Bernardo Alexandre dos Santos Costa Leite

Simulação e deteção de ciberataques

Dissertação no âmbito do Mestrado em Engenharia Eletrotécnica e de Computadores, ramo de Computadores orientada pelo Professor Doutor Tony Richard de Oliveira de Almeida apresentada ao Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologias da Universidade de Coimbra.

Setembro de 2023

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Simulação e deteção de ciberataques

Bernardo Alexandre dos Santos Costa Leite

Coimbra, Setembro 2023

Simulação e deteção de ciberataques

Supervisor:

Prof. Doutor Tony Richard de Oliveira de Almeida

Jury:

Prof. Doutor Jorge Miguel Sá Silva

Prof. Doutor Jorge Nuno de Almeida e Sousa Almada Lobo

Prof. Doutor Tony Richard de Oliveira de Almeida

Dissertation submitted in partial fulfillment for the degree of Master of Science in Electrical and Computer Engineering.

Coimbra, Setembro 2023

Agradecimentos

Primeiramente, expresso o meu profundo agradecimento ao Professor Doutor Tony Richard de Oliveira de Almeida, não só pela disponibilidade, pela orientação durante esta etapa, aconselhando-me sempre da melhor maneira possível, mas também por ser a imagem de uma docência que em tudo dignifica o Departamento de Engenharia Eletrotécnica e de Computadores e a própria Universidade de Coimbra.

Aos membros da família *eletrões*, na qual destaco o João Castilho, Francisco Santos (e seus respetivos afilhados), Tiago Carvalho, Diogo Soares (e seu respetivo afilhado Guilherme Lins), Isidro Ribeiro, Duarte Cruz (e seu respetivo afilhado), Roberto Pereira, Sergio Vaz e o Gonçalo Arsénio (a ordem define o grau de admiração) pela ajuda, de forma direta ou indireta, ao longo de todo percurso académico e pelos vários jantares e momentos incríveis que certamente nunca serão esquecidos. A vocês um enorme obrigado!

Ao meu chinês preferido (e também o único, diga-se de passagem) Ângelo Huang, pelos trabalhos realizados durante o mestrado, e por ser o meu grande companheiro nesta fase final do curso.

Ao Afonso Carvalho, por todo o companheirismo, por sempre estar ao meu lado durante toda esta fase, por me dar um grande neto (de seu nome Luis Ventura) e acima de tudo, por ser um excelente afilhado (salvo algumas exceções) e amigo que Coimbra me deu. Espero que consigas alcançar todos os teus objetivos ao longo da tua vida, pois la estarei para festejar contigo.

Ao André Araújo, por ser o meu fiel companheiro em 90% dos trabalhos realizados durante esta jornada, por ser o melhor colega de quarto, por sempre me auxiliar nas fases mais críticas, por todos os conselhos e ensinamentos transmitidos, por me ter incentivado a nunca desistir, por ser o exemplo do que uma verdadeira amizade significa e acima de tudo, por ser o melhor afilhado (a par do seu irmão) que alguma vez podia ter imaginado. Sem dúvida que és das pessoas mais trabalhadoras e empenhadas que conheço, e certamente terás um futuro cheio de sucesso e alegrias.

Por último e não menos importante, expresso o meu profundo agradecimento à Dr. Regina Santos e ao Eng. António Leite, por sempre acreditarem em mim mesmo quando eu próprio não acreditava. Sem dúvida foram um dos pilares fundamentais para a conclusão desta etapa, da qual estarei sempre grato.

Resumo

A constante dependência da *Internet*, fez com que houvesse um elevado desenvolvimento tecnológico. A maioria das organizações governamentais e empresariais é forçada a usar tecnologias de redes modernas e flexíveis para proporcionar uma maior interconexão. Contudo, isso também abre portas para um vasto leque de ciberataques com a finalidade de sabotar e interromper os recursos do sistema que é alvo desses ataques. Existem vários motivos por detrás desses ciberataques, como roubo de informações financeiras e informações confidenciais, interrupção dos serviços disponíveis para utilizadores legítimos e obtenção de acesso não autorizado. Devido a essas razões, ferramentas de simulação de redes de computadores são sem dúvida uma aposta a ter em conta, pois permitem simular e detetar num ambiente virtual, diferentes tipos de ameaças.

No âmbito desta dissertação, os trabalhos passaram por explorar a ferramenta de simulação *Graphical Network Simulator 3* com o intuito de simular e detetar várias tentativas de ciberataques. Para tal, foi desenvolvido, num ambiente virtual totalmente controlado, um modelo adaptado para simular ciberataques que ocorrem com alguma frequência no nosso quotidiano. Foram simuladas as situações de ataque do tipo DDoS e *Man in The Middle*, e dos resultados obtidos foi possível avaliar o comportamento da rede em análise. Os resultados mostram de forma evidente e clara que a rede teve os seus serviços e recursos comprometidos devido aos danos provocados na comunicação e integridade da rede. Com este trabalho, foi possível verificar a potencialidade desta ferramenta para análise simulada do comportamento de uma rede em condições de ciberataque.

Palavras-chave: Ciberataques; Cibersegurança; Simulação de Redes de Computadores; GNS3, Ambiente Virtual; DDoS; MITM;

Abstract

The constant dependence on the Internet has resulted in a huge amount of technological development. Most government and business organisations are forced to use modern and flexible network technologies to provide greater interconnectivity. However, this also opens the door to a wide range of cyberattacks aimed at sabotaging and disrupting the resources of the targeted system. There are several reasons behind these cyberattacks, such as stealing financial and confidential information, interrupting the services available to legitimate users and obtaining unauthorised access. For these reasons, computer network simulation tools are definitely worth considering, as they allow different types of attacks to be simulated and detected in a virtual environment.

Within the scope of this dissertation, the work has involved exploring the Graphical Network Simulator 3 simulation tool in order to simulate and detect various cyberattack attempts. To this end, an adapted model was developed in a fully controlled virtual environment to simulate cyberattacks that occur with some frequency in our daily lives. DDoS and Man in the Middle type attacks were simulated. From the results obtained, it was possible to evaluate the behaviour of the network under analysis. The results clearly show that the network had its services and resources compromised due to the damage caused to communication and network integrity. With this work, it was possible to verify the potential of this tool for simulated analysis of the behaviour of a network under cyberattack conditions.

Keywords: Cyberattack; Cybersecurity; Computer Network Simulation; GNS3; Virtual Environment; DDoS; MITM;

"If you don't communicate your ideas, you can just as well do Sudoku."

— Bjarne Stroustrup

Conteúdo

Lista de Acrónimos

DoS	Denial of Service
DDoS	Distributed Denial of Service
USB	Universal Serial Bus
IDS	Intrusion Detection system
IA	Inteligência Artificial
SVM	Support Vector Machine
MITM	Man In The Middle
SSL	Secure Sockets Layer
TLS	Transport Layer Security
BGP	Border Gateway Protocol
PCA	Principal Component Analysis
DSR	Dynamic Source Routing
GNS3	Graphical Network Simulator 3
ARP	Address Resolution Protocol
MAC	Media Access Control
IP	Internet Protocol
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
VLAN	Virtual Local Area Network

UDP	User Datagram Protocol
TCP	Transmission Control Protocol
NAT	Network Address Translation
HSRP	Hot Standby Routing Protocol
OSPF	Open Shortest Path First
ABR	Area Border Router
CAM	Content Addressable Memory

Lista de Figuras

1 Introdução

1.1 Motivação e Contexto

Nos últimos anos, as redes de computadores evoluíram de um mero meio de comunicação para uma infraestrutura computacional imprescindível na nossa sociedade. As redes de computadores tornaram-se maiores, mais rápidas e altamente dinâmicas. O seu uso generalizado nas mais diversas áreas do nosso quotidiano, transformou a questão da cibersegurança num problema constante, e de elevada importância. As tendências dos últimos anos, colocaram a cibersegurança como uma das áreas de atenção prioritária para os governos e empresas. A crescente interconexão entre governos, empresas e utilizadores particulares é, por um lado, uma fonte de oportunidades para a melhoria da sociedade e o aumento do bem-estar social, mas, por outro, a origem de importantes desafios, sobretudo no que toca à segurança informática. O crescimento dos riscos derivados das ameaças à segurança da informação está vinculado ao maior impacto dos ataques e ao aumento da probabilidade de ocorrência, que são consequência de uma maior exposição da sociedade aos dispositivos digitais. Os riscos associados às falhas de cibersegurança continuam a ser das principais categorias de risco a nível económico e empresarial, a curto e médio prazo, quer à escala nacional, quer à escala global.

Em relação ao contexto nacional, o CERT.PT (Computer Emergency Response Team), um serviço integrante do Centro Nacional de Cibersegurança, tem registado um aumento contínuo de incidentes desde 2015, tendo verificado um crescimento de 26% de casos em 2021 face ao ano anterior, que corresponde ao registo de 1418 incidentes em 2020 e 1781 em 2021 [1]. Relativamente ao panorama internacional atual, marcado pelo conflito na Ucrânia, este vem substituir a pandemia da Covid-19 enquanto temática que cria dinâmicas de escala. Pois, se por um lado, a pandemia proporcionou condições de contexto para ciberataques com foco na captura de dados sensíveis, realização de burlas e práticas de extorsão, a conjuntura geopolítica atual demonstra-se propícia a ameaças relacionadas com a ciberespionagem e a

negação de serviço distribuída (DDoS).

Em virtude da situação atual, ferramentas de simulação de redes de computadores são sem dúvida uma aposta a ter em conta. Através delas é possível simular o comportamento de uma rede e os seus componentes, permitindo testar, num ambiente virtual, diferentes configurações e algoritmos de estrutura de segurança centrada na rede para avaliar e comparar a eficiência da deteção de ameaças e os eventuais custos operacionais. Desta forma, uma projeção de uma estrutura de segurança pode ser devidamente testada num ambiente de simulação antes das unidades de deteção reais serem implantadas fisicamente na rede.

1.2 Objetivos

O principal objetivo deste trabalho consiste fundamentalmente em explorar uma ferramenta de simulação de redes de computadores já existente, neste caso a ferramenta escolhida foi o *Graphical Network Simulator 3*, onde se irá desenvolver uma topologia de rede devidamente adaptava para esse fim. O foco deste trabalho será então, simular e detetar várias tentativas de ciberataques num ambiente de simulação totalmente controlado. A análise do impacto dos ciberataques simulados, também se enquadra no propósito deste trabalho.

1.3 Estrutura do Documento

Este documento está dividido em seis grandes capítulos. O primeiro, e atual, é o capítulo introdutório, na qual o tema é enquadrado no contexto atual e são apresentados os objetivos do trabalho a desenvolver. No capítulo 2 é apresentado e analisado o conceito de ciberespaço, os diferentes ciberataques e os seus respetivos mecanismos de deteção. O capítulo seguinte, capitulo 3, aborda o simulador de redes de computadores utilizado para realização do trabalho. O capítulo 4, descreve os métodos usados para o desenvolvimento do modelo, desde da criação das diferentes redes aos protocolos implementados. A simulação dos diferentes ciberataques, bem como os impactos causados na rede, é demonstrado no capítulo 5. Por fim, o capítulo 6 apresenta as conclusões obtidas sobre a dissertação realizada, sendo propostas algumas sugestões para trabalhos futuros.

2 Ciberataques e mecanismos de deteção

Neste capitulo, é feito um enquadramento do conceito de ciberespaço, onde é analisado as diferentes camadas que lhe constituem. Além disso, é examinado os diferentes tipos de ciberataques de modo a dar a compreender a sua metodologia. No final, é apresentado diferentes mecanismos deteção que ajudam a identificar os vários ciberataques.

2.1 Ciberespaço

O ciberespaço alcançou um crescimento avassalador ao longo das últimas décadas e globalmente falando está presente no quotidiano de todas as pessoas, coisa que nos primórdios da existência da *Internet* ninguém poderia antever o alcance que iria ter [2]. Em contrapartida, podemos identificar um número idêntico de vulnerabilidades, relacionadas, sobretudo, com a dependência criada nas pessoas em torno da *Internet*, muito por culpa da excessiva utilização dos meios tecnológicos, potenciando o surgimento de novas ameaças e perigos, que possibilitam a exploração das vulnerabilidades dos sistemas informáticos.

Segundo David Clark [3], o ciberespaço refere-se ao ambiente virtual criado por redes de computadores, incluindo a *Internet*, onde as informações e os dados são armazenados, processados e transmitidos para permitir a comunicação e interação entre pessoas e os dados armazenados. Contundo, a definição de ciberespaço não está limitada apenas a uma definição, pois o constante desenvolvimento tecnológico deu azos a novos conceitos. Desta forma, ao não estar restrito a uma única definição, permite uma maior flexibilidade para descrevê-lo à medida que a tecnologia se desenvolve e os requisitos evoluem. Assim sendo, Adrian Venables [4] propôs um novo modelo de ciberespaço com o intuito de dar a compreender a sua composição, propriedades, riscos, ameaças e os seus requisitos de segurança. Esse novo modelo, apresentado na figura 2.1, apresenta uma nova forma de representar o ciberespaço

para permitir que todos os aspetos sejam examinados. Em seguida, é efetuada uma breve explicação de cada camada do ciberespaço.

Figura 2.1: Modelo de oito camadas do ciberespaço.

- **Camada Geográfica**: Esta camada foca-se no ambiente do mundo real, ou seja, no local onde a infraestrutura do ciberespaço e os utilizadores estão posicionados. A geografia tem um papel significativo porque afeta os meios pelos quais as redes são formadas e as suas características de propagação, que são fundamentais para as propriedades do ciberespaço. Distinguir a área geográfica é vital quando se trata de segurança, pois aspetos políticos, variações de terreno e limitações de infraestrutura podem afetar a transmissão de dados e a suscetibilidade a ataques de DoS.

- **Camada Serviços**: Esta camada está compreendida entre as funcionalidades necessárias para manter a infraestrutura cibernética, incluindo fontes de alimentação, ar condicionado e a segurança dos edifícios que alojam computadores, servidores e componentes de redes. As vulnerabilidades desta camada podem torná-la um alvo atraente, especialmente devido aos efeitos posteriores aos ataques físicos que são facilmente visíveis e avaliados. A camada serviços está mais exposta a ataques físicos e é essencial considerar a resiliência da camada ao avaliar a segurança geral.

- **Camada Infraestruturas**: A camada infraestrutura do ciberespaço, alberga os componentes físicos que armazenam, processam e transferem dados. Esta camada abrange os utilizadores, servidores, componentes de rede e outros elementos fundamentais para a operação do ciberespaço. A camada de infraestrutura é amplamente dispersa e pertence a diferentes organizações, tanto privadas como governamentais. A segurança

desta camada envolve principalmente a proteção de componentes contra destruição física ou roubo.

- **Camada Física**: Esta camada, inclui os componentes de *hardware* suscetíveis a ataques e destruição física. A camada incorpora recursos que são regidos pelas leis da física e descreve as propriedades e técnicas que "dão vida" à camada de infraestrutura, permitindo que os dados sejam trocados entre os sistemas. As características do ciberespaço numa região distinta dependem do meio de transmissão, da frequência e da potência utilizada, o que pode afetar as taxas e a velocidade de transferência de dados. O espectro de radiofrequência é um ambiente congestionado com bandas de frequência utilizadas para múltiplos propósitos e é estritamente regulamentado a nível nacional e internacional. A segurança da camada física depende dos componentes utilizados e do método de transmissão. As comunicações sem fio são vulneráveis a ataques, interferência, falsificação e roubo de dados, sendo os dados não encriptados particularmente vulneráveis aos dois últimos métodos.

- **Camada Sintática**: A camada sintática, contem os protocolos que permitem a comunicação dentro da camada infraestrutura. Existe um grande número de protocolos utilizados na comunicação e troca de dados, sendo que o papel deles é designado pelo modelo OSI (*Open Systems Interconnection*). Cada camada no modelo tem implicações de segurança e pode revelar o uso de protocolos mais antigos e menos seguros, a atualidade dos componentes de $software$ e a eficiência dos algoritmos de roteamento de rede. Sistemas operacionais não suportados podem introduzir vulnerabilidades que podem ser exploradas.

- **Camada Semântica**: A camada semântica envolve sistemas de controle de $software$ que interagem com humanos por vários meios, como comandos de voz ou através de um *display*. No entanto, essa camada é vulnerável à exploração por invasores que buscam manipular os dados ou o comportamento do utilizador. A camada semântica é particularmente suscetível a ataques porque foi projetada para a interação humana.

- **Camada Humana**: A camada humana no ciberespaço tem um papel crucial para entender como os utilizadores interagem com a tecnologia e interpretam os dados. Porem, também representa uma grande ameaça à segurança devido à imprevisibilidade do comportamento humano. A educação e a supervisão nesta camada são importantes para evitar ataques bem sucedidos, bem como para desenvolver interfaces mais intuiti-

vas. Com a constante interação da tecnologia com as pessoas, a fronteira entre humanos e ciberespaço está-se a tornar cada vez mais ténue, levando a possíveis avanços, como dispositivos inseridos no corpo dos utilizadores que permitem interagir diretamente com eles. Compreender a natureza dos relacionamentos e do comportamento humano também é importante para projetar aplicações eficazes.

- **Camada Missão**: A última camada é a camada missão que controla o relacionamento que os humanos e os dispositivos automatizados têm com o ciberespaço. Isso demonstra novamente a natureza artificial do ciberespaço e que o meio foi projetado e construído para cumprir um propósito. Compreendendo a camada de missão, as outras camadas podem ser contextualizadas e os requisitos gerais de segurança podem ser compreendidos e formulados. Cada camada tem um papel, e existe uma dependência entre elas.

2.2 Ciberataques

Os ciberataques são uma ameaça constante a qualquer sistema e como tal, estes também acompanham as evoluções tecnológicas, tornando-se cada vez mais diversificados. Devido a essa variedade, torna-se necessário categorizá-los para podermos analisá-los de forma mais eficiente. No entanto, importa referir que a definição dada aos diferentes tipos de ciberataques nem sempre é absoluta, uma vez que algumas variantes exibem comportamentos insólitos.

Nesta secção, serão apresentados e analisados diferentes tipos de ciberataques que podem atingir um sistema. A figura 2.2, mostra a onde alguns ciberataques podem ser despoletados na rede, e a direção destes ataques.

2.2.1 Negação de serviço / Negação de serviço distribuída

O ataque de negação de serviço, designado na literatura inglesa por DoS (*Denial of Service*), tem origem quando o invasor pretende restringir ou impedir o acesso dos utilizadores autorizados aos recursos disponíveis, gerando um elevado número de falsas solicitações de maneira a deixar a rede, servidor ou o sistema sobrelotado. O ataque DoS, como demonstrado na figura 2.3 é gerado a partir de uma única fonte. Por outro lado, o ataque de negação de serviço distribuída, designado na literatura inglesas por DDoS (*Distributed Denial of Service*), como o próprio nome indica, é gerado a partir de inúmeras fontes desconhecidas, onde o invasor inicialmente identifica as vulnerabilidades na rede para instalar *malwares* em várias

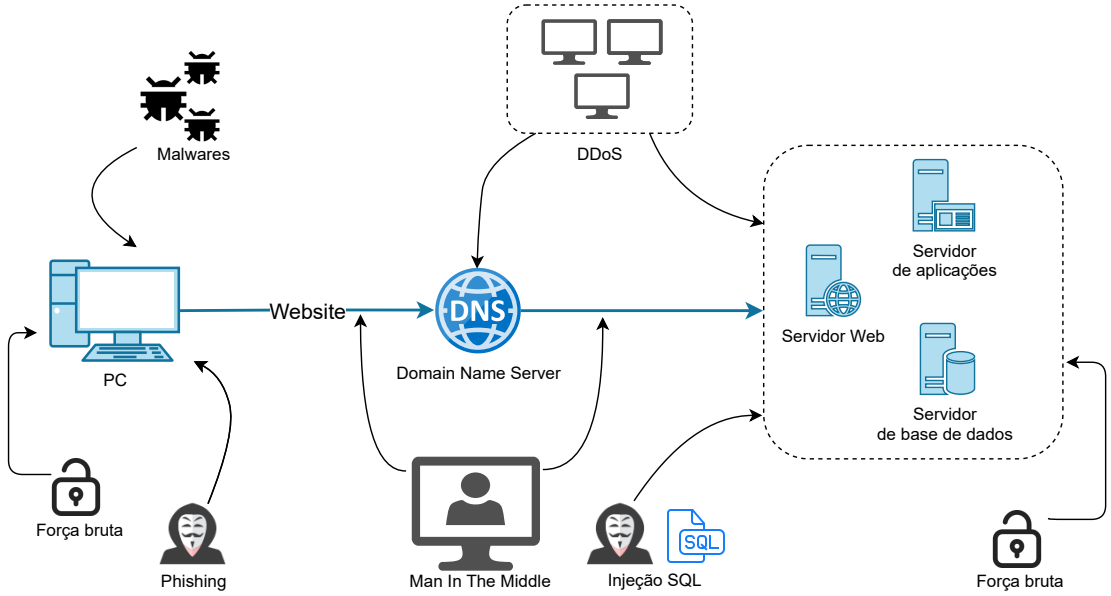

Figura 2.2: Diferentes tipos de ciberataques.

máquinas colocando-as sob o seu controlo [5]. A figura 2.4, ilustra a execução de um ataque DDoS.

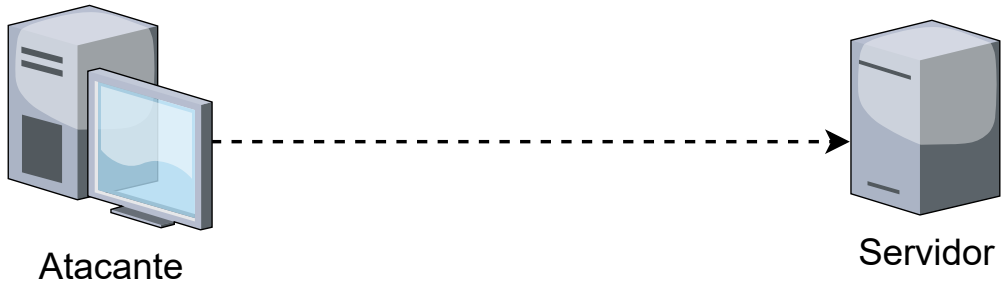

Figura 2.3: Ataque DoS.

Um ataque DDoS exige mais esforços e é normalmente mais prejudicial do que um atacante DoS. Distingue-se de outros ataques pela sua capacidade de utilizar as suas ferramentas de forma coordenada e distribuída na *Internet* e de criar uma grande coleção de dados de tráfego malicioso através da agregação de diferentes elementos. Para além de causar danos a uma vítima, por razões pessoais ou para obter ganhos materiais, o atacante pode também tentar quebrar o sistema de defesa da vítima por diversão ou simplesmente para mostrar a proeza do feito alcançado. Os ataques DDoS podem ser gerados de duas formas distintas [6]: ataque direto e ataque indireto. Num ataque direto, um grande número de pacotes de ataque é enviado diretamente para a máquina da vítima. Neste ataque, o atacante falsifica o endereço IP de origem para que a resposta seja mal direcionada e vá para outro destino. No caso de um ataque indireto, são utilizados muitos nós intermédios, conhecidos como

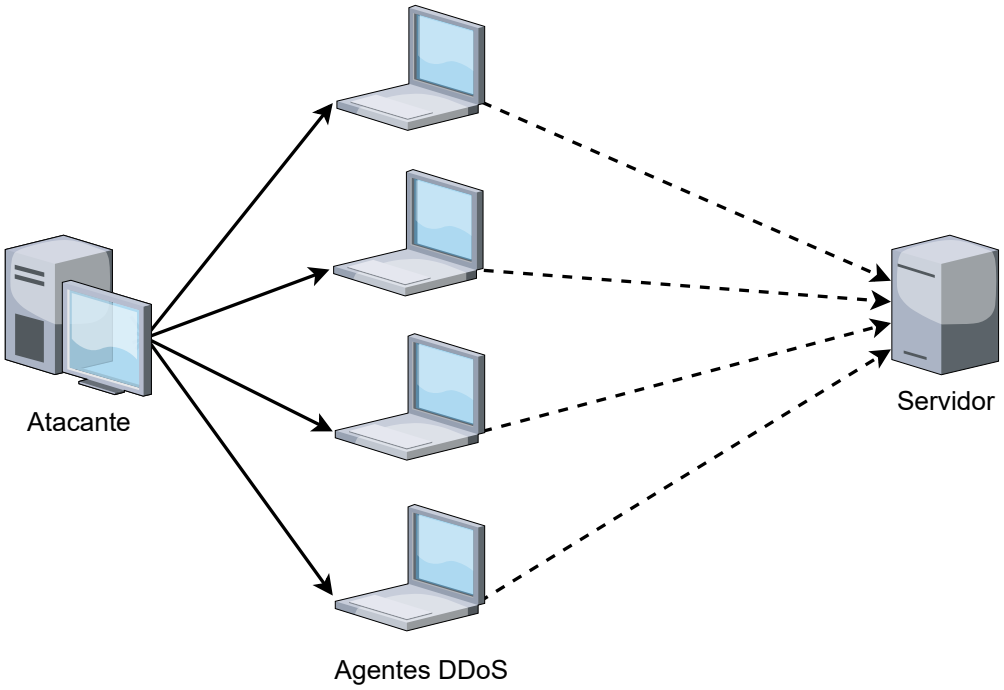

Figura 2.4: Ataque DDoS.

refletores, para gerar um ataque.

Os ataques DoS, estão divididos em dois tipos: ataques de *buffer overflow* e os ataques de *flood* [7]. Onde nos ataques de *buffer overflow*, o invasor consome todos os recursos físicos disponíveis como memoria, processador, etc. Nos ataques de *flood*, o invasor "inunda"o servidor alvo com uma elevada quantidade de solicitações fazendo com que o servidor não consiga processar as solicitações legitimas interrompendo assim os seus serviços.

2.2.2 Botnets

As *botnets*, são redes de computadores comprometidos, controlados remotamente por um ou mais invasores, denominados *botmasters*. Em termos simples, os *bots* são programas maliciosos que operam nos computadores *host* permitindo aos *botmasters* controlarem os computadores *host* remotamente, fazendo com que executem várias ações em simultâneo [8]. Os principais componentes da *botnet* estão representados na figura 2.5, os quatro componentes principais são [9]:

- *Botmaster*: Indivíduo que controla os computadores infetados, enviando e recebendo informações e comandos para um possível ataque.
- Computador *host*: Máquina física ou virtual infetada pelo *bot*.

- Canal de Comando e Controle (Servidor): É a forma como o *botmaster* se comunica, envia ou recebe informações e comandos para os *bots*.
- *Bot*: *Malware* instalado no computador *host*, geralmente usado para ações maliciosas.

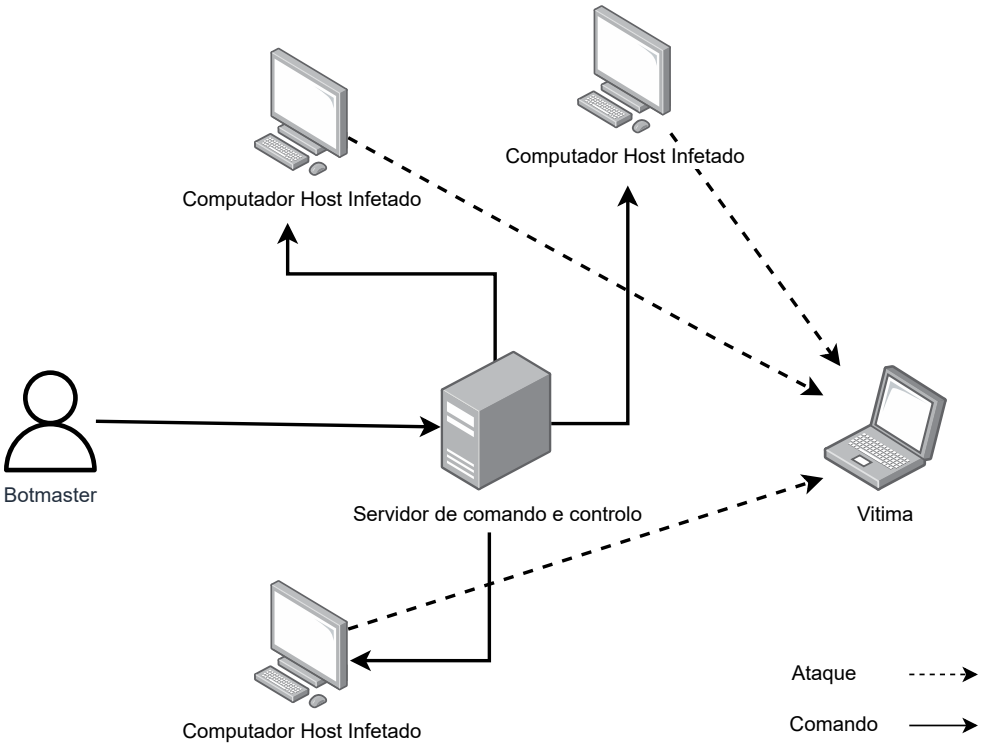

Figura 2.5: Ataque Botnet.

O número crescente de máquinas conectadas à *Internet*, tem criado um ambiente ideal para a disseminação e desenvolvimento de mais *botnets* e por essa razão, estes ataques são considerados uma das maiores ameaças do ciberespaço.

As *botnets* são utilizadas para diversos fins, na grande maioria deles associados a condutas ilegais. Alguns dos seus usos incluem, o lançamento de ataques DDoS, envio de *spam*, *malwares*, e-mails de *phishing*, recolha de informações pessoais, extorsão e manipulação de jogos ou pesquisas *online* [10].

2.2.3 Malware

O *malware* é um código executado num sistema computacional, mas cuja presença ou comportamento é desconhecida pelo administrador do sistema, pois se estivesse ciente da presença e do comportamento do código, ele não permitia a sua execução. O *malware* compromete a confidencialidade, integridade ou a disponibilidade do sistema explorando as suas vulnerabilidades, ou criando novas vulnerabilidades [11].

Em seguida, é apresentado alguns tipos de *malwares*. De salientar que a evolução dos próprios pode levar a diferentes definições, conceitos e variantes. Sendo assim, serão apresentadas as variantes que, no contexto desta dissertação, são tidas como mais relevantes.

- **Vírus**: O vírus é um programa de computador que se transmite de um computador para outro, anexando-se a outro programa. O programa ao qual o vírus se anexa é um dos programas ou arquivos da vítima. Existem muitas formas de transmitir vírus, como enviar um arquivo infetado como anexo de um e-mail ou incorporar cópias de arquivos infetados num dispositivo externo como, por exemplo, uma unidade USB [12].

- **Worms**: Um *worm* é um *software* malicioso que possui uma estrutura independente que lhe permite distribuir de um computador para o outro. Este *malware*, tem a capacidade autónoma de se replicar de forma, a conseguir explorar a vulnerabilidade dos sistemas e de se propagar através da rede [13]. As principais diferenças entre um *worm* e o vírus são: a sua capacidade de replicar cópias de si mesmo automaticamente sem qualquer ação humana e o facto de um *worm* não necessitar de anexar-se a um programa existente [12].

- **Cavalo de Troia (Trojan Horse)**: O cavalo de troia é um tipo de *malware* que se disfarça, ou se comporta como um programa benéfico, fingindo ser inofensivo com o objetivo de conceder ao invasor, informações potencialmente relevantes contidas nos ficheiros do computador da vítima [14]. Há várias maneiras de infetar os computadores das vítimas, como *download* de um site, mas mais recentemente, os cavalos de troia utilizam *worms* e vírus para conseguirem penetrar nos computadores das vítimas [12].

- **Ransomware**: O *ransomware* é um *malware* que encripta, com elevado grau, os ficheiros do utilizador (documentos e fotos), impedindo ou limitando o acesso aos seus dados. Este *malware*, tem como principal objetivo extorquir dinheiro das vítimas. Se o utilizador não realizou uma cópia de segurança dos seus ficheiros antes do ataque, será forçado a escolher entre, pagar um resgate pelos seus dados ou desistir completamente, formatando o seu computador, acabando por perder tudo. A maioria dos *ransomwares*, exige um pagamento rápido, em Bitcoin ou noutra criptomoeda, pois desta forma, permite ao invasor permanecer no anonimato [11]. Dependendo da metodologia utilizada, o *ransomware* é geralmente classificado em dois tipos: *cryptographic ransomware* e *locker ransomware* [15]. O primeiro tipo de *ransomware* é o mais usual, onde os arquivos da vítima são encriptados e posteriormente, é exigido um resgate pela

desencriptação dos mesmos. Já o *locker ransomware*, impede a vítima de aceder ao seu sistema bloqueando o ecrã ou o navegador, e exige o pagamento de um resgate para desbloquear o sistema. Ao contrário do *cryptographic ransomware*, este tipo de *ransomware* não encripta o sistema ou os ficheiros do utilizador.

- **Spyware**: O *spyware* é uma das ameaças mais comuns na *Internet*. É projetado para recolher e monitorizar informações da vítima tais como histórico de navegação, palavras-passe, informações pessoais e as teclas digitadas no teclado sem o seu consentimento. Além disso, *spyware* pode ativar as câmaras e os microfones para vigiar a vítima de forma silenciosa [16].

- **Adware**: O *adware* é utilizado para exibir publicidade ou anúncios, geralmente na forma de *banners*, janelas de *pop-up*, mensagens de correio eletrónico entre outros serviços [17]. Este tipo de *malware*, na maioria das vezes, não costuma prejudicar o sistema, fazendo com que a vítima não tenha noção que o seu sistema foi infetado [11].

- **Rootkit**: Um *rootkit* é uma coleção de uma ou mais ferramentas programadas para ocultar a sua presença num sistema e fornecer ao invasor controlo total desse mesmo sistema. Os *rootkits* por norma, são usados para obter acesso não autorizado a um sistema e realizar várias atividades maliciosas, como roubo de informações confidenciais e monitorização do comportamento da vítima. Inicialmente, os *rootkits* apareceram nos sistemas operativos UNIX e permitiam ao invasor obter e manter acesso ao utilizador com maiores privilégios (nos sistemas operativos UNIX, esse utilizador é chamado "*root*"− dai a origem do nome) [18]. De salientar, que os outros tipos de *malwares* tendem a aplicar as ferramentas de *rootkit* para se instalarem e serem ativados de maneira a não serem detetados, causando enormes prejuízos para o sistema da vítima. Ao ser instalado, o *rootkit* consegue alterar o sistema operativo ou os *softwares* existentes. Por essa razão, torna-se extremamente difícil detetar o *rootkit* e os seus programas associados porque ele consegue corromper programas de segurança ou antivírus [19].

2.2.4 Phishing

Um dos principais objetivos da cibersegurança é proteger dados confidenciais contra ataques de *phishing* e lavagem de dinheiro que usam engenharia social. O ataque de engenharia social consiste na "arte"de manipular as pessoas menos cientes sobre esses tipos de ataque [20]. Para proteger as informações confidenciais do ataque de engenharia social, a questão

da segurança têm sido uma grande preocupação para organizações, utilizadores comuns, desenvolvedores de sites e especialistas. O *phishing* é um problema sério num serviço cada vez mais ilimitado da *Internet*. O ataque de *phishing* é um dos mais usuais e populares entre todos os tipos de ataques de engenharia social [21]. Normalmente é efetuado, enviando um e-mail disfarçado de uma pessoa ou empresa confiável com uma solicitação aparentemente legitima. Os ataques de *phishing* mais comuns, vêm geralmente de bancos populares e por norma contêm algum tipo de ameaça de interrupção de um serviço caso as instruções não sejam seguidas [18]. Também é importante referir que os *links* presentes nos e-mails de *phishing*, embora possam parecer legítimos, quase sempre apontam para um site diferente. Os invasores, clonam sites legítimos e alteram a página de *login* para roubar as informações da credencial de *login* [7].

Os tipos de ataques de *phishing* são [20]:

- **Spoofing email**: *Spoofing* acontece quando um *spammer* (pessoa ou organização que envia mensagens irrelevantes ou não solicitadas pela *Internet*, geralmente para um grande número de utilizadores, para fins de publicidade, *phishing*, etc) envia um e-mail para um utilizador usando outro endereço de correio eletrónico. A falsificação de e-mail é possível devido ao SMTP (*Simple Mail Transfer Protocol*). Ele é utilizado no envio de e-mails e não inclui um processo de autenticação. Portanto, esse tipo de ataque pode manipular o utilizador facilmente para divulgar informações confidenciais.

- **Contas de redes sociais falsas**: Os utilizadores de redes sociais como Facebook, Twitter, LinkedIn por vezes não estão conscientes da exposição das suas contas. Uma conta falsa é facilmente criada em sites de redes sociais pelo invasor. Por meio desses perfis falsos é possível aceder a dados pessoais que o utilizador divulga ao criar a conta. Esses sites populares, têm políticas contra contas falsas, no entanto, ainda existem muitas contas falsas disponíveis nesses sites, porque eles não possuem um sistema real que determine a veracidade do utilizador.

- **Hacking**: Um *hacking* é qualquer esforço técnico para manipular o acesso a um sistema ou recurso. Um *hacker* é o individuo responsável pelas ações, que estão envolvidas nesse processo. O *hacker* pode ser motivado por várias razões, como desafio, lucro e prazer. Os *hackers* utilizam um *scanner* de vulnerabilidade e um *scanner* de porta para verificar os computadores numa rede, em busca de pontos fracos. Neste tipo de ataque, o *hacker* pode utilizar o ataque de força bruta, quebra senhas de acesso e o ataque de dicionário. Neste contexto, a engenharia social é muito eficiente, pois os

utilizadores são a parte mais vulnerável de uma organização. Se por descuido, um funcionário revelar uma senha para uma pessoa não autorizada poderá comprometer a segurança da organização.

2.2.5 Injeção SQL

Devido ao alto custo da mão de obra e ao potencial de erro humano, grande parte das empresas preferem fazer a transição de serviços pessoais para serviços *online*. As empresas que procuram atender as necessidades os seus clientes de forma *online* podem fazê-lo com a ajuda de uma aplicação *web*. As aplicações *web* que usam base de dados para armazenar e recuperar dados são denominados "*Database Driven Web Applications*"[22]. As bases de dados relacionais estão entre os tipos mais populares de base de dados. A Linguagem de Consulta Estruturada (SQL,*Structured Query Language*) é um tipo de linguagem de programação criada para manipular e controlar dados nos sistemas de gestão de base de dados relacional. A SQL, tal como outras linguagens de programação, permite ao utilizador usar comentários embutidos no código, mesmo entre as instruções e também oferece a capacidade de concatenação e combinação de caracteres e valores. As bases de dados são a principal forma de armazenamento de informações confidenciais *online* por essa razão tornam-se um alvo muito a apelativo para os atacantes.

Injeção SQL é uma das ameaças mais comuns num sistema de base de dados no qual o atacante adiciona uma instrução SQL na aplicação *web*, para obter acesso ou efetuar alterações aos dados armazenados na base de dados [23]. A falta de validação de entrada nas aplicações *web* faz com que o ataque seja bem-sucedido, pois a consulta SQL maliciosa será inserida na aplicação *web* e, em vez de variáveis, será concatenada com a consulta legitima. Deste modo, o ataque de injeção SQL compromete a base de dados ao manipular e recuperar, de forma não autorizada, os dados confidenciais.

Na ref.[22] é possível ler informações mais detalhes sobre as diferentes variantes de ataques de injeção SQL.

2.2.6 Man In The Middle

O ciberataque *Man In The Middle* (MITM) é um tipo de ataque em que o invasor assume secretamente o controlo do canal de comunicação entre dois dispositivos. O invasor pode intercetar, alterar ou substituir o tráfego de comunicações das vítimas alvos. Além disso, as vítimas não têm conhecimento do intruso, acreditando que o canal de comunicação

encontra-se protegido [24]. Este ciberataque, tem como principal objetivo comprometer a confidencialidade através da escuta de comunicação, a integridade através da modificação das mensagens e a disponibilidade através da destruição de mensagens de modo a que uma das partes termine a comunicação.

Os ciberataques MITM podem ser divididos em 4 categorias distintas [25]:

- **Spoofing-based MITM**: é um ataque no qual o invasor interceta uma comunicação legítima entre dois hosts por meio de um ataque de falsificação e controla os dados transferidos, enquanto os hosts não têm conhecimento da existência de um intermediário.

- **SSL/TLS MITM**: é uma forma de interação ativa de rede, em que o atacante se insere no canal de comunicação entre duas vitimas. Posteriormente, a atacante estabelece duas ligações SSL (*Secure Sockets Layer*) separadas com cada vítima e retransmite as mensagens entre elas, de maneira a que ambas não tenham conhecimento do intermediário.

- **Border Gateway Protocol MITM**: é um ataque baseado no desvio do endereço IP, mas, no entanto o atacante faz com que tráfego roubado seja entregue ao destino. Desta forma, todo o tráfego passa por estação autónoma, onde é possível manipulá-lo.

- **MITM baseado em estações de base falsas**: é um ataque que ocorre quando o atacante obriga uma determinada vítima a se conectar a uma estação base transcetora falsa, que o atacante usa para manipular o tráfego da vítima.

2.3 Mecanismos de deteção

O constante aumento da regularidade e da complexidade dos ciberataques tornou-se cada vez mais notório de ano para ano. Deste modo, o processo de proteção do ciberespaço contra esse tipo de ataques tem-se mostrado uma tarefa bastante desafiadora. Como resultado, existe uma necessidade urgente de mecanismos capazes de detetar de forma eficaz essas ameaças.

Porém, antes de analisar os mecanismos de deteção, é necessário efetuar um breve enquadramento sobre a *framework* da cibersegurança. Geralmente, a *framework* da cibersegurança, como apresentado na figura 2.6, é composta por cinco etapas [26] : identificar, proteger, detetar, responder e recuperar. Na primeira etapa, as medidas de cibersegurança, consistem

em identificar e compreender quais os recurso e fatores de risco, como quem tem acesso às informações confidenciais. Na etapa de proteção, uma vez ocorrido o ataque, é necessário proteger os dados limitando o seu acesso. Na fase de deteção, é preciso implementar um mecanismo de deteção adequado para identificar e atenuar os ficheiros provenientes da fonte do ataque. Na fase de resposta, é executado um plano de atenuação para os ataques e os incidentes de segurança. Por fim, na fase de recuperação, é implementado mecanismos para recuperar as operações normais com ajuda dos dados disponíveis na cópia de segurança. Esta *framework*, é considerada um guia útil para qualquer organização que pretenda melhorar a sua cibersegurança, sendo a etapa de deteção, a fase que desempenha um papel fundamental ao ajudar no futuro, a defender contra os diferentes ciberataques.

Nesta secção, serão apresentados e analisados diferentes mecanismos de deteção usados para detetar os ciberataques referidos na secção 2.2.

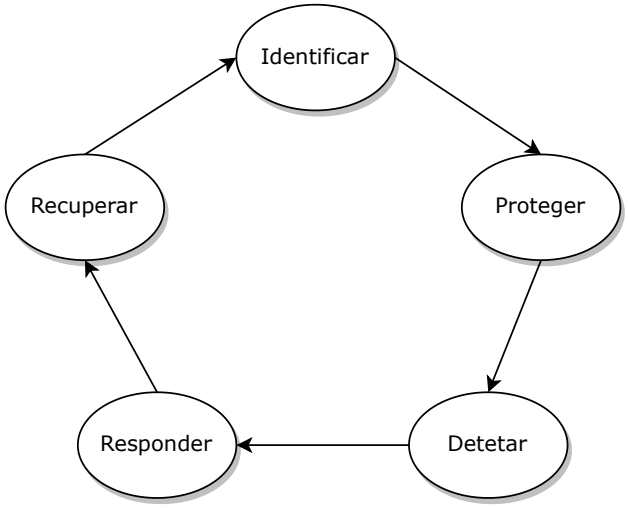

Figura 2.6: *Framework* de cibersegurança.

2.3.1 Mecanismos baseados em assinaturas

Os mecanismos baseados em assinaturas permitem detetar os ataques já conhecidos comparando-os com os padrões de ataques anteriores. É o mecanismo de deteção mais simples de usar, pois apenas é capaz de detetar os ataques conhecidos. Se houver uma pequena alteração no comportamento do ataque, este mecanismo já não consegue detetar a ameaça [27]. Por essa razão, os mecanismos baseados em assinaturas necessitam de atualizar regularmente os padrões de ataque.

As diferentes etapas para a deteção baseada em assinaturas são [28]:

- Preparativos: Neste processo, é extraído os códigos das características, que são posteriormente armazenados na base de dados. Este processo pode ser dividido em três etapas:

    1. Recolha de amostras de códigos maliciosos. Se os códigos maliciosos são capazes de infetar diferentes tipos de ficheiros, devem ser recolhidas diferentes amostras de diferentes tipos de códigos maliciosos.
    2. Extração de características de cada amostra. Trata-se de uma parte importante de todo o sistema. Uma técnica de extração eficiente pode ajudar o sistema a identificar códigos maliciosos.
    3. Armazenamento das características na base de dados. A base de dados armazenará uma quantidade elevada de registos, devido ao grande número de códigos maliciosos que, por sua vez, aumentam os custos de tempo. Um ponto fundamental é a elaboração de uma estrutura de dados que seja eficaz.

- Analise e tomada de decisões:

    1. Pré-processamento de códigos. Refere-se a uma parte importante que inclui análise dos tipos, descompressão, descompactação, analise sintática, etc.
    2. Análise. À medida que cada programa alvo é executado, o sistema compara os seus códigos de características com os da base de dados. Se os códigos coincidirem, podemos inferir que o programa está comprometido. Se o programa for considerado benigno, é necessário efetuar uma análise completa e dispendiosa à base de dados. Os algoritmos de correspondência, são fundamentais para resolver esse problema.

Resumindo, o procedimento de deteção baseado em assinaturas é apresentado na figura 2.7. Em primeiro lugar, algumas amostras de códigos maliciosos são recolhidas. Em seguida, para cada tipo de programa malicioso, é extraída as assinaturas maliciosas que descrevam as suas características. Posteriormente, é armazenada a assinatura na base de dados para ser analisada. Por fim, o programa alvo e a base de dados são analisados para realizar a deteção de algo suspeito.

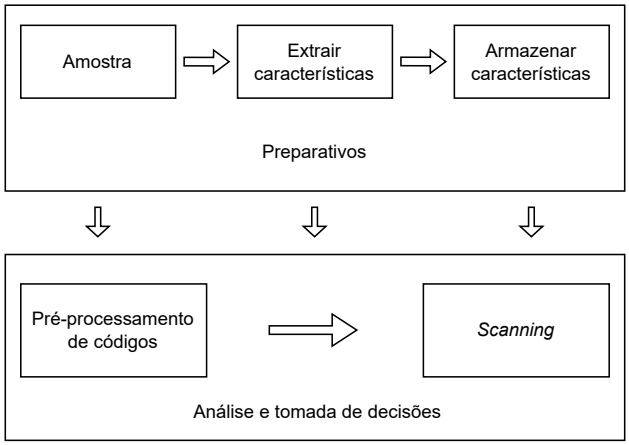

Figura 2.7: Procedimento de deteção baseado em assinaturas.

2.3.2 Mecanismos baseados em anomalias

Os mecanismos baseados em anomalias baseiam-se na definição do comportamento da rede ou do sistema. Quer isto dizer que, se o comportamento da rede não estiver de acordo com o comportamento predefinido, não será aceite e desencadeará um evento na deteção de anomalias. O comportamento da rede que é considerado aceite é preparado ou aprendido através das especificações dos administradores de rede [27].

A deteção baseada em anomalias têm duas grandes vantagens em relação à deteção baseada em assinaturas. A primeira vantagem é a capacidade de detetar ataques desconhecidos, bem como ataques de "dia zero"(termo usado quando uma vulnerabilidade desconhecida pelos administradores é descoberta pelo invasor, onde ele tira proveito dessa falha e a utiliza para um ataque antes dos administradores tentarem mitigá-la) [29]. Isto deve-se à capacidade dos mecanismos de deteção de anomalias para modelar o funcionamento normal de um sistema ou de uma rede e detetar desvios em relação a esse funcionamento. Uma segunda vantagem é que os perfis de atividade normal são personalizados para cada sistema ou rede, o que torna muito difícil para um atacante saber com certeza que atividades pode realizar sem ser detetado [30]. Contudo, este mecanismo, seleciona por vezes uma atividade benigna como uma anomalia que nada tem a ver com uma ameaça de segurança. Como resultado, tende a produzir um grande número de falsos positivos que exigem que o administrador verifique manualmente cada um deles por meio de uma análise dispendiosa e demorada [31].

Em seguida, de entre um vasto conjunto de mecanismos de deteção de anomalias, é analisado os quatro mecanismos principais usados para a deteção de anomalias na rede [32].

Deteção de anomalias utilizando um classificador

A deteção de anomalias depende da ideia de que o comportamento normal das características pode ser distinguido do comportamento anormal. Dado o evento atual, um classificador pode ser usado para prever o próximo evento normal. Se durante a fase de monitorização, o evento subsequente diferir do que o classificador havia previsto, ele é considerado uma anomalia. O processo de classificação normalmente envolve as seguintes etapas:

1. Identificar os atributos e as classes a partir dos dados de treino.
2. Identificar atributos para a classificação
3. Aprender um modelo utilizando os dados de treino
4. Utilizar o modelo aprendido para classificar as amostras de dados desconhecidos.

Existem vários mecanismos de classificação. Estes incluem técnicas de geração de regras indutivas, técnicas baseadas em algoritmos genéticos e lógica difusa. Para categorizar os dados de auditoria, os algoritmos de geração de regras indutivas normalmente aplicam um conjunto de regras de associação e padrões de episódios frequentes. O benefício da adoção de regras é que elas geralmente são simples e intuitivas, não estruturadas e menos rígidas. Como desvantagens, são difíceis de manter e, em certas situações, inadequadas para representar muitos tipos de informação. Os algoritmos genéticos são uma técnica de pesquisa utilizada para encontrar soluções aproximadas para problemas de otimização e pesquisa. Também têm sido amplamente utilizados no domínio da deteção de intrusões para diferenciar o tráfego de rede normal das ligações anómalas. A principal vantagem dos algoritmos genéticos é a sua flexibilidade e robustez como método de pesquisa global. As técnicas de lógica difusa são há muito, utilizadas no domínio da segurança de rede tendo sido desenvolvido um motor de reconhecimento de intrusões difusas (*Fuzzy Intrusion Recognition Engine* - FIRE) utilizando conjuntos difusos e regras difusas. O motor de reconhecimento de intrusões difusas utiliza técnicas simples de extração de dados para processar os dados de entrada da rede e gerar conjuntos difusos para cada característica observada.

Deteção de anomalias utilizando estatística

Nos métodos estatísticos de deteção de anomalias, o sistema acompanha as atividades dos indivíduos e cria perfis para representar o seu comportamento. Normalmente, são mantidos dois perfis para cada sujeito: o perfil atual e o perfil armazenado. O sistema atualiza o perfil

atual à medida que os eventos de rede são processados e calcula uma pontuação de anomalia periodicamente, comparando o perfil atual com o perfil armazenado usando uma função de anormalidade de todas as medidas dentro do perfil. Se a pontuação da anomalia for superior a um determinado limiar, o sistema gera um alerta. A deteção estatística de anomalias tem uma série de vantagens. Em primeiro lugar, estes sistemas não requerem conhecimento prévio das falhas de segurança ou dos próprios ataques. As abordagens estatísticas também podem, fornecer notificações com precisão sobre atividades maliciosas, que normalmente ocorrem durante longos períodos de tempo. No entanto, os sistemas estatísticos de deteção de anomalias também têm inconvenientes. Em alguns casos, pode ser difícil determinar limiares que equilibrem a probabilidade de falsos positivos com a probabilidade de falsos negativos. Além disso, os métodos estatísticos necessitam de distribuições estatísticas exatas, mas nem todos os comportamentos podem ser modelados utilizando métodos puramente estatísticos.

Deteção de anomalias utilizando redes Bayesianas

Uma rede Bayesiana é um modelo gráfico que codifica relações probabilísticas entre variáveis de interesse. As redes Bayesianas apresentam vários benefícios quando combinadas com métodos estatísticos para análise de dados. Desta forma, é possível criar modelos de deteção de anomalias que utilizam redes Bayesianas ingénuas para efetuar a deteção de intrusões no tráfego da rede. Apesar da utilização de redes Bayesianas para a deteção de intrusões possa ser eficaz em determinadas aplicações, as suas limitações devem ser tidas em conta na sua implementação. O aspeto mais importante na solução do problema é a seleção de um modelo preciso uma vez que, a precisão desse método depende de várias suposições que normalmente são baseadas no modelo comportamental do sistema alvo.

Os conjuntos de dados típicos para a deteção de intrusões são muito grandes e multidimensionais. Como forma de resolver esse problema, é utilizada uma técnica de redução da dimensionalidade conhecida como análise de componentes principais (*Principal Component Analysis* - PCA). A análise de componentes principais é uma técnica em que n variáveis aleatórias correlacionadas são transformadas em $d < n$ variáveis não correlacionadas. As variáveis não correlacionadas são combinações lineares das variáveis originais e podem ser utilizadas para expressar os dados numa forma reduzida. Nesse sentido, a análise de componentes principais é utilizada como um esquema de deteção de anomalias com objetivo de reduzir a dimensionalidade dos dados de auditoria e chegar a um classificador que é uma função dos componentes principais.

Deteção de anomalias utilizando máquinas de estados finitos

Uma máquina de estados finitos é um modelo de comportamento composto por estados, transições e ações. Neste modelo, um estado guarda informações sobre o passado, uma transição indica uma mudança de estado sendo descrita por uma condição que teria de ser cumprida para permitir a transição. Uma ação é a descrição de uma atividade que deve ser executada num determinado momento. A maquina de estados finitos é utilizada para detetar ataques ao protocolo DSR (*Dynamic Source Routing*). Esta técnica utiliza um algoritmo de seleção de monitores para monitorizar todos os nós da rede e os comportamentos corretos dos nós que foram atribuídos manualmente. A utilização deste método tem a vantagem de detetar intrusões sem a necessidade de dados ou assinaturas treinadas, podendo também ser detetadas intrusões desconhecidas com poucos falsos alarmes. Como resultado, foi proposta uma arquitetura de monitorização de rede distribuída que rastreia o fluxo de dados em cada nó através de uma máquina de estados finitos.

2.3.3 Mecanismos baseados em data mining

As técnicas baseadas em *data mining* ajudam a detetar os ataques internos, analisando cuidadosamente os padrões de ataques ocorridos anteriormente [7]. *Data mining* consiste no processo de extração de modelos ou padrões relevantes, não detetados pelos sistemas de armazenamento de dados. Encontrar padrões ocultos numa enorme quantidade de dados é o objetivo principal do processo de *data mining*. Assim, o *data mining* também pode ser considerado como uma forma de descobrir novos comportamentos.

Como demonstrado na figura 2.8, a deteção de ciberataques com base em técnicas de *data mining* envolve cinco etapas diferentes, nomeadamente a monitorização do sistema e a captura de dados através de vários sensores, agentes de registo e deteção de redes, sistemas, processos e dispositivos de segurança, o pré-processamento de dados em sistemas de armazenamento de dados locais, a correlação de eventos e a extração de características, *data mining* para detetar utilizações indevidas ou ameaças, por fim, visualização e a interpretação dos resultados da análise de dados.

Quando configurada corretamente, a deteção baseada em *data mining* tem a capacidade de se tornar o cérebro da rede, pois disponibiliza algumas funções úteis como a, monitorização em tempo real e a gestão de incidentes relacionados com a segurança fornecendo um fluxo de trabalho que ajuda a seguir e a escalar o incidente. Pode tambem ser usada para a gestão e consolidação de registos e na criação de relatórios de conformidade. De forma geral, estas

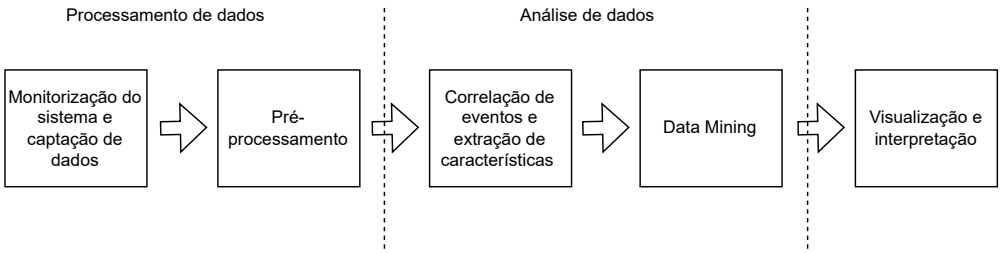

Figura 2.8: Fases da deteção de ciberataques com base em técnicas de *data mining*.

etapas podem ser divididas em três camadas [33]:

- **Processamento de dados**

  O processamento de dados é a primeiro camada da deteção de ciberataques com base em técnicas de *data mining*. Envolve a organização e a formatação dos dados para análise. Vários dispositivos servem como fontes de dados, incluindo dispositivos de rede e de segurança. A captação de dados é responsável pela recolha de registos e eventos de segurança de dispositivos como *firewalls*, IDS (*Intrusion Detection System*) e sistemas antivírus. Também capta informações sobre ataques e dados sensíveis do tráfego espelhado. Os dados recolhidos são depois armazenados numa base de dados ou num sistema de armazenamento de dados. Em todos esses dados, são aplicadas regras de correlação para extrair informações significativas dos dados armazenados. O pré-processamento de dados é efetuado para normalizar e limpar os dados recolhidos. Isto inclui tarefas como normalização, filtragem e limpeza. A normalização assegura a consistência nos formatos e valores, enquanto a filtragem melhora a eficiência e a precisão reduzindo os dados irrelevantes. A limpeza de dados remove o ruído e elimina os valores de dados em falta. A monitorização da segurança engloba várias funções, como a aquisição de dados, a correlação de eventos, a análise de segurança e o fornecimento de uma visão geral do estado da segurança. Envolve a monitorização em tempo real das fronteiras da rede, correlacionando eventos de segurança e gerando relatórios de análise de segurança para obter informações sobre a estabilidade da segurança da rede.

- **Análise de dados**

  Esta camada é considerada o processo central da deteção de ciberataques com base em técnicas de *data mining*. Os dados maciços recolhidos durante a camada de captação destinam-se ao armazenamento e analise centralizada, de modo a extrair as principais informações. Os dados recolhidos são analisados nesta camada para determinar se os

dados são anómalos ou não. A correlação de ameaças utiliza a inteligência artificial (IA) para classificar vários registos e entradas de registo para identificar os atacantes. A camada de análise realiza funções de análise em tempo real e análise profunda. Entre elas, a análise em tempo real inclui o posicionamento de eventos, a análise da correlação temporal, a base de conhecimentos, a análise da situação de segurança, a geração de alarmes e a indexação do armazenamento; enquanto a análise profunda inclui o armazenamento de dados e as estatísticas, a análise baseada na inteligência artificial e a elaboração de relatórios. Em seguida, são utilizados diferentes técnicas baseadas em *data mining* para extrair dados da base de dados. Aqui, pode ser utilizado alguns procedimentos de transformação para transformar os dados no formato desejado. Posteriormente, são utilizados vários algoritmos de extração de dados para processar os dados. A classificação dos dados é o principal passo nesta fase. Os dados são, portanto, organizados segundo um padrão. Com base nos esquemas de análise que estão a ser utilizados, a classificação é determinada.

- **Resposta**

  No final, é necessário tomar as medidas certas em resposta aos ataques detetados. No caso de situações de análise manual, esta pode ser efetuada manualmente, o que significa que a averiguação final é feita por administradores humanos para determinar os ciberataques e a sua mitigação. O sistema de deteção utiliza várias ferramentas, como correio eletrónico, ícones de alarme e técnicas de visualização, para notificar o administrador do sistema da ocorrência de um ataque. Um ataque também pode ser interrompido e controlado por um sistema de deteção através do bloqueio de portas de rede ou do encerramento de processos. Uma abordagem mais abrangente para monitorizar uma série de fontes de eventos diversificados para a deteção de ciberataques pode permitir um melhor conhecimento da situação das ameaças no ciberespaço, melhorando assim a deteção.

2.3.4 Mecanismos baseados em machine learning

O *Machine Learning* tem inúmeras aplicações na cibersegurança. Estas aplicações incluem a identificação de ciberameaças, o combate à cibercriminalidade e deteção de ciberameaças utilizando capacidades de IA. Uma das tarefas mais difíceis na cibersegurança é a deteção de atividades suspeitas na fase de receção e envio de dados. O *Machine Learning* consegue automatizar a deteção da ocorrência de uma ameaça durante essa troca de dados [24]. As

tarefas podem ser automatizadas com ajuda do *Machine Learning*, permitindo aos administradores do sistema concentrarem-se em tarefas mais importantes em vez de tarefas mais repetitivas. Com o auxílio do *Machine Learning*, podem ser detetadas partículas de código malicioso em execução nos servidores. Utilizando conjuntos de dados de ciberataques anteriores, o *Machine Learning* pode detetar determinados tipos de ataques à rede e, com esta análise, pode ser efetuada uma classificação dos riscos da rede [34].

A deteção, classificação de ataques e análise são os três principais objetivos das abordagens de *Machine Learning* na deteção de ataques. Antes de serem utilizados no treino do modelo, os dados de treino passam por uma série de pré-processamentos. Estas operações consistem na transformação e normalização dos dados. Para aumentar o desempenho do modelo a partir desses dados é fundamental usar o processo de redução da dimensão. Para este processo são aplicadas métodos estatísticos de redução de dimensão ou algoritmos de seleção de características. Após o treino do modelo, os dados de teste são usados para detetar um ataque [24]. O sucesso do modelo é avaliado através de várias métricas de desempenho, comparando os resultados previstos com os resultados reais. Na figura 2.9 pode ser observado o diagrama de fluxo destas fases.

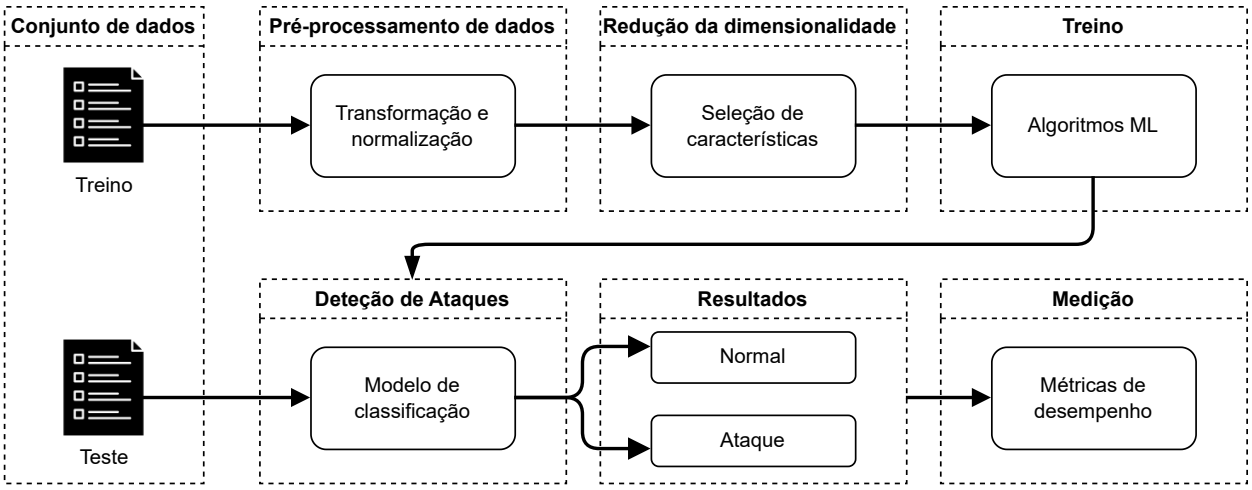

Figura 2.9: Fluxograma de deteção de ataques.

O *Machine Learning* inclui uma grande variedade de paradigmas que oferecem ligações cruzadas e estão em constante evolução. Por esse motivo, torna-se difícil escolher quais os métodos a utilizar para cada problema. Assim sendo, é apresentado quatro métodos de *Machine Learning* que são frequentemente usados na deteção de ciberataques.

Support Vector Machine

Introduzida pela primeira vez em 1995, a *Support Vector Machine* (SVM) ganhou recentemente destaque. Este método pode ser utilizado tanto para problemas de classificação como de regressão, embora seja mais frequentemente utilizado para problemas de classificação. A SMV é um método baseado no espaço vetorial que encontra um limite de decisão entre duas classes que estão mais afastadas de qualquer ponto nos dados de treino [35].

Redes Neuronais Profundas

As redes neuronais com um grande número de camadas ocultas são designadas por redes neuronais profundas. Com a propagação progressiva, os valores de entrada na rede neuronal percorrem a rede, atingindo as camadas ocultas. Cada camada oculta aceita esses valores de entrada, processa-os de acordo com a função de ativação e transmite-os à camada seguinte. Com a retropropagação, a rede neuronal é treinada de forma eficaz. O objetivo da retropropagação é minimizar o valor da função de perda, ajustando os pesos e os desvios da rede. Para cada camada, há uma propagação progressiva e uma retropropagação correspondente. Durante estas operações, é utilizada uma *cache* para transferir os valores calculados e as informações entre si. Num modelo de rede neural simples, o número de camadas é limitado. Com o desenvolvimento das tecnologias de *hardware*, o número de camadas ocultas e o número de neurónios em cada camada podem ser aumentados. À medida que a profundidade dos modelos de redes neuronais artificiais aumenta, são necessários mais dados para obter um maior sucesso [24].

Random Forest

A *Random Forest* (RF), baseia-se no conceito de aprendizagem em conjunto e no processo de combinação de vários classificadores para resolver um problema complexo e melhorar o desempenho do modelo. Uma árvore é construída a partir de um subconjunto retirado aleatoriamente do conjunto de aprendizagem. Cada árvore individual na RF prevê uma classe, e a classe com mais votos torna-se a previsão. Esta técnica resolve o problema do estimador de alta variância presente na aprendizagem de árvores de decisão [24][35].

Redes Neuronais Convolucionais

A rede neuronal convolucional é uma rede neural concebida para processar valores de entrada armazenados em matrizes. As redes neuronais convolucionais são frequentemente

utilizadas para processar sequências de imagens 2D, frequências de áudio e imagens de vídeo ou volumétricas em matrizes 3D. Tal como um modelo de rede neural tradicional, um modelo de rede neuronal convolucional tem uma camada de entrada, uma camada de saída e muitas camadas ocultas. As camadas ocultas deste modelo são normalmente constituídas por uma ou mais camadas convolucionais, camadas de *pooling* e camadas totalmente ligadas. A camada de *pooling* procura identificar a presença de um padrão para ajustar os parâmetros, permitindo desta forma, reduzir o tamanho dos dados para conservar apenas os elementos mais essenciais e limitar o risco de *sobreajustamento*.

3 Simulador de redes de computadores

A simulação é uma técnica utilizada em várias áreas cientificas para imitar o comportamento de certas situações, sistemas ou cenários do mundo real através de um modelo que permita examinar as características estáticas ou dinâmicas. Por conseguinte, simuladores de redes podem ser ferramentas muito úteis na pesquisa e no desenvolvimento de novas redes, ao permitirem modelar, analisar e avaliar o comportamento da rede sem a necessidade de implementar *hardware* físico. Desta forma, os simuladores de redes fornecem uma alternativa mais económica e eficiente de estudar protocolos, aplicações e topologias de rede num ambiente virtual totalmente controlado. Contudo, o desenvolvimento do modelo deve refletir o mais fielmente possível a realidade, de modo a garantir a fiabilidade dos resultados.

Em seguida, é apresentada a ferramenta de simulação utilizada neste trabalho, bem como as razões que levaram à escolha dos simuladores.

3.1 GNS3

O *Graphical Network Simulator 3* (GNS3), é um simulador de redes *open source* desenvolvido em *python* que permite não só simular, mas também testar e emular topologias de redes complexas. Foi lançado em 2008 e rapidamente alcançou uma grande popularidade na comunidade de redes devido aos recursos e funcionalidades que oferece. O GNS3 é divido em dois componentes de *software*: interface gráfica e uma máquina virtual.

A interface gráfica é o local onde é possível criar laboratórios de rede com uma variedade de *routers*, *switches* e PCs configuráveis. A figura 3.1 ilustra a versão 2.2.41 do GNS3 na interface do *Windows*. Esta figura mostra um exemplo de uma rede simples.

Após a criação da topologia no GNS3, os dispositivos criados necessitam de ser hospedados e executados por um processo de servidor. O GNS3 apresenta duas opções para a parte do servidor. A primeira opção seria utilizar um servidor local executado localmente pelo mesmo computador onde foi instalado o GNS3. Porem, esta configuração é limitada e não oferece

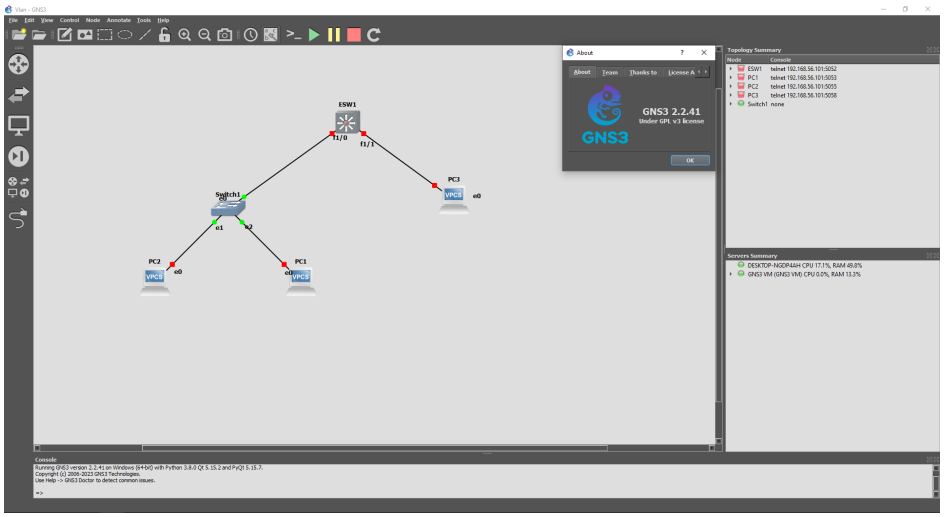

Figura 3.1: Interface gráfica do utilizador GNS3

tantas opções em relação ao tamanho da topologia e aos dispositivos suportados. A segunda opção, que neste caso foi a configuração escolhida, seria instalar uma máquina virtual através de um *software* de virtualização como *VirtualBox*. A figura 3.2 mostra o servidor que foi utilizado durante o desenvolvimento do trabalho.

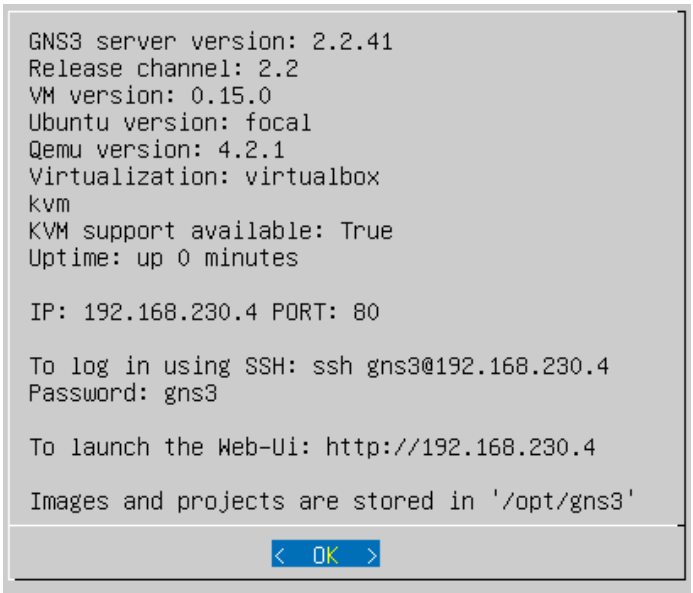

Figura 3.2: Servidor GNS3

O GNS3 utiliza a tecnologia cliente-servidor, da mesma forma que um navegador *web* se conecta a um servidor *web* para aceder e exibir páginas *web*, o programa da interface gráfica do GNS3 acede ao servidor GNS3, permitindo iniciar, parar e controlar todos os dispositivos inseridos na topologia. Isso permite que os projetos criados sejam escalonados porque não estão restritos à execução num único computador. No caso de topologias grandes ou complexas, também é possível executar o programa do servidor GNS3 num computador

diferente do programa da interface gráfica.

A principal característica deste simulador, comparativamente aos outros simuladores, é sem dúvida o facto desta ferramenta suportar vários programas de emulação como *Qemu, VMware* e *VirtualBox* que permitem testar *routers, firewalls* e *swicthes* de fabricantes como a *Cisco* e *Huawei* em diferentes sistemas operativos. O GNS3 oferece uma elevada flexibilidade para os seus projetos através de uma combinação de dispositivos de *hardware* emulados que executam sistemas operativos de rede reais, como o *Cisco IOS*, e a capacidade de partilhar recursos entre vários computadores. De salientar que, o GNS3 também é capaz de conectar *hardware* virtual com *hardware* real. Desta forma é possível criar uma topologia parcialmente virtual e física.

4 Modelos de simulação desenvolvidos

Com o intuito de simular diferentes ciberataques, foi desenvolvido no GNS3, uma topologia de rede especificamente adaptada para simular ciberataques que correm com alguma frequência no nosso quotidiano. Desta forma, será possível compreender e analisar quais os danos e os prejuízos causados na rede. Na figura 4.1, é representada a topologia de rede desenvolvida para esse efeito.

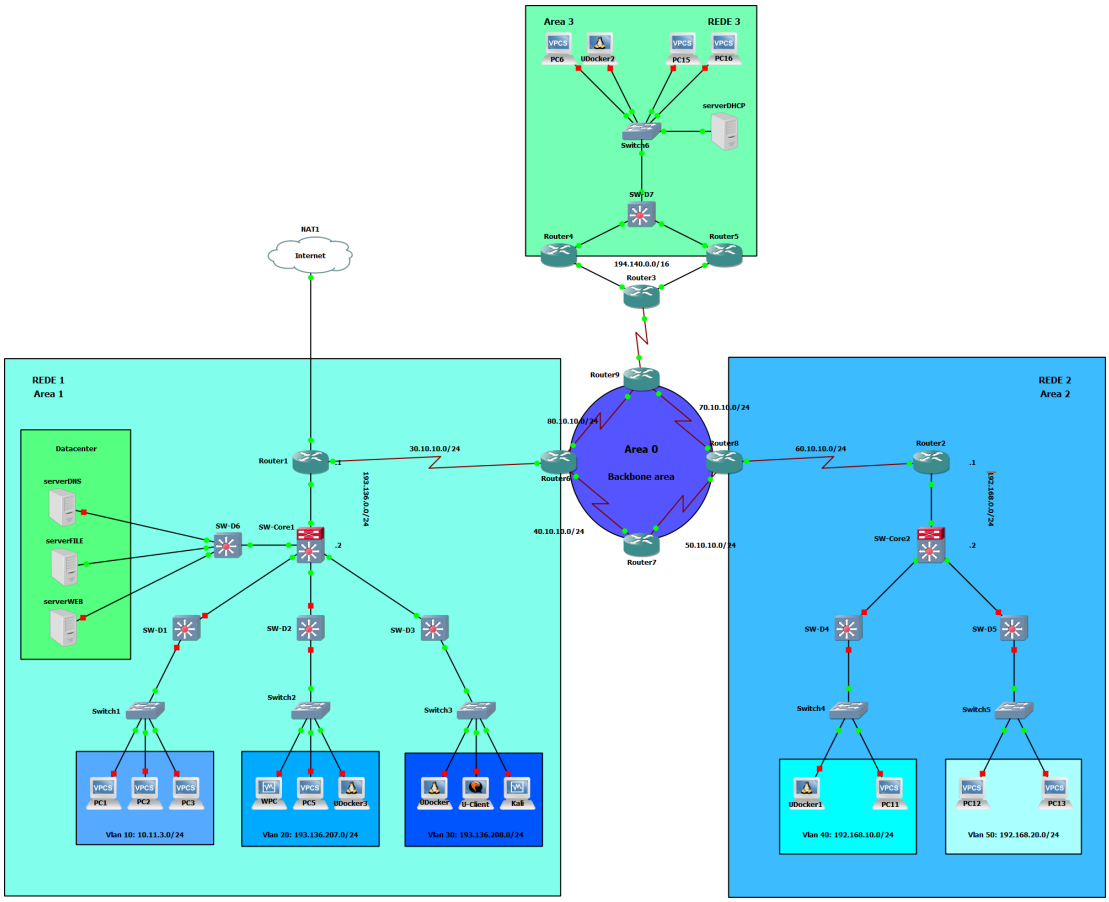

Figura 4.1: Topologia de rede GNS3.

Como podemos averiguar pela figura, esta topologia é constituída por três redes principais. Sendo que a Rede 1, foi desenvolvida com uma estrutura semelhante à do Departamento de Engenharia Eletrotécnica e de Computadores da Universidade de Coimbra. As demais

redes (Rede 2 e Rede 3) foram criadas para simular a interação com a Rede 1. A seguir, é feita uma análise detalha da implementação da topologia de rede.

4.1 Topologia de rede

4.1.1 Rede 1

Na implementação da Rede 1, foi adotado o modelo de 3 Camadas criado pela *Cisco* para auxiliar na gestão da rede, tornando-a mais escalável. Este modelo *Cisco*, como mostra a figura 4.2, define três camadas hierárquicas. Conforme se avança no modelo, as conexões têm a sua largura de banda aumentada. Assim, à medida que se sobe na hierarquia, há uma maior concentração de tráfego de dados. Por isso, é importante planear a rede de forma eficiente e cuidadosa, evitando o surgimento de conflitos de tráfego. Esta rede é constituída por três VLANs (*Virtual Local Area Network*) distintas, sendo elas a VLAN 10, VLAN 20 e a VLAN 30. Desta forma, conseguimos segmentar a rede sem a necessidade de utilizarmos um elemento da camada 3. Assim podemos organizar a rede em diferentes departamentos, melhorando significativamente o desempenho e a gestão da mesma.

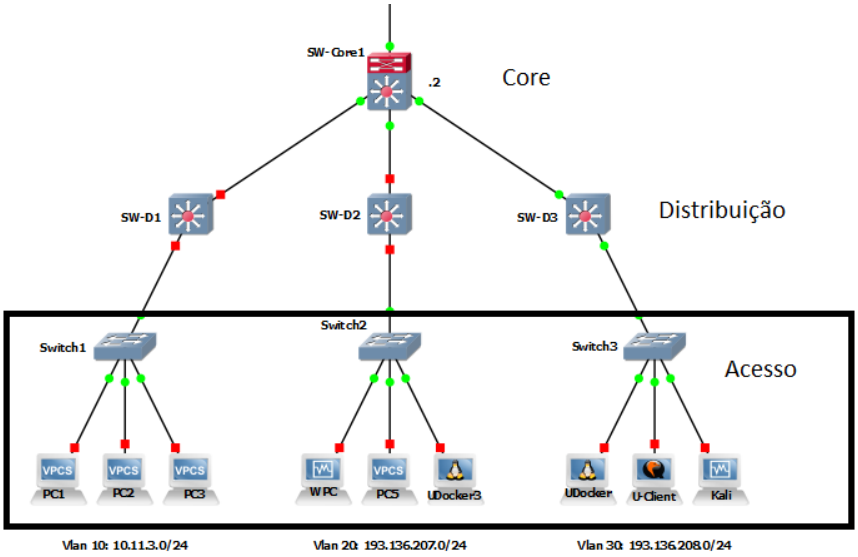

Figura 4.2: Modelo de 3 camadas Cisco.

A camada *Core*, consiste num *switch core* que é considerado a "espinha dorsal"da Rede 1. O seu objetivo é agregar e fazer o roteamento de todo o tráfego entre as diferentes partes da rede. No entanto, embora contenha os protocolos da camada 2, este *switch* não está limitado à camada de ligação do modelo OSI, pois fornece suporte a vários protocolos de camadas

superiores, sendo considerado um *switch multilayer*. Neste caso em concreto, o *switch core* está encarregue da configuração e comunicação das diferentes VLANs, da comunicação com o *router* 1 através do protocolo IPv4 e da comunicação com os *switches* da camada Distribuição. O protocolo ARP (*Address Resolution Protocol*) é usado no *switch core* para localizar o endereço MAC (*Media Access Control*) do *host* destino na rede. Se a associação do endereço IP com o endereço MAC não existir na tabela chamada "ARP *cache*"do *host*, o protcolo ARP será usado para formá-la, ao enviar uma mensagem *broadcast* para todos os *hosts* da rede, informando o endereço IP do *host* para o qual a mensagem é destinada. O *switch core* apresenta também, um serviço de DHCP (*Dynamic Host Configuration Protocol*) embutido, com a função de atribuir os detalhes da configuração de rede, como o endereço IP, a *subnet mask*, a *default gateway* e o servidor DNS (*Domain Name System*) automaticamente aos *hosts* da rede.

Os *switches* de distribuição, situados na segunda camada do modelo da Cisco, têm a função de fazer o roteamento entre as VLANs e conectar a camada de Acesso com a camada *Core*. Ao contrário do *switch core*, que possui a capacidade de abranger diversos protocolos de camadas superiores, os *switches* de distribuição estão limitados à camada 2 do modelo OSI, pois têm o propósito de analisar e encaminhar *frames* que cruzam a rede. Utilizam o endereço MAC para determinar o caminho por onde os *frames* devem ser encaminhados. Durante o processo de encaminhamento de *frames*, os *switches* de distribuição criam tabelas associando endereços MAC às interfaces pelas quais eles foram aprendidos. A tabela é gerada à medida que os *hosts* enviam *frames* para a interface do *switch* à qual se encontram conectados. Então, os *switches* de distribuição relacionam o endereço aprendido com a interface por onde foram recebidos. Os *switches* de distribuição ficam continuamente a atualizar as suas tabelas, com o objetivo de manter a conetividade o mais íntegra possível. Conforme *hosts* são adicionados ou removidos da rede, os *switches* de distribuição dinamicamente procedem com a atualização das tabelas. Caso um *host* seja removido da rede, os dados correspondentes a ele na tabela são removidos após um período predeterminado de tempo.

A camada de Acesso, tem o propósito de controlar e fornecer o acesso de diferentes grupos de utilizadores aos recursos da rede e de conectar os dispositivos finais com a camada de Distribuição. Assume que a maioria dos recursos necessários aos utilizadores deve estar disponível localmente. Esta camada também desempenha o papel de gerir e monitorizar os dispositivos finais ao coletar informações sobre o estado e o desempenho para o diagnóstico de problemas. Os *switches* da camada de acesso tem a função de agregar várias portas numa única conexão para a camada de Distribuição. Isso ajuda a reduzir a complexidade da rede

e otimiza a utilização dos recursos.

Data center

Na Rede 1, foi implementado um *data center* com a finalidade de fornecer diferentes tipos de serviços e aplicações para armazenar e gerir dados e informações de forma centralizada. Atualmente os *data centers* são indispensáveis para o funcionamento de muitas organizações, o que faz com seja um alvo atraente para a maioria dos invasores. Como é possível observar pela figura 4.3, o *data center* da Rede 1 é composto por um servidor *Web*, servidor DNS e por um servidor de arquivos.

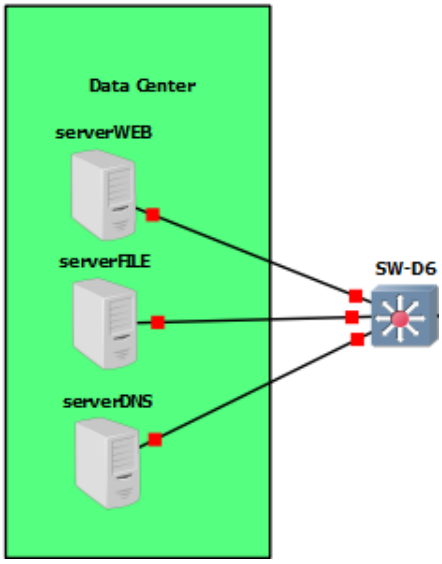

Figura 4.3: *Data center*.

Para simulação do servidor *Web* foi usado o *Metasploitable*, que consiste num sistema operativo que foi concebido para fins de treino de cibersegurança. Ele foi criado como uma plataforma deliberadamente vulnerável que permite realizar testes de penetração e avaliar os danos causados por diversos ciberataques. Por outras palavras, o *Metasploitable* é uma máquina virtual vulnerável criada de propósito para explorar ameaças cibernéticas. A pagina principal do servidor *Web Metasploitable*, representada na figura 4.4, é acedida por um *host* pertencente à VLAN 30, pelo que podemos concluir que o servidor foi implementado com sucesso. No entanto, embora o servidor apresente cinco aplicações *web*, apenas a *Multillidae* foi utilizada.

O servidor DNS, foi implementado através de um *Docker container*. O *Docker* é uma ferramenta usada para automatizar a implantação de aplicações em *containers* leves, permitindo que as aplicações funcionem de forma eficiente em diferentes ambientes, neste caso, o

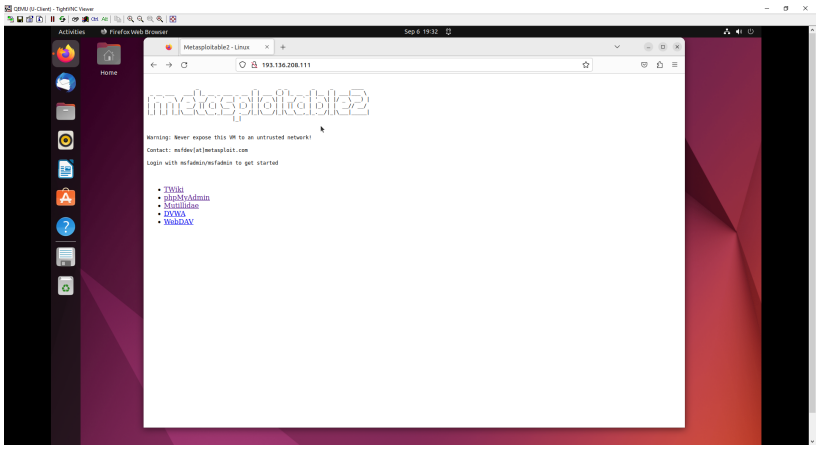

Figura 4.4: Interface do servidor *Web Metasploitable*.

sistema operativo utilizado foi o *Ubuntu Linux*. Possui também um tempo de inicialização mais rápido e desempenho melhor e mais consistente em comparação com máquinas virtuais. Um *container* é um pacote de software que inclui todas as dependências necessárias para executar a aplicação. Vários *containers* podem ser executados no mesmo *hardware* e cada *container* é mantido num ambiente isolado. Para simular o servidor DNS, foi implementado, o *software DNSmasq* que tem a capacidade de gerir redes de tamanho médio e pequeno, podendo atuar como um servidor DNS ou DHCP. A utilização do servidor DNS tem como finalidade, associar nomes de domínios mais facilmente memorizáveis a endereços IP necessários à localização e identificação de serviços e dispositivos, processo esse denominado por "resolução de nome". Por norma, o servidor DNS usa o protocolo UDP (*User Datagram Protocol*) na porta 53 para servir as solicitações e as requisições. Os nomes de domínios locais definidos no servidor estão representados na figura 4.5. Desta forma, a comunicação entre os *hosts* da camada de Acesso e o serviços da rede será mais fácil e intuitiva.

A figura 4.6, mostra um dispositivo da camada de Acesso a comunicar com servidor *Web Metasploitable* através do protocolo ICMP (*Internet Control Message Protocol*), fazendo uso do nome de domínio definido para o servidor. Neste caso, o nome atribuído foi *webServer.deec* e como podemos observar pela figura a comunicação foi bem sucedida.

Por meio de uma ferramenta de captura e análise de tráfego como *Wireshark*, somos capazes de verificar a comunicação do servidor DNS com o dispositivo da camada de Acesso durante o processo de comunicação entre o dispositivo e o servidor *Web Metasploitable*. Como demonstrado na figura 4.7, o dispositivo (IP = 193.136.208.106) envia uma consulta padrão ao servidor DNS (IP = 193.136.208.110) que é respondida com sucesso pelo servidor DNS. Podemos então concluir, que a implementação do servidor DNS foi bem sucedida.

Para a implementação do servidor de arquivos, foi utilizado uma máquina virtual com

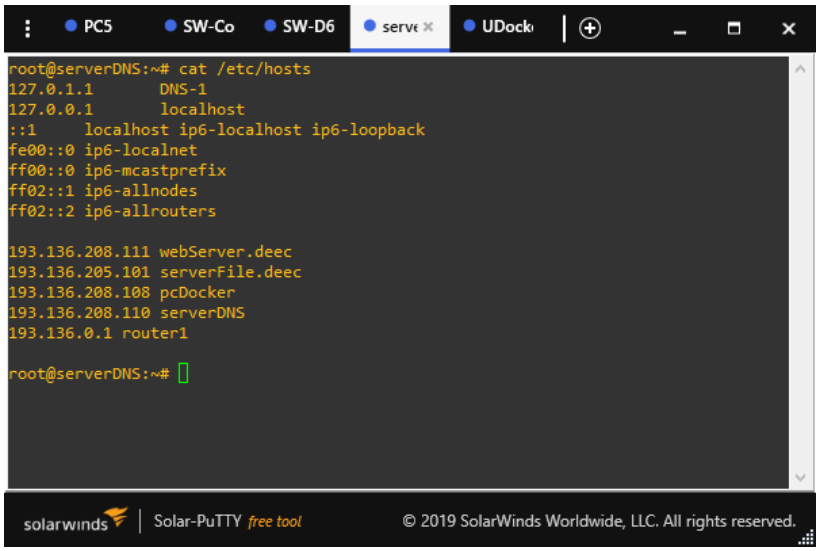

Figura 4.5: Servidor DNS *Web*.

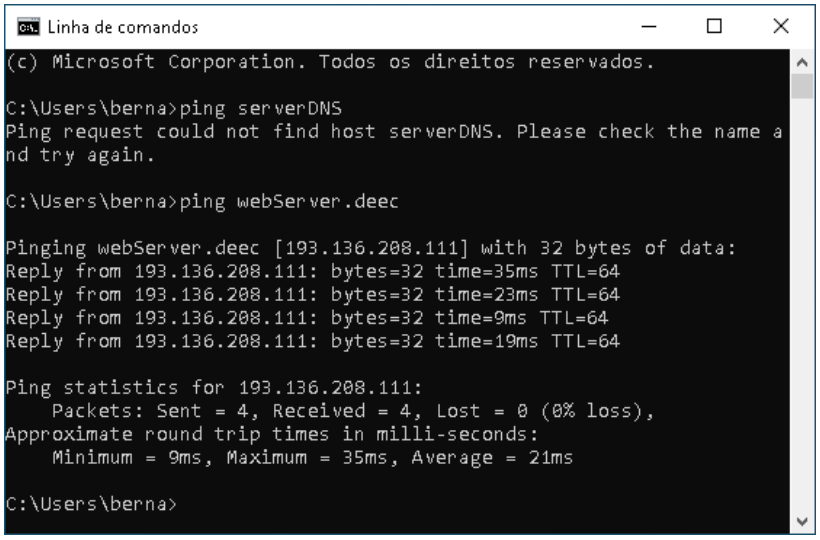

Figura 4.6: Comunicação com *webServer.deec*.

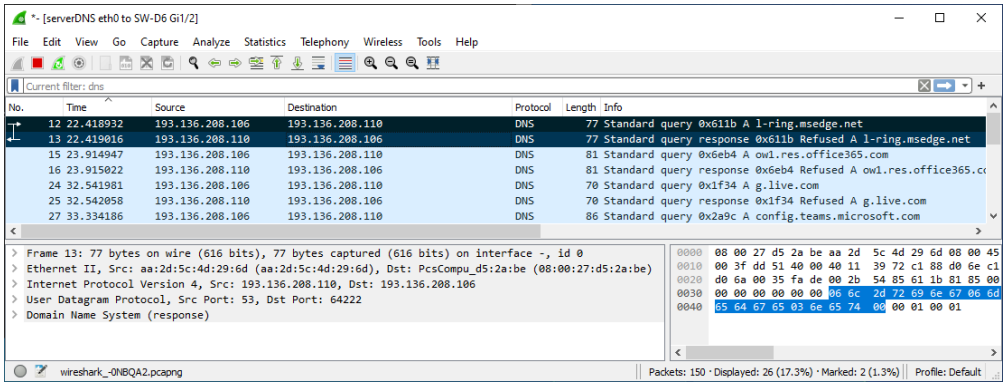

Figura 4.7: Captura do protocolo DNS.

o servidor *Ubuntu 22.04.3 LTS*. O *software SAMBA* foi adotado no servidor para permitir compartilhamento de arquivos entre os diferentes dispositivos da rede, independentemente do seu sistema operativo. Os utilizadores da rede podem assim, aceder a pastas e arquivos compartilhados, podendo até apagá-los ou acrescentar novos dados. Sendo assim, como mostra a figura 4.8, foi criado a pasta *deec_ alunos* que contem cinco ficheiros de texto com uma lista de alunos de diferentes anos.

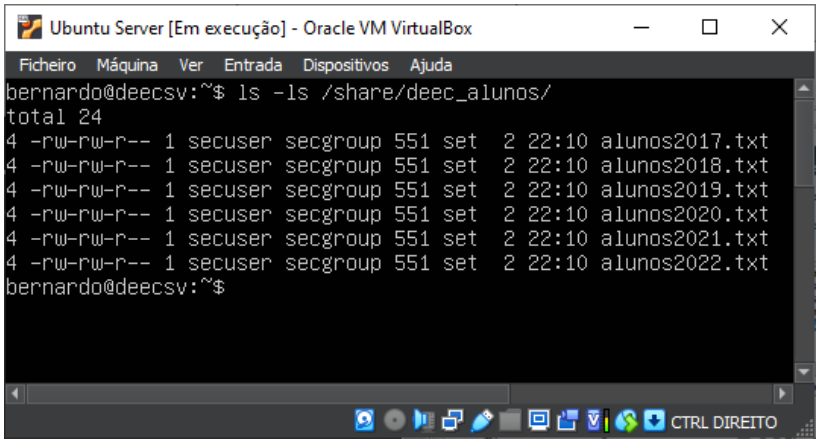

Figura 4.8: Servidor de arquivos.

Na figura 4.9 é possível verificar que dispositivos de diferentes sistemas operativos, neste caso *Linux* e *Microsoft Windows*, são capazes de aceder à pasta *deec_ alunos* e visualizar os ficheiros. Constatamos então, que o servidor de arquivos foi implementado com sucesso.

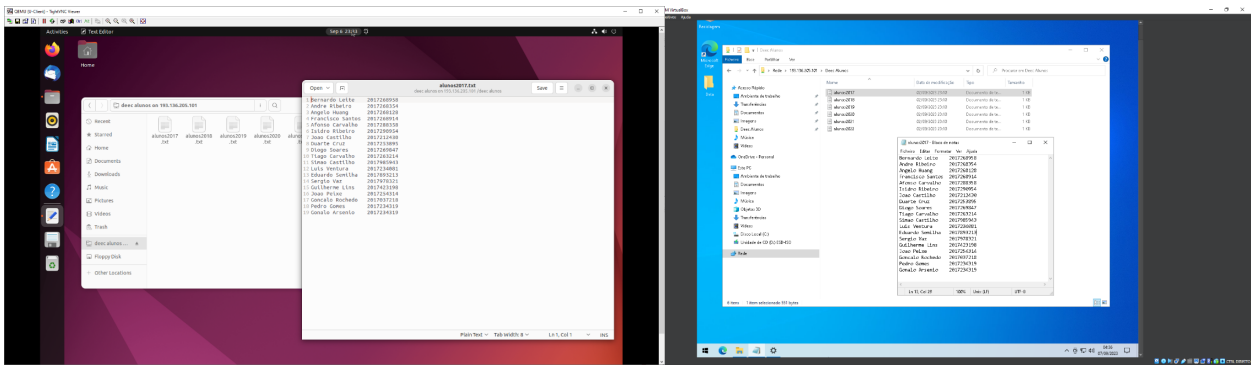

Figura 4.9: Servidor de arquivos acedido por diferentes sistemas operativos.

Router e Bloco NAT

De modo a orientar e direcionar os dados da Rede 1 para outras redes, foi implementado o *router Cisco* 7200, pois oferece uma ampla variedade de recursos e capacidades, tornando-o adequado para uma diversidade de cenários de rede. O *router* é responsável pelo encaminhamento de pacotes de dados e pela comunicação de dispositivos das diferentes redes. Foi

também acrescentado um bloco NAT (*Network Address Translation*), como demonstrado na figura 4.10, a uma das interfaces do *router* para fazer a tradução dos endereços IP da rede com a intenção de os tornar endereços roteáveis.

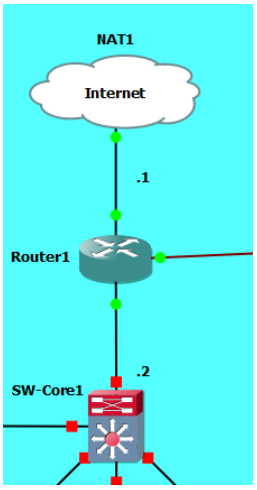

Figura 4.10: Ligação do bloco NAT a uma das interfaces do *router*.

O bloco NAT vai utilizar uma tabela com endereços IP e portas para encaminhar tráfego entre os *hosts* da rede interna e os *hosts* de uma rede externa, neste caso a *Internet*. Para que seja possível aceder os recursos na *Internet*, os endereços da rede local precisão de ser substituídos por endereços que a *Internet* compreenda. De salientar, que o *router* presente na Rede 1, também está encarregue de fornecer os recursos da *Internet* às outras redes da topologia.

4.1.2 Rede 2

A Rede 2 foi implementada, tendo por base a Rede 1. Quer isto dizer, que também foi adotado o mesmo modelo de 3 camadas criado pela *Cisco*. Logo por essa razão, os *switches* da Rede 2 funcionam do mesmo modo que os *switches* da Rede 1. Como se pode observar pela figura 4.11, a Rede 2 apresenta uma estrutura idêntica à da Rede 1 com a exceção do *data center* que não existe.

O número de VLANs também foi reduzido comparativamente à Rede 1, tendo apenas a VLAN 40 e a VLAN 50. O *router* presente na Rede 2 é o mesmo que foi implementado na Rede 1 (*Cisco* 7200) logo ele tem a função de permitir a conexão da Rede 2 com as restantes redes. Os detalhes da configuração da Rede 2 são atribuídos automaticamente pelo serviço de DHCP embutido no *switch core*.

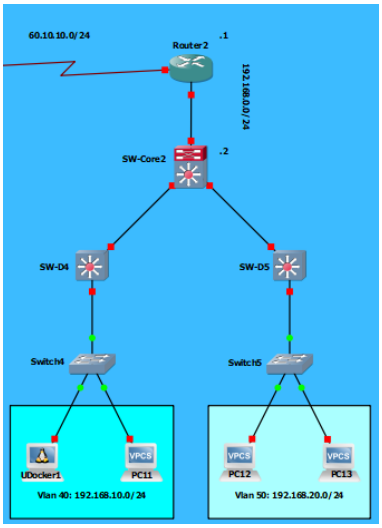

Figura 4.11: Rede 2.

4.1.3 Rede 3

Para a implementação da Rede 3 foi utilizado apenas dois *switches*, sendo que nenhum deles tem o papel de um *switch core*. Em contrapartida, foram implementados 2 *routers* (que por sinal são iguais aos que foram usados nas outras redes), como demonstrado na figura 4.12, através do protocolo HSRP (*Hot Standby Routing Protocol*) que possibilita um grupo de *routers* operar como um único *router* virtual.

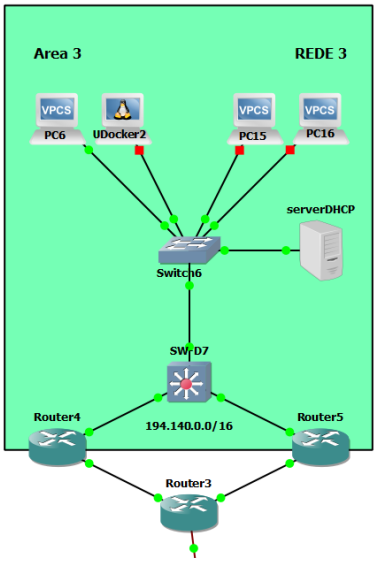

Figura 4.12: Rede 3.

Assim, podemos ter dois ou mais *routers* agrupados logicamente respondendo a um mesmo endereço IP e MAC. O HSRP é um protocolo proprietário *Cisco*, que trabalha em modelo *active* e *standby*, ou seja, um dos *routers* pertencentes ao grupo HSRP será eleito o *router* ativo e atuará no encaminhamento de pacotes dos *hosts* para as outras redes e

vise-versa. Dos *routers* remanescentes (neste caso só existe um), um deles será colocado no modo *standby* e os outros no modo *listen*. Os *routers* num modo diferente de *active* não encaminham pacotes originados pelos *hosts* com destino a outras redes, mas podem atuar no encaminhamento dos pacotes originados por outras redes com destino aos *hosts*. Os hosts da rede são configurados apenas com o endereço IP do grupo HSRP, e o processo de resolução ARP para este IP fará com que o *router* no modo *active* encaminhe ao *host* o MAC do grupo HSRP, permitindo a comunicação entre ambos. Se o *router* que se encontra no modo *active* (neste caso o *router* 5) ficar indisponível, o *router* que está no modo *standby* (neste caso o *router* 6), assumirá instantaneamente o papel de *active*, permitindo que o fluxo de dados prossiga sem qualquer intervenção manual e com o mínimo de impacto para os *hosts* da rede. Isto só é possível porque, os *routers* pertencentes ao mesmo grupo HSPR trocam mensagens de controlo. Se o *router* que está no modo *standby* deixar de receber mensagens do *router active*, ele considerará que o mesmo encontra-se inacessível e alterará o seu status para *active*.

Servidor DHCP

A Rede 3, apresenta também um servidor DHCP responsável por fornecer automaticamente os detalhes necessários para a configuração da rede nos *hosts*. Este servidor foi implementado de forma similar ao servidor DNS da Rede 1. Isto significa que, foi utilizado um *Docker container* com o sistema operativo *Ubuntu Linux*, onde também foi implementado o *software DNSmasq*. No instante em que um *host* se conecta à rede, este envia um pacote UDP destinado a todos os *hosts* da rede, com uma requisição DHCP para a porta 67. Posteriormente, o servidor DHCP que captura esse pacote responderá para a porta 68 do *host* requisitante com um pacote com as configurações essenciais da rede. Através da ferramenta *Wireshark*, como mostra a figura 4.13, conseguimos verificar o momento em que o servidor DHCP responde com sucesso a uma requisição de um *host* da Rede 3. Podemos então concluir, que o servidor DHCP foi implementado com sucesso.

4.1.4 Processo de roteamento

Para conseguirmos transmitir dados de uma rede IP para outra, um *router* precisa de fazer o direcionamento dos pacotes, analisado os seus cabeçalhos e consultando a rota para a rede IP destino na sua tabela de roteamento. Sempre que um pacote sai de uma rede para a outra, um *router* está sempre por de trás do processo. A forma como o pacote

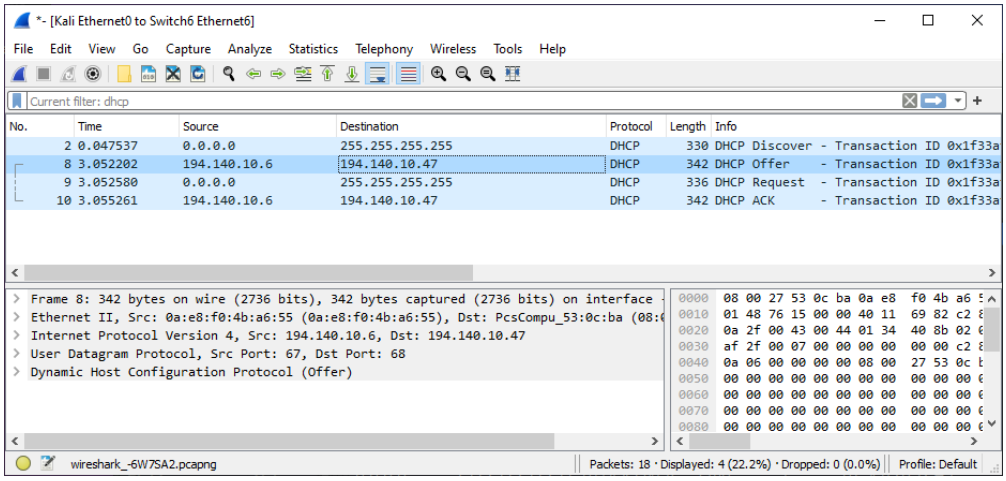

Figura 4.13: Captura do protocolo DHCP.

de dados gerado por um *host* de uma rede deve ser encaminhado para um *host* de outra é determinada pelo processo de roteamento. Nesta topologia, foi adotado um processo de roteamento dinâmico. O processo de roteamento dinâmico utiliza protocolos de roteamento para mapear a rede e atualizar dinamicamente as tabelas de roteamento dos *routers*. Este processo é menos complexo comparativamente ao processo estático, pois simplifica bastante o processo de configuração da rede e é mais viável e recomendável em redes de médio e grande porte.

O protocolo adotado para esta topologia de rede é do tipo *link state*. Os protocolos deste tipo, exercem métricas muito mais eficientes e complexas para determinar o melhor caminho para uma rede remota. Eles produzem e gerem três tabelas distintas. Uma dessas tabelas contem informações sobre a topologia lógica de toda a rede. Outra mantém informações sobre todos os *routers* diretamente conectados e a última seria a tabela de roteamento propriamente dita. A forma como é determinado o melhor caminho para uma rede remota, por um protocolo do tipo *link state* é através análise da largura de banda disponível entre a origem e o destino, escolhendo sempre o caminho com mais banda disponível.

Protocolo OSPF

De modo a conseguirmos fazer a comunicação entre as diferentes redes da topologia, foi implementado o protocolo OSPF (*Open Shortest Path First*). O protocolo OSPF permite a hierarquização da rede através da sua divisão em domínios de roteamento, denominados áreas. As áreas são usadas para controlar a forma de como as informações de roteamento devem ser compartilhadas na rede. A área 0, também designada por *backbone area* representa o núcleo de uma rede OSPF. Essa área tem obrigatoriamente de existir numa rede OSPF,

pois todas as áreas remanescentes devem se conectar a ela. Caso contrario a rede OSPF não funcionará corretamente. Na figura 4.14, encontra-se representada a área 0 que foi implementada. Foi também, implementado mais 3 áreas que correspondem às diferentes redes existentes, ou seja, cada rede pertence a uma área distinta, que se encontra conectada à área 0.

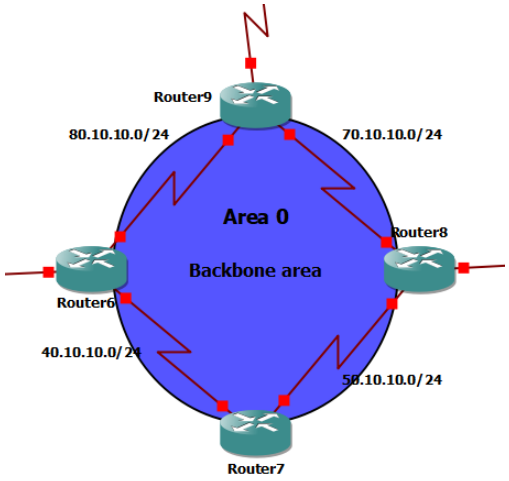

Figura 4.14: Área 0 implementada na topologia.

O tráfego entre áreas é coordenado por um *router* de interconexão, designado por *router* de borda de área (*Area Border Router* ou ABR). Na figura 4.15, é possível analisar a tabela de roteamento do *router* ABR da topologia implementada, neste caso, o *router* definido para esse efeito foi o *router* 6.

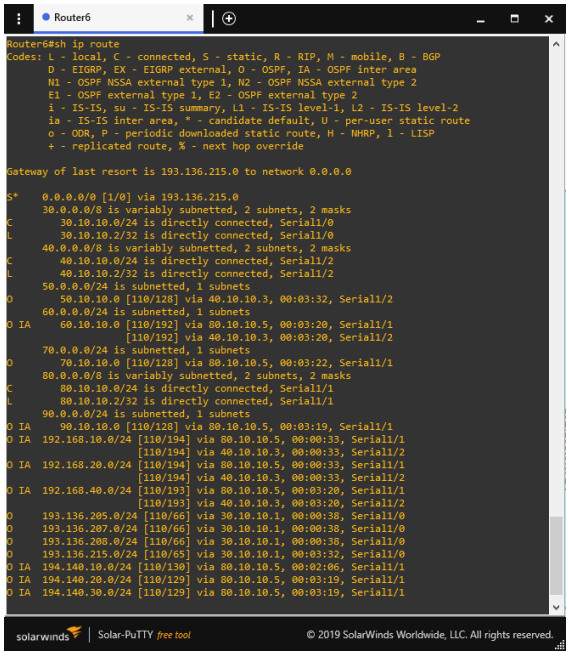

Figura 4.15: Tabela de roteamento do *router* ABR.

O único critério adotado pelo OSPF durante o processo de seleção da melhor rota para uma rede remota é o custo. Quanto menor o custo, melhor o caminho. O custo é inversamente proporcional à largura da banda de um *link* ou interface, ou seja, quanto maior a largura de banda, menor será o custo equivalente. O custo associado a cada interface OSPF é incluído na tabela topológica, e o custo total de uma rota é dado pela soma dos custos das interfaces no seu caminho. Para o cálculo de custo por interface, a *Cisco* adotou a seguinte fórmula:

$$Custo = \frac{10^8}{Largura\ de\ banda\ da\ interface} \tag{4.1}$$

O valor 10^8 corresponde à largura de banda de referência em *bits* por segundo. Assumindo esta fórmula, uma interface que opere a 100 *Mbps* teria o custo OSPF de 1.

Antes de enviar informações de roteamento, o protocolo OSPF necessita de criar uma "parceria"com os *routers* vizinhos. Basicamente, *routers* OSPF que partilhem o mesmo segmento podem construir uma relação de vizinhança. O estabelecimento desta relação ocorre por intermédio da mensagem *"Hello"*. Assim que identificam o "*router* id"listado no pacote "*Hello*"envidado pelo *router* vizinho, os *routers* OSPF tornam-se oficialmente vizinhos.

5 Cenários de ataque

Neste capitulo, é feita uma análise detalhada dos ciberataques que foram realizados, dos seus respetivos cenários aplicação bem como dos impactos e danos causados. Para a simulação dos ciberataques, foi utilizado o *Kali Linux*. Sendo este uma distribuição *Linux* especializada e uma das mais conhecidas na área da cibersegurança e testes de penetração. O *Kali Linux* dispõe de inúmeros *softwares* pré-instalados para atender os mais diversos propósitos.

5.1 TCP SYN Flood

O ataque *TCP SYN Flood*, é um tipo de ataque DoS que tem como alvo o processo de *handshake* TCP (*Transmission Control Protocol*), utilizado para estabelecer uma conexão entre dois *host* de uma rede. O objetivo principal deste ataque é sobrecarregar um *host*, enviando-lhe uma quantidade elevada de solicitações TCP SYN (*synchronization*). Desta forma, os recursos do *host* alvo são consumidos ao ponto de torná-lo inacessível. A figura 5.1, ilustra o cenário onde foi aplicado o ataque.

Neste cenário, um *host* da Rede 2 com o endereço IP 192.168.20.101 pertencente à VLAN 50, quer comprometer os recursos de um *host* com endereço IP 193.136.208.102 localizado na VLAN 30 de Rede 1. Para tal, foi utilizado inicialmente o *software Nmap*, como demonstrado na figura 5.2, com objetivo de descobrir quais as portas disponíveis e quais os serviços estão a ser executados no *host* alvo.

Para proceder à execução do ataque, foi utilizado a ferramenta de rede *H3ping* pois oferece uma grande variedade de recursos para exploração e manipulação de redes. O comando executado nessa ferramenta com o propósito de comprometer o *host* da Rede 1 foi o *-c 100000 -d 100 -S -p 23 –flood –rand-source 193.136.208.102*. Este comando, conforme escrito, envia 100000 pacotes SYN (onde cada pacote ocupa 100 bytes) para porta 23 (usada para o serviço Telnet) o mais rápido possível para o endereço IP de destino (193.136.208.102).

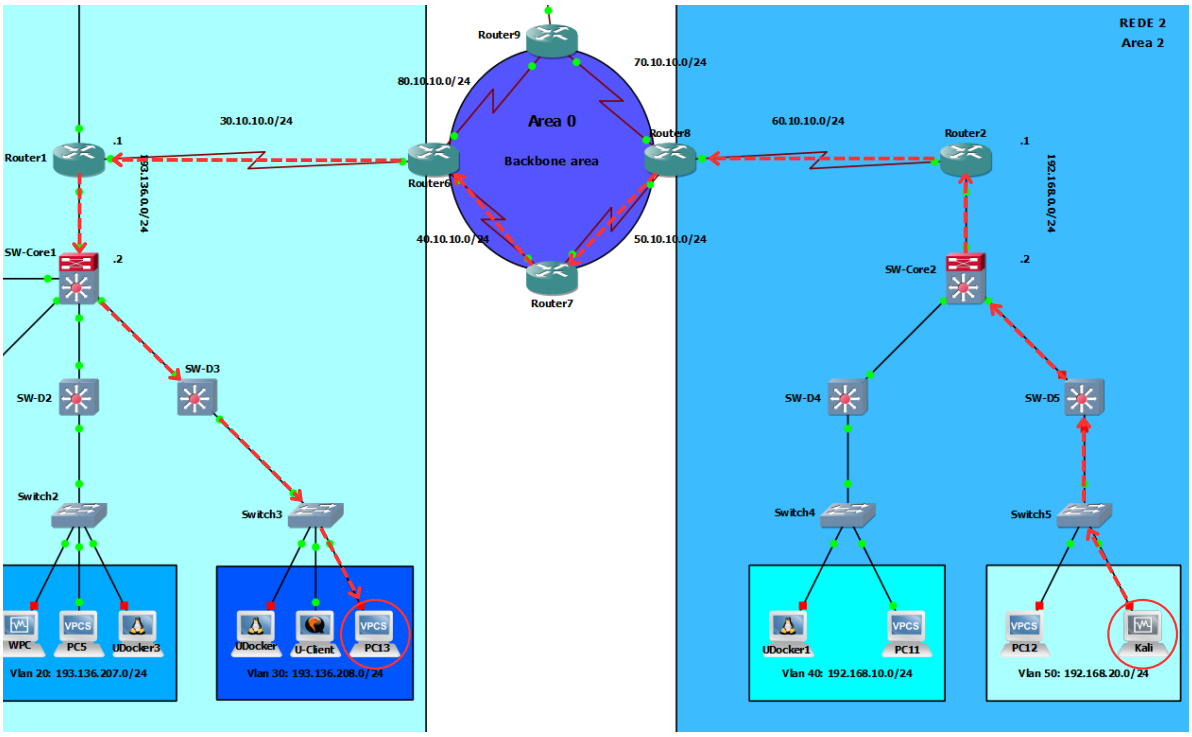

Figura 5.1: Cenário de aplicação do ataque *TCP SYN Flood*.

```
root@kali: ~
File  Actions  Edit  View  Help
Nmap done: 1 IP address (1 host up) scanned in 11.70 seconds

┌──(root㉿kali)-[~]
└─# nmap 193.136.208.102
Starting Nmap 7.93 ( https://nmap.org ) at 2023-09-07 18:59 EDT
Nmap scan report for 193.136.208.102
Host is up (3.3s latency).

PORT      STATE    SERVICE
1/tcp     open     tcpmux
3/tcp     open     compressnet
4/tcp     open     unknown
6/tcp     open     unknown
7/tcp     open     echo
9/tcp     open     discard
13/tcp    open     daytime
17/tcp    open     qotd
19/tcp    open     chargen
20/tcp    open     ftp-data
21/tcp    open     ftp
22/tcp    open     ssh
23/tcp    open     telnet
24/tcp    open     priv-mail
25/tcp    open     smtp
26/tcp    open     rsftp
30/tcp    open     unknown
32/tcp    open     unknown
33/tcp    open     dsp
37/tcp    open     time
42/tcp    open     nameserver
43/tcp    open     whois
49/tcp    open     tacacs
53/tcp    open     domain
70/tcp    open     gopher
79/tcp    open     finger
80/tcp    open     http
81/tcp    open     hosts2-ns
82/tcp    open     xfer
83/tcp    open     mit-ml-dev
84/tcp    open     ctf
85/tcp    open     mit-ml-dev
```

Figura 5.2: Scan de portas e serviços do *host* alvo.

A *flag -rand-source* foi usada para criar endereços de origem aleatórios, dificultando assim o rastreamento da origem do ataque. Através da figura 5.3, é possível verificar o envio das várias solicitações TCP SYN de diferentes IPs de origem.

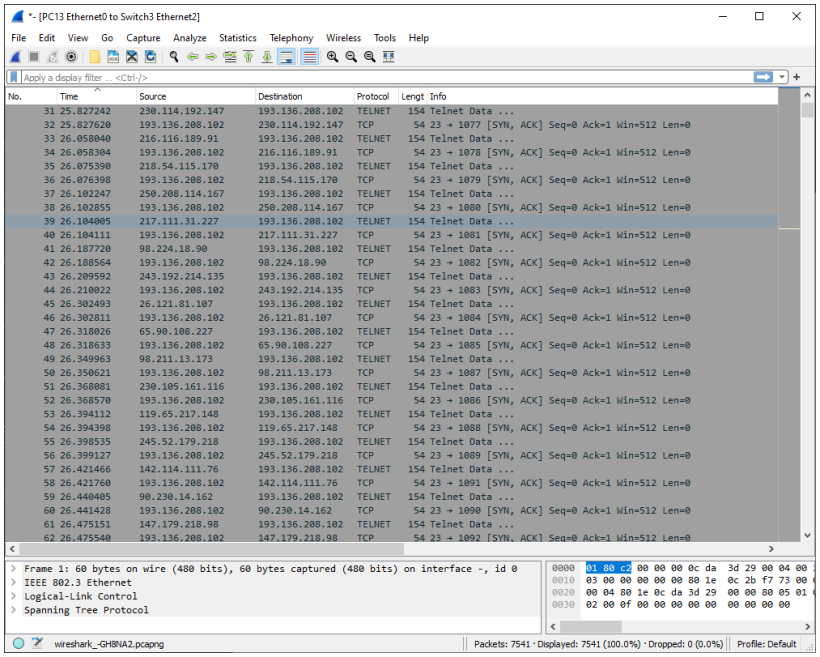

Figura 5.3: Tráfego capturado durante o ataque TCP SYN Flood.

O gráfico da figura 5.4, mostra-nos o número de pacotes recebidos pelo *host* alvo durante a execução do ataque. Analisando o gráfico, detetamos que houve um aumento significativo no número de pacotes recebidos num curto espaço de tempo. No entanto, apesar do ataque ter sido bem sucedido, não existiu uma grande consistência no número de pacotes recebidos ao longo do tempo. Isto deve-se ao facto da máquina que executou o ataque ter recursos limitados ao nível de *hardware* o que fez com que o envio de pacotes para o *host* alvo não fosse consistente.

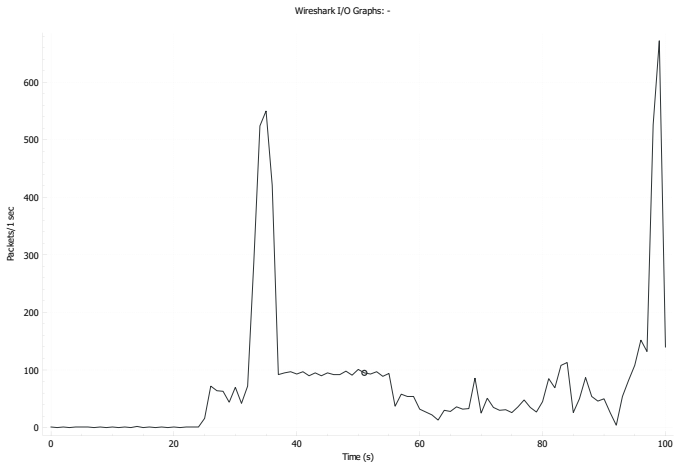

Figura 5.4: Gráfico do número de pacotes recebidos pelo *host* alvo durante o ataque TCP SYN *Flood*.

5.2 CAM table overflow

O ataque CAM (*Content Addressable Memory*) *table overflow*, é um tipo de ataque que tem como alvo as tabelas CAM de um *switch* da rede. Essas tabelas são usadas para armazenar a associação entre os endereços MAC e a interface correspondente. Por norma, um *switch* atualiza a sua tabela CAM dinamicamente à medida que aprende quais endereços MAC podem ser acedidos através de cada uma das suas interfaces. O principal objetivo deste ataque é inundar o *switch* com vários *frames*, cada um contendo um diferente endereço MAC de origem falsificado. A figura 5.5, ilustra o cenário onde foi aplicado o ataque.

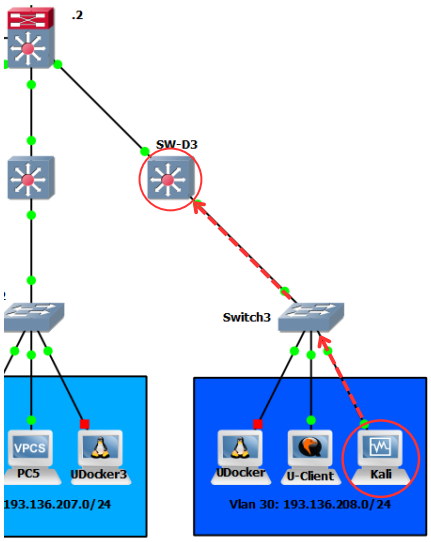

Figura 5.5: Cenário de aplicação do ataque CAM *table overflow*.

Neste cenário, o *host* localizado na VLAN 30 da Rede 1 com endereço IP 193.136.208.101

pretende fazer a negação de serviços do *switch* de distribuição que dá acesso à VLAN 30. Podemos verificar pela figura 5.6, que antes da execução do ataque o *switch* apenas tem conhecimento de 5 endereços MAC da VLAN 30.

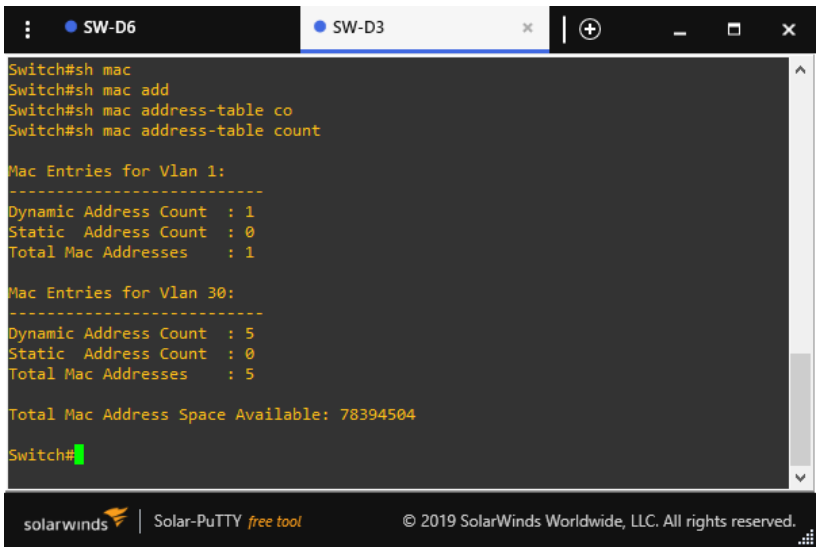

Figura 5.6: Número de endereços MAC antes do ataque.

Para dar início à execução do ataque, foi aplicado o comando *macof -i eth0 -n 10000000* através da consola do *host* responsável pela realização do ataque. Quando este comando é executado, é gerado um elevado número de *frames*, cada um com um endereço MAC de origem falsificado, que serão enviados através da interface de rede *eth0*. A *flag -n* define o número de *frames* a ser enviado. Podemos observar pela figura 5.7, que o durante o período que ocorreu o ataque o *switch* de distribuição tomou conhecimento de mais 150000 novos endereços MAC.

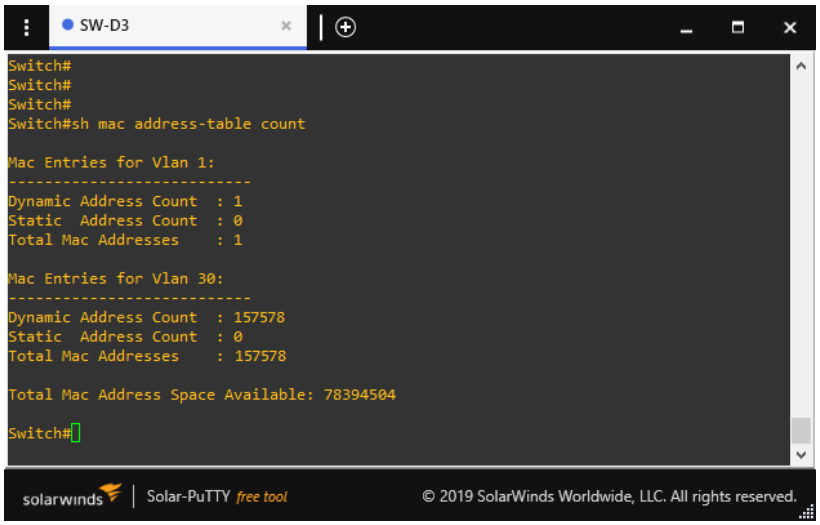

Figura 5.7: Número de endereços MAC após o ataque.

O gráfico da figura 5.8, demonstra-nos o número de *frames* que foram recebidos pelo

switch alvo ao longo de aproximadamente 45 segundos em que ocorreu o ataque. Examinando o gráfico, é possível detetar que o ataque em questão foi extremamente intenso e consistente, tendo o*switch* recebido em média 3500 *frames* por segundo.

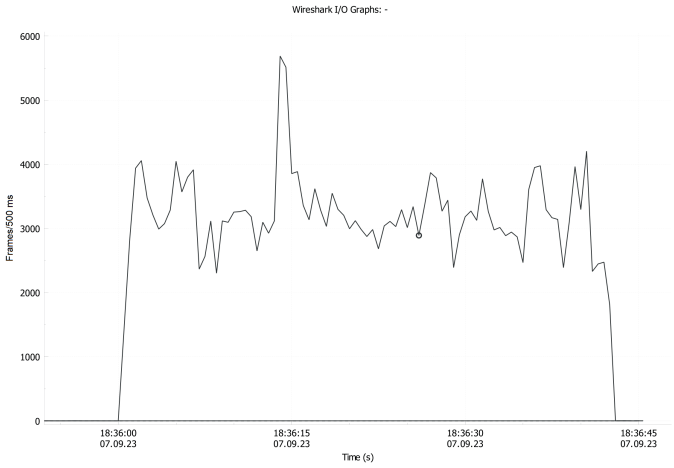

Figura 5.8: Gráfico do número de *frames* recebido pelo *switch*

O impacto deste ataque, levou à interrupção momentânea da comunicação dos *hosts* localizados na VLAN 30, uma vez que o *switch* não conseguiu tomar decisões de encaminhamento por conta do elevado número de *frames* recebidos, que por sua vez levaram à negação do serviço.

5.3 ARP Poisoning

O ARP *poisoning*, é tipo de ciberataque em que o invasor manipula as mensagens ARP numa rede local. Como já foi explicado anteriormente, o protocolo ARP é utilizado para descobrir o endereço MAC associado ao endereço IP do *host* que pretendemos comunicar. Num ataque ARP *poisoning*, o invasor envia mensagens ARP forjadas para a rede local, alegando que o seu endereço MAC corresponde aos endereços IP de outros *hosts* da rede. O principal objetivo é encaminhar incorretamente o tráfego destinado a esses *hosts* para a sua própria maquina. A figura 5.9, ilustra o cenário onde foi aplicado o ataque.

Neste cenário, o *host* invasor localizado na VLAN 30 da Rede 1 com endereço IP 193.136.208.101 tenciona direcionar o tráfego entre um *host* alvo, localizado na mesma VLAN, com endereço IP 193.136.208.120 e o servidor *Web Metasploitable* com endereço IP 193.136.208.111 para a sua própria maquina. Para tal, foi usado o *software Ettercap* que contem várias funcionalidades de análise de rede e de ataques *Man in the Middle*.

Como podemos analisar pela figura 5.10, que demonstra a tabela ARP do servidor *Web*

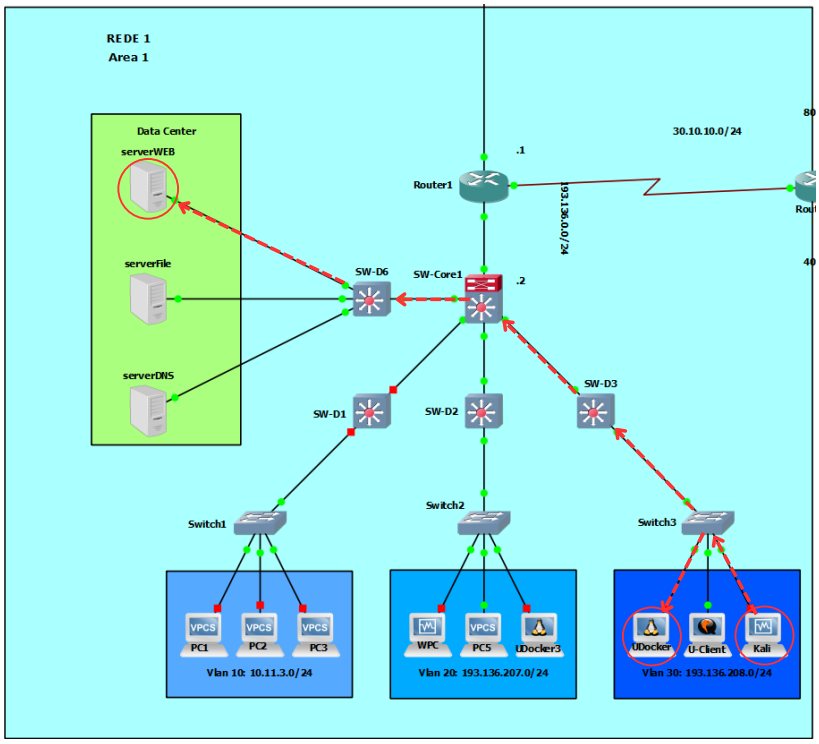

Figura 5.9: Cenário de aplicação do ataque ARP *poisoning*.

Metasploitable, o endereço MAC dos *hosts* com endereço IP 193.136.208.120 e 193.136.208.101, encontra-se distinto um do outro momentos antes da execução do ataque.

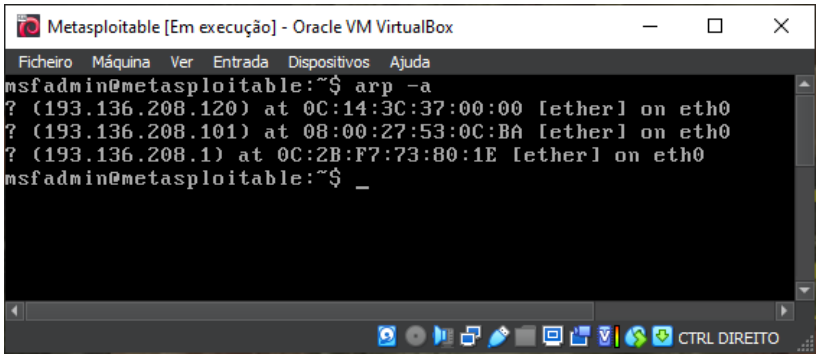

Figura 5.10: Tabela ARP do servidor *Web Metasploitable* antes do ataque ARP *poisoning*.

Para inicializar o ataque, é necessário efetuar primeiro uma busca pelos endereços IPs dos *hosts* presentes na rede. A figura 5.11, mostra o momento em que esses endereços IPs são descobertos pelo *Ettercap*.

Após a busca dos endereços IPs, definimos os dois alvos que tencionamos intercetar o tráfego de rede, como demonstrado na figura 5.12. Desta forma, estamos aptos para iniciar o ataque.

No momento em que o ataque começa, da-se então inicio à troca de mensagens ARP falsificadas. Como podemos averiguar pela figura 5.13, que demostra a tabela ARP do

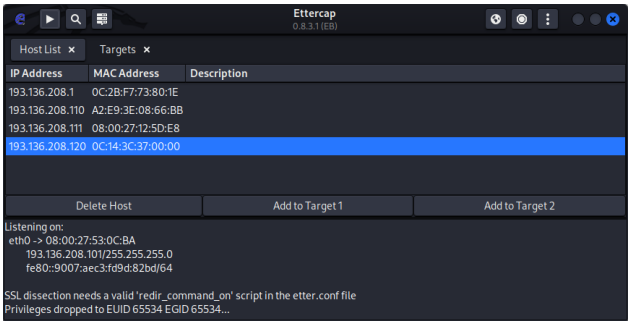

Figura 5.11: Endereços IPs capturados pelo *sofware Ettercap*.

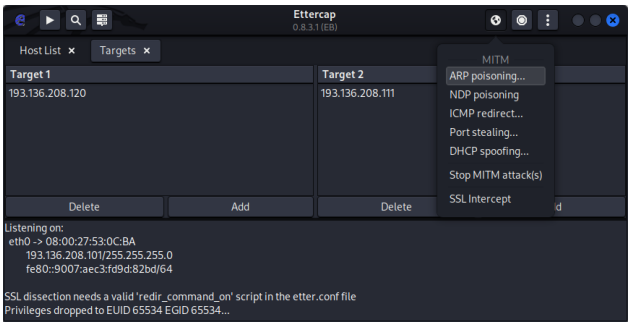

Figura 5.12: Alvos definidos para o ataque.

servidor durante o ataque, o *host* alvo com o endereço IP 193.136.208.120 adquiriu o mesmo endereço MAC do *host* invasor. Neste instante, o *host* invasor já consegue ter acesso ao tráfego entre servidor *Web Metasploitable* e o *host* alvo.

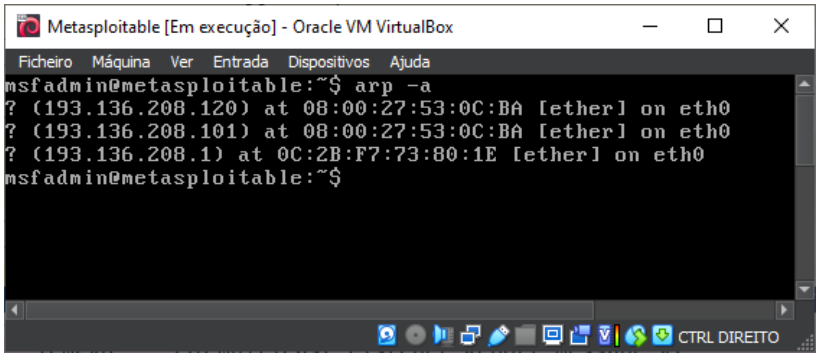

Figura 5.13: Tabela ARP do servidor *Web Metasploitable* durante o ataque ARP *poisoning*.

Se porventura, o *host* alvo tentar iniciar uma sessão na aplicação *web* do servidor *Web Metasploitable*, inserindo as suas credenciais de *login*, como demonstra a figura 5.14, o *host* invasor irá ter acesso aos dados de *login* do *host* alvo.

Através de um *sofware* de captura de tráfego como *Wireshark*, conseguimos captar o *username* e *password* usada pelo *host* alvo para aceder à aplicação *web* uma vez que essas informações não se encontram encriptadas. Na figura 5.15, podemos observar os dados de *login* através da interceção de tráfego HTTP (*Hypertext Transfer Protocol*) não encriptado

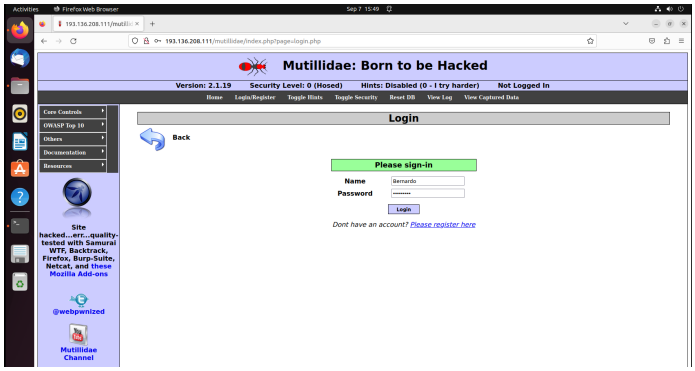

Figura 5.14: Acesso à aplicação *Multillidae* pelo *host* alvo.

pela aplicação *web*.

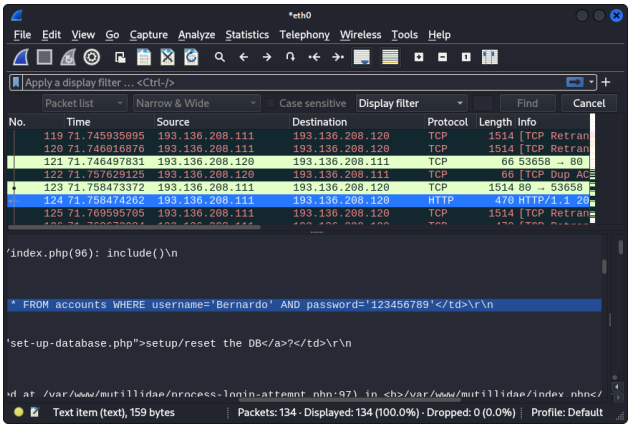

Figura 5.15: Captura de tráfego HTTP não encriptado.

O ataque ARP *poisoning* apesar de não causar danos visíveis aos serviços da rede, compromete a sua fiabilidade ao permitir o roubo e espionagem de dados sensíveis. Portanto, é necessário estar cientes desses riscos e implementar medidas adequadas, como a monitorização de tráfego ARP para mitigar e prevenir o reaparecimento de novos ataques ARP *poisoning*.

5.4 DHCP denial of service

O ataque DHCP *denial of service*, é um tipo de ataque DoS que tem como alvo servidores DHCP com objetivo de perturbar o seu funcionamento e impedir que atribuam endereços IP aos *hosts* da rede. Como automatiza o processo de atribuição de endereços IP e de configuração de rede para *hosts* que se conectam na rede, o DHCP é um serviço critico na maioria das redes. A figura 5.16, ilustra o cenário onde foi aplicado o ataque.

Neste cenário, o *host* invasor cogita comprometer o servidor DHCP da Rede 3 de forma

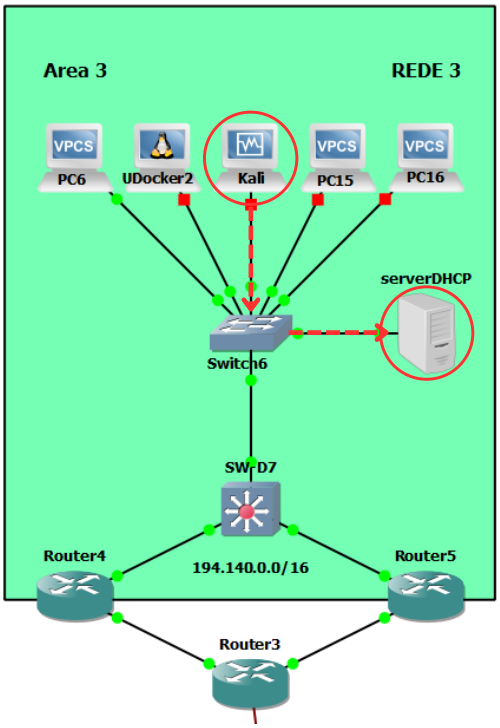

Figura 5.16: Cenário de aplicação do ataque DHCP *denial of service*.

a que mais nenhum host consiga conectar à rede via DHCP. Para esse efeito, foi usado o *software Yersina* usado para testes de penetração de rede e avaliação da segurança. O ataque consiste em inundar o servidor DHCP com várias mensagens *discovery broadcast*, fazendo com que servidor aloque uma enorme quantidade de endereços IPs. A figura 5.17, mostra os endereços IPs alocados no servidor DHCP antes da realização do ataque.

```
root@serverDHCP:~# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
194.140.10.121           ether   00:50:79:66:68:08   C                     eth0
194.140.10.122           ether   00:50:79:66:68:09   C                     eth0
194.140.10.119           ether   4e:43:32:6b:46:0c   C                     eth0
194.140.10.120           ether   00:50:79:66:68:07   C                     eth0
```

Figura 5.17: Tabela de endereços IPs alocados no servidor DHCP

Para dar início ao ataque, como mostra a figura 5.18, temos de dar *launch* no ataque que queremos efetuar escolhendo o seu protocolo. Depois disso, o *Yersina*, encarrega-se de enviar automaticamente os pacotes para o servidor DHCP.

A figura 5.19, mostra os endereços IPs alocados no servidor DHCP após a execução do ataque.

Neste ataque, o *host* invasor consegui inundar com sucesso o servidor DHCP com inúmeras mensagens de *discovery*. Na figura 5.20, é possível verificar o tráfego de rede entre o *host* invasor e o servidor DHCP. Podemos observar que os pacotes são enviados de uma fonte desconhecida (0.0.0.0) para um endereço de *broadcast* (255.255.255.255).

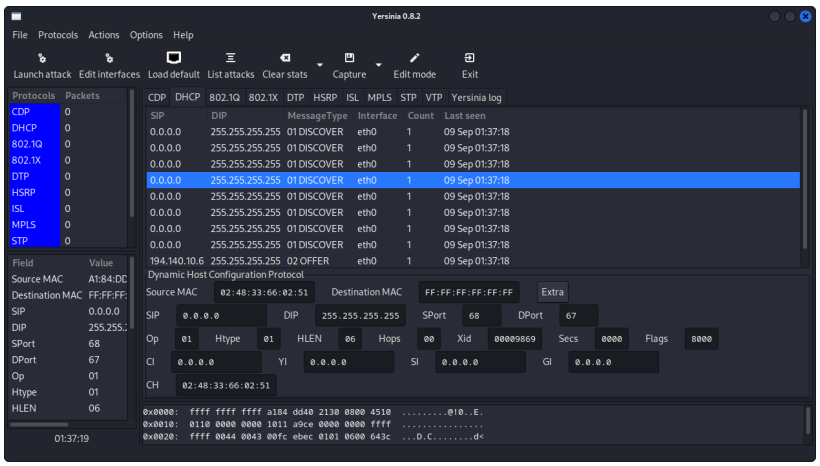

Figura 5.18: Inicialização do ataque DHCP *denial of service*

Figura 5.19: Tabela de endereços IPs alocados no servidor DHCP após o ataque

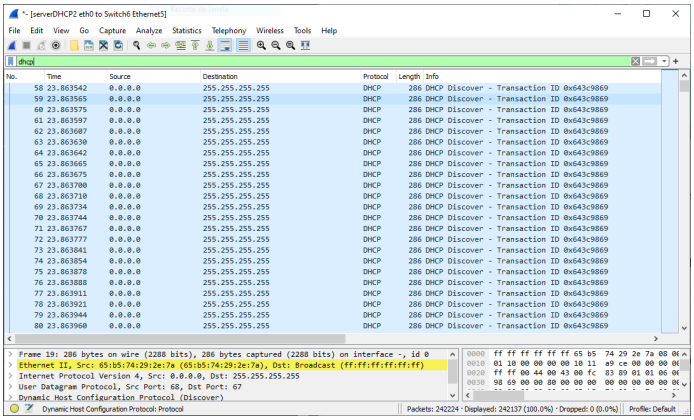

Figura 5.20: Tráfego capturado durante o ataque DHCP *denial of service*

O gráfico da figura 5.21, mostra-nos o número de pacotes recebidos pelo servidor DHCP durante o intervalo de tempo que ocorreu o ataque. O elevado número de pacotes recebidos, levou ao esgotamento dos recursos do servidor DHCP fazendo com que mais nenhum *host* da Rede 3 conseguisse conectar com a rede local. Todos os endereços IP disponíveis foram alocados, fazendo com que o servidor DHCP não consiga atender às necessidades da Rede 3. No entanto, os *hosts* que já tinham obtido endereços IP podem continuar a seu funcionamento normal.

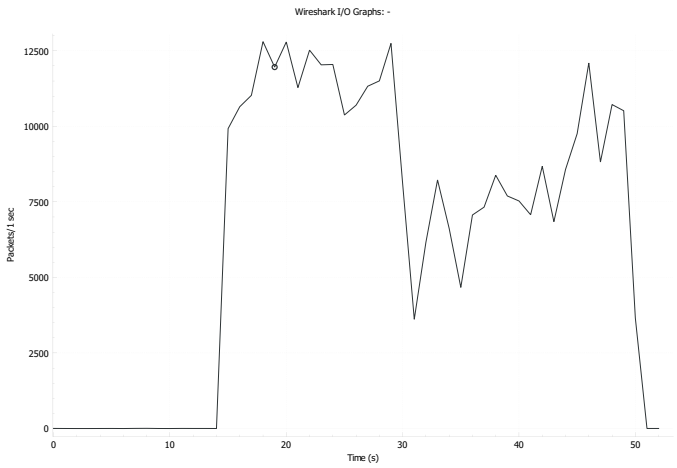

Figura 5.21: Gráfico do número de pacotes recebido pelo servidor DHCP

O impacto causado por este ataque leva a prejuízos significativos, uma vez que a interrupção de serviços críticos, como acesso à *Internet*, comunicação interna e acesso a recursos de outras redes é essencial para o bom funcionamento da rede.

6 Conclusão e Trabalho Futuro

Este capítulo visa abordar as considerações finais deste documento, no sentido de analisar, de forma conclusiva, os procedimentos experimentais realizados ao longo desta dissertação. Ainda, este capítulo termina com algumas propostas de trabalho futuro para complementar e continuar o estudo e análise de ciberataques.

6.1 Conclusão

O número de ciberameaças e de ciberataques, bem como a complexidade e a constante evolução destas, eleva os desafios e a necessidade de uma adaptação, estudo e compreensão constante. Por essa razão, a simulação de ciberataques desempenha um papel fundamental na avaliação dos impactos causados e no treino dos métodos de execução.

Toda a implementação e, posterior, trabalho experimental centrou-se na simulação e análise dos danos que podem ser causados por ciberataques numa rede real. Para tal foi desenvolvido, no simulador de redes *Graphical Network Simulator 3*, uma topologia de rede realista com o objetivo de simular vários ciberataques em diferentes tipos de cenários. Todo o processo de implementação da rede, foi documentado detalhadamente para dar a entender todas as metodologias e protocolos usados no seu desenvolvimento. Através desta topologia, foi possível simular num ambiente virtual totalmente controlado diversos cenários de ataques realistas. Assim, somos capazes de analisar os métodos utilizados, identificar vulnerabilidades existentes e os eventuais prejuízos causados na rede.

Neste trabalho, foram simulados quatro ciberataques que surgem com alguma frequência no nosso quotidiano. Sendo eles, TCP SYN *flood*, CAM *table overflow*, ARP *poisoning* e o DHCP *denial of service*. Cada um destes ciberataques foi simulado em cenários distintos. Desta forma, foi possível demonstrar e avaliar o modo de funcionamento de cada um. A captura e deteção do tráfego malicioso, bem como o impacto causado durante a simulação dos ciberataques, foi posteriormente relatado e analisado de maneira a, comprovar o sucesso

da simulação dos mesmos. Através dos resultados obtidos, verificou-se que os ciberataques simulados conseguiram prejudicar os serviços e recursos da rede ao comprometer a comunicação e integridade da topologia implementada.

Finalmente, podemos concluir que este tipo de ferramentas de simulação permitem avaliar com rigor diversos cenários de ataque de forma controlada e segura. Com base no ataque efetuado, elas fornecem informações valiosas sobre as vulnerabilidades da rede, revelando detalhes importantes que por vezes passariam despercebidos em situações reais.

6.2 Trabalho futuro

Um dos pontos de trabalho futuro passa pela implementação de mecanismos para mitigar os diferentes ciberataques simulados. Saber o método de funcionamento das ciberameaças não é suficiente para resolver o problema. É necessário desenvolver estratégias capazes de atuar de forma rápida e eficaz para que o ciberataque não cause grandes transtornos. À medida que os ciberataques se tornam mais sofisticados, existe uma necessidade ainda maior de adaptação e inovação. É crucial haver um conteste acompanhamento das tendências atuais para que se possa atuar de forma a resolver as adversidades que vão surgindo.

7 Bibliografia

[1] *Relatório Riscos Conflitos de 2022*. Observatório de Cibersegurança do CNCS, Lisboa, 3 edition, 2022.

[2] Nazli Choucri, Stuart Madnick, and Jeremy Ferwerda. Institutions for cyber security: International responses and global imperatives. *Information Technology for Development*, 20(2):96–121, 2014.

[3] David D Clark. Characterizing cyberspace: past, present and future. 2010.

[4] Adrian Venables. Modelling cyberspace to determine cybersecurity training requirements. *Frontiers in Education*, 6, 2021.

[5] Nazrul Hoque, Dhruba K. Bhattacharyya, and Jugal K. Kalita. Botnet in ddos attacks: Trends and challenges. *IEEE Communications Surveys Tutorials*, 17(4):2242–2270, 2015.

[6] Nazrul Hoque, Dhruba K. Bhattacharyya, and Jugal K. Kalita. Botnet in ddos attacks: Trends and challenges. *IEEE Communications Surveys Tutorials*, 17(4):2242–2270, 2015.

[7] K.Muthamil Sudar, P. Deepalakshmi, P. Nagaraj, and V. Muneeswaran. Analysis of cyberattacks and its detection mechanisms. In *2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pages 12–16, 2020.

[8] Anestis Karasaridis, Brian Rexroad, David A Hoeflin, et al. Wide-scale botnet detection and characterization. *HotBots*, 7:7–7, 2007.

[9] Noor Mohd Safar, Noryusliza Abdullah, Hazalila Kamaludin, Suhaimi Abd Ishak, and M. R. M. Isa. Characterising and detection of botnet in p2p network for udp protocol.

Indonesian Journal of Electrical Engineering and Computer Science, 18:1584–1595, 06 2020.

[10] Sérgio S.C. Silva, Rodrigo M.P. Silva, Raquel C.G. Pinto, and Ronaldo M. Salles. Botnets: A survey. *Computer Networks*, 57(2):378–403, 2013. Botnet Activity: Analysis, Detection and Shutdown.

[11] Ori Or-Meir, Nir Nissim, Yuval Elovici, and Lior Rokach. Dynamic malware analysis in the modern era—a state of the art survey. *ACM Comput. Surv.*, 52(5), sep 2019.

[12] Hossein Rouhani Zeidanloo, Farzaneh Tabatabaei, Payam Vahdani Amoli, and Atefeh Tajpour. All about malwares (malicious codes). In *Security and Management*, pages 342–348, 2010.

[13] Zhida Li, Ana Laura Gonzalez Rios, and Ljiljana Trajković. Detecting internet worms, ransomware, and blackouts using recurrent neural networks. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2165–2172, 2020.

[14] Monika Agrawal, Heena Singh, Nidhi Gour, and Mr Ajay Kumar. Evaluation on malware analysis. *International Journal of Computer Science and Information Technologies*, 5(3):3381–3383, 2014.

[15] Harun Oz, Ahmet Aris, Albert Levi, and A. Selcuk Uluagac. A survey on ransomware: Evolution, taxonomy, and defense solutions. *ACM Comput. Surv.*, 54(11s), sep 2022.

[16] S. Lysenko, K. Bobrovnikova, P. T. Popov, V. Kharchenko, and D. Medzatyi. Spyware detection technique based on reinforcement learning. *CEUR Workshop Proceedings*, 2623:307–316, June 2020. Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[17] Siva K Balasubramanian and Monica alexandra Hodis. Spyware and adware.

[18] Battula Trivikrama Rao. A detailed survey on malware and vulnerability scanners.

[19] Sungkwan Kim, Junyoung Park, Kyungroul Lee, Ilsun You, and Kangbin Yim. A brief survey on rootkit techniques in malicious codes. *J. Internet Serv. Inf. Secur.*, 2(3/4):134–147, 2012.

[20] Surbhi Gupta, Abhishek Singhal, and Akanksha Kapoor. A literature survey on social engineering attacks: Phishing attack. In *2016 international conference on computing, communication and automation (ICCCA)*, pages 537–540. IEEE, 2016.

[21] Francois Mouton, Mercia M Malan, Louise Leenen, and Hein S Venter. Social engineering attack framework. In *2014 Information Security for South Africa*, pages 1–9. IEEE, 2014.

[22] Amirmohammad Sadeghian, Mazdak Zamani, and Shahidan M Abdullah. A taxonomy of sql injection attacks. In *2013 International Conference on Informatics and Creative Multimedia*, pages 269–273. IEEE, 2013.

[23] Zainab S Alwan and Manal F Younis. Detection and prevention of sql injection attack: a survey. *International Journal of Computer Science and Mobile Computing*, 6(8):5–17, 2017.

[24] Huseyin Ahmetoglu and Resul Das. A comprehensive review on detection of cyber-attacks: Data sets, methods, challenges, and future research directions. *Internet of Things*, page 100615, 2022.

[25] Mauro Conti, Nicola Dragoni, and Viktor Lesyk. A survey of man in the middle attacks. *IEEE Communications Surveys Tutorials*, 18(3):2027–2051, 2016.

[26] Jangirala Srinivas, Ashok Kumar Das, and Neeraj Kumar. Government regulations in cyber security: Framework, standards and recommendations. *Future Generation Computer Systems*, 92:178–188, 2019.

[27] VVRPV Jyothsna, Rama Prasad, and K Munivara Prasad. A review of anomaly based intrusion detection systems. *International Journal of Computer Applications*, 28(7):26–35, 2011.

[28] Jing Li, Qinyuan Li, Sheng Zhou, Ying Yao, and Jing Ou. A review on signature-based detection for network threats. In *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, pages 1117–1121. IEEE, 2017.

[29] Rasheed Ahmad, Izzat Alsmadi, Wasim Alhamdani, and Lo'ai Tawalbeh. Zero-day attack detection: a systematic literature review. *Artificial Intelligence Review*, pages 1–79, 2023.

[30] Manasi Gyanchandani, JL Rana, and RN Yadav. Taxonomy of anomaly based intrusion detection system: a review. *International Journal of Scientific and Research Publications*, 2(12):1–13, 2012.

[31] Md Amran Siddiqui, Jack W. Stokes, Christian Seifert, Evan Argyle, Robert McCann, Joshua Neil, and Justin Carroll. Detecting cyber attacks using anomaly detection with explanations and expert feedback. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2872–2876, 2019.

[32] Weiyu Zhang, Qingbo Yang, and Yushui Geng. A survey of anomaly detection methods in networks. In *2009 International Symposium on Computer Network and Multimedia Technology*, pages 1–3, 2009.

[33] Huaglory Tianfield. Data mining based cyber-attack detection. *System simulation technology*, 13(2), 2017.

[34] Antoine Delplace, Sheryl Hermoso, and Kristofer Anandita. Cyber attack detection thanks to machine learning algorithms, 2020.

[35] Emilie Bout, Valeria Loscri, and Antoine Gallais. How machine learning changes the nature of cyberattacks on iot networks: A survey. *IEEE Communications Surveys Tutorials*, 24(1):248–279, 2022.

Apêndice A

Configuração do *Router* 1 implementado na Rede 1

```
! Last configuration change at 22:53:21 UTC Mon Sep 4 2023
upgrade fpd auto
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router1
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
no ip icmp rate-limit unreachable
!
!
!
!
!
```

```
!
ip name-server 193.136.205.106
ip name-server 193.136.208.110
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
!
!
redundancy
!
!
ip tcp synwait-time 5
!
!
!
!
!
!
!
!
!
!
!
interface FastEthernet0/0
 ip address dhcp
```

```
 ip nat outside
 ip virtual-reassembly in
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 193.136.215.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
 standby 1 ip 193.136.215.254
 standby 1 priority 120
 standby 1 preempt
 duplex full
 speed auto
!
interface Serial1/0
 no ip address
 ip nat inside
 ip virtual-reassembly in
 shutdown
 serial restart-delay 0
!
interface Serial1/1
 ip address 30.10.10.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
```

```
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
router ospf 100
 network 30.10.10.0 0.0.0.255 area 1
 network 193.136.208.0 0.0.0.255 area 1
 network 193.136.215.0 0.0.0.255 area 1
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
ip nat inside source list 1 interface FastEthernet0/0 overload
ip route 192.168.10.0 255.255.255.0 192.168.40.2
ip route 192.168.20.0 255.255.255.0 192.168.40.2
ip route 192.168.40.0 255.255.255.0 20.10.10.2
ip route 193.136.205.0 255.255.255.0 193.136.215.2
ip route 193.136.207.0 255.255.255.0 193.136.215.2
ip route 193.136.208.0 255.255.255.0 193.136.215.2
!
access-list 1 permit any
no cdp log mismatch duplex
!
!
!
control-plane
!
!
!
mgcp profile default
```

```
!
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
 transport input all
!
!
end
```