UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE

Breno Fabrício Lira Melo Sousa

# *Um Sistema de Detecção de Intrusão para Detecção de Ataques de Negação de Serviço na Internet das Coisas*

São Luís - MA
2016

Breno Fabrício Lira Melo Sousa

# *Um Sistema de Detecção de Intrusão para Detecção de Ataques de Negação de Serviço na Internet das Coisas*

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão como requisito parcial para a obtenção do grau de MESTRE em Engenharia de Eletricidade.

**Orientador: Denivaldo Cicero Pavão Lopes**

**Doutor em Ciência da Computação – UFMA**

São Luís - MA

2016

Sousa, Breno Fabrício Lira Melo

Um Sistema de Detecção de Intrusão para Detecção de Ataques de Negação de Serviço na Internet das Coisas / Breno Fabrício Lira Melo Sousa. – São Luís - MA, 2016.

93 f.

Orientador: Denivaldo Cicero Pavão Lopes.

Impresso por computador (fotocópia).

Dissertação (Mestrado) – Universidade Federal do Maranhão, Programa de Pós-Graduação em Engenharia de Eletricidade. São Luís - MA, 2016.

1. Sistema de Detecção de Intrusão. 2. Internet das Coisas. 3. Segurança em Redes. 4. Negação de Serviço I. Lopes, Denivaldo, orient. II. Título.

CDU

Breno Fabrício Lira Melo Sousa

*Um Sistema de Detecção de Intrusão para Detecção de Ataques de Negação de Serviço na Internet das Coisas*

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por Breno Fabrício Lira Melo Sousa e aprovada pela comissão examinadora.

Aprovada em ___ de _____ de 2016

**BANCA EXAMINADORA**

---

Denivaldo Cicero Pavão Lopes (orientador)

Doutor em Ciência da Computação – UFMA

---

Daniela Barreiro Claro

Doutora em Ciência da Computação – UFBA

---

Francisco José da Silva e Silva

Doutor em Ciência da Computação – UFMA

*A todos que desde o início de minha jornada acadêmica me incentivaram. Aos meus pais Manoel da Cruz de Sousa e Maria dos Anjos Lira Melo de Sousa e minha irmã, Camila Lorena Lira Melo Sousa.*

# Resumo

O paradigma da Internet das Coisas (em inglês, Internet of Things - IoT) surgiu para possibilitar a intercomunicação entre os diferentes objetos através da Internet, e, com isso, facilitar a forma de como o usuário final interagirá com a grande variedade de dispositivos que o cerca no dia a dia. A disponibilidade de recursos que estes dispositivos possuem é um fator que merece uma grande atenção, pois o uso de tais recursos de forma não apropriada pode gerar graves danos. Para tanto, uma vez que tais dispositivos estão conectados à Internet, estes estão vulneráveis a diversas ameaças, como, por exemplo, ataque de negação de serviço (DoS). A fim de enfrentar ameaças do tipo DoS em IoT, propõe-se um IDS (*Intrusion Detection System*) para IoT, objetivando a detecção de alguns ataques do tipo DoS.

Palavras-chaves: Sistema de Detecção de Intrusão, Internet das Coisas, Segurança, Negação de Serviço.

# Abstract

The paradigm of the Internet of Things (in english, Internet of Things - IoT) came to allow intercommunication between different objects via Internet, and thereby facilitate the form of how the end user will interact with a wide variety of devices that surround him in everyday life. The availability of features that these devices have is a factor that deserves great attention because the use of such resources inappropriately can cause serious damage. Therefore, since such devices are connected to the internet, they are vulnerable to various threats, such as, denial-of-service attack (DoS). In order to tackle DoS type threats in IoT, an Intrusion Detection System (IDS) is proposed for IoT, aiming at detecting some types of DoS attacks.



Keywords: Intrusion Detection System, Internet of Things, Security, Denial of Service.

# Agradecimentos

Primeiramente, agradeço a Deus por estar sempre me guiando pelos caminhos da sabedoria, oferecendo-me força para nunca desistir e me desviar de meus objetivos.

Agradeço especialmente aos meus pais, Manoel [*in memorian*] que mesmo não estando fisicamente presente neste plano terreno nunca deixou de me acompanhar em minha vida, e à minha mãe, Dos Anjos, que sempre foi uma mãe exemplar e batalhadora, dando-me carinho e conforto nos momentos difíceis.

À minha irmã, que sempre me deu força para eu nunca desistir de meus sonhos.

Aos meus tios Ferdinand e César, que apesar de tios, tiveram um papel de pai em minha vida.

À minha namorada, Tainara, que com seus carinhos e compreensão consegui me manter no foco e nunca me desviar de meus reais objetivos.

Aos meus avós maternos e paternos, em especial aos meus avós maternos Maria dos Anjos e Almir [*in memorian*] (vulgo Almir das meninas) por tudo que representaram e representam em minha vida.

Ao meu orientador Zair Abdelouhab [*in memorian*] por ter me acolhido no inicio da pesquisa desenvolvida. Ao professor Denivaldo Lopes por ter me acolhido após o professor Zair falecer, assumindo o papel de orientador.

Ao Willian Ribeiro por me auxiliar na produção deste sistema e aos colegas de laboratório.

E, por fim, à todos que me ajudaram nessa etapa de minha vida.

*"Somente após ter visualizado o que nos reserva o futuro disporemos de força e determinação suficientes para investigar o passado de maneira honesta e imparcial"*

*Erich Von Däniken*

# Lista de Figuras

# Lista de Tabelas

# Lista de Siglas

| | |
|---|---|
| DDoS | *Distributed Denial-of-Service* . |
| DoS | Denial of Service. |
| ICMP | *Internet Control Message Protocol* . |
| IDS | *Intrusion Detection System* . |
| IoT | *Internet of Things* . |
| IP | *Internet Protocol* . |
| IPv6 | *Internet Protocol version 6* . |
| LABSAC | Laboratório de Sistemas e Arquiteturas Computacionais. |
| MIT | *Massachusetts Institute of Technology* . |
| RFID | *Radio-Frequency IDentification* . |
| TLS | Transport Layer Security. |
| TPC | *Transmission Control Protocol* . |
| UDP | *User Datagram Protocol* . |
| UFMA | Universidade Federal do Maranhão. |
| UML | *Unified Modeling Language* . |
| VPN | *Virtual Private Network* . |

# Sumário

# 1 Introdução

Esta dissertação apresenta um Sistema de Detecção de Intrusão para Internet das Coisas denominado de IDS-IoT. Por sua vez, o IDS-IoT deverá apresentar resultados favoráveis para a detecção de alguns tipos de ataques de negação de serviço (DoS), tais como: *Syn Flood*, *Land Attack*, *Smurf Attack*, *ICMP Flood* e *UDP Flood*.

O desenvolvimento desta pesquisa terá como base a seguinte hipótese: Sistema de Detecção de Intrusão é o sistema ideal para analisar o comportamento da rede e identificar ataques de Negação de Serviço em tempo real.

Este capítulo apresenta uma visão geral do trabalho desenvolvido. A seção 1.3 apresentar a motivação e a justificativa para o desenvolvimento desta dissertação de Mestrado. A seção 1.2 expõe a problemática do trabalho. A seção 1.5 descreve o objetivo geral e os objetivos específicos. Por último, a seção 1.7 apresenta a estrutura na qual a dissertação está organizada.

## 1.1 Contexto

A computação ao longo dos anos vem crescendo desde quando surgiram os primeiros computadores, que tinham finalidades militares. As informações tornaram-se acessíveis para todos, a qualquer hora e lugar. Em seu processo de aprimoramento, os computadores foram ganhando cada vez mais melhorias: poder de processamento e armazenamento; e até mesmo de tamanho. Chegaram a um ponto de mudar o modo de vida da população mundial, dando uma maior comodidade e solucionando problemas que até então poderiam ser quase que impossível de serem resolvidos sem a utilização dos mesmos.

Todos os dias, nos deparamos com máquinas cada vez menores e mais potentes. É perceptível que o crescente número de pessoas que possuem dispositivos com um grande poder de processamento e armazenamento.

Com esse acelerado surgimento de novas tecnologias, a segurança dos dados trocados entre os dispositivos é de suma importância, pois, caso estes sejam acessados por indivíduos não autorizados, podem provocar danos irreparáveis.

Todavia, a características da possibilidade de interconexão e troca de informações entre os dispositivos, através da Internet, constitui um novo paradigma denominado de *Internet of Things* (IoT). No qual tais dispositivos geram uma quantidade massiva de dados, que devem ser gerenciados com o maior grau de segurança possível.

Em cenário de IoT a segurança é um dos fatores principais que devem ser garantidos, uma vez que existem uma variedade de dispositivos que trocam mensagens entre si. Como tentativa de barrar um eventual invasor, deve-se fazer uso de ferramentas que garantam tal segurança.

## 1.2 Problemática

A necessidade de novas ferramentas que forneçam segurança, no contexto de IoT, é um fator primordial. O crescimento no número de novos dispositivos englobados em IoT — em alguns casos, sem mecanismos que garantam um maior nível de segurança — fez com que o crescimento no interesse no estudo de novos mecanismos de segurança surgissem.

Como alternativa para tentar garantir um maior nível de segurança, destacamos algumas ferramentas, tais como: *Firewall*, *Intrusion Detection System* (IDS), *Virtual Private Network* (VPN), etc. Porém, na escolha de qual ferramenta melhor satisfaz a solução do problema, deve ser levado em conta a disponibilidade dos recursos que cada dispositivo possui, como: poder de processamento, memória e armazenamento, são exemplos de recursos que devem ser considerados ao escolher uma ferramenta de segurança para dispositivos IoT.

Contudo, ataques de negação de serviços são ameaças contantes em ambientes computacionais, merecendo cada vez mais atenção. Dentre os ataques do

tipo DoS, o presente trabalho tem por finalidade detectar e barrar o seguintes ataques DoS: *Syn Flood*[1], *Land Attack*[2], *Smurf Attack*[3], *ICMP Flood*[4] e *UDP Flood*[5].

## 1.3 Motivação e Justificativa

Após o surgimento dos primeiros computadores, e o surgimento da Internet, o modo de vida da população mudou drasticamente. À medida que a 'evolução' dos computadores ocorre de forma acelerada, novas formas de tecnologias surgem com esse avanço.

Com a decorrência do surgimento de novas tecnologias, surgiu um novo paradigma: IoT. Segundo Gendreau [14], IoT trata-se de uma rede de computadores heterogênea, em que uma variedade de dispositivos podem comunicar entre si através da Internet.

Com a possibilidade de diversos dispositivos poderem se comunicar, uma nova gama de possibilidades surgiu. As aplicações de IoT podem estar presentes em diversas áreas auxiliando na resolução de problemas como também em áreas de entretenimento, ou até mesmo no meio econômico.

Com o ganho da comunicação entre os diversos dispositivos, ressalta-se que tal comunicação gera uma quantidade massiva de dados trafegados na Internet. Esses dados, merecem uma atenção especial, pois como se trata de um novo paradigma, as soluções existentes de segurança demonstram-se limitadas na maioria dos casos.

Bhattasali et al. [4] relatam que para garantir uma maior confiabilidade nos dados trafegados, a rede deve ser configurada para trabalhar com o protocolo *Internet Protocol version 6* (IPv6) para lidar com a quantidade massiva de diferentes dispositivos.

[1]Disponível em: https://tools.ietf.org/html/rfc4987. Acessado em: 02/01/2017.

[2]Disponível em: https://www.corero.com/resources/glossary.html#Same Source/Dest Flood (LAND Attack). Acessado em: 02/01/2017.

[3]Disponível em: https://www.symantec.com/pt/br/security_response/glossary/define.jsp?letter=s&word=sm dos-attack. Acessado em: 02/01/2017.

[4]Disponível em: https://www.corero.com/resources/glossary.html#ICMP Flood. Acessado em: 02/01/2017.

[5]Disponível em: https://www.corero.com/resources/glossary.html#UDP Flood. Acessado em: 02/01/2017.

Destaca-se ainda que a Internet das Coisas pode trabalhar em harmonia com outras tecnologias (por exemplo, computação em nuvem), trabalhando em conjunto para garantir uma melhor experiência em sua adoção.

Contudo, as ameaças estão em constante evolução, fazendo necessário a utilização de ferramentas que se adequem às novas eventualidades. Por exemplo, os sistemas de detecção de intrusão (do inglês, *Intrusion Detection System* - IDS) possuem a capacidade de mudar as suas configurações de acordo com a percepção atual do ambiente, ou seja, conforme o contexto do ambiente em que estão inseridos com o intuito de melhorar a detecção de intrusões.

## 1.4 Solução Proposta

Com relação às ferramentas de segurança abordadas na seção 1.2 , IDS foi o mecanismo de segurança utilizado. Neste trabalho, propõe-se um IDS para IoT para detectar ataques do tipo Denial of Service (DoS) que tem por finalidade sobrecarregar os recursos disponíveis dos dispositivos, principalmente, largura de banda, memória e processamento.

## 1.5 Objetivos Gerais e Específicos

O objetivo deste trabalho é desenvolver um sistema de detecção de intrusão para Internet das Coisas, sem que este interfira no desempenho dos dispositivos que estão presentes na rede de computadores.

Esta dissertação tem por objetivos específicos:

1. Desenvolver uma ferramenta que seja capaz de garantir a segurança para dispositivos em IoT, utilizando uma quantidade mínima de recursos possíveis;
2. Ter comportamento ativo, ou seja, tomar ações para barrar atividades maliciosas de atacantes;
3. Realizar testes do sistema desenvolvido;
4. Analisar os resultados gerados pela ferramenta.

## 1.6 Metodologia

Esta dissertação objetiva desenvolver um estudo para solução do problema de acesso à informações por dispositivos não autorizados em ambiente IoT. O primeiro passo consiste no levantamento bibliográfico em livros, teses, artigos científicos, anais de congressos, dissertações, periódicos, *websites* e biblioteca desta instituição de ensino, visando à aquisição da fundamentação teórica e à produção de relatórios das áreas de Internet das Coisas, Segurança da Informação e Sistema de Detecção de Intrusão.

No segundo passo, a definição do escopo do problema será feita, delimitando-se até que ponto será a contribuição da pesquisa. Esperou-se que nesta fase fossem conhecidos os requisitos de segurança, protocolo de comunicação seguro de implantação na IoT, e a implementação do IDS.

Em seguida, a implementação do ambiente de testes foi feita, que tem o objetivo de testar o funcionamento do IDS desenvolvido, que utilizará como dispositivo de teste um *Raspberry Pi*.

## 1.7 Estrutura da Dissertação

Esta dissertação encontra-se organizada em 6 capítulos, descritos a seguir:

No Capítulo 2, uma contextualização tecnológica sobre IoT é realizada, ameaças que circundam as redes de computadores e a IoT; é dado a definição de Sistema de Detecção de Intrusão, e em seguida, é abordado o *firewall Iptables*.

O Capítulo 3 apresenta o Estado da Arte, no qual são expostos alguns dos trabalhos já apresentados à Comunidade Científica sobre os Sistemas de Detecção de Intrusão para a Internet das Coisas, no qual cada um possui suas particularidades.

O sistema proposto é apresentado no Capítulo 4, no qual será exposta a arquitetura utilizada no desenvolvimento da aplicação proposta, bem como os diagramas *Unified Modeling Language* (UML) utilizados, pseudo algoritmos utilizados na implementação e a demonstração do protótipo da aplicação.

O Capítulo 5 é uma avaliação dos resultados obtidos a partir da aplicação desenvolvida, bem como o ambiente de teste no qual este foi empregado.

Por fim, o Capítulo 6 apresenta as conclusões obtidas no desenvolvimento deste trabalho, possíveis trabalhos futuros e publicação de trabalho científico realizado.

# 2 Contexto Tecnológico

Neste capítulo, as tecnologias necessárias para o desenvolvimento do IDS para IoT serão apresentadas. O entendimento de cada tecnologia exerce um papel fundamental para a integração entre as tecnologias.

## 2.1 Internet das Coisas (IoT)

A IoT teve seu surgimento nos laboratórios do *Massachusetts Institute of Technology* (MIT). Evans [13] destaca que os estudos sobre rede de sensores sem fios (RFID) teve o papel primordial para Internet das Coisas. Evans [13] destaca ainda que o grupo de pesquisa *Cisco Internet Business Solutions Group* (IBSG) relata que a IoT teve o seu surgimento a partir do momento que foi detectado que o número de dispositivos conectados foi maior que o número da população mundial.

As "coisas", mencionadas até aqui, tratam-se de quaisquer dispositivos que tenham a capacidade de conectar-se à Internet, podendo, ou não, trabalhar em cooperação na resolução de atividades específicas, conforme o ambiente em que se encontram.

Como destacam Roman *et al.* [36], a IoT apresenta a característica de conectar seres humanos às "coisas", provendo informações em qualquer lugar ou momento sobre qualquer dispositivo desejado, uma vez que este tenha autorização para solicitar as informações solicitadas.

Conforme Medaglia *et al*. [25], ao se falar em IoT temos que ter em mente de que o conceito deste paradigma está relacionado com a intercomunicação entre dispositivos inteligentes e heterogêneos.

Como os dispositivos de IoT apresentam a característica de ter a capacidade de intercomunicação, no qual estes podem compartilhar informações sobre determinado cenário e cooperação para execução de uma dada tarefa. Mendes [26] destaca alguns dos cenários que IoT pode ser utilizada de forma harmoniosa, tais como: *Healthcare*, *Tracking*, *Smart Cities* e *Smart Agriculture*.

*Healthcare* está voltada para a saúde de pacientes, que devido às suas características de saúde, não precisam obrigatoriamente fazer seus tratamentos em hospitais. Dispositivos IoT podem ser utilizados neste campo para fazer a medição de glicose de um paciente, monitorar a pressão sanguínea do paciente, entre outros.

As aplicações de *tracking* podem ser utilizadas em grande gama de situações. O rastreamento pode ser utilizado, por exemplo, para encontrar animais perdidos (a coleira dos animais precisa possuir algum tipo sensor) e encontrar o posicionamento dos produtos em um armazém.

Já em relação às *smart cities*, a IoT pode trazer um grande benefício para a população, melhorando assim a sua qualidade de vida. Zanella *et al*. [47] destaca algumas aplicações em cidades inteligentes: monitoramento da saúde estrutural de edifícios, gerenciamento de lixo, monitoramento da qualidade do ar, monitoramento do consumo de energia da cidade, entre outros.

*Smart agriculture* objetiva a redução dos custos para agricultores, no qual podem existir dispositivos que possuem a finalidade de coletar informações sobre a área a ser cultivada, provendo assim os dados necessários ao agricultor.

Como a ideia central de IoT é a possibilidade de uma variedade de dispositivos heterogêneos se intercomunicar, trocar informações entre si, e fornecer resultados — conforme o ambiente no qual estes estarão a ser utilizados — úteis para os usuários. Porém, as variedades de trocas de mensagens podem se tornar um grande problema de segurança da informação [22].

Li *et al*. [22] destacam alguns dos problemas enfrentados por IoT, que merecem uma atenção maior, tais como:

- **Desafios técnicos**: os desafios técnicos que circundam IoT variam muito de contexto, no qual estes vão desde dificuldades em implantação de *web services* (devido aos recursos disponíveis em dispositivos IoT, podendo refletir diretamente em seu desempenho); a características de heterogeneidade (gerenciar uma quantidade muito grande de dispositivos diferentes não é uma tarefa trivial de se realizar); outra problemática é a falta utilização de uma linguagem de descrição de serviços;

- **Padronização**: pode-se dizer, que a padronização é um elemento primordial para o crescimento de IoT. Uma vez que esta seja estabelecida, poderá resultar em: redução de obstáculos na adesão de novos prestadores de serviços, resolver problemas de interoperabilidade (definir quais protocolos de comunicação são mais seguros e mais confiáveis);

- **Segurança e proteção da privacidade**: aceitação às novas tecnologias depende de muitos fatores. Um desses fatores é a segurança das informações. Sistemas de segurança para IoT precisam garantir que a privacidade de seus usuários será garantida.

## 2.2 Ameaças em Redes de Computadores

É um fato notório o crescimento da quantidade de dispositivos conectados à rede mundial de computadores, ou seja, a Internet. Embora esse número seja massivo, em muitos dos casos, usuários não possuem habilidades mínimas de segurança da informação, tornando-se, assim, alvos "fáceis" para *hackers*.

Em um relatório sobre as ameaças na Internet — ocorridas no ano de 2015 —, divulgado em abril de 2016 por Symantec [39], é possível se ter uma noção de como o surgimento de novas ameaças não para de crescer, colocando em risco cada vez mais o mundo digital. No relatório, é destacado que em 2015 teve um aumento de 36% (trinta e seis por cento) de *malware* [6] detectados, que corresponde a um acréscimo de 430 milhões de detecção de novos casos.

Embora a variedade de tipos de ameaças seja muito grande, CERT.br [5] faz uma explanação dos tipos de ameaças existente em redes de computadores, que segue:

- **Interceptação de tráfego**: o invasor tem por objetivo coletar pacotes que estão trafegando na rede, através de *sniffers*. Uma vez os pacotes capturados, o invasor coleta informações de tais pacotes;

[6] Ozsoy [31] menciona que, de forma geral, *malware* pode ser definido como qualquer programa que possa causar danos a outros programas.

- **Força bruta**: em realizar este tipo de ataque, o atacante tenta ganhar acesso em algum sistema que necessite de informações de usuário e senha. Múltiplas tentativas de usuários e senhas são utilizados a fim de ganhar acesso ao sistema;

- **Falsificação de *e-mails***: *e-mail spoofing*, como é mais conhecido este tipo de ataque, consiste em utilizar informações falsas com o intuito de enganar uma vítima. Geralmente este tipo de e-mail falso possui *links* de sites falsificados;

- **DoS - Denial of Service**: basicamente caracteriza-se por consumir de forma inapropriada os recursos (largura de banda, memória e processamento, por exemplo) do dispositivo-vítima.

### 2.2.1 Ataques de Negação de Serviço

Os ataques de DoS tem por principal característica consumir os recursos de uma máquina, ocasionando o não fornecimento dos serviços fornecidos por tal dispositivo. Quando este tipo de ataque é bem-sucedido, o dano causado pode ser incalculável, pois, em muito dos casos, os alvos são grandes corporações que lidam com uma grande quantidade de transações financeiras [7] [8].

Uma variação deste tipo de ameaça, é o ataque de *Distributed Denial-of-Service* (DDoS), que se demonstra ser mais perigoso e mais complexo para a sua detecção. Um atacante para realizar um DDoS necessita de uma quantidade significativa de *botnets*, ou seja, máquinas "zumbis". No qual tais maquinas estão sob o controle do atacante, denominado de *botmaster* [16].

CERT.br [5] destaca ainda que os ataques de DoS e DDoS podem ser realizados de várias formas, que segue:

- **Número de requisições**: um sistema pode sofrer um ataque de DDoS quando possui muitos clientes, no qual o sistema recebe uma grande quantidade de requisições e ultrapassa a quantidade suportada pelo servidor. Esta forma pode ser considerada um ataque de DDoS não intencional;

[7] Disponível em: http://g1.globo.com/tecnologia/noticia/2010/12/ataque-hacker-em-defesa-wikileaks-afeta-pagamentos-com-mastercard.html. Acessado em: 25/11/2016.

[8] Disponível em: http://g1.globo.com/tecnologia/noticia/2011/05/entenda-o-ataque-rede-line-do-playstation-3-psn.html. Acessado em: 25/11/2016.

- **Quantidade de dados na rede**: ocorre quando o numero de dados trafegados em uma rede de computadores é maior que a largura de banda suportada, ocasionando a indisponibilidade dos serviços em rede;

- **Exploração de vulnerabilidades**: após um dispositivo ser comprometido por algum *software* malicioso, este pode deixar os serviços disponibilizados indisponíveis.

## 2.3 Ameaças em IoT

Ameaças em ambientes computacionais é um fator inevitável, pois à medida que novas tecnologias surgem, as técnicas e métodos de invasão se aprimoram. Os dispositivos de Internet das Coisas não ficaram de fora das ameaças já enfrentadas em redes de computadores.

Em função da grande variedade (heterogeneidade) e quantidade de dispositivos que podem se intercomunicar, as medidas de segurança devem estar constantemente se aprimorando, e se adequando às novas situações, pois o surgimento de novos ataques é constante.

A possibilidade de ameaças internas existe, em que nós (dispositivos) pertencentes à rede são responsáveis pelos ataques aos outros nós. Contramedidas nessas eventuais situações são necessárias, garantindo assim, a segurança da rede.

Outra possibilidade de ameaças, são as ameaças externas, ou seja, nós não pertencentes à rede utilizam mecanismos para ganhar acesso não autorizado aos recursos da rede ou até mesmo capturar informações dos nós, provocando grandes prejuízos.

Ainda sobre o relatório divulgado por Symantec [39], menciona que é crescente o número de detecções de atividades maliciosas em IoT, no qual tem por principais motivos os mecanismos de seguranças para autenticação e problemas de criptografia utilizado, embora a pesquisa de soluções favoráveis seja algo constante. O relatório menciona, ainda, exemplos onde tais problemas foram detectados, que segue:

- **Carros**: a empresa Fiat Chrysler, após pesquisadores demonstrarem que conseguiram controlar seus veículos remotamente, fez um *recall* de 1.4 milhões de veículos;

- **Dispositivos de *smart homes***: ao realizar o relatório, a Symantec identificou cerca de 50 dispositivos que tem por finalidade fornecer segurança domiciliar apresentam falhas de segurança;

- **Dispositivos médicos**: pesquisadores detectaram que vários dos dispositivos (bombas de insulinas e sistemas de raio-x, por exemplo) utilizados por pacientes estão susceptíveis à ameaças externas, podendo provocar a morte de tais pacientes;

- ***Smart* TVs**: é crescente o número de televisores com a capacidade de se conectar à Internet, porém, tais dispositivos não possuem meios que garantam sua segurança, podendo, assim, serem vítimas de roubo de dados, *botnets*.

Hu [17] destaca uma quantidade significativa de ameaças aos dispositivos. Porém, é destacado algumas delas, que segue: *Sinkhole attack*, *Selective forwarding*, *Wormhole attack*, *IP spoof attack* e *Distributed denial of service*.

No *Sinkhole attack* o invasor tem por finalidade atrair os pacotes de rede dos demais nós, para algum dispositivo que teve a sua segurança comprometida. Em suma, trata-se de um ataque de roteamento, no qual o atacante fornece novas rotas para os pacotes, podendo, ainda, executar outro ataque: encaminhamento seletivo (*selective forwarding*).

Ao executar o ataque *selective forwarding*, o ataque determina quais pacotes e tipos de informações este deseja capturar, podendo ocasionar uma negação de serviço.

No *wormhole attack*, o atacante não necessita comprometer os dispositivos que compõem a rede [17]. Esse ataque caracteriza-se pela criação de um túnel entre atacantes, no qual os dados capturados pelos invasores serão transportados.

Após conseguir acesso à rede, o invasor simula a sua identidade, no qual este assume um endereço IP da rede. Ao assumir um IP da rede, o invasor camufla-se como um dispositivo legítimo e começa a ter acesso aos recursos da rede [17]. Estas ações caracterizam o *IP spoof attack*.

Um ataque *distributed denial-of-service* tem por finalidade sobrecarregar os recursos de um dispositivo/servidor, a fim de ocasionar o não fornecimento dos serviços disponibilizados, bem como ocasionando o esgotamento dos recursos disponíveis para o dispositivo.

Segundo Machaka *et al*. [24], ao ser realizado um ataque de DDoS, o invasor tenta interromper o fornecimento dos serviços aos demais dispositivos da rede. Em meio empresarial, um ataque DDoS pode ocasionar uma grande perda financeira para as organizações.

### 2.3.1 Ataques de Negação de Serviço na IoT

À medida que a IoT ganha mais espaço no cotidiano, torna-se cada vez mais importante prover mecanismo que garantam a segurança dos dispositivos, pois uma vez que estão conectados à Internet, estão susceptíveis à ameaças externas e uma delas são os ataques de negação de serviço.

Weber [45] menciona que os ataques de DDoS são provavelmente o maior problema de segurança enfrentado em IoT. De fato, o fato de se ter uma quantidade gigantesca torna um alvo bastante atrativo para atacantes. Embora os motivos que levam a realização dos ataques, sejam eles de DoS/DDoS ou não, não sejam imprevisíveis, não sabendo ao certo o real motivo, são destacados alguns [6]:

- **Demonstração de poder**: provar para as empresas que estas não estão seguras;
- **Prestígio**: Esbanjar-se de ter conseguido com sucesso uma invasão/ataque para outros atacantes;
- **Motivação financeira**: roubar informações de usuários para realizar golpes na Internet;
- **Motivações ideológicas**: atacar *sites* contrários aos seus preceitos religiosos;
- **Motivações comerciais**: atacar sistemas de empresas concorrentes, a fim de deixá-los inacessíveis.

Um grupo de *hackers*, denominados *Lizard Squad*, no qual ficaram conhecidos por ataques de DDoS a servidores de jogos online, anunciaram que

possuem um poderio de 400Gbps para realização de ataques de DDoS. Os ataques são realizados utilizando a técnica *LizardStressor* que invade dispositivos para torná-los *botnets*. O alvo deste grupo são dispositivos IoT que apresentam configurações padrões, ou seja, são mais "fáceis" de tornarem-se "zumbis", para contribuir no poderio de 400Gbps de ataque [9].

A variedade de dispositivos que são utilizados pelo grupo *Lizard Squad* é enorme. Este utilizando desde câmeras de vigilância à *webcams* espalhadas pelo mundo para a realização dos ataques de DDoS. Embora a capacidade de tais dispositivos não ser bastante poderoso para a realização com eficiência (ou seja, deixar um sistema inacessível), a junção de vários dispositivos "zumbis" torna isso possível[10].

## 2.4 Bibliotecas para Captura de Pacotes

Uma vez que existe a necessidade de captura de pacotes que estão sendo trafegados na rede, esta seção visa apresentar algumas das bibliotecas que visam suprir tão necessidade. Seguem algumas das bibliotecas para a captura de pacotes:

- **Biblioteca PCAP:**: trata-se de uma biblioteca que realiza a captura de pacotes que estão trafegando na rede, independentemente de protocolo [3];
- **Biblioteca libpcap:** é uma biblioteca C/C++ portátil para realizar a captura de pacotes na rede. Libpcap trabalha em conformidade com a ferramenta *tcpdump*. Tal ferramenta é utilizada para a captura e análise de pacotes trafegados de rede [40];
- **Biblioteca Jpcap:** é uma biblioteca Java *open source* e possui licença GNU LGPL. Ao utilizar Jpcap, esta possibilita tanto o envio e captura de pacote na rede. Jpcap está habilitada para capturar pacote dos protocolos IPv4, IPv6, ARP/RARP, TCP, UDP e ICMPv4 [15].

[9]Disponível em: http://www.iottechnews.com/news/2016/jun/30/lizard-squads-iot-botnet-launches-400gbps-ddos-attack/. Acessado em: 29/11/2016.

[10]Disponível em: http://themerkle.com/iot-devices-are-being-hacked-by-lizard-squad-to-execute-ddos-attacks/. Acessado em: 29/11/2016.

## 2.5 Iptables

*Iptables* faz parte do projeto Netfilter e trata-se de *firewall* do sistema operacional Linux, no qual sua configuração é feita em linha de comando, em que os usuários podem criar suas próprias regras de filtragem, para determinar quais pacotes são permitidos trafegar na rede [28]. *Iptables* tem a funcionalidade de suportar os protocolos *Transmission Control Protocol* (TPC), *Internet Control Message Protocol* (ICMP) e *User Datagram Protocol* (UDP).

Este possui três chains que realizam toda a filtragem dos pacotes trafegados, que são: *INPUT*, *FORWARD* e *OUTPUT*. O *chain INPUT* fica responsável por filtrar os pacotes que então chegando ao dispositivo, ou seja, que estão "entrando" na placa de rede. Já o *chain FORWARD* fica responsabilizado por filtrar os pacotes que estão a passar pelo dispositivo. E, por fim, o *chain OUTPUT*, que filtra os pacotes que estão a sair do dispositivo. De forma resumida, filtra os pacotes de resposta do dispositivo.

## 2.6 Sistema de Detecção de Intrusão

Em linhas gerais, IDS constitui-se de um mecanismo de defesa que tem por finalidade capturar e analisar os pacotes trafegados em uma rede de computadores. Após a análise, caso seja detectado alguma anomalia, este tem por objetivo barrar a atividade maliciosa e gerar um histórico de tais atividades para futuras modificações nas políticas de segurança da rede.

Através do histórico, é possível extrair informações que auxiliarão nas mudanças da política de segurança, tais como: tipo de ataque realizado, IPs da máquina atacante e vítima, porta do IP do atacante e vítima, etc. IDS ainda pode ser subdivido em outras duas subcategorias: IDS baseado em host (HIDS) e IDS baseado em rede (NIDS); podendo ainda possuir, ainda, um verificador de integridade de arquivos.

O HIDS tem como por função analisar se existe características de invasão no host. Executam sua análise através de estudos de *logs* do sistema operacional da máquina, nesta atividade utilizam uma grande parte dos recursos do computador.

Verificam se houve comportamentos anormais, como por exemplo, múltiplas tentativas de login, alterações de privilégios, entre outros.

NIDS é constituído por dois componentes, os sensores e estação de gerenciamento. Os sensores são alocados de forma aleatórios na rede, a fim de monitorar o tráfego da rede. Já a estação de gerenciamento, tem como finalidade receber alerta dos sensores, caso algum alerta seja gerado.

Verificador de integridade verifica se ocorreu alguma manipulação dos arquivos desde a última verificação. Para assegurar a integridade dos arquivos, estes utilizam uma chave criptografada para identificar se tal arquivo foi modificado ou não, caso seja, um alerta será enviado para o administrador de rede.

Vitali e Silva [43] destacam que existem duas categorias no qual um Sistema de Detecção de Intrusão pode ser classificado em duas categorias: IDS baseado em assinatura e IDS baseado em anomalias.

A detecção por assinatura apresenta um índice de detecção mais favorável, em relação à detecção por anomalia, no qual este possui o conhecimento das características de uma atividade maliciosa, quais características são pertencentes aos ataques. Porém, apresenta a limitação de conseguir detectar apenas ataques que estão em sua base de conhecimento [12]. Assinaturas são atividades/eventos que podem acontecer na rede.

Já a detecção por anomalia corresponde ao aprendizado convencional da utilização do dispositivo, em que ao longo do tempo é construído um perfil de utilização, caso ocorra alguma variação de uso do dispositivo, um evento de alerta é gerado [12].

## 2.7 Síntese

Neste capítulo, vários conceitos relacionados a Internet das Coisas foram apresentados. Logo a seguir, algumas das ameaças que colocam em risco as redes de computadores e as ameaças enfrentadas pela IoT foram discutidas, principalmente a negação de serviço (DoS).

A Internet das Coisas, é um novo paradigma que tem por finalidade interconectar qualquer tipo de dispositivo que tenha capacidade de conexão com a Internet. Alguns dispositivos utilizados na IoT podem ser considerados “inteligentes”, pois podem cooperar entre si para a realização de alguma atividade em comum.

Na seção de ameaças em redes de computadores foi relatado que as ameças em redes de computadores a cada dia crescem consideravelmente, e as ameças estão sempre se aprimorando, sendo que foi detectado um crescimento de 36% no número de novos casos detectados de *malwares*.

Foi realizado uma explanação sobre os ataques de negação de serviço, relatando as maneiras que estes podem ocorrer. E, por fim, foram relatados os tipos de ameças que envolvem IoT, as motivações que levam os atacantes a realizar tais ataques, e mencionado um grupo de *hackers* que possuem várias *botnets* que ameaçam os dispositivos de IoT.

# 3 Estado da Arte

Este capítulo tem como objetivo relatar os tipos de ameaças que circundam a Internet das Coisas, os sistemas de detecção de intrusão para IoT e as contramedidas utilizadas em IoT.

## 3.1 Ameaças para IoT

Garantir a segurança em meios computacionais não é uma tarefa fácil. IoT não ficou de fora dos problemas de seguranças e ameaças que colocam em risco os ambientes computacionais. Wangham *et al*. [44] destacam que ao garantir aplicações seguras para os dispositivos de IoT, é necessário manter a característica de autonomia e interoperabilidade dos dispositivos.

Como tentativa de sanar as ameaças de IoT, Neto [29] destaca que se deve garantir proteção aos dispositivos IoT contra ataques de DoS/DDoS; prover proteção contra programas maliciosos que tem por objetivo infectar os dispositivos IoT; e garantir que as informações pessoais dos usuários não sejam acessadas de forma indevida.

Babar *et al*. [2] enfatizam os requisitos de segurança de dispositivos IoT que são:

- **Identificação de usuários**: somente usuários legítimos devem ser capazes de utilizar os serviços dos dispositivos de IoT, sendo necessário passar por um processo de validação de usuário;
- **Resistência contra invasores**: embora o risco de um dispositivo ser invadido seja muito grande, estes devem ser capazes de garantir a segurança das informações; mesmo que invasores consigam acesso de forma física ao dispositivo;
- **Ambiente de execução segura**: o bom funcionamento das aplicações deve ser garantido, devendo ser protegidos contra aplicações maliciosas que provocam o mal funcionamento de tais aplicações;

- **Conteúdo seguro**: deve ser garantido a segurança dos conteúdos digitais que estão sendo utilizados pelo sistema;
- **Acesso seguro à rede**: somente dispositivos com autorização devem está conectados à rede, devendo ser tomado alguma contramedida, caso seja identificado que dispositivos não autorizados estejam conectados;
- **Comunicação de dados segura**: deve ser garantido a confidencialidade das informações trafegadas na rede, bem como a sua integridade;
- **Gerenciamento de identidade**: deve ser realizado um gerenciamento do que cada coisa (ou seja, cada dispositivo) pode ou não ter acesso na rede;
- **Armazenamento seguro**: as informações armazenadas no sistema deve estar armazenada de forma segura e confiável.

A Figura 3.1 apresenta os principais requisitos de segurança segundo Neto [29].


**Figura 3.1:** Principais requisistos de segurança para a Internet das Coisas [29]

As ameaças de IoT podem ser classificadas por categorias como a seguir [2]:

- **Ataques físicos**: este tipo de ataque está relacionado com comprometimento físico do *hardware* do dispositivo;
- **Ataques de canais de comunicação**: baseia-se na recuperação de dados dos dispositivos responsáveis pela criptografia das informações, com o intuito de conseguir recuperar a chave utilizada na criptografia dos dados;
- **Ataques de análise de criptografia**: semelhante ao ataque de canais de comunicação, busca capturar a chave de criptografia para conseguir ler as informações capturadas;

- **Ataques de *software***: têm por finalidade explorar falhas nos *softwares* presentes nos dispositivos;

- **Ataques de rede**: como exemplo de ataques de rede são os ataques de DoS e/ou DDoS, ataques de roteamentos, captura de tráfego, etc.

Riahi [35] *et al.* mencionam os três aspectos principais para garantir a segurança ao interagir diretamente com os dispositivos de IoT, ou seja, (*intelligent object*), conforme a Figura 3.2, que são: pessoas, processo e ecossistema tecnológico.

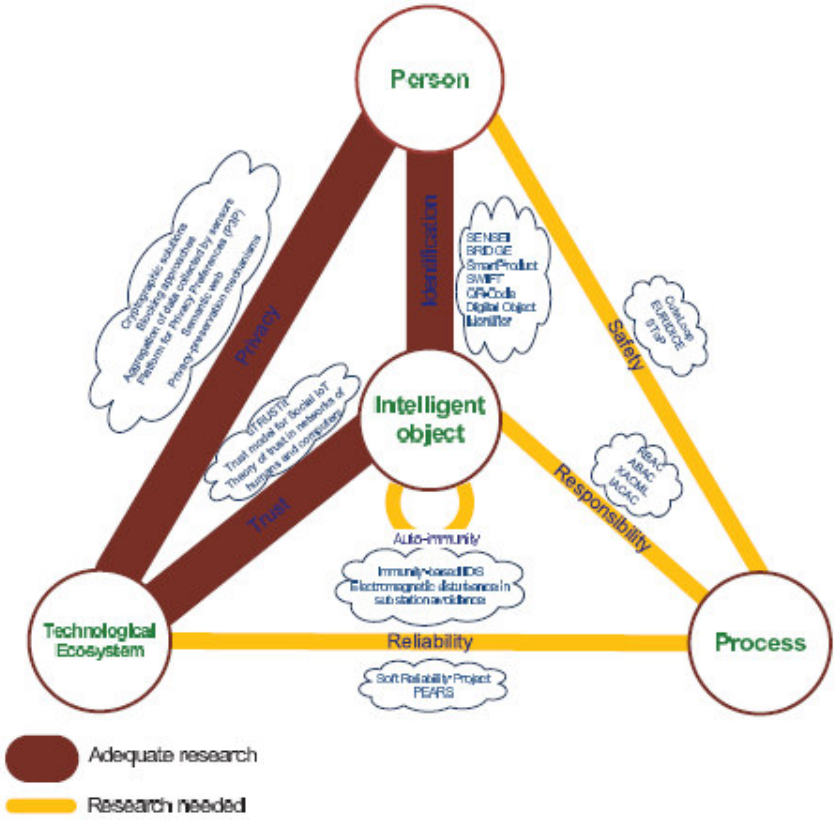



**Figura 3.2:** Uma abordagem sistemática para a segurança da Internet das Coisas [35]

O fator humano, ou seja, as **pessoas** que utilizam os dispositivos, é de suma importância, para que a regras de segurança sejam aplicadas. Estes devem ser capazes de avaliar até que ponto determinada solução está suprindo as necessidades, e até que ponto as soluções existentes são viáveis, devendo sempre buscar a 'evolução' de tais soluções existentes.

Os **processos** são relacionados diretamente com as políticas de segurança, no qual o ambiente em que os dispositivos estão inseridos devem estar em conformidade com tais políticas.

E, por fim, o **ecossistema tecnológico** faz menção às escolhas de quais mecanismos de segurança estarão disponíveis para garantir a segurança dos dispositivos que estão ali presentes.

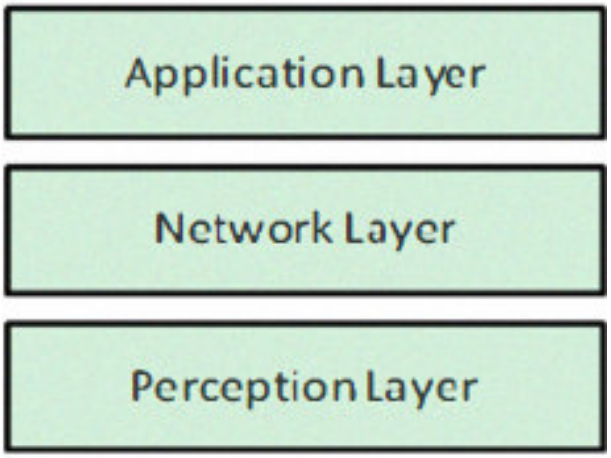



**Figura 3.3:** *3-layer architecture of the Internet of Thing* [46]

Embora a arquitetura de IoT não apresente uma padronização, ou seja, não possui uma arquitetura única a ser seguida, a Figura 3.3 mostra uma arquitetura de IoT apresentada por Wu *et al*. [46], no qual possui três camadas básicas: percepção, rede e aplicação.

**Camada de percepção** possibilita que os sensores, por exemplo, realizem a captura das informações do ambiente. Wu *et al*. [46] alguns dos dispositivos que são utilizados para a captura de informações do ambiente: GPS, etiquetas *Radio-Frequency IDentification* (RFID), sensores de rede, etc.

**Camada de rede** é o "cérebro da IoT", como destacam Wu *et al*. [46]. Esta camada tem por finalidade permitir a intercomunicação entre os dispositivos. Pois é através dela que os dispositivos se conectam à Internet, transportando dados e informações das aplicações.

**Camada de aplicação** compreende as aplicações que são utilizadas pelos dispositivos IoT. A gama de aplicações vai desde aplicações industriais, bem como aplicações de uso cotidiano. Lin *et al*. [23] mencionam que a camada de aplicação proporciona uma interface interativa para os usuários interagirem com suas aplicações.

Puthal *et al*. [33] mencionam os tipos de ameaças para IoT, no qual cada tipo de ameaça está relacionado com as camadas de sua arquitetura, que segue:

- Ataques da **camada de percepção**:

  *Jamming*: ocorre com a emissão de sinal de rádio na mesma frequência de um dispositivo-vítima, com finalidade substituir o sinal legítimo;

  *Timing attack*: tem por finalidade conseguir a chave de criptografia dos dados transmitidos;

  *Routing threats*: o atacante cria rotas adulteradas para o encaminhamento das informações, no qual estas sofrem atrasos na entrega das mensagens.

- Ataques da **camada de rede**:

  *HELLO Flood*: ataca redes sem fios, no qual engana um nó vítima com informações falsas, para convencer que este é a melhor rota, para que as informações transmitidas pelo nó vítima passem pelo nó atacante;

  *Man-in-the-middle*: invasores conseguem acesso à rede, no qual todo o tráfego da rede passa por seus dispositivos, sabendo, assim, de tudo o que acontece na rede;

  *Exploit attack*: explora vulnerabilidades nos dispositivos, a fim de provocar comportamentos anômalos na rede.

- Ataques da **camada de aplicação**:

  *Autenticação*: em um dispositivo podem existir uma variedade de aplicações pertencentes a vários usuários, fazendo, assim, a necessidade do uso mecanismo que autentique os usuários, evitando acesso não autorizado;

  *Disponibilidade de dados*: os dados e informações só devem está disponíveis somente para usuários autenticados, evitando, assim, uso indevido de dados e/ou informações;

  *Proteção e recuperação de dados*: os dados armazenados devem possuir mecanísmos de seguranças, devendo, também, ter a possibilidade de serem recuperados ao ocorrer eventos inesperados.

## 3.2 Sistemas de Detecção de Intrusão para IoT

Visando expandir nossos conhecimentos sobre IoT e IDS, esta sessão irá apresentar alguns trabalhos que já foram desenvolvidos e apresentados para a comunidade científica, no qual será tomado como base quais mecanismo já foram desenvolvidos.

### 3.2.1 *Svelte: Real-time intrusion detection in the internet of things*

SVELTE trata-se de um IDS de detecção em tempo real para IoT, sendo o primeiro IDS implementado para este paradígma [34]. Este sistema tem como principal função a detecção de ataques de falsificação e modificações de informações, *sinkhole* e *selective forward*. Embora os autores destaquem os principais tipos de ataques, estes relatam, ainda, que, SVELTE pode ser estendido e solucionar outros tipos de ataques.

O sistema operacional utilizado para a realizações dos testes, foi o *Contiki OS*[11]. No qual foi implementado um *mini-firewall* para poder realizar a proteção dos tráfegos dos dispositivos.

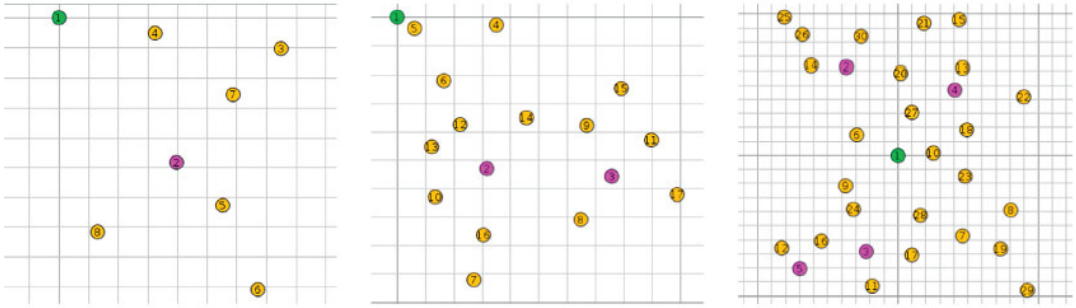
Basicamente, SVELTE é voltado para redes 6LoWPAN. Que consiste de uma rede *wifi*, que utiliza uma versão otimizada do protocolo IPv6, utilizando uma quantidade mínima dos recursos dos dispositivos, a fim de dar uma maior segurança para as informações trafegadas nos dispositivos pertencentes à rede.

Para a detecção dos ataques de roteamento (*sinkhole* e *selective forward*), foi utilizado o protocolo RPL, do próprio sistema operacional. Ao se utilizar o RPL, os nós da rede possuem o conhecimento de todos seus nós descendentes.

RPL é um protocolo utilizado em roteamentos do protocolo IPv6 em *Low-power Lossy Networks* (LLns), em que objetiva redes que possuem dispositivos com poucos recursos e uma grande quantidade de nós que são gerenciados por um nó centralizado [10].

[11] http://www.contiki-os.org/

Quanto aos experimentos, estes foram realizados utilizando um simulador do próprio *Contiki OS*, *Cooja* [12]. A Figura 3.4 mostra os cenários no qual SVELVE foi testado.



(a) 8 nodes with 1 (ID 1) malicious node. (b) 16 nodes with 2 (ID 2 and 3) malicious nodes. (c) 32 nodes with 4 (ID 2, 3, 4, 5) malicious nodes.

**Figura 3.4:** Cenários utilizandos com SVELTE [34]

Em um primeiro momento (a), foram simulados 8 nós, sendo um malicioso, ou seja, responsável por realizar os ataques. Em um segundo momento (b), foram simulados 16 nós, com 2 nós maliciosos. E, por último, foram simulados 32 nós, tendo 4 nós realizando os ataques.

Relata-se, ainda, que, os autores Raza *et al*. [34] demonstram a efetividade do SVELTE nos testes realizados. Ressalva-se que estes não conseguiram 100% de detecção dos casos de tentativa de invasão, pois foram gerados avisos falso-positivos nas análises do nós.

Na Figura 3.5 é demonstrado as porcentagens de alertas verdadeiros-positivos detectados pelo SVELTE, no qual os autores destacam que obtiveram um taxa de 90% de verdadeiros-positivos na detecção de ataques *sinkhole*.

### 3.2.2 *Demo: An ids framework for internet of things empowered by 6lowpan*

Kasinathan *et al*. [19] mencionam que DEMO é um *framework* IDS para IoT, capacitado para trabalhar tanto com o protocolo IPv6, como para 6LoWPAN – o 6LoWPAN nos permite trabalhar com o protocolo IPv6 em dispositivos de baixo poder

[12]Disponível em: http://anrg.usc.edu/contiki/index.php/Cooja_Simulator. Acessado em: 28/12/2016.

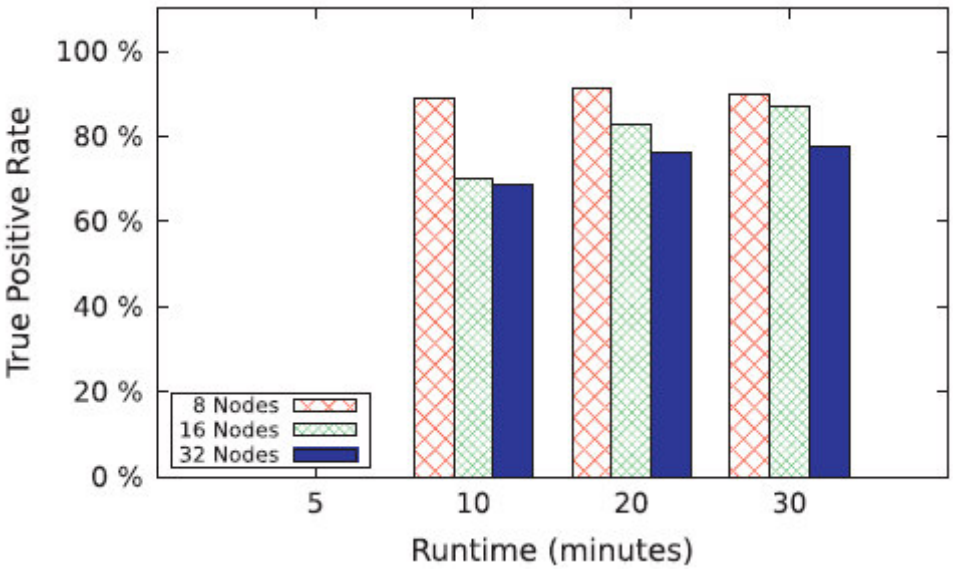



**Figura 3.5:** Porcentagem de verdadeiro-positivo [34]

de processamento e armazenamento. DEMO possui um sistema de monitoramento e um motor de detecção de anomalias que podem ocorrer em algum momento na rede.

Para a realização dos testes, e fazer uma análise do comportamento de DEMO, estes realizaram os testes de penetração utilizando um sistema de PenTest[13] (Scarpy). Foram realizados ataque de DoS, no qual a Figura 3.6 apresenta a arquitetura utilizada.

Na arquitetura utilizada, Scarpy (Sp) está conectado à rede 6LoWPAN, para a realização do ataque DoS, e Suricata (P) (que trata-se de um IDS *open souce*) — para a detecção, Suricata foi modificado com o *framework* DEMO —, responsável por capturar e analisar o tráfego da rede.

Foram inseridos quatro dispositivos (H) conectados à outros dois roteadores (R), no qual estão conectados a um roteador de borda 6LoWPAN (6LBR), que atua como *gateway* de acesso à Internet.

Demo visa a identificação de ataques de *flooding* e *jamming*. Ao ser identificado um dos dois tipos de ataques, é provido a informação do nível de interferência dos canais IEEE 802.15.5, através de uma interface gráfica para o usuário.

Como contribuição do trabalho, os autores destacam que a implementação original de Suricata não foi desenvolvida para trabalhar com os protocolos de

[13]Disponível em: http://www.forbes.com/sites/ericbasu/2013/10/13/what-is-a-penetration-test-and-why-would-i-need-one-for-my-company/. Acessado em: 30/12/2016.

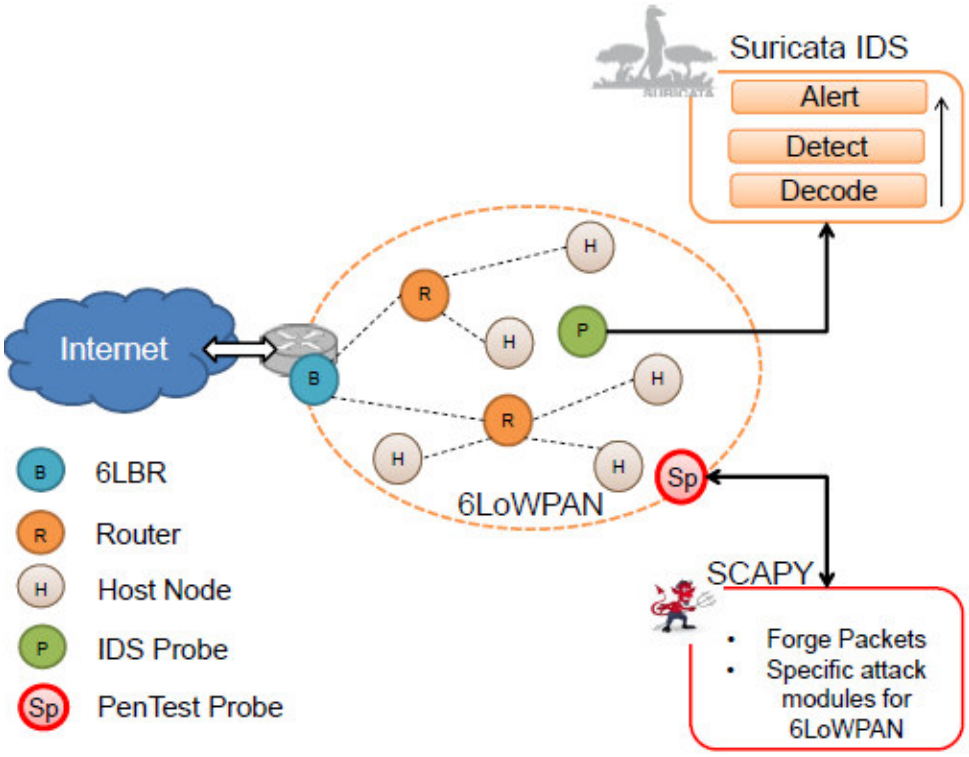



**Figura 3.6:** Arquitetura de proteção DoS [19]

6LoWPAN, no qual se tornou possível após sua modificação com o *framework* desenvolvido por eles.

### 3.2.3 *Distributed Internal Anomaly Detection System for Internet-of-Things*

Thanigaivelan [41] *et al.* propuseram um IDS distribuído para Internet das Coisas, que tem por finalidade a detecção de comportamento anômalos dos dispositivos internos da rede, bem como as características dos comportamento das atividades dos nós vizinhos.

O sistema tem por objetivo aprender as características normais dos dispositivos, obtendo constantemente as informações utilizadas por todos os dispositivos, como tamanho de pacotes e taxa de dados que estão sendo trafegadas em cada nó. Como a abordagem necessita da cooperação de todos os nós da rede, é utilizado uma abordagem distribuída.

A lógica utilizada na organização dos dispositivos é uma estrutura de árvore, ou seja, nós pais recebem relatórios dos comportamentos dos nós filhos.

A distribuição do IDS desenvolvido si dá pela análise comportamental que os nós vizinhos realizam. Em outras palavras, cada nó vizinho irá capturar e analisar o comportamento de seus vizinhos.

Uma vez que seja identificado um comportamento não permitido, o dispositivo terá seus todos seus pacotes bloqueados. Para que seja realizado tal bloqueio, é gerado uma mensagem de bloqueio para o nó pai de tal dispositivo. Para que a mensagem de bloqueio seja replicada até o nó pai, esta é integrada juntamente com o protocolo de roteamento RPL.

O IDS desenvolvido foi projetado para redes 6LoWPAN, em que possui três subsistemas: subsistema de monitoramento e classificação (MGSS), subsistema de reporte (RSS) e subsistema de isolamento (ISS).

Os subsistemas MGSS e RSS operam em camada de rede, em que estes ficam responsáveis pela coleta dos pacotes recebidos pelos nós vizinhos.

MGSS realiza a coleta e análise dos pacotes capturados. Uma vez identificado um comportamento anormal, RSS fica responsável pela geração da mensagem de bloqueio que será transmitida ao nó pai. Em relação ao ISS, este realiza o bloqueio ou liberação dos pacotes dos dispositivos, conforme a análise realizada pelo MGSS.

### 3.2.4 *Design of Complex Event-Processing IDS in Internet of Things*

A variedade de padrões utilizadas pelos dispositivos na IoT, gera uma grande quantidade de dados variados. O processamento de tais dados torna-se complexo e oneroso, devido a grande quantidade da variedade de suas características.

Os autores Jun e Chi [18] desenvolveram um IDS para IoT, no qual fazem uso de Processamento de Eventos Complexos (CEP) para auxiliar no processamento e identificação dos padrões dos dados transmitidos pelos dispositivos.

Com utilização de CEP, é possível o processamento da variedade de eventos que possam ocorrer na rede que não estejam em conformidade com as regras definidas, ou seja, padrões. CEP realiza filtragem e combinação dos eventos para determinar se um evento está em conformidade com as regras estabelecidas [11].

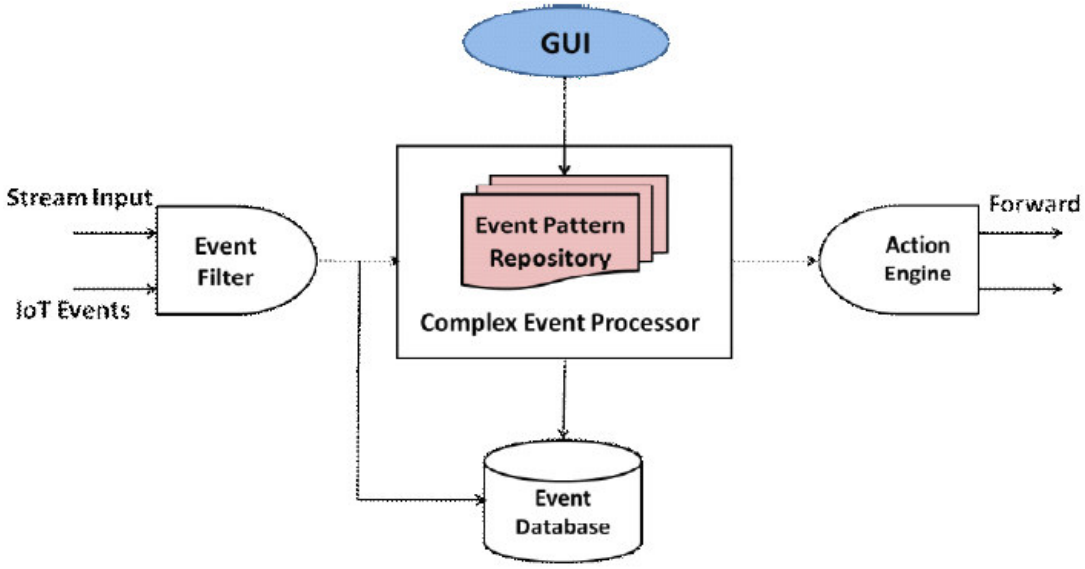



**Figura 3.7:** Arquitetura do IDS baseado em CEP [18]

A Figura 3.7 mostra a arquitetura utilizada pelos autores, que possui alguns componentes, que são:

- ***Event filter***: componente responsável pera captura dos pacotes trafegados na rede. Em algumas situações, os pacotes capturados podem ser duplicados e/ou inconsistentes, precisando, assim, ser filtrados;
- ***Complex event processor***: responsável pela geração dos eventos de EPM (*Event Processing Model)*. EPM trata-se de modelos de eventos já ocorridos, e, após surgir um novo fluxo de dados, estes são comparados, caso sejam correlacionados, um novo evento é criado;
- ***Action engine***: conforme os eventos gerados pelas regras EPM, ações serão tomadas para garantir a segurança dos dispositivos;
- ***Event database***: responsável pelo armazenamento do histórico dos eventos filtrados e resultados dos eventos processados por CEP;

Para que os autores pudessem ter uma noção do comportamento do IDS desenvolvido, estes coletaram informações de utilização de processamento, tempo de processamento e utilização de memória. A Figura 3.8 apresenta os resultados obtidos utilizando duas abordagens: regras CEP e instruções SQL.

Utilizando a mesma quantidade de dados nas duas abordagens, a utilização de CEP demonstrou resultados desfavoráveis em relação ao consumo de taxa de

| Data Type | Data Scale(k) | CPU Utilization (%) | Memory Consumption (MB) | Processing Time (ms) |
|---|---|---|---|---|
| CEP-based IDS | 200 | 48 | 556 | 287 |
| | 400 | 50 | 684 | 368 |
| | 800 | 62 | 730 | 422 |
| Traditional IDS | 200 | 42 | 782 | 477 |
| | 400 | 45 | 964 | 2042 |
| | 800 | 57 | 1064 | 8688 |

**Figura 3.8:** Resultados obtidos

processamento. Porém, em relação à utilização de memória e tempo de processamento apresentou menor utilização de memória e menor tempo para analisar as informações.

### 3.2.5 Outros trabalhos de IDS para IoT

Nesta subseção, será apresentado um *overview* de outros trabalhos que relacionam Sistema de Detecção de Intrusão e Internet das Coisas já apresentados à Comunidade Científica.

#### 3.2.5.1 *Detection of Sinkhole Attacks for Supporting Secure Routing on 6LoWPAN for Internet of Things*

Cervantes *et al*. [8] descrevem que as constantes ameaças para IoT são crescentes, e um dos mais frequentes ataques é o *sinkhole* – este tipo de ataque impede a comunicação entre os dispositivos conectados em uma rede de computadores. Os mencionados autores relatam, ainda, que algumas das alternativas de proteção para dispositivos inteligentes em IoT não oferecem soluções eficazes. Com isso, em seu trabalho eles propõem um IDS para IoT denominado de INTI. O INTI tem a função de detectar ataques de *sinkhole* nos dispositivos constituintes em uma rede 6LoWPAN.

#### 3.2.5.2 *Real Time Intrusion and Wormhole Attack Detection in Internet of Things*

Pongle *et al*. [32] mencionam que a quantidade de dispositivos conectados à Internet chegou a ultrapassar a quantidade de pessoas no mundo todo. Com este

crescimento os dispositivos tendem a ganhar cada vez mais novas funcionalidades e com isso, contribuindo para a evolução da Internet das Coisas.

Com o advento deste crescimento, faz-se necessário o constante surgimento de novas ferramentas que auxiliem na prevenção de ataques em IoT. Os referidos autores desenvolveram um sistema de detecção de intrusão para Internet das Coisas, em que esta ferramenta tem como objetivo a detecção de ataques *wormhole*.

## 3.3 Contramedidas para preservação da IoT

Garantir a segurança de dispositivos é um fator primordial para o bom funcionamento de qualquer sistema e/ou dispositivo. Embora o paradigma de IoT seja relativamente novo, garantir a segurança de tais dispositivos é de suma importância.

Stankovic [38] destaca que garantir a segurança dos dispositivos de IoT é um fator primordial, porem a garantia de tal segurança, bem como a detecção dos ataques e aplicação de contramedidas, está relacionada diretamente com a pouca disponibilidade dos recursos que estes possuem.

Ning *et al*. [30] classificam em quatro categorias as ameaças, e suas contramedidas, que os dispositivos de IoT estão expostos, no qual tais ameaças vão interferir na confiabilidade e disponibilidade dos dados. As categorias são [30]:

- Ao **coletar** dados dos pacotes, o atacante pode realizar quatro tipos de ataques: *Skimming*, *Tampering*, *Eavesdropping* e *Traffic analysis*. As contramedidas para tais ameaças vão desde a utilização de criptografia, uso de funções *hash*, até à análise de comportamento anormal de dispositivos na rede;

- O termo **imitação** relaciona-se com clonagem de informações, dos pacotes em questão, para ganhar acesso a um sistema, por exemplo. Este tipo de ameaça pode ser realizado através dos ataques *Spoofing*, *Cloning* e *Replay*. As contramedidas para tais ameaças vão desde a utilização de assinaturas digitais à utilização de chaves seriais;

- Ataques de **bloqueio** têm por finalidade deixar um sistema inacessível para dispositivos clientes. Este tipo de ameaça pode ser realizado através dos ataques *Denial of service*, *Jamming* e *Malware*. As contramedidas para essas ameaças estão

relacionadas com a utilização de antivírus, *Firewall* e sistemas de detecção de intrusão;

- Os ataques de **privacidade** podem tanto ser voltados para um indivíduo específico, bem como para um grupo alvo. Como contramedidas, pode-se utilizar os conceitos de agregação de dados anônimos (CDA) ou utilizar divulgação seletiva, no qual é selecionado quais dados estarão públicos.

Kasinathan *et al*. [20], menciona algumas das contramedidas utilizadas para garantir a segurança dos dispositivos em IoT, que são: *Secure Bootstrapping*, Segurança da Camada de Aplicação e utilização de Sistemas de Detecção de Intrusão.

***Secure Bootstrapping*** tem por finalidade garantir que apenas dispositivos autorizados possam acessar a rede e seus recursos, no qual cada dispositivo deva ser identificado com uma chave única.

A **Segurança da Camada de Aplicação** deve ter uma atenção especial quanto ao quesito segurança. Utilizar o protocolo Transport Layer Security (TLS) é de máxima importância para garantir a segurança dos dados trafegados pelos dispositivos, no qual garante a privacidade e integridade dos dados.

**Sistemas de Detecção de Intrusão** demonstram-se bastante úteis para a detecção de anomalias, no qual podem ser utilizados como um importante mecanismo de medida preventiva em redes computacionais, incluindo a IoT.

## 3.4 Síntese

Neste capítulo, as ameaças que circulam a Internet das Coisas foram abordadas, mencionando os principais requisitos de segurança para IoT, relacionando os tipos de ataques conforme as camadas da arquitetura de IoT.

Em seguida, os Sistemas de Detecção de Intrusão voltados para a Internet das Coisas foram destacados, relatando qual ameaças cada IDS visa combater. E, por fim, as contramedidas necessárias para garantir a segurança dos dispositivos de IoT foram apresentadas.

# 4 Proposta de um IDS para IoT

Este capítulo tem como objetivo apresentar o Sistema de Detecção de Intrusão para Internet das Coisas, IDS-IoT, que, por sua vez, tem a finalidade de barrar alguns dos ataques de Negação de Serviço (DoS). A arquitetura do IDS-IoT será apresentada bem como alguns diagramas UML e o algoritmo utilizado para a detecção dos ataques.

Para a realização da captura dos pacotes que estão na rede, a biblioteca *open source* Jpcap foi utilizada. O Jpcap pode ser tanto utilizado para a captura de pacotes, como para o envio de pacotes[14]. O banco de dados utilizado foi o MySQL, versão 14.14 e distribuição 5.5.50.

O banco de dados encontra-se localizado no mesmo dispositivo no qual o IDS-IoT está executando, no qual foi identificado um menor tempo gasto para inserir as características dos pacotes capturados na tabela *pacote*.

## 4.1 Arquitetura do IDS-IoT

A arquitetura do IDS-IoT consiste em uma arquitetura em camadas, em que estão organizadas de forma *top-down*; ou seja, as camadas superiores oferecem dados/informações para as camadas inferiores.

Como ressalva, as camadas inferiores são dependentes das camadas superiores, uma vez que estas necessitam das informações providas pelas camadas superiores para poderem realizar suas respectivas funcionalidades.

Na Figura 4.1 pode ser visualizado a arquitetura do IDS-IoT. A arquitetura utilizada possui os seguintes camadas: *Sensor*, *Analisador_Pacote*, *Deteccao_Ataque*, *Geracao_Regra*, que tem como saída as regras do *Firewall* geradas.

[14]http://jpcap.gitspot.com

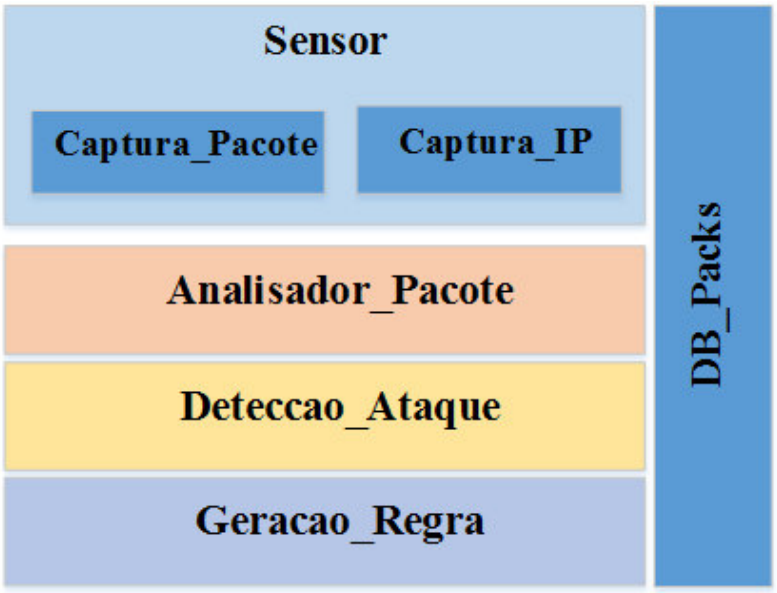



**Figura 4.1:** Arquitetura utilizada pelo IDS-IoT

A camada *Sensor* está responsável tanto pela captura dos pacotes, que estão sendo trafegado na rede, utilizando o módulo *Captura_Pacote*, bem como pela captura dos IPs ativos na rede, utilizando o módulo *Captura_IP*, e inserindo-os no *DB_Packs*.

*Analisador_Pacote* recupera as informações inseridas no *DB_Packs* pela camada *Sensor* e realiza a análise das características dos pacotes capturados. A camada *Deteccao_Ataque* fica responsável por determinar qual tipo de ataque está a acontecer. E, por fim, a camada *Geracao_Regra* tem por finalidade gerar a regra de bloqueio do *Firewall Iptables*.

A Figura 4.2 apresenta o fluxograma de execução do IDS-IoT, para uma melhor compreensão de seu funcionamento.

Ao iniciar o IDS-IoT, este irá utilizar a ferramenta nmap[15] para **capturar os IPs** pertencentes à rede que estão ativos. Após a identificação de quais IPs estão ativos, estes serão inseridos na tabela *ip_ativo* do banco de dados *DB_Packs*.

A **captura dos pacotes** é realizada pela camada *Sensor*, responsável por capturar informações da rede. Possui ainda dois módulos, *Captura_Pacote* e *Captura_IP*. O módulo *Captura_IP* tem como função capturar os *Internet Protocol* (IP) dos dispositivos que estão ativos na rede.

[15]Disponível em: https://nmap.org. Acessado em: 20/11/2016.

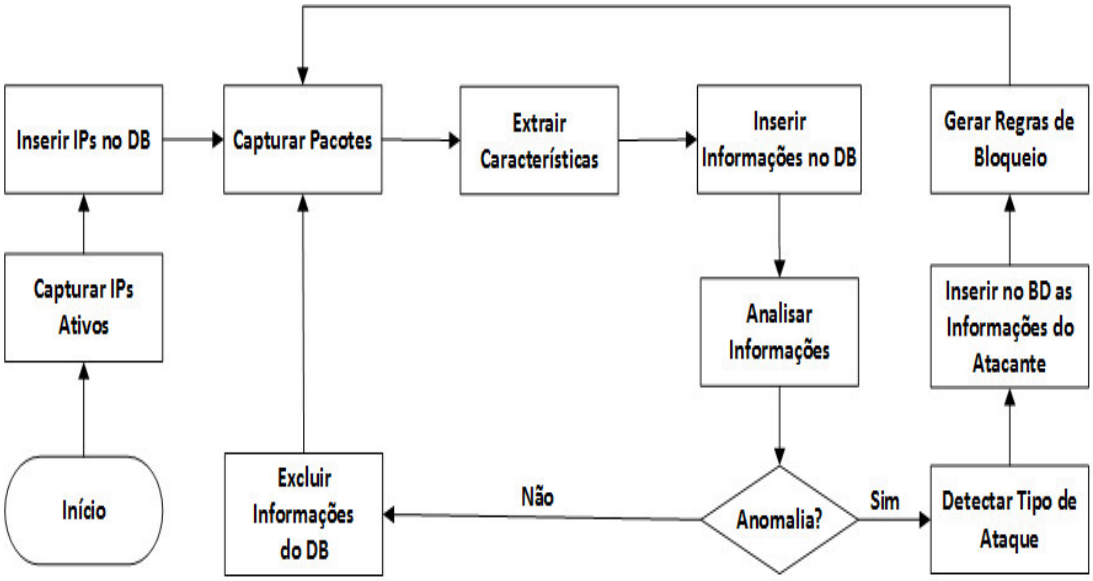



**Figura 4.2:** Fluxograma de Execução

Após a **extração das características**, dos pacotes capturados, são inseridas na tabela *pacote* do *DB_Packs*. Tais características são utilizada pelo *Analisador_Pacote*, para realizar a **análise das características** e determinar se o tráfego é de dispositivos autorizados ou não. As características extraídas dos pacotes são: IP de origem e destino, tamanho do pacote, são exemplo de características extraídas.

Após ser realizada a análise, é verificado se seu resultado é característico dos ataques realizados. Caso a análise não detecte nenhuma características de anomalias, é realizada a **exclusão dos registros** das características capturas. Caso contrário, é a lizada a identificação do ataque conforme às assinaturas que as ameaças possuem.

Ao ser identificado que as características capturadas são de ameaças, é identificada qual tipo de ataque as informações são pertencentes, pois cada tipo de ameaça possui suas características particulares. Uma vez **detectado o tipo de ataque**, as **informações do atacante** são inseridas na tabela *historico* do banco de dados, no qual as regras *Iptables* — ou seja, as **regras de bloqueio** — serão geradas para barrar a atividade maliciosa.

Após a finalização deste ciclo, a captura dos pacotes é retomada, sendo que todos os novos pacotes capturados serão analisados pela camada *Analisador_Pacote*.

## 4.2 Modelagem

Alguns diagrama UML foram criados para modelar o IDS-IoT. Diagrama de Classe, Caso de Uso, Sequência, Atividade e Implantação do IDS-IoT são apresentados a seguir.

### 4.2.1 Diagrama de Caso de Uso do IDS-IoT

Esta subseção, tem por finalidade apresentar o diagrama de caso de uso correspondente à utilização do IDS-IoT, no qual é exposto as funcionalidades da aplicação desenvolvida. A Figura 4.3 corresponde ao diagrama de caso de uso da aplicação.

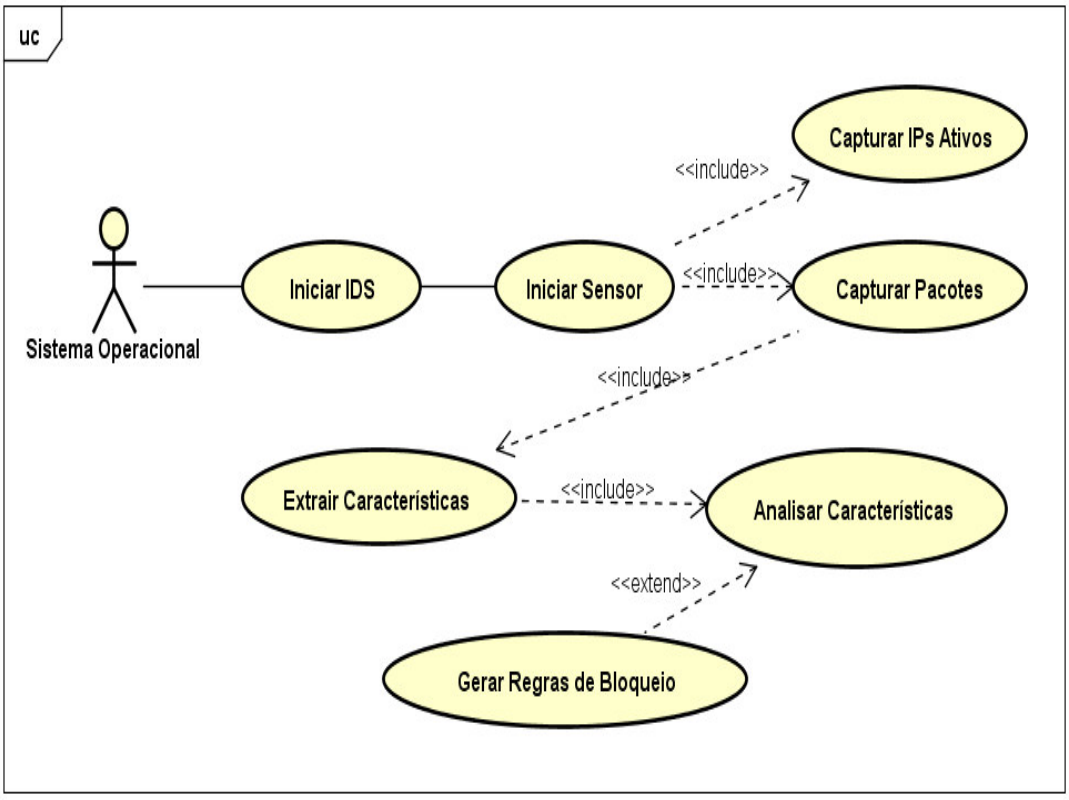



**Figura 4.3:** Diagrama de Caso de Uso

Após a inicialização da aplicação, é dado inicio a fase captura de IPs ativos pelo *Sensor*, em que após a identificação de quais IPs da rede estão ativos para que estes sejam inseridos no banco de dados.

Após a captura e inserção no banco de dados dos IPs ativos, é iniciada a captura dos pacotes que estão sendo recebidos e transmitidos pelos dispositivos

ativos na rede; depois de sua captura, estes são inseridos no banco de dados para que sejam analisados posteriormente. Para que seja identificado algum comportamento anormal, são extraídos dos pacotes capturados suas características para que estes sejam analisados pela fase de análise.

A fase de análise começa após a extração das características dos pacotes, no qual é analisado de tais características são correlatas aos comportamentos das ameaças implementadas. Uma vez detectado tal relacionamento, a fase de geração de regras *iptables* se inicia, no qual as regras são geradas para o *Firewall iptables*.

### 4.2.2 Diagrama de Classe do IDS-IoT

As classes, métodos e atributos do IDS-IoT podem ser visualizados no diagrama de classes da Figura 4.4. O IDS-IoT possui 5 classes: *Sensor*, *AnalisadorPacote*, *DetectaAtaque* e *GeraRegra* e *Conexao*. Em que a classe *Sensor* é a classe principal e está responsável por inicializar a captura dos pacotes e dos IPs ativos na rede. A classe *Sensor* possui 4 métodos que são: *Sensor()*, *main*, *ipAtivo()* e *captura()*.

A Figura 4.4 apresenta o diagrama de classe utilizado para a realização da implementação, juntamente com seus métodos e atributos. Descrevendo cada um dos métodos que compõem a classe Sensor, temos a seguinte descrição:

- **Método *Sensor()***: um construtor para a realização da conexão com o banco de dados, no qual é utilizada a classe *Conexao*, e possibilitar que os pacotes e IPs ativos capturados sejam inseridos nas tabelas *pacote* e *ip_ativo*, respectivamente;
- **Método *main()***: é o método principal, responsável por inicializar toda a execução do IDS-IoT;
- **Método *ipAtivos()***: executa um comando[16] nmap e verifica quais os IP estão ativos, para que estes sejam inseridos na tabela *ip_ativo*;
- **Método *captura()***: este método utiliza a biblioteca Jpcap para detectar e captura se existem pacotes trafegando na rede. Ao ser detectado a existência de pacotes na rede, será determinado qual tipo de protocolo estes correspondem (TPC, ICMP ou UDP). Após ser detectado, serão inseridos na tabela *pacote* algumas de

[16] *nmap -v -sn -oG - 192.168.42.0/24 | grep 'Up'*

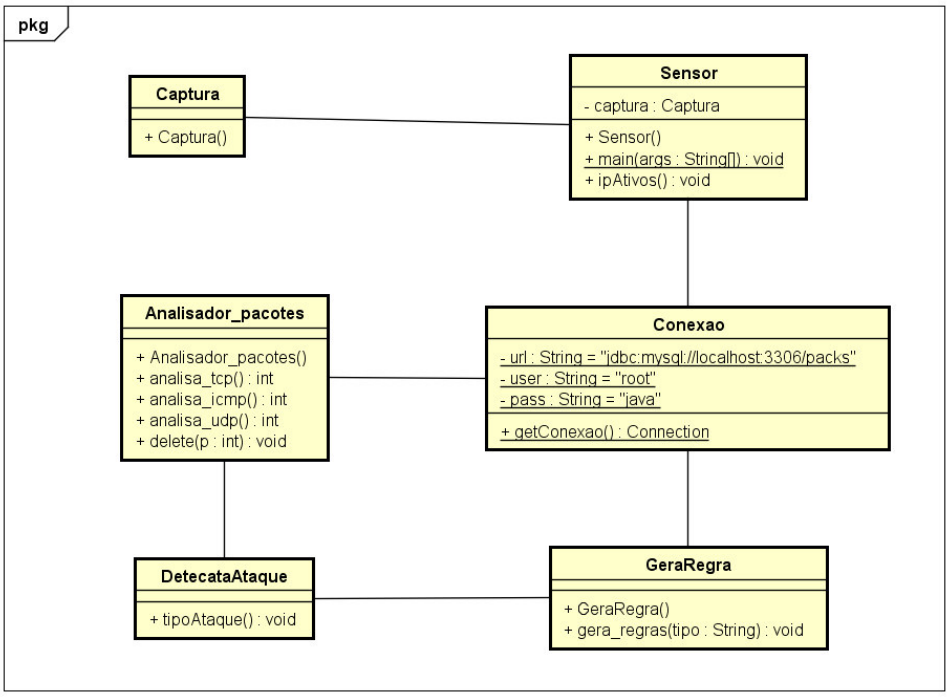



**Figura 4.4:** Diagrama de Classe

suas características, como: endereço IP e porta do emissor, endereço IP e porta de destino, tamanho do pacote, quantidade de pacotes de um mesmo emissor, *flags* TCP (*syn*, *ack* e *fin*) de pacotes TCP e a data e hora de recebimento do pacote.

Em relação à classe *AnalisadorPacote*, este possui alguns métodos, que serão descritos a seguir:

- **Método *AnalisadorPacote()***: igualmente ao construtor da classe Sensor, este método tem como objetivo realizar a conexão com o banco de dados. A conexão com o banco se faz necessário pela necessidade de ter acesso aos registros da tabela *pacote*;
- **Método *analisa_tcp()***: tem por finalidade analisar as características dos pacotes que correspondem ao protocolo TCP;
- **Método *analisa_icmp()***: tem por finalidade analisar as características dos pacotes que correspondem ao protocolo ICMP;
- **Método *analisa_udp()***: tem por finalidade analisar as características dos pacotes que correspondem ao protocolo UDP;

- **Método *delete()***: este método fica responsável por limpar os registros da tabela *pacote* após a análise ser realizada.

Após a finalização da análise dos pacotes, a classe *DetectaAtaque* tem por finalidade identificar qual tipo de ataque está a acontecer. No qual as informações do ataque devem ser inseridas na tabela *historico* do banco de dados.

A classe *GeraRegra* tem por finalidade gerar as regras *iptables* para a realização do bloqueio do ataque, ou seja, bloquear os pacotes oriundos dos IPs atacantes. Esta classe possue duas classe, como segue:

- **Método *GeraRegra()***: um construtor para a realização da conexão com o banco de dados, para o método *gera_regras()* ter acesso à tabela *historico*;
- **Método *gera_regras()***: utiliza as informações inseridas na tabela *historico* do ataque detectado para a geração da regra *iptables* correspondente ao tipo de ataque.

### 4.2.3 Diagrama de Sequência do IDS-IoT

A sequência das interação que ocorrem no processo de execução do IDS-IoT serão demonstrados na Figura 4.5. As interações irão desde o inicio da execução do IDS-IoT, até o processo de geração das regras de bloqueio, caso seja detectado uma atividade maliciosa.

Ao inciar o IDS-IoT, o Sensor é ativado para capturar os IPs que estão ativos naquele dado momento. Ao ser executado o comando para a identificação dos IPs, estes são enviados para o banco de dados pare sejam registrados. Após serem registrados, é enviado uma mensagem para o Sensor com a confirmação do registro.

Finalizado o processo de identificar quais IPs estão ativos, o Sensor inicia o processo de captura de pacotes para que estes sejam analisados. Porém, antes da ocorrência da análise, é feita a extração de suas características e armazenadas no banco de dados, no qual o Sensor recebe uma confirmação da inserção no banco de dados. Tais avisos são úteis para a identificação de falhas, caso venham ocorrer.

O Analisador se comunica com o banco de dados para selecionar as características extraídas pelo Sensor. Após a seleção dos registros, contidas no

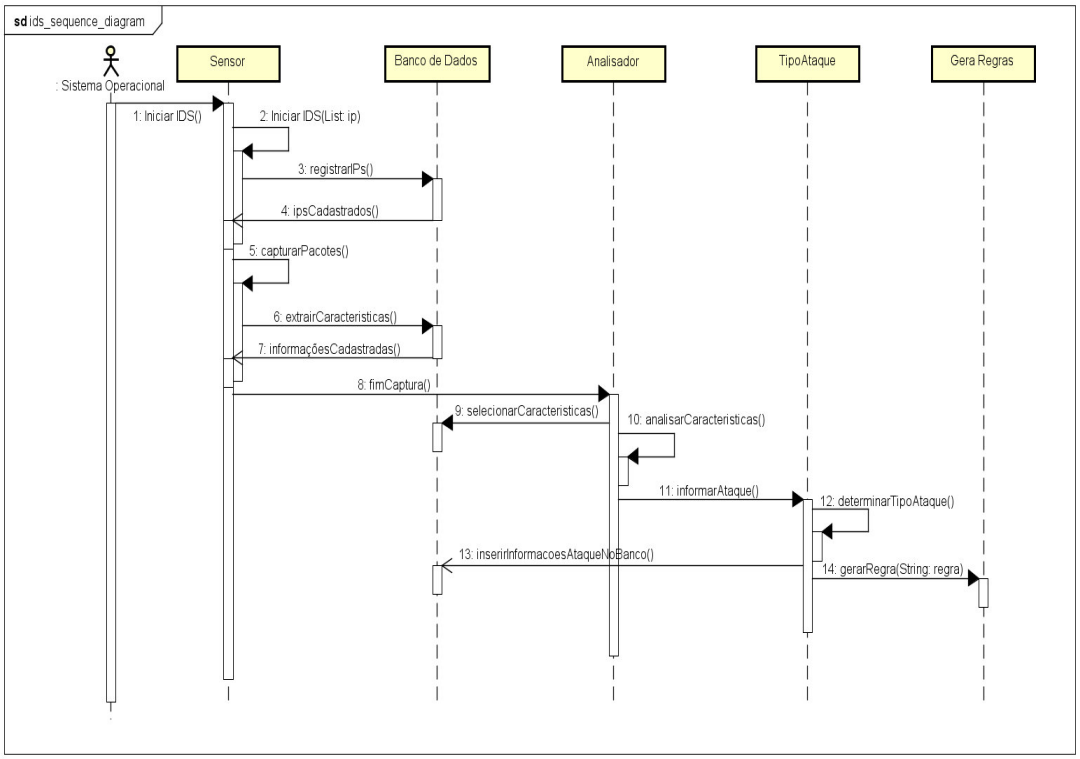



**Figura 4.5:** Diagrama de Sequência

banco de dados, é feita a análise de todas as características. Caso seja detectado a correspondência com algum tipo de ataque, a classe *DetectaAtaque* é informada e esta fica responsável por determinar qual tipo de ataque está ocorrendo.

Ao ser detectado o tipo de ataque, é enviado uma mensagem para classe *GeraRegra* para a geração da(s) regra(s) de bloqueio(s). Ao ser gerado a regra, a classe *DetectaAtaque* recebe uma mensagem de confirmação da regra *iptables* gerada.

### 4.2.4 Diagrama de Atividades do IDS-IoT

Nesta subseção, alguns dos diagramas de atividades do IDS-IoT serão apresentados. Na figura 4.6 é apresentado o diagrama de funcionamento de toda a execução do IDS-IoT. As Figuras 4.7 e 4.8 apresentam os diagramas para a captura dos IPs ativos e captura de pacotes, respectivamente.

A Figura 4.7 mostra as atividades que são realizadas para a realização da captura dos IPs ativos na rede, no qual tal detecção é realizada executando um arquivo *Shell scrip* que possui um comando Nmap que irá checar todos os IPs pertencentes à rede 192.168.42.0/24 que se encontram ativos. Em seguida, os IPs identificados ativos

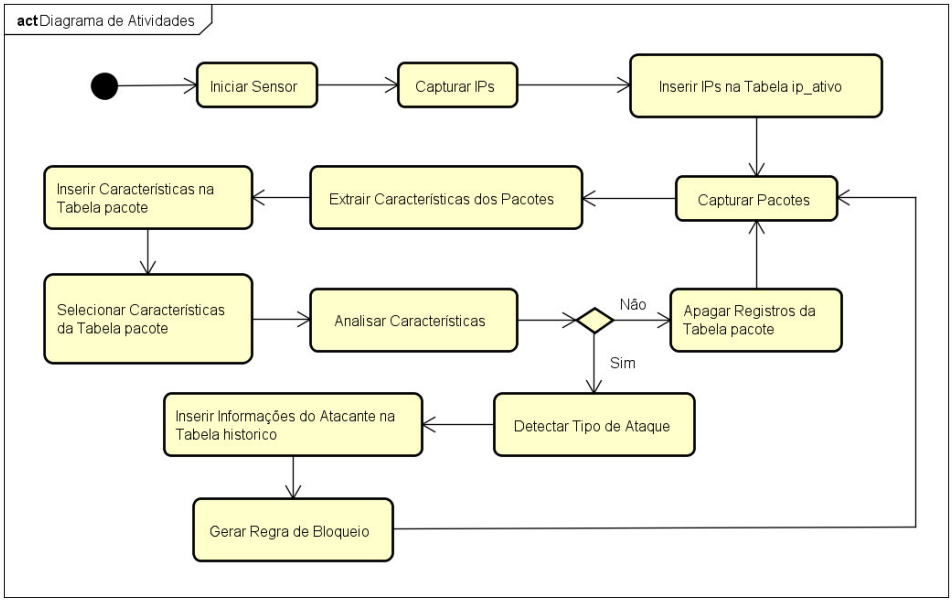



**Figura 4.6:** Diagrama de Atividades Gerais do Sistema

serão inseridos na tabela *ip_ativo* do banco de dados. Após a inserção no banco, a tarefa de verificação de IP ativo é finalizada dando inicio à captura de pacotes.

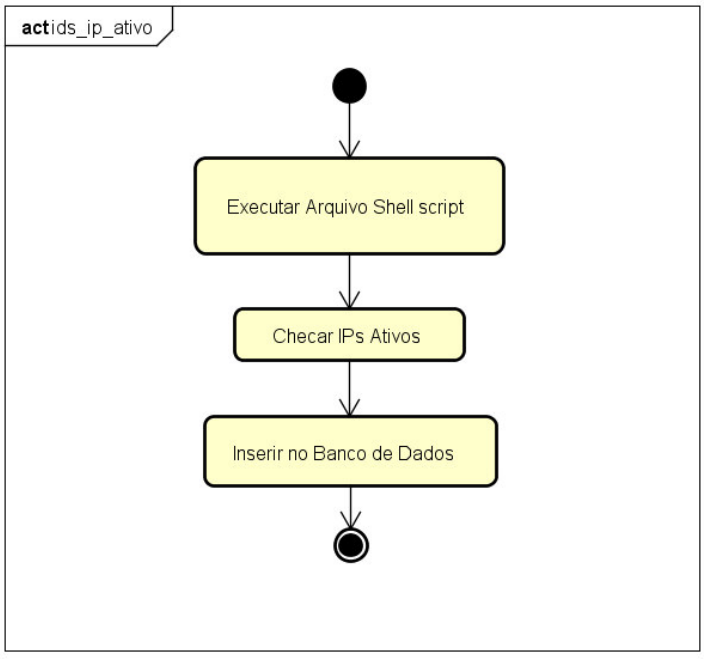



**Figura 4.7:** Diagrama de Atividades (Captura de IPs Ativos)

A Figura 4.8 apresenta as etapas para inicializar a captura dos pacotes que estão sendo recebidos e enviados por todos os dispositivos pertencentes à rede. No qual ao ser iniciado a captura dos pacotes, é identificado a qual tipo de protocolo (TCP, ICMP ou UDP) os pacotes são pertencentes. Ao ser identificado o protocolo,

começa a fase de extração de características para que estes sejam analisados. Esta fase é encerrada para dar inicio à fase de análise.

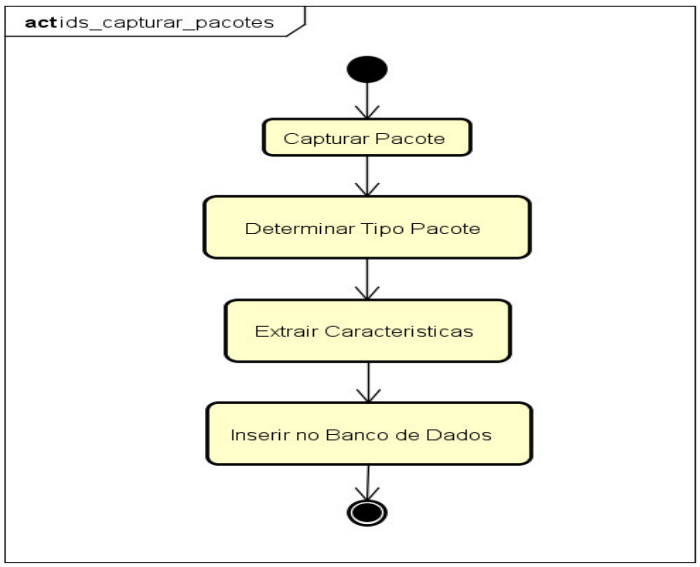



**Figura 4.8:** Diagrama de Atividades (Captura de Pacotes)

A Figura 4.9 apresenta o diagrama com as etapas de análise dos pacotes. No qual esta etapa de execução se inicia com a seleção das características existentes na tabela *pacote* do banco de dados. Após a seleção é verificado se as características são pertencentes aos tipos de ataques implementados. Caso não corresponda a nenhuma ameaça, os registros das características são excluídos, finalizando a etapa de análise. Caso seja detectado um ataque, é dado inicio para fase de identificação de ataque, no qual as informações do atacante é inserido no banco de dados, na tabela *historico*.

A Figura 4.10 apresenta a ultima etapa da execução do IDS-IoT, que é dependente das etapas de análise do pacotes e identificação do tipo do ataque, acarretando na inserção das informações do atacante na tabela *historico*, pois as regras geradas necessitam de tais informações. Após a finalização da etapa de geração de regras, é tomada a captura dos pacotes.

### 4.2.5 Diagrama de Implantação do IDS-IoT

Por fim, esta subseção apresentará o ultimo diagrama utilizado, o Diagrama de Implantação, no qual tem por finalidade expor as tecnologias utilizadas para o desenvolvimento da aplicação proposta. A Figura 4.11 apresenta o diagrama mencionado.

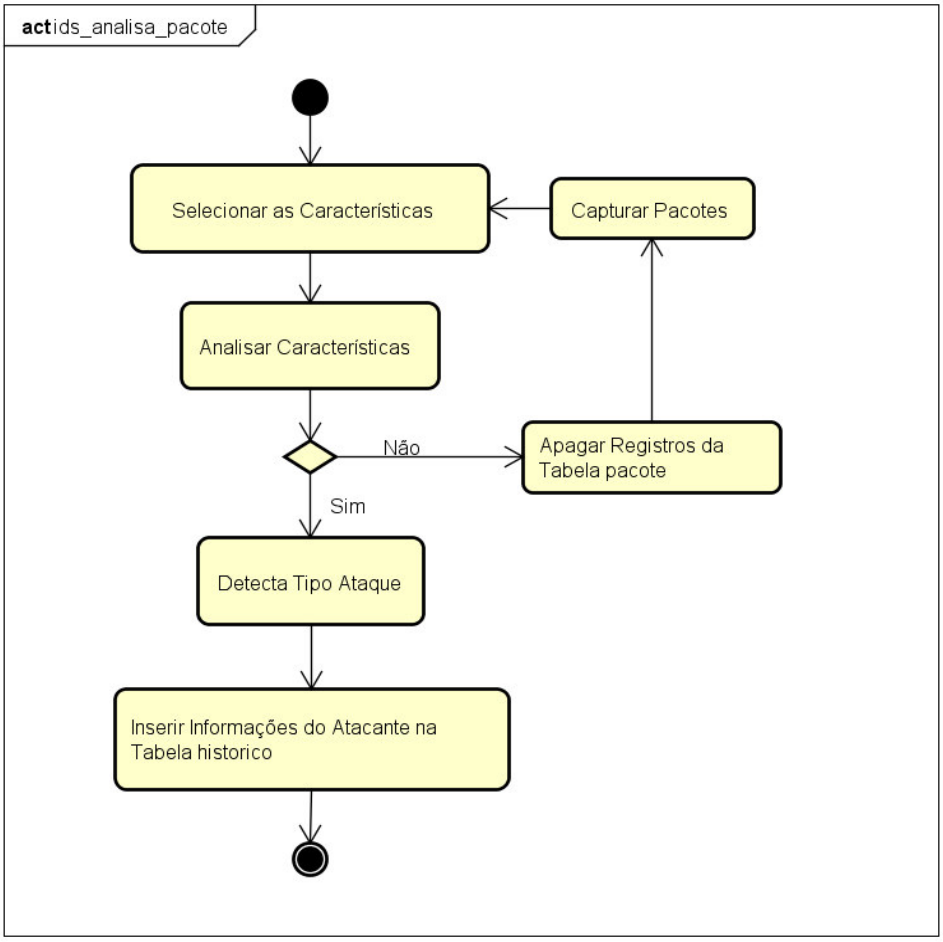



**Figura 4.9:** Diagrama de Atividades (Análise dos Pacotes)

O *hardware* utilizado foi o dispositivo *Raspberry Pi* 3 modelo B, no qual possui as seguintes configurações: 1.2Ghz 64-bit *quad-core* ARMv8 CPU, 10/100Mbps *Lan Speed*, 802.11n *Wireless LAN*. O IDS-IoT foi desenvolvido no sistema operacional *Raspbian*[17].

Para a realização da captura dos IPs e pacotes na rede, o IDS-IoT é dependente de outras duas tecnologias: Nmap e a biblioteca Jpcap. Para a execução do Jpcap em sistemas operacionais de distribuição Linux, este necessita de um arquivo de configuração com a extensão ".so".

Este arquivo de configuração necessita ser recompilado, para que este seja compatível com a arquitetura do processador do dispositivo em questão; o dispositivo utilizado possui a arquitetura do processador ARMv8.

[17]Disponível em: https://www.raspberrypi.org/downloads/raspbian/. Acessado em: 01/12/2017.

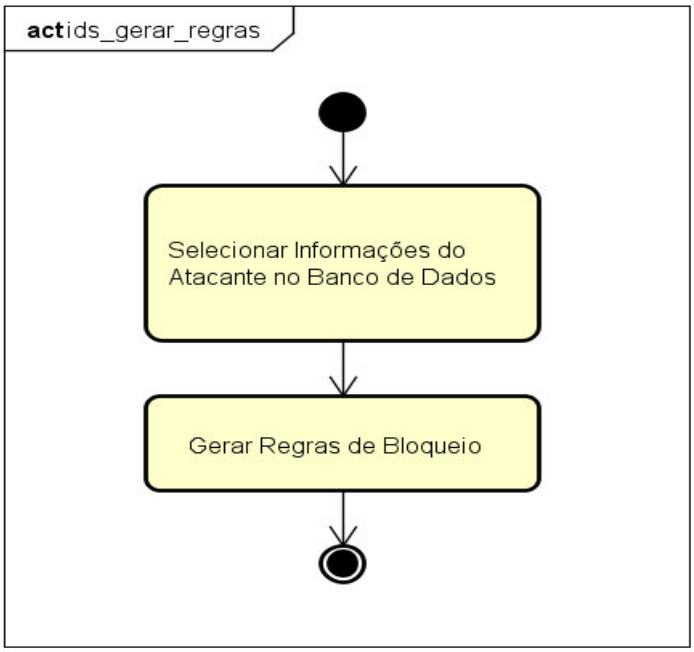



**Figura 4.10:** Diagrama de Atividades (Geração de Regras)

O IDS-IoT depende, ainda, do *Firewall iptables* para a realização do bloqueio de pacotes oriundos de atacantes. Para que ocorra a geração das regras, é feita uma conexão com o banco de dados, utilizando o protocolo TCP, para que seja identificado as informações do atacante — pois as regras são geradas conforme as informações extraídas do ataque.

## 4.3 Algoritmo Utilizado para Detecção de Negação de Serviço do IDS-IoT

Esta subseção, irá apresentar alguns dos pseudo códigos utilizados na implementação do IDS-IoT, no qual a Figura 4.12 relaciona-se com a execução geral do sistema desenvolvido.

Inicialmente, as variáveis são definidas. A variável *verdadeiro* é do tipo *boolean*, com atribuição *true* — para que a execução entre em *loop* de execução sem condição de parada. A variável *tempo* é do tipo *Timer*, que estipula o tempo para a realização da captura dos pacotes. Inicialmente *tempo* é definido com o valor 0 (segundos).

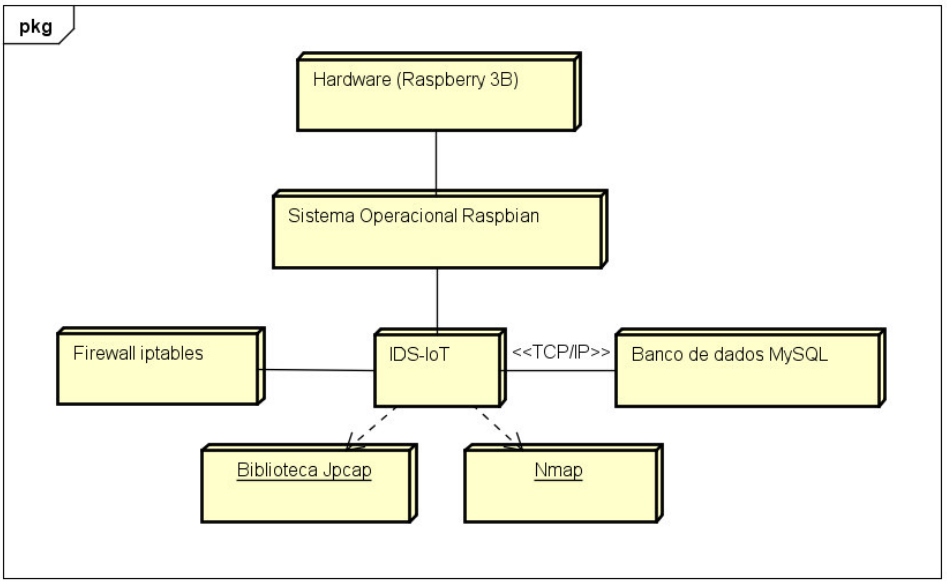



**Figura 4.11:** Diagrama de Implantação da Aplicação Desenvolvida

Ao ser inicializado a fase de análise dos pacotes capturados, é verificado a quantidade de pacotes que são pertencentes a um mesmo emissor com destino a um mesmo receptor. De fato, é de suma importância esta verificação, pois o objetivo da realização de um ataque de negação de serviço é enviar um quantidade de pacotes de forma desnecessária a uma vítima, realizando assim, um ataque de *flood*. A Figura 4.13 encontra-se a o processo de análise dos pacotes capturados.

Caso seja detectada que a quantidade de pacotes é maior que o limite estabelecido, é iniciado o processo de análise de características de pacotes — extraídas na fase de captura —, para que seja capaz de identificar a correspondência com o tipo de ataque que o dispositivo está recebendo.

Após a identificação de qual tipo de ataque está sendo realizado, as informações do atacante, bem como o tipo de ataque, são inseridos na tabela *historico* do banco de dados. Finalizada a fase de identificar as informações do atacante, inicia-se a fase de geração das regras de bloqueio.

A Figura 4.14 apresenta o pseudo código para a captura e identificação de dos pacotes conforme o tipo de protocolo correspondente a cada pacote. Cada tipo de ameaça possui o protocolo correspondente para a realização dos ataques de DoS.

```
1  Variáveis:
   boolean: verdadeiro = true;
2  Timer: tempo = 0; //tempo definido em segundos
   begin
3      Iniciar Sensor
       while verdadeiro do
4          //captura IPs ativos;
5          while tempo ≤ 10 do
6              //capturar pacotes
               //inserir pacotes na tabela pacote
7          end
8          // realizar análise de pacotes
9      end
10 end
```

**Figura 4.12:** Pseudo Algoritmo de Execução do IDS-IoT

## 4.4 Prototipagem do IDS-IoT

A fim de avaliar a proposta no IDS exposta, foi desenvolvido um protótipo para que se possa ter uma maior compreensão de como este irá se comportar em uma aplicação real. No Capitulo 5 será exposto os resultados obtidos ao ser utilizado o IDS-IoT como Sistema de Detecção de Intrusão em um *Raspberry Pi*.

A Figura 4.15 mostra as mensagens exibidas no processo de execução do IDS-IoT, no qual apresenta o inicio da etapa de captura de IPs, sendo exibido o tempo em milissegundos gastos para a execução do comando *nmap*.

Após a finalização da captura, é exibida a mensagem de início da captura de pacotes, no qual é aguardado o tempo definido de captura encerrar-se para dar inicio à fase de análise. São realizadas três análise: análise de pacotes TCP, ICMP e UDP. É exibido, ainda, o tempo gasto para a análise de cada tipo de pacotes.

```
Entrada: pacotes capturados
Saída: definir a ocorrência de ataques DoS
1  Variáveis:
   int: num_packs = 0;
2  int: max_pacotes = 600;
3  begin
4      Iniciar Analisador;
5      //verificar a quantidade de pacotes de um mesmo IP emissor
       if num_packs ≥ max_pacotes then
6          //verificar características dos pacotes
           //classificar tipo de ataque
           //inserir informações atacante na tabela historico;
7          //gerar regras iptables
8      end
9      else
10         //excluir registros da tabela pacote
11     end
12 end
```

**Figura 4.13:** Pseudo Algoritmo de Detecção de Ataque DoS

```
   Entrada: pacotes trafegando na rede
1  Variáveis:
   boolean: verdadeiro = true;
2  Packet: pacote = null;
3  begin
4      while verdadeiro do
5          if pacote == TCP then
6              /*inserir no banco de dados as informações: IP origem,
               porta origem, IP destino, porta destino, tipo pacote,
               tamanho, syn, ack, fin, data */
7          end
8          if pacote == UDP then
9              /*inserir no banco de dados as informações: IP origem,
               porta origem, IP destino, porta destino, tipo pacote,
               tamanho e data */
10         end
11         if pacote == ICMP then
12             /*inserir no banco de dados as informações: IP origem,
               porta origem, IP destino, porta destino, tipo pacote,
               tamanho e data */
13         end
14     end
15 end
```

**Figura 4.14:** Pseudo Algoritmo para Captura de Pacotes

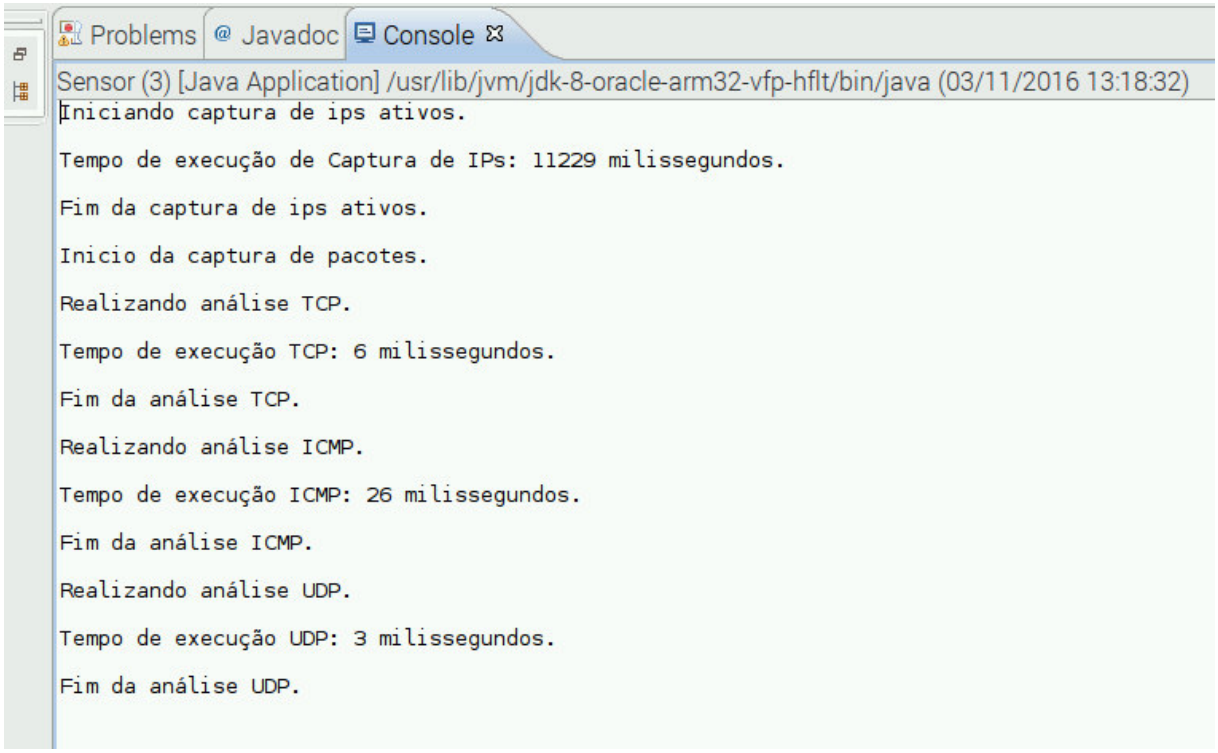



**Figura 4.15:** Protótipo do IDS-IoT

## 4.5 Síntese

Neste capítulo, a arquitetura do IDS-IoT foi apresentada, possuindo 5 (cinco) camadas, na qual cada uma possui suas funcionalidades. Também, os diagramas UML do IDS-IoT foram apresentados bem como os pseudo códigos utilizados. Finalizando-se com apresentação do protótipo desenvolvido para que se possa avaliar a proposta apresentada.

# 5 Testes do IDS-IoT

Avaliar o comportamento do IDS-IoT, bem como os resultados obtidos por ele é de suma importância. Para isto, este capítulo tem por objetivo apresentar os resultados dos testes do IDS-IoT.

## 5.1 Ambiente de Teste

Os testes do IDS-IoT foram feitos em um ambiente de testes controlado e composto por *desktops* e *Raspberry Pi* 3. Dois cenários foram propostos para o ambiente de testes: o primeiro cenário contém uma aplicação de *streaming*; o segundo cenário consiste em uma aplicação para medição de temperatura e umidade. O dispositivo utilizado para executar o IDS-IoT foi um *Raspberry Pi* 3, modelo B, com as seguintes configurações: 1.2Ghz 64-*bit quad-core* ARMv8 CPU, 10/100Mbps *Lan Speed*, 802.11n *Wireless LAN*, executando o Sistema Operacional *Raspbian* — uma distribuição Linux.

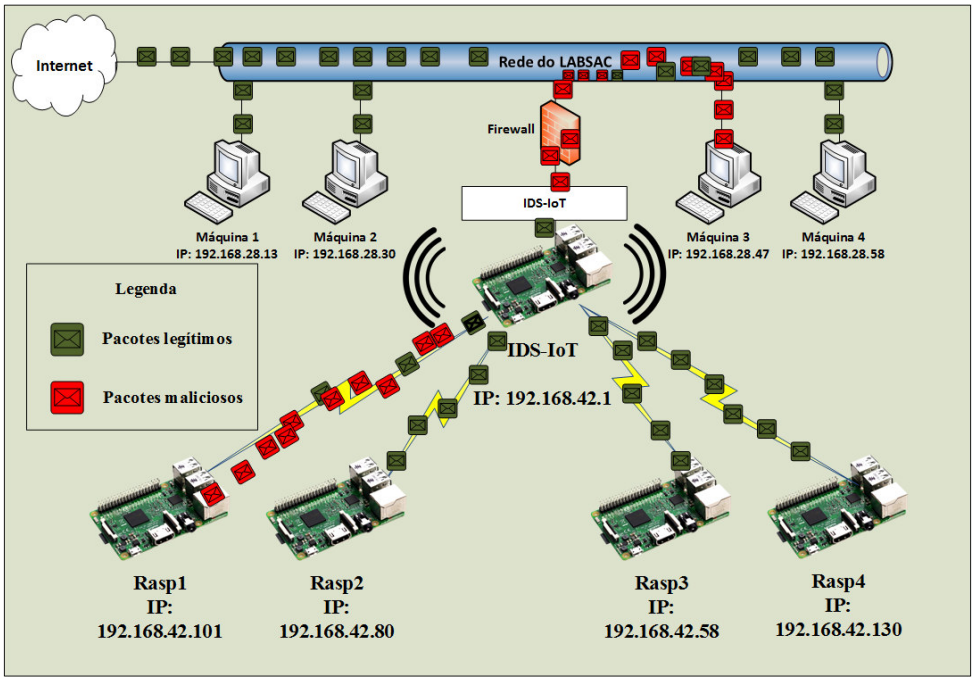



**Figura 5.1:** Cenário utilizado para teste do IDS-IoT

A Figura 5.1 apresenta o primeiro cenário do ambiente de teste utilizado para o teste do IDS-IoT. Conforme os tipos de ameaças abordados nesta dissertação, o

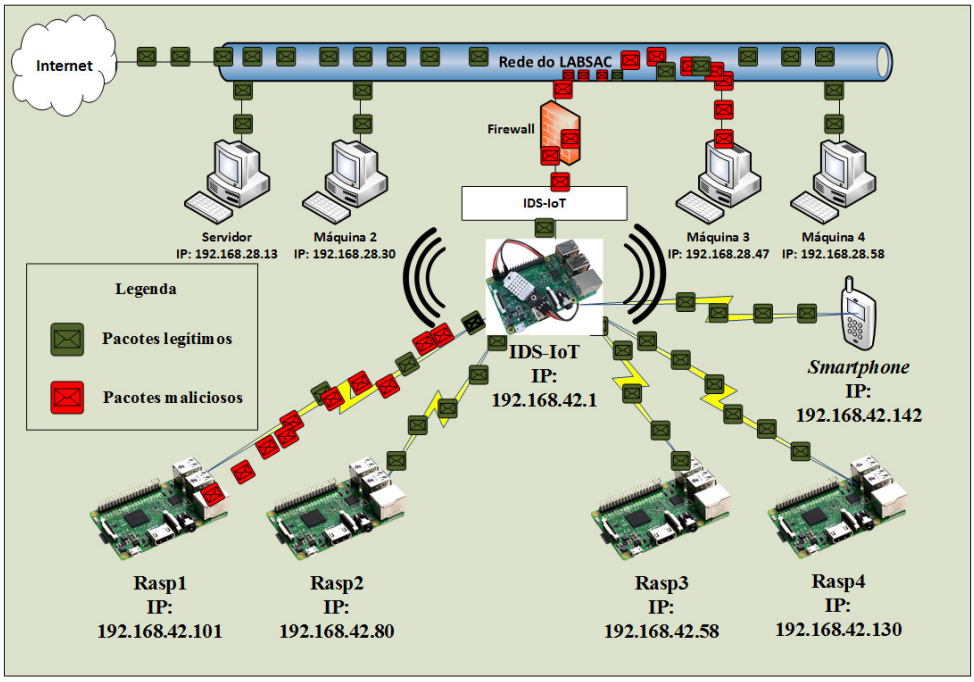



**Figura 5.2:** Cenário de Coleta de Temperatura

escopo do problema avaliado nos testes foi o ataque de Negação de Serviço (DoS). Os ataques realizados foram agrupados conforme o tipo de protocolo (TCP, ICMP e UDP).

A Figura 5.2 apresenta o segundo cenário de testes no qual IDS-IoT foi empregado, no qual foi utilizado um sensor DHT11 para realizar a coleta da temperatura do laboratório Laboratório de Sistemas e Arquiteturas Computacionais (LABSAC).

O sistema de software utilizado para captura de temperatura e umidade foi desenvolvido por Moraes [27], no qual fez uso do protocolo XMPP (eXtensible Messaging and Presence Protocol) para garantir a segurança e a confiabilidade na entrega das mensagens trocadas entre os dispositivos na IoT.

O dispositivo *IDS-IoT* (IP: 192.168.42.1) foi configurado para ser um ponto de acesso *wifi*, no qual os demais dispositivos (Rasp1 ao Rasp4) estão conectados via conexão *wifi*, constituindo uma rede apenas composta por dispositivos de baixos recursos.

O dispositivo *IDS-IoT* foi conectado à rede do LABSAC da Universidade Federal do Maranhão (UFMA), o qual mantém uma conexão com à Internet através da

placa de rede LAN e distribui, pelo sinal *wifi*, utilizando a placa *Wireless*. A ferramenta utilizada para a realização dos ataques foi o *hping3*[18].

Os seguintes ataques foram realizados nos dois cenários: *Syn Flood* e *Land Attack*, baseados no protocolo TCP. Bem como ataques que utilizam o protocolo ICMP, foram realizados: *ICMP Flood* e *Smurf Attack*. E, com relação ao protocolo UDP, o ataque *UDP Flood* foi executado.

## 5.2 Dados do Teste

O sistema de software empregado no teste do primeiro cenário foi um servidor de *streaming* (servidor *Web*) baseado no servidor *Apache* versão 2.4.10 e Sistema Operacional *Raspibian* implantados em um *Raspberry Pi* 3. Conforme o cenário apresentado na Figura 5.1, o servidor *web* está sendo executado no dispositivo *Rasp4* (IP: 192.168.42.130).

Nos testes realizados, todos os demais dispositivos estão acessando as páginas *web* e apresentando os vídeos providos pelo servidor, inserindo, assim, uma quantidade significativa de pacotes na rede.

A Figura 5.3 apresenta a página inicial da aplicação *web* desenvolvida [19] no qual os clientes possuem a opção de assistir 4 (quatro) diferentes vídeos. Todos os vídeos estão no formato MPEG-4 (.mp4). O detalhamento das informações dos vídeos pode ser visualizado na Tabela 5.1.

**Tabela 5.1:** Detalhes dos vídeos

| Vídeo | Duração (s) | Tamanho (MB) |
|---|---|---|
| 01 | 30 | 70.80 |
| 02 | 19 | 48.30 |
| 03 | 45 | 44.00 |
| 04 | 53 | 40.03 |

[18]Disponível em: http://www.hping.org/. Acessado em: 02/01/2017.

[19]ressalva-se que foi utilizado um *layout* de código aberto disponível no site da W3School (Disponível em: http://www.w3schools.com/css/tryit.asp?filename=tryresponsive_video3. Acessado em: 20/11/2016.)

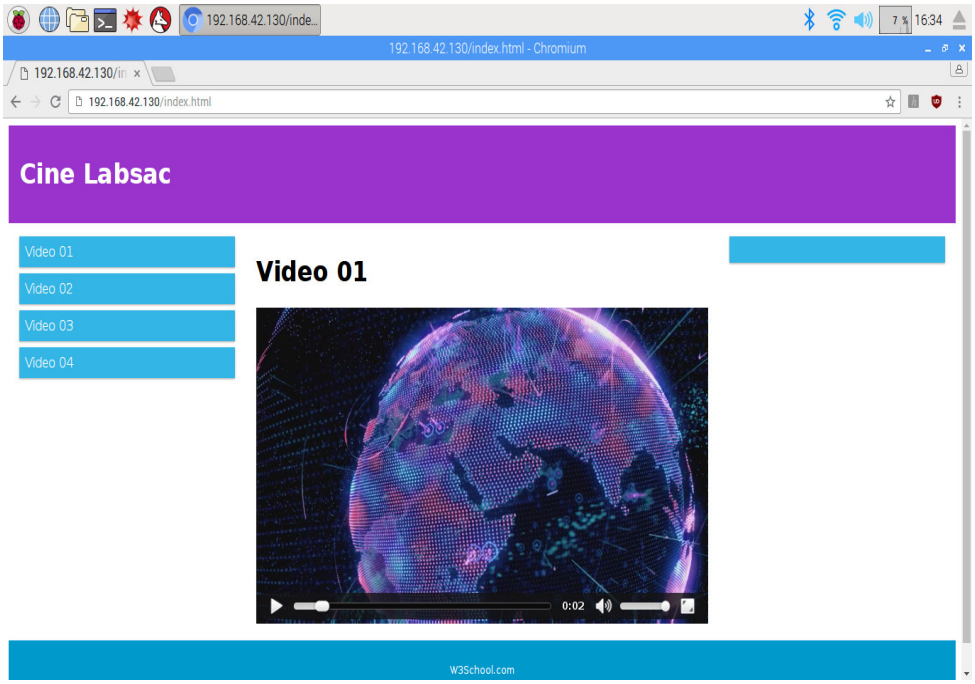



**Figura 5.3:** Aplicação *Web* utilizada para gerar tráfego de dados na rede de teste

Como pode ser visualizado na Tabela 5.1, a duração de cada vídeo está em segundos. Porém, duas configurações foram feitas nas páginas *web*: atualização automaticamente da página a cada 60 segundos; e carregamento automático do vídeo após o carregamento da página. Com essas configurações, o constante envio de pacotes na rede foi possibilitado.

Com relação ao segundo cenário de testes, a coleta de temperatura e umidade é feita a cada 3 (três) segundos e enviada para o servidor (IP: 192.168.28.13). Após o recebimento das informações pelo servidor, este envia as informações coletadas para o dispositivo *Smartphone* (IP: 192.168.42.142), que possui uma aplicação cliente que recebe as informações de temperatura e umidade do sensor DHT11.

## 5.3 Resultados dos testes

Nesta seção, os resultados obtidos ao utilizar o IDS-IoT são apresentados. Os comandos utilizados na ferramenta *hping3*, utilizados para a realização dos ataques serão abordados. Todos os ataques descritos nessa seção foram realizados utilizando o dispositivo denominado ”Máquina 3” no cenário apresentado na Figura 5.1.

### 5.3.1 *Syn Flood*

Quanto ao ataque *Syn Flood*, este é caracterizado por uma variância nas etapas de estabelecimento da comunicação TCP, o *handshake* de três vias. Nas etapas normais de estabelecimento de comunicação, um cliente ao tentar se comunicar com o servidor, envia uma requisição para o servidor com uma *flag* SYN (*synchronize*). O servidor recebe esta *flag* e responde com uma outra *flag* SYN-ACK (*synchronize-acknowledgment*) e por fim, o cliente envia uma *flag* ACK [7]. A figura 5.4 expõe as etapas do handshak.

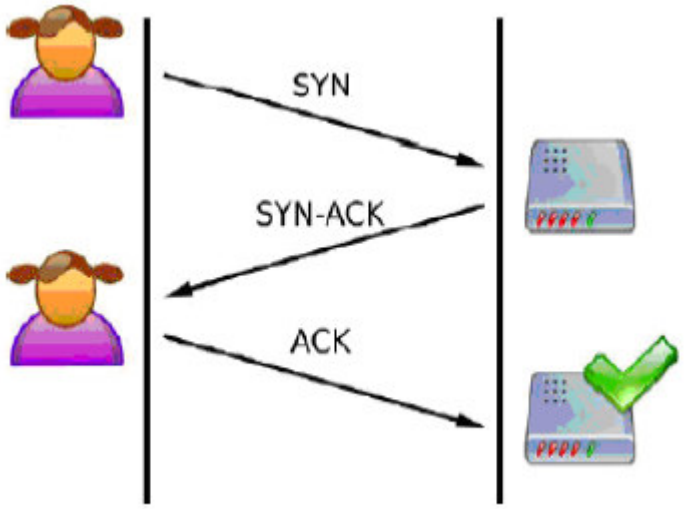



**Figura 5.4:** *Three way handshak* [1]

Ao realizar um ataque de *Syn Flood*, o atacante faz uso das etapas de estabelecimento de comunicação com o servidor. Este envia uma quantidade massiva de pacotes com a *flag* SYN para o servidor e recebe a *flag* SYN-ACK, porém não confirma a conexão estabelecida com a *flag* ACK, deixando a conexão semiaberta [7], consumindo, assim, os recursos do servidor, tais como: largura de banda, memória e processamento. A Figura 5.5 demonstra o estado da conexão no momento de um ataque *Syn Flood*.

Ao ser realizado o ataque *Syn Flood*, o IDS-IoT conseguiu detectar de forma satisfatória as informações do atacante e as características dos pacotes que caracterizam este tipo de ataque.

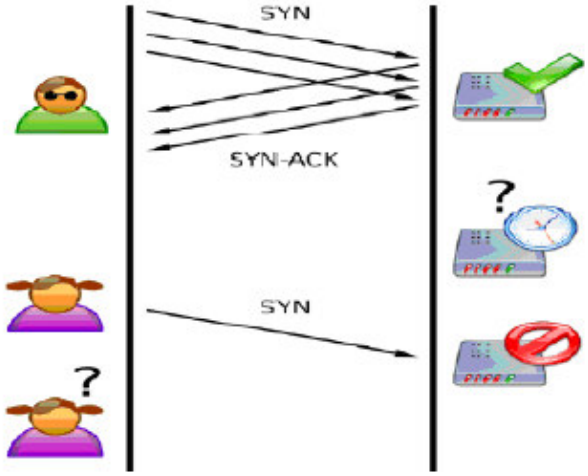



**Figura 5.5:** *Syn Flood Attack* [1]

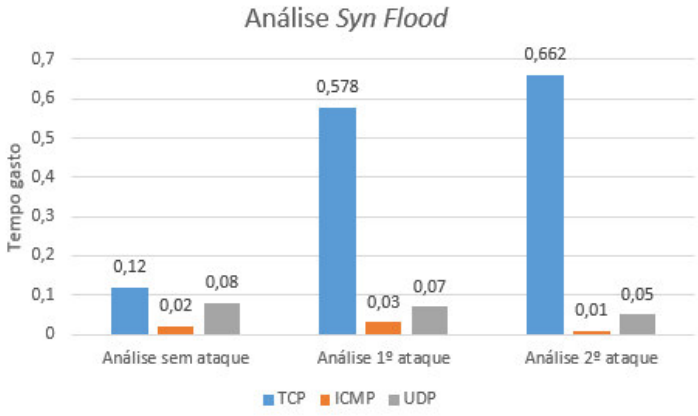



**Figura 5.6:** Tempo de Análise *Syn Flood*

Na Figura 5.6, é apresentado o tempo gasto para a realização das análises de pacotes capturados conforme cada tipo de protocolo. O tempo gasto para realização de todas as análises foi medido em milissegundos.

A legenda "Análise sem ataque" faz menção ao tempo gasto para a realização da análise do pacotes sem ocorrer ataques na rede. Já "Análise 1º ataque" e "Análise 2º ataque" são referentes ao tempo gasto para a realização dos pacotes no momento da realização do ataque *Syn Flood*. O ataque *Syn Flood* foi realizado duas vezes em momentos diferentes, com duração de 2 minutos cada ataque.

Como pode ser analisado na Figura 5.7, a ocorrência o ataque *Syn Flood* fez com que houvesse uma variação na utilização dos recursos do dispositivo. A utilização de processamento inicialmente estava em 5%, atingindo um pico de 15% de utilização.

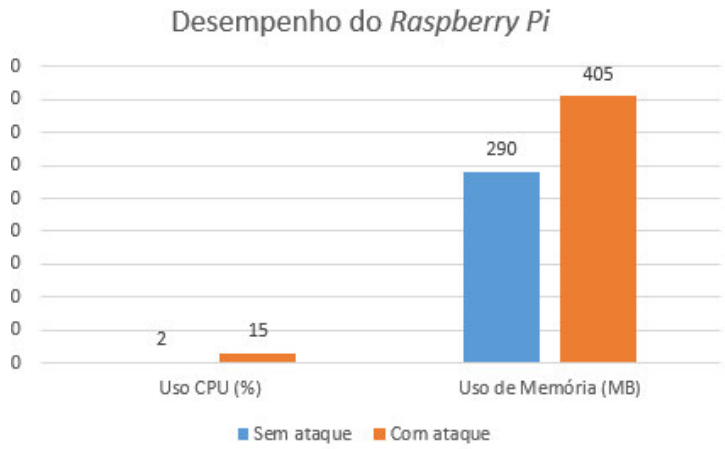



**Figura 5.7:** Desempenho do Dispositivo (*Syn Flood*)

Em relação ao uso de memória, inicialmente o dispositivo está utilizando 290 *megabytes* (MB) variando até chegar o pico de 405 *megabytes* (MB).

O comando utilizado para a realização do ataque em questão, foi o seguinte: *hping3 -V -c 1000000 -d 120 -S -w 64 -p 135 -s 135 - -flood -a 177.88.55.2 192.168.42.101* e *hping3 -V -c 1000000 -d 120 -S -w 64 -p 135 -s 135 - -flood -a 45.66.3.22 192.168.42.101*.

Explicando a sintaxe utilizada, temos: *hping3* corresponde à ferramenta utilizada; *-V* corresponde à *verbose*; *-c 1000000* corresponde à contagem de um milhão de pacotes a ser enviados; *-d 120* está relacionado com o tamanho dos dados; *-p 133 -s 135* correspondem às portas TCP de origem e destino; *-S* faz com que somente a *flag* SYN esteja presente nos pacotes enviados; *- -flood* comando que possibilita que os pacotes sejam enviados o mais rápido possível; *-a* este comando possibilita mascarar o IP original da máquina, forjando um falso (177.88.55.2 e 45.66.3.22, neste exemplo, estes foram os IPs forjados); e por fim, o IP da vítima, 192.168.42.101.

Com relação a regra de bloqueio gerada, segue: *iptables -A FORWARD -s 177.88.55.2 -d 192.168.42.101 -j DROP* e *iptables -A INPUT -s 45.66.3.22 -d 192.168.42.101 -j DROP*. Explicando as regras de bloqueio geradas, segue: *iptables* corresponde ao *firewall* utilizado para a realização dos bloqueios; *-A FORWARD* é o *chain* por qual os pacotes do ataque são recebidos pelo IDS-IoT e trasmitido ao IP de destino; *-s* corresponde ao IP do atacante; *-d* corresponde ao IP da vítima; e por fim, *-j DROP* faz com que todos os pacotes do IP de origem para o destino específico sejam "*dropados*". O motivo da regra não especificar o tipo de protocolo deva ser "*dropado*" de todos os

pacotes do emissor, é porque foi levado em conta que todos os tipos de pacotes, sejam estes de protocolos diferentes, devam ser "*dropados*".

### 5.3.2 *Land attack*

Em relação ao *Land attack*, possui a característica de enviar pacotes para o servidor com o mesmo endereço de IP que o servidor possui, e/ou com a mesma porta de origem e destino, fazendo que com isso o servidor ao tentar responder a requisição do cliente entre em *loop* infinito, uma vez que este está respondendo a requisição para o mesmo endereço de IP de origem e destino. Na Figura 5.8 é possível visualizar o tempo gasto para a realização da análise dos pacotes recebidos. O ataque *Land Attack* foi realizado duas vezes em momentos diferentes, com duração de 2 minutos cada ataque.

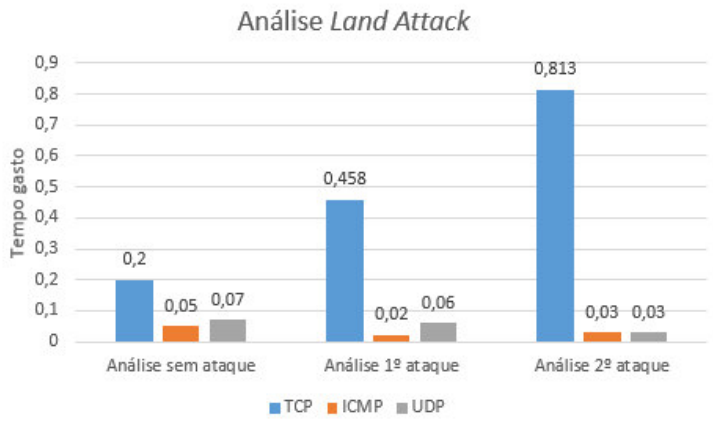



**Figura 5.8:** Tempo de Análise *Land Attack*

A legenda "Análise sem ataque" faz menção ao tempo gasto para a realização da análise do pacotes sem ocorrer ataques na rede. Já "Análise 1º ataque" e "Análise 2º ataque" são referentes ao tempo gasto para a realização dos pacotes no momento da realização do ataque *Land Attack*. O ataque *Land Attack* foi realizado duas vezes em momentos diferentes, com duração de 2 minutos cada ataque.

O comando utilizado para a realização do ataque em questão, foi o seguinte: *hping3 -V -c 1000000 -d 120 -w 64 -p 139 -s 139 - -flood –a 192.168.42.101 192.168.42.101*. Explicando a sintaxe utilizada, temos: *hping3* corresponde à ferramenta utilizada; *-V* corresponde à *verbose*; *-c* 1000000 corresponde à contagem de um milhão de pacotes a ser enviados; *-d 120* está relacionado com o tamanho dos dados; *-p 139 -s 139*

corresponde à porta TCP de origem e destino; - *-flood* comando que possibilita que os pacotes sejam enviados o mais rápido possível; *-a* este comando possibilita mascarar o IP original da máquina, forjando o IP do dispositivo vítima; e por fim, o IP da vítima, *192.168.42.101*.

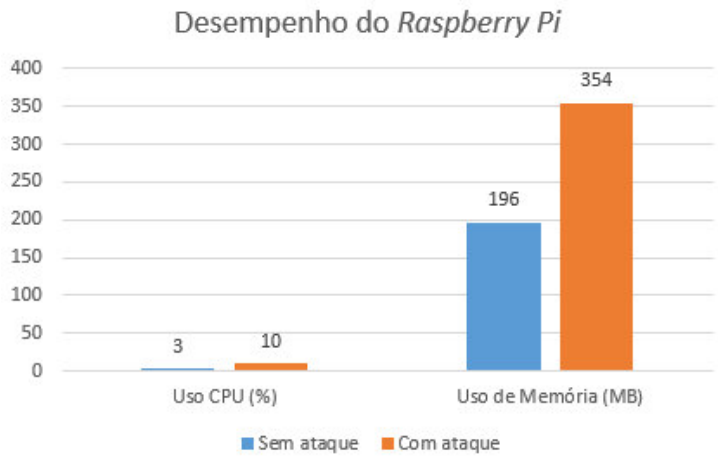



**Figura 5.9:** Desempenho do Dispositivo (*Land Attack*)

Com relação a regra de bloqueio gerada, segue: *iptables -A FORWARD -s 177.88.55.2 -d 192.168.42.101 -j DROP* e *iptables -A FORWARD -s 192.168.42.101 -d 192.168.42.101 -j DROP*. Explicando as regras de bloqueio geradas, segue: *iptables* corresponde ao *firewall* utilizado para a realização dos bloqueios; *-A FORWARD* é o *chain* por qual os pacotes do ataque são recebidos pelo IDS-IoT e trasmitido ao IP de destino; *-s* corresponde ao IP do atacante; *-d* corresponde ao IP da vítima; e por fim, *-j DROP* faz com que todos os pacotes do IP de origem para o destino específico sejam *"dropados"*.

Como pode ser analisado na Figura 5.9, a ocorrência o ataque *Land Attack* fez com que houvesse uma variação na utilização dos recursos do dispositivo. A utilização de processamento inicialmente estava em 3%, atingindo um pico de 10% de utilização. Em relação ao uso de memória, inicialmente o dispositivo está utilizando 196 MB variando até chegar o pico de 354 MB.

### 5.3.3 *ICMP Flood*

O ataque *ICMP Flood*, o comando utilizado foi *hping3 -a 55.88.123.12 -1 192.168.42.101 - -flood*. A saber, o parâmetro -1 indica que pacotes ICMP serão enviados

para o IP de destino, a fim de realizar o ataque de negação de serviço, inundando a largura de banda da vítima. Na Figura 5.10 é possível visualizar o tempo gasto para a realização da análise dos pacotes recebidos.

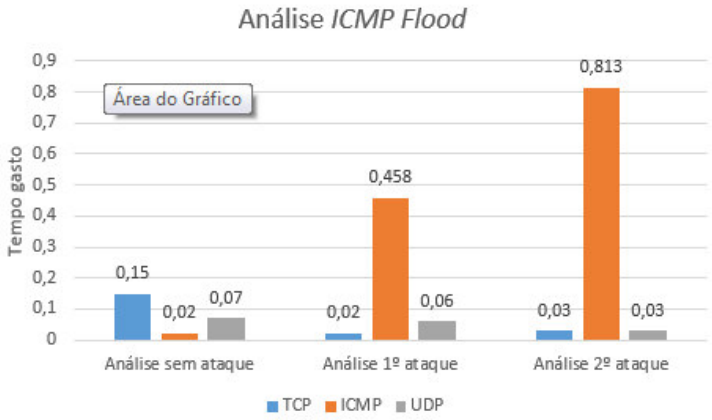



**Figura 5.10:** Tempo de Análise (*ICMP Flood*)

A legenda "Análise sem ataque" faz menção ao tempo gasto para a realização da análise do pacotes sem ocorrer ataques na rede. Já "Análise 1º ataque" e "Análise 2º ataque" são referentes ao tempo gasto para a realização dos pacotes no momento da realização do ataque *ICMP Flood*. O ataque *ICMP Flood* foi realizado duas vezes em momentos diferentes, com duração de 2 minutos cada ataque.

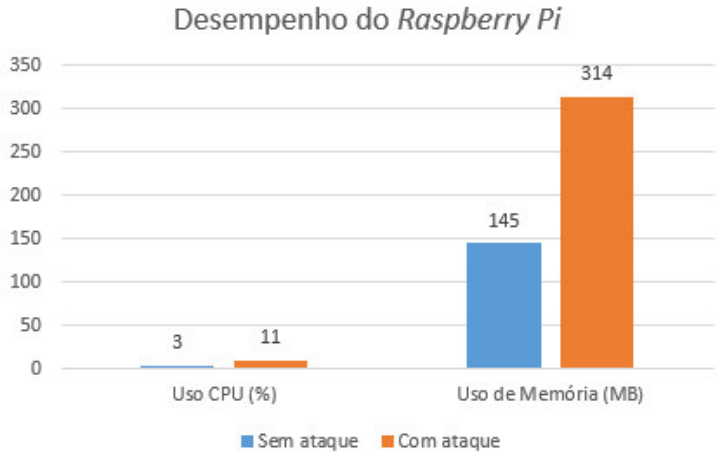



**Figura 5.11:** Desempenho do Dispositivo (*ICMP Flood*)

Como pode ser analisado na Figura 5.11, a ocorrência do ataque *ICMP Flood* fez com que houvesse uma variação na utilização dos recursos do dispositivo. A utilização de processamento inicialmente estava em 3%, atingindo um pico de 11% de utilização. Em relação ao uso de memória, inicialmente, o dispositivo está utilizando 145 MB variando até chegar o pico de 314 MB.

### 5.3.4 *Smurf Attack*

Outro ataque realizado do tipo ICMP, foi o *Smurf Attack*. Esta ameaça tem por finalidade atacar o endereço de *broadcast* da rede. O ataque foi realizado utilizando o seguinte comando:*hping3 -a 55.4.1.3 -1 192.168.42.255 - -flood* e *hping3 -a 60.78.1.4 -1 192.168.42.255 - -flood*. Nota-se que os comandos de ataque de *Smurf Attack* e *ICMP Flood* são semelhantes, diferindo que ao invés de inundar a largura de banda do dispositivo, o *Smurf Attack* irá inundar o broadcast da rede, inundando com requisições falsas.

As regras iptables geradas foram: *iptables -A INPUT -s 55.4.1.3 -d 192.168.42.255 -j DROP* e *iptables -A INPUT -s 60.78.1.4 -d 192.168.42.255 -j DROP*. A Figura mostraram o tempo de análise que o IDS-IoT precisou para analisar os pacotes deste tipo de ataque.

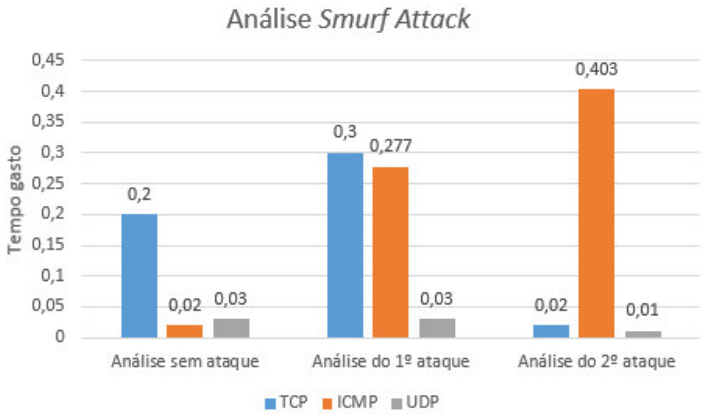



**Figura 5.12:** Tempo de Análise (*Smurf Attack*)

A legenda "Análise sem ataque" faz menção ao tempo gasto para a realização da análise do pacotes sem ocorrer ataques na rede. Já "Análise 1º ataque" e "Análise 2º ataque" são referentes ao tempo gasto para a realização dos pacotes no momento da realização do ataque *Smurf Attack*. O ataque *Smurf Attack* foi realizado duas vezes em momentos diferentes, com duração de 2 minutos cada ataque.

Ao analisar a Figura 5.13, a ocorrência o ataque *Smurf Attack* fez com que houvesse uma variação na utilização dos recursos do dispositivo. A utilização de processamento inicialmente estava em 1%, atingindo um pico de 13% de utilização. Em relação ao uso de memória, inicialmente o dispositivo está utilizando 160 MB variando até chegar o pico de 336 MB.

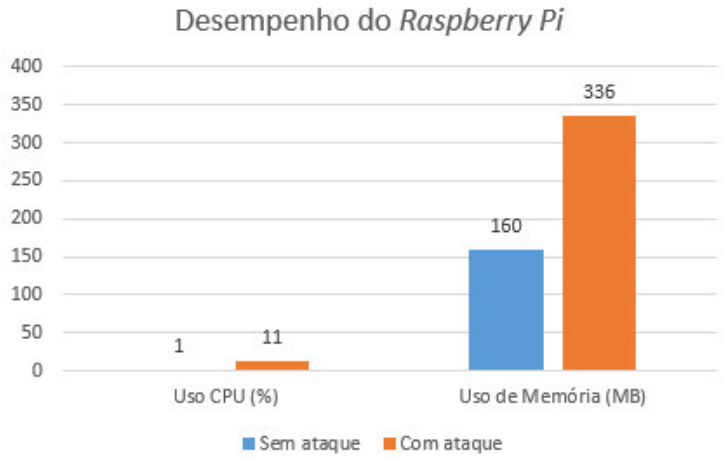



**Figura 5.13:** Desempenho do Dispositivo (*Smurf Attack*)

### 5.3.5 *UDP Flood*

Por fim, o último ataque realizado foi *UDP Flood*. Ao realizar um ataque de *UDP Flood*, o atacante envia uma quantidade massiva de datagramas UDP, sobrecarregando o limite de banda de sua vítima. Para tanto, o ataque foi realizado utilizando o seguinte comando: *hping3 -2 -a 200.78.96.2 192.168.42.101 - -flood* e *hping3 -2 -a 118.98.75.21 192.168.42.101 - -flood*.

Após a detecção do ataque, foi gerado a seguinte regra para barrar os pacotes do emissor: *iptables -A INPUT -s 144.78.96.33 -d 192.168.42.101 -j DROP* e *iptables -A FORWARD -s 118.98.75.21 -d 192.168.42.101 -j DROP*. O ataque *UDP Flood* foi realizado duas vezes em momentos diferentes, com duração de 2 minutos cada ataque.

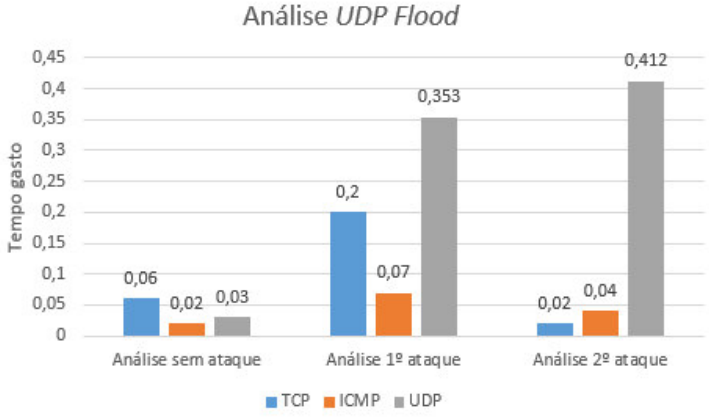



**Figura 5.14:** Tempo de Análise (*UDP Flood*)

Ao analisar a Figura 5.15, a ocorrência o ataque *UDP Flood* fez com que houvesse uma variação na utilização dos recursos do dispositivo. A utilização de processamento inicialmente estava em 1%, atingindo um pico de 11% de utilização. Em relação ao uso de memória, inicialmente o dispositivo está utilizando 177 MB variando até chegar o pico de 352 MB.

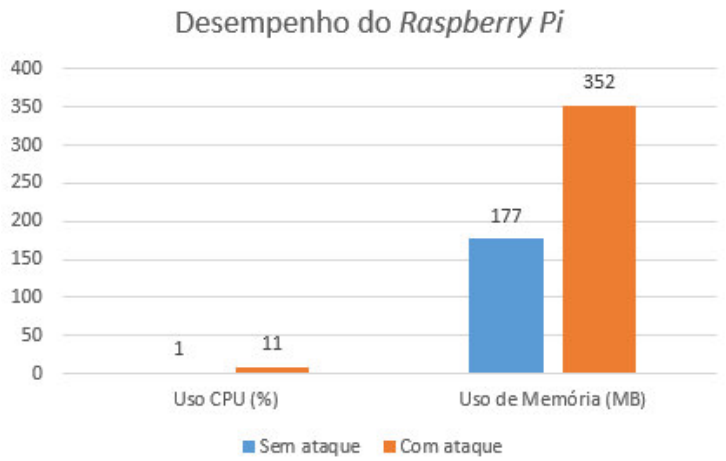



**Figura 5.15:** Desempenho do Dispositivo (*UDP Flood*)

### 5.3.6 Utilização de Recursos de Entrada e Saída

Uma vez que a solução proposta faz a utilização de banco de dados para o armazenamento das características dos pacotes capturados, a Figura 5.16 apresenta a utilização dos recusos de entrada e saída (I/O).

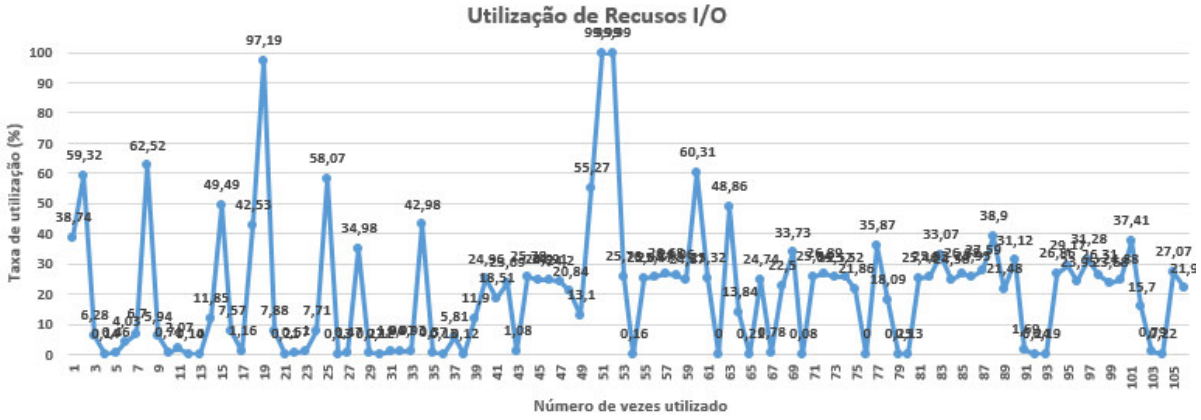



**Figura 5.16:** Utilização de Recusos I/O

Os dados apresentados na Figura 5.16 foram coletados utilizando a ferramenta *iotop*[20] (versão 0.6). No período de 3 minutos foram realizados 105 operações de I/O, no qual obteve um pico, em dois momentos distintos, de utilização de 99% dos recursos de I/O.

Como pode ser visualizado é constante a variação da utilização de tais recursos, no qual deve ser levado em consideração a quantidade de registros que estão sendo inseridos no banco de dados.

### 5.3.7 Verdadeiros-Positivos e Falsos-Positivos

Embora a detecção dos ataques realizados tenha ocorrido, a geração de falsos alertas/avisos de ameaças foram gerados. A geração de falsos-positivos ocorreu no período de três horas que o IDS-IoT ficou executando.

Os dispositivos Rasp1, Rasp2 e Rasp3 (clientes) estavam acessando os vídeos disponibilizados pelo dispositivo Rasp4 (servidor), através do serviço *web*. O constante envio de requisições dos clientes fez com que ocorresse a geração de alertas falsos-positivos. A Tabela 5.2 expõe a quantidade de alertas gerados durante a execução do IDS-IoT.

**Tabela 5.2:** Quantidade de Alertas Gerados - Serviço de *Streaming*

| *max_pacotes* | Verdadeiros-Positivos | Falsos-Positivos |
|---|---|---|
| 600 | 10 | 29 |
| 800 | 10 | 21 |
| 1000 | 10 | 12 |

Como pode ser analisado, à medida que o número de pacotes que determinam um ataque DoS aumenta, o número de falsos-positivos diminui. Porém, a determinação do valor da variável *max_pacotes* não é algo trivial, pois o IDS-IoT pode deixar de identificar um real ataque.

Sobre os valores utilizados para determinar um ataque DoS no cenário de coleta de temperatura, os valores da variável *max_pacotes* foram reduzidos. Pode-se observar que o número de falsos-positivos foi zero, mesmo variando o valor de

[20]Disponível em: http://guichaz.free.fr/iotop/. Acessado em: 02/01/2017.

**Tabela 5.3:** Quantidade de Alertas Gerados - Coleta de Temperatura

| *max_pacotes* | Verdadeiros-Positivos | Falsos-Positivos |
|---|---|---|
| 600 | 10 | 0 |
| 400 | 10 | 0 |
| 200 | 10 | 0 |

*max_pacotes*. Acredita-se que isto ocorreu devido a pequena quantidade de pacotes trafegados no exemplo do sensor de temperatura. Quanto aos verdadeiros-positivos, nota-se que os 10 ataques foram detectados.

## 5.4 Trabalhos Relacionados e Comparações

Utilizar trabalhos já apresentados à comunidade científica é de suma importância para o desenvolvimento de qualquer trabalho científico. Tendo conhecimento desta prerrogativa, esta seção tem por objetivo analisar alguns dos trabalhos já desenvolvidos.

Durante à analise de tais trabalhos serão extraídos algumas informações (ameaças detectadas e taxa de detecção, por exemplo) que servirão de base base para o desenvolvimento do trabalho proposto. Tais trabalhos apresentam limitações, como o número de ataques que estes são capazes de detectar.

De fato, a limitação é inevitável, uma vez que tais IDS são baseados em assinaturas. Os trabalhos mencionados foram desenvolvidos pelos seguintes autores: Kasinathan *et al*. [20], Sonar e Upadhyay [37], Machaka *et al*. [24] e Chen *et al*. [9].

### 5.4.1 *Denial-of-Service detection in 6LoWPAN based Internet of Things*

Kasinathan *et al*. [20] desenvolveram um IDS que visa combater ataques DoS na IoT, mas precisamente em redes 6LoWPAN, no qual o trabalho foi desenvolvido no âmbito projeto *ebbits* da EU FP7[21] . A Figura 5.17 apresenta a arquitetura utilizada pelos autores.

[21]Disponível em: http://www.ebbits-project.eu/. Acessado em: 15/10/2016.

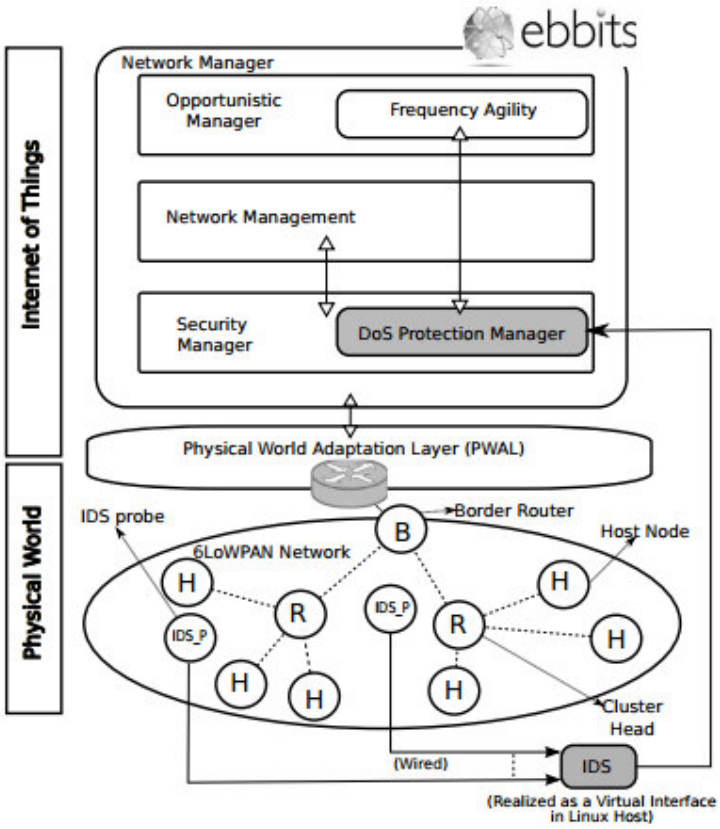



**Figura 5.17:** Arquitetura para Detecção DoS [20]

*Physical world* é representado pela rede 6LoWPAN, no qual existe uma variedade de *hosts* (H) conectados aos seus respectivos *cluster* (R). Os dados trafegados na rede são entregues ao *Network Manager* pelo roteador de borda (B). *Network management* possui três componentes: *Network management*, *Opportunistic manager* e *Security manager*.

O componente ***network management*** tem por finalidade realizar o monitoramento da rede, no qual através deste é possível ter algumas informações, como: latência e nível de interferência na rede, por exemplo.

***Opportunistic manager*** permite a obtenção de informações relevantes para o bom funcionamento da rede. Fornece, ainda, a obtenção de informações de interferências na rede, no qual é determinar o melhor canal de transmissão para ser utilizado.

***Security manager*** tem por finalidade fornecer mecanismos de segurança para a comunicação entre os dispositivos pertencentes à rede, em que faz uso de criptografia. Este componente possui dois subcomponentes: *DoS protection manager* e IDS.

- ***DoS protection manager***: este componente objetiva receber os alertas gerados pelo IDS, em que é realizado a extração de informações (quantidade de pacotes rejeitados, por exemplo) de outros gerenciadores *ebbits* (*opportunistic manager* e *network management*) para a verificação se os alertas gerados são realmente de ameaças detectadas;

- **IDS**: o sistema proposto trata-se de um NIDS. Os pacotes capturas pelas sondas *IDS_P*[22] são enviadas para para o módulo IDS que será responsável pelo processamento e determinar se os pacotes capturados são característicos de uma atividade anômala (Figura 5.17).

Uma vez que, 6LoWPAN é uma versão otimizada do protocolo IPv6 para dispositivos com uma quantidade limitada de recursos [21], os autores utilizaram um IDS (componente IDS apresentado na Figura 5.17) de código aberto, denominado de Suricata[23], que tem suporte para IPv6.

Quanto aos testes obtidos, uma aplicação foi desenvolvida para a realização dos ataques de *UDP Flood* baseado em IPv6 na rede 6LoWPAN, em que esta aplicação foi desenvolvida utilizando o sistema operacional *Contiki*. A Figura expõe um exemplo de alerta gerado ao ser detectado um ataque.

```
alert udp any any -> any any (msg:\"My Threshold
rule works \";threshold: type threshold,
track by_dst, count 30, seconds 1; sid:999999;
classtype: misc-activity;rev:1; content:"Hello";
priority:1;)
```

**Figura 5.18:** Alerta Gerado [20]

A detecção de um ataque está em consonância com as regras existentes no Suricata, no qual se uma atividade detectada corresponder as regras podem ser tomadas algumas ações: *pass* (no qual libera os pacotes a trafegarem), *drop* (negar o recebimento dos pacotes), *reject* (no qual os pacotes devem ser rejeitados) e *alert*(geração da mensagem de ataque detectado).

[22]IDS_P foi primeiramente apresentado em [42]. Este tem a capacidade de operar como um *firmware* e possibilita a captura dos pacotes em modo promíscuo.

[23]Disponível em: https://oisf.net/. Acessado em: 03/01/2017.

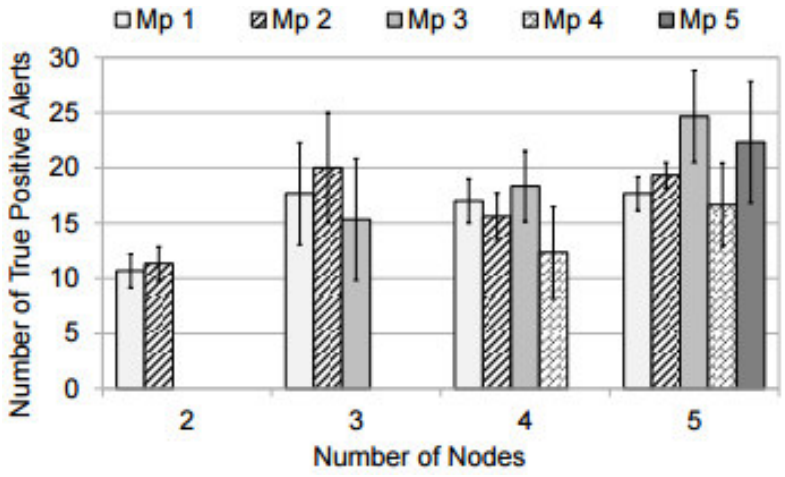



**Figura 5.19:** Taxa de Verdadeiro-Positivo [20]

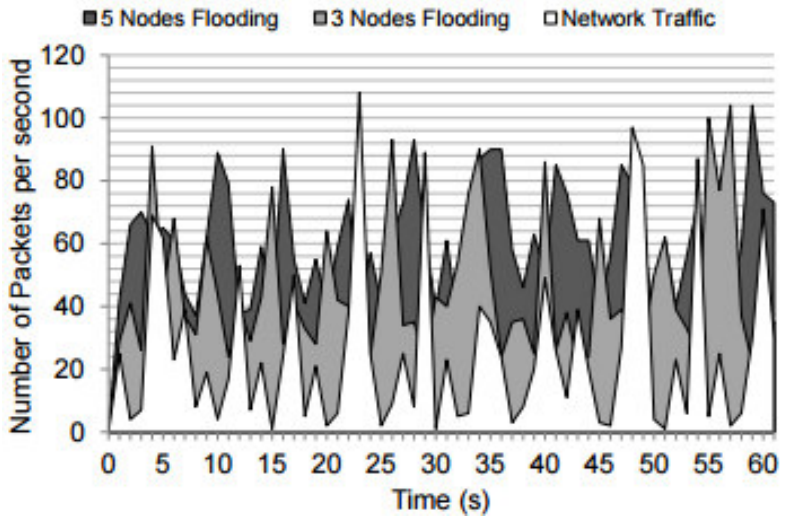



**Figura 5.20:** Tráfego de Rede durante Ataque DoS [20]

O alerta contido na Figura 5.18, foi gerado devido a quantidade de pacotes UDP ultrapassarem a quantidade de 30 pacotes no período de 1 segundo. Ao ser um alerta este possui algumas informações:

- **Ação**: *pass*, *drop*, *reject* e *alert*;
- **Protocolo dos Pacotes**: TCP, ICMP ou UDP;
- **Endereços**: Endereço de Origem, porta de origem, endereço de destino e Destino;

Os ataques de *UDP Flood* foram realizados no período de 1 (um) minuto, sendo que os testes foram realizados do seguinte modo: em primeiro momento foi utilizado apenas um dispositivo (Mp) para a realização do ataque; chegando até a utilização de cinco dispositivos para a realização do ataque.

A obtenção das informações da taxa de verdadeiro-positivo é possível se obter ao analisar a Figura 5.19. Na Figura 5.20 é possível analisar tráfego de rede durante os ataques DoS.

### 5.4.2 *An Approach to Secure Internet of Things Against DDoS*

Sonar e Upadhyay [37] projetaram um IDS para Internet das Coisas, no qual tem por finalidade a detecção de ataques de Negação de Serviço Distribuído na IoT. A ideia geral do sistema desenvolvido é possuir duas listas, *GreyList* e *BlackList*, para o controle de acesso dos pacotes de rede que estão trafegando.

Ao ser identificado um ataque, as informações do atacante são inseridas primeiramente na *GreyList*. Endereço IP e porta de origem, bem como a quantidade de pacotes são exemplos de informações extraídas ao ser detectado um eventual ataque.

A primeira lista, *GreyList* — lista de possível ameaça —, fica responsável por realizar bloqueios de acesso temporários. Já a *BlackList* — lista de ameaça confirmada—, fica responsável por realizar o bloqueio permanente e revogar acesso de um dispositivo;

Para a redução de alertas falso-positivo, as listas são constantemente atualizadas, analisando novamente o comportamento do dispositivo. A cada 40 segundos a *GreyList* é atualizada. Em relação à *BlackList*, esta é atualizada a cada 300 segundos.

O algoritmo de detecção consiste em 6 (seis) etapas, que segue:

- **Primeira etapa**: Verificar a ocorrência de geração de alerta positivo (*true*);
- **Segunda etapa**: Após a identificação de alerta positivo, verificar se o endereço de IP emissor é pertencente à *BlackList*;
- **Terceira etapa**: Determinar o nível de confiança que um dispositivo possui;
- **Quarta etapa**: Após a detecção, inserir o endereço IP do atacante na *GreyList*;
- **Quinta etapa**: Entrar em modo de filtragem;
- **Sexta etapa**: Verificar existência de novos alertas do mesmo IP gerado na primeira etapa; caso seja detectado novamente, inserir IP na *BlackList*.

Foi utilizado o *Contiki OS* como sistema operacional para a execução do IDS proposto. Para a realização dos testes, foi utilizado o simulador *Cooja* pertencente ao sistema operacional utilizado.

Para a simulação da conexão entre os dispositivos, foi realizado a simulação de um roteador de borda baseado no protocolo de roteamento RPL. Para a realização da inundação de pacotes na rede, a ferramenta *Ostinato* foi utilizada. Foi utilizado apenas pacotes IPv6 e UDP.

Quanto aos resultados obtidos, verdadeiro-positivo e falso-positivo, as figuras 5.21 e 5.22 relacionam ao tempo de execução no período de 10 e 20 segundos, respectivamente. À medida que o tempo de utilização é maior, é perceptível que a taxa de detecção de verdadeiro-positivo aumenta.

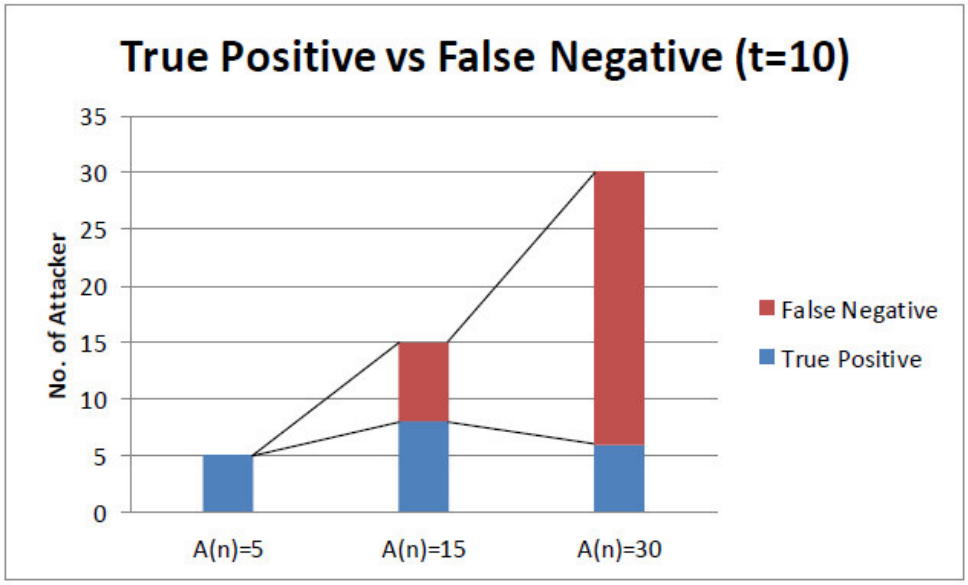



**Figura 5.21:** Verdadeiro-Positivo vs. Falso-Positivo para 10 segundos [37]

### 5.4.3 *Using the Cumulative Sum Algorithm Against Distributed Denial of Service Attacks in Internet of Things*

Machaka *et al*. [24] desenvolveram um Sistema de Detecção de Intrusão para Internet das Coisas em que objetiva a detecção de ataques de negação de serviço *Syn Flood*. Para a realização da detecção dos ataques foi utilizado o algoritmo *Cumulative Sum* (CUSUM).

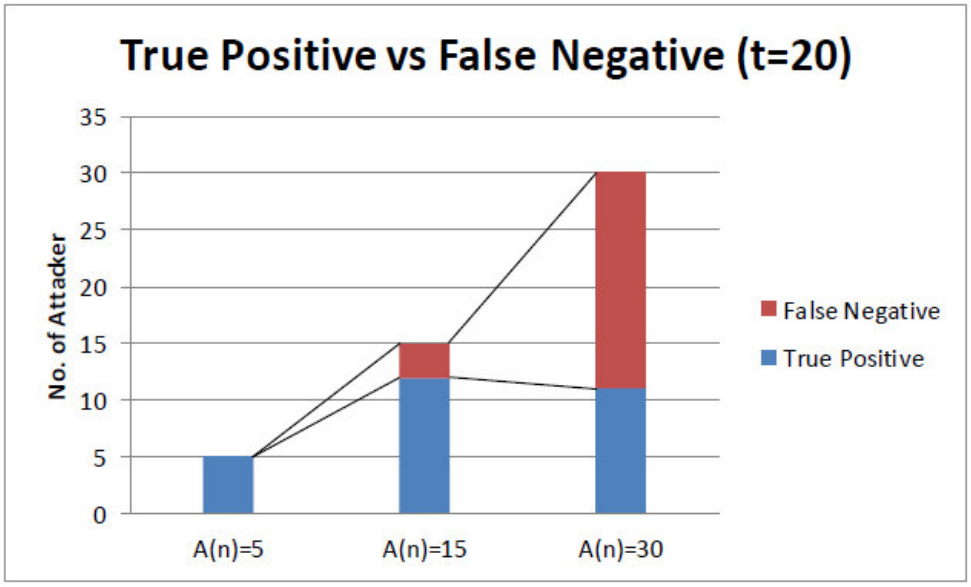



**Figura 5.22:** Verdadeiro-Positivo vs. Falso-Positivo para 20 segundos [37]

Estes mencionam que CUSUM trata-se de um algoritmo que se baseia-se em hipóteses, no qual utiliza variáveis aleatórias independentes e identicamente distribuídas $\{y_i\}$ e utiliza as variações nos comportamentos de rede que podem eventualmente ocorrer. As hipóteses são agrupadas em dois grupos: $\theta_1$ e $\theta_2$.

O grupo $\theta_1$ corresponde aos comportamentos detectados antes da detecção da variação no comportamento. Já o $\theta_2$ faz menção ao comportamento após a detecção da variação comportamental. A Figura 5.23 apresenta a formula para calcular a mudança no comportamento (Sn).

$$S_n = \sum s_i$$

Where,

$$s_i = \ln \frac{P_{\theta_1}(y_i)}{P_{\theta_2}(y_i)}$$

**Figura 5.23:** Cálculo de Variação

A geração de um alerta irá depender se $g_n \geq$ h (h é um parâmetro de limiar). Se for satisfeito essa condição, irá ser gerado um alerta no tempo $n$. A Figura 5.24 apresenta o fórmula para a realização dos cálculos.

Os testes foram realizados utilizaram tráfegos reais do *MIT Lincoln Laboratory*, em que foram utilizados dados capturados num período de 11 horas para

$$\text{where } g_n = S_n - m_n$$

and

$$m_n = \min_{1 \le j \le n} S_j.$$

**Figura 5.24:** Fórmula para Calcular $g_n$

a análise. Os ataques foram realizados no período de minutos, porém, tiveram um intervalo de 30 segundos entre os ataques.

Foram considerados dois tipos de características de ataque: de alto e baixa intensidade. Ataques de baixa intensidade no qual a intensidade de pacotes são elevados gradativamente. Ataques de alta intensidade é quando é identificado uma quantidade muito grande inesperada de pacotes num intervalo de tempo. Os resultados dos testes levou em conta somente ataque de baixa intensidade.

As Figuras 5.25 e 5.26 representam os resultados de verdadeiro-positivo e falso-positivo obtidos, respectivamente. É destacado que a para verdadeiro-positivo a detecção teve uma variação entre 0% e 81%; e falso-positivo variou entre 0% e 7% nos testes realizados.

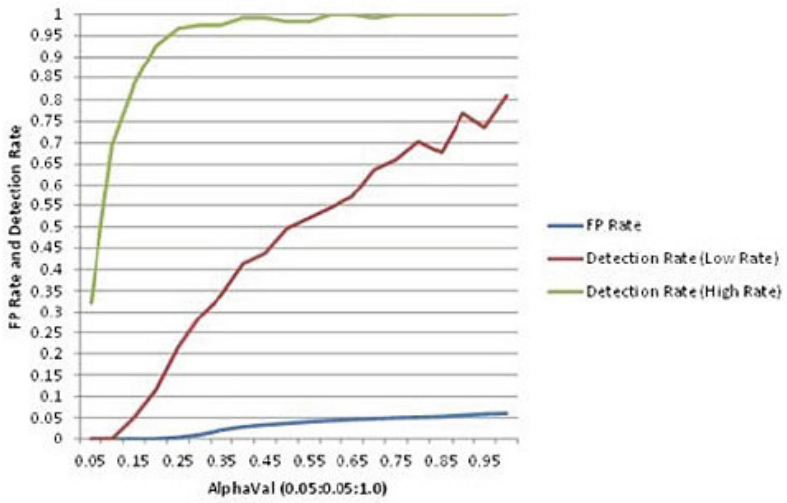



**Figura 5.25:** Resultados de Verdadeiro-Positivo

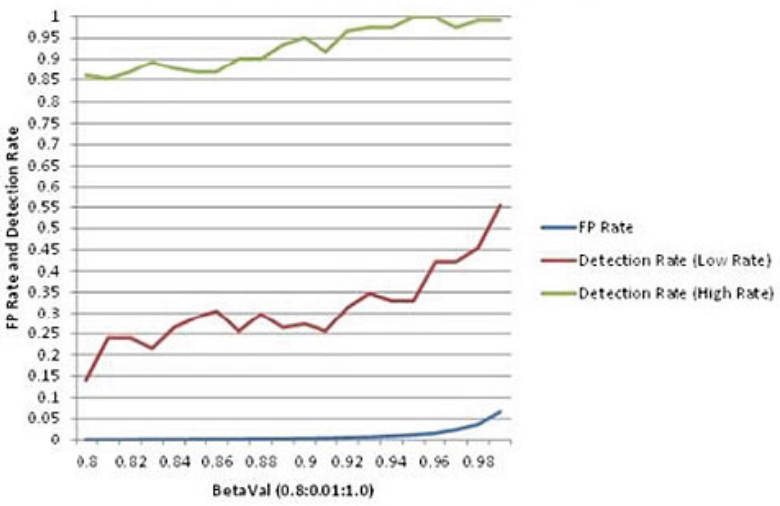



**Figura 5.26:** Resultados de Falso-Positivo

### 5.4.4 *Defense Denial-of Service Attacks on IPv6 Wireless Sensor Networks*

Chen *et al*. [9] desenvolveram um Sistema de Detecção de Intrusão para *Wireless Sensor Network* (WSN) que objetiva a identificação e prevenção de ataques de *wormhole* e *flooding* IPv6.

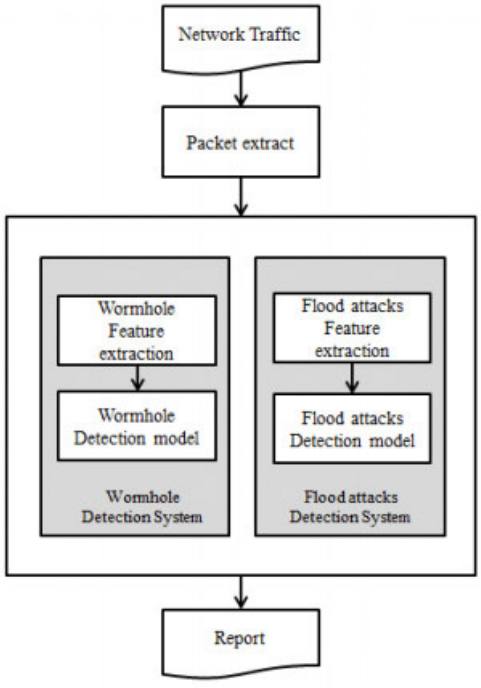



**Figura 5.27:** *Framework* utilizado

Para a identificação das ameaças os autores utilizaram um *framework*. As fases de captura de pacotes, extração de características e identificação dos ataques são demonstrados na Figura 5.27.

A identificação do ataque *wormhole* está relacionado com a utiliza do protocolo RPL, uma vez que este é relacionado como ameaça de roteamento. *Wormhole* tem como base comunicação ICMP. Já a identificação do ataque de *flooding* está relacionado com a comunicação entre os nós vizinhos.

Ao ser extraído as informações dos pacotes ICMPv6 é iniciado a análise dos comportamento dos dispositivos. Após a extração, é realizada a checagem se o endereço dos nós que estão enviando as mensagens estão inseridos na *black list*. Caso seja detectado a a permanência na *black list*, as mensagens são descartadas, impedindo a comunicação entre os nós.

| | Identified as Malicious | Identified as Benign |
|---|---|---|
| Malicious ICMPv6 | TP | FP |
| Benign ICMPv6 | FN | TN |

**Figura 5.28:** Matriz de Confusão

A Figura 5.28 foi utilizada pelos autores para mostrar que as mensagens correspondentes ao ataque *wormhole* foram identificadas corretamente. As sigla TP corresponde à verdadeiro-positivo e FP corresponde à falso-negativo.

$$SP = \frac{TP}{TP + FP} \quad (1)$$

$$SR = \frac{TP}{TP + FN} \quad (2)$$

$$A = \frac{TN + TP}{TP + FN + FP + TN} \quad (3)$$

$$MR = \frac{FP}{TN + FN} \quad (4)$$

**Figura 5.29:** Cálculo de Medidas

Para a avaliação do sistema desenvolvido foram utilizados quatro tipos de medidas: *precision* (SP), *recall* (SR), *accuracy* (A) e *miss rate* (MR). A Figura 5.29 mostra como é realizado o calculo de cada medida.

As Figuras 5.30 e 5.31 correspondem aos parâmetros de simulação utilizados e o resultados obtidos, respectivamente.

| Map Size | 500*500 (M) |
|---|---|
| Number of nodes | 100 |
| Transmission distance | 100 M |
| Distance of wormhole tunnel | 200 M |

**Figura 5.30:** Parâmetros de Simulação

| SP | 100% |
|---|---|
| SR | 100% |
| A | 100% |
| MR | 0 |

**Figura 5.31:** Parâmetros de Simulação

## 5.5 Síntese

Neste Capitulo, os elementos utilizados para a realização da avaliação do IDS-IoT foram apresentados. O ambiente de teste utilizado foi apresentado, em que foram utilizados quatro dispositivos *Raspberry Pi* conectados a um *Raspberry Pi* configurado como *gateway* para o compartilhamento da Internet via *wireless* para os demais dispositivos conectados.

Os dados de testes utilizado foram apresentados, em que foi implementado um serviço *web* de *stream* de vídeo utilizando o servidor *Apache*. Tal implementação tem por objetivo inserir pacotes na rede para que se possa ter uma melhor avaliação do desempenho do IDS-IoT.

Dando continuidade, na subseção 5.3 foram expostos o resultados obtidos ao ser realizados os tipos de ataques no qual o IDS-IoT foi implementado para detectar, bem como a geração das regras *iptables* correspondentes.

Por fim, foram apresentados os trabalhos relacionados utilizados durante o desenvolvimento do sistema proposto. As particularidades de cada trabalho foram apresentadas. A Tabela 5.4 faz um comparativo de cada trabalho desenvolvido com o IDS-IoT.

Na Tabela 5.4 é possível observar que, embora existam trabalhos sobre IDS para IoT, estes estão limitados à detecção de apenas um ou dois tipos de ameaças. O IDS-IoT tem a possibilidade de realizar a detecção de até cinco tipos de ameaças.

**Tabela 5.4:** Tabela Comparativa

| | Kasinathan *et al*. [20] | Sonar e Upadhyay [37] | Machaka *et al*. [24] | Chen *et al*. [9] | IDS-IoT |
|---|---|---|---|---|---|
| Sistema Operacional | *Contiki* | *Contiki* | Não informado | Não informado | *Raspbian* |
| *Syn Flood* | - | - | ✓ | - | ✓ |
| *Land Attack* | - | - | - | - | ✓ |
| *Smurf Attack* | - | - | - | - | ✓ |
| *ICMP Flood* | - | - | - | ✓ | ✓ |
| *UDP Flood* | ✓ | ✓ | - | - | ✓ |
| *Wormhole* | - | - | - | ✓ | - |

# 6 Conclusão

Como conclusão, será apresentado neste capítulo os objetivos alcançados com o desenvolvimento do IDS-IoT, bem como as limitações identificadas nos testes realizados.

Em seguida, os trabalhos futuros são mencionados a fim de tentar suprir as limitações do IDS-IoT. Por fim, a publicação realizada é apresentada, no qual as informações do evento são salientadas.

## 6.1 Objetivos alcançados

Tomando como referência os IDS existentes para a IoT, mencionados na seção 5.4, o presente trabalho apresentou como principal contribuição a identificação dos ataques DoS descritos no Capitulo 5 desta dissertação.

O sistema desenvolvido (IDS-IoT) fez uso de uma arquitetura que é composta por 5 (cinco) camadas. Da camada inferior à superior temos: *Geracao_Regra*, *Deteccao_Ataque*, *Analisador_Pacote*, *Sensor* e *DB_Packs* que se interliga com todas as demais camadas.

*Geracao_Regra* objetiva gerar as regras do *firewall* utilizado para barrar os pacotes de origem de eventuais atacantes identificados, no qual a regra irá barrar os pacotes de qualquer tipo de protocolo enviado pelo atacante, após a geração da regra.

*Deteccao_Ataque* recebe da camada *Analisador_Pacote* uma resposta se os pacotes capturados pelo IDS-IoT são característicos de um ataque DoS, ou não. Caso a análise seja positiva, esta camada fica responsável por identificar qual tipo de ataque está a ocorrer.

Como mencionado, *Analisador_Pacote* analisa todos os pacotes capturados. Ao ser capturos, é analisado as informações de todos os pacotes e verificado se estes são correlatos às características dos ataques implementados. Um vez identificado, *Deteccao_Ataque* identifica com qual tipo de ataque o dispositivo está sendo atacado.

A captura dos pacotes que estão trafegando na rede fica por responsabilidade da camada *Sensor*. No qual as informações de tais pacotes são inseridas no banco de dados (camada *DB_Packs*).

Retomando os objetivos abordados na seção 1.5 desta dissertação, destacam-se os seguintes objetivos alcançados no decorrer da elaboração do presente trabalho:

- Os testes realizados no cenário utilizado demonstrou que o IDS-IoT obteve resultados favoráveis na detecção dos tipos de ataques implementados, embora a ocorrência de alertas falsos-positivos tenha sido detectado nos testes realizados;
- A constante atualização das regras do *firewall iptables* é outro fator importante, pois à medida que um novo ataque é identificado é gerado um nova regra correspondente à ameaça detectada;
- Outro importante objetivo alcançado foi a utilização mínima de recursos (memória e processamento) ao executar o IDS-IoT em um dispositivo (*Raspberry Pi*) que possui disponibiliza de baixos recursos;

  — Embora tenha sido identificado o baixo consumo de memória e processamento, IDS-IoT apresentou altos índices de utilização de recursos I/O, que a depender em qual cenário este será empregado, poderá influenciar no desempenho do dispositivo.

Mesmo que os objetivos tenham sido atingidos, é sabido que IDS-IoT apresenta limitações que necessitam uma maior atenção. Para tanto, a seção 6.2 apresenta as limitações do IDS-IoT identificadas.

## 6.2 Limitações e Trabalhos Futuros

Como IDS-IoT trata-se de um Sistema de Detecção de Intrusão baseado em anomalias, este está habilitado apenas para a detecção dos ataques implementados durante o desenvolvimento do sistema proposto nesta dissertação.

Como um dos trabalhos futuros, propomos fazer uso de aprendizado de máquina para determinar o valor ideal para a variável *max_pacotes* (Figura 4.12) conforme o ambiente no qual IDS-IoT será utilizado.

De fato, o uso de adaptabilidade ao contexto (largura de banda, quantidade de dispositivos, frequência na utilização da porta de comunicação, tipo de serviço ligado a porta de comunicação, dentre outros, são exemplos de contexto) é de suma importância para que ocorra a redução da geração de alertas falsos-positivos, como pode ser visualizado nos cenários utilizados.

IDS-IoT será utilizado em uma rede com um número maior de dispositivos, assim podemos ter uma analise mais ampla, uma vez que este foi testado em um laboratório de pesquisas, ao ser aplicado em cenários reais os resultados obtidos poderão ter uma variação significante.

Em cenários reais a quantidade de ataques implementados pode demonstra-se ineficiente. Com isso, outros tipos de identificação serão acrescentadas ao IDS-IoT, tais como: *Ping of Death*, *Probe attack* e *Zero Day*. Os ataques mencionados são apenas exemplos de ataques que podem ser implementados em um trabalho futuro.

## 6.3 Publicações

O presente trabalho gerou uma aceitação de um artigo em uma conferência que ocorrerá nos dias 22 e 23 de março de 2017, no Reino Unido. O evento acontecerá na Universidade de Cambridge. A conferência é intitulada de *International Conference on Internet of Things, Data and Cloud Computing* (ICC).

O artigo mencionado é intitulado de *An Intrusion Detection System for Denial of Service Attack Detection in Internet of Things*. Os resultados apresentados no artigos são referentes aos resultados preliminares obtidos durante o desenvolvimento da pesquisa.

# Referências Bibliográficas


[1] B. Q. M. AL-Musawi. Mitigating dos/ddos attacks using iptables. *International Journal of Engineering & Technology*, 12(3), 2012.

[2] S. Babar, A. Stango, N. Prasad, J. Sen, and R. Prasad. Proposed embedded security framework for internet of things (iot). In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on*, pages 1–5. IEEE, 2011.

[3] B. Barr, S. Yoo, and T. Cheatham. Network monitoring system design. *ACM SIGCSE Bulletin*, 30(1):102–106, 1998.

[4] T. Bhattasali, R. Chaki, and N. Chaki. *Study of Security Issues in Pervasive Environment of Next Generation Internet of Things*, pages 206–217. 2013.

[5] CERT.br. *Cartilha de Segurança para Internet*, page 410. Comitê Gestor da Internet no Brasil, São Paulo, 2012.

[6] CERT.br. Ataques na internet, Disponível em: http://cartilha.cert.br/ataques/. Acessado em: 16 de setembro de 2016.

[7] Cert.org. Tcp syn flooding and ip spoofing attacks, Disponível em: http://www.cert.org/historical/advisories/CA-1996-21.cfm. Acessado em: 16 de outubro de 2016.

[8] C. Cervantes, D. Poplade, M. Nogueira, and A. Santos. Detection of sinkhole attacks for supporting secure routing on 6lowpan for internet of things. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 606–611. IEEE, 2015.

[9] C.-M. Chen, S.-C. Hsu, and G.-H. Lai. Defense denial-of service attacks on ipv6 wireless sensor networks. In *Genetic and Evolutionary Computing*, pages 319–326. Springer, 2016.

[10] T. Clausen, U. Herberg, and M. Philipp. A critical evaluation of the ipv6 routing protocol for low power and lossy networks (rpl). In *2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 365–372. IEEE, 2011.

[11] G. Cugola and A. Margara. Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys (CSUR)*, 44(3):15, 2012.

[12] DevMedia. Sistema de detecção de intrusão - artigo revista infra magazine 1, Disponível em: http://www.devmedia.com.br/sistema-de-deteccao-de-intrusao-artigo-revista-infra-magazine-1/20819. Acessado em: 8 de setembro de 2016.

[13] D. Evans. A internet das coisas: Como a próxima evolução da internet está mudando tudo. *Cisco Internet Business Solutions Group (IBSG)*, 2011.

[14] A. A. Gendreau. Situation awareness measurement enhanced for efficient monitoring in the internet of things. In *Region 10 Symposium (TENSYMP), 2015 IEEE*, pages 82–85, 2015.

[15] Gitspot.com, Disponível em: http://jpcap.gitspot.com/. Acessado em: 27 de dezembro de 2016.

[16] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita. Botnet in ddos attacks: trends and challenges. *IEEE Communications Surveys & Tutorials*, 17(4):2242–2270, 2015.

[17] F. Hu. *Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations*. CRC Press, 2016.

[18] C. Jun and C. Chi. Design of complex event-processing ids in internet of things. In *2014 Sixth International Conference on Measuring Technology and Mechatronics Automation*, pages 226–229. IEEE, 2014.

[19] P. Kasinathan, G. Costamagna, H. Khaleel, C. Pastrone, and M. A. Spirito. Demo: An ids framework for internet of things empowered by 6lowpan. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 1337–1340. ACM, 2013.

[20] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits. Denial-of-service detection in 6lowpan based internet of things. In *WiMob*, pages 600–607, 2013.

[21] V. H. La, R. Fuentes, and A. R. Cavalli. A novel monitoring solution for 6lowpan-based wireless sensor networks. In *Communications (APCC), 2016 22nd Asia-Pacific Conference on*, pages 230–237. IEEE, 2016.

[22] S. Li, L. Da Xu, and S. Zhao. The internet of things: a survey. *Information Systems Frontiers*, 17(2):243–259, 2015.

[23] N. Lin and W. Shi. The research on internet of things application architecture based on web. In *Advanced Research and Technology in Industry Applications (WARTIA), 2014 IEEE Workshop on*, pages 184–187. IEEE, 2014.

[24] P. Machaka, A. McDonald, F. Nelwamondo, and A. Bagula. Using the cumulative sum algorithm against distributed denial of service attacks in internet of things. In *International Conference on Context-Aware Systems and Applications*, pages 62–72. Springer, 2015.

[25] C. M. Medaglia and A. Serbanati. *An Overview of Privacy and Security Issues in the Internet of Things*, pages 389–395. Springer New York, New York, NY, 2010.

[26] J. M. M. L. Mendes. Security techniques for the internet of things. Master's thesis, Universidade de Aveiro, 2013.

[27] L. C. O. Moraes. Framework de comunicação seguro e confiável para internet das coisas usando o protocolo xmpp. Master's thesis, Universidade Federal do Maranhão, 2016.

[28] Netfilter.org, Disponível em: https://www.netfilter.org/. Acessado em: 9 de setembro de 2016.

[29] O. Neto et al. Síntese de requisitos de segurança para internet das coisas baseada em modelos em tempo de execução. 2015.

[30] H. Ning, H. Liu, and L. T. Yang. Cyberentity security in the internet of things. *Computer*, (4):46–53, 2013.

[31] M. Ozsoy, K. N. Khasawneh, C. Donovick, I. Gorelik, N. Abu-Ghazaleh, and D. V. Ponomarev. Hardware-based malware detection using low level architectural features.

[32] P. Pongle and G. Chavan. Real time intrusion and wormhole attack detection in internet of things. *International Journal of Computer Applications*, 121(9), 2015.

[33] D. Puthal, S. Nepal, R. Ranjan, and J. Chen. Threats to networking cloud and edge datacenters in the internet of things. *IEEE Cloud Computing*, 3(3):64–71, 2016.

[34] S. Raza, L. Wallgren, and T. Voigt. Svelte: Real-time intrusion detection in the internet of things. *Ad hoc networks*, 11(8):2661–2674, 2013.

[35] A. Riahi, Y. Challal, E. Natalizio, Z. Chtourou, and A. Bouabdallah. A systemic approach for iot security. In *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, pages 351–355. IEEE, 2013.

[36] R. Roman, J. Zhou, and J. Lopez. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10):2266–2279, 2013.

[37] K. Sonar and H. Upadhyay. An approach to secure internet of things against ddos. In *Proceedings of International Conference on ICT for Sustainable Development*, pages 367–376. Springer, 2016.

[38] J. A. Stankovic. Research directions for the internet of things. *IEEE Internet of Things Journal*, 1(1):3–9, 2014.

[39] Symantec. Internet security threat report, Disponível em: https://www.symantec.com/pt/br/security_response/publications/threatreport.jsp. Acessado em: 29 de setembro de 2016.

[40] Tcpdump.org, Disponível em: http://www.tcpdump.org/. Acessado em: 26 de dezembro de 2016.

[41] N. K. Thanigaivelan, E. Nigussie, R. K. Kanth, S. Virtanen, and J. Isoaho. Distributed internal anomaly detection system for internet-of-things. In *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 319–320. IEEE, 2016.

[42] R. Tomasi, L. Bruno, C. Pastrone, and M. Spirito. Meta-exploitation of ipv6-based wireless sensor networks. In *3rd international workshop on Security and Communication Networks-IWSCN,(Gjøvik-Norway)*, 2011.

[43] M. M. Vitali and H. F. A. d. Silva. Sistema inteligente de detecção de intrusão, 2008.

[44] M. S. Wangham, M. C. Domenech, and E. R. de Mello. Infraestrutura de autenticaçao e de autorizaçao para internet das coisas. *Minicursos do XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais—SBSeg*, 2013.

[45] R. H. Weber and R. Weber. *Internet of Things*, volume 12. Springer, 2010.

[46] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun, and H.-Y. Du. Research on the architecture of internet of things. In *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, volume 5, pages V5–484. IEEE, 2010.

[47] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *IEEE Internet of Things Journal*, 1(1):22–32, 2014.