TÉCNICO LISBOA

Multi-Túneis:

Implementação, configuração e avaliação prática de túneis de rede com múltiplas camadas de segurança

Ricardo Jorge Pimenta Lopes

Dissertação para obtenção do Grau de Mestre em

Segurança de Informação e Direito no Ciberespaço

Orientador(es): Prof. Miguel Filipe Leitão Pardal

Júri

Presidente: Prof. Carlos Manuel Costa Lourenço Caleiro
Orientador: Prof. Miguel Filipe Leitão Pardal
Vogal: Prof. José Manuel de Campos Lages Garcia Simão

novembro 2021

Para a minha família...

Agradecimentos

Em primeiro lugar, ao meu orientador, Professor Doutor Miguel Filipe Leitão Pardal, agradeço todo o conhecimento que me transmitiu, a sua disponibilidade e a sua dedicação. Foi uma ajuda muito importante para a realização deste meu objetivo académico.
Aos meus maiores apoios. À minha família por me incentivar e sempre me apoiar.
Quero ainda agradecer a todas as pessoas que de alguma forma contribuíram para a realização deste projeto.
Por último gostaria de agradecer a todos os colegas e professores que ao longo do meu percurso no Instituto Superior Técnico, na Faculdade de Direito, da Universidade de Lisboa, e na Escola Naval, me apoiaram e influenciaram de forma positiva.

A todos,
O meu mais sincero agradecimento.

Resumo

A comunicação em rede de computadores está na base do funcionamento do mundo moderno. Quer seja para uso pessoal ou organizacional, a comunicação em rede interliga pessoas e máquinas. Porém, na grande maioria das vezes, esta comunicação necessita de utilizar uma infraestrutura de rede pública de forma a interligar os dois pontos da comunicação. Ao sair da esfera do utilizador, essa comunicação fica vulnerável aos mais diversos ataques, quer seja por escuta passiva da comunicação, quer seja por interferência ativa no conteúdo e no fluxo da comunicação. É aqui que protocolos de segurança na camada de transporte como o TLS (*Transport Layer Security*, em Inglês) têm um papel fundamental, pois podem garantir confidencialidade, integridade, frescura e autenticidade da comunicação. Neste trabalho é abordada especificamente uma implementação estendida do TLS denominada MultiTLS, que permite compor vários canais TLS para obter segurança acrescida. Ao utilizar múltiplos conjuntos de cifras, uma vulnerabilidade explorada num dado algoritmo não compromete a mensagem porque existem outros conjuntos de cifras, tornando-se tolerante a vulnerabilidades. Foi ainda avaliado o desempenho desta ferramenta de forma a comprovar a viabilidade da sua utilização em diversos contextos pessoais e empresariais.

Palavras-chave: Segurança na Camada de Transporte, Segurança, Comunicação Segura, Confidencialidade, Túnel, Encapsulamento, Rede Privada Virtual

Abstract

Computer network communication is at the foundation of how the modern world works. Whether for personal or organisational use, network communication connects people and machines. However, most of the time this communication needs to use a public network infrastructure in order to connect the two points of communication. When leaving the sphere of the user, this communication becomes vulnerable to a diversity of attacks, either by passive listening of the communication, or by active interference in the content and flow of the communication. This is where transport layer security protocols like TLS (Transport Layer Security) play a key role, as they can guarantee confidentiality, integrity, freshness and authenticity of the communication. In this paper, an extended implementation of TLS called MultiTLS is specifically addressed, which allows composing multiple TLS channels for enhanced security. By using multiple cipher suites, a vulnerability exploited in a given algorithm does not compromise the message because there are others, thus becoming vulnerability tolerant. The performance of this tool was also evaluated in order to prove the viability of its use in various personal and corporate contexts.

Keywords: Transport Layer Security, Security, Secure Communication, Confidentiality, Tunnel, Encapsulation, Virtual Private Network

Conteúdo

Agradecimentos . . . iii
Resumo . . . v
Abstract . . . vii
Lista de Tabelas . . . xi
Lista de Figuras . . . xiii

1 Introdução **1**
1.1 Retrospectiva histórica da comunicação segura . . . 1
1.2 Segurança nas comunicações com base na diversidade criptográfica . . . 2
1.3 Aspetos práticos na utilização de múltiplos túneis . . . 3
1.4 Visão global do documento . . . 4

2 Enquadramento Teórico **5**
2.1 Comunicação segura . . . 5
2.1.1 IPsec . . . 5
2.1.2 SSL/TLS . . . 6
2.1.3 SSH . . . 9
2.2 Tunelamento . . . 10
2.2.1 PPP . . . 11
2.2.2 PPTP . . . 11
2.2.3 L2TP . . . 12
2.2.4 IPv6 . . . 13
2.3 Redes Privadas Virtuais . . . 14
2.4 vtTLS . . . 15
2.5 Resumo . . . 16

3 Solução Multi-Túneis **17**
3.1 MultiTLS . . . 17

3.2 Arquitetura do MultiTLS . . . 18
3.2.1 Interfaces TUN . . . 18
3.2.2 Biblioteca . . . 19
3.2.3 Ferramenta . . . 20
3.3 Atualização do MultiTLS . . . 22
3.4 Conjuntos de cifra . . . 23
3.5 Configurações de ligação . . . 24
3.6 Manual de utilização . . . 25
3.7 Resumo . . . 26

4 Estudo de Casos **27**
4.1 Rede-para-Rede . . . 27
4.1.1 Comunicação segura entre áreas de operação . . . 28
4.1.2 Comunicação segura entre duas soluções na nuvem . . . 29
4.2 Máquina-para-Rede . . . 31
4.3 Máquina-para-Máquina . . . 33
4.4 Resumo . . . 35

5 Avaliação **37**
5.1 Preparação dos testes . . . 37
5.2 Realização dos testes . . . 38
5.3 Resultados . . . 39
5.4 Discussão . . . 41

6 Conclusão **45**
6.1 Contribuições . . . 45
6.2 Trabalho futuro . . . 46

Bibliografia **47**

A Tabelas de resultados **51**

B Readme **57**

Lista de Tabelas

3.1 Bibliotecas/Ferramentas Atualizadas 22
3.2 Conjuntos de cifra 24
3.3 Configurações de encaminhamento de portos 25

4.1 Comunicação segura entre áreas de operação 29
4.2 Comunicação segura entre duas soluções na nuvem 30
4.3 Comunicação segura entre colaborador e organização 32
4.4 Comunicação segura entre aplicações legadas 35

5.1 Valores de teste para k = 0 39
5.2 Valores de teste para k = 1 40
5.3 Valores de teste para k = 2 40
5.4 Valores de teste para k = 3 40
5.5 Valores de teste para k = 4 40
5.6 Velocidade de transferência 43

A.1 Valores para k = 0 52
A.2 Valores para k = 1 52
A.3 Valores para k = 2 53
A.4 Valores para k = 3 54
A.5 Valores para k = 4 55

Lista de Figuras

2.1 Diagrama de Tunelamento 11
2.2 Diagrama da VPN para o cenário Máquina-para-Rede 15
2.3 Diagrama da VPN para o cenário Rede-para-Rede 15
2.4 Diagrama da VPN para o cenário Máquina-para-Máquina 15

3.1 Diagrama de arquitetura do MultiTLS 19
3.2 Diagrama de mensagens com dois níveis de proteção 20
3.3 Diagrama da ferramenta Socat - 1ªIteração 21
3.4 Diagrama da ferramenta Socat - 2ª Iteração 21
3.5 Menu inicial da ferramenta MultiTLS 26

4.1 Diagrama - Comunicação segura entre áreas de operação 28
4.2 Diagrama - Comunicação segura no teletrabalho 31
4.3 Diagrama - Comunicação segura entre aplicações legadas 33

5.1 Gráfico - Resultados dos testes 42
5.2 Gráfico - Velocidade de transferência (MB/s) 44

Capítulo 1

Introdução

Nos dias de hoje, os seres humanos vivem em sociedade, o que leva a uma constante necessidade de comunicação. Nesta nova era tecnológica onde grande parte da comunicação passou a ser online, é importante manter estas comunicações seguras. Contudo, a necessidade de comunicar em segurança já não vem dos dias de hoje. Se observarmos o percurso do ser humano desde que se começou a organizar em pequenas civilizações, é possível notar que não tardou até existirem disputas entre estes grupos organizados de indivíduos. Já nesta altura, era importante assegurar a comunicação de forma segura entre os indivíduos. Durante esses conflitos, comunicar em segurança era essencial pois caso as mensagens fossem comprometidas o inimigo ganharia uma grande vantagem estratégica. Desta necessidade de dois indivíduos comunicarem em segurança, sem que um terceiro consiga retirar sentido da comunicação caso esta seja intercetada, nasceu o conceito de criptografia.

1.1 Retrospectiva histórica da comunicação segura

O termo criptografia é proveniente das palavras gregas *kryptos* e *graphein*, que significam *escondida* e *escrita* respetivamente [19], ou seja, escrita escondida. A primeira forma de criptografia era simplesmente a escrita, já que na altura a maior parte das pessoas não sabiam ler, contudo, outras formas mais sofisticadas de ocultar o significado de uma mensagem foram elaboradas. Em 1900 a.c. os Egípcios utilizavam os seus famosos hieróglifos, já que apenas um grupo muito restrito de pessoas os conseguia decifrar. Por outro lado, também os gregos inventaram a sua própria forma de comunicar em segredo, utilizando uma ferramenta chamada *Scytale*, através da qual era possível realizar uma cifra de transposição [27]. Neste tipo de cifra, era utilizado um pedaço de madeira com uma ligadura enrolada, na qual seria escrita a mensagem. Esta ligadura, quando desenrolada, apresentava uma mensagem impercetível, sendo que apenas alguém

que utilizasse um pedaço de madeira com o mesmo diâmetro poderia decifrar a mensagem [31]. Os romanos, por sua vez, utilizavam o algoritmo de César, que consistia em substituir os caracteres da mensagem pela posição da letra no alfabeto + k [31]. Assim, se numa mensagem existir a palavra "CESAR" e for utilizado o algoritmo de César com k = 3, a mensagem cifrada será "FHVDU". Para decifrar a mensagem bastaria conhecer o valor de k e subtrair a posição dos caracteres da mensagem. Porém, estas cifras ficaram obsoletas, sendo que com o evoluir do tempo foram criados algoritmos de criptografia cada vez mais complexos, contudo, também cada vez mais existiam indivíduos com a capacidade de quebrar estes algoritmos. Um ótimo exemplo disto foi a segunda guerra mundial, onde a cifra de mensagens foi muito utilizada, tendo mesmo um papel fundamental. Do lado da Alemanha existia uma poderosa máquina de cifrar mensagens denominada Enigma. O algoritmo desta máquina seria tão complexo que se acreditava na impossibilidade deste ser quebrado, tendo sido utilizado para cifrar até as mensagem mais secretas durante a segunda guerra mundial. Esta máquina atribuía aos alemães uma enorme vantagem estratégica em relação às forças aliadas. Porém, devido aos trabalhos do matemático Alan Turing, durante a segunda guerra mundial, foi possível quebrar a segurança da máquina Enigma e decifrar as mensagens do lado alemão [14]. Acredita-se que os trabalhos de Alan Turing, não só contribuíram em grande parte para o desfecho do conflito, como reduziram também a sua duração em vários anos [14]. Um dos grandes erros das forças alemãs foi acreditar que as mensagens trocadas estavam perfeitamente seguras, já que o sistema de cifra utilizada era considerado por muitos perfeito. Esta crença de que uma comunicação está perfeitamente segura quando utilizados algoritmos criptográficos no âmbito de cifrar a informação, resulta numa falsa sensação de segurança que é notória mesmo nos dias de hoje, onde os algoritmos criptográficos utilizados são cada vez mais sofisticados.

1.2 Segurança nas comunicações com base na diversidade criptográfica

Se antes os algoritmos apenas garantiam a confidencialidade da mensagem, hoje garantem também outras propriedades como integridade, frescura e autenticidade da comunicação. Ainda assim, não é possível garantir que dado algoritmo é inquebrável e que determinada comunicação é cem por cento segura. Porém, é possível adicionar camadas de segurança a uma comunicação, de forma a tornar o trabalho de um atacante extremamente difícil.

O objetivo deste trabalho é perceber de que forma estas camadas de segurança podem ser adicionadas, bem como o seu impacto, através da implementação, configuração e avaliação prática

de túneis de rede com múltiplas camadas de segurança. Neste trabalho, baseado no trabalho de Moura R. [26], será analisado de que forma a utilização de algoritmos criptográficos e, mais concretamente a utilização de túneis, pode contribuir para uma comunicação de segurança reforçada entre dois pontos, mesmo que para tal seja necessário utilizar uma infraestrutura de rede pública.

1.3 Aspetos práticos na utilização de múltiplos túneis

Durante a realização deste trabalho, foi feita uma nova implementação e respetivos testes de uma ferramenta já existente, denominada MultiTLS [26]. Esta ferramenta, visa estabelecer uma comunicação de segurança reforçada através de múltiplos canais cifrados numa arquitetura cliente/servidor. Através desta ferramenta, é possível comunicar de forma segura em comunicações que antes não apresentavam qualquer tipo de segurança, ou cujo nível de segurança era obsoleto. Um exemplo disso é utilização de aplicações legadas. Aplicações legadas, são programas que, derivado da sua idade, nos dias de hoje apresentam várias vulnerabilidades de segurança [33]. Contudo, estas aplicações são ainda parte importante do funcionamento de certas áreas de negócio. Com recurso ao MultiTLS, é possível transportar a comunicação entre aplicações legadas mantendo-a segura através de algoritmos criptográficos modernos.
Para a implementação da ferramenta, foi em primeiro lugar necessário, atualizar as dependências utilizadas pela ferramenta, de modo a que funcionassem de forma otimizada em versões mais recentes de sistemas operativos *Linux*. De seguida, foram criados vários módulos de forma a automatizar e facilitar a utilização da ferramenta:

- Cliente - executa a ferramenta em modo cliente;
- Servidor - executa a ferramenta em modo servidor;
- Instalador/Desinstalador - instala/desinstala as dependências necessárias ao funcionamento da ferramenta;

A nova implementação da ferramenta foi pensada de modo a facilitar a sua configuração. Para tal, quando o utilizador inicia uma ligação MultiTLS, será guiado por todas as configurações necessárias para estabelecer a comunicação através da ferramenta.

Por último, foi criada uma interface de modo a permitir ao utilizador controlar os diferentes módulos de forma mais simples e intuitiva. O utilizador apenas necessita de iniciar a ferramenta e seguir os passos apresentados no ecrã, consoante este pretenda iniciar o modo cliente, iniciar o modo servidor, instalar/desinstalar dependências ou terminar a execução da ferramenta.

De forma a avaliar o comportamento da ferramenta, foram ainda realizados testes de desempenho. Para estes testes, foram utilizadas duas máquinas nas quais foi executada a ferramenta MultiTLS (uma em modo cliente e outra em modo servidor). Após estabelecida a comunicação entre cliente e servidor, foi medido o tempo de envio de ficheiros com tamanhos diferentes e utilizando diferentes configurações da ferramenta.

1.4 Visão global do documento

Neste primeiro capítulo, na introdução, foi realizada uma exposição inicial a esta investigação, bem como aos seus objetivos e ao que foi desenvolvido. O presente trabalho encontra-se apresentada nos próximos cinco capítulos, organizados da seguinte forma:

- No capítulo 2 (Enquadramento Teórico), são apresentados os principais conceitos base necessários à compreensão do trabalho;
- No capítulo 3 (MultiTLS), é apresentada a ferramenta MultiTLS ao nível do seu funcionamento, arquitetura, dependências e configurações necessárias;
- No capítulo 4 (Estudo de Caso), são definidos casos onde a utilização da ferramenta MultiTLS pode contribuir para reforçar a segurança de uma comunicação;
- No capítulo 5 (Avaliação), são explicados os testes realizados à ferramenta, bem como os resultados encontrados e a sua análise;
- Finalmente, no capítulo 6 (Conclusão), é realizada uma exposição do que se pode sumariar acerca da viabilidade da ferramenta MultiTLS no decurso dos testes realizados.

De forma a auxiliar a leitura do documento, é feita, no início de cada capítulo, uma introdução ao tema que será tratado (exceto na conclusão), bem como, no final desses capítulos, um resumo dos conceitos apresentados nos mesmos. Com este trabalho é possível avaliar o comportamento que uma solução como o MultiTLS poderá ter num cenário de implementação real, bem como as vantagens e desvantagens da sua utilização.

Capítulo 2

Enquadramento Teórico

Neste capítulo são apresentados os principais protocolos utilizados para estabelecer comunicações seguras. Mais concretamente é apresentado o conjunto de protocolos que formam o protocolo IPsec (*Internet Protocol Security*), bem como uma descrição de todos os processos que compõem o protocolo TLS (*Transport Layer Security*) e é ainda apresentado o protocolo SSH (*Secure Shell Protocol*). De seguida, é apresentado o conceito de tunelamento e a sua utilização em alguns protocolos como PPTP e L2TP, e no processo de tornar possível a comunicação de dois *routers* IPv6 através de uma rede IPv4. E por fim, é explicado o conceito de VPN, assim como os seus diferentes cenários de utilização.

2.1 Comunicação segura

Com a adesão ao meio tecnológico, em especial por parte das organizações, foi necessário criar mecanismos que permitissem estabelecer trocas de dados de forma segura e o mais eficiente possível. Ao longo dos anos, foram desenvolvidos vários protocolos que, não só permitem estabelecer comunicações de forma segura, como também agilizar processos, contribuindo para a evolução do novo mundo tecnológico.

2.1.1 IPsec

O IPsec ou *Internet Protocol Security* é um conjunto de protocolos que permite comunicações seguras através da Internet, baseado nas normas desenvolvidos pelo IETF, garantindo confidencialidade, integridade, autenticidade e frescura dos dados, o que o torna uma ótima ferramenta VPN [3]. O IPsec pode correr em dois modos de operação: o modo de transporte e o modo de túnel. Sendo que em cada modo podem ser utilizados dois protocolos diferentes: o AH (*Authentication Header*) e o ESP (*Encapsulation Security Protocol*). O modo de transporte é

normalmente utilizado entre duas máquinas (VPN máquina-para-máquina), porém, pode ser utilizado também entre uma máquina e uma rede, se este último for destinatário da mensagem. Neste modo o cabeçalho IP original é utilizado para roteamento pelo que os endereços IP não estão protegidos. O modo de túnel é utilizado em comunicações VPN rede-para-rede ou máquina-para-rede, sendo que o router atua como *proxy* para o *host* que se encontra por detrás. Neste modo o cabeçalho IP original é encapsulado e um novo cabeçalho IP é adicionado, desta maneira os endereços de IP das máquinas são protegidos e apenas os VPN *gateways* têm conhecimento dos mesmos [25]. O protocolo AH providencia autenticação, integridade e frescura das mensagens, porém não define mecanismos de forma a garantir a confidencialidade das mensagens trocadas. Para isso é utilizado o protocolo ESP que além de autenticação, integridade e frescura das mensagens também permite que as mesmas sejam cifradas, garantindo a confidencialidade dos dados [15]. Um dos grandes problemas da utilização do IPsec são as *firewalls NAT (Network Address Translation)* que impossibilitam a utilização do IPsec AH em ambos os modos (transporte e túnel) e IPsec ESP em modo de transporte. A solução mais fácil passa por utilizar o IPsec com ESP em modo de túnel, sendo também a opção mais segura em relação as outras opções do IPsec. Contudo, desta segurança advém um custo, que é o aumento do *overhead*.

2.1.2 SSL/TLS

SSL ou *Secure Sockets Layer* foi originalmente desenvolvido em 1994 pela *Netscape* como uma forma de comunicação segura através da rede numa arquitetura cliente/servidor, garantindo autenticidade, confidencialidade e integridade da ligação. Ao longo dos anos o SSL foi alvo de várias melhorias tendo existido várias versões do SSL, até que em 1999 a primeira versão do TLS ou *Transport Layer Security* (TLS 1.0) foi apresentada, surgindo assim o sucessor do SSL. O TLS passou também ele por várias alterações até aos dias de hoje, onde a versão TLS 1.3 permanece a mais atual [26]. O protocolo TLS é composto por vários protocolos: o *TLS Record Protocol*, o *TLS Handshake Protocol*, o *Change Cipher Spec Protocol* e o *Alert Protocol*.

TLS Record Protocol

O *TLS Record Protocol* é utilizado pelo *TLS Handshake*, *TLS Alert Protocol*, *TLS Change Cipher Spec Protocol* e pelos dados de aplicação [9] e "permite desenvolver mecanismos para enviar e receber mensagens" [26]. A camada do *TLS Record* admite dados não interpretados de camadas superiores em blocos não vazios de tamanho arbitrário. De seguida esses dados sofrem um processo de fragmentação, onde são transformados em blocos de tamanho menor ou igual

a 214 bytes, estes blocos denominam-se *TLS Plaintext*. Na fragmentação não é tido em conta os limites da mensagem do cliente, ou seja, tanto pode haver múltiplas mensagens dentro do mesmo *TLS Plaintext* como uma mensagem dividida em vários *TLS Plaintext* [9]. De seguida o *TLS Plaintext* sofre um processo de compressão onde se originam blocos cujo tamanho não deve exceder 214 + 1024 bytes, ou seja, o processo de compressão não deve aumentar o conteúdo em mais de 1024 bytes. Depois deste processo os blocos passam a ser *TLS Compressed* [9]. Após a fase de compressão é adicionado ao *TLS Compressed* o respetivo MAC (*Message Authentication Code*) e de seguida todo o bloco (*TLS Compressed* + MAC) é cifrado, originando assim o *TLS Ciphertext*, que não poderá exceder o tamanho de 214 + 2048 bytes. Terminado esta última fase o bloco está pronto a ser enviado ao destino, onde serão efetuadas as operações inversas seguida da entrega dos dados às camadas superiores [9]. Durante a primeira execução do *TLS Handshake* o *TLS Record* não inicia os processos de compressão, cifra, nem existe a computação do MAC, pois o servidor e o cliente ainda não acordaram os algoritmos que vão ser utilizados nestas etapas [26].

TLS Handshake Protocol

O *Handshake Protocol* é responsável por negociar a sessão que consiste nos seguintes pontos: um identificador de sessão (escolhido pelo servidor), os certificados (norma x509), o algoritmo de compressão utilizado para originar os bloco *TLS Compressed* no *TLS Record Protocol*, as especificações de cifra (algoritmo de MAC e de cifra utilizados no *TLS Record Protocol* para originar os *TLS Ciphertext*), um *master secret* (partilhado entre o cliente e o servidor) e a *flag is resumable* que indica se a sessão pode ser utilizada para iniciar novas ligações [9].

Change Cipher Spec Protocol

O *Change Cipher Spec Protocol* consiste numa mensagem cifrada e comprimida de acordo com o estado atual da ligação, para sinalizar uma mudança no conjunto de cifras negociadas. O estado de ligação do TLS é uma característica que pertence ao *TLS Record*, na qual é especificado o algoritmo de compressão, de cifra e MAC, bem como as chaves utilizadas para o MAC e para o algoritmo de cifra, tanto para leitura como para escita (*read/write*). No TLS existem quatro estados diferentes: *pending read state*, *pending write state*, *current read state* e *current write state*. Todos os processos que acontecem no *TLS Record* são processados com base nos parâmetros do estado atual (*current read/write state*). Os parâmetros de segurança para os estados pendentes (*pending read/write state*) são definidos pelo *TLS Handshake*, e o *Change Cipher Spec Protocol* faz com que o o estado pendente passe a ser o atual, eliminando o antigo.

O estado atual inicial não especifica algoritmo de cifra, compressão ou MAC. A mensagem de *ChangeCipherSpec* pode ser enviada tanto pelo cliente como pelo servidor. O emissor da mensagem deve imediatamente comunicar à camada de *record* para alterar o estado de escrita atual para o pendente, após envio da mensagem. O recetor, após receber a mensagem, deve imediatamente alterar o estado de leitura atual para o pendente [9].

Alert Protocol

O *Alert Protocol* envia uma mensagem de alerta que, dependendo da gravidade, pode ser do tipo aviso ou fatal (*warning/fatal*). Ao receber uma mensagem do tipo fatal a ligação é terminada e o identificador de sessão invalidado. Neste caso, outras ligações da mesma sessão podem continuar ativas mas não poderão ser estabelecidas novas ligações para aquela sessão. Este tipo de mensagens, tal como as outras, são cifradas e comprimidas com base no estado de ligação atual [9].

Handshake

O cliente começa por enviar uma mensagem *ClientHello* à qual o servidor responde através de uma mensagem *ServerHello*. Quando uma nova sessão começa os algoritmos de cifra, MAC e compressão, do estado de ligação do *TLS Record*, são inicializados a *null*. Por essa razão estas mensagens são utilizadas para estabelecer atributos de sessão como: o identificador de sessão, algoritmos de cifra utilizados, métodos de compressão e dois valores aleatórios). No *ClientHello*, o cliente envia para o servidor uma lista de conjuntos de cifra suportados pelo cliente, por ordem de preferência. O servidor escolhe o conjunto de cifra a ser utilizado nessa sessão e envia a resposta ao cliente. Caso não existam opções que o servidor considere satisfatórias, é enviada uma mensagem de alerta e é terminada a ligação. O servidor pode também começar uma sessão através de uma mensagem de *HelloRequest*, à qual o cliente responderá com uma mensagem de *ClientHello* [9]. Após essa fase concluída, e se não se tratar de uma ligação anónima, o servidor envia o seu certificado para ser autenticado (*Server Certificate message*). Se a mensagem, enviada pelo servidor com o certificado, não possuir informação suficiente para que o cliente consiga trocar o *premaster secret*, então adicionalmente é enviada uma mensagem de *ServerKeyExchange*. Se o servidor for autenticado, e caso tenha sido acordado no conjunto de cifras utilizado na sessão, o servidor pode efetuar um pedido de certificado ao cliente (*CertificateRequest*). De seguida o servidor envia a mensagem *ServerHelloDone* a indicar que a fase de *hello* do *handshake* acabou, e fica a aguardar o certificado do cliente. O cliente envia o certificado e de seguida uma mensagem *ClientKeyExchange*. Esta mensagem permite definir o *premaster*

secret, ou através do envio da chave RSA ou enviando os parâmetros de *Diffie-Hellman* que serão utilizados pelas duas partes para calcularem a mesma chave. Caso o servidor não faça o pedido de certificado do cliente esta deve ser a primeira mensagem enviada pelo cliente depois do *ServerHelloDone* [9]. Por último o cliente envia uma mensagem *ChangeCipherSpec*, copia as especificações de cifra pendentes para as atuais, e envia uma mensagem *Finished* para o servidor já com base nas novas especificações, acordadas no *handshake*. Em resposta, o servidor efetua o mesmo processo, ficando assim completo o *handshake*. A informação da camada de aplicação pode agora ser trocada [9].

Os dados de aplicação nunca são enviados sem que o primeiro *handshake* aconteça, pois por padrão nenhum conjunto de cifra está estabelecido (TLS_NULL_WITH_NULL_NULL) [9]
Um exemplo de um conjunto de cifra acordado numa sessão TLS é: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384_P384.

- ECDHE - *Elliptic-Curve Diffie-Hellman Ephemeral* para troca de chaves
- ECDSA - *Elliptic-Curve Digital Signature Algorithm* para assinatura digital
- AES_256_GCM - AES com chave de 256 bits com *Galois/Counter Mode* para cifrar as mensagens
- SHA384 - utilizado no MAC (*Message Authentication Code*)

2.1.3 SSH

O SSH ou *Secure Shell Protocol* foi originalmente desenvolvido por Tatu Ylonen em 1995 em resposta a um ataque informático a uma Universidade Finlandesa que comprometeu milhares de utilizadores e *passwords*. O SSH surgiu da necessidade de criar uma ferramenta que permitisse administrar sistemas e transferir ficheiros de forma segura através de redes não seguras. O SSH veio substituir algumas ferramentas como Telnet, FTP, FTP/S, etc., ainda que algumas delas ainda sejam utilizadas nos dias de hoje [29]. O SSH é composto por três principais componentes: o *Authentication Protocol*, o *Transport Protocol* e o *Connection Protocol* [5].

- *Transport Protocol*: fornece confidencialidade através de criptografia, autenticação do servidor e proteção da integridade [7];
- *Authentication Protocol*: funciona em cima do *Transport Protocol* e fornece um túnel autenticado ao *Connection Protocol*. [6];
- *Connection Protocol*: funciona em cima do *Transport Protocol* e *Authentication Protocol*, fornecendo sessões de *login*, execução remota de comandos e *forwarding* de ligações TCP/IP

e X11, através de um único túnel cifrado. [8].

Estes protocolos proporcionam ao SSH uma grande capacidade para realizar comunicações seguras através de túneis cifrados.

2.2 Tunelamento

Desde sempre que sistemas de túneis foram utilizados por vários seres vivos que habitam ou habitaram o planeta terra, incluindo o ser humano. A palavra túnel está associada a um meio para obter comunicação entre dois pontos. Este conceito remete para algo que, não só permite a comunicação, como também permite que esta seja efetuada de forma mais segura e eficiente atravessando um obstáculo, como uma montanha, por exemplo, como sucede com alguns túneis rodoviários. O conceito de túnel existe também em computação e é conseguido através da utilização de protocolos de tunelamento. Nos dias de hoje, a comunicação em rede tornou-se de certo modo fundamental para o ser humano, sendo utilizada tanto para uso pessoal como para uso profissional. Visto que a comunicação em rede está em todo lado, e é utilizada para vários fins, poderá existir, por vezes, a necessidade de que informação sensível seja transportada de um ponto da rede para outro, como por exemplo a utilização de serviços de *homebancking*. Nesse caso, é necessário garantir que essa informação seja enviada em segurança de forma a que apenas os destinatários de tal informação tenham acesso. O tunelamento é uma ferramenta que pode contribuir para essa segurança. Contudo, para que a conexão seja estabelecida com sucesso é necessário que ambas as partes entendam e utilizem o mesmo protocolo. Num protocolo de tunelamento a mensagem é encapsulada dentro do datagrama de outra mensagem permitindo assim que esta seja enviada de forma segura. Essa característica neste tipo de protocolos permite tornar viável o envio de dados entre duas redes privadas, utilizando uma infraestrutura de rede pública. Porém, a utilização deste tipo de tecnologia permite que esta seja utilizada para outros fins como, por exemplo, conceder passagem a certos protocolos em algumas *firewalls* menos sofisticadas que, de outra forma, seriam bloqueados (ao encapsular um protocolo ao qual a *firewall* aplicaria a ação de *drop*, dentro do datagrama de outro protocolo em que a *firewall* permitiria a passagem). Isto acontece porque a *firewall* não consegue detetar que dentro do datagrama daquele protocolo existe na verdade outro protocolo encapsulado [18]. Na figura em baixo é possível observar duas máquinas, que comunicam entre elas através de redes diferentes, utilizando para isso um protocolo "Y" que é encapsulado dentro de outro protocolo "X". A mensagem enviada é então constituída pelo cabeçalho do protocolo "X" e o cabeçalho e *payload* do protocolo "Y". Tanto o cabeçalho como o *payload* do protocolo "Y" podem ser cifrados para garantir a privacidade da comunicação. Por norma o cabeçalho do protocolo encapsulador (neste

caso o protocolo "X") não é cifrado.

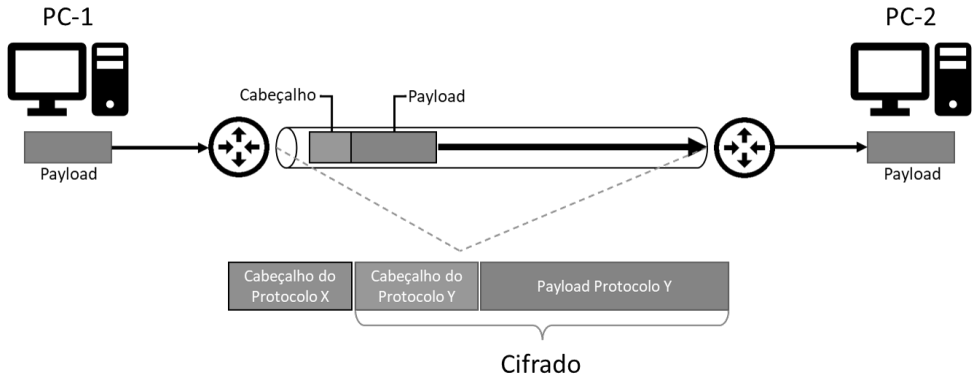


Figura 2.1: Diagrama de Tunelamento

2.2.1 PPP

PPP ou *Point-to-Point Protocol* é um protocolo de camada dois (*data-link*) utilizado para estabelecer uma conexão direta entre dois nós, sendo que permite o encapsulamento de múltiplos protocolos de camada 3. O Encapsulamento PPP define como os pacotes da camada de rede são encapsulados na frame do PPP. Para isso necessita de três campos: protocolo, informação e *padding*. "O campo protocolo permite identificar o protocolo encapsulado no campo informação. E o campo de informação pode ser preenchido por um número arbitrário de bits (*padding*) até ao *Maximum Receive Unit* (MRU)" [26].

O PPP suporta dois protocolos de autenticação: o PAP (*Password Authentication Protocol*) e o CHAP (*Challenge Handshake Authentication Protocol*). O problema da utilização do PAP é que tanto utilizador como palavra chave são enviados em formato texto pelo que é um protocolo pouco seguro [22]. O processo de autenticação no CHAP acontece em três passos: o autenticador envia um desafio, a parte que pretende ser autenticada envia um *hash* do desafio e da palavra chave e por último o autenticador verifica se o *hash* recebido corresponde ao *hash* calculado pelo autenticador [26]. Pode ainda ser utilizado o EAP (*Extensible Authentication Protocol*), com um servidor de autenticação como o *Radius*. O PPP suporta também cifra de mensagens através do ECP (*Encrypton Control Protocol*). Os protocolos utilizados são o "PPP DES Encrypton Protocol (DESE), PPP Triple-DES Encrypton Protocol e o DESE-bis" [26].

2.2.2 PPTP

O protocolo PPTP (*Point-to-Point Tunneling Protocol*) é um dos mais antigos protocolos VPN utilizados ainda nos dias de hoje, foi criado pela *Microsoft* com o objetivo de efetuar, de forma

segura, a troca de mensagens do protocolo PPP (*Point-to-Point Protocol*) através da rede, tendo sido muito utilizado como VPN comercial. Isto porque fazia parte desde cedo de sistemas operativos como o Windows 95, 98 e NT até aos dias de hoje, onde ainda faz parte da maioria dos sistemas operativos modernos [10].

Para que a ligação PPTP aconteça estão envolvidos três processos, os quais requerem que o anterior seja completo com sucesso.

- *PPP Conection and Comunication*: O cliente PPTP utiliza o protocolo PPP para estabelecer a ligação e cifrar a informação.
- *PPTP Control Connection*: Através da ligação à Internet estabelecida pelo PPP, o protocolo PPTP cria uma ligação de controlo entre o cliente PPTP e o servidor PPTP através da Internet. Esta ligação utiliza o protocolo TCP e é denominada túnel PPTP.
- *PPTP Data Tunneling*: Por último o protocolo PPTP origina os datagramas IP, os quais contêm os pacotes PPP cifrados, que de seguida são enviados através do túnel PPTP para o servidor PPTP. O servidor PPTP por sua vez desmonta o frame recebido, decifra os pacotes PPP no interior e de seguida redireciona-os para o destino[4].

Este protocolo tem como vantagem a fácil configuração, pois praticamente todas as plataformas (Windows, macOS, Android, iOS e até alguns routers) contêm o cliente PPTP instalado, pelo que não é necessário instalar software adicional para configurar uma VPN com PPTP. Contudo, este protocolo é há muito considerado como inseguro, sendo a razão para isso o facto deste utilizar a senha de utilizador para gerar as suas chaves de cifra, pelo que se a senha for fraca as chaves também o serão. É, portanto, aconselhado que seja utilizado apenas em casos de compatibilidade quando a utilização de outros protocolos mais modernos e seguros não seja possível [4].

2.2.3 L2TP

O L2TP ou *Layer 2 Tunneling Protocol*, é um protocolo VPN criado pela *Cisco* e tem as suas origens em dois protocolos: o *Layer 2 Forwarding Protocol(L2F)* da *Cisco* e o PPTP. Tal como o PPTP, este protocolo é utilizado para enviar tráfego PPP através de um túnel, herdando assim os seus métodos de autenticação e de cifra [11]. No L2TP existem dois agentes: o LAC (*L2TP Access Concentrator*) e o LNS (*L2TP Network Server*) [12]:

- O LAC atua como um dos *endpoints* do túnel e encontra-se entre o LNS e o sistema remoto encaminhando pacotes entre ambos [12];

- O LNS atua em conjunto com o LAC como *endpoint* do túnel [12].

O L2TP utiliza dois tipos de mensagens: as de controlo, utilizadas para gerir, manter e terminar as ligações, e as de dados, utilizadas para encapsular os pacotes PPP [23]. Embora o L2TP inclua também suporte para autenticação de túnel, não define mecanismos para proteger o mesmo [11]. Para isso faz uso do IPSec (*Internet Protocol Security*). Em primeiro lugar, é estabelecido um canal seguro entre o LAC e o LNS, através do IKE (*Internet Key Exchange*) como gestor de chaves, de seguida o canal seguro é utilizado para estabelecer o túnel L2TP. Os dados L2TP encapsulados são depois transportados através da ligação IPSec com ESP. "Neste caso é utilizado IPSec em modo de transporte, pois encapsular o pacote L2TP, que por si só já contem encapsulada uma frame da camada dois, num novo pacote IP resultaria numa sobrecarga excessiva no protocolo" [30]. O L2TP/IPSec é mais seguro que o PPTP, mantendo a sua fácil configuração. Em relação ao IPSec, tem como vantagem permitir que também sejam transportados outros protocolos além do IP. Outra razão pela qual se utiliza o L2TP é o facto de este permitir mecanismos de *login* aos utilizadores, ao contrário do IPSec por si só. Porém tem como desvantagem o facto de ser facilmente bloqueado por uma *firewall* devido à utilização do UDP (*User Datagram Protocol*) com o porto 500.

2.2.4 IPv6

As ferramentas de tunelamento são ainda utilizadas para que dispositivos com protocolos mais antigos possam comunicar com dispositivos que suportem protocolos mais recentes, como é o caso do *Internet Protocol* (IPv4 vs IPv6). Já na década de 90, o IETF (*Internet Engineering Task Force*), criador da norma TCP/IP, havia previsto que, num futuro próximo, o IPv4 deixaria de ser suficiente para endereçar todos os dispositivos conectados à Internet. Isso deve-se ao facto do IPv4 utilizar apenas 32 bits para endereçamento o que "apenas fornece uma quantidade inferior de 4,3 x 10^{12} de endereços, isto significa que não existe um endereço IP para cada pessoa ou dispositivo que usa a Internet" [28], "nem mesmo técnicas como NAT (*Network Address Translation*) ajudaram, apenas adiaram o inevitável" [28], o que levou à proposta de uma nova versão do protocolo IP, o IPv6. Porém, ainda existe alguma resistência quanto à sua utilização. Segundo a *Google*, num estudo acerca dos utilizadores dos seus serviços, constatou-se que desde 2010 existe um aumento significativo do número de utilizadores de IPv6, no entanto, em novembro de 2020 a percentagem de utilização ainda é inferior a 35% [13]. À fraca adesão junta-se o problema de que dispositivos mais antigos podem não suportar a nova versão do protocolo o que, tendo em conta que as duas versões não são diretamente compatíveis, a sua coexistência seria um problema. É aqui que mais uma vez o tunelamento tem um papel fundamental, pois

permite a comunicação entre dois dispositivos que comunicam em ipv6 através de uma rede de dispositivos IPv4, encapsulando o protocolo ipv6 num datagrama de IPv4.

2.3 Redes Privadas Virtuais

Uma VPN ou *Virtual Private Network* é uma conexão efetuada por um dispositivo pela rede pública para uma rede privada. Esta conexão é realizada com recurso a técnicas de tunelamento e de criptografia que garantem que dados sensíveis possam ser transmitidos de forma segura, prevenindo que pessoas não autorizadas tenham acesso a essa informação [2].

A utilização de uma VPN acarreta várias vantagens, sendo elas a possibilidade de comunicar de forma segura entre duas redes privadas ou de uma rede pública para uma rede privada, trabalhar remotamente desde de casa (*home office*) e, ainda, manter a privacidade dos dados tanto de potenciais atacantes como também dos próprios ISP (*Internet Service Providers*). Contudo, a utilização de VPN também tem algumas desvantagens, como a menor velocidade de conexão devido ao processo de cifra dos dados. Outra desvantagem está relacionada com o custo de uma VPN. Embora existam vários serviços de VPN gratuitos, muitos deles não oferecem uma proteção completa, pelo que é importante que seja feito um investimento numa boa VPN de forma a tornar a conexão mais segura. Por último, existem ainda problemas relacionados com a configuração da VPN, que podem ser um problema para o utilizador comum. E, ainda, existem os problemas legais como, por exemplo, o facto de em alguns países, como a China, as VPN serem ilegais.

Quando se fala de VPN existem vários tipos a considerar. O tipo Máquina-para-Rede permite uma comunicação segura entre uma rede privada e uma máquina que se encontra fora dessa rede. Este é o tipo mais comum, pois permite no contexto corporativo que um colaborador tenha acesso à rede do escritório da organização através de casa, como é demonstrado na figura 3.1. A VPN do tipo Rede-para-Rede é utilizada principalmente em contexto corporativo para criar um meio de comunicação seguro entre escritórios em localizações geográficas diferentes, tal como é ilustrado na figura 2.3. Por último, na figura 2.4, é demonstrado o tipo Máquina-para-Máquina que é utilizado para criar um meio de comunicação seguro entre duas máquinas. Utilizando mais uma vez o contexto corporativo como exemplo, este tipo de VPN utiliza-se quando um colaborador tem necessidade de aceder de forma segura a um recurso específico da organização, como um servidor ou base de dados, através do seu posto de trabalho.

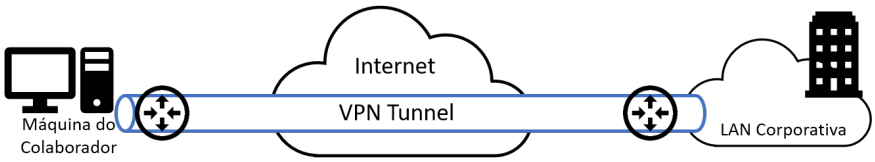


Figura 2.2: Diagrama da VPN para o cenário Máquina-para-Rede

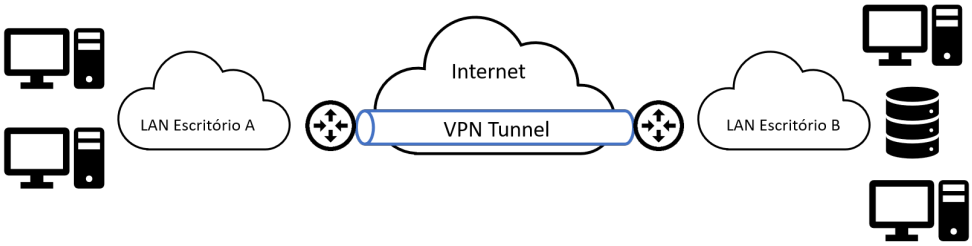


Figura 2.3: Diagrama da VPN para o cenário Rede-para-Rede

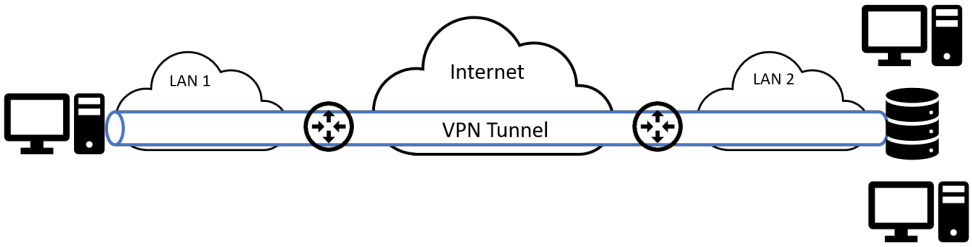


Figura 2.4: Diagrama da VPN para o cenário Máquina-para-Máquina

2.4 vtTLS

O vtTLS ou *Vulnerability-Tolerant TLS* [16], é um protocolo baseado no protocolo TLS que fornece uma ligação segura e tolerante a vulnerabilidades. Este protocolo foi desenvolvido com o objetivo de solucionar o problema de numa ligação TLS apenas existir um conjunto de cifra negociado entre cliente e servidor. Para tal, o vtTLS permite negociar vários conjuntos de cifra entre o cliente e servidor, que depois serão aplicados durante a comunicação. Assim, caso seja encontrada uma vulnerabilidade num conjunto de cifra os restantes conjuntos mantêm a comunicação segura. Contudo, este protocolo tem como principal desvantagem ser necessário alterar a biblioteca OpenSSL, sendo que terá de ser mantido a par das novas versões da biblioteca.

2.5 Resumo

Neste capítulo, foram descritos os protocolos IPSec, TLS e SSH e como estes permitem estabelecer comunicações de forma segura. De seguida, foi apresentado o conceito de tunelamento, bem como diferentes abordagens e protocolos que concretizam esse conceito. De igual forma, foi também apresentado o conceito de VPN e os seus diferentes cenários de utilização e o protocolo tolerante a vulnerabilidades vtTLS.

Capítulo 3

Solução Multi-Túneis

Neste capítulo, será apresentada a ferramenta MultiTLS, baseada no trabalho de Moura R. [26] que tem como objetivo incrementar o nível de segurança de uma comunicação através de várias camadas de proteção com cifras diversas [21]. Em primeiro lugar, será feita uma apresentação da ferramenta e arquitetura utilizadas. Na descrição da arquitetura serão também apresentadas as dependências e tecnologias utilizadas, bem como os algoritmos criptográficos aplicados. Por fim, será explicado de que forma o utilizador poderá facilmente configurar e utilizar a ferramenta MultiTLS.

3.1 MultiTLS

O estabelecimento de túneis cifrados permite uma comunicação segura através de uma infraestrutura onde a segurança não é garantida. Porém, derivado da sensibilidade da informação trocada, poderá existir a necessidade de garantir uma segurança reforçada. O MultiTLS é um *midlleware*, com vários níveis de proteção, que oferece a possibilidade de estabelecer uma comunicação segura entre dois pontos, para tal utiliza o encapsulamento de vários túneis TLS. Porém, é de realçar que o MultiTLS não se limita apenas a encapsular vários túneis TLS idênticos. Através do MultiTLS a comunicação é protegida por várias cifras diferentes (uma para para cada túnel TLS), reforçando assim o nível de segurança da comunicação. A ideia é obrigar o adversário a ter de quebrar um conjunto de diferentes cifras caso queira quebrar as propriedades da confidencialidade e integridade. Ou seja, caso um atacante encontre uma vulnerabilidade num dos algoritmos criptográficos utilizados na implementação do TLS, e consiga decifrar o primeiro túnel, a probabilidade de encontrar uma vulnerabilidade contida nos conjuntos de cifra utilizados nos restantes túneis será muito reduzida. Utilizando o MultiTLS para proteger uma comunicação, onde serão estabelecidos k * *túneis* TLS, em que $k > 1$, caso

k-1 túneis TLS sejam comprometidos a comunicação permanece segura [26]. Outra vantagem do MultiTLS é a capacidade de ser utilizado para tornar segura uma comunicação, independentemente da aplicação/programa que está a ser utilizado, já que toda a comunicação efetuada será cifrada através de múltiplos túneis TLS, sem ser necessário alterar o código-fonte destas aplicações/programas.

3.2 Arquitetura do MultiTLS

Nesta secção, serão abordados os componentes da arquitetura do MultiTLS. Em primeiro lugar, é apresentado o conceito de interfaces TUN, as quais permitem ao MultiTLS criar múltiplas interfaces de rede virtuais e de seguida são introduzidas as dependências utilizadas pela ferramenta o OpenSSL e o Socat. O OpenSSL permite realizar toda a parte criptográfica, desde criar e assinar certificados do cliente e do servidor à cifra das mensagens. No caso do Socat, é através desta dependência que o MultiTLS estabelece os vários túneis.

3.2.1 Interfaces TUN

Como referido anteriormente, o MultiTLS utiliza o encapsulamento de vários túneis TLS de modo a garantir que uma comunicação é segura. Para realizar esse encapsulamento utiliza interfaces TUN (abreviatura de TUNnel). As interfaces TUN, são interfaces de rede virtuais, atuando por isso no nível 3 do modelo OSI. Estes dispositivos podem ser utilizados para estabelecer comunicações VPN, já que permitem ao *software* responsável pela VPN, cifrar a informação antes desta ser enviada. As interfaces TUN, como operam ao nível da camada de rede, têm ainda um endereço de IP atribuído. O MultiTLS utiliza várias interfaces TUN, já que cada interface vai permitir estabelecer um túnel TLS que será encapsulado pelo túnel TLS da interface TUN seguinte. A figura 3.1 demonstra a utilização do MultiTLS para proteger uma comunicação entre dois *hosts* no qual são encapsulados dois túneis TLS (k=2).

1. A aplicação quer estabelecer e enviar a mensagem do IP 10.2.1.2 para o IP 10.2.1.1;
2. A mensagem é enviada da interface C-TUN 2 (10.2.1.2) para a interface C-TUN 1 (10.1.1.2);
3. A mensagem sofre o primeiro processo de cifra, previamente acordado, sendo enviada da interface C-TUN 1 (10.1.1.2) para a interface de rede do Cliente (192.168.50.11);
4. Nesta fase, é realizado o encapsulamento da primeira mensagem cifrada, e aplicado à mensagem uma segunda cifra, previamente acordada. A mensagem é então enviada da interface de rede do Cliente (192.168.50.11) para a interface de rede do Servidor (192.168.50.10);

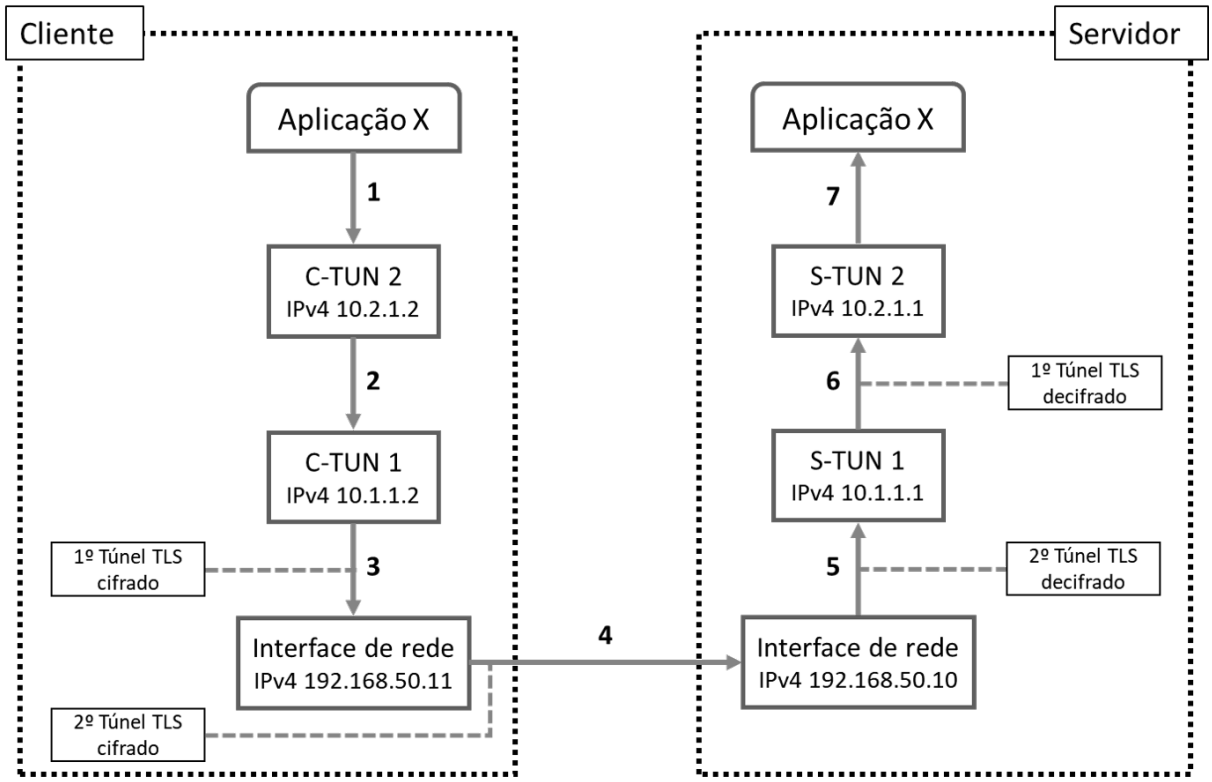


Figura 3.1: Diagrama de arquitetura do MultiTLS

5. A mensagem cifrada chega à interface de rede do servidor (192.168.50.10). Como o método utilizado para cifrar a mensagem foi previamente acordado, o servidor/interface de rede (192.168.50.10) possui a chave de sessão, e como tal, é capaz de decifrar a primeira camada de proteção;

6. A interface S-TUN 1 recebe a mensagem. Como foi acordado o método utilizado para cifrar a mensagem, o servidor/S-TUN 1 (10.1.1.1) possui a chave de sessão e portanto, é capaz de decifrar a segunda (e última para *k=2*) camada de proteção;

7. A interface S-TUN 2 recebe a mensagem. Nesta fase, a mensagem encontra-se completamente decifrada e pode ser entregue à aplicação.

Na figura 3.2 é possível observar uma demonstração do datagrama enviado no ponto 4 da figura 3.1. Para esta figura, tal como na anterior, considera-se *K=2*, ou seja, utilização da ferramenta MultiTLS com dois túneis TLS.

3.2.2 Biblioteca

O MultiTLS utiliza como dependência o OpenSSL. O OpenSSL é uma ferramenta de *software* que, por possuir uma implementação *open-source* do protocolo TLS, permite desenvolver

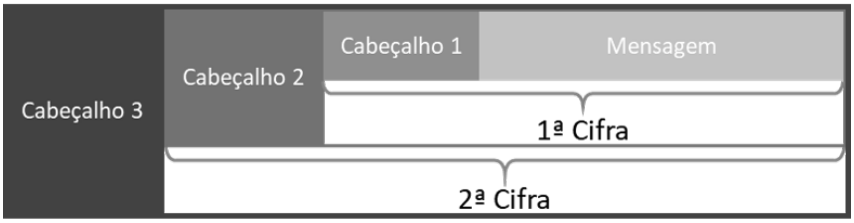


Figura 3.2: Diagrama de mensagens com dois níveis de proteção

aplicações que necessitem de comunicar de forma segura, utilizando para isso uma *suite* de algoritmos criptográficos. O OpenSSL é ainda responsável por gerar os ficheiros CSR (*Certificate Signing Request*), que serão utilizados para gerar o certificado junto da autoridade certificadora (*certificate authority* ou CA). Em ambiente de produção, estes certificados permitem a autenticação do servidor perante o cliente, e em alguns casos também do cliente perante o servidor. Para efeitos de teste no âmbito deste estudo, foram gerados através desta ferramenta certificados auto-assinados (*self-signed certificates*). Este tipo de certificados podem ser facilmente gerados, sem que para isso seja necessária a intervenção de uma autoridade certificadora, contudo, devem apenas ser utilizados para efeitos de teste e não em ambiente de produção, salvo raras exceções como, por exemplo, serviços que apenas correm dentro da rede interna de uma organização.

3.2.3 Ferramenta

O MultiTLS utiliza também como dependência a ferramenta Socat. Esta ferramenta permite a transferência de dados entre dois canais independentes, sendo responsável por criar as interfaces TUN e pela utilização do OpenSSL. Ou seja, é através do Socat que é estabelecido o túnel entre a interface TUN do lado do cliente e a interface TUN do lado do servidor, utilizando a implementação do OpenSSL de forma a proteger a ligação.
Para interligar as duas interfaces é necessário que uma das instâncias do Socat esteja à escuta de ligações (lado do servidor) e que a outra faça o pedido de ligação (lado do cliente). A figura 3.3 tem em conta uma implementação do MultiTLS com as seguintes configurações:

- IP do servidor: 192.168.50.10;
- Porto: 4444;
- Número de túneis: 2.

Através da primeira iteração é estabelecido o túnel TLS entre as interfaces eth0 - 192.168.50.10 (servidor) e eth0 - 192.168.50.11 (cliente). Na segunda iteração será estabelecido o túnel TLS

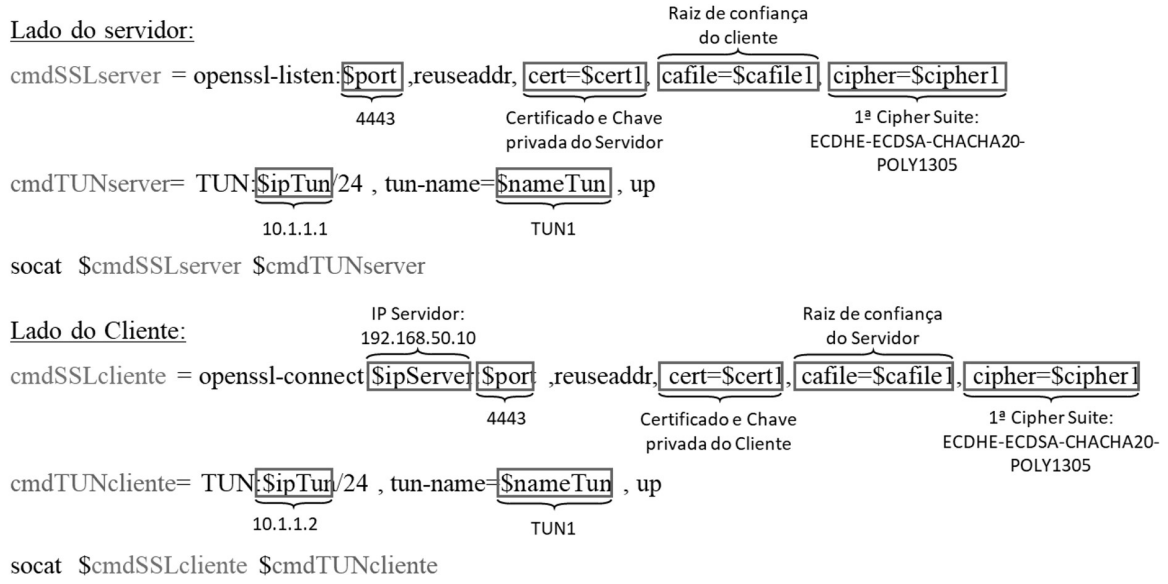


Figura 3.3: Diagrama da ferramenta Socat - 1ªIteração

entre as interfaces TUN1 - 10.1.1.1 (servidor) e TUN1 - 10.1.1.2 (cliente), o qual será encapsulado pelo túnel anterior. Para isso, do lado do cliente, o comando *openssl-connect* irá realizar o pedido de ligação para o endereço IP da interface TUN criada na primeira iteração no lado do servidor (TUN1 - 10.1.1.1 (servidor)). Nesta segunda iteração, o conjunto de cifra utilizado para configurar o túnel TLS, incluindo os certificados do servidor e cliente, utilizam cifras diferentes de modo a conferir ao MultiTLS a tolerância contra vulnerabilidades já mencionada anteriormente. A figura 3.4 representa os comandos utilizados na segunda iteração do MultiTLS.

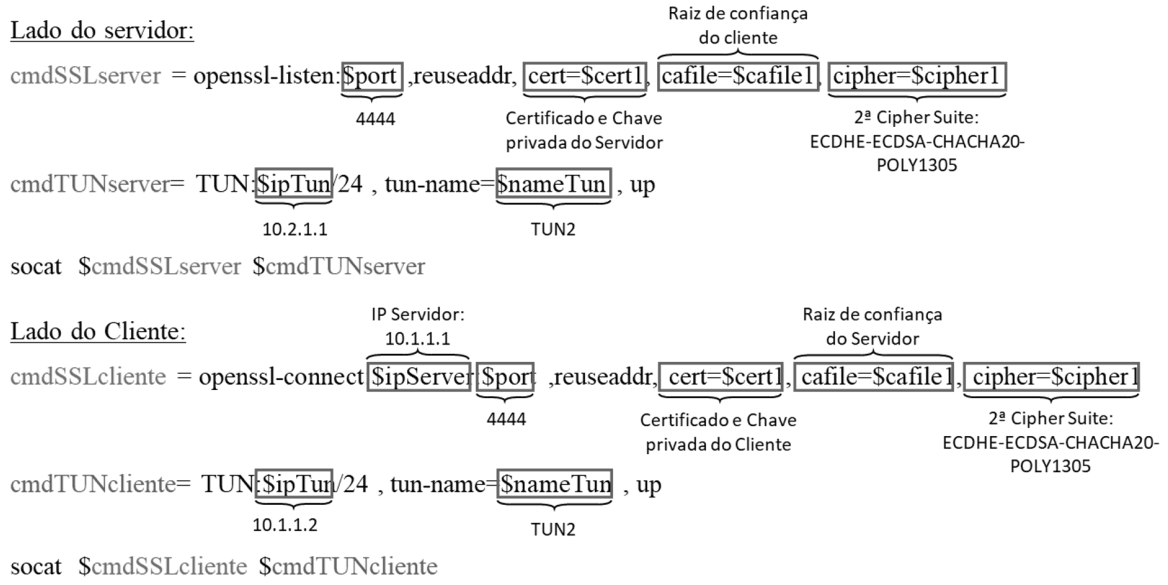


Figura 3.4: Diagrama da ferramenta Socat - 2ª Iteração

3.3 Atualização do MultiTLS

No decorrer deste trabalho, a ferramenta MultiTLS foi alvo de algumas atualizações. Em primeiro lugar foram atualizadas as dependências de forma a que fossem utilizadas as versões mais recentes, mais concretamente foram atualizadas a biblioteca OpenSSL e a ferramenta Socat. As versões podem ser observadas na tabela 3.1.

Bibliotecas/Ferramentas Atualizadas		
Nome	Versão Antiga	Versão Atualizada
OpenSSL	1.1.0g	1.1.1k
Socat	1.7.3.2	1.7.4.1

Tabela 3.1: Bibliotecas/Ferramentas Atualizadas

De seguida, foi também implementado um menu, pelo qual o utilizador passa a ter controlo da ferramenta. Além disso, foram criados também os módulos que serão executados através do menu. Estes módulos possibilitam realizar as ações de instalação e desinstalação das dependências do MultiTLS, bem como optar por iniciar a ferramenta em modo servidor ou cliente. Também ao nível da arquitetura do MultiTLS foram efetuadas algumas alterações. Na versão anterior, era necessário a configuração do porto tanto do lado do cliente como do servidor. Na versão mais recente da ferramenta MultiTLS, todas as configurações passaram a ser da responsabilidade do cliente. Ao iniciar a comunicação, o cliente indica todos os parâmetros da comunicação como:

- IP do servidor - informa a ferramenta do endereço IP que deve contactar;
- Porto - o porto que será utilizado pelo cliente e servidor para estabelecer o primeiro túnel;
- Número de túneis - informa o servidor do número de túneis que pretende estabelecer;
- IP do cliente - informa o servidor qual o endereço de IP que este deve contactar

Através do envio destes parâmetros, o cliente controla todas as configurações necessárias para estabelecer a comunicação. Do outro lado, o servidor, apenas aguarda pelo início de comunicação por parte do cliente. Com as atualizações efetuadas, a ferramenta, ficou mais otimizada, pois utiliza as últimas versões das suas dependências, e também ao nível da sua utilização e configuração, ficou mais intuitiva.

3.4 Conjuntos de cifra

Como referido anteriormente, a grande vantagem do MultiTLS é a sua tolerância a vulnerabilidades, de modo a possuir esta tolerância o MultiTLS utiliza um conjunto de cifra diferente para cada túnel TLS criado. A implementação do MultiTLS permite ao utilizador iniciar uma ligação utilizando até quatro túneis TLS encapsulados, sendo que são utilizados os seguintes conjuntos de cifra baseados no trabalho de Moura R. [26]:

1. TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305 - Neste primeiro conjunto de cifra (o qual estabelece o primeiro túnel), é utilizado o algoritmo *ECDHE* (*Elliptic Curve Diffie-Hellman Ephemeral*) para troca de chaves, o *ECDSA* (*Elliptic Curve Digital Signature Algorithm*) é utilizado como algoritmo de assinatura digital. De seguida, o *CHACHA20* é utilizado para cifrar as mensagens da comunicação e finalmente, o algoritmo *POLY1305* tem a função de autenticação da mensagem;

2. TLS_RSA_WITH_AES_128_CCM8 - O segundo conjunto (utilizado para estabelecer o segundo túnel), utiliza o *RSA* (*Rivest–Shamir–Adleman*) como algoritmo de troca de chaves e assinatura digital. De seguida, as mensagens são cifradas através do algoritmo *AES128* (*Advanced Encryption Standard 128 bits*) e por fim é utilizado o algoritmo *CCM8* para autenticação das mensagens;

3. TLS_DHE_DSS_WITH_CAMELIA_256_SHA256 - O terceiro conjunto de cifra (através do qual é estabelecido o terceiro túnel), utiliza como algoritmo de troca de chaves o *DHE* (*Diffie-Hellman Ephemeral*), o *DSS* (*Digital Signature Standard*) é utilizado na assinatura digital. De seguida, o algoritmo *CAMELIA de 256 bits* trata da cifra de mensagens e, por último, é utilizado o algoritmo *SHA256 (Secure Hash Algorithm 256 bits)* para autenticação de mensagens;

4. TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 - O quarto e último conjunto de cifra (utilizado para estabelecer o quarto túnel), utiliza para troca de chaves *ECDHE* (*Elliptic Curve Diffie-Hellman Ephemeral*), o *RSA* (*Rivest–Shamir–Adleman*) é utilizado para assinatura digital, para cifra de mensagens o algoritmo escolhido foi o *AES256 GCM (Advanced Encryption Standard 256 bits in Galois Counter Mode)* e finalmente, o *SHA384 (Secure Hash Algorithm 384 bits)* é utilizado para autenticação de mensagens;

A tabela 3.2, resume os conjuntos de cifra, utilizados pela ferramenta MultiTLS, os quais estão organizados consoante a sua função.

Conjuntos de Cifra				
	Troca de Chaves	Assinatura Digital	Cifra de Mensagem	Autenticação de Mensagem
1	ECDHE	ECDSA	CHACHA20	POLY1305
2	RSA	RSA	AES128	CCM8
3	DHE	DSS	CAMELIA256	SHA256
4	ECDHE	RSA	AES256-GCM	SHA384

Tabela 3.2: Conjuntos de cifra

3.5 Configurações de ligação

Para estabelecer com sucesso uma comunicação, através da ferramenta MultiTLS, é necessário garantir algumas configurações. Numa primeira fase, o cliente MultiTLS utiliza o porto 4040 para realizar o envio de informação que será utilizada para estabelecer a comunicação cifrada. Esta informação inclui: o seu endereço IP, o número de túneis a considerar na comunicação MultiTLS a estabelecer, o porto no qual essa comunicação será feita e o certificado do cliente utilizado no primeiro túnel. Do lado do servidor, este recebe a informação no porto 4040 e envia o seu certificado para o primeiro túnel. O cliente recebe o certificado do servidor no porto 4040. Devido a estas negociações iniciais, é necessário configurar quaisquer *firewalls* que possam interferir na comunicação, de forma a aceitarem fluxo de dados *inbound* e *outbound* para o porto 4040.

Terminadas as negociações iniciais, a ferramenta pode agora estabelecer k * túneis (em que k é o número de túneis previamente definido pelo utilizador e menor ou igual a quatro). O primeiro túnel é estabelecido através do porto indicado pelo utilizador ao iniciar o programa do lado do cliente. Para os restantes túneis, o número do porto utilizado será incrementado a partir do porto indicado inicialmente pelo utilizador. A título de exemplo, se o utilizador indicar para o primeiro túnel o porto 4444, o segundo túnel será estabelecido no porto 4445, o terceiro no porto 4446 e finalmente o quarto túnel será estabelecido através do porto 4447. Neste caso, derivado do entunelamento feito pela ferramenta MultiTLS, os vários túneis estabelecidos serão encapsulados uns nos outros de forma sucessiva. Deste modo, o primeiro túnel atua como um "túnel exterior", o qual contém encapsulados os restantes túneis. Devido a esta propriedade da ferramenta MultiTLS, apenas é necessário ter em conta que nenhuma *firewall* faça o bloqueio do fluxo de dados para o porto utilizado para estabelecer o primeiro túnel, já que os restantes túneis utilizam a estrutura previamente criada pelo primeiro túnel. É de realçar ainda, que o número indicado pelo utilizador para o primeiro porto deve ser escolhido e configurado ao nível da *firewall* antes de iniciar qualquer comunicação através da ferramenta MultiTLS.

Feitas estas configurações, estão reunidas as condições para estabelecer a comunicação entre cli-

ente e servidor MultiTLS que se encontrem dentro da mesma rede. Porém, caso seja necessário realizar essa comunicação entre redes diferentes, deve-se proceder à configuração de *port forwarding*. O *port forwarding*, é o encaminhamento ou mapeamento de portos, de modo a redirecionar um pedido de comunicação de uma combinação de endereço IP e porto, para outra, enquanto os pacotes atravessam um *gateway* de rede, como um *router* [32]. Assim sendo, é possível através de um *gateway* redirecionar pedidos de ligação vindos de um conjunto *IP:porto* externos para um IP e porto pertencentes a uma máquina na rede interna. No caso da ferramenta MultiTLS é necessário realizar o *port forwarding* do porto 4040 e do porto indicado pelo utilizador, que a título de exemplo considera-se o porto 4444. As configurações devem ser realizadas nos *gateways* situados nas redes que se pretende interligar. A tabela 3.3 assume a existência de duas redes (rede 1 e rede 2) e apresenta a configuração de *port forwarding* efetuada no *gateway* da rede 2.

Endereço IP origem	Porto externo	Endereço IP interno	Porto interno	Protocolo
IP externo rede 1	4040	IP máquina MultiTLS	4040	TCP
IP externo rede 1	4444	IP máquina MultiTLS	4444	TCP

Tabela 3.3: Configurações de encaminhamento de portos

Com o *port forwarding* configurado nos *gateways* das duas redes, o pedido de ligação do cliente situado na rede 1 chega ao *gateway* da rede 2 que, por sua vez, já sabe que uma mensagem vinda daquele endereço IP para aquele porto deve ser entregue à máquina onde se encontra o servidor MultiTLS e vice-versa.
Através destas configurações, será possível estabelecer comunicações seguras através da ferramenta MultiTLS, quer o par cliente/servidor se encontre na mesma rede ou em redes diferentes.

3.6 Manual de utilização

Para utilizar a ferramenta MultiTLS é necessário executar a ferramenta via terminal nos dois pontos que se pretende interligar. Após a execução, o programa apresenta um menu ao utilizador conforme é demonstrado na figura 3.5.

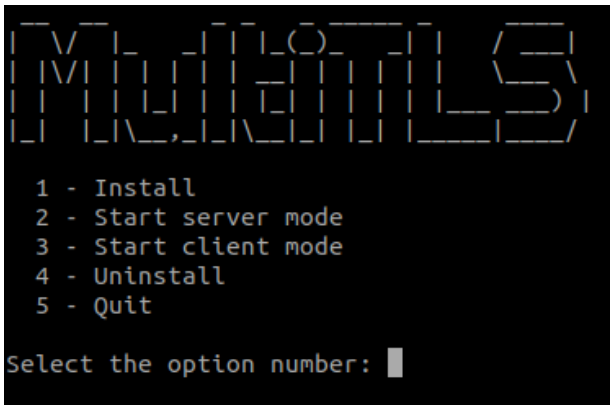


Figura 3.5: Menu inicial da ferramenta MultiTLS

Na primeira execução o utilizador deve selecionar a primeira opção de forma a iniciar a instalação. Neste passo, serão instaladas as dependências da ferramenta como o OpenSSL e o Socat. É necessário realizar este processo em ambos os pontos que se pretende interligar. Ao terminar a fase de instalação o utilizador é novamente redirecionado para o menu anterior. Nesta fase, o utilizador deve iniciar a ferramenta em Modo Servidor (Server Mode) num dos pontos e no outro em Modo Cliente (Client Mode). No ponto que se encontra em Modo Cliente será necessário indicar o endereço de IP do servidor, o porto a utilizar e o número de túneis que se pretende encapsular (este número deverá estar entre 1 e 4). As configurações iniciais são apenas necessárias do lado do cliente, pois este ao estabelecer a ligação com o servidor envia também as informações acerca do porto e do número de túneis selecionados. Do lado do servidor este recebe a informação do cliente e estabelece a ligação com base nas definições recebidas. Assim que o utilizador completa este passo, a ferramenta estabelece os túneis TLS e interliga os dois pontos estabelecendo uma comunicação de segurança reforçada.

3.7 Resumo

Neste capítulo foi apresentada em detalhe a ferramenta MultiTLS. Numa fase inicial, foi apresentada a sua arquitetura, onde foi explicado o funcionamento da ferramenta MultiTLS, bem como as suas dependências e tecnologias utilizadas. De seguida foram apresentados os diferentes conjuntos de cifra utilizados numa comunicação MultiTLS. E finalmente é demonstrado como o utilizador pode configurar facilmente a ferramenta através de um manual de utilização.

Capítulo 4

Estudo de Casos

Nos dias de hoje, são várias as situações onde a solução de segurança passa pela utilização de canais cifrados, estes podem ser utilizados para reforçar o nível de segurança de uma comunicação, ou até mesmo para tornar seguros mecanismos que de outra forma apresentariam um nível de segurança obsoleto.
De forma a demonstrar situações, onde o MultiTLS poderá ser uma ferramenta extremamente útil, foram definidos quatro estudos de caso onde a utilização de uma ferramenta como o MultiTLS pode apresentar vantagens ao nível da segurança:

- Comunicação segura entre áreas de operação;
- Comunicação segura entre duas soluções *cloud*;
- Comunicação segura entre o colaborador e a rede da organização;
- Comunicação segura entre aplicações legadas.

No decorrer deste capítulo, serão apresentados os diferentes estudos de caso considerados, de forma a contextualizar a utilização da ferramenta MultiTLS em cenários práticos.

4.1 Rede-para-Rede

Nesta secção, será mencionado, em primeiro lugar, um estudo de caso da utilização da ferramenta MultiTLS, que será utilizada para garantir a segurança na comunicação entre duas áreas de operação de uma organização. De seguida, será abordado o estudo de caso onde se pretende estabelecer uma comunicação de segurança reforçada entre serviços de computação em nuvem (serviços *cloud*) diferentes.

4.1.1 Comunicação segura entre áreas de operação

Para este estudo de caso será necessário configurar a ferramenta MultiTLS em ambas as áreas de operação, as quais irão funcionar como *reverse proxy*, tal como ilustra a figura 4.1. Nesta situação, seria da responsabilidade da organização realizar a configuração da ferramenta nas duas áreas de operação.

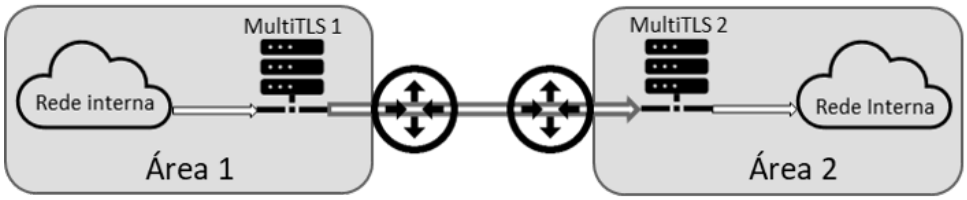


Figura 4.1: Diagrama - Comunicação segura entre áreas de operação

A tabela 4.1 descreve de forma mais concreta o estudo de caso Rede-para-Rede no qual é utilizada a ferramenta MultiTLS. Através da ferramenta MultiTLS, é possível estabelecer uma ligação segura através de múltiplos canais TLS cifrados, os quais reforçam o nível de segurança entre duas áreas de operação de uma organização (ex: edifícios, escritórios, etc). Desta forma, apesar de distanciadas geograficamente, é possível realizar troca de dados de forma segura entre ambas as áreas de operação.

ID:	Rede-para-Rede – Comunicação segura entre áreas de operação
Título:	Comunicação segura entre áreas de operação
Descrição:	Utilizar o MultiTLS como meio de garantir uma comunicação segura entre duas áreas de operação (ex. escritórios), que contenham requisitos de segurança mais rigorosos.
Agente principal:	MultiTLS
Pré-condição	A comunicação entre áreas é feita através dos mecanismos de segurança habituais como um canal TLS estabelecido entre áreas.
Pós-condição:	A comunicação entre áreas é feita com recurso à ferramenta MultiTLS, e, portanto, adicionando um nível mais elevado de segurança.

Descrição dos passos:	1. Área1 e Área2 pretendem que toda a comunicação entre si seja feita de forma mais segura devido à sensibilidade da informação que necessita de ser trocada entre estas áreas. 2. Área1 e Área2 estabelecem através do MultiTLS, múltiplos canais TLS com diferentes tipos de cifra para oferecer segurança com diversidade de mecanismos. 3. O tráfego entre áreas é redirecionado por esses canais estabelecidos pela ferramenta MultiTLS, adicionando assim um forte fator de segurança.
Observações:	Para estabelecer os canais TLS será necessário configurar a ferramenta MultiTLS, bem como as suas dependências.
Cenários de utilização:	Este caso de uso tem por base incrementar o nível de segurança entre áreas de operação em organizações que têm a necessidade de manusear informação sensível e que esta seja transportada virtualmente para outra área de operação (ex. agências governamentais que manuseiam informação classificada). Utilizando o MultiTLS é possível incrementar o nível de segurança, estabelecendo múltiplos canais TLS, de modo a tornar seguro o processo de envio de informação sensível entre áreas, escritórios, edifícios, etc.

Tabela 4.1: Comunicação segura entre áreas de operação

4.1.2 Comunicação segura entre duas soluções na nuvem

A tabela 4.2 descreve outro exemplo Rede-para-Rede, onde a utilização da ferramenta MultiTLS adiciona um forte fator de segurança. Neste estudo de caso, pretende-se reforçar a segurança na comunicação entre duas soluções *cloud*, quer sejam do mesmo fornecedor, ou de fornecedores diferentes. À semelhança do estudo de caso descrito anteriormente, também aqui será necessário configurar uma máquina em cada solução *cloud* que servirá de *reverse proxy* aos restantes ativos. A ferramenta MultiTLS será configurada em ambas as máquinas de modo a estabelecer os canais TLS, pelos quais a comunicação será redirecionada.

A utilização de uma ferramenta como o MultiTLS, adiciona um fator extra de segurança o que viabiliza a existência de um sistema composto por várias soluções *cloud* (*cloud-of-clouds*) com

alto nível de segurança da informação trocada.

ID:	Rede-para-Rede – Comunicação segura entre soluções *cloud*
Título:	Comunicação segura entre duas soluções *cloud*
Descrição:	Utilizar o MultiTLS como meio de garantir uma comunicação segura entre duas soluções *cloud*.
Agente principal:	MultiTLS
Pré-condição	A comunicação entre soluções *cloud* é feita através dos mecanismos de segurança habituais.
Pós-condição:	A comunicação soluções *cloud* é feita com recurso à ferramenta MultiTLS, e, portanto, adicionando um nível mais elevado de segurança.
Descrição dos passos:	1. Pretende-se estabelecer uma comunicação com segurança reforçada entre duas soluções *cloud*. 2. Cloud1 e Cloud2 estabelecem através do MultiTLS, múltiplos canais TLS com diferentes tipos de cifra. 3. O tráfego entre soluções *cloud* é redirecionado através dos canais estabelecidos pela ferramenta MultiTLS, adicionando assim um forte fator de segurança.
Observações:	Para estabelecer os canais TLS será necessário configurar a ferramenta MultiTLS, bem como as suas dependências.
Cenários de utilização:	Este caso de uso tem por base incrementar o nível de segurança na comunicação entre duas soluções *cloud*, quer sejam do mesmo fornecedor ou de fornecedores diferentes. Através da ferramenta MultiTLS é possível estabelecer múltiplos canais TLS, através dos quais será redirecionada a comunicação entre a Cloud1 e a Cloud2. A utilização da ferramenta MultiTLS possibilita assim comunicações entre *clouds* com um nível de segurança reforçado, o que viabiliza a existência de um sistema de *cloud-of-clouds* seguro.

Tabela 4.2: Comunicação segura entre duas soluções na nuvem

4.2 Máquina-para-Rede

No caso, ilustrado na figura 4.2, é apresentada a comunicação segura entre um colaborador, que se encontra fora da rede da organização, e a rede interna da organização. Será da responsabilidade da organização garantir a disponibilidade do serviço (ferramenta MultiTLS em modo servidor), de forma a aceitar pedidos de ligação. Do lado do colaborador, este deve configurar a ferramenta MultiTLS em modo cliente e estabelecer a ligação com o servidor.

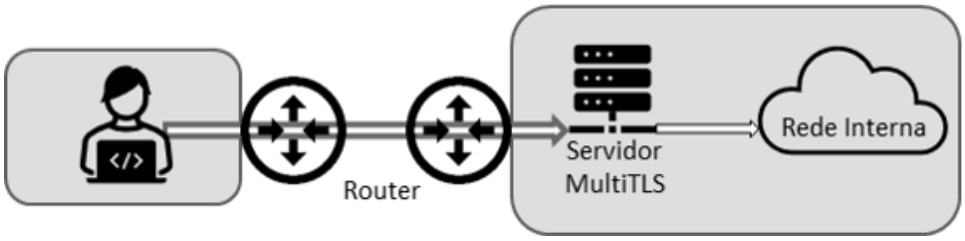


Figura 4.2: Diagrama - Comunicação segura no teletrabalho

O estudo de caso, descrito na tabela 4.3, pretende retratar a situação de um colaborador que se encontra a trabalhar fora do escritório, e portanto, não se encontra protegido pela infraestrutura da sua organização. Porém, devido à sensibilidade da informação trocada, existe a necessidade de reforçar a segurança da comunicação. Através da ferramenta MultiTLS o colaborador pode estabelecer uma ligação segura entre a sua máquina e a rede da sua organização. Para este estudo de caso será necessário configurar o MultiTLS na máquina do colaborador, e do lado da organização, esta deve possuir uma máquina/servidor para aceitar os pedidos de comunicação do MultiTLS.

ID:	Máquina-para-Rede – Teletrabalho seguro
Título:	Comunicação segura entre o colaborador e a rede da organização
Descrição:	Utilizar o MultiTLS como meio de garantir uma comunicação VPN com segurança reforçada entre o colaborador de uma organização, que se encontra a trabalhar fora das instalações, e a rede da organização, mantendo a simplicidade de utilização das aplicações existentes.
Agente principal:	MultiTLS
Pré-condição	A comunicação entre colaborador e rede da organização é feita através de uma ligação VPN.

Pós-condição:	A comunicação entre colaborador e rede da organização é feita através de VPN com segurança reforçada por múltiplos canais TLS estabelecidos pela ferramenta MultiTLS.
Descrição dos passos:	1. O colaborador pretende estabelecer uma comunicação com segurança reforçada por VPN à rede da organização. 2. O colaborador estabelece através do MultiTLS, múltiplos canais TLS com diferentes tipos de cifra que são selecionadas de forma automática e transparente para o utilizador. 3. O tráfego entre o colaborador e a organização é redirecionado por esses canais estabelecidos pela ferramenta MultiTLS, adicionando assim uma segurança reforçada à ligação VPN.
Observações:	Para estabelecer os canais TLS será necessário configurar a ferramenta MultiTLS, bem como as suas dependências.
Cenários de utilização:	Este caso de uso tem por base incrementar o nível de segurança na comunicação de um colaborador com a rede da sua organização. Por vezes derivado das suas atividades ou do papel que desempenha dentro da organização, será necessário que o colaborador tenha uma preocupação extra com o fator segurança. A ferramenta MultiTLS permite adicionar uma segurança reforçada à ligação VPN que por si já estabelece uma ligação segura.

Tabela 4.3: Comunicação segura entre colaborador e organização

Neste estudo de caso, seria necessário existir um mecanismo de controlo de identidade, através do qual os utilizadores se deveriam autenticar. Desse modo, apenas utilizadores que provassem a sua identidade, por meio das suas credenciais, poderiam ter acesso à ferramenta e consequentemente ao conteúdo do outro lado da comunicação. Estes fornecedores de identidade podem ser distinguidos em dois tipos, segundo o seu lugar espacial. Enquanto os *cloud* se configuram em algum lugar da internet (na nuvem), os *on-premises* encontram-se algures na empresa (ou seja, no local). A título exemplificativo existe o caso da *Microsoft*, que dispõe de duas soluções: o *Active Directory*[17] (*on-premises*) e o *Azure Active Directory* [20] (*cloud*). Pegando no exem-

plo da plataforma *cloud* da *Microsoft*, através da integração do *Azure Active Directory* com a plataforma *Microsoft*, para gestão de dispositivos moveis, não só seria possível realizar a distribuição da ferramenta para todos os utilizadores da organização, como seria possível também aplicar mecanismos de segurança como autenticação *multifator* e/ou exigir que para o dispositivo aceder a essa ferramenta passasse por verificações de segurança. A única desvantagem deste método, seria o licenciamento necessário obter para ter acesso a estas ferramentas de controlo de dispositivos e gestão de identidade. Porém, uma outra opção seria integrar a autenticação da ferramenta com o *Active Directory on-premises*, que embora tenha algumas limitações, continua a ser um fornecedor de identidade viável.

4.3 Máquina-para-Máquina

Este caso, ilustrado na figura 4.3, foca-se na comunicação entre aplicações legadas e como tornar segura a sua utilização em ambientes críticos. Para este caso será necessário que ambas as máquinas corram uma instância da ferramenta MultiTLS. Qualquer uma das máquinas poderá correr em modo cliente ou servidor, contudo para que a ligação seja possível as máquinas devem operar em modos diferentes, ou seja, uma em modo servidor e outra em modo cliente.

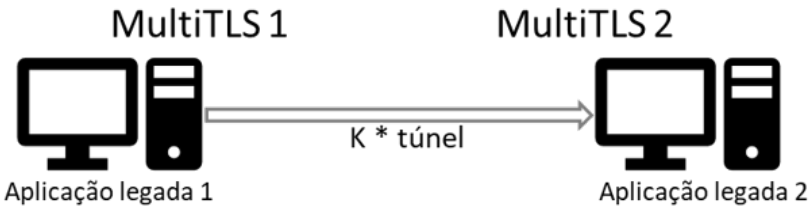


Figura 4.3: Diagrama - Comunicação segura entre aplicações legadas

As aplicações legadas são *softwares*, que derivado da sua idade, foram desenvolvidos utilizando técnicas que nos dias de hoje já se encontram desatualizadas, o que as torna num alvo fácil para os atacantes visto que muitas possuem vulnerabilidades conhecidas. Porém, a utilização deste tipo de *software* continua a desempenhar um papel vital no funcionamento de algumas organizações/empresas. Muitas das vezes, as próprias organizações não sabem como lidar com aplicações legadas. Isto porque, se de um lado têm um papel fundamental na organização, e como tal não podem ser desativadas, por outro a sua atualização/migração para um sistema mais moderno encontra muitas vezes desafios técnicos e não técnicos, desde elevados custos financeiros à falta de profissionais experientes em lidar com tecnologias que nos dias de hoje se encontram desatualizadas [1].

A utilização da ferramenta MultiTLS neste estudo de caso, permite que aplicações legadas com

vulnerabilidades de segurança conhecidas possam ser utilizadas de forma segura.

Para este estudo de caso, será necessário configurar a ferramenta MultiTLS nas máquinas onde estas aplicações que pretendem comunicar se encontram. A utilização do MultiTLS permite que a comunicação seja efetuada de forma segura ao encapsular a mensagem da aplicação legada através de protocolos criptográficos seguros. Um exemplo mais concreto seria a interligação de um servidor aplicacional com uma base de dados que não suporte uma versão do protocolo TLS recente. A tabela 4.4 apresenta uma descrição deste estudo de caso.

ID:	Máquina-para-Máquina – Aplicações legadas
Título:	Comunicação segura entre aplicações legadas
Descrição:	Utilizar o MultiTLS como suporte para fornecer segurança através de canais TLS em aplicações legadas.
Agente principal:	MultiTLS
Pré-condição	Utilização de aplicações legadas contendo vulnerabilidades de segurança conhecidas.
Pós-condição:	Utilização de aplicações legadas de forma segura através do MultiTLS.
Descrição dos passos:	1. Host1 e Host2 pretendem estabelecer uma ligação através de uma aplicação que utiliza protocolos legados (por exemplo, SSL 2.0, 3.0 e TLS 1.0), e, portanto, não segura devido a existirem vulnerabilidades conhecidas. 2. Host1 e Host2 estabelecem através do MultiTLS, múltiplos canais TLS que são apresentados como um túnel. 3. O tráfego entre os dois é redirecionado por esses canais estabelecidos pela ferramenta MultiTLS, tornando assim viável a utilização da aplicação legada.
Observações:	Para estabelecer os canais TLS será necessário configurar a ferramenta MultiTLS, bem como as suas dependências em ambos os hosts.

Cenários de utilização:	Este estudo de caso tem por base a utilização de aplicações legadas e como as tornar seguras para utilização nos dias de hoje. Aplicações legadas ainda existem em utilização por parte de vários setores, como saúde e setor financeiro. É, por isso, de extrema importância que estas aplicações, contendo elas várias falhas graves de segurança, como por exemplo protocolos legados (versões do SSL e as primeiras versões do TLS) sejam protegidas através de métodos criptográficos mais recentes. O MultiTLS é assim uma ferramenta fundamental para este tipo de situações, pois permite a utilização em segurança deste tipo de aplicações, já que toda a comunicação efetuada será encriptada através de múltiplos canais TLS, sem ser necessário alterar o código-fonte destas aplicações.

Tabela 4.4: Comunicação segura entre aplicações legada

4.4 Resumo

Neste capítulo foram apresentados quatro casos onde a solução MultiTLS pode incrementar o nível de segurança. No primeiro caso, *Comunicação segura entre áreas de operação*, é demonstrado como o MultiTLS pode estabelecer uma ligação de segurança reforçada entre duas áreas de operação de uma organização (ex: edifícios, escritórios, fábricas, etc). O segundo caso, *Comunicação segura entre soluções cloud*, demonstra como o MultiTLS viabiliza a existência de um sistema de *cloud-of-clouds* seguro, ou seja, é possível interligar, através do MultiTLS, vários fornecedores ou infraestruturas diferentes. No terceiro caso, *Comunicação segura entre o colaborador e a rede da organização*, é apresentado um cenários no qual um colaborador de uma organização, por se encontrar fora da mesma, utiliza o MultiTLS de forma a aceder aos recursos organizacionais de forma segura. Por último, no quarto caso, *Comunicação segura entre aplicações legadas*, é demonstrado como, através do MultiTLS, é possível transformar uma aplicação legada (e portanto com vulnerabilidades de segurança), numa aplicação segura para uso empresarial.

Para estes estudos de caso, considera-se que a solução poderia passar pela utilização de uma normal VPN TLS, no entanto a propriedade do MultiTLS que o torna tolerante a vulnerabilidades confere uma segurança de nível superior, o que em casos de informação extremamente sensível faz a total diferença. Através deste capítulo, foi possível verificar um conjunto de casos práticos onde a utilização da ferramenta MultiTLS contribui para um aumento significativo do

nível de segurança.

Capítulo 5

Avaliação

De forma a perceber o comportamento da ferramenta MultiTLS num contexto de utilização real foram preparadas e realizadas um conjunto de experiências. Numa primeira fase, será explicado a preparação e realização das experiências, e de seguida será feita a análise e discussão dos resultados obtidos.

5.1 Preparação dos testes

Na realização dos testes foram utilizadas duas máquinas com o sistema operativo Ubuntu 20.04 LTS instalado e situadas na mesma rede. Ambas as máquinas possuem poder computacional e configurações semelhantes, tendo ambas um processador de quatro núcleos e oito *gigabytes* de memoria RAM. Nas máquinas foi configurada a ferramenta MultiTLS (assim como as suas dependências) em modo cliente e servidor respetivamente. Após a instalação das máquinas e da ferramenta MultiTLS em cada *host*, iniciou-se o processo de configuração das ferramentas necessárias à realização dos testes.

Nos testes realizados, foi considerado o estudo de caso, máquina-para-máquina para comunicação segura entre aplicações legadas, descrito na secção 4.3. Neste estudo de caso, é utilizada a ferramenta MultiTLS para tornar viável a utilização de aplicações legadas, e portanto com vulnerabilidades conhecidas, tornando-as assim seguras para utilização num contexto real. Mais concretamente, foi utilizado o FTP (*File Transfer Protocol*) o qual foi apresentado pela primeira vez há cerca de 50 anos [24]. Através da sua arquitetura cliente/servidor, o FTP é capaz de estabelecer uma ligação entre dois pontos, a qual pode ser utilizada para transferir ficheiros e realizar outras operações. Contudo, devido à sua idade, o FTP na sua forma original não possui nenhum tipo de cifra durante a comunicação, sendo essa a razão pela qual é cada vez menos utilizado, especialmente em contexto corporativo. Esta característica que o torna inseguro, faz

ao mesmo tempo com que seja um excelente exemplo de como uma ferramenta como MultiTLS pode ter um papel fundamental, garantindo o processo de cifra de mensagens e tornando o FTP uma alternativa viável.
Antes da realização dos teste, foi então necessário proceder à instalação e configuração do cliente e servidor FTP nas respetivas máquinas. De seguida foram considerados quatro ficheiros de tamanhos distintos:

- 25MFile - ficheiro de 25MB;
- 50MFile - ficheiro de 50MB;
- 75MFile - ficheiro de 75MB;
- 100MFile - ficheiro de 100MB.

De forma a definir o tamanho para os quatro ficheiros, foi tido em conta que os tamanhos de ficheiros escolhidos não deveriam ser nem demasiado pequenos, pois dificultaria o processo de recolha de dados, nem deveriam ter um tamanho demasiado grande, visto que, a ideia seria realizar várias medições, e um ficheiro de grandes dimensões tornaria o processo mais complicado. Após algumas tentativas iniciais, chegou-se à conclusão de que um ficheiro de tamanho 25 MB cumpria com esses dois requisitos. Os tamanhos dos restantes ficheiros, são múltiplos do primeiro valor. Isto resulta num aumento constante do tamanho dos ficheiros o que facilita a análise dos dados.

5.2 Realização dos testes

Os testes realizados consistiram em transferir os diferentes ficheiros (25MB, 50MB, 75MB e 100MB) e medir o tempo de transferência para números de túneis diferentes. Inicialmente, como forma de referência, foi realizado um teste para $k = 0$, ou seja, as medições foram feitas sem que fosse utilizado qualquer túnel, apenas uma comunicação FTP normal. De seguida foi feito o mesmo procedimento para $k = 1$, $k = 2$, $k = 3$ e $k = 4$, em que k representa o número de túneis utilizados pela ferramenta MultiTLS. De modo a minimizar o impacto de eventuais perturbações na rede, todos os resultados obtidos foram recolhidos na mesma altura do dia, em condições semelhantes. Outra vertente que foi tida em conta foi a representatividade dos dados. De forma a garantir que a amostra de dados recolhida era representativa, foi calculado o desvio padrão dos dados obtidos. Após recolhidas as amostras, constatou-se que o desvio padrão para todos os casos considerados era inferior a um segundo. Tendo em conta que a rede, pode por

vezes sofrer algumas perturbações, considera-se esse desvio padrão um valor bastante aceitável. Por fim, foi realizada uma média das medições obtidas de forma a poder ser analisada.

5.3 Resultados

Os resultados obtidos através dos procedimentos descritos na secção 5.2 (Realização dos testes), serão apresentados ao longo da secção 5.3 (Resultados) sobe a forma de tabelas. As tabelas encontram-se divididas pelo número de túneis, sendo apresentadas cinco tabelas (k = 0, k = 1, k = 2, k = 3 e k = 4). Cada tabela tem ainda cinco colunas, a primeira indica a ordem pela qual os dados foram inseridos, e as restantes quatro indicam o valor das medições para os diferentes ficheiros considerados. Por último, cada tabela indica ainda a média dos tempos de transferência de cada ficheiro. As tabelas presentes neste capítulo, representam apenas uma versão resumida das mesmas. As versões completas estão disponíveis no Apêndice A.

Na tabela 5.1, estão representados os dados referentes à transferência dos ficheiros utilizando apenas o FTP (k = 0).

k = 0				
	25MFile	50MFile	75MFile	100MFile
1º	0.08 s	0.20 s	0.24 s	0.31 s
2º	0.08 s	0.18 s	0.24 s	0.33 s
3º	0.07 s	0.15 s	0.24 s	0.26 s
4º	0.08 s	0.15 s	0.23 s	0.44 s
5º	0.07 s	0.14 s	0.23 s	0.28 s
	...	...	...	...
Média	0.08 s	0.17 s	0.22 s	0.33 s
σ	0.01 s	0.02 s	0.01 s	0.06 s

Tabela 5.1: Valores de teste para k = 0

Utilizando apenas o FTP (k = 0), foi possível constatar que o envio dos ficheiros foi praticamente instantâneo. De seguida, foi realizado o mesmo teste, contudo, desta vez para k = 1, ou seja, utilizando a ferramenta MultiTLS com apenas um túnel configurado. Os resultados obtidos podem ser observados na tabela 5.2. Através dos dados observados foi possível verificar que ao configurar uma ligação com apenas um túnel, já é notório um aumento da média dos tempos. Os restantes dados, referentes a k = 2, k = 3 e k = 4, ou seja, a ferramenta MultiTLS configurada com dois, três e quatro túneis encapsulados, podem ser observados nas tabelas 5.3, 5.4 e 5.5 respetivamente.

Na secção 5.4 (Discussão), será feita a análise dos resultados obtidos.

k = 1				
	25MFile	50MFile	75MFile	100MFile
1º	2.90 s	3.78 s	6.93 s	9.86 s
2º	2.25 s	3.63 s	6.63 s	9.81 s
3º	3.25 s	2.86 s	7.50 s	11.90 s
4º	2.58 s	2.87 s	8.77 s	8.77 s
5º	1.53 s	3.59 s	5.44 s	8.09 s
	...	...	...	...
Média	2.87 s	3.30 s	7.39 s	9.54 s
σ	0.96 s	0.88 s	0.96 s	0.92 s

Tabela 5.2: Valores de teste para k = 1

k = 2				
	25MFile	50MFile	75MFile	100MFile
1º	3.00 s	8.38 s	14.41 s	19.16 s
2º	4.14 s	7.28 s	16.09 s	20.38 s
3º	2.07 s	7.63 s	13.92 s	21.38 s
4º	5.81 s	7.04 s	15.78 s	19.98 s
5º	3.99 s	6.16 s	13.24 s	21.12 s
	...	...	...	...
Média	3.41 s	7.70 s	14.46 s	20.26 s
σ	0.81 s	0.94 s	0.90 s	0.95 s

Tabela 5.3: Valores de teste para k = 2

k = 3				
	25MFile	50MFile	75MFile	100MFile
1º	7.61 s	16.95 s	19.42 s	30.48 s
2º	6.81 s	14.89 s	20.39 s	29.51 s
3º	5.93 s	15.27 s	20.05 s	28.43 s
4º	5.48 s	15.54 s	19.51 s	30.76 s
5º	8.13 s	17.71 s	20.05 s	29.78 s
	...	...	...	...
Média	7.20 s	16.00 s	20.28 s	30.17 s
σ	0.95 s	0.93 s	0.93 s	0.96 s

Tabela 5.4: Valores de teste para k = 3

k = 4				
	25MFile	50MFile	75MFile	100MFile
1º	10.95 s	19.91 s	26.68 s	39.35 s
2º	10.44 s	18.54 s	27.11 s	38.83 s
3º	9.07 s	19.66 s	26.31 s	40.08 s
4º	10.13 s	18.91 s	26.54 s	41.02 s
5º	9.00 s	17.71 s	20.05 s	29.78 s
	...	...	...	... s
Média	9.33 s	19.33 s	27.18 s	39.64 s
σ	0.90 s	0.84 s	0.99 s	0.96 s

Tabela 5.5: Valores de teste para k = 4

5.4 Discussão

O gráfico da figura 5.1, foi obtido através dos dados, recolhidos nas experiências realizadas, os quais podem ser observados nas tabelas 5.1, 5.2, 5.3, 5.4 e 5.5 da secção 5.2 (Realização dos testes). Este gráfico, permite fazer uma comparação do tempo envolvido na transferência dos ficheiros, quando utilizados ficheiros de diferentes tamanhos e diferentes números de túneis. Ou seja, através do gráfico da figura 5.1 é possível observar de que forma o número de túneis, utilizados na configuração da ferramenta MultiTLS, e o tamanho dos dados a transferir durante a comunicação, podem afetar o desempenho do MultiTLS. Observando o eixo das ordenadas, é possível verificar que este é representado em segundos (s). Este eixo, representa o tempo de transferência de cada ficheiro. Quanto ao eixo das abcissas, este está dividido pelo tamanho dos ficheiros utilizados no conjunto de experiências (25 MB, 50MB, 75 MB e 100 MB) e consequentemente, para cada ficheiro são apresentados cinco valores diferentes. Estes valores correspondem ao número de túneis utilizados, durante as experiências, para transferir cada ficheiro, sendo que:

- O zero, significa que nenhum túnel foi considerado, e portanto, apenas o FTP foi utilizado na transferência do ficheiro;
- O número um, representa a transferência do ficheiro utilizando a ferramenta MultiTLS com apenas um túnel configurado;
- O número dois significa que foram utilizados 2 túneis durante a transferência do ficheiro;
- No número três o envio do ficheiro é realizado com três túneis configurados na ferramenta MultiTLS;
- Finalmente, no número quatro, é utilizada a ferramenta MultiTLS com quatro túneis configurados.

Analisando o gráfico da figura 5.1, é possível verificar que quanto maior for o tamanho do ficheiro e o número de túneis utilizado, maior será o tempo de transferência. Algo que já era esperado, tendo em conta que não só foi aumentado o tamanho do ficheiro em si como também o *overhead* provocado pela adição de túneis à comunicação. É possível concluir, que o tamanho do ficheiro a transferir e o número de túneis utilizados para proteger a ligação, têm um impacto direto no desempenho da ferramenta MultiTLS.

Através das conclusões retiradas, seria fácil cair no erro de desconsiderar a utilidade de uma ferramenta como o MultiTLS, tendo por base apenas os elevados tempos de transferência para ficheiros de maior tamanho e nos casos onde são utilizados o maior número de túneis. Porém,

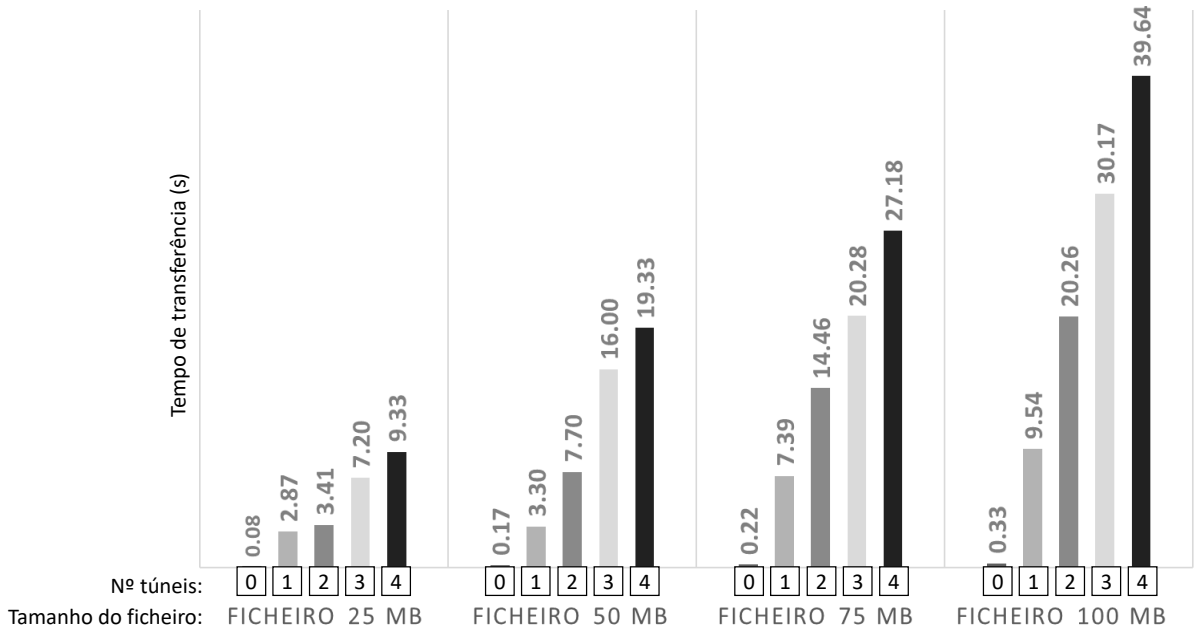


Figura 5.1: Gráfico - Resultados dos testes

é necessário clarificar que tudo depende do caso de uso em questão. Uma ferramenta só será eficaz se o trabalho realizado for adequado à sua utilização. O MultiTLS, sendo uma ferramenta, não é exceção. Se o objetivo é enviar o ficheiro o mais rápido possível, talvez seja melhor optar por outra alternativa. Contudo, o que o MultiTLS oferece, é a capacidade de enviar de forma segura, informação extremamente sensível devido à propriedade que o torna tolerante a falhas. A título de exemplo, uma empresa X pretende enviar o seguinte documento:

- Tipo de ficheiro: .txt;
- Número de caracteres: cerca de 595 000 (equivalente a 100 páginas);
- Tamanho do ficheiro: 0.582 MB;
- Classificação: Altamente Confidencial.

Se um ficheiro de 25 MB nos testes realizados foi transferido numa média de 9.33 segundos quando k = 4, o ficheiro de texto da empresa X, classificado como Altamente Confidencial, e equivalente a 100 páginas (0.582 MB), seria transferido via MultiTLS num período de tempo muito reduzido, considerando condições semelhantes às encontradas quando foi feita a recolha dos dados. Observando este exemplo, é possível concluir que não basta apenas ter em conta o tamanho da informação e o tempo de transferência. Neste caso, é importante também perceber o que ela significa, e qual o impacto caso essa informação seja intercetada por indivíduos não

autorizados. Tendo isso em conta, considera-se que o MultiTLS é uma ferramenta capaz de grande utilidade sempre que o objetivo seja reforçar o fator segurança de uma comunicação, ou até mesmo adicionar esse fator onde em primeira instância não existe.

Adicionalmente, foi calculada a velocidade média de transferência de cada ficheiro para os diferentes números de túneis. Estes valores, presentes na tabela 5.6, foram calculados a partir dos valores médios dos tempos de transferência, apresentados na secção 5.3.

Velocidade de transferência (MB/s)				
	25MFile	50MFile	75MFile	100MFile
k = 0	312.50 MB/s	294.11 MB/s	340.90 MB/s	303.03 MB/s
k = 1	8.71 MB/s	15.15 MB/s	10.14 MB/s	10.48 MB/s
k = 2	7.33 MB/s	6.49 MB/s	5.18 MB/s	4.93 MB/s
k = 3	3.47 MB/s	3.12 MB/s	3.69 MB/s	3.31 MB/s
k = 4	2.67 MB/s	2.58 MB/s	2.89 MB/s	2.52 MB/s

Tabela 5.6: Velocidade de transferência

O gráfico 5.2, criado a partir dos dados da tabela 5.6, permite verificar que existe uma diferença notória na velocidade média calculada, ao ser transferido o ficheiro, sem qualquer tipo de proteção (k = 0), quando comparado com os restantes casos onde foram utilizados túneis para proteger a comunicação (k = 1, k = 2, k = 3 e k = 4). Contudo, é de notar também que embora seja evidente a diferença de desempenho de k = 0 para k = 1, para os restantes casos essa diferença é menos acentuada. Isto significa que, ao serem utilizados os mecanismos para proteger a comunicação, estes têm um forte impacto no desempenho da mesma. Sendo que isto se verifica até para k = 1, que seria equivalente a utilizar apenas o protocolo TLS. Porém, quando são adicionados mais túneis à comunicação (k = 2 , k = 3 e k = 4), embora exista um impacto negativo no desempenho, esse impacto é menos evidente quando comparado com a diferença entre o caso sem segurança (k = 0) e a utilização de um nível de segurança equivalente a utilizar apenas o protocolo TLS (k = 1).

Assim, tem-se que: o tamanho do ficheiro condiciona de forma significativa o desempenho da ferramenta, começando a notar-se um menos bom desempenho à medida que se adicionam túneis; contudo, também se verifica que o simples facto de adicionar encriptação TLS (k = 1) diminui significativamente a velocidade de transferência.

Assim, é possível verificar que existem dois grandes custos: o custo de cifra e o custo que envolve a comunicação, o que se traduz no aumento dos tempos de transferência.

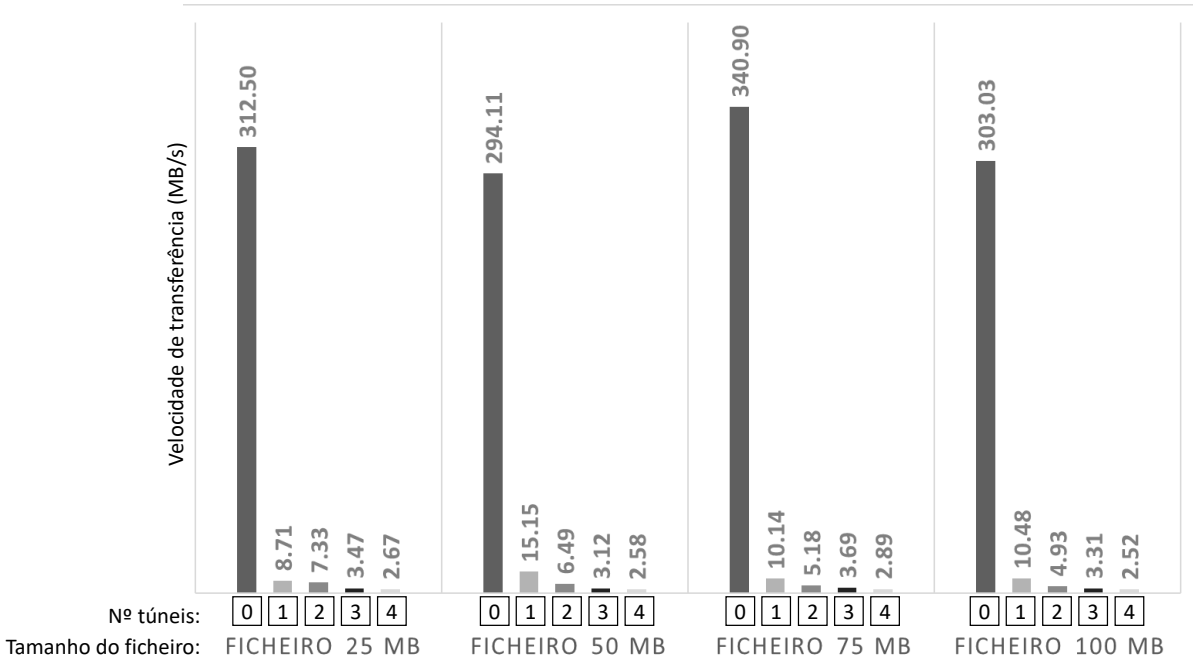


Figura 5.2: Gráfico - Velocidade de transferência (MB/s)

Capítulo 6

Conclusão

Os algoritmos criptográficos, como os utilizados nos protocolos de comunicação, têm um papel fundamental na comunicação em rede. No decorrer desta pesquisa, verificou-se a existência de uma vasta gama de algoritmos criptográficos, uns sendo protocolos legados, e portanto já não tão utilizados e outros mais modernos e utilizados nos dias de hoje. O MultiTLS, sendo uma implementação estendida do TLS, garante uma segurança acrescida, através do encapsulamento de vários túneis TLS, onde se podem utilizar cifras diversas e, deste modo, permitir tolerar ataques que comprometem a segurança de uma cifra mas que não afeta as restantes.

6.1 Contribuições

Ao longo deste trabalho foram apresentados vários algoritmos e conceitos que contribuíram e contribuem, nos dias de hoje, para que seja possível a comunicação *online* de forma segura. Foi igualmente apresentada, a implementação da ferramenta MultiTLS, incluindo a sua arquitetura, as suas dependências e como pode ser facilmente instalada e configurada. De seguida, foi realizada uma exposição de quatro estudos de caso onde a solução de segurança passou por utilizar a ferramenta MultiTLS e, finalmente, foram efetuadas um conjunto de experiências com o objetivo de avaliar o desempenho da ferramenta MultiTLS.

Com base nos resultados e conclusões obtidas, comprovou-se que uma ferramenta como o MultiTLS pode ser uma forma de garantir segurança reforçada em comunicações que assim o exijam, quer para utilização individual ou a nível organizacional. A sua utilidade foi ainda demonstrada por meio da aplicação dos quatro estudos de caso já referidos. Em cada um deles, o MultiTLS revelou que a sua tolerância a vulnerabilidades é uma característica que lhe confere uma segurança mais sólida, e que o evidencia quando comparado com a utilização de uma normal VPN. Adicionalmente, ainda se verificou de que forma o MultiTLS se comporta face a ficheiros de

tamanhos diferentes e diferentes números de túneis, verificando-se que estas duas variáveis influenciam significativamente o desempenho do MultiTLS. Sendo que, quanto maior for o tamanho do ficheiro, maior é o impacto do número de túneis escolhidos, no tempo de transferência. Este aumento do tempo de transferência, quando utilizado um maior número de túneis, deve-se essencialmente aos processos criptográficos envolvidos na criação dos diferentes túneis.

6.2 Trabalho futuro

A ferramenta MultiTLS, é capaz de fornecer segurança reforçada a comunicações que assim o exijam, contudo é de notar que a ferramenta ainda apresenta espaço para evolução.
A utilização do MultiTLS no contexto empresarial exigirá a existência de um mecanismo de autenticação de utilizadores. Para tal ser exequível, seria necessário que a referida ferramenta possuísse um mecanismo de integração que permitisse fornecer a identidade, ou seja, que permitisse extrair de uma base de dados a informação necessária acerca dos utilizadores da empresa. Assim sendo, os utilizadores apenas conseguiriam aceder, através do MultiTLS, aos recursos da empresa por meio de uma autenticação, o que permitiria também a limitação de acesso a certos recursos. Posto isto, seria interessante num trabalho futuro trabalhar sobre estas questões, na medida em que a falta de autenticação se constitui como uma lacuna da ferramenta MultiTLS. Ainda que, como se verificou, o MultiTLS possui diversas vantagens aquando da sua utilização, não prevê na sua raiz a segurança contra utilizadores indesejados. Pelo que seria uma grande consolidação da ferramenta garantir que nem todos os utilizadores que instalem o MultiTLS tenham acesso ao outro lado da comunicação. Para tal acesso, seria necessário provar que o utilizador é quem realmente diz ser e que possui autorização e, portanto, credenciais para aceder a todas as funcionalidades do sistema.
Outro aspeto que seria interessante ser trabalhado é adaptar o MultiTLS de forma a ser compatível com *multi-client*. Na atual versão o MultiTLS apenas permite uma ligação *sigle-client*.
Também ao nível dos conjuntos cifras utilizados a ferramenta apresenta espaço para melhorias. Seria importante trabalhar no sentido de perceber quais as melhores combinações de algoritmos que podem ser utilizadas de forma a reforçar ainda mais o nível de segurança numa comunicação MultiTLS.
Por outro lado, poderia revelar-se interessante e mesmo bastante útil, trabalhar na relação do MultiTLS com os sistemas operativos. De momento, esta ferramenta apenas suporta alguns ambientes *Linux*, deixando de lado sistemas como *Windows*, *Android* e *iOS*, que são sistemas bastante relevantes no mercado mundial.

Bibliografia

[1] Keith H. Bennett. Legacy systems: Coping with success. *IEEE Software*, 12(1):19–23, 1995.

[2] CISCO. What is a VPN? - virtual private network. `https://www.cisco.com/c/en/us/products/security/vpn-endpoint-security-clients/what-is-vpn.html`.

[3] CISCO. Introduction to cisco IPsec technology. `https://www.cisco.com/c/en/us/td/docs/net_mgmt/vpn_solutions_center/2-0/ip_security/provisioning/guide/IPsecPG1.html`, agosto 2007.

[4] Microsoft Corporation. *Understanding Point-to-Point Tunneling Protocol (PPTP)*. Microsoft, 1997.

[5] Ylonen T. e Lonvick C. *RFC4251 - The Secure Shell (SSH) Protocol Architecture*. IETF, janeiro 2006.

[6] Ylonen T. e Lonvick C. *RFC4252 - The Secure Shell (SSH) Authentication Protocol*. IETF, janeiro 2006.

[7] Ylonen T. e Lonvick C. *RFC4253 - The Secure Shell (SSH) Transport Layer Protocol*. IETF, janeiro 2006.

[8] Ylonen T. e Lonvick C. *RFC4254 - The Secure Shell (SSH) Connection Protocol*. IETF, janeiro 2006.

[9] Dierks T. e Rescorla E. *RFC5246-The Transport Layer Security (TLS) Protocol Version 1.2*. IETF, agosto 2008.

[10] Schneier B. Mudge e Wagne D. Cryptanalysis of microsoft's pptp authentication extensions (ms-chapv2). `https://people.eecs.berkeley.edu/~daw/papers/pptpv2.pdf`, outubro 1999.

[11] Patel et al. *RFC3193-Securing L2TP using IPsec*. IETF, novembro 2001.

[12] Townsley et al. *RFC2661-Layer Two Tunneling Protocol "L2TP"*. IETF, agosto 1999.

[13] Google. Ipv6 statistics. `https://www.google.com/intl/en/ipv6/statistics.html`.

[14] Andrew Hodges. *Alan Turing: the Enigma: The Book That Inspired the Film "The Imitation Game"*. Princeton University Press, 2014.

[15] IBM. What is the difference between the ah and esp protocols of IPSec? `https://www.ibm.com/support/pages/what-difference-between-ah-and-esp-protocols-ipsec`, maio 2017.

[16] André Joaquim, Miguel L. Pardal, and Miguel Pupo Correia. Vulnerability-tolerant transport layer security. In *OPODIS*, 2017.

[17] Jaroslav Kadlec, David Jaros, and Radek Kuchta. Implementation of an advanced authentication method within microsoft active directory network services. In *2010 6th International Conference on Wireless and Mobile Communications*, pages 453–456, 2010.

[18] Kaspersky. What is a tunneling protocol? `https://www.kaspersky.com/resource-center/definitions/tunneling-protocol`.

[19] Pawlan M. Cryptography: the ancient art of secret messages. *U. Michigan Press*, fevereiro 1998.

[20] Microsoft. What is azure active directory? `https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-whatis`, julho 2021.

[21] Ricardo Moura, David Matos, Miguel Pardal, and Miguel Correia. *MultiTLS: Secure Communication Channels with Cipher Suite Diversity*, pages 64–77. 09 2020.

[22] Computer Networking Notes. PPP protocol and encapsulation method explained. `https://www.computernetworkingnotes.com/ccna-study-guide/ppp-protocol-and-encapsulation-method-explained.html`, agosto 2018.

[23] Celestino P. Redes virtuais privadas. Master's thesis, Universidade de Taubaté, Brasil, 2005.

[24] Reynolds J. Postel J. *File Transfer Protocol (FTP)*. IETF, outubro 1985.

[25] Cisco Press. IPSec overview part two: Modes and transforms. *Cisco Press*, fevereiro 2002.

[26] Moura R. Multitls: Secure communication channel with diversity. Master's thesis, Instituto Superior Técnico, junho 2018.

[27] Frank Santi Russell and Frank S Russel. *Information gathering in classical Greece / Frank Santi Russell*. University of Michigan press, Ann Arbor, 1999.

[28] Ferreira S. Ipv6 nas redes de sensores sem fio. Master's thesis, Escola Superior de Tecnologia e Gestão de Oliveira do Hospital, 2013.

[29] Melese S. Honeypot system for attacks on ssh protocol. Master's thesis, Andhra University, India, setembro 2016.

[30] Berger T. Analysis of current VPN technologies. Technical report, IEEE, abril 2006.

[31] Damico T. A brief history of cryptography. *Inquiries Journal*, 1(11):1, 2009.

[32] Jha A. Verma N, Kashyap M. *Extending Port Forwarding Concept to IOT*. IEEE, India, 2018.

[33] Weiqing Zhang, Arne Berre, Dumitru Roman, and Hans Huru. Migrating legacy applications to the service cloud. 10 2009.

Apêndice A

Tabelas de resultados

k = 0				
	25MFile	50MFile	75MFile	100MFile
1º	0.08 s	0.20 s	0.24 s	0.31 s
2º	0.08 s	0.18 s	0.24 s	0.33 s
3º	0.07 s	0.15 s	0.24 s	0.26 s
4º	0.08 s	0.15 s	0.23 s	0.44 s
5º	0.07 s	0.14 s	0.23 s	0.28 s
6º	0.08 s	0.14 s	0.23 s	0.36 s
7º	0.08 s	0.16 s	0.26 s	0.28 s
8º	0.08 s	0.17 s	0.22 s	0.32 s
9º	0.09 s	0.17 s	0.24 s	0.36 s
10º	0.15 s	0.18 s	0.24 s	0.33 s
11º	0.10 s	0.15 s	0.20 s	0.28 s
12º	0.12 s	0.14 s	0.21 s	0.28 s
13º	0.09 s	0.22 s	0.21 s	0.28 s
14º	0.08 s	0.24 s	0.21 s	0.43 s
15º	0.07 s	0.17 s	0.21 s	0.26 s
16º	0.07 s	0.19 s	0.21 s	0.46 s
17º	0.08 s	0.18 s	0.22 s	0.28 s
18º	0.08 s	0.22 s	0.22 s	0.45 s
19º	0.07 s	0.18 s	0.21 s	0.27 s
20º	0.08 s	0.14 s	0.20 s	0.28 s
Média	0.08 s	0.17 s	0.22 s	0.33 s

σ	0.01 s	0.02 s	0.01 s	0.06 s

Tabela A.1: Valores para k = 0

k = 1				
	25MFile	50MFile	75MFile	100MFile
1º	2.90 s	3.78 s	6.93 s	9.86 s
2º	2.25 s	3.63 s	6.63 s	9.81 s
3º	3.25 s	2.86 s	7.50 s	11.90 s
4º	2.58 s	2.87 s	8.77 s	8.77 s
5º	1.53 s	3.59 s	5.44 s	8.09 s
6º	1.81 s	3.53 s	7.78 s	9.23 s
7º	1.03 s	3.34 s	8.62 s	9.00 s
8º	2.86 s	3.30 s	7.38 s	9.54 s
9º	4.71 s	2.04 s	6.56 s	9.69 s
10º	1.97 s	2.28 s	6.77 s	9.30 s
11º	1.65 s	3.73 s	7.23 s	10.52 s
12º	3.16 s	4.59 s	6.42 s	9.10 s
13º	2.92 s	3.93 s	7.50 s	9.27 s
14º	2.46 s	2.66 s	6.47 s	10.45 s
15º	4.09 s	3.62 s	9.60 s	8.26 s
16º	4.33 s	2.03 s	8.75 s	9.14 s
17º	3.08 s	3.14 s	7.29 s	8.48 s
18º	2.92 s	2.79 s	7.92 s	11.29 s
19º	3.70 s	2.43 s	7.51 s	9.28 s
20º	4.13 s	5.93 s	6.65 s	9.83 s
Média	2.87 s	3.30 s	7.39 s	9.54 s
σ	0.96 s	0.88 s	0.96 s	0.92 s

Tabela A.2: Valores para k = 1

k = 2				
	25MFile	50MFile	75MFile	100MFile

1º	3.00 s	8.38 s	14.41 s	19.16 s
2º	4.14 s	7.28 s	16.09 s	20.38 s
3º	2.07 s	7.63 s	13.92 s	21.38 s
4º	5.81 s	7.04 s	15.78 s	19.98 s
5º	3.99 s	6.16 s	13.24 s	21.12 s
6º	3.33 s	7.44 s	14.91 s	21.89 s
7º	2.64 s	5.65 s	13.17 s	20.21 s
8º	3.41 s	7.70 s	14.45 s	20.25 s
9º	1.90 s	7.07 s	15.92 s	19.77 s
10º	3.79 s	7.80 s	12.56 s	19.01 s
11º	3.32 s	8.11 s	14.15 s	20.86 s
12º	3.28 s	7.04 s	13.80 s	21.90 s
13º	3.53 s	8.19 s	14.29 s	20.63 s
14º	3.56 s	8.53 s	14.42 s	21.72 s
15º	3.54 s	9.31 s	14.83 s	19.34 s
16º	2.71 s	7.09 s	14.04 s	19.11 s
17º	3.27 s	7.28 s	14.52 s	20.59 s
18º	4.00 s	9.64 s	13.94 s	19.10 s
19º	2.85 s	7.83 s	15.08 s	19.35 s
20º	4.13 s	8.85 s	15.61 s	19.42 s
Média	3.41 s	7.70 s	14.46 s	20.26 s
σ	0.81 s	0.94 s	0.90 s	0.95 s

Tabela A.3: Valores para k = 2

k = 3				
	25MFile	50MFile	75MFile	100MFile
1º	7.61 s	16.95 s	19.42 s	30.48 s
2º	6.81 s	14.89 s	20.39 s	29.51 s
3º	5.93 s	15.27 s	20.05 s	28.43 s
4º	5.48 s	15.54 s	19.51 s	30.76 s
5º	8.13 s	17.71 s	20.05 s	29.78 s
6º	6.81 s	16.64 s	21.17 s	31.41 s

7º	8.98 s	15.97 s	22.81 s	31.54 s
8º	7.19 s	16.00 s	20.28 s	30.17 s
9º	7.89 s	17.81 s	21.22 s	29.46 s
10º	5.48 s	15.01 s	19.79 s	29.52 s
11º	6.92 s	15.89 s	20.47 s	32.47 s
12º	8.01 s	15.24 s	18.25 s	30.82 s
13º	8.03 s	14.54 s	20.91 s	29.74 s
14º	6.54 s	15.55 s	20.63 s	30.17 s
15º	7.36 s	17.62 s	19.40 s	28.86 s
16º	7.90 s	15.24 s	20.54 s	30.25 s
17º	6.22 s	15.81 s	19.84 s	30.52 s
18º	7.94 s	16.48 s	19.72 s	28.63 s
19º	6.38 s	16.50 s	19.76 s	30.32 s
20º	8.34 s	15.45 s	21.48 s	30.63 s
Média	7.20 s	16.00 s	20.28 s	30.17 s
σ	0.95 s	0.93 s	0.93 s	0.96 s

Tabela A.4: Valores para k = 3

k = 4				
	25MFile	50MFile	75MFile	100MFile
1º	10.95 s	19.91 s	26.68 s	39.35 s
2º	10.44 s	18.54 s	27.11 s	38.83 s
3º	9.07 s	19.66 s	26.31 s	40.08 s
4º	10.13 s	18.91 s	26.54 s	41.02 s
5º	9.00 s	17.71 s	20.05 s	29.78 s
6º	8.98 s	18.61 s	27.10 s	39.70 s
7º	9.06 s	18.17 s	28.48 s	40.14 s
8º	8.21 s	18.54 s	28.32 s	39.34 s
9º	8.88 s	18.76 s	27.52 s	38.22 s
10º	11.76 s	18.30 s	25.70 s	38.06 s
11º	8.94 s	18.20 s	27.51 s	39.37 s
12º	8.93 s	20.84 s	27.65 s	41.17 s
13º	8.99 s	19.86 s	28.75 s	39.22 s

14º	9.79 s	18.94 s	28.86 s	38.57 s
15º	9.09 s	20.79 s	26.71 s	39.92 s
16º	8.22 s	18.59 s	26.16 s	41.35 s
17º	8.99 s	19.67 s	26.49 s	39.72 s
18º	8.58 s	20.36 s	28.09 s	39.32 s
19º	8.31 s	19.98 s	26.93 s	38.75 s
20º	10.10 s	19.76 s	25.09 s	41.44 s
Média	9.33 s	19.33 s	27.18 s	39.64 s
σ	0.90 s	0.84 s	0.99 s	0.96 s

Tabela A.5: Valores para k = 4

Apêndice B

Readme

MultiTLS

O MultiTLS é um midlleware, com vários níveis de proteção, que oferece a possibilidade de estabelecer uma comunicação segura entre dois pontos. Para isso, o MultiTLS utiliza o encapsulamento de vários túneis TLS.

Informação Geral

O MultiTLS tem como objetivo oferecer uma forma de comunicação de segurança reforçada, para tal utiliza o encapsulamento de vários túneis TLS. Porém, é de realçar que o MultiTLS não se limita apenas a encapsular vários túneis TLS idênticos. Através do MultiTLS a comunicação é protegida por várias cifras diferentes (uma para cada túnel TLS), reforçando o nível de segurança da comunicação. A ideia é obrigar o adversário a ter de quebrar um conjunto de diferentes cifras se quiser quebrar as propriedades da confidencialidade e integridade. Ou seja, caso um atacante encontre uma vulnerabilidade num dos algoritmos criptográficos utilizados na implementação do TLS, e consiga decifrar o primeiro túnel, a probabilidade de encontrar uma vulnerabilidade contida nos conjuntos de cifra utilizados nos restantes túneis será muito reduzida. Utilizando o MultiTLS para proteger uma comunicação, onde serão estabelecidos k * túneis TLS, em que $k > 1$, caso k-1 túneis TLS sejam comprometidos a comunicação permanece segura. Outra vantagem do MultiTLS é a capacidade de ser utilizado para tornar segura uma comunicação, independentemente da aplicação/programa que está a ser utilizado, já que toda a comunicação efetuada será cifrada através de múltiplos túneis TLS, sem ser necessário alterar o código-fonte destas aplicações/programas.

Dependências

O MultiTLS utiliza como dependências o Socat e o OpenSSL, os quais podem ser instaladas a partir da interface da ferramenta.

- OpenSSL - openssl-1.1.1k
- Socat - socat-1.7.4.1

Testar Ferramenta

Estas instruções demonstram de que forma poderá utilizar a ferramenta MultiTLS num ambiente para fins de desenvolvimento e testes. Consulte a aba Cenário Real para notas sobre como implementar o projeto em casos reais em que o servidor e cliente podem estar na mesma rede ou em redes diferentes.

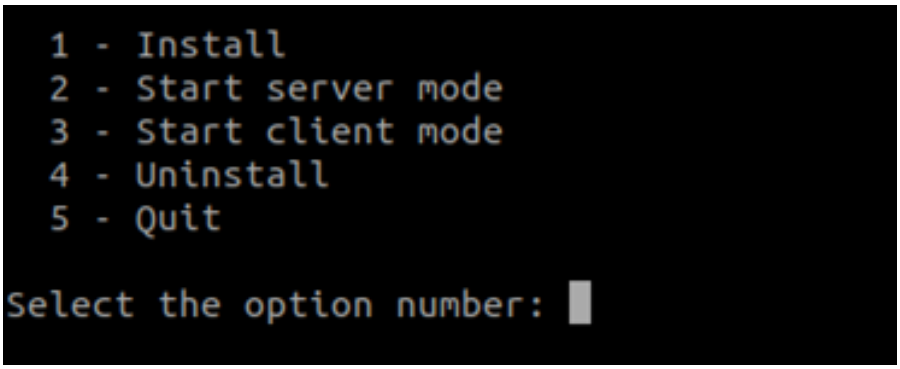
Pré-requisitos

Para correr a ferramenta é necessário utilizar como sistema operativo uma distribuição Linux, contudo a ferramenta foi implementada e testada apenas para a versão Ubuntu 20.04 LTS. Neste cenário de teste serão utilizadas duas máquinas virtuais com Ubuntu 20.04 LTS como sistema operativo, as quais serão referidas como Cliente e Servidor. Em ambas as máquinas, faça o download da ferramenta e navegue até à pasta onde se encontra via terminal. De seguida execute a ferramenta através do seguinte comando:

```
sudo ./multiTLS_init
```

Instalação

Ao executar o comando anterior, a ferramenta é iniciada apresentando ao utilizador o menu inicial, tal como na imagem seguinte:



No menu, caso não tenha efetuado a instalação das dependências de forma manual, selecione a opção: 1-Install. O OpensSSL e o Socat serão instalados automaticamente. Após este processo termine a execução da ferramenta.

Teste

Após a instalação verifique se as dependências foram corretamente instaladas. Para isso execute os seguintes comandos e verifique o resultado.

Opens SSL:

```
openssl version
```

Socat:

```
socat -V
```

O resultado dos comandos deve ser igual às versões instaladas. Caso tenha instalado as dependências de forma automática através da ferramenta, estas devem corresponder às versões indicadas na aba Dependências.

Demonstração

Concluídos os passos de instalação, pode agora iniciar a ferramenta MultiTLS em ambas as máquinas. No menu da ferramenta, execute a ferramenta em modo servidor numa máquina e na outra em modo cliente. Em ambas as máquinas, a ferramenta irá pedir que seja configurado o primeiro certificado. Ao configurar o certificado, no campo Common Name, insira corretamente o nome/endereço IP da máquina.
Do lado do cliente, após criar o certificado, será necessário indicar: o seu endereço IP, endereço IP do servidor ao qual se vai ligar, a porta utilizada no primeiro túnel e o número de túneis utilizados na comunicação (entre 1 e 4). De seguida, as configurações da comunicação são enviadas para o servidor e os certificados do cliente e servidor são trocados.
Após estes passos, são criados os túneis entre cliente e servidor, resultando numa comunicação de segurança reforçada.

Cenário real

Num cenário real, deve garantir que cliente e servidor conseguem comunicar através da porta 4040 (utilizada na comunicação inicial e troca de certificados) e a porta indicada do lado do cliente quando iniciada a ferramenta (utilizada para estabelecer o primeiro túnel). Para cenários em que, o cliente e servidor não se encontrem na mesma rede, será necessário também configurar o port forwarding para as portas anteriormente referidas.