IPSec/VPN

Pedro Fernando Moreira Silva Antunes
Departamento de Ciências de Computadores
Faculdade de Ciências da Universidade do Porto
Porto, Portugal
up201507254fc.up.pt

André Ferreira Monteiro Lopes Rodrigues
Departamento de Ciências de Computadores
Faculdade de Ciências da Universidade do Porto
Porto, Portugal
up201505639@fc.up.pt

I. Introdução

Neste trabalho vamos abordar o conceito de Redes Privadas Virtuais, VPN, que utilizam o protocolo IPSec como forma de assegurar a segurança da comunicação. Começaremos o artigo explorando o conceito de VPN, sua arquitetura e protocolos, bem como as diversas vantagens desta tecnologia. Depois iremos abordar o protocolo IPSec, bem como outros protocolos que em conjunto com o IPSec permitem a criação de uma VPN segura. Depois iremos abordar o tema da performance da VPN usando IPsec/IKEv2 e por fim a conclusão que podemos tirar sobre a performance de uma VPN usando os protocolos IPSec/IKEv2.

Este trabalho está inserido na unidade curricular de Segurança em Engenharia de Software do Mestrado de Segurança Informática da Faculdade de Ciências da Universidade do Porto. Vamos abordar conceitos de Segurança em Engenharia de Software, como forma de assegurar a segurança do sistema, dos intervenientes do sistema e da comunicação entre os intervenientes e o sistema. O nosso objetivo foi perceber de que forma é que se constrói o design, o planeamento e a implementação de software destacando a segurança como primitiva para cada etapa. Para isso iremos construir um software numa arquitetura cliente-servidor e os clientes podem comunicar com os outros clientes do servidor numa arquitetura cliente-cliente. Os clientes têm de efetuar um pré registo presencial onde são validados os dados nacionais do utilizador (como o nome, identificação nacional, etc). Posto isto, os clientes podem então registar uma conta de utilizador e autenticar-se no servidor. O servidor fornece serviços de acordo com o nível de autorização de cada utilizador, ou seja, não permite a utilização de alguns serviços aos utilizador que não estão autorizados para a utilização de serviços com elevadas permissões de acesso. O serviço principal é a comunicação cliente-cliente para realizar trocas de mensagens. Primeiramente através do servidor, os clientes podem pedir uma lista de clientes, escolhem um cliente da lista e iniciam uma comunicação cliente-cliente com o cliente escolhido. Começaremos o artigo com a apresentação da ideia do software, bem como os requisitos de segurança. Depois, iremos apresentar o design e a arquitetura do sistema. De seguida, relatamos como é que o software será implementado e por fim descrevemos a fase de testagem da implementação, a análise ao trabalho realizado e aos resultados obtidos.

II. Virtual Private Networks (VPN)

Virtual Private Network ou Rede Privada Virtual é uma rede privada composta por duas ou mais redes privadas distintas geograficamente entre si, ou seja, não partilham uma ligação física entre si, portanto utilizam uma rede de comunicação pública, como a Internet, para se conectarem de forma virtual. [2] As VPN surgiram quando a Internet ainda não usava protocolos de criptografia, como o TLS, portanto um atacante que dominasse um router intermediário em uma comunicação, poderia receber e visualizar toda a informação que estaria a ser trocada entre dois pontos (ataque conhecido como man-in-the-middle). Este motivo levou à criação de VPNs por parte das forças armadas, onde era criada uma conexão segura e criptografada, um túnel, entre duas redes privadas. Contudo o que levou a popularização desta tecnologia foi a vontade que as empresas tinham de conectar diferentes redes privadas para poderem aceder a recursos que estavam em cada uma das redes. Antes das VPNs, era necessário pagar por um aluguer exclusivo ou não de uma linha física que conectava de forma física ambas as redes privadas. Só para esclarecimento, uma rede privada é uma rede onde endereços IP são normalmente privados, ou seja, os dispositivos na rede não tem a necessidade de possuir um endereço público pois a sua comunicação será feita com os dispositivos na rede privada, até mesmo para aceder à Internet, onde o dispositivo fala com o gateway e este trata de fazer o pedido e depois encaminhar para o endereço privado. Isto permite a reutilização dos endereços noutras redes privadas bem como aumenta a segurança pois não é possível, de forma geral, o mundo externo estabelecer uma conexão com uma máquina que use um IP privado. Posto isto, existem várias formas de ligar redes privadas, formas essas que caracterizam os tipos de VPNs existentes. Existe o tipo de VPNs no qual designamos por acesso remoto, no qual permite um utilizador que está numa rede privada conectar-se a uma outra rede privada que não seja através de uma forma física, daí o nome acesso remoto, de forma a poder aceder aos recursos e serviços presentes nessa outra rede. O outro tipo de VPN é designado por site-site ou router-router pois a VPN precisa de ser configurado no router de forma a permitir que ambas as redes privadas possam aceder aos recursos e serviços de cada uma, diferentemente do acesso remoto onde a ligação é unidireccional.

Quando ambas as redes privadas são geridas pela mesma

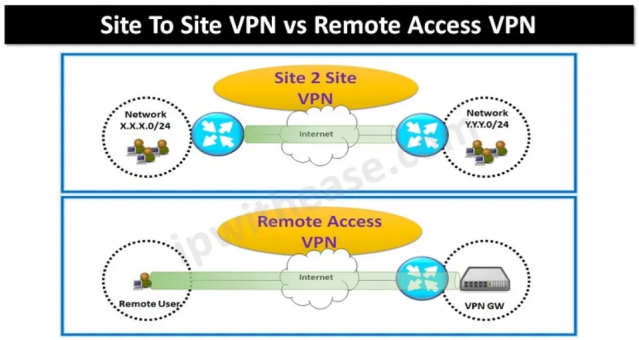

Fig. 1. Tipos de VPN

organização chamamos de Intranet e quando é uma ligação onde cada rede privada acede pertence a organizações diferentes chamamos de Extranet. As VPNs vieram trazer mais segurança para o mundo da internet, que é conhecida na sua forma natural por ser extremamente insegura devido a quando a sua criação não terem pensado na sua segurança. O ponto de partida são as firewalls que embora não estejam relacionadas diretamente com o uso ou aplicação de VPNs, as firewalls são uma parte integral pois elas tem a função de permitir utilizadores da VPN de entrarem na rede privada, ou seja dentro do perímetro de segurança, mantendo todos os outros visitantes indesejados do lado de fora. O próximo ponto é a autenticação, como vimos a forma de entrar dentro do perímetro de segurança é sendo um utilizador da VPN portanto é necessário assegurar a autenticação deste utilizador e portanto o padrão hoje em dia é o uso de algoritmos criptográficos como o MAC que assegura a integridade da comunicação, o RSA e o DH que permitem a troca de chaves e asseguram a integridade de cada interveniente na comunicação e o uso de chaves simétricas, como o AES. O que nos leva ao próximo ponto que é a encriptação da informação, como referido antes de existir o TLS e mesmo após a criação do TLS ainda existem muitas possibilidades de haver ataques de sniffing nas redes, daí é um dos maiores pilares que levou a criação desta tecnologia é a possibilidade de poder-se realizar comunicações seguras onde a integridade, autenticidade e confidencialidade é mantida mesmo usando redes públicas como intermediários.

A. Protocolos VPN

As conexões lógicas podem ser feitas na camada 2 ou 3 da pilha OSI, ou seja, nas camadas de Dados ou de Rede, respectivamente. Exemplos da camada 2 são ATMs e Frame Relays, que são tecnologias implementadas por grandes organizações ou ISPs onde são configurados circuitos virtuais através de toda a rede para conectar diversas redes privadas. A vantagem deste sistema é que oferece a melhor performance, contudo também é o mais dispendioso. O foco do nosso artigo será na camada 3, camada de rede, pois é a mais utilizada e onde existem diversos protocolos no qual passaremos a enumerar.

- LT2P/IPSec
  Layer 2 Tunnel Protocol é uma extensão do PPTP, e foi desenvolvido com a intenção de o substituir. A versão original não proporciona nenhum tipo de criptografia e portanto é muitas vezes utilizado em conjunto com o protocolo IPSec que garante segurança a nível da rede (IP). O LT2P opera com duas encapsulações, primeiro é responsável pela criação das conexões da VPN, utilizando o PPP (Point-to-Point protocol) similar ao PPTP. E na segunda fase utiliza o protocolo IPSec que é responsável pela integridade e confidencialidade da informação. Apesar de seguro, o próprio protocolo pode ser fraco, bastante complexo e lento devido às duas encapsulações. Como utiliza o protocolo IPSec que por consequência utiliza o protocolo de transporte UDP faz com que exista uma probabilidade alta de ser bloqueado por firewalls.
- OpenVPN
  É um protocolo open source, o que faz com que o seu código seja acessível e já foi visto por diversas pessoas, o que garante uma segurança maior. Este protocolo utiliza o protocolo TLS para garantir a integridade e confidencialidade da comunicação. É altamente configurável, onde podemos definir a porta, bem como entre os protocolos UDP ou TCP, o que em conjunto com o TLS torna este protocolo indistinguível de uma comunicação HTTPs o que permite circundar as firewalls. Para implementar um serviço recorrendo a este protocolo é necessário uma aplicação third-party porque este protocolo não é suportado de forma nativa por nenhuma distribuição.

- IKEv2/IPSec
  Internet Key Exchange version 2 é um protocolo desenvolvido pela Cisco e Microsoft, que não foi desenvolvido especificamente para criar VPNs, pois a sua funcionalidade é proporcionar autenticidade dos intervenientes na comunicação e a troca de chaves e parâmetros criptográficos de forma segura, similar ao estabelecimento de uma sessão no protocolo TLS. Este protocolo será depois utilizado pelo IPSec como forma de garantir a integridade e confidencialidade da comunicação. A junção dos dois protocolos garante um túnel seguro entre duas redes privadas.

III. IPSec

O protocolo IPSec é a resposta à combinação insegura TCP/IP, onde fornece autenticação e encriptação a nível da rede, ou seja, toda a estrutura construída acima estaria também segura. Desenvolvido pelo IETF, com o intuito de proporcionar vários serviços de segurança para o protocolo IP, tanto IPv4 como IPv6. O IPSec foi desenhado para proporcionar uma rede segura, não pondo a camada aplicacional em risco, basicamente proporciona compatibilidade para todas as aplicações e garante a segurança de toda a internet pois não dependente da camada aplicacional, pois corre no kernel e portanto não depende que aplicações usem por exemplo TLS para proporcionar segurança. O protocolo foca-se em três vertentes: algoritmos de encriptação, algoritmos de autenticação e nas chaves. O IPSec não especifica nenhum desses pontos, o IPSec pode ser visto como uma framework que utiliza vários protocolos para cada um desses pontos pois devido a esses algoritmos serem efêmeros no mundo da segurança informática, como o IPSec não quer comprometer a compatibilidade e a segurança ele delega essa informação como parte da sua configuração, ou seja, quem efetua a configuração pode escolher que protocolos quer usar, apenas uma ressalva para os protocolos que lidam com as chaves pois têm de ser da família IKE. Para além disso o IKE tem dois modos de operação, no modo de transporte apenas o payload dos pacotes IP é autenticado e/ou cifrado. Os cabeçalhos IP não são modificados. Este modo é normalmente usado para proteger uma transmissão ponto-a-ponto. Já no modo túnel o datagrama IP é completamente autenticado e/ou cifrado pois este é encapsulado num novo datagrama IP com um novo cabeçalho IP. Este modo é o indicado para ser utilizado para formar uma VPN, criando um túnel seguro que pode ser site-to-site (entre dois routers gateway), host-to-site (acesso remoto de um host a um router gateway), ou host-to-host. [3] O IPSec utiliza o protocolo IKEv2 para o estabelecimento de SAs. Associação de Segurança (SA) é uma conexão lógica entre entidades de uma comunicação, onde são estabelecidos os algoritmos e parâmetros de segurança acordados entre ambas as partes. É uma ligação unidireccional, onde cada entidade tem de estabelecer uma SA. A finalidade deste protocolo é fazer com que ambas as partes partilhem uma chave de forma segura e automática, através de chaves pre partilhadas ou de certificados. O IPSec divide-se em duas fases, na fase inicial é estabelecido um túnel seguro, utilizando o protocolo ISAKMP que é um protocolo que faz parte do protocolo IKE. Assim ambas as entidades negociam os algoritmos de cifra, de verificação de integridade, os métodos de autenticação, o grupo Diffie-Hellman e o tempo de vida do túnel. No final desta fase, ambos terão uma chave compartilhada e terão diversos métodos de segurança configurados para serem usados durante a comunicação. A segunda fase é onde será estabelecido o túnel da comunicação onde será transportada a informação propriamente dita. Como referido o IPSec é responsável pela autenticação e encriptação de toda a informação do pacote desde o nível aplicacional até ao nível da rede, excepto alguns parâmetros do pacote IPSec pois precisam de ser vistos e até mesmo alterados por alguns nós intermediários da rede. O IPSec faz isso utilizando uma função de apenas Autenticação, Authentication Header (AH), que só existe para oferecer compatibilidade, mas que não deverá ser usada pois não fornece encriptação. Portanto a função que deve ser usado chama-se Encapsulating Security Payload (ESP) e é responsável como dito por encriptar toda a informação desde o cabeçalho TCP até ao cabeçalho aplicacional, e oferece também autorização desde o cabeçalho ESP até ao cabeçalho aplicacional. ESP suporta vários algoritmos de cifra como AES, DES, entre outros, estes algoritmos são definidos por cada SA. No caso de uma VPN que utiliza o IPSec no modo túnel, final obtemos um datagrama constituído por um cabeçalho IP mais o resto do payload, ambos cifrado utilizando a função ESP que cifrou e autenticou o resto do payload com os padrões definidos pelos SA durante a fase do IKE (1ºfase). Sobre esse datagrama é também adicionado um cabeçalho ESP com os campos SPI, que identifica a que SA o datagrama pertence, o número de sequência (usado para prevenir replay attacks). Depois, é criado um hash de autenticação sobre todo o pacote resultante, e esse valor é adicionado ao fim do pacote (ESP auth). No fim, é adicionado um novo endereço IP, o número do protocolo do novo cabeçalho IP passa a ser 50 que identifica o protocolo ESP. [4]

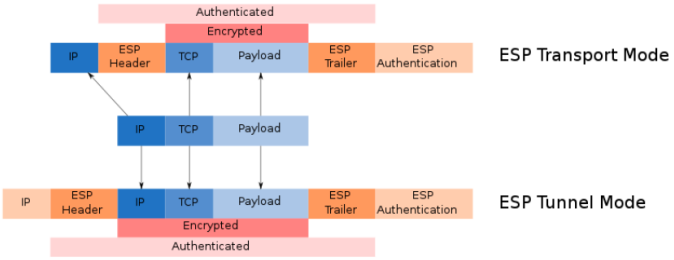

Fig. 2. Modo túnel com ESP

IV. Objetivo

Implementação, teste, e análise de uma VPN implementada através dos protocolos IKEv2/IPSec. [1] Vamos implementar uma VPN host-to-host / host-to-site, tal como uma VPN comercial, onde nos conseguimos conectar a uma outra rede privada, o que nos permite aceder a serviços na internet como

se estivéssemos nessa rede privada e também conseguimos aceder aos recursos e serviços dessa rede, por exemplo o serviço de SSH no próprio servidor VPN. As VPNs seguem o modelo cliente-servidor e portanto precisamos de configurar um servidor na rede privada pretendida e um outro cliente numa outra rede privada distinta e de nenhuma forma conectar fisicamente. A ideia é que o cliente e o servidor estabeleçam uma conexão segura através do IPSec e depois os pacotes do cliente são redireccionados do servidor para o gateway da rede privada do servidor, basicamente o servidor atua como um proxy e faz com que o cliente esteja de forma virtual na rede. Para tal utilizamos a ferramenta strongswan que nos proporciona uma implementação IPSec, ou seja, podemos usar a framework IPSec que já engloba protocolos IKE para criar túneis seguros entre entidades.

A. Implementação do Servidor

1) Instalar Strongswan e suas dependencias: No caso, o nosso servidor esta a correr a distribuição Ubuntu versão 20.04, portanto para instalar o strongswan:

```
sudo apt install strongswan strongswan-pki libcharon
    -extra-plugins libcharon-extauth-plugins
    libstrongswan-extra-plugins
```

Depois precisamos de criar um certificado de autoridade porque o IKEv2 requer um certificado para que os clientes possam identificar o servidor. Depois precisamos de gerar o certificado do servidor, que será assinado pelo certificado de autoridade. Neste exemplo, o próprio servidor assina ambos os certificados, o que é algo inseguro, num contexto profissional o certificado do nosso servidor deveria ser assinado por uma autoridade credenciada para tal.

```
mkdir -p ~/pki/{cacerts,certs,private}
chmod 700 ~/pki
pki --gen --type rsa --size 4096 --outform pem > ~/
    pki/private/ca-key.pem
pki --self --ca --lifetime 3650 --in ~/pki/private/
    ca-key.pem --type rsa --dn  CN =VPN root  C A
    --outform pem > ~/pki/cacerts/ca-cert.pem
pki --gen --type rsa --size 4096 --outform pem > ~/
    pki/private/server-key.pm
```

No nosso caso o servidor não estava instalado no router/gateway, estava por detrás da NAT e tinha IP 192.168.1.78. O nosso IP público era 176.79.17.216. No nosso caso não implementamos Dynamic DNS (DDNS) para atualizar o DNS quando o nosso IP público alterar, portanto desta forma se o IP público alterar é necessário gerar um novo certificado e configurar-lo também no cliente, bem como alterar as configurações que vamos ver mais a frente.

```
pki --pub --in ~/pki/private/server-key.pem --type
    rsa | pki --issue --lifetime 1825 --cacert ~/pki
    /private/ca-key.pem --cakey ~/pki/private/ca-key
    .pem --dn  CN =<ip_vpn>    --san @<ip_vpn> --
    san <ip_vpn> --flag serverAuth --flag
    ikeIntermediate --outform pem > /pki/certs/
    server-cert.pem
```

Depois vamos copiar os nossos certificados para o directório que o IPSec e por consequente o Strongswan espera encontrar os certificados.

```
sudo cp -r ~/pki/* /etc/ipsec.d/
```

3) Configurar o Strongswan do lado do servidor: Agora vamos configurar o Strongswan, ou seja, o nosso servidor VPN para criar túneis IPSec.

```
sudo nano /etc/ipsec.conf
```

```
version 2.0

config setup
    charondebug="ike 1, knl 1, cfg 0"
    uniqueids=no

conn ikev2-vpn
    auto=add
    compress=no
    type=tunnel
    keyexchange=ikev2
    fragmentation=yes
    forceencaps=yes
    dpdaction=clear
    dpddelay=300s
    rekey=no
    left=%any
    leftid=                    <- IP Publico
    leftcert=server-cert.pem
    leftsendcert=always
    leftsubnet=192.168.1.0/24
    right=%any
    rightid=%any
    rightauth=eap-mschapv2
    rightsourceip=192.168.1.0/24
    rightdns=8.8.8.8,8.8.4.4
    rightsendcert=never
    eap_identity=%identity
    ike=aes256-sha512-modp8192
    esp=aes256-sha512
```

Fig. 3. Configuração do Servidor

Como podemos observar pela imagem, nós vamos dizer ao Strongswan para criar uma conexão chamada “ikev2-vpn” onde utilizará o IPSec no modo túnel, onde o método de troca de chaves será o IKEv2. Adicionamos uns campos que começam por “dpd” que significa dead-peer detection e serão utilizados caso um cliente se desconecte. Depois, configuramos o lado IPSec do servidor usando a tag left, %any garante que o servidor irá usar a interface de rede que recebe comunicações dos clientes, leftid é referente ao nome que o servidor mostra para os clientes, que combinado

com o leftcert faz com que o nome e o Distinguished Name(DN) tenham que coincidir. Depois temos o caminho para o certificado do servidor que será usado para autenticar o servidor para os clientes e assim poderem negociar a chave através de IKEv2. Depois iremos configurar o lado do cliente usando a tag right, onde %any neste caso diz ao servidor para aceitar conexões vindas de qualquer cliente remoto. A tag "eap-mschapv2" refere-se ao método de autenticação que será usado pelos clientes para se autenticar com o servidor. Depois definimos o conjunto de IPs privados que serão atribuídos aos clientes remotos, no caso iremos utilizar novamente a subnet 192.168.1.0/24 para que simule um cliente local na rede privada do servidor. Escolhemos também os DNSs que será usado pelo cliente. eap_identity=%identity diz ao servidor para perguntar pelas credenciais do cliente e as tags "ike" e "esp" irão definir os algoritmos que o cliente poderá escolher para depois serem usados tanto pelo IKE na troca de chaves e pela função ESP do IPSec.

4) Configurar as credenciais dos clientes: A próxima etapa consiste em registar um utilizador no servidor. Primeiro precisamos de dizer ao Strongswan onde ele pode encontrar a chave do nosso servidor e depois é que podemos definir as credenciais dos utilizadores.

```
sudo nano /etc/ipsec.secrets
    : RSA "server-key. p e m
    pedro : EAP    tar
    andre : EAP    tar
sudo systemctl restart strongswan-starter
```

5) Configurar a firewall: Para que os pacotes do cliente possam chegar ao seu destino passando antes pelo servidor VPN é preciso fazer umas alterações na firewall do servidor e no caso também no próprio router/gateway da rede do servidor pois como referido o servidor encontra-se por detrás da NAT e portanto o cliente tem de comunicar primeiro com o router/gateway que depois sim irá direcionar o tráfego para o servidor VPN. Primeiro precisamos de abrir as portas UDP que são usadas pelo protocolo IPSec: "$ sudo ufw allow 500,4500/udp". Depois precisamos de saber qual é a interface que o servidor utiliza para comunicar com o gateway e por tanto usamos o comando: "$ ip route show default" que nos mostrará que o servidor comunica com o gateway 192.168.1.254 através da interface de rede enp3s0. Portanto o próximo passo consiste em adicionar regras à firewall do servidor para que ele possa encaminhar os pacotes recebidos pelos clientes para o gateway e para que o gateway utilize o seu ip público para enviar os pacotes até ao servidor público na internet, tudo isso será feito sobre a tabela nat (*nat) enquanto que a regra *mangle apenas serve para evitar potenciais problemas limitando o tamanho dos pacotes TCP. É preciso também dizer a firewall para encaminhar tráfego ESP para que os clientes se possam conectar ao servidor pois utilizam túneis IPSec.

```
*nat
-A POSTROUTING -s 192.168.1.0/24 -o enp3s0 -m policy
     --pol ipsec --dir out -j ACCEPT
-A POSTROUTING -s 192.168.1.0/24 -o enp3s0 -j
    MASQUERADE
COMMIT

*mangle
-A FORWARD --match policy --pol ipsec --dir in -s
    192.168.1.0/24 -o enp3s0 -p tcp -m tcp --tcp-
    flags SYN,RST SYN -m tcpmss --mss 1361:1536 -j
    TCPMSS --set-mss 1360
COMMIT

-A ufw-before-forward --match policy --pol --ipsec
    --dir in --proto esp -s 192.168.1.0/24 -j ACCEPT
-A ufw-before-forward --match policy --pol ipsec --
    dir out --proto esp -d 192.168.1.0/24 -j ACCEPT
```

Depois é necessário permitir o encaminhamento de pacotes ip e para tal fazemos:

```
sudo nano /etc/ufw/sysctl.conf
    net/ipv4/ip_forward=1
```

Normalmente também é bloqueado o envio e recepção de pacotes ICMP para evitar ataques man-in-the-middle, mas neste caso optamos por não ativar essa opção porque queremos usar os serviços disponíveis na rede privada. Como por exemplo comunicar com o servidor através de ssh. Contudo vamos desabilitar o "Path MTU": net/ipv4/ip_no_pmtu_disc=1. Depois é só fazer: "$ sudo ufw reload" para que as regras da firewall entrem em vigor.

As definições no router podem ser vistas na imagem abaixo. Basicamente adicionamos regras para encaminhar os pedidos da porta externa 4500 para a porta 4500 do servidor (porta utilizada pelo IPSec).

Fig. 4. Port Forwarding no router

B. Implementação do Cliente

Neste exemplo criamos um cliente numa máquina com Ubuntu 20.04. Para criar o cliente foi necessário instalar o strongswan também no cliente: "$ sudo apt install strongswan libcharon-extra-plugins". Depois será necessário ter o certificado de autoridade que assinou o certificado do servidor na nossa máquina e copiá-lo para o diretório /etc/ipsec.d/cacerts. Depois temos de configurar as nossas certificações similar ao que é feito no servidor (no caso do utilizador pedro):

```
sudo nano /etc/ipsec.secrets
    pedro : EAP    tar
```

E similar também ao servidor é necessário configurar o Strongswan, ou seja, temos de adicionar as nossas configurações ao ficheiro /etc/ipsec.conf:

```
config setup

conn ikev2-rw
    right=<ip_publico_servidor>
    rightid=<ip_publico_servicor>
    rightsubnet=0.0.0.0/0
    rightauth=pubkey
    leftsourceip=%config
    leftid=andre
    leftauth=eap-mschapv2
    eap_identity=%identity
    keyexchange=ikev2
```

Nota: O cliente precisa de assegurar que não tem nenhuma interface de rede com a subnet 192.168.1.0/24 porque senão ira haver interferência e não será possível conectar aos recursos da rede do servidor pois ele tem essa mesma subnet.

Depois de ter o cliente configurado temos duas opções: ou utilizamos o strongswan-starter para termos uma ligação mais permanente ou então utilizamos o charon-cmd para uma conexão isolada. No primeiro caso convém desabilitar o strongswan-starter para que ele não se ligue automaticamente quando a máquina é iniciada: ”$ sudo systemctl disable –now strongswan-starter”. E depois podemos correr o comando ($ sudo systemctl start strongswan-starter) para iniciar a ligação e stop para terminar a ligação. Ou então podemos utilizar a segunda opção:

```
sudo charon-cmd --cert /etc/ipsec.d/ca-cert.pem --
    host <ip_vpn> --identity pedro
```

V. Resultados

Neste capítulo apresentamos os resultados e o desempenho da nossa implementação IPSec VPN.

A. Teste da VPN

Primeiramente, o protocolo IKEv2 negoceia os protocolos de segurança entre o host e o servidor VPN (proxy) através do envio de pares de mensagens ISAKMP. Isto para estabelecer um SA (Security Association). Na Fig. 5 vemos que o host cliente com o IP 10.02.15 é o initiator que faz um pedido de troca de chaves e o servidor com o IP 176.79.17.216 é o responder que responde ao pedido.

No primeiro par IKE_SA_INIT, o initiator envia sugestões para os parâmetros de segurança dos algoritmos de cifração, algoritmos de integridade, funções pseudo-aleatórias e o grupo de Diffie-Hellman (DH). As sugestões dos algoritmos enviados são pré-definidos pelo strongswan. Estes parâmetros protegem a partilha das mensagens seguintes IKE_AUTH.

O responder responde ao pedido enviando os algoritmos escolhidos para cada parâmetro (algoritmos pré-definidos pelo strongswan). Tanto o initiator como o responder, também enviam os valores correspondentes de Diffie-Hellman no payload Key Exchange.

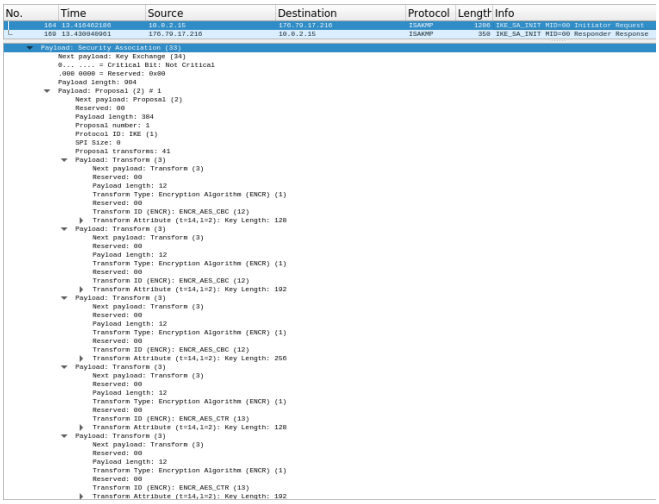

Fig. 5. Mensagem IKE_SA_INIT pelo ”Initiator”

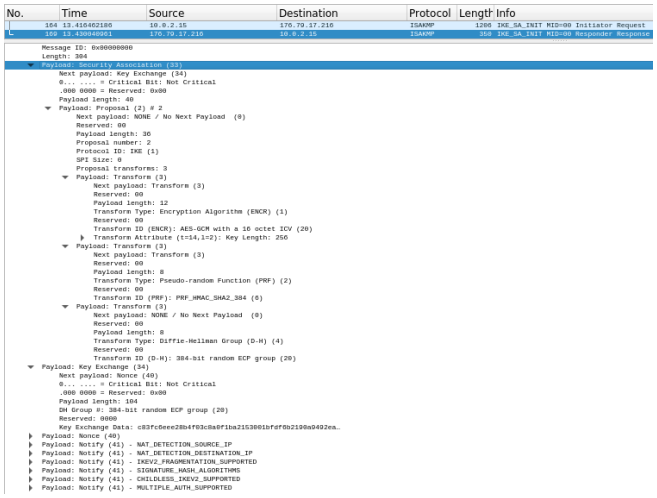

Fig. 6. Mensagem IKE_SA_INIT pelo ”Responder”

Após a troca de mensagens INIT, ambos os extremos geram uma chave de sessão de segurança diferente composta pela chave de cifra e pela chave de autenticação. Posto isto, os dois compartilham as suas chaves que são utilizadas na troca de mensagens IKE_SA_AUTH.

Numa segunda fase, há uma troca de mensagens cifradas IKE_SA_AUTH que são usadas para a criação dos CHILD_SAs.

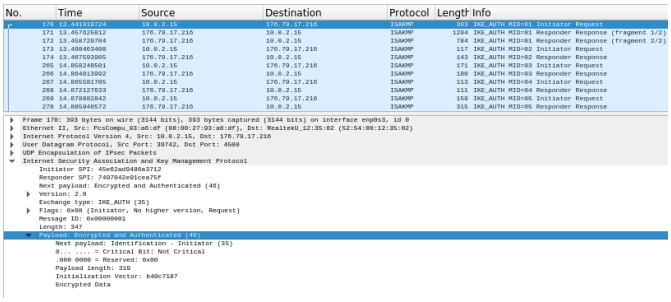

Fig. 7. Mensagens IKE_SA_AUTH pelo ”Initiator”

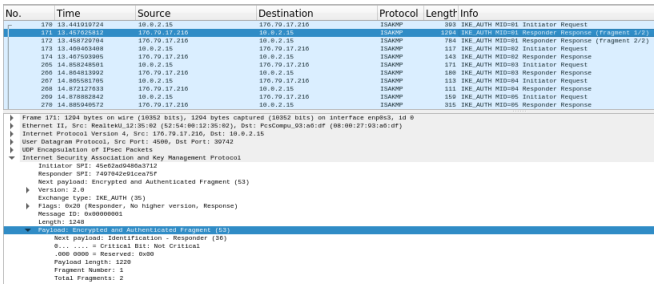

Fig. 8. Mensagens IKE_SA_AUTH pelo ”Responder”

A partir desta fase, as SAs do protocolo IPSec estão configuradas e periodicamente são enviadas mensagens cifradas INFORMATIONAL que podem conter mensagens keep alive para reconfirmar os hosts ativos e entre outras notificações. Na Fig. 9 temos um exemplo de uma mensagem keep alive.

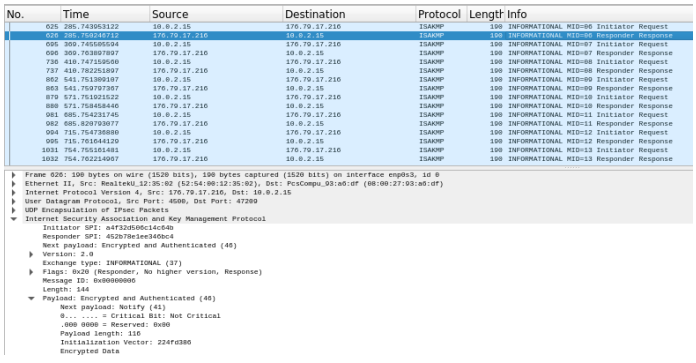

Fig. 9. Mensagens INFORMATIONAL para manter a sessão entre os dois hosts

Posto isto, conseguimos verificar que há uma conexão segura entre o host cliente e o router do servidor VPN. Os pacotes são encapsulados em pacotes cifrados ESP no percurso da rede Internet entre o cliente e o servidor. Todo o tráfego da Internet passa pelo servidor VPN que atua neste caso como proxy.

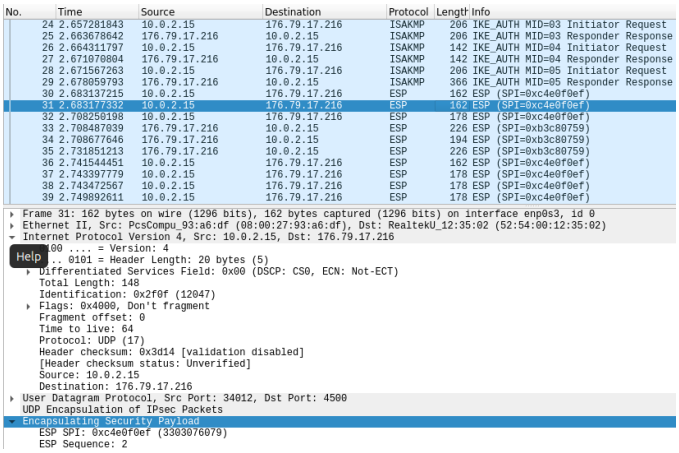

Fig. 10. Tráfego encapsulado em pacotes cifrados ESP

O túnel de encapsulamento permite a ligação do cliente à LAN do servidor, fazendo assim parte da mesma. Verificamos que havia conetividade entre o cliente VPN e todos os hosts da LAN pertencente ao servidor VPN. Também verificamos que o ip público que nos identifica é o ip público que está por detrás da LAN do servidor VPN.

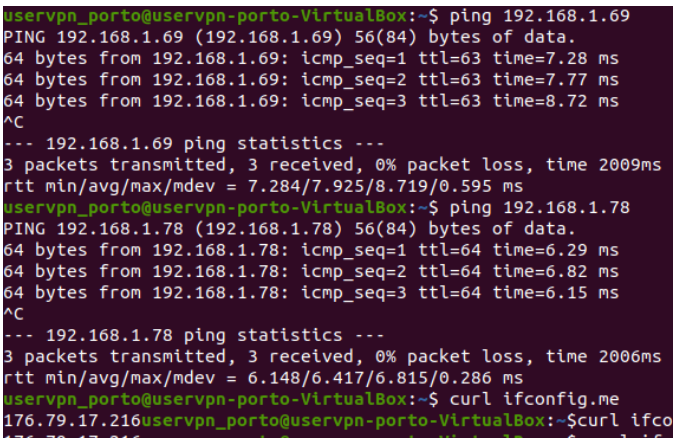

Fig. 11. Ping para hosts da LAN 192.168.1.0/24 do servidor VPN e ip público

B. Internet

Para avaliar o desempenho de uma VPN devemos ter em conta alguns fatores:

- **Velocidade da Internet:** Testar o desempenho e a velocidade da Internet sem a VPN ativa para depois comparar com a velocidade da Internet pela VPN.
- **Distância para a localização dos servidores:** As melhores velocidades surgem de servidores mais próximos. Quanto mais viajar um pedido, mais tempo demorará para se conectar com o servidor e retornar para o dispositivo.
- **Componentes de Hardware do servidor VPN:** A velocidade da Internet pode diminuir consoante os utilizadores que estão a utilizar o servidor VPN. Mais recursos de Hardware nos servidores VPN providenciam melhores velocidades com vários utilizadores a utilizarem o serviço em simultâneo.

- **Cifração na VPN:** A principal característica para que a velocidade da Internet seja afetada na utilização das VPN é devido à cifração. Uma VPN tem de cifrar os dados, para os manter seguros de possíveis espetadores na rede, e em seguida decifrá-los. Algoritmos de cifras robustos podem afetar e diminuir as velocidades.

Nesta implementação realizamos testes de desempenho sobre os algoritmos de cifração utilizados, assim como os servidores de teste da Internet. Nos algoritmos de cifração para a troca de chaves e para o encapsulamento (par IKE/ESP) testamos os algoritmos:

```
IKE: CHACHA20_POLY1305/PRF_HMAC_SHA2_512/CURVE_25519
ESP: AES_CBC_256/HMAC_SHA1_96/NO_EXT_SEQ

IKE: AES_CBC_256/HMAC_SHA2_512_256/PRF_HMAC_SHA2_512
    /MODP_8192
ESP: AES_CBC_256/HMAC_SHA2_512_256/NO_EXT_SEQ

IKE: AES_GCM_16_256/PRF_HMAC_SHA2_384/ECP_384
ESP: AES_CBC_256/HMAC_SHA2_512_256/NO_EXT_SEQ
```

Com qualquer destes pares obtivemos os resultados idênticos, ou seja, a velocidade não sofreu com a mudança de cifras. Contudo achamos que não sofreu alterações significativas por estarmos perante uma distância de cliente-servidor de 40kms. Com distâncias maiores normalmente iria haver uma quebra de velocidades quando utilizássemos o algoritmo robusto SHA_512.

Os resultados apresentados na Fig. 12 foram obtidos com a ferramenta open-source speedtest. A Fig. 12 mostra dois testes. Um com a VPN ativada (ISP=MEO) e outro sem VPN (ISP=Vodafone Portugal). Testamos em vários servidores diferentes mutuamente e obtivemos sempre os mesmos resultados.

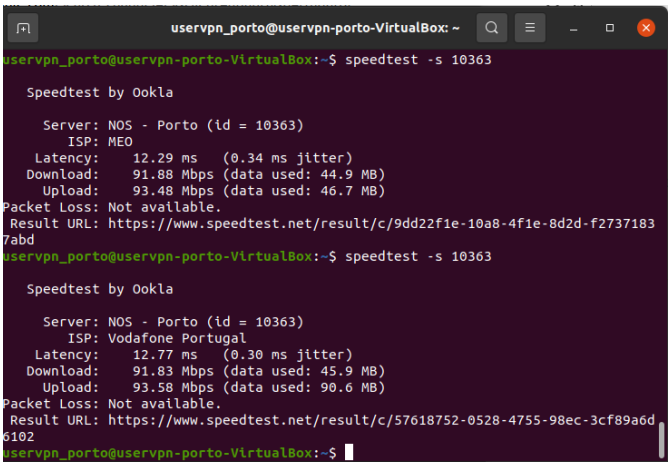

Fig. 12. Teste de métricas da Internet

VI. Conclusões

Uma VPN possui inúmeras vantagens e é uma tecnologia bem vinda que permitiu a criação de novos modelos de implementação software como a cloud (software as a service), permitiu também muitos trabalhos serem realizados de forma remota, entre outros. Contudo as VPNs acarretam várias dificuldades tanto a nível de segurança como de desempenho. Como esta tecnologia requer o uso de redes públicas, como a internet, nós não temos nenhum controle sobre esse meio, portanto cada caso pode ser bastante diferente um do outro, pois nesse meio passamos por diversas conexões, e por exemplo como sabemos a nossa conexão é sempre limitada pela conexão com menor banda larga. Contudo o desempenho também varia consoante o horário pois pode haver mais afluência na rede, o hardware do cliente e do servidor, a distância entre eles, a banda larga de cada um, etc. Mas em relação ao que podemos controlar, é possível escolher os protocolos usados para implementar as VPN. No artigo falamos de vários protocolos populares, onde todos oferecem segurança das comunicações embora cada um tenha as suas particularidades. O OpenVPN é considerado o mais estável, o mais seguro por ser open source e possui também a possibilidade de passar por firewalls. O IKEv2/IPsec normalmente é o segundo colocado porque também é seguro pois possui algoritmos e protocolos com provas dadas e é considerado um protocolo bastante eficiente a nível de hardware e possui um conjunto de particularidades na sua arquitetura que é favorável para dispositivos móveis. Por último, outro protocolo que utiliza o IPsec, o LT2P/IPSec, é o mais lento dos três devido a sua encapsulação, o processo de encriptação tem de ser realizado duas vezes. Possui segurança igual ao IKEv2.

Apresentamos os resultados da implementação IPSec de uma VPN. Através do wireshark apresentamos uma visão de como é que a troca de chaves é estabelecida entre o cliente e o servidor VPN. Neste processo demos ênfase aos protocolos IKEv2 e aos algoritmos de cifra utilizados. Também verificamos que o tráfego da rede foi encapsulado sobre o protocolo ESP, fazendo assim um túnel sobre a Internet do cliente VPN até ao servidor VPN. Por fim, apresentamos a alteração do ip público do cliente para o ip público referente ao servidor VPN, assim como métricas de atraso dos pacotes (latência) e velocidade da Internet (download). As métricas que foram apresentadas são limitadas pela banda larga mais baixa entre o cliente e o servidor VPN. O servidor VPN tinha uma velocidade de 100mbps na Internet.

References

[1] Referência da implementação, URL https://www.digitalocean.com/community/tutorials/how-to-set-up-an-ikev2-vpn-server-with-strongswan-on-ubuntu-20-04

[2] Charlie Scott, Paul Wolfe, Mike Erwin. Virtual Private Network, 2nd edition. O'reilly, 1998

[3] What is IPSec, URL www.cloudflare.com/learning/network-layer/what-is-ipsec/

[4] IPSec protocol detais for implementing VPNs, URL https://searchnetworking.techtarget.com/feature/IPsec-protocol-details-for-implementing-VPNs