UNIVERSIDADE D
COIMBRA

Relatório do Assignement 1

Segurança em Tecnologias da Informação

Bruno José Silvério da Silva 2021232021

Diogo Emanuel Matos Honório 2021232043

Introdução

O presente relatório foi realizado no âmbito da cadeira de Segurança em Tecnologias da Informação com objetivo de implementar um túnel VPN com autenticação de dois-fatores (2FA) e verificação de certificados.

Para alcançar estes objetivos, foram utilizados os serviços OpenVPN e Apache, além do protocolo Online Certificate Status Protocol (OCSP) e One Time Password (OTP) que ira ser descrito no decorrer deste relatório.

1. Arquitetura

A arquitetura deste projeto consiste em duas máquinas virtuais, com o sistema operativo CentOS 9, onde:

- Uma máquina atuou como servidor, responsável por hospedar o OpenVPN, o OCSP e o sistema de autenticação OTP.
- A outra máquina funcionou como cliente, configurada para estabelecer uma conexão segura com o servidor OpenVPN.

A comunicação entre essas máquinas foi viabilizada por uma rede NAT com IPs atribuídos manualmente, conforme o enunciado do projeto.

2. Certificados

2.1 Preparação da estrutura das CAs

Para a configuração dos certificados, foi necessária a criação de uma estrutura organizada de diretorias para armazenar os diferentes tipos de certificados, tanto no servidor como no cliente:

- /certs Armazena os certificados criados.
- /crl Contém a lista de certificados revogados.
- /newcerts Utilizado para novos certificados.
- /private Guarda os certificados privados.

```
sudo mkdir -p /etc/pki/CA/{certs,crl,newcerts,private}
```

Após a criação dessas diretorias, procedemos com a inicialização da Autoridade Certificadora (CA) privada usando OpenSSL. Para garantir o correto funcionamento, foram criados os seguintes ficheiros:

```
sudo touch /etc/pki/CA/index.txt
echo 01 | sudo tee /etc/pki/CA/serial
```

O index.txt regista os certificados emitidos pela CA e o serial define o número de série inicial dos certificados.

2.2 Criação da CA privada

A CA privada foi criada com o OpenSSL, seguindo as boas práticas de segurança e definição de políticas de certificação adequadas ao ambiente de VPN.

```
openssl genrsa -out ca.key –des3
openssl req -new -key ca.key -out ca.csr
openssl x509 -req -days 365 -in ca.csr -out ca.crt -signkey ca.key -extfile v3_ca.ext
```

2.3 Criação do certificado para o Servidor OpenVPN

Foi gerado um certificado específico para o servidor OpenVPN, permitindo a comunicação segura com os clientes, como também o arquivo dh2048.pem.

```
openssl genrsa -out server.key –des3
openssl req -new -key server.key -out server.csr
openssl ca -in server.csr -cert ca.crt -keyfile ca.key -out server.crt -extfile v3_server.ext
```

```
openssl dhparam -out dh2048.pem 2048
```

2.4 Criação do certificado para o Cliente OpenVPN

Para o cliente OpenVPN exportamos a CA criada na máquina do servido através do *scp* para a criação de um certificado individual, garantindo uma identificação exclusiva e segura na rede.

```
openssl genrsa -out client.key –des3
openssl req -new -key client.key -out client.csr
openssl ca -in client.csr -cert ca.crt -keyfile ca.key -out client.crt -extfile v3_client.ext
```

Assim, podemos importar e visualizar o nosso certificado no browser com o ficheiro cliente.p12 após a criação do mesmo através do seguinte comando.

```
openssl pkcs12 -export -clcerts -in user.crt -inkey client.key -out client.p12
```

3. OpenVPN

Em relação ao OpenVPN, as seguintes configurações foram implementadas para permitir a conexão entre o cliente e o servidor:

```
client
dev tun
proto udp
remote 192.168.0.53 1194
persist-tun
persist-key
ca /etc/openvpn/ca.crt
cert /etc/openvpn/client.crt
key /etc/openvpn/client.key
tls-auth ta.key 1
remote-cert-tls server                                    // client.conf
```

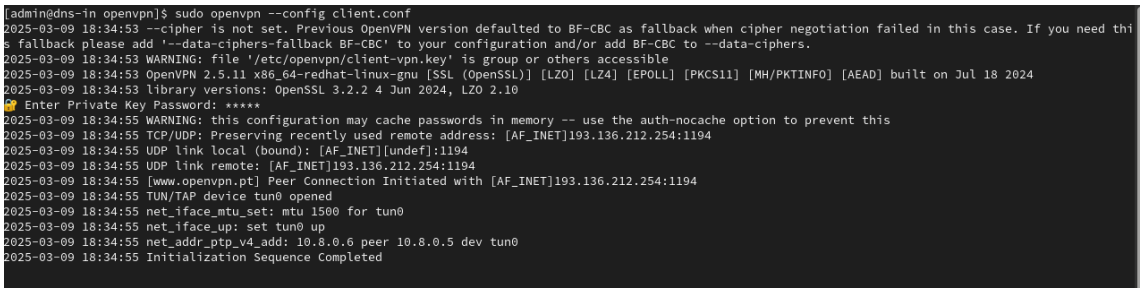
```
local 192.168.0.53
port 1194
proto udp
dev tun
ca /etc/openvpn/ca.crt
cert /etc/openvpn/server.crt
key /etc/openvpn/server.key
dh /etc/openvpn/dh2048.pem
server 10.8.0.0 255.255.255.0
tls-auth ta.key 0                                         // server.conf
```

Para executar OpenVPN executamos os seguintes comandos:

Server:

```
[admin@localhost openvpn]$ sudo openvpn --config server.conf
2025-03-09 18:34:38 WARNING: --topology net30 support for server configs with IPv4 pools will be removed in a future release. Please migrate to --topology sub
net as soon as possible.
2025-03-09 18:34:38 --cipher is not set. Previous OpenVPN version defaulted to BF-CBC as fallback when cipher negotiation failed in this case. If you need thi
s fallback please add '--data-ciphers-fallback BF-CBC' to your configuration and/or add BF-CBC to --data-ciphers.
2025-03-09 18:34:38 OpenVPN 2.5.11 x86_64-redhat-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Jul 18 2024
2025-03-09 18:34:38 library versions: OpenSSL 3.2.2 4 Jun 2024, LZO 2.10
2025-03-09 18:34:38 WARNING: --keepalive option is missing from server config
🔐 Enter Private Key Password: *****
2025-03-09 18:34:40 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
2025-03-09 18:34:40 TUN/TAP device tun0 opened
2025-03-09 18:34:40 net_iface_mtu_set: mtu 1500 for tun0
2025-03-09 18:34:40 net_iface_up: set tun0 up
2025-03-09 18:34:40 net_addr_ptp_v4_add: 10.8.0.1 peer 10.8.0.2 dev tun0
2025-03-09 18:34:40 Could not determine IPv4/IPv6 protocol. Using AF_INET
2025-03-09 18:34:40 UDPv4 link local (bound): [AF_INET]193.136.212.254:1194
2025-03-09 18:34:40 UDPv4 link remote: [AF_UNSPEC]
2025-03-09 18:34:40 Initialization Sequence Completed
2025-03-09 18:34:55 193.136.212.1:1194 peer info: IV_VER=2.5.11
2025-03-09 18:34:55 193.136.212.1:1194 peer info: IV_PLAT=linux
2025-03-09 18:34:55 193.136.212.1:1194 peer info: IV_PROTO=6
2025-03-09 18:34:55 193.136.212.1:1194 peer info: IV_NCP=2
2025-03-09 18:34:55 193.136.212.1:1194 peer info: IV_CIPHERS=AES-256-GCM:AES-128-GCM
2025-03-09 18:34:55 193.136.212.1:1194 peer info: IV_LZ4=1
2025-03-09 18:34:55 193.136.212.1:1194 peer info: IV_LZ4v2=1
2025-03-09 18:34:55 193.136.212.1:1194 peer info: IV_LZO=1
2025-03-09 18:34:55 193.136.212.1:1194 peer info: IV_COMP_STUB=1
2025-03-09 18:34:55 193.136.212.1:1194 peer info: IV_COMP_STUBv2=1
2025-03-09 18:34:55 193.136.212.1:1194 peer info: IV_TCPNL=1
2025-03-09 18:34:55 193.136.212.1:1194 [client1-vpn] Peer Connection Initiated with [AF_INET]193.136.212.1:1194
2025-03-09 18:34:55 client1-vpn/193.136.212.1:1194 MULTI_sva: pool returned IPv4=10.8.0.6, IPv6=(Not enabled)
```

Client:



4. Apache

Primeiramente devemos criar certificado para o servidor Apache e para o cliente da mesma forma que foi feito para o OpenVPN.

Após isso a configuração do apache para futuro uso com o OSCP e OTP é necessário instalá-lo através dos seguintes comandos.

```
sudo yum install http mod_ssl -y
sudo systemctl enable httpd
sudo systemctl start httpd
```

De seguida, foram alteradas as seguintes linhas do ficheiro ssl.conf.

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

```
SSLCertificateFile /etc/openvpn/apache_server.crt
SSLCertificateKeyFile /etc/openvpn/apache_server.key
SSLCACertificateFile /etc/openvpn/ca.crt
SSLVerifyClient require
SSLVerifyDepth 10                                    // ssl.conf
```

Para implementar as alterações do ssl.conf devemos reiniciar o servidor apache.

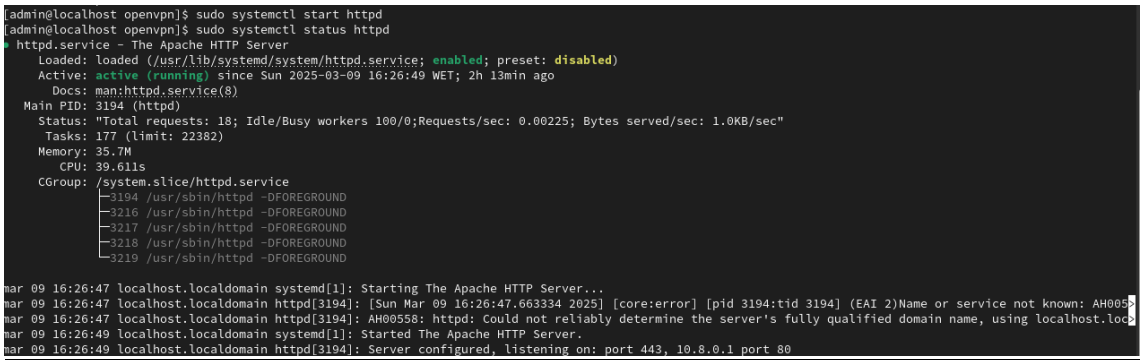
```
sudo systemctl restart httpd
```

Para o funcionamento em conjunto com o OpenVPN, é necessário fazer alguns ajustes no ficheiro de configuração do Apache.

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

```
listen 10.8.0.1:80                                   // httpd.conf
```

Server:



Para testar se temos acesso ao apache via VPN realizamos o seguinte comando:

```
sudo curl -v cert apache_client.crt --key apache_client.key http://10.8.0.1
```

Posteriormente será possível ver o conteúdo da página caso esteja tudo bem configurado.

5. OCSP

O OCSP foi configurado para permitir a verificação em tempo real dos certificados trocados entre cliente e servidor. Para testar sua funcionalidade, foi criado um certificado revogado e submetido a uma verificação, assegurando que conexões com certificados comprometidos sejam corretamente bloqueadas.

Essa implementação melhora a segurança das comunicações dentro da infraestrutura VPN, garantindo que apenas clientes certificados válidos possam estabelecer conexões seguras.

```
openssl ca -revoke revoked_client.crt
```

Em caso de erro devemos alterar as seguintes linhas do openssl.cnf:

```
nano /etc/ssl/openssl.cnf
```

```
certificate       = /etc/openvpn/ca.crt
private_key       = /etc/openvpn/ca.key                    // openssl.cnf
```

Para a configuração e a ativação do Server OCSP devemos realizar as seguintes mudanças no openssl.cnf, para que posteriormente os certificados possuam a informação OCSP.

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

```
certificate       = /etc/openvpn/ca.crt
private_key       = /etc/openvpn/ca.key                    // openssl.cnf
```

Após isto procede-se a ativação do Servidor OCSP com o recurso ao OpenSSL, seguindo as seguintes configurações o serviço irá responder no porto 81, fazendo uso da informação relativa à CA.

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

6. OTP

6.1 Configuração do servidor

A implementação e configuração do OTP (One-Time Password) ocorre após a instalação do google-authenticator. Este processo envolve as seguintes etapas:

```
sudo apt install google-authenticator
```

Após a instalação do google-authenticator, prosseguimos à configuração do serviço. A execução do comando necessário gera um QR code e uma chave secreta.

```
google-authenticator
```

Ao efetuar a leitura do QR code gerado utilizando o aplicativo Google Authenticator, disponível para dispositivos móveis, é possível gerar uma senha temporária baseada no tempo (TOTP – Time-based One-Time Password). Esta senha é dinâmica, ou seja, muda periodicamente, o que aumenta a segurança do processo de autenticação.

Além disso, é necessário configurar o PAM (Pluggable Authentication Modules) para integração com o OpenVPN.

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

```
auth required pam_google_authenticator.so                    // openvpn
```

Com isto, podemos configurar o OpenVPN para utilizar as senhas temporárias. Para validar a palavra-passe enviada pelo cliente e verificar se corresponde à gerada pelo OTP, é necessário criar um script chamado authenticate.sh, que será responsável por essa verificação. O *authenticate.sh* pode ser estruturado da seguinte forma:

```
#!/bin/bash

PAM_SERVICE="openvpn"
USERNAME=$username
PASSWORD=$password

# Verifica se o usuário existe no sistema
if id "$USERNAME" &>/dev/null; then
  echo "Usuário $USERNAME encontrado no sistema."
else
  echo "Erro: Usuário $USERNAME não existe!"
  exit 1
fi

# Valida a senha e o OTP
echo "$PASSWORD" | /usr/bin/pam_exec -a -- /bin/login -f "$USERNAME" > /dev/null 2>&1
EXIT_CODE=$?

if [ $EXIT_CODE -eq 0 ]; then
  # Se a senha for válida, verifica o OTP
  echo "$PASSWORD" | google-authenticator -s
/home/$USERNAME/.google_authenticator -t -u
  if [ $? -eq 0 ]; then
    exit 0
  else
    echo "Erro: OTP inválido!"
    exit 1
  fi
else
  echo "Erro: Senha inválida!"
  exit 1
fi                                                    // authenticate.sh
```

Este script recebe como parâmetros o nome do utilizador e a palavra-passe fornecida, verifica a existência do arquivo de configuração do OTP e, em seguida, valida se o código fornecido corresponde ao gerado pelo Google Authenticator.

Para que o cliente consiga conectar-se ao servidor, é necessário configurar o ficheiro de configuração do OpenVPN. Para isso, deve-se adicionar a seguinte linha ao arquivo de configuração do servidor OpenVPN:

```
sudo nano /etc/openvpn/server.conf
```

```
plugin /usr/lib64/openvpn/plugins/openvpn-plugin-auth-pam.so openvpn
auth-user-pass-verify /etc/openvpn/authenticate.sh via-env
script-security 2                                          // openvpn
```

6.2 Configuração do cliente

Após configurar o servidor OpenVPN para utilizar autenticação baseada em OTP, é necessário configurar o cliente para fornecer as credenciais corretamente durante a autenticação.

```
sudo nano /etc/openvpn/client.conf
```

```
auth-user-pass                                             // client.conf
```

Para autenticação, o utilizador precisa inserir as credenciais corretamente. Como a autenticação agora envolve OTP, a senha que será pedida no cliente OpenVPN deve ser composta por: [senha do usuário][código OTP].

6.3 Problemas na Autenticação do Cliente

Durante os testes de autenticação do cliente OpenVPN com OTP, não foi possível estabelecer a conexão devido a um erro na validação da palavra-passe. O cliente foi configurado corretamente para enviar as credenciais conforme esperado pelo servidor. No entanto, ao tentar estabelecer a conexão, o servidor rejeitou a autenticação.

Conclusão

A implementação do túnel VPN com 2FA e verificação de certificados reforçou a segurança da comunicação entre cliente e servidor. O uso do OpenVPN, OCSP e OTP garantiu autenticação robusta, apesar de desafios na validação das credenciais. No geral, o projeto demonstrou a viabilidade da solução, com possibilidade de futuras melhorias na autenticação.