Politécnico
de Viseu

Escola Superior
de Tecnologia
e Gestão de Viseu

HomeSec: Privacidade e Segurança para Smart Homes

Filipe Manuel Sequeira Oliveira

Trabalho de Projeto
Mestrado em Sistemas e Tecnologias de Informação para as Organizações

Trabalho efetuado sob a orientação de
Professor Doutor Filipe Manuel Simões Caldeira

Junho de 2023

Politécnico
de Viseu

Escola Superior
de Tecnologia
e Gestão de Viseu

HomeSec: Privacidade e Segurança para Smart Homes

Filipe Manuel Sequeira Oliveira

Trabalho de Projeto

Mestrado em Sistemas e Tecnologias de Inf. para as Organizações

Trabalho efetuado sob a orientação de

Professor Doutor Filipe Manuel Simões Caldeira

Junho de 2023

DECLARAÇÃO

Eu, Filipe Oliveira, do Departamento de Informática, Escola de Tecnologia e Gestão - Instituto Politécnico de Viseu, confirmo que este é um trabalho original da minha autoria e que figuras, tabelas, equações, trechos de código, obras de arte e ilustrações neste relatório são originais e não foram extraídas do trabalho de qualquer outra pessoa, exceto onde as obras de terceiros foram explicitamente reconhecidas, citadas e referenciadas. Compreendo que, caso isso não seja respeitado, será considerado um caso de plágio. O plágio é uma forma de má conduta académica e será penalizado de acordo.

Filipe Manuel Sequeira Oliveira

DEDICATÓRIA

Dedico este trabalho à minha esposa, Beatriz, cujo apoio constante se revelou um pilar fundamental nesta jornada. A sua paciência e compreensão face à minha frequente ausência para a conclusão deste trabalho foram cruciais para a sua realização.

Este trabalho, que se realizou ao mesmo tempo que experienciamos as alegrias, os medos e os desafios de sermos novos pais, misturou-se de maneira intensa com este projeto, tornando este período um marco inesquecível nas nossas vidas, marcado por momentos de tensão e lágrimas na tentativa de equilibrar as responsabilidades familiares, profissionais e académicas. A minha gratidão será sempre insuficiente para expressar tudo o que me ajudou a alcançar.

Agradeço também ao meu orientador, cuja orientação incansável e compreensão face aos inevitáveis contratempos foram essenciais para a realização deste trabalho. A sua disponibilidade constante e os conselhos precisos foram imprescindíveis para a entrega deste projeto.

Este trabalho é um resultado de um esforço de várias pessoas. A cada uma delas, a minha mais profunda e sincera gratidão. Sem o seu apoio, este projeto não teria sido possível.

RESUMO

Nos últimos anos, tem-se observado um aumento exponencial de dispositivos *Internet of Things* (IoT). Como resultado, têm sido desenvolvidas inúmeras soluções para diversas áreas, nomeadamente, na indústria, saúde, agricultura, *smart cities* e *smart homes*, tornando este tipo de soluções praticamente ubíquas e permitindo estabelecer uma conexão entre os equipamentos e os utilizadores.

As *smart homes* são um dos exemplos onde este tipo de soluções IoT são aplicadas através da conexão de múltiplos dispositivos, embora isso levante preocupações consideráveis de segurança e privacidade. A complexidade e a diversidade inerente a estes dispositivos, incluindo múltiplos protocolos de comunicação e transmissão de dados, aliadas a restrições no poder de processamento, armazenamento e autonomia, representam um desafio significativo na garantia da segurança dos mesmos.

Um dispositivo IoT inseguro pode apresentar várias vulnerabilidades, expondo os utilizadores a vários tipos de falhas de segurança, podendo comprometer a confidencialidade, privacidade e integridade da informação, até à disponibilidade dos dispositivos. Um dos motivos para a proliferação de ameaças, como os *botnets,* é a utilização de dispositivos IoT, onde a aplicação de políticas de cibersegurança foi negligenciada.

Vários dispositivos são constantemente vítimas de ataques, sendo utilizados para, por exemplo, o lançamento de ataques *replay*, *man-in-the-middle, zero-day, spoofing* ou *Denial of Service* (DoS). É, portanto, pertinente desenvolver soluções que ajudem a monitorizar, prevenir e solucionar possíveis ataques, avaliando as vulnerabilidades desses dispositivos e preservando a privacidade dos utilizadores.

Neste contexto, o trabalho proposto centra-se na criação uma solução acessível, *low-cost* e *open source.* Esta solução pretende oferecer aos utilizadores a capacidade de monitorizar a informação transmitida, avaliar possíveis vulnerabilidades dos dispositivos em uso e detetar ataques em tempo real, garantindo assim uma *smart home* segura e protegida.

PALAVRAS-CHAVE

Internet das Coisas (IoT);
Smart Homes;
Cibersegurança;

ABSTRACT

In recent years, an exponential increase in Internet of Things (IoT) devices has been observed. As a result, numerous solutions have been developed for various areas such as industry, health, agriculture, smart cities, and smart homes, making these types of solutions practically ubiquitous and enabling a connection between equipment and users.

Smart homes are one of the examples where this type of IoT solution is applied through the connection of multiple devices, although this raises considerable concerns about security and privacy. The complexity and inherent diversity of these devices, including multiple communication and data transmission, combined with restrictions in processing power, storage and autonomy, represent a significant challenge in ensuring their security.

An insecure IoT device can present several vulnerabilities, exposing users to several types of security breaches, which could compromise the confidentiality, privacy, and integrity of the information, even the availability of the devices. One of the reasons for the proliferation of threats such as botnets is the use of IoT devices, where the application of cybersecurity policies has been neglected.

Various devices are constantly victims of attacks, being used for, for example, the launching of replay attacks, man-in-the-middle, zero-day, spoofing, or DoS. It is therefore pertinent to develop solutions that help to monitor, prevent and solve possible attacks, evaluating the vulnerabilities of these devices and preserving the privacy of users.

In this context, the proposed work focuses on creating an accessible, low-cost, and open-source solution. This solution aims to offer users the ability to monitor transmitted information, evaluate potential vulnerabilities of the devices in use and detect attacks in real time, thus ensuring a secure and protected smart home.

KEYWORDS

Internet of Things (IoT);
Smart Homes;
Cybersecurity;

ÍNDICE GERAL

ÍNDICE DE TABELAS

ÍNDICE DE FIGURAS

ÍNDICE DE GRÁFICOS

LISTA DE SIGLAS / ABREVIATURAS

API - Application Programming Interface

CoAP - Constrained Application Protocol

DDoS - Distributed Denial-of-Service

DNS - Domain Name System

DPI – Deep Packet Inspection

ELK - Elasticsearch, Logstash, and Kibana Stack

FATS - Fingerprint And Timing-based Snooping

HIDS – Host Intrusion Detection System

HTTPS - Hypertext Transfer Protocol Secure

ICDS - Intelligent Cyber-Defense System

IDS – Intrusion Detection System

IP – Internet Protocol

IoT – Internet of Things

IPS – Intrusion Prevention System

MITM – Man-in-the-Middle

MQTT - MQ Telemetry Transport

NIDS – Network Intrusion Detection System

OWASP - Open Web Application Security Project

RNN - Recurrent Neural Network

SDASL - Semi-supervised Domain Adaptation with Subspace Learning

SDN - Software Defined Network

SSH - Secure Shell Protocol

SFTP - Secure File Transfer Protocol

SPAN - Switched Port Analyzer

VPN – Virtual Private Network

WSNs – Wireless sensor network

XSS - Cross-Site Scripting

1. INTRODUÇÃO

A Internet das Coisas (IoT) é uma das tecnologias que apresenta maior crescimento nos últimos anos. De acordo com a pesquisa conduzida pela Gartner (Gartner Research, 2021), estima-se que o número de dispositivos IoT ligados em 2021 foi de 10 mil milhões, com um crescimento esperado até 27 mil milhões para 2025. A proliferação dos dispositivos IoT apresenta inúmeros desafios, entre os quais se destacam a segurança dos dados, a privacidade do utilizador, a integração de sistemas e a escalabilidade. Estes desafios são transversais a vários domínios da tecnologia. Para lidar eficazmente com os desafios criados, torna-se imperativo desenvolver estratégias para administrar a informação que está presente no nosso quotidiano de uma forma cada vez mais ubíqua, adaptando-as à heterogeneidade típica dos dispositivos IoT, garantindo assim, soluções escaláveis, seguras e confiáveis.

A crescente adoção de soluções IoT é transversal a vários setores (Javed et al., 2018), desde a indústria e saúde até às nossas casas. Isto ocorre, em parte, devido à adaptabilidade e diversidade dos dispositivos IoT, que possibilitam a criação de um vasto conjunto de soluções para diferentes tipos de problemas. A investigação em IoT abrange um amplo conjunto de áreas tecnológicas, incluindo comunicações, *machine learning*, *edge computing*, *hardware e* segurança, de forma a dar resposta aos desafios que surgem diariamente. O domínio de investigação é tão abrangente que serve como impulsionador de várias tecnologias, como as redes 5G (Chettri & Bera, 2020), as *wireless sensor networks(*WSNs) (Butun et al., 2020) ou os *digital twins* (Fuller et al., 2020).

As tecnologias IoT revolucionaram a forma como interagimos com o mundo que nos rodeia, interligando vários objetos do nosso quotidiano e dotando-os de novas capacidades, como conectividade, recolha de dados e atuação. Estas características permitem aos dispositivos IoT comunicar e partilhar informações entre si, melhorando a eficiência e a comodidade das nossas vidas. Por exemplo, num contexto de uma

smart home, um dispositivo IoT pode recolher informação sobre a temperatura ou a luminosidade e partilhar essa informação com outros dispositivos, como um sistema de climatização ou de iluminação, que atuam de acordo com essa informação, otimizando o consumo energético e o conforto dos habitantes.

Contudo, apesar do seu tremendo potencial, estes dispositivos têm sido gravemente afetados por vulnerabilidades de segurança. Um exemplo ilustrativo disso é o ataque *Distributed Denial of Service (*DDoS*)* em larga escala, perpetrado pelo *botnet Mirai* (Kolias et al., 2017)*,* que provocou a indisponibilidade de vários *websites* a nível mundial. Estes tipos de ataques vêm alertar para a facilidade com que dispositivos IoT podem ser infetados, evidenciando a sua fragilidade perante ciberataques, que resulta da negligência na implementação de boas práticas de segurança durante os últimos anos.

Websites como o *shodan.io,* possibilitam a identificação de um número significativo de dispositivos IoT vulneráveis e facilmente acessíveis na Internet. Estas falhas de segurança apresentam vários aspetos em comum, entre os quais a pressão constante para que os fabricantes lancem novos produtos rapidamente. Esta pressão muitas vezes leva a uma priorização da usabilidade em detrimento da segurança (Schiller et al., 2022). Adicionalmente, as características intrínsecas destes dispositivos, tais como recursos limitados de processamento, armazenamento e baixo consumo energético, tornam a proteção contra ciberataques um desafio significativo (Conti et al., 2018). Estes constrangimentos, somados à diversidade de protocolos e plataformas existentes no universo IoT, complicam ainda mais a implementação de estratégias de segurança robustas e eficazes.

As *smart homes,* em particular, representam um dos setores com maior adoção de tecnologias IoT. As soluções propostas abrangem desde automação até segurança residencial, proporcionando a simplificação de tarefas e maior eficiência na utilização de recursos. No entanto, o desconhecimento dos utilizadores acerca dos riscos e vulnerabilidades associados à segurança e privacidade destes dispositivos torna as *smart homes* particularmente suscetíveis a ataques. (Zeng et al., 2017) demonstram que muitos utilizadores priorizam fatores como usabilidade e baixo custo dos dispositivos, em detrimento da segurança, comprometendo os benefícios oferecidos por essas soluções.

Perante estes desafios, o estudo apresentado tem como objetivo examinar as vulnerabilidades de segurança dos dispositivos IoT e o potencial impacto na segurança e privacidade em *smart homes*. Adicionalmente, busca-se propor uma solução

baseada na combinação de diferentes recursos, com o intuito de criar um sistema *open-source* que possa ser utilizado de forma autónoma e intuitiva pelos utilizadores, fornecendo uma camada adicional de segurança ao ambiente existente. A eficácia da solução proposta será avaliada e discutida neste trabalho.

1.1. Enquadramento

A crescente integração de soluções IoT em diversos setores, tem transformado significativamente a forma como vivemos e interagimos com o ambiente que nos rodeia. As *smart homes*, em particular, têm-se destacado como uma área de rápido desenvolvimento, com inúmeros dispositivos IoT a serem implementados, para aumentar a automação, a eficiência de recursos e a segurança residencial. Estas casas inteligentes, integram uma ampla gama de dispositivos conectados à internet, como, sensores, fechaduras inteligentes, sistemas de iluminação e outros aparelhos, que podem ser facilmente controlados por comandos de voz ou automação, através de *gateways* inteligentes. Contudo, a crescente prevalência de dispositivos IoT nas nossas vidas, também apresenta desafios significativos em termos de segurança e privacidade dos utilizadores.

A aplicação dos princípios fundamentais de confidencialidade, integridade e disponibilidade, que são cruciais nas soluções de tecnologias de informação tradicionais, é igualmente vital nas implementações de soluções IoT. Apesar das inúmeras potencialidades e vantagens oferecidas por essas tecnologias (El-Azab, 2021; Sovacool & Furszyfer Del Rio, 2020; Sowah et al., 2018), elas também carregam riscos de segurança associados à sua implementação e adoção. As vulnerabilidades presentes nestes sistemas são variadas, abrangendo desde aplicações inseguras (Fernandes et al., 2016), protocolos de comunicação vulneráveis, falta de atualizações de segurança, uso de credenciais padrão ou fracas e armazenamento inseguro de dados. A exploração destas vulnerabilidades pode conduzir a ataques em larga escala a dispositivos inteligentes, com consequências potencialmente devastadoras para a segurança e privacidade dos utilizadores (Ronen et al., 2017).

As *smart homes* representam um caso específico de preocupação, dado que estão intrinsecamente associadas à recolha e transmissão de informações pessoais sensíveis, como fotografias, vídeos e outros dados confidenciais. Esta característica torna-as alvos preferenciais para ciberataques, pois a violação dessas informações pode acarretar graves consequências para os utilizadores e as suas famílias. Adicionalmente, é demonstrado em (N. Zhang et al., 2019) que dispositivos como,

câmaras IP e microfones, podem ser acedidos remotamente por meio de vulnerabilidades de segurança nos *gateways*, representando uma ameaça significativa à segurança das residências e dos seus habitantes.

A implementação de soluções IoT é frequentemente condicionada por fatores como baixo poder computacional, memória limitada e recursos energéticos escassos. Esta realidade intensifica a complexidade de garantir a segurança e privacidade dos utilizadores. Acrescenta-se a esta problemática o facto de muitos dos utilizadores finais das soluções IoT não possuírem um conhecimento aprofundado acerca das práticas de cibersegurança, aumentando assim o risco de exposição das *smart homes.*

Neste sentido, torna-se imperativo que os fabricantes e programadores assumam a responsabilidade de garantir a segurança das soluções IoT desde a fase inicial do processo de desenvolvimento até à sua implementação. Isto envolve também a partilha de informações acerca de vulnerabilidades e ameaças, garantindo uma abordagem unificada e eficiente na promoção da segurança no contexto das *smart homes.* Esta colaboração poderá incluir a criação de padrões e protocolos de segurança comuns, bem como a realização de auditorias e testes de penetração regulares, para identificar e corrigir possíveis vulnerabilidades.

Por outro lado, os utilizadores finais das soluções IoT nas *smart homes* também têm um papel fundamental na garantia da segurança dos seus dispositivos e redes (Duezguen et al., 2021). Neste contexto, é fundamental que sejam promovidas ações de sensibilização e formação sobre cibersegurança, permitindo que os utilizadores compreendam os riscos associados à utilização destes dispositivos e que possam aplicar as melhores práticas de segurança, tais como a atualização regular do *firmware*, a utilização de palavras-passe fortes e a monitorização constante das atividades dos dispositivos na rede.

Em suma, perante os desafios acima mencionados, este trabalho visa contribuir para a compreensão dos perigos e ameaças associados à segurança nas *smart homes,* promovendo a adoção de práticas e estratégias eficazes na mitigação de riscos e na salvaguarda da privacidade e integridade dos dados dos utilizadores.

1.2. Objetivos

A intensificação da adoção das *smart homes* exige a implementação de soluções eficazes de segurança, de forma a garantir a privacidade e a proteção dos utilizadores. Tendo em conta os desafios mencionados na secção anterior, o presente

projeto tem como objetivo principal o desenvolvimento e a implementação de uma solução que contribua para aumentar o nível de segurança e privacidade nas *smart homes*, introduzindo uma camada adicional de proteção aos sistemas já existentes nas residências. Além disso, a solução proposta deve ser de fácil instalação (*plug&play*) e acessível ao utilizador comum, contribuindo para a sensibilização acerca dos riscos e vulnerabilidades inerentes às soluções IoT.

Para que os objetivos desta solução sejam alcançados, propõe-se o desenvolvimento de um sistema de monitorização passiva de uma rede dedicada aos dispositivos IoT, isolada dos restantes dispositivos mais sensíveis presentes numa *smart home*. A implementação baseia-se em várias ferramentas de código aberto (*open-source*), que não acarretarão custos adicionais para o utilizador final, nem causarão perturbações no funcionamento dos dispositivos já instalados.

A solução consiste em dois componentes principais: um dedicado à monitorização dos dispositivos e informações de uma *smart home*, e outro à gestão dos dispositivos IoT, atuando como um *gateway*, através do qual serão geridas as comunicações entre os vários dispositivos. No que concerne ao componente de monitorização de rede, propõe-se um sistema de deteção de intrusões (IDS) baseado na ferramenta *open-source Suricata*, que será integrado num *stack* ELK (*Elasticsearch-Logstash-Kibana*) para visualização de *dashboards* personalizados e alertas de segurança correspondentes. O componente responsável pela rede IoT compreenderá um conjunto de ferramentas, nomeadamente, um sistema de prevenção de intrusões (IPS) *fail2ban*, um DNS *Sinkhole* (*PI-Hole*) e um conjunto de regras de *firewall* (*iptables*), que ajudarão a mitigar eventuais ataques e a isolar os dispositivos IoT dos restantes dispositivos presentes na habitação do utilizador, mediante a criação de uma rede específica.

A fim de assegurar a acessibilidade financeira do projeto, a implementação será realizada em minicomputadores *Raspberry Pi*, cuja capacidade para proporcionar uma camada adicional de segurança num ambiente típico de *smart home* será avaliada ao longo do projeto.

Em resumo, os objetivos do projeto podem ser definidos na seguinte ordem:

- Incrementar o nível de segurança de uma *smart home* com uma solução *plug&play*;
- Monitorizar os dispositivos IoT;
- Isolar os dispositivos IoT dos dispositivos mais sensíveis de uma rede doméstica;

- Realizar testes de penetração da solução proposta;
- Avaliar a capacidade de um *Raspberry Pi* para a implementação de um IDS.

Os objetivos mencionados visam abordar os desafios inerentes à segurança nas *smart homes* e contribuir para a criação de um ambiente mais seguro e protegido para os seus utilizadores.

1.3. Abordagem metodológica

A crescente adoção das *smart homes* tem levantado preocupações sobre a segurança dessas soluções. Para abordar esse problema, realizou-se uma revisão bibliográfica abrangente, focada na investigação atual na área de segurança em *smart homes*, recorrendo às principais publicações científicas. O enfoque da revisão bibliográfica incidiu, sobretudo, em sistemas de deteção de vulnerabilidades em redes de *smart homes* e em abordagens cujo produto final não requer interação por parte dos utilizadores.

A fase de investigação extensiva revelou uma quantidade significativa de estudos e propostas realizados na área das *smart homes*. Após a análise da investigação disponível, elaborou-se o desenho de uma arquitetura típica de uma *smart home*, incluindo diversos dispositivos de diferentes tipologias, como *Smart Speakers*, *Smart TVs*, sistemas de videovigilância, tomadas e lâmpadas inteligentes. Adicionalmente, foram considerados protocolos de comunicação como *Zigbee*, *Bluetooth* e *Wi-Fi*, a fim de garantir que a fase de implementação da solução não se concentrasse nas características de um único dispositivo ou protocolo específico, procurando ser o mais abrangente possível dentro do domínio proposto.

Para avaliar a eficácia da solução proposta, realizaram-se vários testes de penetração, cujo principal objetivo residia em avaliar a capacidade da solução em resistir e reagir a possíveis violações de segurança, bem como identificar potenciais vulnerabilidades. Todas estas etapas são descritas e detalhadas sucintamente nas seções subsequentes deste trabalho.

A abordagem metodológica adotada neste projeto baseia-se, portanto, na revisão bibliográfica, na elaboração de uma arquitetura típica de uma *smart home* e na avaliação da solução proposta por meio de testes de penetração.

1.4. Estrutura do documento

O presente documento estrutura-se em sete capítulos. No primeiro capítulo, é apresentada uma introdução ao trabalho proposto, delineando o enquadramento do mesmo. Estabelecem-se os objetivos e descreve-se a metodologia aplicada. No segundo capítulo, procede-se a uma revisão bibliográfica, contemplando um levantamento dos artigos científicos mais recentes que abordam os temas relacionados com este trabalho.

No terceiro capítulo, dedicado à arquitetura da solução, apresenta-se o desenho da proposta e discute-se a forma como os vários componentes se interligam. O quarto capítulo, centrado nas ferramentas e tecnologias, detalha as soluções tecnológicas adotadas, como o *Raspberry Pi, o Suricata*, o *iptables* e o *stack* ELK, e os seus respetivos papeis na criação de uma infraestrutura de rede segura para dispositivos IoT.

O capítulo cinco foca-se nos passos dados durante a execução e implementação deste projeto, desde a instalação dos módulos "*hotspot*" e *"monitor"*, à criação de regras específicas para o sistema. No capítulo seguinte, são expostas as observações resultantes dos testes de avaliação da solução proposta, incluindo a deteção e mitigação de vulnerabilidades, bem como a necessidade de melhoramentos contínuos face a novas ameaças.

Por fim, no último capítulo, realiza-se uma reflexão sobre o trabalho desenvolvido, culminando nas conclusões retiradas deste estudo. Destaca-se a importância de soluções de segurança adaptáveis e resilientes para proteger ambientes de *smart homes*, considerando a constante evolução do mundo tecnológico, e propõem-se direções para futuros trabalhos neste campo.

2. IoT e Segurança em *Smart Homes*

2.1. IoT

A Internet das Coisas (IoT) é um conceito abrangente que engloba várias definições e perspetivas, conforme discutido por diversos autores (Alansari et al., 2018; Guth et al., 2018). Introduzido por Kevin Ashton em 1999 (Ashton, 1999), o IoT, foi descrito como um ecossistema de dispositivos e serviços interligados, capazes de se comunicar e partilhar dados em diversos contextos de aplicação. Embora não exista uma definição padrão para o IoT, (Sorri et al., 2022) identificam várias características principais que se lhe associam, nomeadamente:

- **Interação**: Capacidade de interoperabilidade e troca de informações entre dispositivos;
- **Dispositivos**: Sensores inteligentes, microcomputadores e sistemas embebidos;
- **Serviços**: Aplicações e soluções que aprimoram processos e agregam valor ao ecossistema;
- **Dados**: Dados em bruto adquiridos pelos sensores, o que irá adicionar valor aos utilizadores;
- **Tecnologias e Informação**: Processamento dos dados e protocolos de comunicação;
- **Utilizadores**: Quem vai utilizar os serviços disponibilizados;
- **Ubiquidade**: Sistema transversal a tudo o que nos envolve em tempo-real;
- **Singularidade**: Cada dispositivo deve ser passível de identificação única, facilitando a gestão e rastreabilidade.

O IoT representa um enorme avanço na forma como os dispositivos e sistemas interagem entre si, proporcionando inúmeras oportunidades para melhorar a vida das

pessoas e otimizar processos em diversos sectores. No entanto, a crescente prevalência e complexidade desses dispositivos acarretam desafios significativos, especialmente no que diz respeito à segurança, aspeto que tem recebido considerável atenção na literatura recente (Mohamad Noor & Hassan, 2019).

A segurança é uma questão crítica no contexto da IoT (O. Garcia-Morchon et al., 2019) devido ao enorme volume de dados sensíveis que são gerados, processados e transmitidos por esses dispositivos. As vulnerabilidades e falhas de segurança podem resultar em graves consequências, incluindo violação de privacidade, perda de dados confidenciais e até mesmo riscos à integridade física dos utilizadores e dos sistemas. Além disso, uma violação de segurança num dispositivo IoT pode ter impactos negativos significativos na reputação e credibilidade do fabricante.

Desta forma, é crucial abordar a segurança na IoT de forma abrangente e eficaz, para que se possa tirar partido das oportunidades oferecidas por este ecossistema em constante evolução, sem comprometer a privacidade e a segurança dos utilizadores e das informações envolvidas.

2.2. Segurança em IoT

Para abordar as vulnerabilidades de segurança em IoT, é necessário adotar uma abordagem multifacetada que considere diversas estratégias em vários níveis, desde o *hardware* dos dispositivos até à camada aplicacional das soluções. Segundo (Ahanger & Aljumah, 2019), várias medidas de segurança podem ser implementadas para mitigar os riscos de segurança e garantir a proteção das soluções.

No nível de *hardware*, a utilização de encriptação, a implementação de *designs fail-secure* e a adoção de mecanismos de controlo de acesso são exemplos de estratégias que podem ser aplicados para proteger os dispositivos e os seus dados. No nível de comunicação, a implementação de *firewalls*, v*irtual private networks* (VPNs) e sistemas de deteção e prevenção de intrusões (IDS/IPS) pode melhorar significativamente a segurança das redes IoT.

No nível aplicacional é fundamental que os programadores de *software* sigam as melhores práticas de desenvolvimento seguro, incluindo a realização de testes de segurança e a implementação de metodologias *zero-trust* nas suas soluções. Além disso, a adoção de diretrizes específicas para prevenir ataques, como *SQL injection* e *cross-site scripting* (XSS), é crucial para garantir a segurança das aplicações.

Conforme a análise de (Ahanger & Aljumah, 2019), os riscos de segurança predominantes no IoT podem ser categorizados em confidencialidade, integridade, disponibilidade, autenticação e não-repúdio. A confidencialidade visa garantir a privacidade das informações dos utilizadores, tornando-as inacessíveis a entidades não autorizadas. A integridade procura proteger a precisão e a consistência dos dados, prevenindo modificações não autorizadas. A disponibilidade assegura que os serviços e dados estejam sempre acessíveis aos utilizadores autorizados quando necessário. A autenticação valida a identidade dos utilizadores e dispositivos para prevenir acessos não autorizados, enquanto o não-repúdio garante que as partes envolvidas numa transação não possam negar a sua participação.

Segundo (Schiller et al., 2022), garantir a segurança no IoT, contrariamente às soluções tradicionais, apresenta desafios intensificados por características singulares destes sistemas. A heterogeneidade dos dispositivos em comunicação complica a adoção de protocolos padrão, criando múltiplos pontos de ataque. A priorização da usabilidade em prejuízo da segurança, os recursos limitados, o curto período até a entrada no mercado (*time-to-market*) e a alta disponibilidade dos dispositivos são outros fatores que acentuam as vulnerabilidades de segurança. Estas questões estão diretamente associadas às falhas de segurança mais comuns e aos principais objetivos de segurança no IoT, como ilustrado na Figura 1.

Figura 1 - Principais objetivos na segurança IoT

Em adição às categorias de riscos de segurança apresentados, é importante considerar também as vulnerabilidades específicas associadas às soluções de IoT. De acordo com um relatório da *Open Worldwide Application Security Project* (OWASP, 2019), várias vulnerabilidades críticas afetam dispositivos IoT, na Tabela 1, são apresentadas as principais vulnerabilidades identificadas pela OWASP.

Tabela 1 - Principais vulnerabilidades das soluções IoT (OWASP, 2019)

Vulnerabilidade	Descrição	Mitigação
Passwords fracas	▪ *Passwords* fracas são de fácil previsão, podendo ficar facilmente expostas publicamente; ▪ Credenciais imutáveis ou *backdoors* no *firmware* dos sistemas, permitem o acesso aos mesmos; ▪ *Passwords* fracas ou *hardcoded*, são a forma mais fácil de comprometer um dispositivo, permitindo, lançar ataques em larga escala, como por exemplo, o *botnet Mirai.*	▪ Cada dispositivo deve ter um conjunto exclusivo de credenciais; ▪ Desabilitar senhas fracas e remover *backoors* criados durante a fase de *debugging*.
Serviços de rede inseguros	▪ Serviços de rede inseguros, expostos externamente, como por exemplo, portas abertas, podem facilmente comprometer a confidencialidade, integridade e autenticidade dos dados, bem como, permitir o acesso remoto não autorizado, possibilitando ataques, como por exemplo, *Man-in-the-Middle* (MITM).	▪ Usar protocolos seguros como HTTPS*,* SFTP *e* SSH; ▪ Desabilitar portas que não são essenciais; ▪ Manter os dispositivos numa rede separada; ▪ Instalação de atualizações regulares.
Interfaces inseguras	▪ Interfaces inseguras como *Web* APIs*, cloud* ou aplicações móveis, podem comprometer os dispositivos ou os componentes associados; ▪ Problemas como falta de autenticação, autorização ou de encriptação, são bastante comuns; ▪ Os requisitos de segurança devem ser estendidos para além do dispositivo até todos os componentes integrantes da solução.	▪ Implementar o princípio do privilégio mínimo; ▪ Autenticação forte dos *endpoints* IoT.
Falta de mecanismos de atualização seguros	▪ Incapacidade de atualizar um dispositivo com segurança; ▪ A falta de validação do *firmware* e da fonte no dispositivo, ou o envio de um *firmware* a partir de uma comunicação insegura.	▪ Implementar atualizações assinadas digitalmente; ▪ Implementar mecanismos *anti-rollback*; ▪ Proteger e verificar os acessos às atualizações.

Uso de componentes inseguros ou desatualizados	▪ Componentes que possam comprometer o dispositivo, como por exemplo, configurações inseguras ou o uso de aplicações de terceiros.	▪ Evitar e substituir tecnologias obsoletas; ▪ Substituir componentes ultrapassados; ▪ Garantir a fiabilidade dos componentes de hardware e software terceiros.
Proteção da privacidade insuficiente	▪ Dados confidenciais guardados num dispositivo que possam estar comprometidos, ou que possam ser usados de forma insegura.	▪ Limitar o armazenamento de dados pessoais em dispositivos; ▪ Aplicar uma política de proteção de dados; ▪ Preparar um plano de resposta a incidentes e ataques futuros.
Transferência e armazenamento de dados inseguros	▪ Falta de encriptação ou controlo no acesso a informação confidencial durante o armazenamento, processamento ou transmissão de dados.	▪ Garantir encriptação a todos os níveis; ▪ Utilizar protocolos seguros (HTTPS, SSH e SFTP).
Incapacidade de administração dos dispositivos	▪ Falta de suporte para dispositivos em ambientes produtivos; ▪ Gestão de dispositivos (*asset management*); ▪ Gestão de atualizações; ▪ Desativação segura e monitorização do sistema.	▪ Garantir a desativação segura; ▪ Colocar *endpoints* em quarentena; ▪ Aplicar *blacklists*; ▪ Integrar os dispositivos numa solução que permita fazer a gestão de ativos, *bug tracking*, e gestão de *patches*.
Configurações padrão inseguras	▪ Os dispositivos são entregues com configurações inseguras ou são incapazes de restringir possíveis alterações efetuadas pelos utilizadores.	▪ Usar apenas configurações padrão seguras; ▪ Permitir aos utilizadores alterar as passwords padrão.
Segurança física inadequada	▪ Falta de medidas de proteção física, permitindo que, atacantes tenham acesso físico ao dispositivo, de forma a obterem informação confidencial.	▪ Entender como um utilizador final pode modificar o dispositivo; ▪ Antecipar proactivamente os danos que qualquer utilizador possa causar ao dispositivo.

Para desenvolver soluções de segurança eficazes e compreender de forma abrangente a complexidade dos ataques a que um sistema IoT pode estar sujeito, torna-se necessário examinar a sua estrutura e os principais níveis que a constituem: aquisição, transporte, processamento e aplicação. Cada camada ou nível tem funções distintas dentro do sistema, e portanto, podem estar associados a diferentes tipos de ataques de segurança. A camada de aquisição lida com a recolha de dados dos dispositivos físicos, a camada de transporte é responsável pela transmissão desses dados, a camada de processamento realiza a computação dos dados, e a camada de aplicação lida com a interação do sistema com os utilizadores ou outras aplicações

A Figura 2 apresenta um esquema adaptado (Hassija et al., 2019), no qual se identificam os ataques principais associados a cada nível de uma arquitetura IoT. Enquanto alguns estudos focam-se em problemas específicos de um determinado nível, outros abordam temas que se estendem a vários níveis de uma arquitetura IoT (J. Zhang et al., 2019). É importante ressaltar que os riscos de segurança são comuns a todas os níveis, em (Khan & Salah, 2018), são identificados os principais problemas de segurança presentes nos diversos níveis de uma arquitetura IoT.

Aprofundando a análise no contexto específico das *smart homes*, verifica-se que a segurança na IoT assume uma importância ainda maior. As casas inteligentes integram uma variedade de dispositivos e sistemas interligados que visam melhorar o conforto, a eficiência energética e a segurança dos seus habitantes. Contudo, à medida que os dispositivos IoT proliferam neste ambiente, intensificam-se também os potenciais riscos associados à segurança e privacidade dos utilizadores (Fernandes et al., 2016).

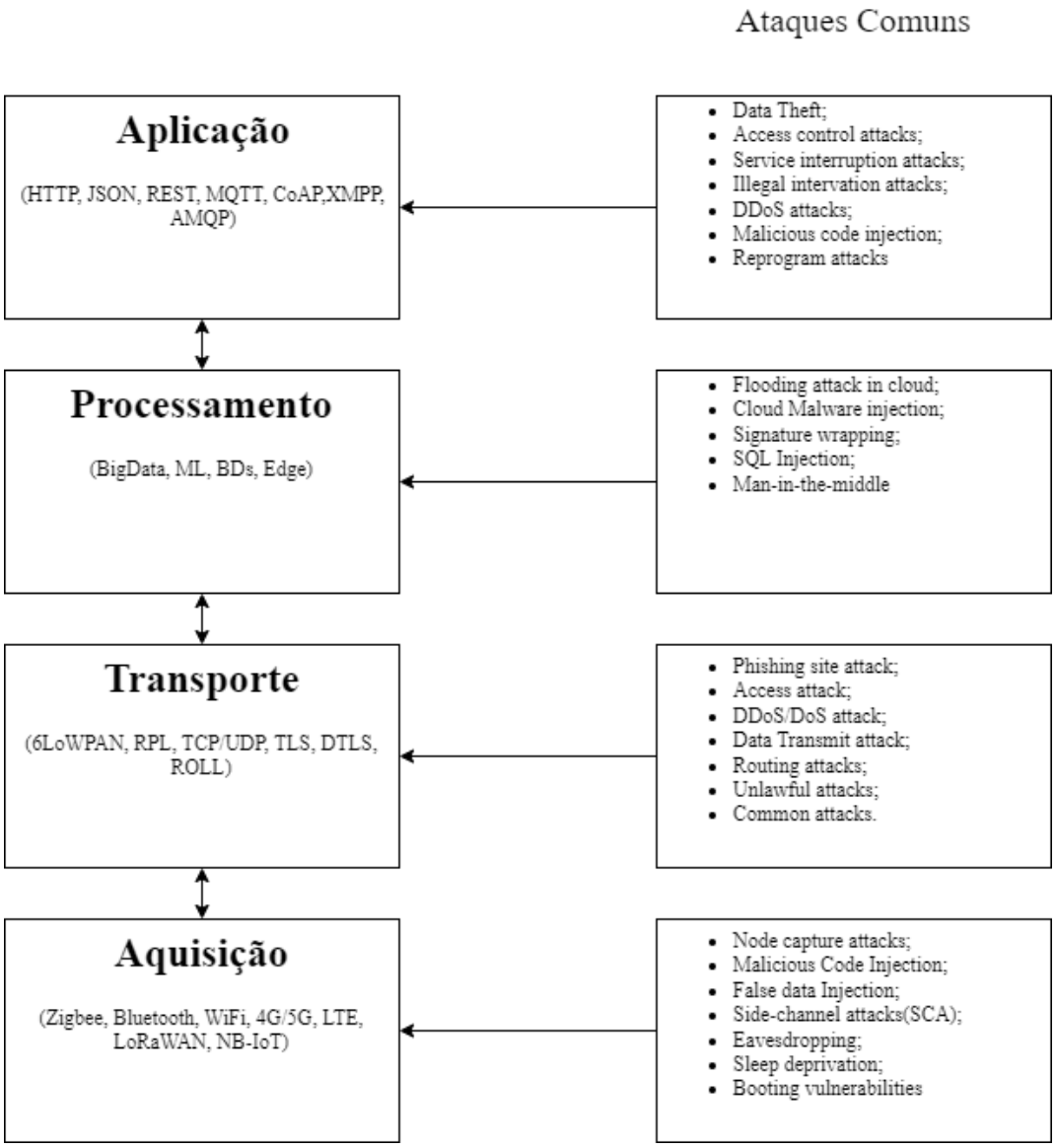


Figura 2 - Camadas de uma solução IoT e as suas principais ameaças (Hassija et al., 2019).

2.3. Segurança em *Smart Homes*

As *smart homes* têm despertado um interesse crescente (D. Kumar et al., 2019), devido à sua capacidade de integrar uma variedade de dispositivos interligados, desde *smart* TVs, termostatos inteligentes, fechaduras inteligentes, até objetos mais simples, como escovas de dentes. Estes dispositivos proporcionam diversos benefícios aos utilizadores, incluindo maior eficiência nas tarefas quotidianas, redução de custos energéticos e aumento da segurança, entre outros (Marikyan et al., 2019; Nižetić et al., 2020).

Contudo, juntamente com os benefícios e potencialidades associadas, surgem também riscos de segurança que podem comprometer a privacidade dos utilizadores, chegando mesmo a colocar em causa a sua integridade física. Um exemplo disso é a possibilidade de um indivíduo mal-intencionado aceder a uma fechadura inteligente e realizar um assalto a uma residência com facilidade. Atualmente, vários estudos têm sido publicados alertando para os riscos de segurança existentes (Ammar et al., 2018; Apthorpe et al., 2017), tornando-se essencial consciencializar os utilizadores acerca dos riscos a que estão expostos.

Uma casa inteligente, ou *smart home*, caracteriza-se pela presença de uma diversidade de dispositivos que interagem entre si através de diferentes protocolos de comunicação, conforme se pode observar na Figura 3. Destes protocolos, os mais populares incluem o *Wi-Fi, Bluetooth, ZigBee* e *Z-Wave*, com os dois últimos sendo especialmente conhecidos por criarem redes de malha (*mesh*) de curto alcance e baixo consumo energético.

Nestes ambientes, é comum a presença de dispositivos que funcionam como portas de acesso ou gateways. Frequentemente designados de "inteligentes", estes dispositivos incluem soluções amplamente reconhecidas como a *Alexa (Amazon)*, *Google Home* ou *Samsung SmartThings*, que desempenham o papel de assistentes pessoais. Estes equipamentos constituem um ponto central, permitindo a integração e o controlo dos vários dispositivos que compõem uma *smart home*.

Os assistentes pessoais oferecem uma interface de utilização notavelmente acessível, tornando possível o controlo dos dispositivos através de comandos de voz simples ou de uma aplicação móvel. Contudo, apesar de toda a comodidade que proporcionam, estes dispositivos processam informação confidencial, originando preocupações legítimas acerca da privacidade e segurança dos utilizadores (Acar et al., 2020). A garantia absoluta de privacidade e segurança dos utilizadores não pode ser assegurada, dada a natureza intrínseca do processamento de dados nestes dispositivos.

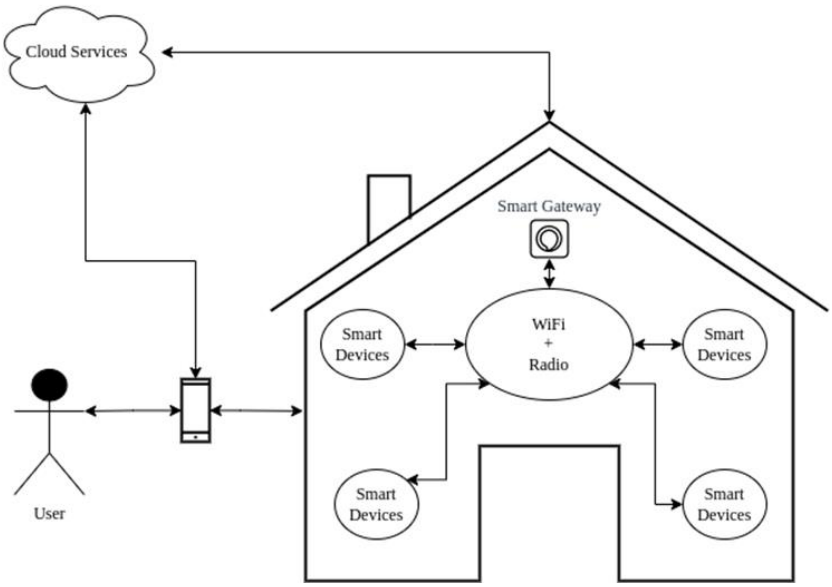


Figura 3 - Arquitetura típica de uma *smart home*

A diversidade de dispositivos numa *smart home* é vasta, abrangendo desde termostatos inteligentes (por exemplo, *Nest*), sistemas de iluminação (como *Philips Hue*), videovigilância, sensores de movimento, luz e humidade, até tomadas inteligentes. Todos estes dispositivos, conectados à Internet através de uma rede *Wi-Fi*, podem potenciar o risco de violações de privacidade e ataques, gerando preocupações significativas no que se refere à segurança (Ali & Awad, 2018).

Como mencionado anteriormente, o forte crescimento na adoção de tecnologias IoT nas nossas habitações levanta diversas questões de segurança (Meneghello et al., 2019). Portanto, torna-se vital identificar as vulnerabilidades destas soluções IoT em ambiente de *smart home*, assegurando a privacidade, integridade e autenticidade dos dados gerados e transmitidos.

Neste contexto, diversos estudos têm sido realizados para enfrentar os vários desafios que as soluções *smart home* apresentam. Por exemplo, (Nassiri Abrishamchi et al., 2022) propõem uma ferramenta baseada em amostras de dados e "*supervised learning*" (SDASL), com o objetivo de ocultar os padrões de utilização de uma rede, numa solução adaptada às *smart homes*, com baixo consumo energético, baixa latência, adaptabilidade e proteção dos seus utilizadores, evitando ataques do tipo FATS (*fingerprint and timing-based snooping*). (Miettinen et al., 2017) desenvolvem uma ferramenta*, IoT Sentinel*, recorrendo a *software defined networks* (SDNs), através da qual é realizada a identificação automática de dispositivos vulneráveis, aplicando modelos de *machine learning* e técnicas de *traffic fingerprinting* para garantir a implementação de medidas de segurança que mitiguem possíveis riscos de

segurança, proporcionando uma solução que proteja a rede dos utilizadores, isolando comunicações para dispositivos considerados inseguros.

No que se refere às comunicações em IoT, diversos estudos têm abordado protocolos característicos de um ambiente IoT, como o MQTT e o CoAP, identificando cenários de aplicação, vantagens e desvantagens de cada um (Hedi et al., 2017; Z. Shelby et al., 2014). (Ling et al., 2017) estudam a utilização destes protocolos de comunicação e analisam as comunicações de um "*smart socket*" (tomada inteligente), recorrendo a ataques *brute force, device scanning* e *spoofing*, verificando que o dispositivo não utiliza protocolos de comunicação e autenticação seguros. (Abu Waraga et al., 2020) propõem uma plataforma *open source* para identificar vulnerabilidades em redes IoT, permitindo a automatização de testes sem intervenção dos utilizadores, gerando relatórios de problemas detetados e monitorizando ativamente atividades maliciosas.

(Seralathan et al., 2018) analisam as ameaças que podem comprometer os dispositivos IoT e apresentam um caso de estudo para uma câmara IP, aplicando uma análise da rede e ataques *man-in-the-middle*, discutindo a importância de aplicar as melhores práticas de segurança a estes dispositivos. Atualmente, a ideia generalizada é que qualquer dispositivo adquirido poderá ter vulnerabilidades de segurança, tornando-se assim essencial identificar os vários desafios associados às especificidades típicas destas soluções IoT. Algumas soluções direcionam-se para tecnologias específicas, como as *wired sensor networks* (WSNs), enquanto outras visam soluções mais heterogéneas, como *testbeds*, que identifiquem vulnerabilidades a diversos níveis, tais como protocolos de comunicação, *firmware* ou integridade dos dados.

(Jacobsson et al., 2016; Nurse et al., 2017) avaliam os riscos associados às *smart homes,* particularmente no que diz respeito à confidencialidade, integridade e disponibilidade, expondo os vários riscos e classificando-os de acordo com a sua criticidade. São ainda mencionadas várias abordagens baseadas em análise de risco, de segurança e de privacidade. (Apthorpe et al., 2017) sugerem estratégias para proteger a privacidade das redes *smart home*, limitando o tipo de dados que possam estar acessíveis externamente.

Em (OConnor et al., 2019), é desenvolvida uma ferramenta denominada *HomeSnitch*. Esta ferramenta recorre a técnicas de *machine learning* para classificar as comunicações de uma *smart home,* baseando-se em padrões de comportamento. Recorre ainda a SDNs (*software defined networks*) para monitorizar as comunicações

entre dispositivos, conseguindo identificar comportamentos desconhecidos mesmo quando as comunicações são efetuadas por canais seguros e com recurso a encriptação, demonstrando assim, os reais perigos de uma *smart home*.

(Ramapatruni et al., 2019), por sua vez, criam um modelo *Hidden Markov* para a deteção de anomalias, com base na rotina diária de um utilizador numa *smart home*. Este modelo tem o potencial de proteger a *smart home* de possíveis ataques, tais como o *botnet Mirai* ou *Reaper*. Numa abordagem semelhante, mas com base no consumo energético dos dispositivos IoT, (Alsabilah & Rawat, 2021) apresentam uma solução para deteção de ataques com base em filtros *Kalman*, visando prever o consumo de energia nas *smart homes*. Recorrendo a testes *Shapiro-Wilk*, é detetado o desvio entre as medidas observadas e estimadas.

(Nguyen et al., 2018) apresentam um sistema de autoaprendizagem para deteção de dispositivos comprometidos numa rede IoT, recorrendo a modelos de redes neuronais recorrentes (RNN). Esta abordagem ambiciona detetar automaticamente um tipo específico de dispositivo, com o mínimo de intervenção humana. O estudo demonstra um elevado grau de eficiência e uma taxa bastante reduzida de falsos alarmes num ambiente heterogéneo, ajudando a ultrapassar algumas das dificuldades inerentes à criação de um modelo eficaz para a deteção de anomalias, devido às características específicas das soluções IoT. A grande variedade de dispositivos e o respetivo volume de dados transmitidos, geralmente baixos, dificultam a criação de modelos baseados em comportamentos, tornando a sua construção mais complexa.

(Huc & Trcek, 2021) desenvolvem um algoritmo de *machine learning* para deteção de anomalias, otimizado para dispositivos IoT, de forma a colmatar as restrições típicas deste tipo de soluções para as quais os modelos de *machine learning* tradicionais não são adequados. Este trabalho evidencia a crescente importância da segurança nas redes IoT e a necessidade de uma deteção de anomalias eficiente.

(Mandalari et al., 2021) apresentam a solução *IoTrimmer*, que permite bloquear de forma eficaz o tráfego não essencial dos dispositivos IoT, limitando, deste modo, a informação exposta a terceiros, sem interferir com o correto funcionamento dos dispositivos. Os autores do estudo salientam que praticamente metade dos dispositivos analisados estabelece contato com destinos não necessários para o desempenho das suas funções principais.

(Doshi et al., 2018) apresentam uma solução que visa a deteção de ataques DDoS em dispositivos IoT, recorrendo a técnicas de *machine learning*. Os autores debruçam-se sobre as características peculiares do tráfego de rede proveniente dos

dispositivos IoT para informar a seleção de características para os seus modelos. Na sua abordagem, simularam uma rede de dispositivos IoT de consumo, conseguindo identificar de forma bem-sucedida o tráfego associado a ataques, com uma precisão superior a 0.999. A pesquisa sugere que os *routers* dos *gateways* domésticos poderiam detetar automaticamente as fontes de ataques DDoS, utilizando algoritmos de *machine learning* de baixo custo e dados de tráfego baseados em fluxo, independentemente do protocolo.

Em (Hafeez et al., 2020), é proposto um sistema, *IoT-Keeper*, para deteção de anomalias e isolamento de dispositivos IoT comprometidos na rede e em tempo real. A execução de um modelo de classificação no *edge gateway*, otimizado para sistemas com recursos limitados, demonstra resultados eficientes na identificação de anomalias. Ao identificar um dispositivo IoT como fonte de atividade maliciosa, o sistema restringe o acesso à rede do dispositivo através de redes *ad-hoc*. A implementação do *IoT-Keeper* revela eficiência na classificação de tráfego em tempo real, com uma precisão elevada, poucos falsos alarmes e baixa latência. O sistema é adaptável, podendo ser aplicado a *gateways* de rede e dispositivos *edge*, protegendo a comunicação local e remota de dispositivos IoT.

A crescente necessidade de soluções seguras e robustas na área de IoT tem motivado diversos estudos (Aneja et al., 2018; Bezawada et al., 2018; Ferman & Ali Tawfeeq, 2021; Kuzniar et al., 2022) que exploram técnicas de identificação automática de dispositivos, inclusive em cenários onde as comunicações são encriptadas. (Marchal et al., 2019) propõem um modelo baseado em algoritmos de *machine learning*, autónomo e sem supervisão humana, com o objetivo de identificar o tipo de dispositivos numa rede IoT e, a partir dessa identificação, desenvolver um sistema de deteção de anomalias. Este modelo utiliza *fingerprints* criados a partir de fluxos periódicos, sendo agnóstico à encriptação do tráfego, dado que não recorre a informações do *payload* dos pacotes transmitidos. De forma semelhante, (Sun et al., 2019) apresentam um método de *fingerprinting* que prescinde de técnicas de *Deep Packet Inspection* (DPI), um método que analisa o conteúdo dos pacotes de dados à medida que atravessam um ponto de controle na rede, possibilitando a classificação de dispositivos em comunicações encriptadas. Os autores coletam dados de telemetria das comunicações dos dispositivos ao nível do *edge* e, com base nessa informação, extraem características dos fluxos gerados pelos dispositivos. Posteriormente, aplicam um modelo de *deep learning* (*Multi-layer Perception*) para classificar os fluxos. Estas abordagens destacam-se pela sua capacidade de identificar dispositivos IoT de forma autónoma e segura, mesmo em contextos de comunicações encriptadas.

A utilização de sistemas de deteção e prevenção de intrusões (IDS/IPS) nas *smart homes* representa um avanço significativo na garantia da segurança das redes domésticas. Estes sistemas monitorizam o tráfego de rede, procurando padrões suspeitos e alertando os utilizadores quando detetam atividades maliciosas e não autorizadas. Ao integrar algoritmos de inteligência artificial, os sistemas IDS/IPS tornam-se progressivamente mais eficientes na deteção de ameaças e na prevenção de intrusões, contribuindo para a proteção dos dispositivos IoT e das informações pessoais dos utilizadores. Em (Chaabouni et al., 2019) é realizada uma pesquisa abrangente sobre os vários tipos de ferramentas IDS existentes, evidenciando-se como este tipo de sistemas em IoT tem incorporado, cada vez mais, algoritmos de inteligência artificial.

Os sistemas de deteção de intrusões (IDS) atuam na identificação de potenciais ameaças, emitindo alertas aos utilizadores. Ao detetar ameaças em tempo real, os sistemas IDS possibilitam a implementação de medidas preventivas e corretivas para proteger a infraestrutura de rede doméstica e os dispositivos IoT conectados. Por outro lado, os sistemas de prevenção de intrusões (IPS) são responsáveis pela inspeção e rejeição de pacotes suspeitos, mitigando ataques em tempo real. No entanto, a implementação de sistemas IDS/IPS em ambientes de *smart homes* enfrenta desafios específicos, tais como a heterogeneidade dos dispositivos e protocolos, a limitação de recursos e a necessidade de escalar as soluções de segurança para lidar com o grande número de dispositivos conectados. Para abordar esses desafios, os pesquisadores têm explorado várias abordagens e técnicas, incluindo redes neuronais e *deep learning* (Jayalaxmi et al., 2022).

Neste contexto, as técnicas de *Deep Packet Inspection (*DPI*)* surgem como um componente importante da maior parte dos sistemas IDS/IPS*.* Ao analisar os pacotes de dados em detalhe, o DPI ajuda os sistemas IDS/IPS a identificar tráfego malicioso e a tomada de decisões com base no conteúdo desses pacotes. Contudo, à medida que a adoção de protocolos mais seguros e uma maior encriptação no tráfego de rede se intensifica (David Warburton, 2021), os sistemas tradicionais de DPI veem a sua eficiência comprometida, dado que os *payloads* estão encriptados. Para se adaptarem à evolução tecnológica e à crescente encriptação de dados em redes, os sistemas de DPI estão a adaptar novas abordagens baseadas em técnicas de *machine learning* e *deep learning* para analisar o tráfego encriptado (Papadogiannaki & Ioannidis, 2021). Estas técnicas, quando utilizadas em combinação com sistemas IDS/IPS, permitem enfrentar a complexidade e a diversidade dos ambientes de *smart homes*, mantendo,

simultaneamente, a capacidade de identificar e mitigar possíveis ameaças de forma eficiente.

A exploração eficiente das informações geradas pelos sistemas de deteção e prevenção de intrusão (IDS/IPS) é crucial para assegurar a segurança das infraestruturas de rede. Neste contexto, torna-se necessário optar por um sistema que permita a recolha sistemática, indexação e visualização desses dados de forma eficaz. Na literatura científica, destaca-se o conjunto de ferramentas *Elasticsearch-Logstash-Kibana (*ELK*),* pela sua facilidade de utilização e pela capacidade de produzir visualizações relevantes para os utilizadores finais (Dharur & Swaminathan, 2018).

Um estudo recente realizado por (Tokar et al., 2022) demonstrou a viabilidade do *stack* ELK em diversos cenários de *Internet* das Coisas (*IoT*), ressaltando a sua capacidade de gerir grandes volumes de dados, fornecendo escalabilidade, balanceamento de carga automático e tolerância a falhas num sistema de gestão relativamente simples de utilizar. Adicionalmente, foi realizada uma análise aprofundada da eficiência e desempenho desta solução, validando assim a escolha do *stack* ELK como ferramenta de análise de dados IDS/IPS*,* como uma solução eficaz para a gestão de grandes volumes de dados no contexto de IoT.

No âmbito da segurança de uma *smart home*, a adoção da *stack* ELK pode trazer inúmeros benefícios. Ao permitir a análise e visualização de dados gerados pelos sistemas IDS/IPS, os utilizadores podem identificar padrões, detetar anomalias e compreender melhor o comportamento da rede, possibilitando a implementação de medidas de segurança mais eficientes e uma resposta rápida a potenciais ameaças.

A utilização do *Raspberry Pi*, um minicomputador com capacidades limitadas em termos de memória e processamento, tem ganho espaço como uma solução viável e de baixo custo para implementar sistemas de segurança em ambientes IoT, como as *smart homes.* Estudos como o de (Sforzin et al., 2016), demonstram a capacidade do *Raspberry Pi* em correr um sistema IDS adaptado a uma arquitetura IoT. Baseado no IDS *Snort,* conclui-se que é possível executar este tipo de sistemas em dispositivos com recursos limitados. Um outro exemplo, o estudo de (Tirumala et al., 2022), explora a implementação de um sistema inteligente de defesa para redes típicas em Pequenas e Médias Empresas (PMEs) e *smart homes.* A solução utiliza modelos de inteligência artificial e *machine learning* para detetar tráfego malicioso e *malware,* avaliando também a capacidade de resposta e desempenho do *Raspberry Pi.* Este tipo de equipamento tem-se mostrado uma opção viável para implementar soluções em ambientes IoT devido à sua flexibilidade e a uma vasta comunidade de contribuidores.

No entanto, é importante ter em conta as suas limitações de desempenho, entre outras desvantagens, de modo a garantir a adequação desta solução às necessidades específicas de cada ambiente IoT.

O crescimento da popularidade das *smart homes* e a integração de dispositivos IoT acarretam diversos perigos, incluindo preocupações relacionadas com a privacidade dos utilizadores. Uma prática comum entre fabricantes de dispositivos IoT é o rastreamento do comportamento dos utilizadores, sem o seu conhecimento, para fins de publicidade direcionada, gerando lucros significativos (Niu, 2018; Samsung, 2020). Neste contexto, é crucial explorar soluções para proteger a privacidade dos utilizadores, como a utilização de um DNS *Sinkhole*, nomeadamente a solução *Pi-Hole* (Mazhar & Shafiq, 2020).

O *Pi-Hole* é uma solução eficaz que funciona como bloqueador de anúncios e *trackers* a nível da rede, impedindo que dispositivos IoT acedam a domínios associados a serviços de rastreamento e publicidade (Mohajeri Moghaddam et al., 2019). Ao implementar soluções baseadas em DNS *sinkhole*, os utilizadores protegem a sua privacidade e controlam o fluxo de informações nas redes domésticas. O *Pi-Hole*, além de proteger a privacidade, melhora o desempenho da rede ao reduzir o consumo de largura de banda. A implementação de soluções como o *Pi-Hole* representa uma estratégia de proteção de privacidade viável em ambientes IoT residenciais. No entanto, é fundamental considerar medidas de segurança adicionais para garantir uma proteção abrangente aos utilizadores.

Em suma, a segurança das *smart homes* é um domínio de investigação multidisciplinar e complexo, que engloba áreas como, protocolos de comunicação, gestão de privacidade e integridade de dados e dispositivos, autenticação e autorização, bem como modelos de governo das soluções IoT. A heterogeneidade destas soluções cria desafios na identificação de uma abordagem universal que resolva todos os problemas de segurança. Contudo, vários estudos têm sido desenvolvidos para abordar problemas específicos, ainda que nem sempre sejam diretamente aplicáveis a casos reais ou sejam dispendiosos e complexos para o utilizador final.

Nesta revisão da literatura, foram analisados diversos trabalhos e soluções que visam melhorar a segurança e privacidade nas *smart homes*. Estes esforços têm sido de grande importância para o avanço do conhecimento nesta área, identificando lacunas e oportunidades para o desenvolvimento de soluções mais eficazes e acessíveis.

Com base no conhecimento adquirido nesta revisão, o próximo capítulo focar-se-á na proposta de uma solução que melhore a segurança das *smart homes*. Esta solução será concebida com o objetivo de ser simples e acessível ao utilizador final, destacando-se pela sua abordagem centrada no desenvolvimento de uma solução económica e eficiente.

3. Arquitetura da solução

Neste capítulo, apresentamos a solução proposta com base na revisão da literatura realizada e nos objetivos estabelecidos para melhorar a segurança das *smart homes*. A solução foi concebida levando em consideração os desafios e lacunas identificados na literatura, procurando oferecer uma abordagem inovadora, prática e acessível ao utilizador final. É composta por várias ferramentas e tecnologias, que serão detalhadas nas próximas secções deste capítulo.

3.1. Arquitetura

Nesta secção, descrevemos a arquitetura geral da solução proposta, abordando os seus componentes principais, assim como as interações entre eles. A arquitetura deve ser flexível e escalável, permitindo a integração de diferentes dispositivos IoT e a adaptação a diferentes cenários de *smart homes*.

A solução proposta para melhorar a segurança e privacidade das *smart homes* assenta na implementação de dois componentes principais que estão interligados e se complementam, ambos baseados em dispositivos *Raspberry Pi*. Estes são: um *hotspot* e um sistema de monitorização que incorpora um sistema de deteção de intrusão (IDS). Esta comunicação recíproca entre os dois módulos é de crucial importância para a garantia de eficácia da solução, dado que assegura a privacidade e a segurança dos utilizadores das *smart homes*.

A Figura 4 evidencia a arquitetura da solução proposta, dando especial enfoque a todas as ferramentas e tecnologias que desempenham um papel fundamental na garantia da segurança das *smart homes*. Esta estrutura ilustrativa oferece uma visão clara da integração dos dois componentes principais, o *hotspot* e o sistema de monitorização com o IDS, além de demonstrar os mecanismos de comunicação existentes entre eles, que reforçam a coesão e a interdependência entre as diferentes partes da solução.

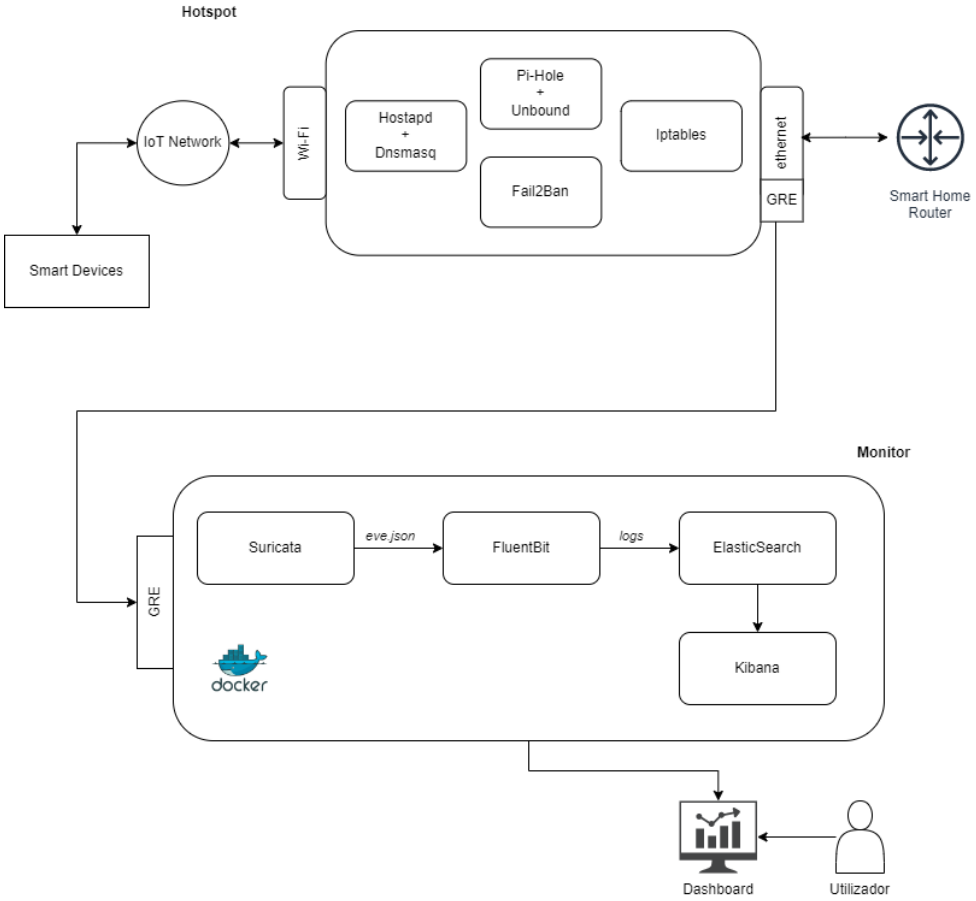


Figura 4 - Esquema da arquitetura proposta

O módulo *hotspot*, sendo o ponto central da arquitetura proposta, estabelece uma rede segregada à qual os dispositivos *IoT* da *smart home* estão conectados, isolando-os dos demais dispositivos presentes na rede doméstica. Esta segregação é crucial para evitar a propagação de eventuais falhas de segurança a dispositivos sensíveis, como os computadores pessoais dos habitantes da *smart home*. O *hotspot* utiliza protocolos de criptografia e autenticação robustos, garantindo a privacidade e integridade dos dados transmitidos entre os dispositivos IoT e a rede isolada.

O segundo módulo, que é responsável por monitorizar o tráfego de rede na *smart home* e identificar anomalias e atividades suspeitas, inclui um IDS. Este IDS recorre a técnicas avançadas de análise de tráfego de rede, como análise de assinaturas e inspeção profunda de pacotes (DPI), para detetar potenciais ameaças e alertar o utilizador em tempo real. A integração entre o *hotspot* e o IDS permite a partilha de informações relevantes sobre o estado da rede e eventuais incidentes de segurança, possibilitando uma resposta rápida e eficiente a possíveis ameaças.

A opção de utilizar exclusivamente um sistema de deteção de intrusão (IDS), em detrimento de um sistema de prevenção de intrusão (IPS), no contexto *das smart*

homes, deve-se sobretudo à intenção de preservar a simplicidade e a facilidade de gestão da solução proposta. O IDS é responsável apenas pela monitorização da rede, enquanto o IPS requer capacidades adicionais para bloquear ou prevenir automaticamente atividades maliciosas.

No cenário das *smart homes*, a adoção de um IDS simplifica a implementação e a administração da solução, proporcionando maior controlo ao utilizador, que pode analisar os alertas e decidir sobre as ações a serem tomadas. Isto é particularmente relevante em ambientes domésticos, onde os utilizadores podem não possuir conhecimentos técnicos avançados e, portanto, beneficiar de uma solução menos complexa.

Adicionalmente, os falsos positivos são menos problemáticos em sistemas IDS quando comparados aos sistemas IPS. Nos sistemas IDS, os falsos positivos geram alertas que podem ser investigados pelo utilizador, ao passo que, nos sistemas IPS, a ocorrência de falsos positivos pode levar a interrupções indesejadas no funcionamento normal da rede, devido à interferência automática do sistema na tentativa de bloquear a atividade identificada como maliciosa.

3.2. Módulo 1 - *Hostpot*

O módulo "*hotspot*", apresentado na Figura 5, desempenha um papel fundamental na criação de um ponto de acesso central para todos os dispositivos IoT numa *smart home*. Diversas ferramentas e soluções foram aplicadas no desenvolvimento deste módulo, visando salvaguardar a segurança e a privacidade dos utilizadores. Seguidamente, apresenta-se uma descrição detalhada e técnica de como as ferramentas interagem entre si para garantir uma segurança aprimorada.

- ***Hostapd e Dnsmasq*:** O *Hostapd* e o *Dnsmasq* trabalham em conjunto para criar um ponto de acesso isolado e gerir os serviços de DNS e DHCP. Através da segmentação da rede, os dispositivos IoT são isolados dos restantes dispositivos presentes na habitação dos utilizadores, reduzindo assim, a superfície de ataque e minimizando a propagação de potenciais ameaças.
- ***IPTables* e *Fail2Ban*:** As *IPTables* e o *Fail2Ban* são utilizados simultaneamente para proporcionar uma defesa em camadas contra ataques e intrusões. As *IPTables*, filtram o tráfego de rede com base em regras predefinidas, bloqueando o tráfego suspeito e malicioso. Já o *Fail2Ban* analisa os registos do sistema em tempo real, identificando e bloqueando endereços

IP com comportamento suspeito, como por exemplo, sucessivas tentativas de autenticação falhadas. Esta combinação de filtragem de pacotes e análise de registos gera uma barreira de segurança mais resistente a potenciais ataques e intrusões, tais como ataques DDoS.

- ***Unbound* e *Pi-Hole***: O *Unbound* e o *Pi-Hole* são utilizados em conjunto para garantir a segurança e a privacidade das comunicações DNS. O *Unbound* proporciona resolução de nomes de domínio segura e privada, evitando a dependência de servidores DNS públicos ou fornecidos pelo ISP. Já o *Pi-Hole*, enquanto DNS *Sinkhole*, bloqueia o acesso a sites maliciosos ou suspeitos. A interação entre o *Unbound* e o *Pi-Hole* permite a proteção contra ataques de *spoofing* e *phishing*, garantindo a integridade das comunicações DNS e protegendo a privacidade dos utilizadores.

A interligação das ferramentas acima mencionadas permite criar um ambiente de segurança abrangente e robusto. Os dispositivos IoT comunicam com o *hotspot* através dos serviços *hostapd* e do *dnsmasq*, que fornecem serviços de rede essenciais e mantêm o tráfego isolado. Paralelamente, *IPTables* e *Fail2Ban* protegem a rede contra-ataques e intrusões. O *Unbound* e *Pi-Hole* asseguram, por sua vez, a segurança das comunicações DNS. A integração destas ferramentas cria uma arquitetura de segurança resiliente, capaz de se adaptar a um cenário de ameaças em constante evolução. No entanto, é importante salientar que a implementação destas medidas de segurança não é algo estático, mas sim um processo em constante atualização, que requer monitorização adequada para garantir a proteção eficaz e contínua dos dados dos utilizadores.

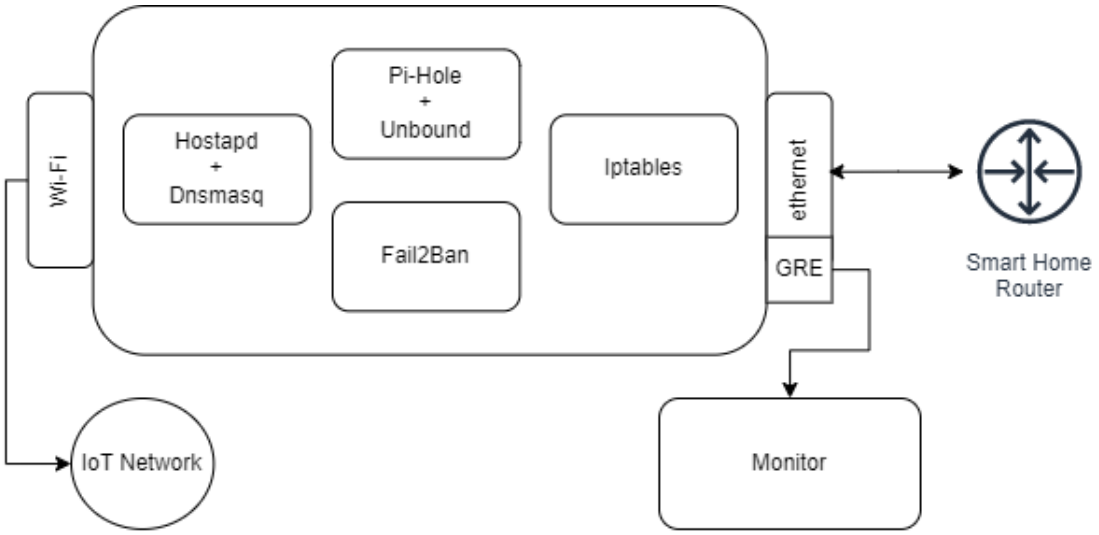


Figura 5 – Esquema do módulo "*Hotspot*"

3.3. Módulo 2 – Monitorização

O segundo módulo, apresentado na Figura 6, incorpora um sistema de deteção de intrusão (IDS), baseado na solução *open-source Suricata*. Este módulo tem como principal objetivo monitorizar o tráfego proveniente do módulo "*hotspot*", alertando o utilizador para eventuais intrusões e atividades suspeitas. Para facilitar a consulta e análise dos alertas emitidos pelo IDS, este módulo utiliza a *stack* ELK (*Elasticsearch, Logstash e Kibana*), com o intuito de criar *dashboards*, proporcionando assim, uma perceção visual mais aprofundada dos eventos de segurança.

A solução *Suricata* é um motor *de* IDS/IPS de código aberto, poderoso e flexível, desenvolvido para deteção e resposta a ameaças emergentes e sofisticadas em ambientes de rede, em tempo real. Algumas das suas principais características incluem integração fácil com outras soluções, a monitorização de segurança de rede a partir de um conjunto de regras e assinaturas de ameaças, mantidas e atualizadas pela comunidade, para identificar padrões de tráfego associados a ataques conhecidos, *malware* e outras atividades suspeitas. Além disso, regista e analisa o tráfego TLS/SSL, HTTP e DNS, apresenta alto desempenho, deteção automática de protocolos, e permite *scripting Lua*.

Para complementar a solução *Suricata* e proporcionar uma melhor visualização e análise dos alertas gerados pelo IDS, este módulo integra a *stack* ELK, composta pelos seguintes componentes:

- ***Elasticsearch*:** É um motor de pesquisa e análise distribuída, que armazena e indexa os dados provenientes do *Suricata*, permitindo consultas rápidas e eficientes.
- ***Logstash*:** Representa uma *pipeline* de processamento de dados, que recebe os alertas e eventos do *Suricata*, processa-os e encaminha-os para o *Elasticsearch*. O *Logstash* é responsável por agregar, transformar e enriquecer os dados antes de serem armazenados no *Elasticsearch*. Contudo, no âmbito deste trabalho, este módulo foi substituído pelo *Fluentbit*. A troca justifica-se pela similaridade de propósitos entre as duas ferramentas, contudo, o *Fluentbit* apresenta-se como uma opção mais leve em termos de consumo de recursos, o que se traduz numa maior eficiência e adaptabilidade para o contexto do projeto.
- ***Kibana*:** É uma plataforma de análise e visualização de dados, que permite criar *dashboards* interativos e personalizados, baseados nos dados armazenados no *Elasticsearch*. Através do *Kibana*, os utilizadores podem

explorar e analisar os alertas e eventos de segurança gerados pelo *Suricata*, identificando tendências, padrões e possíveis ameaças à rede.

Em suma, a implementação do módulo IDS, baseado na solução *Suricata* em conjunto com a *stack* ELK, proporciona uma abordagem abrangente e eficaz na monitorização e deteção de ameaças à segurança no ambiente das *smart homes*. Através da integração destas tecnologias, os utilizadores são capacitados para identificar, analisar e reagir a possíveis intrusões e atividades maliciosas na rede, protegendo, deste modo, os seus dispositivos IoT e dados sensíveis. Adicionalmente, a utilização *da stack* ELK para visualizar e analisar os dados provenientes do *Suricata* facilita a compreensão e a resposta a eventos de segurança, promovendo a adoção de medidas corretivas e preventivas.

É importante salientar que, à semelhança do módulo "*hotspot*", a implementação e manutenção deste módulo de monitorização requer um processo de atualização e adaptação constante, tendo em conta o cenário dinâmico das ameaças à segurança. A atualização regular das regras e assinaturas da *Suricata*, bem como a adaptação e personalização das configurações e *dashboards* da *stack* ELK, são aspetos cruciais para garantir uma proteção eficaz e adaptativa face às ameaças existentes e emergentes.

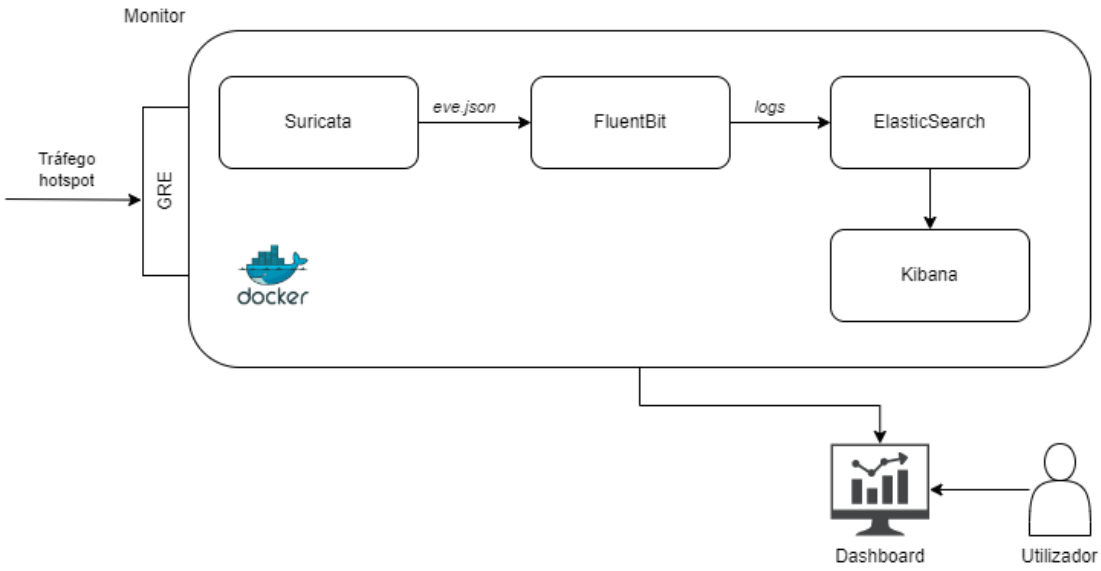


Figura 6 - Esquema do módulo "Monitor"

4. Ferramentas e Tecnologias

Neste capítulo, serão apresentadas as ferramentas e tecnologias selecionadas para suportar a solução proposta. As ferramentas foram escolhidas com base em critérios de eficácia, acessibilidade e facilidade de uso. Além disso, serão explicadas as razões pelas quais estas ferramentas e tecnologias foram escolhidas, em detrimento de outras opções disponíveis no mercado.

4.1. *IPTables*

As *firewalls*, elementos cruciais para a segurança de redes, podem ser classificadas em duas categorias: *firewalls* baseadas em *software* e *firewalls* baseadas em *hardware*. A principal função de ambas é filtrar o tráfego de dados conforme um conjunto de regras predefinidas, desempenhando um papel vital na criação de um ambiente de rede seguro. Contudo, é importante salientar que uma configuração inadequada da *firewall* pode resultar numa proteção ilusória, deixando a rede vulnerável a ataques. Os *hackers* estão constantemente em busca de novas técnicas e vulnerabilidades para explorar. Um simples erro na configuração da *firewall* pode proporcionar a um atacante o acesso indevido à rede.

Soluções de *firewall* tradicionais, como *Pfsense* e *FortiGate*, são reconhecidas pela sua robustez e eficácia na proteção de redes. No entanto, a implementação destas soluções requer uma maior quantidade de recursos em comparação com dispositivos de menor capacidade, como o *Raspberry Pi*. Essas limitações tornam-se mais evidentes ao analisar dois aspetos principais. Em primeiro lugar, o *hardware* do *Raspberry Pi* é menos poderoso em comparação com os dispositivos dedicados a soluções de *firewall*, como *Pfsense* e *FortiGate*. A capacidade de processamento, memória e outros recursos do *Raspberry Pi* são inferiores aos de dispositivos especializados, o que pode afetar a eficiência e a capacidade de lidar com grandes volumes de tráfego de rede. Em segundo lugar, o *Raspberry Pi* carece de uma segunda porta *Ethernet*, o que limita a sua capacidade de atuar como uma *firewall* eficiente. A presença de apenas uma porta *Ethernet* restringe a possibilidade de

separar fisicamente as redes interna e externa, o que é uma prática comum em soluções de *firewall* para garantir a segurança da rede. Esta limitação pode ser parcialmente contornada com o uso de adaptadores USB para *Ethernet* ou *switches* gerenciáveis, mas ainda assim, não é uma solução tão eficiente quanto a encontrada nos dispositivos dedicados a esse fim.

O *iptables* é uma ferramenta poderosa e flexível que faz parte do módulo *netfilter* disponível nos sistemas *Linux,* permitindo a criação de regras personalizadas para filtrar e controlar o tráfego de rede no dispositivo. As principais características do *iptables* incluem a divisão em "*Tables*", "*Chains*" e "*Targets*", que permitem uma organização eficiente das regras de filtragem de tráfego. As "*Tables*" são compostas por quatro tipos pré-definidos: "*Filter*", "*Nat*", "*Mangle*" e "Raw", cada uma com um conjunto de regras específicas. As "*Chains*" são responsáveis por determinar o ponto do fluxo de informação onde as regras serão processadas, sendo divididas em cinco categorias: "*PREROUTING*", "*INPUT*", "*FORWARD*", "*OUTPUT*" e "*POSTROUTING*". Já os "*Targets*" definem a ação a ser executada quando um pacote corresponde a uma regra, podendo ser "*Accept*", "*Drop*", "*Return*" ou "*Reject*". Um esquema de funcionamento do iptables pode ser visualizado na Figura 7.

A eficiência e robustez do *iptables* estão relacionadas com a sua capacidade de lidar com a complexidade das regras de filtragem e a flexibilidade na configuração da *firewall*. Além das regras básicas, é importante adicionar regras mais específicas de acordo com as necessidades da rede doméstica inteligente e dos dispositivos IoT presentes. Por exemplo, regras podem ser adicionadas para limitar o acesso a determinados endereços IP, portas ou protocolos, de acordo com a política de segurança desejada, garantindo assim um maior controlo e segurança no acesso às informações na *smart home.*

Apesar de existir uma alternativa mais recente, chamada *nftables*, a escolha pelo *iptables* neste projeto está relacionada com a maior disponibilidade de informação e documentação existente para esta ferramenta. Além disso, o tempo necessário para migrar as regras do *iptables* para o *nftables* poderia ser demorado e pouco vantajoso, tendo em conta que o *iptables* continua sendo uma solução robusta e eficiente.

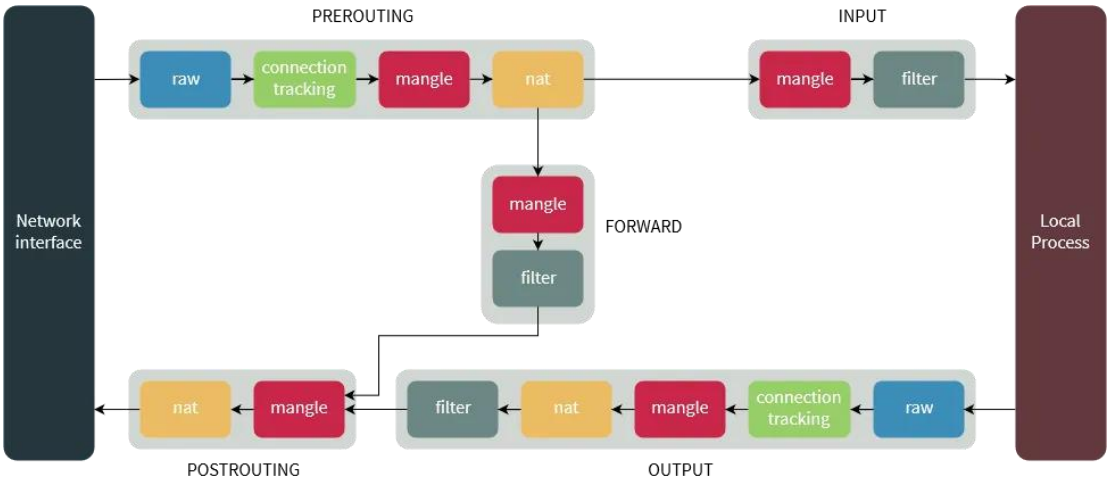


Figura 7 -Esquema *Iptables*(Supriyo Biswas, 2017)

Embora o *Raspberry Pi* não seja tão robusto quanto soluções de *firewall* dedicadas, ainda é possível melhorar significativamente a segurança de uma rede doméstica inteligente utilizando o *iptables* com um conjunto de regras bem definidas.

Em suma, é importante ter em mente as limitações apresentadas pelo *Raspberry Pi* em comparação com soluções especializadas, como *Pfsense* e *FortiGate*. Considerar as necessidades específicas de segurança e recursos da rede em questão é crucial ao escolher a solução de *firewall* mais adequada.

4.2. IPS *Fail2ban*

O *Fail2Ban* é uma ferramenta de segurança *open-source* que atua como um sistema de prevenção de intrusões (IPS), contribuindo na proteção de sistemas contra atividades maliciosas. A sua implementação é especialmente relevante em ambientes de *smart home*, onde a crescente interconexão de dispositivos e sistemas exige medidas de segurança robustas e eficientes.

A ferramenta funciona através da análise dos registos (*logs*) dos sistemas, procurando padrões suspeitos que correspondam a falhas de autenticação, *exploits* e outros registos maliciosos. Ao identificar uma atividade fora do padrão, o *Fail2Ban* interage com a *firewall* para bloquear o endereço IP associado. Este bloqueio tem uma duração predefinida, sendo o IP automaticamente desbloqueado após este período.

As principais vantagens do *Fail2Ban* incluem a sua capacidade de personalização, permitindo a configuração de regras específicas para proteger serviços como servidores *Apache* ou ligações remotas SSH. Além disso, o envio de

notificações aos administradores sempre que atividades suspeitas são detetadas possibilita uma rápida reação na eventualidade de um ataque.

A utilização do *Fail2Ban* em ambientes de *smart home* contribui para aumentar a segurança e a privacidade do sistema, protegendo contra uma variedade de ataques, como ataques de força bruta, tentativas de exploração de vulnerabilidades e ataques de negação de serviço distribuída (DDoS).

No entanto, apesar das suas vantagens, o *Fail2Ban* apresenta limitações que devem ser consideradas. Uma das desvantagens é a possibilidade de bloquear endereços IP legítimos, o que pode resultar em falsos positivos e afetar a utilização normal dos serviços. Além disso, a ferramenta pode ser insuficiente para proteger contra-ataques mais sofisticados, que exigem soluções de segurança mais avançadas.

Deste modo, é fundamental que a implementação do *Fail2Ban* seja complementada com outras medidas de segurança, como a atualização regular de *software*, a utilização de *firewalls* e a criação de senhas fortes. Desta forma, podemos concluir que o *Fail2Ban* é uma ferramenta valiosa na proteção de redes domésticas contra atividades maliciosas. A adoção de múltiplas camadas de proteção, incluindo ferramentas como o *Fail2Ban*, pode contribuir para a criação de um ambiente de rede doméstica mais seguro e resiliente.

4.3. DNS *Sinkhole*

Uma das várias medidas que podem melhorar significativamente a segurança das redes domésticas é utilizar ferramentas como um DNS *Sinkhole*, um servidor que fornece respostas falsas às consultas DNS, redirecionando o dispositivo cliente para um endereço IP controlado, em vez de permitir a conexão a domínios maliciosos. Esta técnica é eficaz na prevenção de conexões a *botnets* e servidores de comando e controlo (C2), ao interromper os nomes de domínio que estes utilizam para comunicar.

As listas de URLs maliciosos e servidores C2 conhecidos, provenientes de fontes abertas e comerciais, são essenciais para a eficiência do DNS *Sinkhole*. Ao configurar o servidor para fornecer endereços IP falsos para estes URLs específicos, o DNS *Sinkhole* evita conexões a domínios indesejados, protegendo a rede doméstica contra *spyware, botnets* e *downloads* perigosos.

Além disso, o DNS *Sinkhole* permite identificar dispositivos infetados na rede, a partir do registo das tentativas repetidas de conexão aos domínios bloqueados. Esta

informação é especialmente útil quando a *firewall* não consegue identificar a origem do pedido DNS, possibilitando à equipa de segurança agir rapidamente na deteção e resolução do problema.

Outra aplicação interessante do DNS *Sinkhole* é a restrição de acesso a *sites* que violem as políticas de utilização definidas pelos utilizadores das s*mart homes*. Ao tentar aceder a um URL bloqueado, o utilizador é redirecionado para uma página personalizada, que informa sobre a infração cometida. Ferramentas como o *Pi-Hole* (*Pi-Hole*, 2022), baseado em *Linux*, podem ser utilizadas para bloquear *sites* de anúncios em toda a rede doméstica.

O caso do ataque *WannaCry* em 2017 ilustra o potencial surpreendente do DNS *Sinkhole* (Lily Hay Newman, 2017). Ao registar um domínio não registado encontrado no código do *malware*, o investigador Marcus Hutchins ativou acidentalmente um interruptor que reduziu significativamente a propagação do ataque, permitindo a intervenção de especialistas.

A utilização de um DNS *sinkhole*, como o *Pi-Hole*, em *smart homes* apresenta inúmeras vantagens para os utilizadores, proporcionando uma experiência de navegação mais segura e agradável. Com a capacidade de bloquear, de forma centralizada e eficiente, anúncios e rastreadores em todos os dispositivos conectados à rede, o *Pi-Hole* contribui para o menor consumo de largura de banda e uma navegação mais rápida. Além disso, esta solução de código aberto impede a proliferação de *clickbaits* e, ao bloquear *cookies* de rastreamento, assegura uma maior privacidade para os utilizadores nas suas *smart homes*.

A segurança é também um fator crucial na escolha do *Pi-Hole* como solução de DNS *sinkhole*. Ao bloquear anúncios e outros tipos de conteúdos indesejados, cria-se uma proteção adicional que protege os dispositivos contra *malware*, *phishing* e outras ameaças. O *Pi-Hole* destaca-se no mercado pela sua facilidade de instalação, configurabilidade e integração com outras soluções de segurança, tornando-se uma opção eficaz para garantir a segurança e a privacidade nas *smart homes*.

4.4. IPS/IDS

Os sistemas de deteção e prevenção de intrusão (IDS/IPS) são ferramentas essenciais para garantir a segurança de redes de computadores e dispositivos conectados, como por exemplo, em contextos de *smart homes*. Com o aumento do

número de dispositivos conectados à *Internet* e a crescente sofisticação das ameaças cibernéticas, é crucial adotar soluções que ofereçam proteção abrangente e eficaz.

*Intrusion Detection System (*IDS*)* é uma ferramenta de monitorização de rede e *endpoints,* para deteção de intrusão, ameaças ou outras atividades maliciosas. É uma ferramenta passiva, uma vez que apenas informa os utilizadores da ocorrência de uma anomalia em tempo real, não tendo, por isso, uma ação direta na prevenção ou correção de uma possível anomalia. Por outro lado, os sistemas *Intrusion Prevention System (*IPS*)* atuam de forma ativa na prevenção de ataques.

Não existe uma opção "correta" entre a adoção de um outro sistema, uma vez que ambos têm funções distintas e deve-se compreender qual desempenha a melhor função no ambiente em que serão implementados. Na Figura 8 podemos visualizar um exemplo concreto das diferenças mais significativas na implementação de ambos os sistemas. Para entender melhor os sistemas IDS/IPS, é importante explorar as principais abordagens utilizadas para identificar atividades maliciosas: o reconhecimento de assinatura *(Signature Recognition*) e a deteção de anomalias (*Anomaly Detection*).

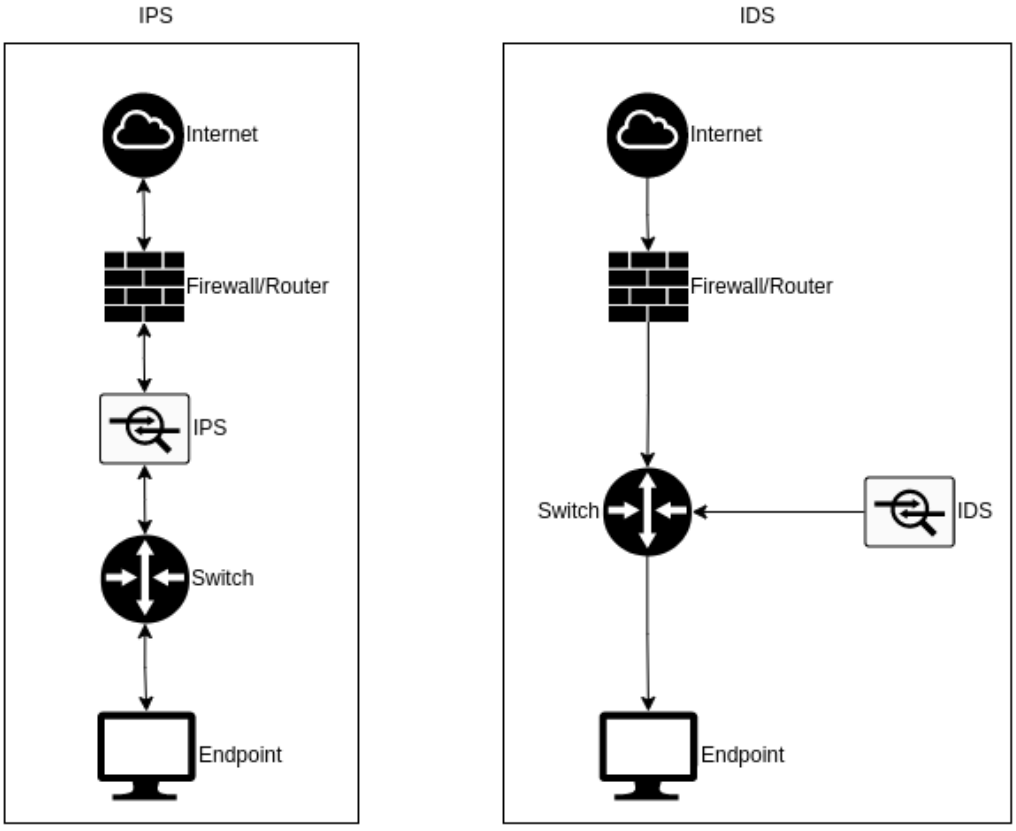


Figura 8 - Diferenças entre IDS e IPS

O reconhecimento de assinaturas envolve o uso de padrões específicos, conhecidos como assinaturas, que representam características de atividades maliciosas. Os sistemas IDS/IPS analisam os pacotes de dados e comparam-nos com as assinaturas, identificando possíveis ameaças. No entanto, essa abordagem tem algumas limitações, como a dificuldade em detetar ataques *zero-day* e a complexidade na configuração de regras a serem aplicadas.

Por outro lado, a deteção de anomalias baseia-se na identificação de desvios em relação ao comportamento normal. Os sistemas IDS/IPS monitorizam a rede e comparam a atividade atual com o padrão normal estabelecido, alertando os utilizadores caso identifiquem atividades incomuns. Esta abordagem permite identificar ataques que um sistema baseado em assinaturas não seria capaz de detetar, mas também apresenta desafios, como a necessidade de estabelecer uma linha de base confiável e a possibilidade de gerar mais falsos positivos.

Os sistemas IDS/IPS podem ser classificados como "*Network Intrusion Detection System"* (NIDS) ou "*Host-based Intrusion Detection System*" (*HIDS*), dependendo de onde são implementados e como operam.

Os sistemas NIDS monitorizam a atividade da rede, rastreando e detetando possíveis anomalias. Eles recebem uma cópia do tráfego da rede local, através de uma porta espelhada (SPAN) ou de um "*network tap*". Com base em regras previamente definidas, os NIDS analisam os dados e decidem se devem ignorar ou lançar alertas ao utilizador. Soluções populares de NIDS incluem *Snort*, *Suricata* e *Zeek*.

Os sistemas HIDS, por sua vez, são instalados diretamente nos dispositivos e monitorizam de perto as atividades realizadas nestes. Procuram alterações em ficheiros, *logs* e processos de sistema, verificam a integridade dos dados e detetam possíveis vulnerabilidades e violações de políticas de segurança. Exemplos de soluções HIDS incluem *OSSEC, Samhain, Wazuh* e *Fail2Ban*.

Neste contexto, o *Suricata* surge como uma solução robusta e flexível, desenvolvida para garantir a deteção e resposta a ameaças emergentes e sofisticadas em ambientes de redes, incluindo *smart homes*. As principais características do *Suricata* incluem:

- **Integração fácil com outras soluções:** O *Suricata* integra-se perfeitamente na rede e pode ser incorporado em várias soluções comerciais e de código aberto respeitadas;

- **Independência:** O projeto e código do *Suricata* são detidos e apoiados pela *Open Information Security Foundation* (OISF), uma organização sem fins lucrativos comprometida em manter o *Suricata* de código aberto permanentemente.
- **Monitorização de Segurança de Rede (NSM):** O *Suricata* não só atua como um Sistema de Deteção e Prevenção de Intrusões (IDS/IPS), mas também pode registar solicitações HTTP, armazenar certificados TLS, extrair arquivos dos fluxos e armazená-los em disco. O suporte completo para captura *pcap* facilita a análise e torna o *Suricata* um componente essencial para o ecossistema de NSM.
- **Registo e análise TLS/SSL:** A ferramenta permite não apenas corresponder a vários aspetos de uma troca TLS/SSL através da sua linguagem de regras, mas também registar todas as trocas de chaves para análise.
- **Registo HTTP:** É possível registar todas as conexões HTTP em qualquer porta para análise posterior, evitando a necessidade de adicionar mais *hardware* à rede.
- **Registo DNS:** A ferramenta regista todas as consultas e respostas DNS.
- **Deteção de ameaças e prevenção de intrusões:** A solução implementa uma linguagem completa de assinaturas para identificar ameaças conhecidas, violações de políticas e comportamentos maliciosos, utilizando regras especializadas do *Emerging Threats Suricata* e do conjunto de regras VRT.
- **Alto desempenho:** Uma única instância do *Suricata* é capaz de inspecionar tráfego de *multi-gigabit*. A ferramenta possui um motor construído em torno de uma base de código moderna, limpa, altamente escalável e *multi-threaded*. Além disso, oferece suporte nativo para aceleração de *hardware* de vários fornecedores e através do PF_RING e AF_PACKET.
- **Deteção automática de protocolos:** Deteção automática de protocolos como HTTP em qualquer porta e aplica a lógica de deteção e registo adequada.
- ***Scripting Lua*:** Permite análises avançadas e funcionalidades para detetar elementos que não seriam possíveis apenas com a sintaxe do conjunto de regras.
- **Saídas padrão do setor:** A principal saída de registo do *Suricata* é chamada de "*Eve*", um registo de eventos e alertas em formato JSON, facilitando a integração com ferramentas como *Logstash*.

Em contextos de *smart homes*, o *Suricata* oferece várias vantagens, incluindo a capacidade de proteger efetivamente um ambiente cada vez mais complexo e interconectado. Esta solução pode ser utilizada para monitorizar o tráfego entre dispositivos inteligentes, como câmaras de segurança, sistemas de climatização, iluminação e outros dispositivos conectados, identificando e bloqueando potenciais ataques. Ao analisar continuamente a atividade da rede, o *Suricata* pode detetar comportamentos incomuns e responder rapidamente a ameaças emergentes, protegendo assim os dispositivos e dados dos utilizadores.

Adicionalmente, a capacidade de integrar-se com outras soluções de segurança torna o *Suricata* uma escolha popular em ambientes de *smart homes*. Ao trabalhar em conjunto com *firewalls*, sistemas de prevenção de perda de dados (DLP) e outras soluções de segurança, o *Suricata* ajuda a construir um ecossistema de segurança robusto e fiável.

Algumas das principais vantagens da utilização do *Suricata* em contextos de *smart homes* incluem:

- **Personalização:** É permitido aos utilizadores criar e personalizar regras de deteção, o que lhes dá a capacidade de adaptar a solução às suas necessidades específicas.
- **Monitorização abrangente:** O *Suricata* monitoriza e analisa várias camadas do tráfego da rede, desde os protocolos de aplicação até os protocolos de transporte e de rede, garantindo uma análise abrangente e a deteção de ameaças em tempo real.
- **Proteção contra ataques *zero-day*:** A capacidade de deteção de anomalias do *Suricata* permite que a ferramenta identifique ameaças desconhecidas e ataques *zero-day*, que são um risco significativo para *smart homes* e dispositivos IoT.
- **Redução do tempo de resposta a incidentes:** A análise em tempo real e a deteção de ameaças do *Suricata* permitem aos administradores de rede e utilizadores identificar e responder a incidentes de segurança rapidamente, reduzindo o potencial de danos.
- **Fácil integração e escalabilidade:** O *Suricata* pode ser facilmente integrada a outros sistemas de segurança e ferramentas de monitorização, e é escalável para acomodar redes em crescimento.

- **Comunidade e suporte:** A ferramenta é apoiada por uma comunidade de desenvolvedores ativa e em crescimento, bem como por empresas de cibersegurança e especialistas em todo o mundo. Isso garante a disponibilidade de suporte e recursos para manter a ferramenta atualizada e eficaz.

Em conclusão, o *Suricata* é uma solução poderosa e flexível para proteger redes e dispositivos em contextos de *smart homes*. A sua capacidade de detetar ameaças em tempo real e integrar-se com outras soluções de segurança faz dela uma escolha ideal para garantir a segurança e privacidade dos utilizadores.

4.5. *Stack* ELK

A implementação de sistemas eficientes e flexíveis de monitorização e análise de dados é fundamental para garantir a segurança e o correto funcionamento dos diversos dispositivos presentes numa *smart home*. Neste contexto, o presente texto explora a utilização do *stack* ELK, composto pelos componentes *Elasticsearch, Logstash* e *Kibana*, como solução para monitorizar e analisar dados de *logs* gerados por vários dispositivos, incluindo aqueles presentes numa *smart home.* A discussão também abordará a sua integração com sistemas de deteção e prevenção de intrusões (IDS/IPS), como o *Suricata*.

O *stack* ELK é uma plataforma de análise de dados formada por três componentes principais: *Elasticsearch, Logstash* e *Kibana*. Em conjunto, estes componentes possibilitam a recolha, armazenamento, análise e visualização de dados de *logs* provenientes de diferentes fontes, facilitando a identificação de padrões e tendências.

- ***Elasticsearch*:** Motor de pesquisa e análise de dados distribuído e baseado em RESTful, capaz de armazenar, pesquisar e analisar grandes volumes de dados em tempo real. Devido à sua escalabilidade e habilidade para lidar com dados não estruturados, o *Elasticsearch* é adequado para monitorar eventos de segurança e analisar *logs* gerados por dispositivos IoT.
- ***Logstash*:** Ferramenta de processamento de dados responsável pela recolha, transformação e envio de *logs* de distintas fontes para o *Elasticsearch*. Através de plugins e configurações personalizadas, o *Logstash* permite a normalização e enriquecimento dos dados, facilitando a análise e correlação de eventos de segurança.

- ***Kibana*:** Interface de visualização de dados que se integra com o *Elasticsearch*. Esta ferramenta permite a criação de gráficos, tabelas, mapas de calor, entre outras visualizações, para facilitar a análise e interpretação dos dados armazenados. Além disso, o *Kibana* oferece funcionalidades como alertas e notificações personalizadas, bem como painéis de controlo interativos e compartilháveis.

A integração do *stack* ELK com sistemas de deteção e prevenção de intrusões, como o *Suricata*, potencia a eficácia na identificação e resposta a ameaças à cibersegurança. O *Suricata* é um IDS/IPS de código aberto e de alto desempenho que analisa o tráfego de rede em tempo real e gera alertas com base em regras e assinaturas de ameaças conhecidas. Essa integração facilita a identificação de ataques e a resposta a incidentes de segurança, reduzindo o tempo de reação e os riscos associados.

Ao combinar as capacidades de análise e visualização do *stack* ELK com os dados gerados pelo *Suricata*, é possível identificar padrões de tráfego anómalos, detetar tentativas de intrusão, prevenir ataques DDoS e identificar dispositivos desconhecidos adicionados à rede. Deste modo, a integração entre o *stack* ELK e o *Suricata* proporciona uma visão mais abrangente e detalhada das atividades de rede, permitindo aos administradores e especialistas em segurança agir de forma mais eficaz e proativa.

O conjunto de ferramentas ELK é altamente flexível e escalável, o que o torna capaz de lidar com um grande volume de dados e adaptar-se às necessidades específicas de cada ambiente de casa inteligente. Esta solução proporciona uma visão abrangente das atividades de rede e facilita a identificação de ameaças e vulnerabilidades de segurança.

Em conclusão, a popularidade do conjunto de ferramentas ELK deve-se a uma combinação de fatores, incluindo o seu carácter de código aberto, flexibilidade, escalabilidade e capacidade de integração com sistemas de segurança, como o *Suricata*. Estas características tornam o conjunto de ferramentas ELK uma solução ideal para monitorização e análise de dados em ambientes *smart home*, contribuindo para uma maior proteção e eficiência na identificação e resposta a incidentes de cibersegurança. Ao adotar o conjunto de ferramentas ELK, os administradores e especialistas em segurança podem obter uma visão detalhada e abrangente das atividades de rede, permitindo uma atuação mais eficaz e proactiva na garantia da segurança e privacidade dos utilizadores de casas inteligentes.

4.7. *Port Mirroring*

De forma a possibilitar a visualização do tráfego do *hotspot* pelo IDS, é crucial ativar a funcionalidade de *port mirroring*. Esta funcionalidade consiste em encaminhar uma cópia do tráfego, tanto enviado como recebido, de uma porta de origem para uma porta de destino. Neste sentido, foram analisadas três opções: ativar o *port mirroring* diretamente no *switch* (SPAN), o redireccionamento através do *IPTables* (Módulo TEE) e a utilização de um túnel GRE.

A ativação do *port mirroring* (SPAN) no *switch*, apesar de ser a opção mais simples, não se adequa aos objetivos deste projeto. Pretende-se que o utilizador tenha a mínima intervenção possível no sistema, e a configuração do *switch* pode ser complexa para utilizadores com menos experiência ou em situações onde o acesso ao *switch* é limitado.

O redireccionamento do tráfego para o dispositivo IDS, utilizando o módulo TEE do *iptables,* apresenta também algumas desvantagens. Esta opção atribui à *firewall* duas funções distintas, aumentando substancialmente a carga sobre a mesma. Além disso, o módulo TEE modifica o endereço MAC da máquina original (*src*) pelo MAC do anfitrião onde é executado, fazendo com que os dados originais sejam perdidos. Esta perda de informação pode dificultar a análise e resposta a incidentes de segurança.

O encaminhamento do tráfego para o IDS através de um túnel com o protocolo *Generic Routing Encapsulation* (GRE) surge como uma alternativa mais viável e eficiente. O túnel GRE tem como principal função encapsular o pacote que se pretende enviar para um ou mais destinos e proceder ao respetivo envio. O seu funcionamento assemelha-se ao do protocolo UDP, que é "*connectionless*" e não possui controlo de receção dos pacotes. Durante a fase de encapsulamento, é possível optar por encapsular apenas o cabeçalho IP (*gretun*) ou todo o *frame* (*gretap*).

A utilização do túnel GRE oferece uma maior flexibilidade na configuração do sistema e facilita a integração com o IDS, garantindo a preservação dos dados originais para análise e resposta a incidentes de segurança. Além disso, esta abordagem pode ser implementada sem a necessidade de intervenção direta no *switch*, simplificando o processo para o utilizador.

Em suma, a implementação de um túnel GRE parece ser a opção mais adequada e eficaz para permitir que o tráfego do *hotspot* seja visualizado pelo sistema IDS. Esta abordagem proporciona a flexibilidade e a preservação dos dados necessários para uma análise eficiente das ameaças à rede e pode ser implementada

com um mínimo de intervenção por parte do utilizador, tornando-se a escolha ideal para o projeto.

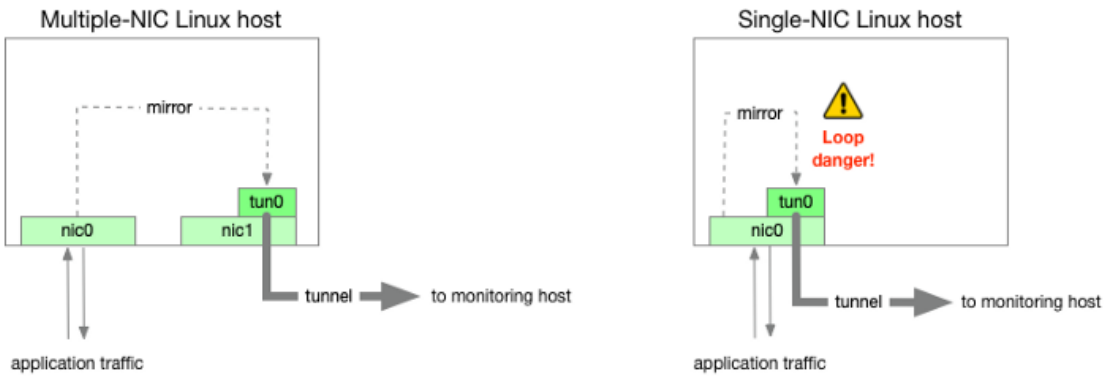


Figura 9 - Esquema do túnel GRE (David Waiting, 2020)

5. Implementação

Neste capítulo, dedicado à implementação do projeto proposto no âmbito deste trabalho, serão abordados os aspetos práticos e técnicos relacionados com o desenvolvimento de uma solução de segurança para *smart homes*. Com base nos objetivos estabelecidos no Capítulo 1, e assente na revisão bibliográfica realizada, ficou clara a importância da segurança e privacidade dos utilizadores de *smart homes*, bem como as diversas abordagens e técnicas que têm sido investigadas neste contexto.

A solução proposta neste capítulo visa aumentar o nível de proteção em ambientes de *smart homes,* através da utilização de ferramentas *open-source*, executadas num minicomputador *Raspberry Pi*. Serão apresentadas as principais etapas do processo de implementação da arquitetura proposta no capítulo 3, desde a seleção das ferramentas adequadas até à integração das mesmas no ambiente de *smart home*. Além disso, serão discutidas as principais considerações técnicas e desafios encontrados ao longo do processo.

Posteriormente, será realizada uma avaliação da solução proposta em termos de segurança, identificando os benefícios e limitações desta abordagem, e analisando o seu impacto no nível de proteção proporcionado aos utilizadores de *smart homes*. Deste modo, o presente capítulo detalha a implementação prática do projeto, fornecendo uma base sólida para a análise dos resultados obtidos e das conclusões a retirar.

5.1. Cenário de implementação

O cenário desenvolvido para a solução proposta, ilustrado na Figura 10, representa a configuração típica de uma *smart home*, abrangendo dispositivos comuns como *smart gateway* (*Amazon Echo Dot*), *smart devices*, que incluem tomadas inteligentes e iluminação, dispositivos de automação com ligação *Zigbee* e uma placa

Arduino, responsável pela aquisição de dados e subsequente envio para a *cloud* através de protocolos MQTT. Neste cenário, assume-se que as comunicações dos dispositivos IoT baseiam-se no protocolo *802.11* ou em algum *gateway/hub* que assegure a troca de informações entre protocolos (por exemplo, *Zigbee* para *Wi-Fi*) para comunicações externas.

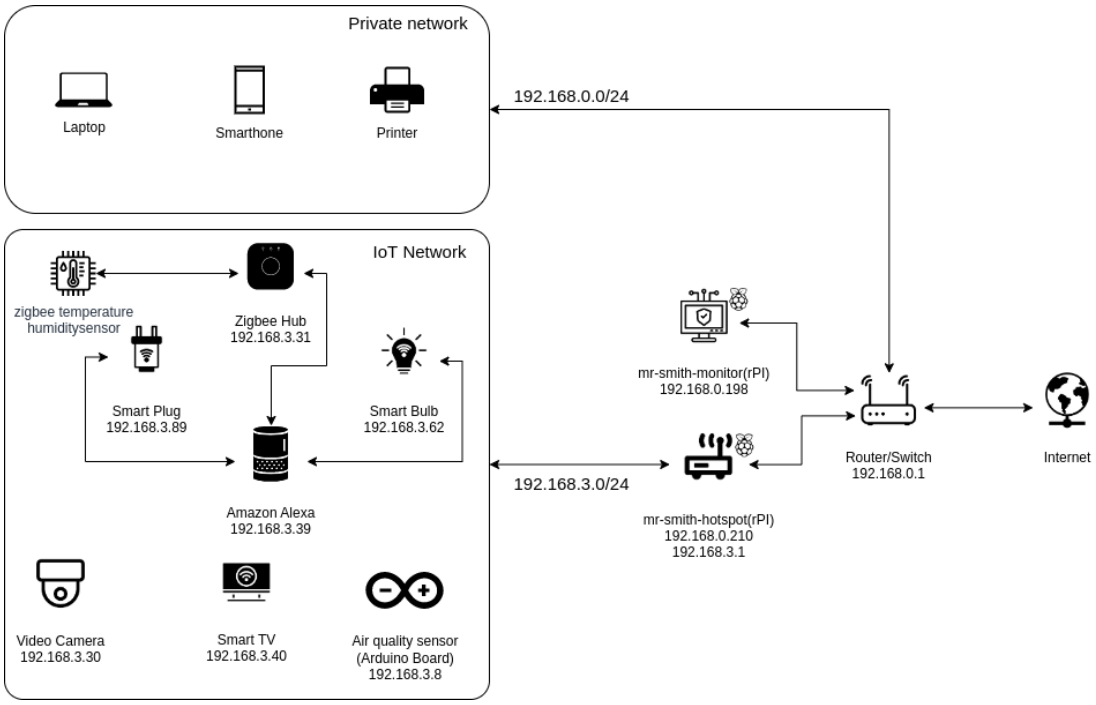


Figura 10 - Cenário teste para *smart home*

Para esta implementação, foram utilizados os seguintes equipamentos:

- Raspberry Pi4 4GB, Gigabit Ethernet, Quad core Cortex-A72 a 1.5GHz;
- Raspberry Pi 3B+ 1GB, 300 Mbps Ethernet, Quad core Cortex-A53 a 1.4GHz;
- Amazon Smart Plug;
- SONOFF Zigbee 3.0 Gateway Hub;
- Sensor de Temperatura/Humidade ZigBee;
- TP-Link Smart Bulb;
- Amazon Echo Dot 4;
- Xiaomi Camera Mi 360;
- Sony Smart TV;
- Router TP-Link AC1750 (Open-WRT);
- Arduino MKR WI-FI 1010 + Sensor de qualidade do ar.

Este conjunto de dispositivos e equipamentos permitiu criar um ambiente de teste representativo e realista, que permite a subsequente avaliação da solução proposta em termos de eficácia, eficiência e segurança. Paralelamente, a diversidade de dispositivos e tecnologias envolvidas no cenário contribui para que a solução seja robusta e adaptável a diferentes configurações de *smart homes*, ampliando a sua aplicabilidade e relevância no contexto da segurança das casas inteligentes.

5.2. Módulo *Hotspot*

Nesta secção, abordaremos a implementação das ferramentas utilizadas no módulo hotspot (IPS *Fail2ban, PI-Hole, IPtables*) com o objetivo de melhorar a segurança dos dispositivos IoT numa *smart home*. Descreveremos os passos práticos tomados, bem como as dificuldades encontradas e como foram superadas, além de explicar como as várias ferramentas se interligam.

O módulo *Hotspot* tem como base um minicomputador *Raspberry Pi,* equipado com uma antena *Wi-Fi* para garantir um maior alcance que cubra toda a habitação. Para colmatar uma das vulnerabilidades mais comuns na instalação destes dispositivos, foram alteradas as configurações por omissão, exemplo da configuração do serviço SSH na Figura 11, uma vez que, os mesmos são configurados com *passwords* por omissão e portas standard abertas, associadas a protocolos comuns.

```
Port 2244

PermitRootLogin prohibit-password
MaxAuthTries 3
MaxSessions 2
PubkeyAuthentication yes
PasswordAuthentication no
PermitEmptyPasswords no
ChallengeResponseAuthentication no
```

Figura 11 - Ficheiro *sshd_config*

Para além da alteração do ficheiro acima descrito e da configuração do *hotspot* a partir do serviço *hostapd*, algumas configurações adicionais foram executadas, que incluem:

- A implementação de atualizações automáticas do sistema, de forma a manter o mesmo constantemente atualizado, a partir da configuração da ferramenta *Unattended Upgrades*;
- A atualização do servidor NTP (Aanchal Malhotra, 2019)

- A realização de uma auditoria de segurança ao sistema, através da ferramenta *lynis*;
- A instalação de uma ferramenta para a deteção de *rootkits* – *rkhunter*;

5.2.1. IPS *Fail2Ban*

O *Fail2Ban* é uma ferramenta que possibilita a configuração de regras de proteção para diversos serviços disponíveis no *host*, como SSH, FTP ou w*eb server*. Esta ferramenta, monitoriza esses serviços, detetando atividades suspeitas como ataques de força bruta. Quando um ataque é detetado, o sistema bloqueia o endereço IP de origem, dificultando assim o acesso não autorizado.

Na implementação do *Fail2Ban*, definiram-se regras personalizadas com o propósito de proteger os serviços críticos do *hotspot*. O *log* do *Fail2Ban* foi ajustado de modo a documentar tentativas de intrusão e a enviar um alerta para o telemóvel do utilizador. A referida notificação é exibida na Figura 12, sendo ativada sempre que um potencial perigo é detetado.

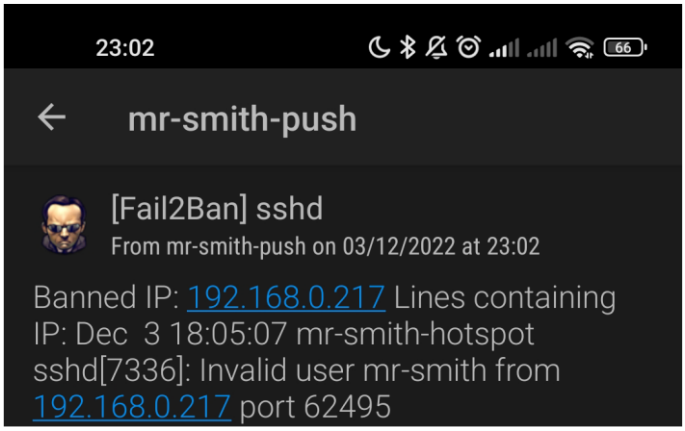


Figura 12 - Notificação telemóvel com alerta do *fail2ban*

A referida figura ilustra o alerta no telemóvel do utilizador quando um ataque é identificado pelo *Fail2Ban*. Este alerta fornece informações relevantes, como o endereço IP bloqueado, o serviço que estava a ser visado e o momento exato do incidente.

Na Figura 13, podemos observar o arquivo *jail.local* personalizado. Este ficheiro contém as configurações específicas para proteger os serviços do *hotspot*, como SSH e o *web server*. Cada seção do ficheiro *jail.local* define as regras para um serviço

específico, incluindo a porta usada, o filtro aplicado e o *logpath* onde o *Fail2Ban* deve monitorar as atividades.

```
[sshd]
# To use more aggressive sshd modes set filter parameter "mode" in
jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage
example and details.
#mode   = normal
port    = ssh
logpath = %(sshd_log)s
backend = %(sshd_backend)s
enabled = true
filter = sshd
banaction = iptables-multiport
action = pushover
bantime = -1
maxretry = 3
```

Figura 13 - Exemplo configuração *fail2ban*

Ao personalizar o arquivo *jail.local* com base nas necessidades específicas do *hotspot*, o *Fail2Ban* oferece uma camada adicional de segurança para a rede, protegendo os dispositivos IoT e outros serviços críticos de atividades maliciosas.

5.2.2. *Pi-Hole*

O *Pi-Hole* é uma ferramenta que permite a criação de listas negras para bloqueio de endereços indesejados, oferecendo maior controlo sobre o conteúdo disponível na rede. O *dashboard* do *Pi-Hole,* ilustrado na Figura 14, foi configurado para exibir informações sobre os pedidos realizados pelos dispositivos conectados ao *hotspot*, incluindo o s*mart gateway Amazon Echo Dot*.

Foi também instalado o servidor DNS recursivo "*Unbound*" para evitar a dependência de servidores DNS de terceiros, aumentando a segurança e prevenindo ataques como o DNS *cache poisoning*. O *Unbound* oferece suporte a funcionalidades como DoT (DNS *over* TLS), DoH (DNS *over* HTTPS) e DNSSEC, aumentando a sua eficácia e eficiência.

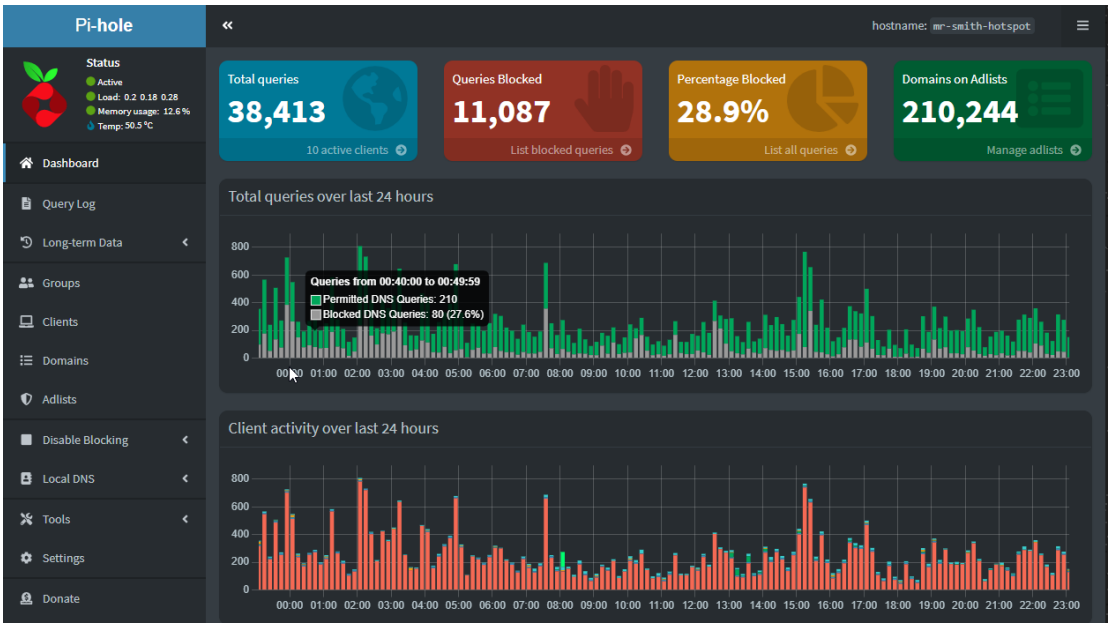


Figura 14 – *Dashboard Pi-Hole*

Após a instalação, foram importadas listas de sites de rastreamento conhecidos a partir do site *https://firebog.net/* para criar uma lista negra personalizada. O *Pi-Hole* atua intercetando os pedidos de DNS e comparando os domínios solicitados com uma série de listas, permitindo ou bloqueando o acesso com base nessas comparações. Estas listas, denominadas de listas negras (*blacklists*) e listas brancas (*whitelists*), são fundamentais para o funcionamento do *Pi-Hole*, ajudando a personalizar a experiência de navegação na Internet, protegendo a privacidade do utilizador e melhorando a segurança da rede.

As listas personalizadas para *o Pi-Hole* são conjuntos de domínios ou endereços IP que o utilizador deseja bloquear ou permitir na sua rede. Porém, elas têm vantagens e desvantagens que devem ser consideradas.

Vantagens das listas personalizadas para o *Pi-Hole*:

- **Controle granular:** As listas personalizadas permitem ao utilizador ter controlo granular sobre o que é permitido e bloqueado na sua rede, incluindo a possibilidade de bloquear anúncios, rastreadores, sites maliciosos ou conteúdos indesejados.
- **Melhoria na privacidade:** Ao bloquear rastreadores e anúncios, as listas personalizadas ajudam a proteger a privacidade do utilizador, impedindo que terceiros rastreiem suas atividades online.

- **Aumento da segurança:** As listas personalizadas podem ser usadas para bloquear *sites* maliciosos ou com histórico de atividades suspeitas, reduzindo a exposição a *malware*, *phishing* e outras ameaças online.
- **Redução do uso de largura de banda:** O bloqueio de anúncios e conteúdo indesejado pelas listas personalizadas pode contribuir para a redução do uso de largura de banda, melhorando a velocidade de navegação e possivelmente reduzindo os custos com a conexão à Internet.
- **Personalização:** As listas personalizadas podem ser adaptadas às necessidades específicas de cada utilizador, permitindo que sejam criadas listas de permissões ou restrições para certos sites ou serviços.

Desvantagens das listas personalizadas para o *Pi-Hole*:

- **Falsos positivos:** O bloqueio excessivo pode originar falsos positivos, em que sites legítimos e úteis são bloqueados acidentalmente, exigindo ajustes frequentes na lista.
- **Manutenção:** As listas personalizadas requerem manutenção contínua, dado que novos domínios e endereços IP podem ser adicionados ou removidos frequentemente. Paralelemente, poderá ser necessário verificar regularmente a atualização e o correto funcionamento das listas.
- **Complexidade:** A criação e a gestão de listas personalizadas podem ser complexas para utilizadores menos experientes, podendo levar a erros de configuração que afetam a funcionalidade do *Pi-Hole* ou a experiência de navegação na Internet.
- **Incompatibilidades:** Algumas listas personalizadas podem ser incompatíveis com determinados dispositivos, navegadores ou aplicativos, levando a problemas de funcionamento ou exibição incorreta de conteúdo.
- **Dependência de listas de terceiros:** Frequentemente, as listas personalizadas são baseadas em listas criadas por terceiros. A qualidade e a eficácia dessas listas podem variar, e os utilizadores podem estar dependentes de atualizações e suporte fornecidos pelos criadores das listas

Ao ponderar as vantagens e desvantagens, é importante que os utilizadores avaliem as suas necessidades e capacidades antes de implementar listas personalizadas no *Pi-Hole*. Para aqueles dispostos a investir tempo e esforço na manutenção e ajuste das listas, os benefícios podem superar as desvantagens. No entanto, para utilizadores menos experientes ou que não desejam envolver-se na

manutenção das listas, talvez seja mais adequado utilizar listas pré-configuradas fornecidas pelo *Pi-Hole* ou outras soluções de filtragem de conteúdo menos complexas.

No contexto do projeto em questão, além das listas pré-configuradas, optou-se também por utilizar as listas do *site Firebog.net.* por diversos motivos, entre os quais se destacam a variedade de listas categorizadas por nível de risco, a transparência do gestor das listas e a comunidade ativa em torno das mesmas. Na Figura 15, podemos visualizar alguns pedidos bloqueados no cenário de testes após a configuração das novas *blacklists*

Time	Type	Domain	Client	Status	Reply	Action
2022-12-04 23:05:01	A	api.amazonalexa.com	192.168.3.39	OK (answered by localhost#5335)	CNAME (0.8ms)	Blacklist
2022-12-04 23:04:47	A	d3p8zr0ffa9t17.cloudfront.net	192.168.3.39	OK (answered by localhost#5335)	IP (0.8ms)	Blacklist
2022-12-04 23:03:20	A	device-metrics-us-2.amazon.com	192.168.3.39	Blocked (gravity)	IP (0.1ms)	Whitelist
2022-12-04 23:03:18	A	device-metrics-us-2.amazon.com	192.168.3.39	Blocked (gravity)	IP (0.1ms)	Whitelist
2022-12-04 23:03:16	A	device-metrics-us-2.amazon.com	192.168.3.39	Blocked (gravity)	IP (0.1ms)	Whitelist
2022-12-04 23:02:49	A	api.amazonalexa.com	192.168.3.39	OK (answered by localhost#5335)	CNAME (1.4ms)	Blacklist
Time	Type	Domain	Client	Status	Reply	Action

Figura 15 - *Pi-Hole*, pedidos bloqueados

5.2.3. *IPTables*

As vantagens do *iptables,* realçadas no capítulo anterior, evidenciam a importância desta ferramenta na promoção de um ambiente seguro. No decorrer da implementação do projeto destinado à proteção da *smart home*, várias regras foram pesquisadas e aplicadas para evitar atividades maliciosas e garantir a segurança da rede.

Essas regras foram configuradas de acordo com os requisitos específicos do ambiente *da smart home*, considerando os dispositivos ligados, os serviços em execução e o tráfego de rede. Algumas das regras implementadas incluem:

- **Bloqueio de tráfego indesejado:** Filtragem de pacotes com base no protocolo, endereço IP de origem/destino ou porta, permitindo apenas conexões necessárias e legítimas.
- **Prevenção de ataques DoS/DDoS:** Limitação da taxa de pacotes para serviços específicos, protegendo o sistema contra-ataques de negação de serviço que tentam sobrecarregar os recursos.

- **Registo e monitorização:** Configuração do *iptables* para registar eventos específicos, como tentativas de conexão a portas fechadas ou bloqueadas, auxiliando na deteção e análise de atividades suspeitas.
- **Proteção contra *spoofing*:** Verificação da integridade dos pacotes para garantir que não sejam originados de endereços IP falsificados, protegendo o sistema contra-ataques de falsificação de identidade.
- **Restrição de acesso aos serviços:** Limitação do acesso aos serviços do sistema apenas para endereços IP confiáveis, evitando acesso não autorizado e possíveis vulnerabilidades de segurança.

O Anexo A apresenta uma lista completa das regras utilizadas na solução proposta.

5.2.4. Visão Geral - *Hotspot*

As ferramentas mencionadas acima articulam-se para estabelecer um *hotspot* seguro para todos os dispositivos IoT de uma *smart home*. O módulo *hotspot*, implementado no *Raspberry Pi*, utiliza o serviço *Unattended Upgrades* e NTP para manter o sistema atualizado e sincronizado. O *Fail2Ban* protege os serviços, como o SSH, enquanto o *PI-Hole* e o *Unbound* trabalham juntos para bloquear endereços indesejados e fornecer um servidor DNS seguro e privado. Por fim, o *iptables* é utilizado para implementar políticas de segurança na rede. Estas ferramentas, ao operarem em conjunto, contribuem para o incremento da segurança de uma rede doméstica e para a proteção dos dispositivos IoT.

5.3. Módulo Monitor

O principal objetivo do segundo módulo do projeto desenvolvido reside na implementação de um sistema de deteção de intrusões (IDS), o *Suricata*, e na sua integração com a *stack* ELK. Esta solução opera a partir de *containers Docker*, tal como ilustrado na Figura 16.

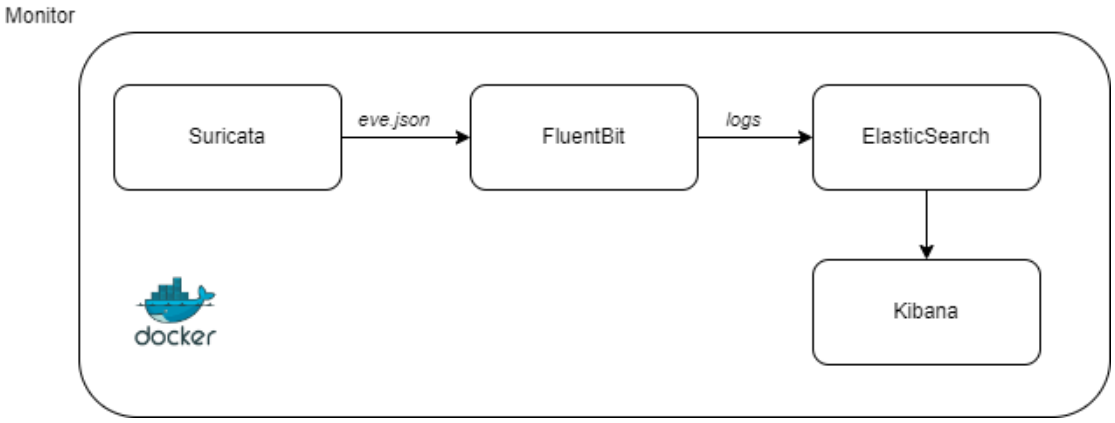


Figura 16 - Arquitetura módulo "Monitor"

A escolha do *Suricata* como IDS, como já mencionado no capítulo anterior, deve-se à sua eficiência e capacidade para analisar o tráfego de rede em tempo real, identificando potenciais ameaças e gerando alertas correspondentes. A integração dos *logs* gerados pelo *Suricata* com a *stack* ELK possibilita uma análise mais profundada dos dados e uma melhor compreensão dos eventos de segurança que ocorrem na rede.

O *FluentBit*, componente alternativo ao *Logstash* da *stack ELK*, é responsável por coletar, processar e armazenar os *logs* gerados pelo *Suricata* no *Elasticsearch*, um motor de busca e análise de dados escalável e distribuído. Por sua vez, o *Kibana*, que é a interface gráfica de utilizador da *stack*, permite a criação de visualizações e painéis de controlo personalizados, auxiliando o utilizador na interpretação e monitorização dos alertas gerados pelo *Suricata* de forma mais eficaz e intuitiva.

Este módulo é fundamental para atingir os objetivos propostos no projeto, pois permite ao utilizador ter uma visão abrangente e detalhada da segurança da sua rede, identificando potenciais vulnerabilidades e ameaças e, consequentemente, tomar decisões informadas para melhorar a proteção do ambiente. Adicionalmente, a utilização de *containers Docker* facilita a implementação, manutenção e escalabilidade da solução, garantindo um maior grau de isolamento e eficiência no seu funcionamento.

5.3.1. *Containers Docker*

A implementação do sistema *IDS* no segundo módulo do projeto foi realizada de forma a garantir a eficiência na deteção de anomalias na rede doméstica. A solução foi desenvolvida em *containers Docker* num *Raspberry Pi*, o que permitiu obter

vantagens em termos de isolamento, flexibilidade e escalabilidade que são apresentadas a seguir:

- **Isolamento:** Os *containers Docker* proporcionam um ambiente isolado para executar aplicações, garantindo que os processos e recursos sejam separados entre si. Esta abordagem evita conflitos de dependências e configurações, permitindo que cada aplicação funcione de forma independente e sem interferir nas demais. No caso do *Raspberry Pi*, isso significa que é possível executar várias aplicações em paralelo, sem que uma afete o desempenho ou a estabilidade das restantes.
- **Flexibilidade:** Os *containers* são portáteis e podem ser facilmente movidos e executados em diferentes sistemas operacionais e arquiteturas de *hardware*. Adicionalmente, os *containers* facilitam a atualização e a manutenção das aplicações, uma vez que as alterações podem ser feitas nos ficheiros de configuração e nas imagens dos *containers*, sem a necessidade de reconfigurar todo o sistema.
- **Escalabilidade:** A utilização de *containers Docker* permite a escalabilidade das aplicações, uma vez que é possível criar e gerir múltiplas instâncias de um *container* para lidar com diferentes cargas de trabalho. No caso do *Raspberry Pi*, embora existam limitações de recursos, a abordagem baseada em *containers* permite uma melhor utilização dos recursos disponíveis e a possibilidade de ajustar a alocação de recursos com base nas necessidades da aplicação. Isto pode ser particularmente útil no contexto de soluções IoT, onde o número de dispositivos e a quantidade de dados podem variar significativamente.

5.3.2. *Suricata*

A configuração do *Suricata* no contexto de uma *smart home* envolve a personalização do ficheiro *"/etc/suricata/suricata.yaml"*, que define as regras e parâmetros específicos para monitorizar e proteger os dispositivos IoT e serviços presentes nesse ambiente.

O *Suricata* foi configurado para analisar o tráfego proveniente de um túnel GRE, que transporta os dados entre o "*hotspot"* e o "monitor", garantindo a monitorização completa do tráfego da rede IoT. O desencapsulamento do tráfego é realizado pelo próprio *Suricata*, não constituindo um obstáculo à análise.

Os *logs* gerados pelo *Suricata* são armazenados no ficheiro *eve.json*, que contém informações detalhadas sobre os alertas e eventos de segurança identificados. Estes *logs* são posteriormente processados e analisados pela *stack ELK*, permitindo a criação de visualizações e *dashboards* personalizados para monitorização dos eventos.

No âmbito deste projeto, optou-se por não utilizar a funcionalidade de prevenção de intrusões (IPS) do *Suricata*, de modo a não comprometer o normal funcionamento da rede doméstica. A implementação de um sistema IPS requer uma análise cuidadosa e a criação de regras específicas para evitar falsos positivos e a interrupção indevida de serviços legítimos. Além disso, um sistema IPS pode apresentar desafios em termos de desempenho e latência, especialmente em dispositivos com recursos limitados, como o *Raspberry Pi*.

Em suma, a implementação e configuração do *Suricata* no segundo módulo do projeto centraram-se na eficiência e precisão na deteção de anomalias na rede doméstica, utilizando *containers Docker* num *Raspberry Pi* e ficheiros de configuração YAML. A escolha de não utilizar a funcionalidade de IPS foi baseada na preservação do funcionamento normal da rede e na minimização de potenciais desafios relacionados ao desempenho e falsos positivos.

5.3.3. ELK + IDS

Na implementação do projeto, foi utilizado o *stack ELK* (*Elasticsearch, Logstash* e *Kibana*) para facilitar a monitorização e análise dos *logs* gerados pelo *Suricata* (IDS), mais especificamente o arquivo *eve.json*. Com o intuito de garantir eficiência e otimização dos recursos no *Raspberry Pi*, optou-se pela substituição do *Logstash* pela ferramenta *FluentBit*, que se releva mais leve e consome menos recursos.

A configuração do *stack ELK* em *containers Docker* num *Raspberry Pi* foi realizada seguindo os seguintes passos:

- **Instalação do *Docker* e *Docker Compose* no *Raspberry Pi*:** Inicialmente, foi necessário instalar o *Docker* e *Docker Compose* no *Raspberry Pi* para administrar os *containers*.
- **Configuração do *Docker Compose*:** Foi criado um arquivo *docker-compose.yml* que descreve os serviços *Suricata, Elasticsearch, Kibana e FluentBit*, além das configurações necessárias, como portas, volumes e variáveis de ambiente.

- **Configuração *Suricata***: Foi criado um *container Docker* para o *Suricata* com um ficheiro de configuração personalizado, integrado no ficheiro *docker-compose.yml*. Garantiu-se o acesso do *Suricata* aos registos de rede do *hotspot* e a acessibilidade dos seus *logs* pelo *FluentBit* para o respetivo processamento.
- **Configuração do *FluentBit*:** O *FluentBit* foi configurado para ler o arquivo *eve.json* gerado pelo *Suricata* e enviar os dados para o *Elasticsearch*. Foi criado um arquivo de configuração "*fluent-bit.conf*", onde foram definidas as entradas (*input*), saídas (*output*) e filtros (*filter*) para o processamento dos *logs* do *Suricata*.
- **Inicialização dos *containers*:** Com o *Docker Compose* e os arquivos de configuração preparados, os *containers* foram iniciados com o comando *docker-compose up -d*, o que garantiu a execução em segundo plano.
- **Configuração do *Kibana*:** Após a inicialização dos *containers*, foi necessário configurar o *Kibana* para criar índices no *Elasticsearch*, baseados nos *logs* processados pelo *FluentBit*. Isso permitiu a criação de visualizações e *dashboards* personalizados para facilitar a análise dos alertas gerados pelo IDS *Suricata*, como pode ser visualizado na Figura 17.

Com esta implementação, foi possível integrar de forma eficiente os *logs* do *Suricata*, mais especificamente o ficheiro *eve.json*, ao *stack* ELK no *Raspberry Pi*. A utilização do *FluentBit* em detrimento do *Logstash* permitiu um melhor desempenho em termos de recursos consumidos, mantendo a funcionalidade desejada e garantindo a eficácia na análise dos dados.

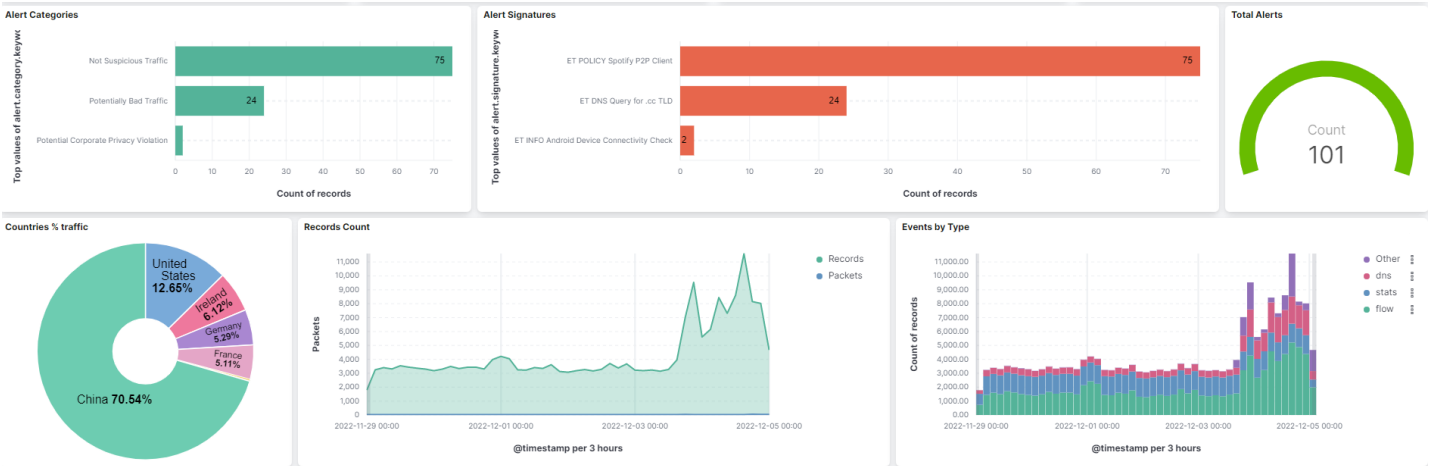


Figura 17 - *Dashboard Kibana* para os alertas do *Suricata*

6. Análise dos resultados

Neste capítulo, procederemos à análise dos resultados obtidos através dos testes realizados com o protótipo implementado, com o objetivo de avaliar a sua eficácia na garantia de segurança e privacidade dos utilizadores em ambientes de *smart home*. Serão apresentados diversos testes realizados e simulações para verificar se os objetivos de segurança estabelecidos inicialmente foram atingidos, e se a solução desenvolvida cumpre os requisitos necessários para assegurar a proteção de dados e comunicações dentro de uma rede doméstica inteligente.

6.1. Limitações *Raspberry Pi*

Um dos elementos fundamentais da solução proposta baseia-se na utilização de dois minicomputadores *Raspberry Pi*, amplamente empregues no desenvolvimento de protótipos em diversas áreas (Dewangan & Sahu, 2021; R. Kumar & Rajasekaran, 2016). Contudo, é crucial avaliar a capacidade real desses dispositivos em ambientes reais e de produção. Para analisar esta questão de forma mais aprofundada, realizaram-se testes de desempenho, focados na taxa de transferência e na latência entre os dispositivos.

Os dispositivos *Raspberry Pi* em questão estão equipados com portas *Gigabit Ethernet*. No entanto, é importante salientar que o RPi 3B+ encontra-se limitado à velocidade do barramento USB *2.0* interno de 300 Mbps. Dada esta limitação, foram realizados testes utilizando a ferramenta *iperf3,* para avaliar o desempenho das comunicações entre os dois dispositivos em várias condições.

Teste 1: Ligação durante 60 segundos, comunicação com base no protocolo TCP e 4 conexões simultâneas ao servidor (*iperf3 -t 60 -P 4 -b 0 -f M -c 192.168.0.210*)

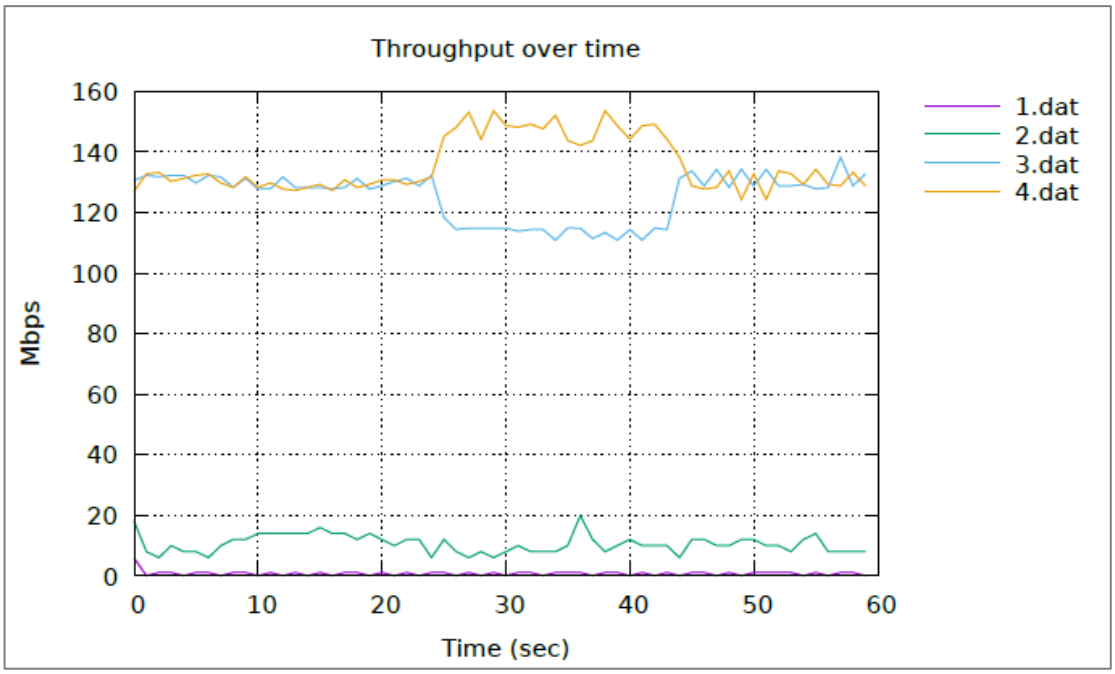


Gráfico 1 - Taxa de transferência - teste 1

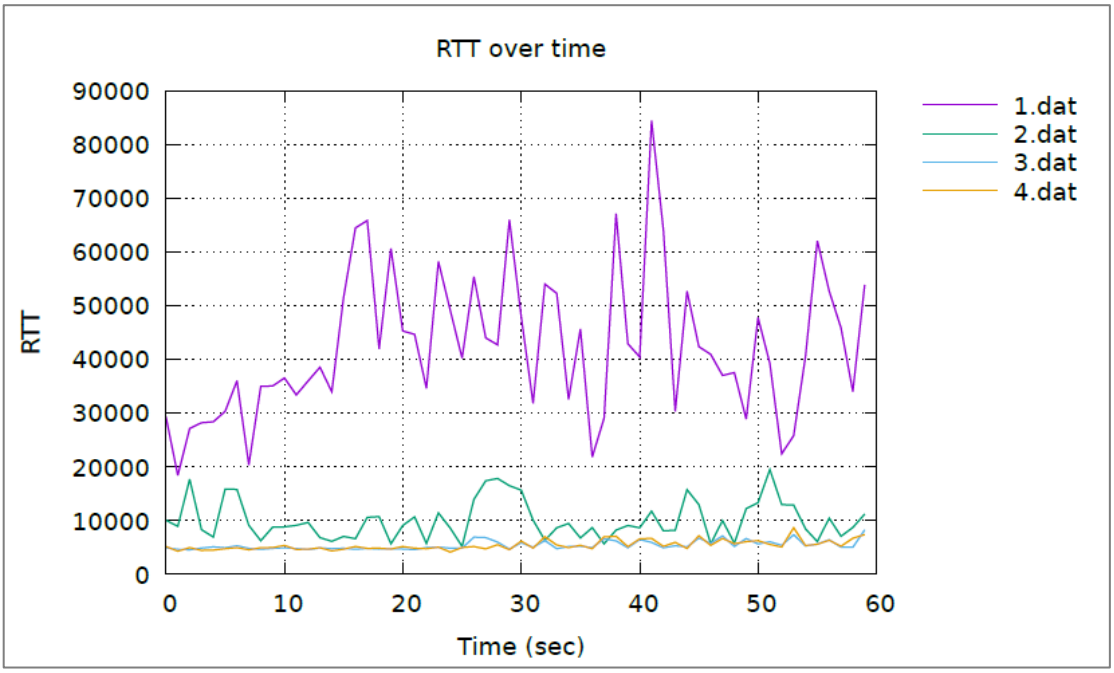


Gráfico 2 - RTT - teste 1

Teste 2: Ligação durante 60 segundos, comunicação com base no protocolo TCP e 1 conexão ao servidor (*iperf3 -t 60 -b 0 -f M -c 192.168.0.210*)

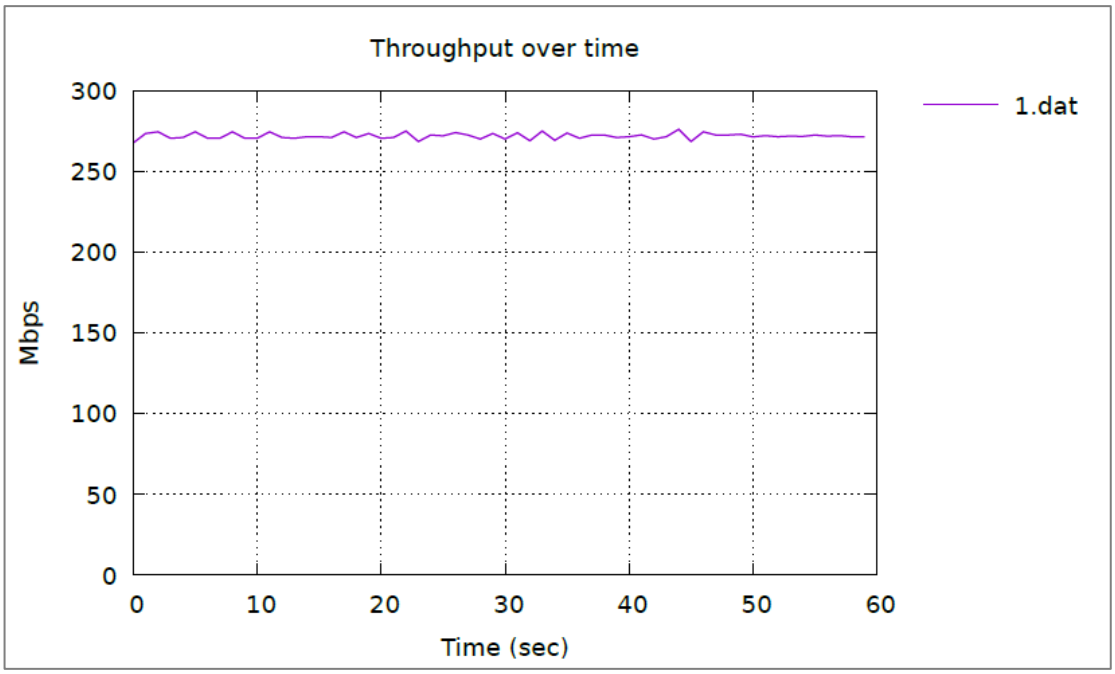


Gráfico 3 – Taxa de transferência RPi – teste 2

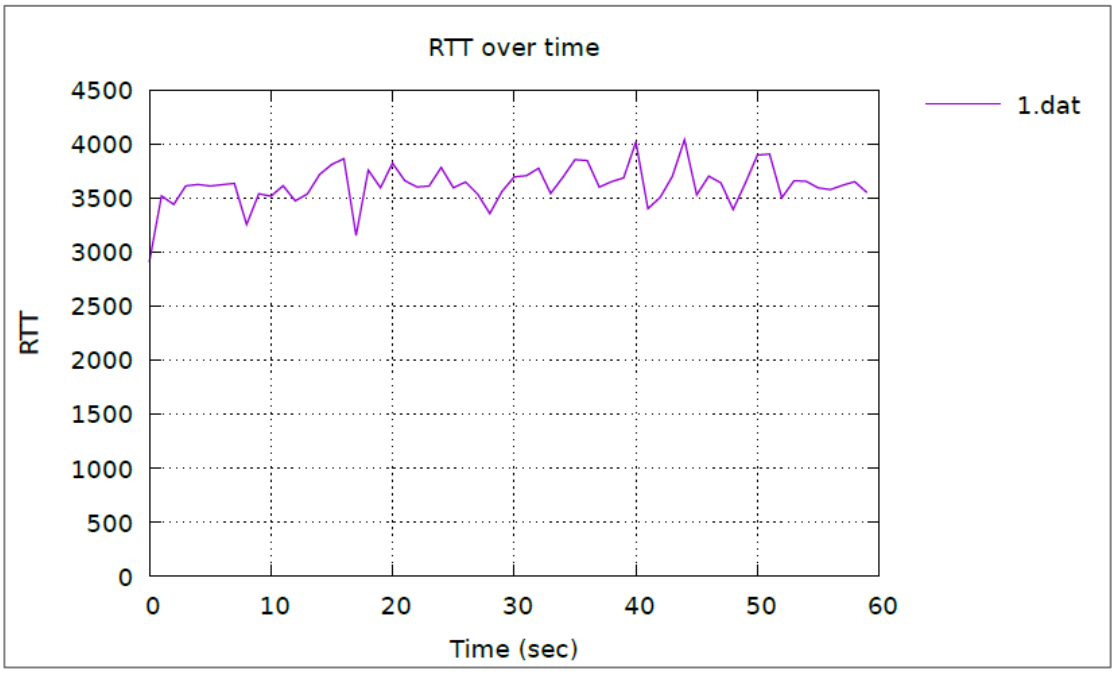


Gráfico 4 – RTT – teste 2

Após a execução dos testes, obteve-se uma taxa de transferência média de 285 *Mbps*, o que se aproxima do limite teórico do barramento USB 2.0 interno do RPi 3B+. Este resultado é um indicativo de que o *Raspberry Pi* é capaz de lidar com a transferência de dados em velocidades próximas ao seu limite máximo teórico.

No que diz respeito à latência, o *Round Trip Time* (RTT) médio alcançado nos testes foi de 3500 microssegundos, ou seja, 3,5 milissegundos. Embora este valor possa parecer elevado, é importante considerar que, no contexto de uma *smart home*, a maioria das operações não requer uma latência extremamente baixa. Mesmo para tarefas mais exigentes, como *streaming* de vídeo, essa latência é suficiente para garantir uma experiência satisfatória ao utilizador.

Considerando os resultados obtidos nos testes realizados, é possível concluir que os dispositivos *Raspberry Pi* conseguem corresponder às expectativas e atender aos requisitos de desempenho em um contexto de *smart home*. A taxa de transferência e a latência alcançadas garantem a execução eficiente das funcionalidades propostas na solução desenvolvida, proporcionando segurança e privacidade aos utilizadores.

6.2. Auditoria rede IoT (*Hotspot*)

Para aprofundar a análise da segurança da rede criada no módulo *hotspot*, foram realizados testes com a ferramenta *Nmap*, permitindo verificar possíveis vulnerabilidades nos dispositivos conectados à rede. O comando utilizado para a análise foi: `sudo nmap –script vuln -Pn -O 192.168.3.0/24.`

Os resultados obtidos são detalhados na tabela abaixo, que apresenta as vulnerabilidades detetadas nos dispositivos analisados:

Tabela 2 - Resultados *scan* nmap

Host	Dispositivo	Portas Identificadas	Vulnerabilidades detetadas
192.168.3.1	Hotspot	53, 80	Slowloris DoS attack (CVE-2007-6750)
192.168.3.8	*Arduino* + sensor	-	-
192.168.3.30	Camara de vídeo	-	-
192.168.3.31	Zigbee Hub	-	-
192.168.3.39	Alexa	1080, 8888	-
192.168.3.62	Smart Bulb	80	-
192.168.3.89	Smart Plug	-	-
192.168.3.40	Smart TV	-	-

Com base nos resultados, é possível observar que apenas o dispositivo *hotspot* apresentou uma vulnerabilidade identificada pelo *Nmap*, o ataque DoS *Slowloris* (CVE-2007-6750). Este ataque tem como objetivo congestionar a ligação ao servidor, realizando vários pedidos HTTP parciais simultâneos.

Apesar da presença desta vulnerabilidade, é importante considerar que os restantes dispositivos não apresentaram vulnerabilidades identificáveis pelo *Nmap*. Isso sugere que a solução proposta oferece um nível adequado de segurança no contexto de uma *smart home.* Contudo, perante a vulnerabilidade detetada no *hotspot*, é imperativo salientar que esta será alvo de uma análise mais aprofundada nos capítulos subsequentes do documento, com o intuito de encontrar estratégias eficazes para colmatar tal falha.

6.3. Teste *Suricata* + ELK

Na fase final do projeto, foram realizados testes de penetração ao sistema para avaliar a eficácia do sistema IDS implementado, bem como a apresentação dos alertas no *stack* ELK.

Um dos principais ataques identificados na literatura a dispositivos IoT e *smart homes* são os ataques DoS. Nesse sentido, foram realizados testes específicos para verificar se a solução proposta impede eficazmente este tipo de ataques e se o sistema IDS emite os alertas correspondentes. Para simular o ataque DoS, utilizou-se um portátil com o sistema *Kali Linux* e a ferramenta *hping3*. Os resultados são apresentados nas imagens seguintes.

Os resultados obtidos evidenciaram um aumento significativo do tráfego durante a simulação do ataque, confirmando a eficácia do sistema IDS na deteção de atividades suspeitas e potencialmente maliciosas. A ferramenta *hping3* permitiu testar a eficácia da *firewall* em diferentes protocolos e detetar pacotes suspeitos ou modificados. Os gráficos apresentados nas Figuras 18 e 19 ilustram o aumento do número de pacotes TCP transmitidos durante o ataque.

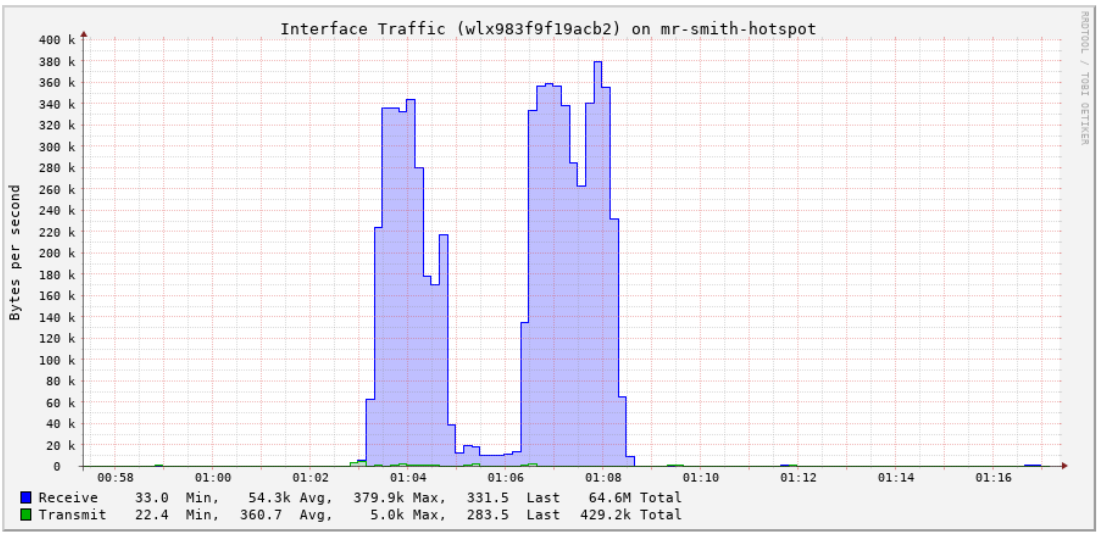


Figura 19 - Simulação ataque *DoS* (*bytes/s*)

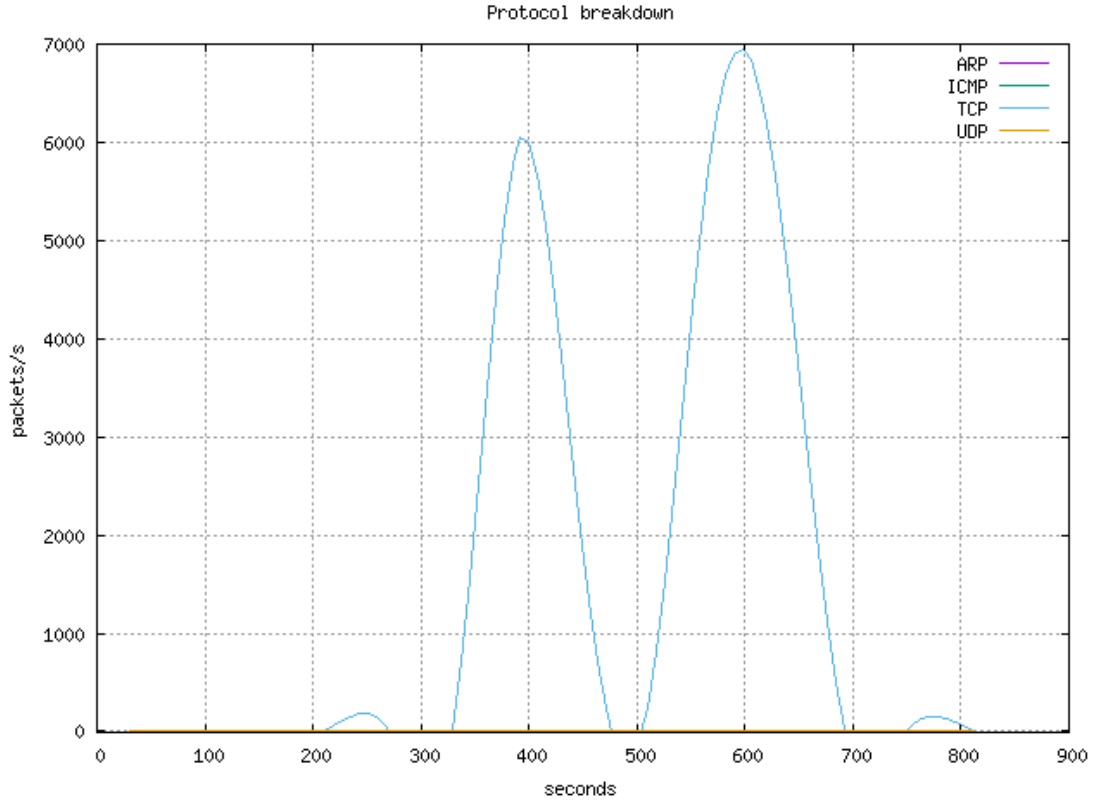


Figura 18 - Simulação ataque DoS - *Packets/s*

No que respeita ao desempenho das regras criadas para o *iptables*, observa-se na Figura 20 o número de pacotes que foram ignorados durante os testes, demonstrando a capacidade do sistema em bloquear o tráfego malicioso.

```
Chain PREROUTING (policy ACCEPT 327K packets, 28M bytes)
 pkts bytes target     prot opt in     out     source               destination
   62  4388 DROP       all  --  any    any     anywhere             anywhere             ctstate INVALID
    6   456 DROP       tcp  --  any    any     anywhere             anywhere             tcp flags:!FIN,SYN,RST,ACK/SYN ctstate NEW
2754K  110M DROP       tcp  --  any    any     anywhere             anywhere             ctstate NEW tcpmss match !536:65535
 9603 1304K DROP       icmp --  any    any     anywhere             anywhere
    0     0 DROP       all  -f  any    any     anywhere             anywhere

Chain INPUT (policy ACCEPT 298K packets, 23M bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 29397 packets, 5279K bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 3070K packets, 378M bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 3099K packets, 383M bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Figura 20 - Estatísticas *Iptables*

Relativamente aos alertas gerados pelo sistema IDS, as Figuras 21 e 22 mostram a efetiva deteção e apresentação dos alertas no *stack* ELK.

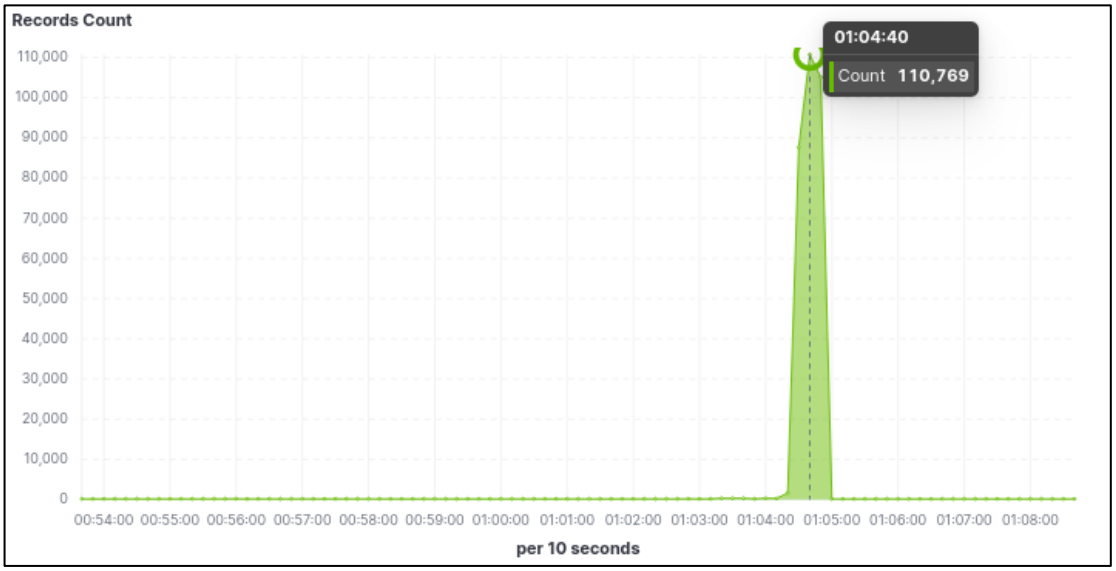


Figura 21 - Número de alertas do IDS

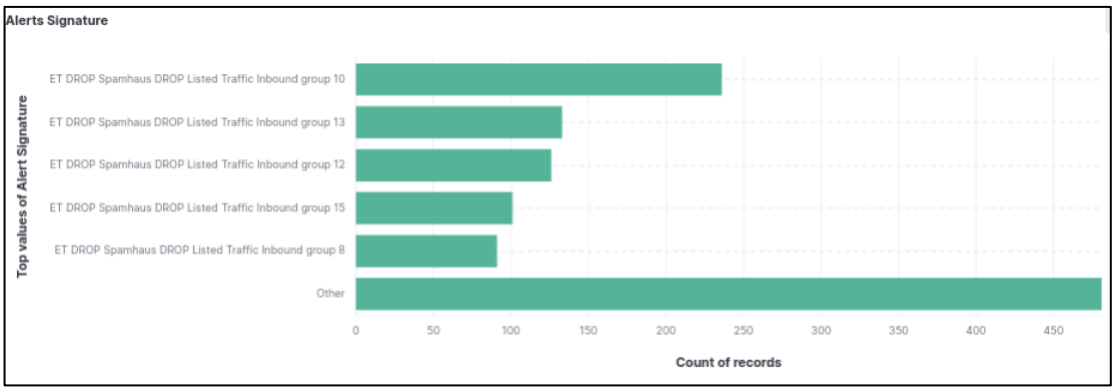


Figura 22 - Alertas *Suricata*

A vulnerabilidade DoS *Slowloris*, identificada pelo *Nmap* no dispositivo responsável pelo *hotspot*, foi submetida a um teste específico utilizando a *framework Metasploit*. O objetivo deste teste era verificar a presença da anomalia e avaliar a resposta do sistema IDS perante uma tentativa de exploração desta vulnerabilidade. Os comandos utilizados foram os seguintes:

```
$: msfconsole
$: use auxiliary/dos/http/slowloris
$: set RHOST 192.168.3.1
$: exploit
```

Durante a execução do teste, constatou-se que a página de administração do *Pi-Hole* ficou indisponível, corroborando a eficácia do ataque *Slowloris*. Paralelamente, não foi gerado nenhum alerta no sistema IDS, sugerindo uma lacuna na deteção deste tipo específico de ataque.

A Figura 23 apresenta o *dashboard* do sistema IDS após a realização dos testes, onde não é possível identificar qualquer alerta relacionado ao ataque *Slowloris*. Esta situação demonstra a importância de realizar testes abrangentes e específicos durante a avaliação de soluções de segurança para *smart homes*, visando a identificação e retificação de possíveis lacunas na deteção de ameaças.

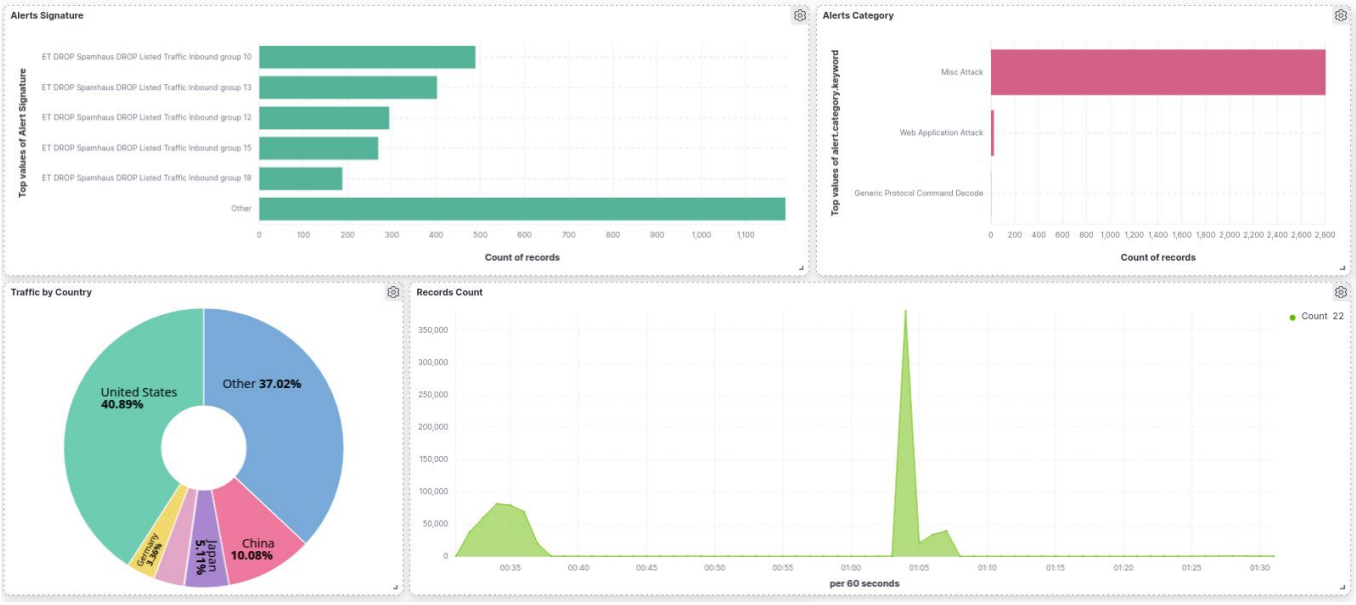


Figura 23 - *Dashboard Kibana* após simulação ataques

A análise dos resultados obtidos nos testes de penetração evidencia que a solução proposta alcança parcialmente os objetivos estabelecidos no projeto, apesar de não ter detetado um tipo de ataques, impediu por exemplo ataques DoS,

contribuindo, apesar de tudo, para uma *smart home* mais segura. A implementação do sistema IDS e a integração com o *stack* ELK permitiram identificar e bloquear ataques, assim como emitir alertas para possíveis tentativas de intrusão. Estas características são fundamentais para garantir a proteção e privacidade dos utilizadores de uma *smart home*, fortalecendo a segurança da rede e dos dispositivos conectados.

Em conclusão, a análise dos resultados obtidos nos testes de penetração e a identificação das vulnerabilidades existentes reforçam a necessidade de uma avaliação contínua e aprimoramento da solução de segurança proposta, a fim de garantir a proteção efetiva dos dispositivos e da rede em um ambiente de *smart home*.

6.4. Mitigação do Ataque *Slowloris*

A identificação e mitigação de um ataque *Slowloris* tornam-se imperativas após os testes realizados nas secções anteriores, com o intuito de colmatar uma falha de segurança significativa no protótipo por nós desenvolvido. Esta vulnerabilidade põe em causa a segurança das *smart homes*, colocando em risco a confidencialidade e a integridade dos seus dados. Assim, a necessidade de aprofundar o nosso conhecimento acerca do ataque *Slowloris* e as formas de o contrariar tornou-se premente. O propósito desta secção é, por conseguinte, explorar e apresentar estratégias eficazes de mitigação deste tipo de ataque, visando o fortalecimento da segurança do nosso protótipo, garantindo a sua resistência e robustez face a esta ameaça.

O *Slowloris* é um ataque de negação de serviço (DoS), caracterizado pela abertura e manutenção de múltiplas conexões HTTP simultâneas entre o atacante e o alvo, sobrecarregando o servidor e tornando-o inoperante. A peculiaridade do *Slowloris* reside na sua forma de operar ao nível da camada de aplicação, mais especificamente, através da exploração de requisições HTTP parciais.

Em contraste com os ataques DDoS mais comuns, que consomem largura de banda, como a amplificação NTP, o *Slowloris* distingue-se pela sua estratégia "lenta e discreta". Esta abordagem tem como objetivo esgotar os recursos do servidor com requisições que parecem ser mais lentas do que o normal, mas que, de outro modo, imitam o tráfego regular. Cada *thread* do servidor tenta manter-se ativa enquanto aguarda a conclusão da requisição lenta, que nunca acontece. Quando o número máximo de conexões possíveis do servidor é ultrapassado, cada conexão adicional não será respondida, resultando numa negação de serviço.

Perante este cenário, torna-se fundamental implementar medidas que possam prevenir ou mitigar os efeitos de um ataque *Slowloris*. Algumas soluções possíveis incluem:

- **Limitação do número de ligações por IP**: A definição de limites no servidor para o número de conexões simultâneas que um único endereço IP pode estabelecer pode contribuir para a mitigação do impacto de um ataque *Slowloris*. Esta medida pode ser implementada através da configuração no próprio servidor web ou pelo uso de um Sistema de Prevenção de Intrusões (IPS) ou de uma *firewall*.
- **Redução do tempo "*keep-alive*"**: Diminuir o tempo durante o qual uma conexão HTTP pode permanecer aberta pode fazer com que as conexões sejam encerradas mais rapidamente, limitando assim o impacto de um ataque *Slowloris*.
- **Balanceamento de carga**: O uso de um balanceador de carga pode ajudar a distribuir o tráfego entre vários servidores, o que pode contribuir para mitigar o impacto de um ataque *Slowloris*.
- **Módulos de segurança específicos**: Alguns servidores web, como o *Apache*, têm módulos específicos que podem ser instalados para ajudar a mitigar ataques *Slowloris*. O módulo *mod_reqtimeout* do *Apache*, por exemplo, pode ser configurado para detetar e bloquear ataques *Slowloris*.
- **Utilização de um serviço de mitigação de DDoS**: Serviços como o *Cloudflare* oferecem proteção contra vários tipos de ataques DDoS, incluindo *Slowloris*.
- **Uso de "*reverse proxy*"**: Configurar um *reverse proxy*, tal como o *Nginx* ou *HAProxy*, pode ajudar a mitigar ataques *Slowloris*. Este pode limitar o número de conexões lentas que chegam ao servidor, ajudando a manter o serviço disponível.

Com base nas medidas acima referidas, e tendo em conta o contexto da nossa solução para proteger as *smart homes*, tomámos as seguintes ações.

Em relação à *firewall*, adicionámos regras específicas ao *iptables* de modo a mitigar este ataque.

```
iptables -I INPUT -p tcp --dport 80 -m connlimit --connlimit-above 20 --
connlimit-mask 32 -j DROP

iptables -A INPUT -p tcp --dport 80 -m state --state NEW -m recent --set
iptables -A INPUT -p tcp --dport 80 -m state --state NEW -m recent --
update --seconds 60 --hitcount 20 -j DROP
```

O *webserver* utilizado pelo *PI-Hole* foi modificado para apenas permitir um número fixo de ligações

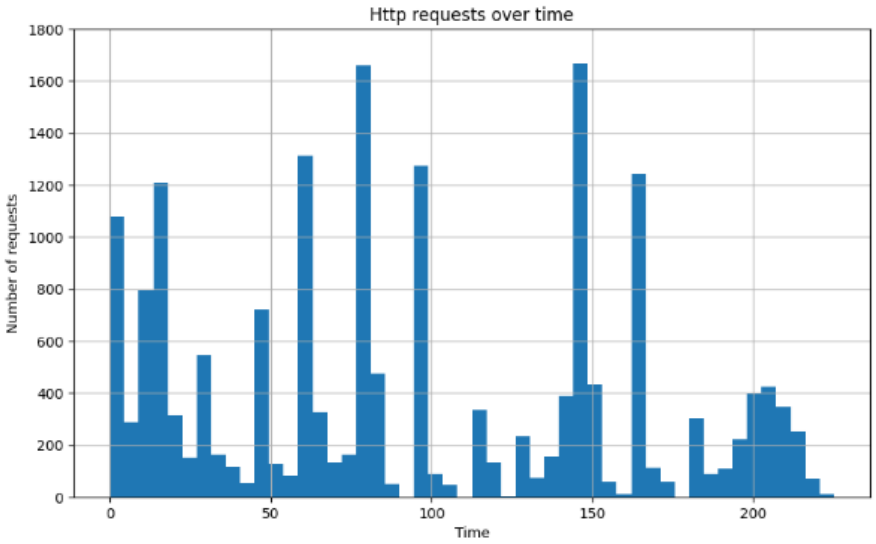
```
server.max-read-idle = 10 # maximum number of seconds until a waiting,
non keep-alive read times out and closes the connection

server.max-write-idle = 10 # maximum number of seconds until a waiting
write call times out and closes the connection
```

Após as alterações efetuadas, lançámos uma nova simulação de um ataque *Slowloris*. Notámos que houve uma melhoria na performance da página que anteriormente tinha ficado indisponível. Nas Figuras 24 e 25, pode-se constatar que grande parte do tráfego gerado pela simulação do ataque foi rejeitado.

```
smith@mr-smith-hotspot:~/dumps $ sudo iptables -L INPUT -v
Chain INPUT (policy ACCEPT 52343 packets, 7509K bytes)
 pkts bytes target     prot opt in     out     source               destination
  259 29091 DROP       tcp  --  any    any     anywhere             anywhere             tcp dpt:http #conn src/32 > 20
    0     0 f2b-sshd   tcp  --  any    any     anywhere             anywhere             multiport dports ssh
    0     0 f2b-sshd   tcp  --  any    any     anywhere             anywhere             multiport dports ssh
    0     0 f2b-sshd   tcp  --  any    any     anywhere             anywhere             multiport dports ssh
    0     0 f2b-sshd   tcp  --  any    any     anywhere             anywhere             multiport dports ssh
    0     0 f2b-sshd   tcp  --  any    any     anywhere             anywhere             multiport dports ssh
    0     0 f2b-sshd   tcp  --  any    any     anywhere             anywhere             multiport dports ssh
    0     0 f2b-sshd   tcp  --  any    any     anywhere             anywhere             multiport dports ssh
    0     0 f2b-sshd   tcp  --  any    any     anywhere             anywhere             multiport dports ssh
 2516  148K REJECT     tcp  --  any    any     anywhere             anywhere             #conn src/32 > 111 reject-with tcp-reset
 1653  101K ACCEPT     tcp  --  any    any     anywhere             anywhere             ctstate NEW limit: avg 60/sec burst 20
```

Figura 24 - Estatísticas *iptables* após simulação *slowloris*



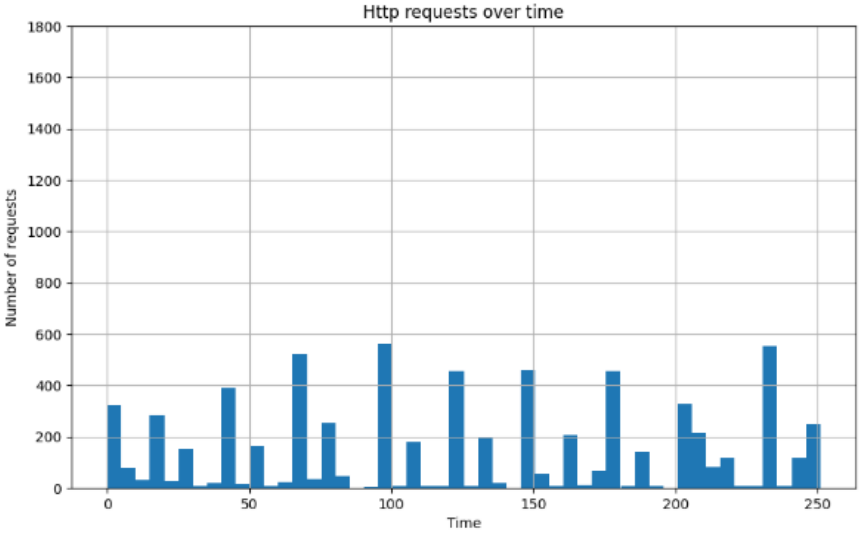


Figura 25 - Pedidos HTTP antes e depois de aplicar as novas regras no *iptables*

Contudo, no que diz respeito ao IDS, não houve qualquer alerta desencadeado, apenas uma visualização do aumento do tráfego que pode ser considerada anómala, conforme ilustrado na Figura 26. Embora o *Suricata* não seja capaz de detetar automaticamente um ataque *Slowloris*, é possível estabelecer regras personalizadas para identificar indícios deste tipo de ataque. Por exemplo, pode-se configurar o *Suricata* para emitir um alerta sempre que um único endereço IP estabeleça um número excessivo de conexões HTTP, ou quando uma ligação HTTP se mantém ativa por um período anormalmente longo. Além disso, é possível monitorizar a quantidade

total de ligações abertas no servidor para detetar um aumento repentino, que pode indicar um ataque em progresso.

Importa frisar que estas regras podem acarretar um número significativo de falsos positivos, dado que também podem ser ativadas por tráfego legítimo. Por exemplo, um utilizador com uma ligação à internet lenta ou instável pode necessitar de estabelecer múltiplas ligações para descarregar um ficheiro de grandes dimensões, ou manter as suas ligações ativas durante um longo período devido à baixa velocidade de descarga.

Assim, a deteção de ataques *Slowloris* pode requerer uma combinação de regras personalizadas, análise de tráfego em tempo real e, talvez mais crucialmente, um conhecimento profundo do padrão de tráfego normal na rede que está a ser monitorizada.

Figura 26 - *Dashboard* após ataque *slowloris*

7. Conclusão

No presente trabalho, abordou-se a problemática inerente à segurança e à privacidade em contextos de *smart homes*, dado o aumento exponencial do número de dispositivos IoT e a sua respetiva utilização no quotidiano doméstico. Através da revisão da literatura, identificaram-se as ameaças e vulnerabilidades principais associadas a este cenário, assim como as tecnologias e estratégias mais adequadas para as contrariar. Assim, foi proposta uma solução de segurança baseada na utilização de tecnologias como o *Raspberry Pi*, o *Suricata*, o *iptables* e o *stack* ELK, visando a implementação de um sistema IDS e a criação de uma infraestrutura de rede segura para dispositivos IoT.

A análise da literatura foi crucial para a identificação dos objetivos principais do projeto, orientando a seleção das tecnologias e a construção da solução. A utilização do *Raspberry Pi* como base para o sistema IDS e o *hotspot* da rede mostrou ser uma opção eficaz e eficiente em termos de recursos, apesar de algumas limitações relativas à capacidade de processamento e velocidade de ligação. A implementação de *containers Docker* para a instalação dos módulos *ElasticSearch* e *Kibana* simplificou o processo e otimizou a utilização dos recursos do *Raspberry Pi.*

A implementação do sistema IDS com recurso ao *Suricata*, e a configuração de regras específicas para o contexto das *smart homes*, possibilitaram a deteção de várias ameaças e a geração de alertas no stack ELK, facilitando a análise e a tomada de ações de mitigação. A substituição do *Logstash* pela ferramenta *FluentBit* veio tornar a solução mais eficiente em termos de recursos consumidos. No entanto, após alguns testes de penetração, como o ataque *Slowloris*, foram identificadas oportunidades de melhoria e necessidade de ajustes na solução proposta.

A solução foi revista e novos testes foram efetuados após a aplicação de regras adicionais de mitigação, resultando num aprimoramento da proteção oferecida. Apesar disso, tornou-se evidente a importância de uma avaliação contínua e adaptação à medida que novas ameaças e vulnerabilidades vão surgindo.

Como trabalho futuro, propõe-se a exploração de mecanismos adicionais de deteção e mitigação, como a implementação de soluções de *Machine Learning* para análise de tráfego e deteção de anomalias. A avaliação de outras tecnologias e abordagens complementares, como o uso de *honeypots* e o reforço da autenticação e autorização em dispositivos IoT, poderá fortalecer a segurança da solução proposta.

Uma outra linha de investigação para trabalhos futuros pode ser a integração da solução proposta com outras ferramentas e sistemas de gestão de segurança, como os Sistemas de Informação e Eventos de Segurança (SIEM), permitindo uma abordagem mais holística e eficiente na proteção de ambientes de *smart homes*. A colaboração entre diferentes sistemas e a partilha de informações sobre ameaças e vulnerabilidades pode ajudar a construir uma rede de proteção mais robusta.

Em conclusão, a constante evolução do cenário tecnológico torna impossível a criação de uma solução de segurança infalível. No entanto, é crucial desenvolver soluções acessíveis aos utilizadores e suficientemente dinâmicas para se adaptarem às mudanças. Esta pesquisa sublinha a importância de se manter uma abordagem proactiva na criação de soluções de segurança resilientes e adaptáveis, que garantam a proteção contínua dos ambientes de *smart homes*.

Referências Bibliográficas

Aanchal Malhotra. (2019, junho 21). *Introducing time.cloudflare.com*. https://blog.cloudflare.com/secure-time/

Abu Waraga, O., Bettayeb, M., Nasir, Q., & Abu Talib, M. (2020). Design and implementation of automated IoT security testbed. *Computers & Security*, *88*, 101648. https://doi.org/10.1016/j.cose.2019.101648

Acar, A., Fereidooni, H., Abera, T., Sikder, A. K., Miettinen, M., Aksu, H., Conti, M., Sadeghi, A.-R., & Uluagac, S. (2020). Peek-a-boo: I see your smart home activities, even encrypted! *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 207–218. https://doi.org/10.1145/3395351.3399421

Ahanger, T. A., & Aljumah, A. (2019). Internet of Things: A Comprehensive Study of Security Issues and Defense Mechanisms. *IEEE Access*, *7*, 11020–11028. https://doi.org/10.1109/ACCESS.2018.2876939

Alansari, Z., Anuar, N. B., Kamsin, A., Belgaum, M. R., Alshaer, J., Soomro, S., & Miraz, M. H. (2018). Internet of Things: Infrastructure, Architecture, Security and Privacy. *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, 150–155. https://doi.org/10.1109/iCCECOME.2018.8658516

Ali, B., & Awad, A. (2018). Cyber and Physical Security Vulnerability Assessment for IoT-Based Smart Homes. *Sensors*, *18*(3), 817. https://doi.org/10.3390/s18030817

Alsabilah, N., & Rawat, D. B. (2021). Anomaly Detection in Smart Home Networks Using Kalman Filter. *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 1–6. https://doi.org/10.1109/INFOCOMWKSHPS51825.2021.9484507

Ammar, M., Russello, G., & Crispo, B. (2018). Internet of Things: A survey on the security of IoT frameworks. *Journal of Information Security and Applications*, *38*, 8–27. https://doi.org/10.1016/j.jisa.2017.11.002

Aneja, S., Aneja, N., & Islam, M. S. (2018). IoT Device Fingerprint using Deep Learning. *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, 174–179. https://doi.org/10.1109/IOTAIS.2018.8600824

Apthorpe, N., Reisman, D., & Feamster, N. (2017). Closing the Blinds: Four Strategies for Protecting Smart Home Privacy from Network Observers. *arXiv:1705.06809 [cs]*. http://arxiv.org/abs/1705.06809

Ashton, K. (1999). *That 'Internet of Things' Thing*.

Bezawada, B., Bachani, M., Peterson, J., Shirazi, H., Ray, I., & Ray, I. (2018). Behavioral Fingerprinting of IoT Devices. *Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security*, 41–50. https://doi.org/10.1145/3266444.3266452

Butun, I., Osterberg, P., & Song, H. (2020). Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures. *IEEE Communications Surveys & Tutorials*, *22*(1), 616–644. https://doi.org/10.1109/COMST.2019.2953364

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., & Faruki, P. (2019). Network Intrusion Detection for IoT Security Based on Learning Techniques. *IEEE Communications Surveys & Tutorials*, *21*(3), 2671–2701. https://doi.org/10.1109/COMST.2019.2896380

Chettri, L., & Bera, R. (2020). A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems. *IEEE Internet of Things Journal*, *7*(1), 16–32. https://doi.org/10.1109/JIOT.2019.2948888

Conti, M., Dehghantanha, A., Franke, K., & Watson, S. (2018). Internet of Things security and forensics: Challenges and opportunities. *Future Generation Computer Systems*, *78*, 544–546. https://doi.org/10.1016/j.future.2017.07.060

David Warburton. (2021, outubro 20). *The 2021 TLS Telemetry Report*. https://www.f5.com/labs/articles/threat-intelligence/the-2021-tls-telemetry-report

Dewangan, D. K., & Sahu, S. P. (2021). Deep Learning-Based Speed Bump Detection Model for Intelligent Vehicle System Using Raspberry Pi. *IEEE Sensors Journal*, *21*(3), 3570–3578. https://doi.org/10.1109/JSEN.2020.3027097

Dharur, S., & Swaminathan, K. (2018). Efficient surveillance and monitoring using the ELK stack for IoT powered Smart Buildings. *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, 700–705. https://doi.org/10.1109/ICISC.2018.8398888

Doshi, R., Apthorpe, N., & Feamster, N. (2018). Machine Learning DDoS Detection for Consumer Internet of Things Devices. *2018 IEEE Security and Privacy Workshops (SPW)*, 29–35. https://doi.org/10.1109/SPW.2018.00013

Duezguen, R., Mayer, P., Berens, B., Beckmann, C., Aldag, L., Mossano, M., Volkamer, M., & Strufe, T. (2021). How to Increase Smart Home Security and Privacy Risk Perception. *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 997–1004. https://doi.org/10.1109/TrustCom53373.2021.00138

El-Azab, R. (2021). Smart homes: Potentials and challenges. *Clean Energy*, *5*(2), 302–315. https://doi.org/10.1093/ce/zkab010

Ferman, V. A., & Ali Tawfeeq, M. (2021). Machine Learning Challenges for IoT Device Fingerprints Identification. *Journal of Physics: Conference Series*, *1963*(1), 012046. https://doi.org/10.1088/1742-6596/1963/1/012046

Fernandes, E., Jung, J., & Prakash, A. (2016). Security Analysis of Emerging Smart Home Applications. *2016 IEEE Symposium on Security and Privacy (SP)*, 636–654. https://doi.org/10.1109/SP.2016.44

Fuller, A., Fan, Z., Day, C., & Barlow, C. (2020). Digital Twin: Enabling Technologies, Challenges and Open Research. *IEEE Access*, *8*, 108952–108971. https://doi.org/10.1109/ACCESS.2020.2998358

Gartner Research. (2021). *Forecast: IT Services for IoT, Worldwide, 2019-2025*. https://www.gartner.com/en/documents/4004741

Guth, J., Breitenbücher, U., Falkenthal, M., Fremantle, P., Kopp, O., Leymann, F., & Reinfurt, L. (2018). A Detailed Analysis of IoT Platform Architectures: Concepts, Similarities, and Differences. Em B. Di Martino, K.-C. Li, L. T. Yang, & A. Esposito (Eds.), *Internet of Everything* (pp. 81–101). Springer Singapore. https://doi.org/10.1007/978-981-10-5861-5_4

Hafeez, I., Antikainen, M., Ding, A. Y., & Tarkoma, S. (2020). IoT-KEEPER: Detecting Malicious IoT Network Activity Using Online Traffic Analysis at the Edge. *IEEE Transactions on Network and Service Management*, *17*(1), 45–59. https://doi.org/10.1109/TNSM.2020.2966951

Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., & Sikdar, B. (2019). A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. *IEEE Access*, *7*, 82721–82743. https://doi.org/10.1109/ACCESS.2019.2924045

Hedi, I., Speh, I., & Sarabok, A. (2017). IoT network protocols comparison for the purpose of IoT constrained networks. *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 501–505. https://doi.org/10.23919/MIPRO.2017.7973477

Huc, A., & Trcek, D. (2021). Anomaly Detection in IoT Networks: From Architectures to Machine Learning Transparency. *IEEE Access*, *9*, 60607–60616. https://doi.org/10.1109/ACCESS.2021.3073785

Jacobsson, A., Boldt, M., & Carlsson, B. (2016). A risk analysis of a smart home automation system. *Future Generation Computer Systems*, *56*, 719–733. https://doi.org/10.1016/j.future.2015.09.003

Javed, F., Afzal, M. K., Sharif, M., & Kim, B.-S. (2018). Internet of Things (IoT) Operating Systems Support, Networking Technologies, Applications, and Challenges: A Comparative Review. *IEEE Communications Surveys & Tutorials*, *20*(3), 2062–2100. https://doi.org/10.1109/COMST.2018.2817685

Jayalaxmi, P. L. S., Saha, R., Kumar, G., Conti, M., & Kim, T.-H. (2022). Machine and Deep Learning Solutions for Intrusion Detection and Prevention in IoTs: A Survey. *IEEE Access*, *10*, 121173–121192. https://doi.org/10.1109/ACCESS.2022.3220622

Khan, M. A., & Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, *82*, 395–411. https://doi.org/10.1016/j.future.2017.11.022

Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and Other Botnets. *Computer*, *50*(7), 80–84. https://doi.org/10.1109/MC.2017.201

Kumar, D., Shen, K., Case, B., Garg, D., Alperovich, G., Kuznetsov, D., Gupta, R., & Durumeric, Z. (2019). All Things Considered: An Analysis of IoT Devices on Home Networks. *28th USENIX Security Symposium (USENIX Security 19)*, 1169–1185. https://www.usenix.org/conference/usenixsecurity19/presentation/kumar-deepak

Kumar, R., & Rajasekaran, M. P. (2016). An IoT based patient monitoring system using raspberry Pi. *2016 International Conference on Computing Technologies*

and Intelligent Data Engineering (ICCTIDE'16), 1–4. https://doi.org/10.1109/ICCTIDE.2016.7725378

Kuzniar, C., Neves, M., Gurevich, V., & Haque, I. (2022). IoT Device Fingerprinting on Commodity Switches. *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, 1–9. https://doi.org/10.1109/NOMS54207.2022.9789865

Lily Hay Newman. (2017, maio 13). *How an Accidental «Kill Switch» Slowed Friday's Massive Ransomware Attack*. https://www.wired.com/2017/05/accidental-kill-switch-slowed-fridays-massive-ransomware-attack/

Ling, Z., Luo, J., Xu, Y., Gao, C., Wu, K., & Fu, X. (2017). Security Vulnerabilities of Internet of Things: A Case Study of the Smart Plug System. *IEEE Internet of Things Journal*, *4*(6), 1899–1909. https://doi.org/10.1109/JIOT.2017.2707465

Mandalari, A. M., Dubois, D. J., Kolcun, R., Paracha, M. T., Haddadi, H., & Choffnes, D. (2021). *Blocking without Breaking: Identification and Mitigation of Non-Essential IoT Traffic*. https://doi.org/10.48550/ARXIV.2105.05162

Marchal, S., Miettinen, M., Nguyen, T. D., Sadeghi, A.-R., & Asokan, N. (2019). AuDI: Toward Autonomous IoT Device-Type Identification Using Periodic Communication. *IEEE Journal on Selected Areas in Communications*, *37*(6), 1402–1412. https://doi.org/10.1109/JSAC.2019.2904364

Marikyan, D., Papagiannidis, S., & Alamanos, E. (2019). A systematic review of the smart home literature: A user perspective. *Technological Forecasting and Social Change*, *138*, 139–154. https://doi.org/10.1016/j.techfore.2018.08.015

Mazhar, M. H., & Shafiq, Z. (2020). Characterizing Smart Home IoT Traffic in the Wild. *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 203–215. https://doi.org/10.1109/IoTDI49375.2020.00027

Meneghello, F., Calore, M., Zucchetto, D., Polese, M., & Zanella, A. (2019). IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices. *IEEE Internet of Things Journal*, *6*(5), 8182–8201. https://doi.org/10.1109/JIOT.2019.2935189

Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A.-R., & Tarkoma, S. (2017). IoT Sentinel: Automated Device-Type Identification for Security Enforcement in IoT. *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2177–2184. https://doi.org/10.1109/ICDCS.2017.283

Mohajeri Moghaddam, H., Acar, G., Burgess, B., Mathur, A., Huang, D. Y., Feamster, N., Felten, E. W., Mittal, P., & Narayanan, A. (2019). Watching You Watch: The Tracking Ecosystem of Over-the-Top TV Streaming Devices. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 131–147. https://doi.org/10.1145/3319535.3354198

Mohamad Noor, M. binti, & Hassan, W. H. (2019). Current research on Internet of Things (IoT) security: A survey. *Computer Networks*, *148*, 283–294. https://doi.org/10.1016/j.comnet.2018.11.025

Nassiri Abrishamchi, M. A., Zainal, A., Ghaleb, F. A., Qasem, S. N., & Albarrak, A. M. (2022). Smart Home Privacy Protection Methods against a Passive Wireless Snooping Side-Channel Attack. *Sensors*, *22*(21), 8564. https://doi.org/10.3390/s22218564

Nguyen, T. D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., & Sadeghi, A.-R. (2018). *DÏoT: A Federated Self-learning Anomaly Detection System for IoT*. https://doi.org/10.48550/ARXIV.1804.07474

Niu, E. (2018). Roku is beefing up ad targeting in a big way. *Retrieved August*, *25*, 2019.

Nižetić, S., Šolić, P., López-de-Ipiña González-de-Artaza, D., & Patrono, L. (2020). Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future. *Journal of Cleaner Production*, *274*, 122877. https://doi.org/10.1016/j.jclepro.2020.122877

Nurse, J. R. C., Creese, S., & De Roure, D. (2017). Security Risk Assessment in Internet of Things Systems. *IT Professional*, *19*(5), 20–26. https://doi.org/10.1109/MITP.2017.3680959

O. Garcia-Morchon, S. Kumar, & M. Sethi. (2019, abril). *Internet of Things (IoT) Security: State of the Art and Challenges*. https://datatracker.ietf.org/doc/html/rfc8576

OConnor, T., Mohamed, R., Miettinen, M., Enck, W., Reaves, B., & Sadeghi, A.-R. (2019). HomeSnitch: Behavior transparency and control for smart home IoT devices. *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, 128–138. https://doi.org/10.1145/3317549.3323409

OWASP. (2019, novembro 1). *OWASP Internet of Things Project*. Internet of Things (IoT) Top 10 2018. https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Top_10

Papadogiannaki, E., & Ioannidis, S. (2021). A Survey on Encrypted Network Traffic Analysis Applications, Techniques, and Countermeasures. *ACM Computing Surveys*, *54*(6), 1–35. https://doi.org/10.1145/3457904

Pi-Hole. (2022). https://pi-hole.net/

Ramapatruni, S., Narayanan, S. N., Mittal, S., Joshi, A., & Joshi, K. (2019). Anomaly Detection Models for Smart Home Security. *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, 19–24. https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2019.00015

Ronen, E., Shamir, A., Weingarten, A.-O., & OFlynn, C. (2017). IoT Goes Nuclear: Creating a ZigBee Chain Reaction. *2017 IEEE Symposium on Security and Privacy (SP)*, 195–212. https://doi.org/10.1109/SP.2017.14

Samsung. (2020, fevereiro). *TV Ad Retargeting*. https://www.samsung.com/us/business/samsungads/resources/tv-ad-retargeting/

Schiller, E., Aidoo, A., Fuhrer, J., Stahl, J., Ziörjen, M., & Stiller, B. (2022). Landscape of IoT security. *Computer Science Review*, *44*, 100467. https://doi.org/10.1016/j.cosrev.2022.100467

Seralathan, Y., Oh, T. T., Jadhav, S., Myers, J., Jeong, J. P., Kim, Y. H., & Kim, J. N. (2018). IoT security vulnerability: A case study of a Web camera. *2018 20th International Conference on Advanced Communication Technology (ICACT)*, 172–177. https://doi.org/10.23919/ICACT.2018.8323686

Sforzin, A., Marmol, F. G., Conti, M., & Bohli, J.-M. (2016). RPiDS: Raspberry Pi IDS — A Fruitful Intrusion Detection System for IoT. *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, 440–448. https://doi.org/10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0080

Sorri, K., Mustafee, N., & Seppänen, M. (2022). Revisiting IoT definitions: A framework towards comprehensive use. *Technological Forecasting and Social Change*, *179*, 121623. https://doi.org/10.1016/j.techfore.2022.121623

Sovacool, B. K., & Furszyfer Del Rio, D. D. (2020). Smart home technologies in Europe: A critical review of concepts, benefits, risks and policies. *Renewable and Sustainable Energy Reviews*, *120*, 109663. https://doi.org/10.1016/j.rser.2019.109663

Sowah, R. A., Ofoli, A. R., Tetteh, M. K., Opoku, R. A., & Armoo, S. K. (2018). Demand Side Management of Smart Homes Using OpenHAB Framework for Interoperability of Devices. *2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST)*, 1–8. https://doi.org/10.1109/ICASTECH.2018.8506917

Sun, J., Sun, K., & Shenefiel, C. (2019). Automated IoT Device Fingerprinting Through Encrypted Stream Classification. Em S. Chen, K.-K. R. Choo, X. Fu, W. Lou, & A. Mohaisen (Eds.), *Security and Privacy in Communication Networks* (Vol. 304, pp. 147–167). Springer International Publishing. https://doi.org/10.1007/978-3-030-37228-6_8

Tirumala, S. S., Nepal, N., & Ray, S. K. (2022). Raspberry Pi-based Intelligent Cyber Defense Systems for SMEs and Smart-homes: An Exploratory Study. *EAI Endorsed Transactions on Smart Cities*, *6*(18), e4. https://doi.org/10.4108/eetsc.v6i18.2345

Tokar, D., Morozova, O., & Kharchenko, V. (2022). The IoT Applications Productivity: Data Management Model and ELK Tool Based Monitoring and Research. *2022 IEEE 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, 162–167. https://doi.org/10.1109/TCSET55632.2022.9766930

Z. Shelby, K. Hartke, & C. Bormann. (2014, junho). *The Constrained Application Protocol (CoAP)*. https://tools.ietf.org/html/rfc7252

Zeng, E., Shrirang, M., & Franziska, R. (2017). End User Security and Privacy Concerns with Smart Homes. Em *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)* (pp. 65--80). USENIX Association. https://www.usenix.org/conference/soups2017/technical-sessions/presentation/zeng

Zhang, J., Chen, H., Gong, L., Cao, J., & Gu, Z. (2019). The Current Research of IoT Security. *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)*, 346–353. https://doi.org/10.1109/DSC.2019.00059

Zhang, N., Mi, X., Feng, X., Wang, X., Tian, Y., & Qian, F. (2019). Dangerous Skills: Understanding and Mitigating Security Risks of Voice-Controlled Third-Party Functions on Virtual Personal Assistant Systems. *2019 IEEE Symposium on Security and Privacy (SP)*, 1381–1396. https://doi.org/10.1109/SP.2019.00016

Anexo A – REGRAS IPTABLE

```
### 1: Drop invalid packets
sudo iptables -t mangle -A PREROUTING -m conntrack --ctstate
INVALID -j DROP

### 2: Drop TCP packets that are new and are not SYN
sudo iptables -t mangle -A PREROUTING -p tcp ! --syn -m
conntrack --ctstate NEW -j DROP

### 3: Drop SYN packets with suspicious MSS value
sudo iptables -t mangle -A PREROUTING -p tcp -m conntrack --
ctstate NEW -m tcpmss ! --mss 536:65535 -j DROP

### 4: Block packets with bogus TCP flags
sudo iptables -t mangle -A PREROUTING -p tcp --tcp-flags
FIN,SYN FIN,SYN -j DROP
sudo iptables -t mangle -A PREROUTING -p tcp --tcp-flags
FIN,SYN,RST,PSH,ACK,URG NONE -j DROP
sudo iptables -t mangle -A PREROUTING -p tcp --tcp-flags
SYN,RST SYN,RST -j DROP
sudo iptables -t mangle -A PREROUTING -p tcp --tcp-flags
FIN,RST FIN,RST -j DROP
sudo iptables -t mangle -A PREROUTING -p tcp --tcp-flags
FIN,ACK FIN -j DROP
sudo iptables -t mangle -A PREROUTING -p tcp --tcp-flags
ACK,URG URG -j DROP
sudo iptables -t mangle -A PREROUTING -p tcp --tcp-flags
ACK,FIN FIN -j DROP
sudo iptables -t mangle -A PREROUTING -p tcp --tcp-flags
ACK,PSH PSH -j DROP
sudo iptables -t mangle -A PREROUTING -p tcp --tcp-flags ALL
ALL -j DROP
sudo iptables -t mangle -A PREROUTING -p tcp --tcp-flags ALL
NONE -j DROP
sudo iptables -t mangle -A PREROUTING -p tcp --tcp-flags ALL
FIN,PSH,URG -j DROP
sudo iptables -t mangle -A PREROUTING -p tcp --tcp-flags ALL
SYN,FIN,PSH,URG -j DROP
sudo iptables -t mangle -A PREROUTING -p tcp --tcp-flags ALL
SYN,RST,ACK,FIN,URG -j DROP

### 5: Block spoofed packets
sudo iptables -t mangle -A PREROUTING -s 224.0.0.0/3 -j DROP
sudo iptables -t mangle -A PREROUTING -s 169.254.0.0/16 -j
DROP
```

```
sudo iptables -t mangle -A PREROUTING -s 172.16.0.0/12 -j
DROP
sudo iptables -t mangle -A PREROUTING -s 192.0.2.0/24 -j
DROP
sudo iptables -t mangle -A PREROUTING -s 10.0.0.0/8 -j DROP
sudo iptables -t mangle -A PREROUTING -s 0.0.0.0/8 -j DROP
sudo iptables -t mangle -A PREROUTING -s 240.0.0.0/5 -j DROP
sudo iptables -t mangle -A PREROUTING -s 127.0.0.0/8 ! -i lo
-j DROP

### 6: Drop ICMP
sudo iptables -t mangle -A PREROUTING -p icmp -j DROP

### 7: Drop fragments in all chains
sudo iptables -t mangle -A PREROUTING -f -j DROP

### 8: Limit connections per source IP
sudo iptables -A INPUT -p tcp -m connlimit --connlimit-above
111 -j REJECT --reject-with tcp-reset

### 9: Limit RST packets
sudo iptables -A INPUT -p tcp --tcp-flags RST RST -m limit -
-limit 2/s --limit-burst 2 -j ACCEPT
sudo iptables -A INPUT -p tcp --tcp-flags RST RST -j DROP

### 10: Limit new TCP connections per second per source IP
sudo iptables -A INPUT -p tcp -m conntrack --ctstate NEW -m
limit --limit 60/s --limit-burst 20 -j ACCEPT
sudo iptables -A INPUT -p tcp -m conntrack --ctstate NEW -j
DROP

### SSH brute-force protection
sudo iptables -A INPUT -p tcp --dport ssh -m conntrack --
ctstate NEW -m recent --set
sudo iptables -A INPUT -p tcp --dport ssh -m conntrack --
ctstate NEW -m recent --update --seconds 60 --hitcount 10 -j
DROP


### Protection against port scanning
sudo iptables -N port-scanning
sudo iptables -A port-scanning -p tcp --tcp-flags
SYN,ACK,FIN,RST RST -m limit --limit 1/s --limit-burst 2 -j
RETURN
sudo iptables -A port-scanning -j DROP

### Slowloris
sudo iptables -I INPUT -p tcp --dport 80 -m connlimit --
connlimit-above 20 --connlimit-mask 32 -j DROP
iptables -A INPUT -p tcp --dport 80 -m state --state NEW -m
recent --set
iptables -A INPUT -p tcp --dport 80 -m state --state NEW -m
recent --update --seconds 60 --hitcount 20 -j DROP
```