visando simular o cenário documentado na Figura 1, utilizou-se o VMWare para simular a interação de três VMs que prestam serviços conforme estão descritos nas tabelas seguintes — **VM1** (VPN Client), na Tabela 2, **VM2** (VPN Gateway), na Tabela 3, e a **VM3** (CA & Apache), na Tabela 4.

ID	**VM 1**
Descrição	Representa o cliente: utiliza o OpenVPN Client, a aplicação cliente do Google Authenticator, o browser para aceder ao website, o Thunderbird para enviar o e-mail assinado e encriptado com GnuPG e o OCSP (Mozilla Firefox).
Serviços Instalados	OpenVPN; Mozilla Firefox; Thunderbird; GnuPG.

Tabela 2: Especificação da **VM1**.

ID	**VM 2**
Descrição	Representa o VPN gateway: utiliza o OpenVPN Server, o Google Authenticator, e o OCSP (OpenSSL).
Serviços Instalados	OpenVPN; Google Authenticator; OpenSSL.

Tabela 3: Especificação da **VM2**.

ID	**VM 3**
Descrição	Representa os serviços da rede interna: possui o Apache, OCSP Responder (OpenSSL) e o Certification Authority (OpenSSL).
Serviços Instalados	OpenSSL; Apache.

Tabela 4: Especificação da **VM3**.

3 Configuração Inicial

Para concretizar o cenário proposto neste trabalho, começámos pela atribuição de endereços a cada uma das máquinas. Desta forma, atribuímos a **VM1** (VPN Client) o endereço **193.136.212.1**, Tabela 5. Para a **VM2** (VPN Gateway) definimos o endereço **193.136.212.2** na zona *Internet* e **10.60.0.129** na zona *Internal*, Tabela 6. Para a **VM3** (CA & Apache) foi definido o endereço **10.60.0.130** e definido como *default gateway* o endereço **10.60.0.129** da VM2, Tabela 7.

Hardware	**Endereço**	**Rede**
VMnet0	193.136.212.1	193.136.212.0/24

Tabela 5: *Interfaces* de rede da **VM1**.

Hardware	**Endereço**	**Rede**
VMnet0	193.136.212.2	193.136.212.0/24
VMnet2	10.60.0.129	10.60.0.128/25

Tabela 6: *Interfaces* de rede da **VM2**.

Hardware	**Endereço**	**Rede**
VMnet2	10.60.0.130	10.60.0.128/25

Tabela 7: *Interfaces* de rede da **VM3**.

Na zona *Internal* foram definidas duas redes a partir da rede **10.60.0.0/24**, que foi previamente dividida em 2 redes (*subnetting*) — a rede **10.60.0.0/25**, que se destina à atribuição de endereços a novos clientes VPN, e a rede **10.60.0.128/25**, que se destina ao alojamento de máquinas ou serviços internos diretamente ligados a esta rede.

4 Gerar Certificados

4.1 Gestão de certificados com OpenSSL

Para a utilização do OpenSSL para gestão da CA, é necessário incluir na sua configuração uma extensão com informação sobre o endereço e o porto do *OCSP Responder*. Esta informação será incluída nos certificados quando a CA assinar um pedido de emissão de certificado (CSR) [1] [2].

```
$ nano /etc/pki/tls/openssl.cnf

# Informação a incluir nos certificados
[ usr_cert ]
authorityInfoAccess = OCSP;URI:http://ocsp.sti.pt:81
```

Antes da criação de certificados é necessária a criação de uma diretoria *newcerts*, de um ficheiro *index.txt* e de um ficheiro *serial* [1].

```
$ cd /etc/pki/CA
$ touch index.txt
$ echo '01' > serial
$ mkdir newcerts
```

4.2 Criação de Certificados

4.2.1 *Certification Authority*

```
# Criação chave simétrica
$ openssl genrsa -des3 -out ca.key
# Criação do CSR
$ openssl req -new -key ca.key -out ca.csr
    PT
    Coimbra
    Coimbra
    UC
    DEI
    STI
    ca@sti.pt
    1234
    sti
# Criação do certificado final a partir do CSR
$ openssl x509 -req -days 365 -in ca.csr -out ca.crt -signkey ca.key
```

4.2.2 OpenVPN

```
# Criação de uma chave privada para o OpenVPN
$ openssl genrsa -des3 -out openvpn.key
# Criação do CSR
$ openssl req -new -key openvpn.key -out openvpn.csr
    PT
    Coimbra
    Coimbra
    UC
    DEI
    openvpn
    openvpn@sti.pt
    1234
    sti_openvpn
# Criação do certificado final a partir do CSR
$ openssl ca -in openvpn.csr -cert ca.crt -keyfile ca.key -out openvpn.crt
```

4.2.3 Apache Web Server

```
# Criação de uma chave privada para o Apache
$ openssl genrsa -des3 -out apache.key
# Criação do CSR
$ openssl req -new -key apache.key -out apache.csr
    PT
    Coimbra
    Coimbra
    UC
    DEI
    www.apache.sti.pt
    apache@sti.pt
    1234
    www.sti.pt
# Criação do certificado final a partir do CSR
$ openssl ca -in apache.csr -cert ca.crt -keyfile ca.key -out apache.crt
```

4.2.4 VPN Client

```
# Criação de uma chave privada para o Cliente VPN
$ openssl genrsa -des3 -out user.key
# Criação do CSR
$ openssl req -new -key user.key -out user.csr
    PT
    Coimbra
    Coimbra
    UC
    DEI
    user
    user@sti.pt
    1234
    sti_user
# Criação do certificado final a partir do CSR
$ openssl ca -in user.csr -cert ca.crt -keyfile ca.key -out user.crt
# Conversão do certificado do user para o formato PKCS#12 para importação no
↪  browser
$ openssl pkcs12 -export -out user.p12 -inkey user.key -in user.crt -certfile
↪  ca.crt
```

4.3 Distribuição das Chaves e dos Certificados

Após a criação dos certificados, foram transferidos para a **VM1** (VPN Client) e para a **VM2** (VPN Gateway) os certificados e as chaves privadas respeitantes a cada um deles. Foi transferido também o certificado da CA. Note-se que os endereços de rede utilizados aqui são temporários que permitem conectar com o *host* e as diferentes máquinas virtuais entre si.

```
# Transferência de ficheiros para o VPN Client
$ scp user.key root@192.168.78.128:/etc/openvpn/user.key
$ scp user.key root@192.168.78.128:/etc/pki/CA/user.key
$ scp user.crt root@192.168.78.128:/etc/openvpn/user.crt
$ scp user.crt root@192.168.78.128:/etc/pki/CA/user.crt
$ scp ca.crt root@192.168.78.128:/etc/openvpn/ca.crt
$ scp ca.crt root@192.168.78.128:/etc/pki/CA/ca.crt
$ scp user.p12 root@192.168.78.128:/etc/pki/CA/user.p12

# Transferência de ficheiros para o VPN Gateway
$ scp openvpn.key root@192.168.78.129:/etc/openvpn/openvpn.key
$ scp openvpn.crt root@192.168.78.129:/etc/openvpn/openvpn.crt
```

```
$ scp ca.crt root@192.168.78.129:/etc/openvpn/ca.crt
```

5 Name Resolution

Para podermos aceder aos endereços definidos anteriormente pelos quais o Apache e o *OCSP Responder* irão responder, é necessário recorrer ao ficheiro **/etc/hosts** para fazer corresponder a esses domínios, o endereço IP da **VM3** (Apache & CA) [1].

5.1 VM3 (Apache & CA)

```
$ nano /etc/hosts
127.0.0.1 ocsp.sti.pt
127.0.0.1 www.apache.sti.pt
```

5.2 VM2 (VPN Gateway)

```
$ nano /etc/hosts
10.60.0.130 ocsp.sti.pt
```

5.3 VM1 (VPN Client)

```
$ nano /etc/hosts
10.60.0.130 ocsp.sti.pt
10.60.0.130 www.apache.sti.pt
```

6 *OCSP Responder*

6.1 *A priori*

É necessário possuir o ficheiro *crlnumer*.

```
$ nano /etc/pki/CA
$ echo '01' > crlnumber
```

6.2 Testar OCSP

Para testar o funcionamento do OCSP iremos através da linha de comandos revogar o certificado do utilizador user [1].

```
$ openssl ca -revoke user.crt
```

A seguir, criamos um CRL com a informação dos certificados revogados pela CA [1].

```
$ openssl ca -gencrl -out ca.crl
```

De forma a verificar o estado de validade do certificado do utilizador com recurso ao OCSP, é necessário ativar o serviço *OCSP Responder* [1].

```
$ openssl ocsp -index /etc/pki/CA/index.txt -port 81 -rsigner ca.crt -rkey
↪  ca.key -CA ca.crt -text -out log.txt
```

O comando seguinte verifica o estado de validade do certificado, com recurso ao servidor OCSP.

```
$ openssl ocsp -CAfile ca.crt -issuer ca.crt -cert user.crt -url
↪  http://ocsp.sti.pt:81 -resp_text
```

6.3 Configuração do Apache para suporte de OCSP

```
$ nano /etc/httpd/conf.d/ssl.conf

# Configuração do Apache para suporte de OCSP
SSLOCSPEnable on
SSLOCSPDefaultResponder http://ocsp.sti.pt:81
SSLOCSPOverrideResponder off
```

Após a alteração da configuração do ficheiro de configuração do Apache, é necessário reinicializar o serviço.

```
$ systemctl restart httpd
```

Do lado do cliente, é necessário confirmar que a opção *Consultar os servidores de resposta OCSP para confirmar a validade de certificados* nas definições do browser está ativa.

6.4 Criação de um novo certificado para o utilizador

```
$ openssl genrsa -out user2.key 1024 -des3
$ openssl req -new -key user2.key -out user2.csr
        PT
       Coimbra
       Coimbra
       UC
       DEI
       user2
       user2@sti.pt
       1234
       sti_user2
 $ openssl ca -in user2.csr -cert ca.crt -keyfile ca.key -out user2.crt
 $ openssl pkcs12 -export -out user2.p12 -inkey user2.key -in user2.crt
  ↪  -certfile ca.crt
```

Após a criação do certificado para o novo utilizador, os ficheiros *user2.key*, *user2.crt* e *user2.p12* foram transferidos para a **VM1** (VPN Client).

6.5 Script para o VPN Server comunicar com o *OCSP Responder*

De seguida, procedeu-se à criação de um script para o VPN Server poder comunicar com o OCSP Responder [1].

```
$ cd /etc/openvpn/
$ nano OCSP.sh
#!/bin/bash

if [ "$1" -ne 0 ]
then
    exit_code=0
else
    if [ -n "${tls_serial_0}" ]
    then
        cmd=$(openssl ocsp -issuer /etc/openvpn/ca.crt -CAfile
        ↪  /etc/openvpn/ca.crt -url http://ocsp.sti.pt:81 -serial
        ↪  "${tls_serial_0}")
        if [ $? -eq 0 ]
        then
            if echo "$cmd" | grep -Fq "${tls_serial_0}: good"
            then
                exit_code=0
            else
```

```
                exit_code=1
            fi
        else
            exit_code=1
        fi
    else
        exit_code=1
    fi
fi
exit "$exit_code"

$ chmod 777 OCSP.sh
```

7 OpenVPN

7.1 Servidor

Para podermos aceitar a receção de ligações VPN autenticadas por certificados X.509 é necessária a configuração do ficheiro *server.conf*. Através da configuração deste ficheiro, será atribuído um endereço na gama definida (10.60.0.2 a 10.60.0.127) a cada um dos clientes VPN [1] [3].

```
$ cd /etc/openvpn
$ nano server.conf

# Endereço IPV4 e porto no qual o servidor irá aceitar novas ligações
local 193.136.212.2
port 1194

# Definição do protocolo e dispositivos a utilizar para encapsular o tráfego
↪  IP
proto udp
dev tun

# Localização dos ficheiros com os certificados da AC e do servidor, bem como
↪  da chave privada do servidor
ca ca.crt
cert openvpn.crt
key openvpn.key

# Nome do ficheiro com os valores para o algoritmo Diffie-Hellman
dh dh1024.pem

# Gama de endereços a utilizar para a atribuição aos clientes VPN. O servidor
↪  atribui à interface "tun0" o primeiro endereço da gama
server 10.60.0.0 255.255.255.128

# Rotas a enviar aos clientes VPN para que estes consigam comunicar com redes
↪  internas à organização, através do serviço VPN.
push "route 10.60.0.128 255.255.255.128"

# Autenticação
tls-auth ta.key 0 # This file is secret
plugin openvpn-plugin-auth-pam.so openvpn
#plugin openvpn-plugin-auth-pam.so login
tls-verify /etc/openvpn/OCSP.sh
script-security 2

## Ativar compressão nas ligações VPN
comp-lzo
```

7.1.1 Iniciar Servidor OpenVPN

Após a alteração do ficheiro de configuração do servidor é necessário fazer *restart* ao serviço. Para garantir o bom funcionamento do OpenVPN é necessário desativar a firewall.

```
$ systemctl stop firewalld
$ cd /etc/openvpn
openvpn --config /etc/openvpn/server.conf
```

7.2 Cliente

De forma a permitir a autenticação do cliente através do uso de certificados, é necessário proceder à configuração do ficheiro *client.conf*, no qual especificamos a localização dos ficheiros com os certificados da CA e do cliente e da chave privada [1].

```
$ cd /etc/openvpn
$ nano client.conf

client
dev tun
proto udp
remote 193.136.212.2 1194

## Localização dos ficheiros com os certificados da AC e do servidor, bem
↪  como da chave privada do servidor
ca ca.crt
cert user.crt
key user.key

# Autenticação
tls-auth ta.key 1 # This file is secret
auth-user-pass

## Ativar compressão nas ligações VPN
comp-lzo
```

7.2.1 Iniciar Cliente OpenVPN

Para a ativação do serviço OpenVPN no cliente é necessário executar os comandos apresentados de seguida.

```
$ systemctl stop firewalld
$ cd /etc/openvpn
openvpn --config /etc/openvpn/client.conf
```

8 Apache Web Server

8.1 Configuração do Apache

De forma a garantir o correto funcionamento do Apache é necessário especificar no seu ficheiro de configuração o seu certificado e chave privada e ainda o certificado da CA. Para garantir a autenticação do cliente através do uso de certificados X.509 é necessária a inclusão desta informação no ficheiro [1] [4].

```
$ nano /etc/httpd/conf.d/ssl.conf

# Configuração do Apache para utilização do seu certificado, chave privada e
↪  certificado da CA
SSLCertificateFile /etc/pki/CA/apache.crt
SSLCertificateKeyFile /etc/pki/CA/apache.key
```

```
SSLCACertificateFile /etc/pki/CA/ca.crt

# Autenticação do cliente através de certificados X.509
SSLVerifyClient require
SSLVerifyDepth 10
```

Após a alteração do ficheiro de configuração do Apache, é necessário efetuar um *restart* ao serviço.

```
$ systemctl stop firewalld
$ systemctl restart httpd
```

8.2 Importação certificados no browser

Neste momento é necessário adicionar ao browser do cliente, o certificado da CA para que esta possa fazer parte da lista de Autoridades aceites pelo browser. Desta forma, o browser será capaz de reconhecer e validar certificados emitidos pela CA adicionada. Isto pode ser feito acedendo às *Definições*, *Ver certificados*, *Autoridades*, *Importar* e escolhendo o certificado da CA (*ca.crt*).

De seguida, adicionámos o certificado criado para o cliente para que o browser consiga fazer a sua autenticação. Para a importação do certificado do cliente, devemos aceder às *Definições*, *Ver certificados*, *Os seus certificados*, *Importar* e escolher o certificado do cliente (*user.p12*).

9 Autenticação

9.1 VPN Gateway

Começamos por permitir que o serviço OpenVPN seja executado noutro porto [5].

```
$ semanage port -m -t openvpn_port_t -p udp 53
```

De seguida, adicionamos a seguinte configuração de encaminhamento [5].

```
$ nano /etc/sysctl.conf
# Ativar forwarding de pacotes
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
$ sysctl -p
```

Instalamos *Google Authenticator*.

```
$ yum install google-authenticator*
```

Demos permissões de execução ao *google-authenticator* [5].

```
$ 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(/.*)?'
```

Script para gerar códigos do *Google Authenticator* [5].

```
$ nano create-gauth.sh
#!/bin/sh

# Parse arguments
USERNAME="$1"

if [ -z "$USERNAME" ]; then
```

```
  echo "Usage: $(basename $0) <username>"
  exit 2
fi

# Set the label the user will see when importing the token:
LABEL='OpenVPN Server'

su -c "google-authenticator -t -d -r3 -R30 -W -f -l \"${LABEL}\" -s
↪  /etc/openvpn/google-authenticator/${USERNAME}" - gauth

$ chmod 700 create-gauth.sh
```

9.1.1 Criação de novo utilizador

Criamos um utilizador, a sua *password* e geramos o *token* do *Google Authenticator* [5].

```
$ useradd -s /sbin/nologin user
$ passwd user
    1234
    1234
$ bash create-gauth.sh user | tee gauth_output_user.txt
$ cat gauth_output_user.txt
```

Depois, adicionamos à aplicação cliente do *Google Authenticator*, através do QR Code ou da chave secreta.

9.1.2 PAM

Criamos o ficheiro **/etc/pam.d/openvpn** e adicionamos esta configuração [5]:

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

#%PAM-1.0
auth [user_unknown=ignore success=ok ignore=ignore default=bad]
↪  pam_securetty.so
auth required /lib64/security/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
```

9.2 Apache Server

No servidor Apache a autenticação de 2 fatores foi implementada através da apresentação de um certificado X.509 válido e com autenticação básica (Nome de utilizador e Palavra passe) em vez de OTP [4].

Para definir o nome de utilizador e palavra passe do utilizador recorremos ao seguinte comando.

```
$ htpasswd -c /etc/httpd/conf.d/.htpasswd sti
    1234
```

De forma a garantir a autenticação dos clientes adicionámos as seguintes configurações ao ficheiro de configuração do Apache. Para aplicar as alterações efetuadas, basta dar *restart* ao serviço Apache.

```
$ nano /etc/httpd/conf.d/ssl.conf
<Location />
    AuthType basic
    AuthName "sti"
    AuthUserFile /etc/httpd/conf.d/.htpasswd
    Require valid-user
```

```
</Location>
```

Referências

[1] J. Granjal, "Segurança prática em sistemas e redes com linux," 2017.

[2] OpenVPN, "Openvpn/ocsp_check.sh at master · openvpn/openvpn," Feb 2016. [Online]. Available: https://github.com/OpenVPN/openvpn/blob/master/contrib/OCSP_check/OCSP_check.sh

[3] bjoernv, "Signer certificate for ocsp responder," Nov 2021. [Online]. Available: https://forums.openvpn.net/viewtopic.php?t=25307

[4] apache, "Apache module mod_auth_basic," 2021. [Online]. Available: https://httpd.apache.org/docs/current/mod/mod_auth_basic.html

[5] urs, "Setup an openvpn server with certificate and two-factor authentication on centos 7," Dec 2016. [Online]. Available: https://nethack.ch/2016/12/08/setup-an-openvpn-server-with-certificate-and-two-factor-authentication-on-centos-7/