Universidade do Minho
Escola de Engenharia

Tiago Filipe Mesquita da Cunha

Deteção de Falsos Alertas de Intrusão em Redes de Computadores

Dissertação de Mestrado
Mestrado Integrado em Engenharia Eletrónica Industrial e Computadores

Trabalho efetuado sob a orientação do
Professor Doutor Henrique Santos
Professor Doutor Sérgio Lopes

janeiro de 2019

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho

Atribuição-NãoComercial-CompartilhaIgual
CC BY-NC-SA
https://creativecommons.org/licenses/by-nc-sa/4.0/

AGRADECIMENTOS

Este trabalho de dissertação representa o término de uma importante etapa de formação na minha vida, com as mais variadas aprendizagens, tanto a nível pessoal como académico. Para tal, devo e quero expressar, neste pequeno texto, a minha gratidão a todos os intervenientes, que direta ou indiretamente contribuíram para este caminho efetuado.

Em primeiro lugar, agradeço aos meus pais, pela oportunidade que me foi concedida de continuar a formação académica e enveredar no ensino superior, assim como pelo acompanhamento, paciência e apoio. De igual modo, um especial agradecimento à minha namorada, pelo carinho, conforto e incentivo, nestas e noutras circunstâncias de maior pressão, que tanto ajudaram a enfrentar as adversidades.

Ao Professor Doutor Henrique Santos, orientador da dissertação, deixo uma palavra de reconhecimento e gratidão, pela ajuda, compreensão, disponibilidade e conhecimento partilhado, em todo o processo de investigação efetuado. Pela oportunidade do paradigma de dissertação e pelo interesse na mesma, agradeço igualmente ao coorientador Professor Doutor Sérgio Lopes.

A todos os meus colegas de curso, em especial aqueles que lidaram comigo dia após dia, o meu obrigado pela amizade, companheirismo, interajuda e bons momentos partilhados.

Por último, mas não menos importante, um agradecimento geral à instituição Universidade do Minho, em especial o departamento de Eletrónica, seus Docentes e pessoal não Docente, por se esforçarem em proporcionar a melhor formação possível dos seus alunos.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Deteção de Falsos Alertas de Intrusão em Redes de Computadores

RESUMO

Nos dias que correm, assiste-se a um aumento significativo e constante dos sistemas de informação e, incontornavelmente, ao acréscimo de dados sensíveis no espaço *online*. Essa informação passa a ser mais facilmente armazenada, acedida e trabalhada pelas entidades fidedignas, mas, por outro lado, passa a estar mais suscetível a ataques, intrusões ou acessos indevidos.

Desta forma, a segurança informática é cada vez mais um assunto fulcral e imprescindível e, apesar de existirem já alguns métodos, técnicas e sistemas capazes de identificar e bloquear a proliferação das ditas intrusões, existem ainda lacunas e aspetos a melhorar constantemente nestas soluções.

Esta dissertação tem como finalidade estudar e avaliar uma das soluções para o problema mencionado: o sistema de deteção de intrusões (IDS). Estes sistemas têm como objetivo monitorar, identificar e notificar a ocorrência de atividades maliciosas ou suspeitas, que tenham como alvo e coloquem em risco ativos de tecnologia de uma rede de computadores.

Pretende-se comparar dois IDS, Snort *versus* Suricata, numa *framework* para o efeito, de forma a monitorizar os seus desempenhos, tanto ao nível da eficiência como da exatidão. Esta comparação servirá de alicerce para um melhor estudo dos alertas gerados por cada IDS, dando especial atenção aos falsos positivos.

Palavras-Chave: Desempenho, intrusões, segurança, sistemas de informação.

Detection of False Intrusion Alerts in Computer Networks

ABSTRACT

These days, there is a significant and constant increase in information systems and, inevitably, a growth of sensitive data in the online space. This information becomes more easily stored, accessed and handled by the trusted entities, but, on the other hand, it becomes more susceptible to attacks, intrusions or improper access.

In this way, computer security is increasingly a central and indispensable subject and, although there are already some methods, techniques and systems able to identify and block the proliferation of said intrusions, there are still gaps and aspects to constantly improve in these solutions.

This dissertation aims to study and evaluate one of the solutions to the mentioned problem: the intrusion detection system (IDS). These systems aim to monitor, identify and report malicious or suspicious activities, that target and threaten the technological assets of a computer network.

It is intended to compare two IDSs, Snort versus Suricata, in a framework for this purpose, in order to monitor their performance, both in terms of efficiency and accuracy. This comparison will serve as a foundation for a better study of the alerts generated by each IDS, paying attention to false positives.

Keywords:, Information systems, intrusions, performance, security.

ÍNDICE

ÍNDICE DE SIGLAS

***A**dress **R**esolution **P**rotocol*	ARP
***D**enial **o**f **S**ervice*	DoS
***D**istributed **D**enial **o**f **S**ervice*	DDoS
***D**esign **S**cience **R**esearch*	DSR
***D**omain **N**ame **S**erver*	DNS
***F**ile **T**ransfer **P**rotocol*	FTP
***G**eneral **P**ublic **L**icense*	GPL
***H**ost-Based **I**ntrusion **D**etection **S**ystem*	HIDS
***H**yper**T**ext **T**ransfer **P**rotocol*	HTTP
***I**nternet **C**ontrol **M**essage **P**rotocol*	ICMP
***I**nternet **P**rotocol*	IP
Internet **R**elay **C**hat	IRC
***I**ntrusion **D**etection **S**ystem*	IDS
***I**ntrusion **P**revention **S**ystem*	IPS
***K**ernel-based **V**irtual **M**achine*	KVM
***M**edia **A**ccess **C**ontrol*	MAC
***N**etwork **A**ddress **T**ranslation*	NAT
***N**etwork-Based **I**ntrusion **D**etection **S**ystem*	NIDS
***N**etwork **S**ecurity **M**onitoring*	NSM
Open **I**nformation **S**ecurity **F**oundation	OISF
Open **S**ource **S**ecurity **I**nformation **M**anagement	OSSIM
***P**rocess **Id**entifier*	PID
***S**ecurity **I**nformation and **E**vent **M**anagement*	SIEM
***S**imple **M**ail **T**ransfer **P**rotocol*	SMTP
***S**imple **N**etWork **M**anagement **P**rotocol*	SNMP
Sistema de **I**nformação	SI
***T**ransmisson **C**ontrol **P**rotocol*	TCP
***T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol*	TCP/IP
***U**ser **D**atagram **P**rotocol*	UDP
***R**everse **A**dress **R**esolution **P**rotocol*	RARP

ÍNDICE DE TABELAS

ÍNDICE DE FIGURAS

1. Enquadramento

"A popularização do computador nas décadas de 80 e 90 fez com que todo o mundo assistisse uma verdadeira revolução tecnológica" [1]. O computador passou a desempenhar um papel fundamental nas mais variadas circunstâncias e organizações. Os avanços resultantes dessa melhora tecnológica vão desde: as grandes empresas, que utilizam redes financeiras e sistemas de comunicação, até aos inúmeros benefícios e facilidades para toda a sociedade, mas, incontornavelmente, cada vez mais dependentes dos sistemas de informação [2].

Nos dias que correm, assiste-se a uma evolução das tecnologias computacionais, acompanhada pelo aumento significativo e constante dos SI e, incontornavelmente, ao acréscimo de dados sensíveis ao mundo online. Essa informação passa a ser mais facilmente armazenada, acedida e trabalhada pelas entidades fidedignas, mas, por outro lado, passa a estar mais suscetível a ataques, intrusões ou acessos indevidos. Desta forma, a "revolução" tecnológica a que temos vindo a assistir, introduziu facilidades por um lado, mas por outro acarreta também inúmeras ameaças e preocupações [3].

Ao mesmo tempo que as tecnologias de rede evoluem, a informação passa a ter um papel fulcral e valioso na sociedade, sendo por isso necessário a implementação de mecanismos para proteger os dados. Isto porque, a um ritmo elevado, surgem também novas ferramentas automatizadas, especificamente projetadas para a procura de falhas em sistemas informáticos, comprometendo a sua segurança.

"Atualmente, não é necessário ter conhecimentos avançados num dado sistema operativo ou protocolo, para explorar as suas vulnerabilidades, basta apenas fazer uma boa pesquisa pelo Google" [1].

1.1. Motivação

Com a constante evolução e modificação dos SI e respetivas tecnologias contíguas, cria-se a necessidade de implementar, estudar e melhorar mecanismos para a proteção de todo o tipo de dados.

Neste momento, são já várias as ferramentas que cooperam no aumento da segurança numa rede, tais como: a criptografia, que cria um nível de proteção para dados, o uso de Firewall, que através de políticas de segurança controla o tráfego de pacotes na entrada e saída da rede, a VPN com um túnel criptografado entre 2 pontos de rede, o sistema de deteção de intrusão (IDS), entre outras. Dentro deste universo de ferramentas, os IDS merecem um relevo especial, visto que

englobam os procedimentos de monitorizar, identificar e notificar um evento de atividades maliciosas, isto é, atividades não-autorizadas, que ponham em risco um sistema informático [1].

Um IDS auxilia e contribui no aumento da segurança dos sistemas informáticos, podendo trabalhar em conjunto com os restantes instrumentos de segurança informática. Isto porque, quando se interseta uma dada intrusão, através da supervisão do tráfego na rede, consegue-se mais facilmente corrigir ou evitar possíveis problemas de segurança, permitindo a atuação do gestor de rede no bloqueio do invasor e respetivo ataque, se necessário.

Os sistemas IDS são cada vez mais usados e investigados, ao mesmo tempo que recebem grandes investimentos, dentro da área da segurança de redes de computadores. Isto deve-se ao crescente aumento da quantidade de computadores interligados no mundo, apresentando-se como uma ótima oportunidade para intrusos, que exploram continuamente novos métodos de ataques, comprometendo seriamente a segurança destes sistemas informáticos.

Desta forma, torna-se pertinente o estudo, emprego, avaliação e comparação das soluções aplicadas no combate às intrusões, contemplando e salientando o sistema IDS. Contudo, estes sistemas possuem algumas dependências e dificuldades na sua instalação, configuração e execução, nomeadamente o NIDS Suricata [4]. Por outro lado, o seu emprego integrado num sistema com ferramentas de segurança de terceiros, como alguns gestores de eventos, apresenta também vantagens na sua utilização, como o tratamento dos dados relativos aos incidentes de segurança, podendo constituir uma mais-valia na análise da veracidade dos alertas gerados, sobretudo os falsos positivos [5].

Deste modo, a implementação e utilização de uma plataforma de trabalho virtualizada, voltada para o âmbito em questão, conseguirá colmatar algumas das dificuldades e particularidades mencionadas, conseguindo-se uma plataforma o mais completa e compatível possível para o uso de IDS, para além de constituir um espaço de execução isolado e mais seguro [6].

1.2. Objetivos

Esta dissertação tem como finalidade estudar alguns dos recursos existentes para o tema apresentado, concluindo com o teste, análise e comparação de dois IDS (Snort vs Suricata), num ambiente simulado e controlado através da virtualização de um sistema Linux. Contribuindo, se possível, com melhorias ou novas ideias na deteção de intrusões em redes de computadores.

De um modo geral, o trabalho centra-se na avaliação de IDS, na vertente da eficiência e eficácia, aferindo sobre o seu desempenho geral, enfatizando os falsos positivos nos alertas gerados.

Numa primeira fase, pretende-se analisar algumas das melhores ferramentas, algoritmos e sistemas desenvolvidos, escolhendo os que melhor se adequem e possibilitem a comparação do desempenho dos IDS. Desta forma e posteriormente, será possível recolher e estudar modelos de tráfego, assim como simular o comportamento geral dos sistemas detetores de intrusões, na estrutura desenvolvida e, propositadamente, explorar determinadas características, como a capacidade de processamento do tráfego, alertas gerados, discrepâncias e vulnerabilidades, dos mesmos.

Em suma, espera-se conseguir uma comparação e cruzamento consistente dos resultados dos dois sistemas IDS, que permita aferir e concluir acerca das potencialidades, vantagens, desvantagens e diferenças de cada sistema. Contribuindo, se possível, com algumas conclusões relevantes sobre os desempenhos dos mecanismos de deteções de intrusões estudados, sobretudo no que diz respeito aos falsos alertas.

1.3. Metodologia

Este trabalho de dissertação foi desenvolvido segundo o método de investigação Desenho de Ciência (*Design Science Research* – DSR). Este método de pesquisa surgiu e cresceu na engenharia e é, cada vez mais, aplicado em sistemas de informação.

Segundo Cruz [7], este processo de trabalho e desenvolvimento engloba seis atividades distintas e complementares: identificar o problema/tema e motivação; definir objetivos; desenvolvimento; demonstração; avaliação e comunicação.

Desta forma, depois de identificada a problemática de estudo e definidos objetivos, foi efetuado um levantamento do estado da arte: estudo das melhores ferramentas, com melhores características, desempenho e eficácia adequada ao propósito, vulnerabilidades e lacunas presentes, analise dos sistemas e algoritmos colaborativos e soluções existentes no mercado, entre outros aspetos relevantes à construção de uma boa base e estrutura, para simulação e deteção de intrusões e testes comparativos de IDS. Esta *framework* é o ponto de partida para todos os ensaios, simulações e aprimoramentos que se pretendem, sendo responsável pela agregação, comunicação e interação de todas as ferramentas e sistemas usados, incluindo por último e mais relevante o sistema de deteção de intrusão.

Após o levantamento e sistematização dos conceitos, processos e técnicas de trabalho, procedeu-se ao estudo experimental e simulações finais, recolhendo todos os resultados pretendidos e relevantes para o universo de estudo. Finaliza-se a investigação com a comparação, cruzamento e análise dos resultados obtidos, assim como as devidas conclusões.

1.4. Organização do Documento

O documento de dissertação encontra-se dividido em cinco capítulos distintos. Inicia-se com o enquadramento do tema escolhido, passando pelos conceitos e estado da arte, o desenvolvimento da experiência em si, resultados e por fim as conclusões do trabalho realizado.

O primeiro capítulo, em que se insere também o presente texto, é constituído pela contextualização do tema proposto para o trabalho de dissertação, assim como os principais objetivos, motivações e métodos de investigação do mesmo.

O segundo capítulo apresenta uma revisão bibliográfica, de todos os conceitos relevantes ao universo de estudo, assim como das soluções existentes e trabalhos efetuados na área.

Na terceira parte deste documento, encontra-se o desenvolvimento e explicação da estrutura de trabalho proposta e implementada, assim como dos vários testes realizados. São descritos os passos de implementação, ferramentas, métricas e *datasets* aplicados.

De seguida, no capítulo quatro, são expostos todos os resultados relevantes dos testes efetuados.

O quinto capítulo aborda a análise e discussão, individual e geral, dos resultados obtidos.

Por último, no sexto capítulo, apresenta-se a conclusão final, desde o trabalho inicial de investigação, até à experiência em si e resultados obtidos. Salientam-se, também, algumas lacunas do trabalho efetuado e, consequentemente, sugestões de aprimoramento e trabalho futuro.

2. Conceitos Teóricos

Neste segundo capítulo do documento, será exposto todo o conteúdo relevante para o universo de estudo. Começando pelas noções mais gerais e primárias, como a segurança da informação, até aos conceitos mais particulares e específicos, como os sistemas detetores de intrusão, finalizando com um apanhado das soluções e trabalhos existentes na área.

2.1. Segurança da Informação

De acordo com Netto e da Silveira [8], "a segurança da informação é o processo de proteção da informação das ameaças à sua integridade, disponibilidade e confidencialidade" e ainda, "(...) uma área do conhecimento dedicada à proteção de ativos da informação contra acessos não autorizados, alterações indevidas ou sua indisponibilidade".

Se pretendermos um padrão internacional da definição de segurança da informação, podemo-nos basear na norma *ISO / IEC 27002 (2005)*, que define segurança da informação "como a preservação da confidencialidade, integridade e disponibilidade de informações". De acordo com a mesma fonte, a informação pode adotar muitos formatos, desde a impressão e escrita em papel, transmissão por correio, eletronicamente ou em conversas, exibição cinematográfica, até ao armazenamento informático, entre outras [9].

Desta forma, podemos dizer que a segurança da informação é uma área do conhecimento, que pretende proteger o seu maior ativo, a informação, das ameaças à sua integridade, disponibilidade e confidencialidade, nas mais variadas formas em que esta se pode encontrar, minimizando os riscos e garantindo a continuidade da mesma. De seguida, pode-se perceber com maior detalhe estas três principais propriedades da informação:

- **Integridade**: Procura preservar a totalidade e exatidão dos dados, evitando que pessoas sem autorização a possam modificar ou destruir;
- **Disponibilidade**: Visa assegurar o contínuo e bom acesso de um sistema de informação, pelas pessoas devidamente autorizadas e bem-intencionadas. Para tal, é necessário travar ataques como o *distributed denial of service* (DDoS), que buscam ocupar os recursos de um dado SI de forma deliberada e mal-intencionada, levando à insuficiência na resposta do mesmo a pedidos autênticos e benignos;

- **Confidencialidade**: Assegura o sigilo da informação de uma determinada entidade, como patentes, projetos em desenvolvimento e dados pessoais, permitindo apenas o acesso de quem está devidamente autorizado [8], [10].

Contudo, a segurança da informação não é algo tão objetivo e simples assim, como proteger um determinado elemento. De acordo com os especialistas no tema, a segurança da informação abrange seis elementos fundamentais, que merecem especial relevância e devem ser protegidos. São eles: o pessoal, os dados, os procedimentos, o *software*, o *hardware* e as redes [10].

Na Figura 1, pode-se observar uma ilustração dos conceitos fundamentais da segurança da informação, abordados até aqui.

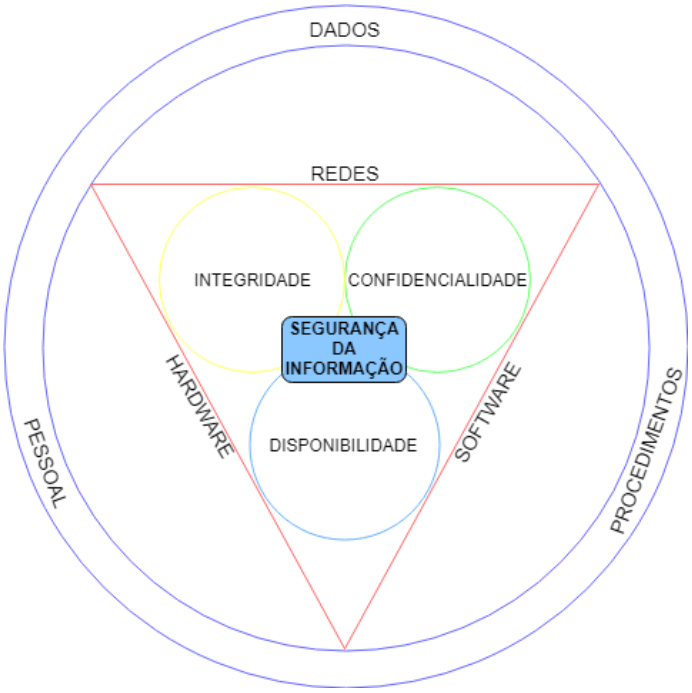


Figura 1 - Componentes da Segurança da Informação[1]

2.1.1. Cibersegurança

Normalmente, ouve-se falar em cibersegurança e segurança da informação de forma interligável e similar. Na maioria dos casos, a cibersegurança é sinónimo de segurança da informação e os incidentes que possam ocorrer na primeira mencionada, também podem ser

[1] Adaptado de Wikipédia [65].

descritos e caracterizados igualmente para a segunda. Portanto, uma violação da cibersegurança conduziria a uma violação da confidencialidade, integridade ou disponibilidade de informações.

Contudo e segundo Von Solms e Van Niekerk [9], "existem ameaças de cibersegurança que não fazem parte do escopo formalmente definido de segurança da informação". A mesma fonte apresenta ainda alguns exemplos elucidativos destas situações:

- *Cyber bullying*: Onde "o alvo de tais atividades é o próprio utilizador e assim, o *cyber bullying* resulta em dano direto à pessoa que está sendo intimidada". Deste modo, a intimidação no ciberespaço não representa uma falha na integridade, confidencialidade ou disponibilidade da informação;
- Automação residencial: "Mais uma vez, neste caso, pode-se argumentar que as informações da vítima não são necessariamente afetadas negativamente. Em vez disso, outros ativos da vítima são o alvo do cibercrime", como o acesso indevido aos sistemas inteligentes que possui na casa, desde os menos inofensivos como o frigorífico ou *Smart TVs*, até aos mais sensíveis como o sistema de segurança;
- Mídias digitais: "Todos os anos enormes quantidades de receita potencial são perdidas para a partilha de filmes ilegais, música e outras formas de mídia digital". Este é, novamente, um caso onde as propriedades da informação não são, necessariamente, postas em causa, mas sim a propriedade intelectual do autor;
- *Cyber* terrorismo: Neste cenário, os ataques podem ser direcionados a uma infraestrutura crítica, como a rede energética de um país, resultando não só na possível perda da integridade ou disponibilidade de recursos de informação, mas também do acesso a esses serviços críticos. "Nesse caso, não é a própria informação nem o utilizador da informação individual que está em risco, mas sim o bem-estar da sociedade como um todo".

Os cenários, acima apresentados, evidenciam alguns aspetos particulares da cibersegurança, onde os interesses pessoais, organizacionais ou nacionais, incluindo ativos que não são diretamente informações, carecem de proteção dos riscos resultantes da interação com o ciberespaço. Assim e como foi referido anteriormente, consegue-se demonstrar algumas possíveis diferenças entre segurança da informação e cibersegurança.

2.1.2. Gestão do Risco

A gestão do risco, uma área em constante desenvolvimento, procura criar mecanismos de controlo para proteger recursos ou ativos de valor. "Um processo de gestão de risco define um conjunto de atividades para suportar a identificação e mitigação de riscos num contexto específico." Se essa identificação dos riscos for rápida o suficiente, há uma maior facilidade na criação de estratégias de combate e redução das potenciais ameaças.

Como foi dito acima, o foco da gestão do risco é a definição de controlos apropriados para tentar travar ameaças, eliminar vulnerabilidades ou, em último caso, diminuir o impacto da ocorrência de um risco, partindo da premissa de que o risco advém de uma vulnerabilidade, que uma determinada ameaça consegue explorar.

Os dados ou ativos de valor a proteger podem variar bastante, dependendo essencialmente da natureza em que se insere, abrangendo desde entidades ou organizações físicas (pessoas, edifícios) até sistemas informacionais e processos. "Quando uma vulnerabilidade é explorada, é introduzido um impacto na obtenção de objetivos da organização, reduzindo-lhe o valor" [11].

Como referido anteriormente, esta é uma área em evolução, onde a base de conhecimento tem vindo a crescer e a relacionar-se com várias áreas de investigação. Entre elas e de forma interligada, podemos falar da gestão da segurança da informação e no papel fundamental que a gestão do risco desempenha nesta. Desta forma e no seguimento, será apresentado um resumo do modelo genérico do processo de gestão do risco, baseado na norma ISO/IEC 27005[2], contendo as seguintes etapas [12]:

- Identificação do contexto: Limita-se o campo de ação do processo, assim como os seus objetivos;
- Análise e identificação dos riscos: Calcula-se o nível de risco, de acordo com o conhecimento acerca dos recursos e fontes de ameaças;
- Avaliação e valorização do risco: Atribui-se um valor ao risco, seguindo um critério previamente definido;

[2] (ISO / IEC 27005: 2018 Tecnologia da informação - Técnicas de segurança - Gestão de risco de segurança da informação, série ISO 27000): Documento estabelecido por unanimidade, pela organização internacional, não-governamental e independente, ISO (International Organization for Standardization), que fornece diretrizes para a gestão de riscos de segurança da informação. Este documento assenta nos conceitos gerais especificados na norma ISO / IEC 27001. (disponível em https://www.iso.org/standard/75281.html; consultado a 15/12/18)

- Tratamento do risco: Escolhe-se e implementa-se ações de controlo, destinadas a diminuir o risco;
- Consciência do risco remanescente: Assume-se a existência de riscos não mitigados, no final do processo;
- Pós-aceitação: Inicia-se a monitorização de todos os controlos de segurança implementados, avaliando a eficiência e eficácia dos controlos;
- Comunicação (tarefa contínua): Interação entre todos os ativos, com vista na troca de informações.

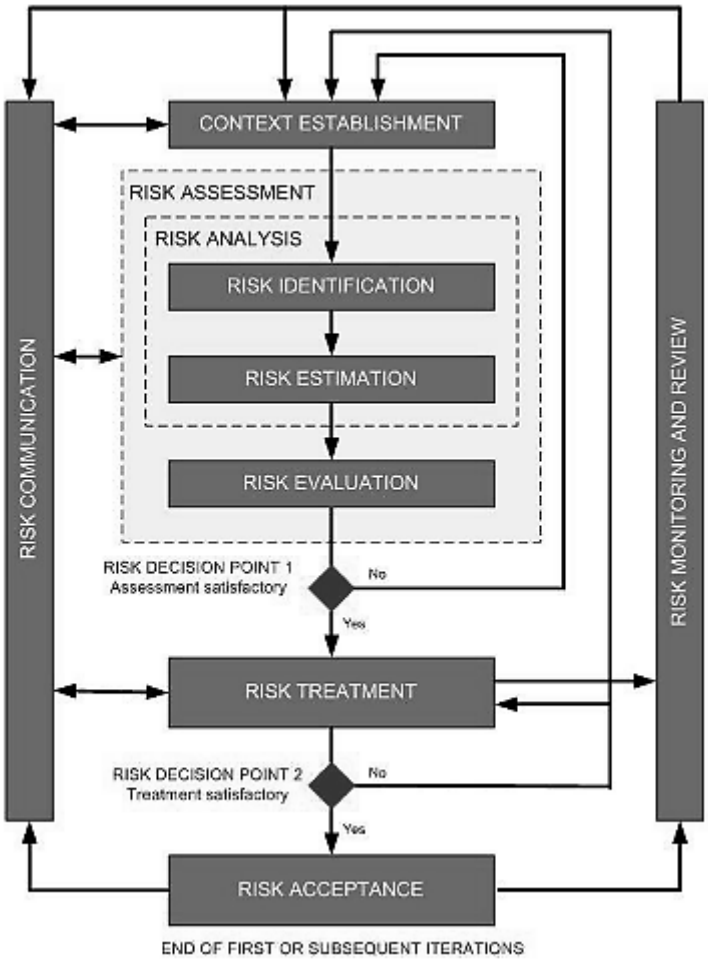


Figura 2 - Etapas no processo de gestão do risco (norma ISO/IEC 27005) [12]

2.1.3. Mecanismos e Controlos de Segurança

Depois de analisados os riscos, as suas consequências e o processo de gestão dos mesmos, deve-se implementar os controlos de segurança, em pontos estratégicos. Procura-se

garantir a maior redução possível dos riscos, partindo do pressuposto de que a anulação dos mesmos não é garantida [13].

Assim, os controlos e mecanismos de segurança podem ser tanto físicos como lógicos.

Os controlos físicos são barreiras, que dificultam o contacto direto com a informação ou as infraestruturas que a suportam. Estes mecanismos de segurança podem ser:

- Portas;
- Fechaduras;
- Paredes;
- Blindagem;
- Guardas, entre outros.

Os controlos lógicos impedem ou limitam o acesso à informação, geralmente em ambiente controlado e eletrónico, de forma a que a mesma não fique exposta a riscos por acessos indevidos. Os mecanismos de segurança, que auxiliam os controlos lógicos, são os seguintes:

- Criptografia;
- Assinatura digital;
- Garantia da integridade da informação;
- Controlo de acesso;
- Certificação;
- Autenticidade;
- *Honeypot*;
- Ferramentas e sistemas de segurança, como os detetores de intrusões, antivírus, *firewalls* e filtros antisspam [14].

Como referido anteriormente, depois de conhecidos os diversos mecanismos e controlos de segurança, estes podem e devem ser aplicados a certos pontos, de forma estratégica e nos seguintes moldes:

1. Políticas de Segurança da informação: Este ponto pretende apoiar toda a organização do que será executado, como a ação de cada elemento ou a abordagem à política de segurança na organização. "Estas políticas são, geralmente, construídas a partir das necessidades da organização e aperfeiçoadas pala experiência do gestor de segurança da informação, que deve transformar o seu trabalho em algo prático, objetivo e que tenha valor corporativo";

2. Organização da Segurança da Informação: Neste caso, são tomadas e organizadas medidas, pelo gestor de segurança da informação, de forma a serem implementadas na entidade, para que a mesma possa estar em segurança;
3. Gestão e Controlo de Ativos: Este é um ponto de controlo e prevenção, sobre tudo aquilo que tenha valor para a entidade em causa. São concebidas estratégias, pelo gestor de segurança da informação, "(..)que visam gerir e controlar o acesso de funcionários, bem como definir o que cada profissional pode acessar de informação da empresa". É ainda imprescindível, para uma boa gestão de ativos, que exista um inventário destes, um profissional responsável por cada ativo e ainda uma política de classificação da importância da informação;
4. Segurança em Recursos Humanos: O gestor de segurança da informação e o departamento de recursos humanos devem estabelecer as seguintes normas: "(...) a realização de análises de idoneidade pessoal e profissional das pessoas que pleiteiam uma vaga na empresa; definir uma política de confidencialidade ou código de ética entre trabalhadores e organização; realizar treinos em segurança da informação para todos os funcionários e não apenas para os profissionais de tecnologia da informação; definir uma política que dê acesso a funcionários ativos e que solicite a remoção de profissionais desligados da empresa". Deste modo, todos os funcionários, colaboradores e a área de recursos humanos têm responsabilidade pela segurança da Informação;
5. Segurança Física e do Ambiente: Este ponto abrange os meios que regulam o controlo de acesso, ao mesmo tempo que antecipa e previne danos, como tempestades, furacões, sismos, acidentes, roubos, entre outros, minimizando o risco de acontecer perda, dano ou adulteração de informação;
6. Gestão das Operações e Comunicações: A comunicação é um ponto relevante para o sucesso na divulgação de políticas. "(...) esta provoca alteração no *status quo* de praticamente todos os colaboradores". Assim, "(...) obriga a mudanças na forma de trabalho e qualquer mudança gera resistência, sendo a comunicação a melhor maneira de reduzir os conflitos inerentes a ela";
7. Controlo de Acessos: Estratégias, no processo de gestão da segurança da informação, que promovem o controlo de acesso, através de permissões, tanto dos sistemas informáticos como das pessoas da organização;

8. Aquisição, Desenvolvimento e Manutenção de Sistemas de Informação: Consiste na obtenção dos instrumentos capitais ao desenvolvimento de um sistema, assim como na revisão e manutenção periódica, contribuindo para o aprimoramento do projeto;
9. Gestão da Continuidade do Negócio: É o ponto responsável pela planificação e continuidade da segurança da informação "a longo prazo", ao mesmo tempo que evita que "(...) haja interrupção das atividades da empresa" e "(...) dos processos críticos", garantindo, em último caso, "(...) a retoma da empresa num tempo consideravelmente rápido". Este processo deve garantir as seguintes metas: Análise do risco no "plano de continuidade do negócio", "desenvolvimento, revisão e testes, no mínimo anualmente", na gestão de pessoas e na administração da tecnologia da informação;
10. Conformidade Legal: É o ponto de controlo reservado a tratar "(...) as leis, normas e requisitos de segurança". Este tipo de controlo deve garantir "o cumprimento das leis e outras legislações vigentes", "zelar pelos direitos de propriedade de todas as aplicações de informática da empresa" e efetuar a "verificação de pontos críticos de melhorias no ambiente da empresa" [13].

2.2. Redes de Computadores

As redes de computadores são, atualmente, indispensáveis a todas as organizações, empresas e sociedade como um todo. O modelo de referência e mais amplamente usado para este tipo de comunicação é o TCP/IP (*Transmission Control Protocol/Internet Protocol*). Este conjunto de protocolos permite a partilha de informação de modo rápido, eficiente e a longas distâncias. Para tal, é usada uma "pilha protocolar", "portável e independente da plataforma usada", possibilitando a ligação e comunicação entre diferentes sistemas e redes. O modelo protocolar TCP/IP é reconhecido como o "protocolo da *Internet*" [15], [16].

Segundo Aurélio e Alencar [16], a pilha protocolar do modelo TCP/IP é constituída por quatro camadas, como se vê na Figura 3:

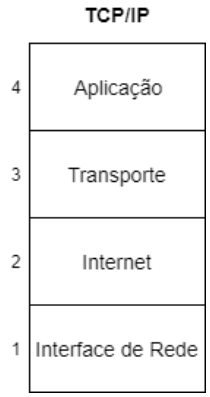


Figura 3 - Pilha protocolar do modelo TCP/IP[3]

- **Camada de Acesso à Rede ou Interface de Rede:** É a primeira camada da pilha, do modelo TCP/IP, com a finalidade de apoiar a camada de rede (*Internet*), utilizando os meios de acesso físico e lógico;
- **Camada de *Internet* ou Inter-rede:** Esta camada é responsável pelo envio dos pacotes de dados (*datagrams*), da origem para o destino (entre computadores). Consoante o percurso, condições e localizações na rede, é calculado a melhor rota para o encaminhamento dos dados;
- **Camada de Transporte:** Este nível é encarregado da comunicação entre máquinas. É, também, responsável por fornecer suporte à camada de aplicação e pelo controlo de fluxo. Este ocorre, de forma fiável e com garantia da entrega de pacotes, quando usado o protocolo TCP orientado à conexão, ou de forma não fiável e sem garantia da entrega dos dados, no caso do protocolo UDP (*User Datagram Protocol*) não orientado à conexão.
- **Camada de Aplicação:** O último nível do modelo TCP/IP é o de aplicação. Nesta camada, são especificados os protocolos, descritos de seguida, que apoiam as aplicações de rede dos utilizadores [15], [16].

Como referido acima, existem vários protocolos, que servem e suportam as várias aplicações e serviços de rede, destacando-se os seguintes:

- HTTP (*HyperText Transfer Protocol*): Empregado sobretudo no acesso de dados na *internet*. "Este protocolo permite a transferência de dados na forma de textos simples, hipertextos, áudios, vídeos, entre muitas outras";
- SMTP (*Simple Mail Transfer Protocol*): Utilizado como "mecanismo padrão" na troca de correspondência eletrónica (emails);

[3] Adaptado de Aurélio e Alencar [16].

- FTP (*File Transfer Protocol*): É o protocolo padrão de transferência de arquivos na *internet*, que possibilita a cópia de dados entre *hosts* (computadores/máquinas hospedeiras);
- SNMP (*Simple NetWork Management Protocol*): Protocolo responsável pela administração da *internet*;
- DNS (*Domain Name Server*): Trata-se de um protocolo de aplicação, que identifica endereços IP's e mantém uma tabela com endereços, de alguns caminhos de redes;
- TCP (*Transmission Control Protocol*): Protocolo responsável por oferecer um serviço orientado à conexão e confiável, entre aplicações;
- UDP (*User Datagram Protocol*): Ao contrário do protocolo TCP, o UDP não é orientado à conexão e, por isso, menos fiável. "Conhecido pela característica de ser um protocolo otimista, ou seja, ele envia todos os seus pacotes, acreditando que eles chegarão sem problemas e em sequência ao destino";
- IP (*Internet Protocol*): Principal protocolo da *Internet* (ou camada Inter-rede), no modelo TCP/IP;
- ICMP (*Internet Control Message Protocol*): Responsável por fornecer mensagens de controlo, entre nós, numa comunicação TCP/IP;
- ARP (*Adress Resolution Protocol*): Efetua o mapeamento de um endereço IP no respetivo endereço MAC (*Media Access Control*);
- RARP (*Reverse Adress Resolution Protocol*): Como o nome sugere, faz o inverso do protocolo anterior, ou seja, associa um endereço MAC conhecido a um endereço IP [16].

2.2.1. Ameaças e Vulnerabilidades

De modo geral, uma ameaça a uma rede constitui, também, uma ameaça à segurança da informação. Estas ameaças põem em causa as três propriedades fundamentais da segurança da informação, anteriormente analisadas: integridade, disponibilidade e confidencialidade [14].

A informação como um todo, está sujeita, desde sempre, a diversas ameaças primárias, como catástrofes naturais, falhas de energia, vandalismo, entre outras. Mais recentemente, com o advento da *Internet* nas organizações e na sociedade em geral, apareceram novas vulnerabilidades, para além das já existentes, entre elas formas de espionagem comercial, fraudes

e erros. Atualmente, existe também uma necessidade de proteção na rede contra os *hackers* e suas invasões, vírus e outras ameaças características deste meio.

Os sistemas ou entidades com maiores vulnerabilidades e, por isso mesmo, expostos a um maior risco são as redes de computadores, as bases de dados, os sistemas de energia, comunicação e informação.

De maneira a alcançar a maior segurança possível, numa aplicação para a *Internet* e na rede como um todo, são necessários cuidados e mecanismos de proteção em quatro elementos capitais: a estação de trabalho do cliente, que pode conter informação sensível e vulnerável, muitas vezes desprovida de proteção ao acesso; o meio de transporte, onde a utilização de algumas ferramentas de segurança pretendem assegurar a privacidade e integridade das informações trafegadas na *Internet*; o servidor, auxiliado por *firewalls* e um "servidor de controlo", consegue proteger e controlar o ponto único de troca de dados, controlando as comunicações e serviços autorizados; e, por último, a rede interna, que outrora era mais descartada e subvalorizada, por se partir do princípio de que as ameaças teriam origem externa à entidade em causa, o que, com a experiência que vai sendo adquirida na área, se percebe não ser verdade, pois muitos dos problemas advêm de vulnerabilidades e ações internas [13].

Relativamente ao tráfego em rede e, mais especificamente, ao modelo TCP/IP, as ameaças e vulnerabilidades são, também, diversas e com tendência para aumentar constantemente. Isto deve-se ao constante crescimento da *Internet* e de alguns aspetos de segurança inicialmente descartados, como a segurança dos protocolos, escalabilidade e espaço de endereçamento, tornando a *Internet* num alvo vulnerável a ciberataques.

"Existe uma variedade de vulnerabilidades na pilha protocolar TCP/IP, que derivam da forma como a pilha protocolar foi desenhada e projetada" [15]. Essas falhas de segurança advêm das tarefas e serviços que cada nível da pilha protocolar fornece. Os vários protocolos de encaminhamento, encontrados na camada de rede, apresentam algumas vulnerabilidades, que quando exploradas com sucesso, podem representar ameaças comprometedoras no funcionamento da rede, como a difusão de falsas rotas e estados de ligação incorretos. Na camada de transporte e como foi dito acima, encontram-se os protocolos TCP e UDP, que possuem igualmente algumas vulnerabilidades, uma vez que estes dois protocolos utilizam o endereçamento IP para reconhecer as várias máquinas da rede, sem formas de autenticação da origem dos pacotes, assim como da integridade e validade dos dados transferidos.

Quanto à generalidade dos protocolos da camada de aplicação do modelo TCP/IP, como DNS, FTP, SNMP e SMTP, as questões de segurança mais recorrentes são as seguintes: autenticação, *sniffinng*, *cache poisoning*, *stack overflow*, *spoofing* e negação de serviço (DoS) [15].

2.2.2. Segurança em Redes

Nas últimas décadas assistimos a uma evolução e expansão exponenciais da *Internet*, e, de facto, "vivemos numa sociedade cada vez mais dependente da Rede, tanto a nível pessoal como empresarial" [16]. As redes de computadores são, hoje em dia, fundamentais para as diversas atividades do quotidiano, sejam elas privadas, individuais ou organizacionais, refletindo-se numa necessidade da sociedade, em geral, dos computadores e redes. Desta forma, um intruso nestas redes é uma séria ameaça, podendo resultar em dispendiosas interrupções da rede e do trabalho. Este tipo de acesso indevido é alcançado pela exploração das vulnerabilidades, no *software* e *hardware*, ou pela tentativa de descoberta das credenciais de acesso dos utilizadores fidedignos. Aos intrusos que, através da modificação ou exploração de vulnerabilidades do software, conseguem acesso a um determinado sistema ou rede atribui-se a designação de *hackers*. Todas as particularidades e progresso da *Internet*, como o facto de ser aberta, distribuída e universal, são por um lado pontos positivos, mas por outro conduzem a um aumento constante das ameaças e riscos, tornando-a cada vez mais vulnerável [18].

Como tentativa de eliminar ou pelo menos combater as vulnerabilidades e questões acima mencionadas, a noção de segurança de redes ganha forma e robustez com a adição de medidas técnicas e melhoria da gestão da infraestrutura de rede. Este tipo de segurança baseia-se e abrange âmbitos como "(...) as ciências da computação, tecnologias de comunicação, redes de computadores, segurança da informação, teoria da informação" e privilegia princípios como a "(...) privacidade, integridade, confiabilidade, disponibilidade e não repudio de transferência de informação" [15].

Como mencionado até aqui, existem diversos problemas e ameaças que põem em risco a segurança nas redes e as propriedades fundamentais da informação. Segundo Rocha [15], essas vulnerabilidades englobam falhas no *design* e implementação dos protocolos utilizados nas redes, debilidades e escassez de processos na autenticação e nos mecanismos de controlo e encaminhamento de rede, bem como todas as fragilidades presentes nos inúmeros *softwares* utilizados, incluindo o malicioso. Há, por isso, motivos para a procura e melhora da segurança nas redes de computadores e da informação, sobretudo contra atividade não permitida e maldosa.

Este tipo de ameaça provém geralmente de duas origens distintas, de dentro da própria entidade a que pertence o sistema alvo, ou por pessoas externas ao mesmo.

Dentro das atividades não autorizadas ou maliciosas, pode-se salientar cinco padrões diferenciados: acesso a informação, alteração de informação, utilização abusiva de recursos, impedimento de prestação de serviço e vandalismo [15].

- O acesso a informação confidencial não autorizado, comporta todo o tipo de dados, guardada em sistemas informáticos ou que estejam em "trânsito na rede".
- A alteração de informação consiste em toda a atividade que modifique ou destrua informação, sem consentimento e de forma implícita ou explícita.
- Uma utilização abusiva de recursos de um dado sistema (memória, tempo de processamento e ocupação da rede), pode levar à escassez desses recursos para atividades benignas e fundamentais, conduzindo ao mau funcionamento ou, em último caso, indisponibilidade do serviço ou sistema informático.
- O impedimento de prestação de serviço acontece, como mencionado anteriormente, aquando de um caso extremo de utilização abusiva de recursos, com intenção de obstruir o acesso aos mesmos.
- Uma atividade de vandalismo constitui uma tentativa de apenas interferir no funcionamento correto dos sistemas computacionais, sem qualquer benefício próprio.

2.2.2.1. Deteção e Controlo de Acesso Indevido

Segundo Nascimento [18], um acesso não autorizado ou intrusão é "(...) qualquer ação ou conjunto delas, que tenham por intuito comprometer a integridade, a confidencialidade ou a disponibilidade de um recurso, sistema ou rede". Ou seja, todo o tipo de ações com potencial para pôr em risco as propriedades fundamentais da informação, conduzindo a alterações, permanentes ou não, da mesma, constitui uma forma de intrusão.

Para que uma intrusão seja efetuada com sucesso, tem que existir um conjunto de iniciativas por parte dos atacantes, na procura de falhas de segurança. Uma tentativa de intrusão inicia-se com a identificação, a nível externo, do tipo de sistema e da existência de sistemas de defesa, fazendo também uma análise aos padrões na troca de dados e aos protocolos utilizados. Posteriormente, é feito uma identificação das características internas, através de testes nos

servidores da rede, objetivando a descoberta de vulnerabilidades e informações sensíveis, como dados e credenciais de utilizadores, tipo de sistema operativo, hardware do sistema e serviços em execução. Reunidos os conhecimentos da rede e dos sistemas, o intruso pode facilmente optar pelo que lhe é mais conveniente atacar e aproveitar, se pretender, a intrusão, vulnerabilidades e dados recolhidos "(...) para transformar a máquina numa base de operações para outros ataques" [18].

Deste modo, a "deteção de intrusão é o processo de monitorização e análise de eventos, ocorridos em sistemas de computadores ou redes, com o propósito de identificar sinais de problemas de segurança", conforme Filho e Filho [19]. É, hoje em dia, fulcral o papel que a deteção de acessos indevidos desempenha nas redes de computadores e nos SI. Uma função que era assegurada por mecanismos primários, como práticas de autenticação e controlo de acesso, até aos antivírus e *firewalls*, mas que, com o ampliar contínuo de novas ameaças e vulnerabilidades, tende a ser insuficiente, passando os IDS a desempenhar um cargo importante no auxílio de deteção de intrusões e atividade maliciosa.

Os antivírus são programas informáticos que monitorizam, detetam e eliminam *malware*, de maneira a proteger os computadores.

As *firewalls* executam um "(...) conjunto de medidas e políticas de segurança e controlo de acesso, entre duas redes com níveis de confiança diferentes, podendo ser vistas como um separador ou um limitador entre duas redes" [15]. As mesmas controlam e impedem, de forma reativa, o acesso não autorizado à informação e recursos do sistema, através da análise e filtragem do tráfego da rede, mediante regras predefinidas.

Os sistemas IDS "(...) baseiam-se no princípio de que o comportamento malicioso e não autorizado de um sistema informático é claramente diferente do comportamento normal" [15]. Deste modo, os IDS procuram atividades, fora desse padrão estabelecido como regular, através da recolha e análise de diversas fontes de dados, reportando alertas de passíveis atividades maliciosas.

2.3. Sistemas de Deteção de Intrusão

A deteção de intrusões é um assunto que vem obtendo maior pertinência e investigação desde a década de oitenta, traduzindo-se no advento de várias estratégias e soluções para o tema em causa.

"Os sistemas de deteção de intrusão são, para as redes e computadores, como os sistemas de vigilância do mundo físico (...)" [19]. Um IDS procura detetar e alertar possíveis anormalidades no comportamento da rede ou dos servidores, passíveis de representarem um ataque ou intrusão com risco para o sistema. No universo dos IDS existem dois tipos de deteção e fonte de informação: o que deteta intrusões ao nível da máquina: *Host Intrusion Detection System* (HIDS), e o que deteta intrusões na rede: *Network-Based Intrusion Detection System* (NIDS) [18]. Relativamente aos métodos ou técnicas de deteção, estes sistemas podem ser: *Signature Based*, quando efetuam a análise do tráfego baseados em assinaturas de ataques conhecidos; ou Anomaly Based, quando o método procura desvios a um padrão de normalidade. Deve-se destacar, também, o tipo de análise dos IDS, podendo ser *Singular Based* ou *Collaborative Based*. A primeira opção representa uma abordagem e configuração mais minimalista e individual, pelo facto de a informação ser recolhida apenas por um sensor. Por outro lado, a segunda configuração recorre a múltiplos sensores, em zonas distintas do sistema, para recolha de dados. Este tipo de análise possibilita o combate a uma das maiores limitações e preocupações no campo da deteção de intrusões, que são os falsos alertas, uma vez que correlaciona os dados e eventos obtidos por cada sensor [15], [20].

Na Figura 4, pode-se observar tanto estes paradigmas, como outras características e categorias de classificação dos IDS, desenvolvendo-se algumas delas no decorrer do texto [1].

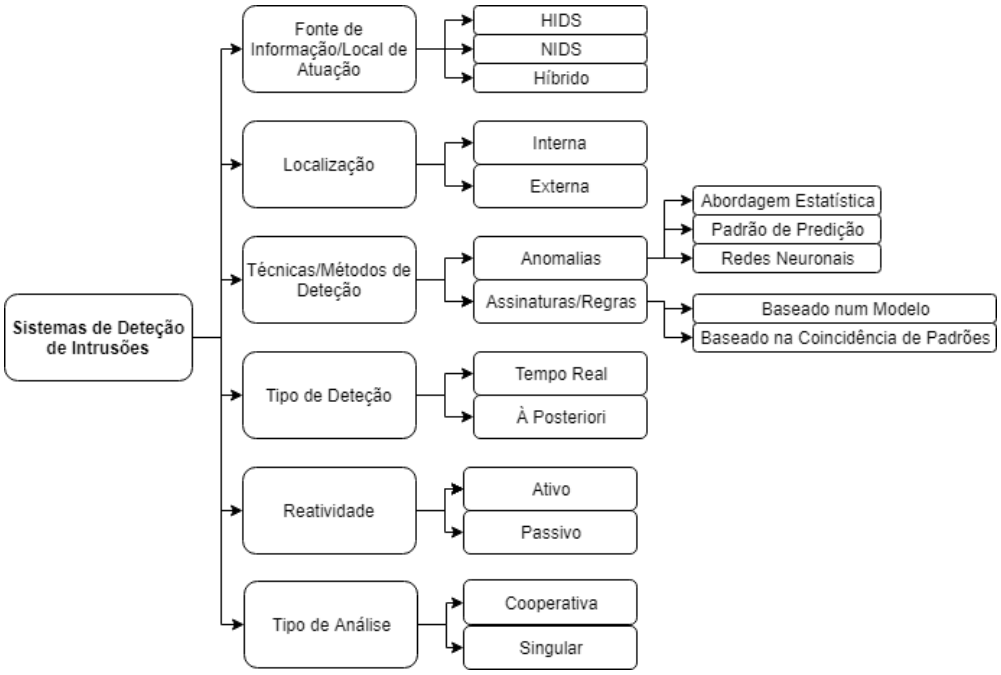


Figura 4 - Classificações de um IDS[4]

[4] Adaptado de Santos [1].

Estes sistemas possuem a capacidade de desempenhar várias funcionalidades, através da observação de *logs*[5] e arquivos, tais como [18]:

- "Monitorizar e analisar atividades de utilizador e sistema;
- Analisar a integridade de arquivos importantes do sistema;
- Analisar estatisticamente padrões de comportamento desconhecidos;
- Analisar comportamento baseado em padrões conhecidos (assinaturas);
- Analisar atividades anormais;
- Identificar origem e destino de ataques".

Relativamente à arquitetura funcional de um IDS, pode-se apresentar a mesma em quatro componentes distintas e complementares: "(...) *E boxes*, *A boxes*, *D boxes* e *C boxes*" [18].

A componente *E box* (*Event*) é a responsável pela recolha dos eventos gerados no sistema controlado, envia os elementos de baixo nível para a *A box* e investiga padrões, que possam revelar ataques ou intrusões. No elemento *A box* (*Analysis*), geram-se "(...) eventos de alto nível, indicadores de atividades de mais alto nível, estes elementos retroalimentam a *A box* e são enviados para as *D box* e *C box*". Na *D box* (*Storage*) são armazenados os dados gerados "(...) (sistema de ficheiros ou base de dados)". A *C box (Countermeasure*) trata os dados enviados pelas *A box* e *E box*, deteções e eventos gerados de baixo ou alto nível, responsabilizando-se pela reação adequada a cada evento, "(...) por exemplo enviando e-mail, relatórios ou alertas" [18].

De seguida (Figura 5), pode-se ver, de forma mais ilustrativa e sucinta, as várias interações entre componentes, mencionadas anteriormente:

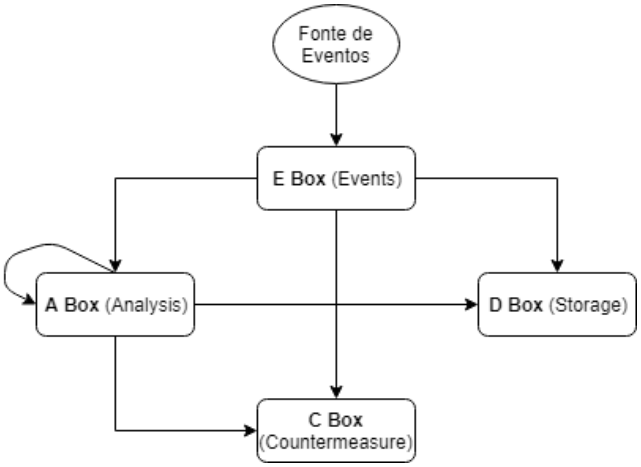


Figura 5 - Arquitetura funcional de um IDS[6]

[5] No universo informático, um arquivo de *log* é um ficheiro que regista eventos ocorridos num programa e/ou sistema operativo (disponível em: https://en.wikipedia.org/wiki/Log_file; consultado a 03/01/19).
[6] Adaptado de Nascimento [18].

2.3.1. HIDS (Host-Based Intrusion Detection System)

Os sistemas de deteção de intrusões, baseados no *host* (máquina), "são sensores locais, que detetam eventos de segurança e atividade maliciosa relativamente à máquina que se encontram a monitorizar". Têm como principal função analisar toda a informação relevante a que tem acesso, como "(...) *logs* do tráfego e da máquina onde está instalado, violações de integridade nos ficheiros do sistema, possíveis portas abertas indevidamente ou mais tipicamente tentativas falhadas de login, entre muitos outros sinais de anomalia".

Os sistemas de deteção de intrusões, baseados em máquina, implicam a instalação de um sistema por cada máquina da rede. Contudo, é possível aglomerar todos os eventos gerados e enviá-los para uma "máquina central", agilizando a supervisão dos eventos.

Estes sensores, embora localmente condicionados e isolados, "(...) fornecem informação sobre eventos, aos quais um NIDS não teria acesso, visto que a informação é restrita à máquina sobre a qual o HIDS se encontra instalado" [5], [10].

De entre as várias funções e métodos, que faz uso para a deteção de intrusões, destacam-se as seguintes:

- Verificação da integridade de ficheiros: Através de uma função de *hash*[7] criptográfica, é guardado o estado dos ficheiros do sistema, procedendo-se à comparação das *hashes* mais atuais com as mais antigas, averiguando se existem alterações e, em caso afirmativo, emitindo um alerta;
- Análise de logs: Os HIDS utilizam os *logs*, do sistema onde estão instalados, para análise e comparação de padrões, que indiquem possíveis desvios às ditas atividades normais, ou seja, suscetíveis de representarem atividade maliciosa. "Exemplo: Um utilizador autenticou-se como *root*[8] fora das horas de trabalho";
- Deteção de rootkits[9]: Busca por *rootkits* alojados no sistema em monitorização.

De seguida, são indicados alguns dos mais relevantes HIDS, em código aberto [5]:

---

[7] Tipo de dados informático (*string* de bits), gerado a partir de um conjunto de informações e de um algoritmo, que modifica a *string* resultante, conforme as informações recebidas (disponível em: https://en.wikipedia.org/wiki/Cryptographic_hash_function; consultado a 03/01/19).

[8] *Root, administrator, admin, supervisor* ou superutilizador é uma conta de utilizador que concede privilégios especiais de administração no sistema operativo (disponível em: https://pt.wikipedia.org/wiki/Super_usu%C3%A1rio; consultado a 03/01/19).

[9] *Rootkit* é um programa ou conjunto deles, normalmente maliciosos, projetado para conceder o acesso não autorizado a um computador ou parte dele, de forma impercetível (disponível em: https://en.wikipedia.org/wiki/Rootkit; consultado a 03/01/19).

- OSSEC [21];
- Samhain [22];
- Prelude-LML [23];
- Sagan [24];
- AIDE [25].

2.3.2. NIDS (Network-Based Intrusion Detection System)

Os IDS que recolhem e analisam o tráfego de rede são chamados de NIDS (sistemas de deteção de intrusões baseados em rede). Estes programas identificam eventos de segurança, através da captura e análise de todo o tráfego na rede, podendo gerar alertas em "tempo real" ou à "posteriori".

Um NIDS procede, essencialmente no seu modo de funcionamento, à análise de duas principais componentes: a verificação da pilha protocolar e a verificação da aplicação protocolar [5], [10]. A primeira "(...) rastreia pacotes que não se encontram de acordo com as regras dos protocolos de rede", enquanto que a segunda certifica a aplicação protocolar, "(...) rastreando violações de utilização de protocolos. Neste tipo de violação, a estrutura do protocolo está intacta, mas este é utilizado de forma errónea, sendo um exemplo muito comum o *syn overflow*".

Este tipo de sensores pode ser usado de forma singular, mas, em redes maiores e mais complexas, o seu uso como "sensores remotos" possibilitam uma melhor recolha e análise de todo o tráfego essencial da rede. Neste tipo de configuração, os alertas podem ser centralizados para uma única máquina ou servidor, onde é possível administrar toda a informação [5].

Os sistemas de deteção de intrusões na rede possuem algumas vantagens, nomeadamente na transparência de atuação, uma vez que a sua finalidade não exige, obrigatoriamente, uma instalação em cada máquina constituinte da rede, assim como na robustez a ataques, isto porque, face à sua configuração e funcionamento, consegue resistir e passar despercebida ao intruso. Por outro lado, existem também algumas desvantagens neste tipo de IDS, salientando-se a impossibilidade de analisar tráfego cifrado, assim como a necessidade de configurações especiais e auxiliares, na recolha de tráfego em algumas estruturas de redes mais complexas [10].

Quanto ao método de deteção nos NIDS, o mesmo pode-se basear em dois tipos [5]:

- **Assinaturas**: A deteção é efetuada com base na análise e comparação do tráfego capturado com "padrões de eventos", predefinidos em regras. Esta forma de

atuação tem o inconveniente de partir em desvantagem, aquando do aparecimento de um novo padrão de ataque, visto ser preciso primeiro detetar, criar e adicionar esse novo padrão à base de regras. Deste modo, "o método de deteção por assinaturas é altamente dependente das regras utilizadas", tanto em número como em fiabilidade das mesmas. Dentro dos conjuntos de regras para sistemas de deteção de intrusões em rede, pode-se salientar os seguintes, disponíveis e atualizados de forma gratuita pela "comunidade de segurança":

  - (VRT) SourceFire Vulnerability Research Team: Trata-se do conjunto de regras, oficialmente concebido, para um dos mais reconhecidos NIDS, o Snort.
  - Emerging Threats: Representa um conjunto adicional de regras, com o intuito de complementar as VRT *rules* e abranger um maior número de ataques. São independentes e compatíveis com qualquer sistema NIDS, em particular o Suricata.

- **Anomalias**: Este tipo de deteção baseia-se em características modelo das redes analisadas (protocolos usados, largura de banda consumida, máquinas presentes, etc.) e, com base na definição de um comportamento normal, assinala tudo o que foge a essa referência. Esta técnica apresenta, normalmente, uma alta percentagem de falsos positivos nos sistemas IDS.

Relativamente ao universo de sistemas NIDS, em código aberto, gratuitos e mais amplamente usados, salientam-se os seguintes:

- Snort [26];
- Suricata [27];
- Bro Network Security Monitor [28].

Entre estes, abordar-se-ão, de seguida e com maior ênfase, o Snort e Suricata, NIDS escolhidos para a análise comparativa desta investigação.

2.3.2.1. Snort

O Snort é um sistema de deteção e prevenção de intrusões de rede, que inspeciona os pacotes, usando a correspondência de padrões, ou seja, pelo método de assinaturas ou regras. Possui três configurações ou modos distintos: o modo *sniffer* ("farejador"), análogo a outras

ferramentas de captura de tráfego na rede, como o tcpdump; o modo *logger* (registo) de pacotes, que permite guardar os pacotes capturados no disco e a configuração principal, que será a utilizada no contexto deste trabalho, modo de deteção de intrusões (IDS) [5], [29].

Este NIDS foi concebido em 1998 por Martin Roesch, um dos fundadores da Sourcefire, empresa que pertence agora à Cisco, responsável pelo atual desenvolvimento do Snort, sob as licenças GNU General Public License (GPL) v.2 e Non-Commercial Use License for the Proprietary Snort Rules [26].

"De acordo com Ghafir [29], "a arquitetura do Snort permite a implementação dos chamados pré-processadores". Estes mecanismos de ajuda à deteção, analisam e acondicionam o pacote para o passo seguinte, a avaliação da regra, possibilitando a implementação adicional de regras-chave. Os pré-processadores, também, facilitam a implementação de funcionalidades mais complexas, para além da relação de padrões, "(...) como a interpretação de dados e a deteção de anomalias".

Este NIDS é uma aplicação *single-threaded*, ou seja, que só trabalha com instâncias únicas de processamento do tráfego. Existe a possibilidade de ter uma configuração similar ao funcionamento *multi-threaded*, utilizando para tal uma divisão em partes do fluxo de tráfego a monitorizar, atribuindo o processamento dessas frações a instâncias singulares do Snort.

O modo de funcionamento de deteção de intrusões e arquitetura do Snort, mencionados anteriormente, é representado abaixo (Figura 6), de forma esquemática.

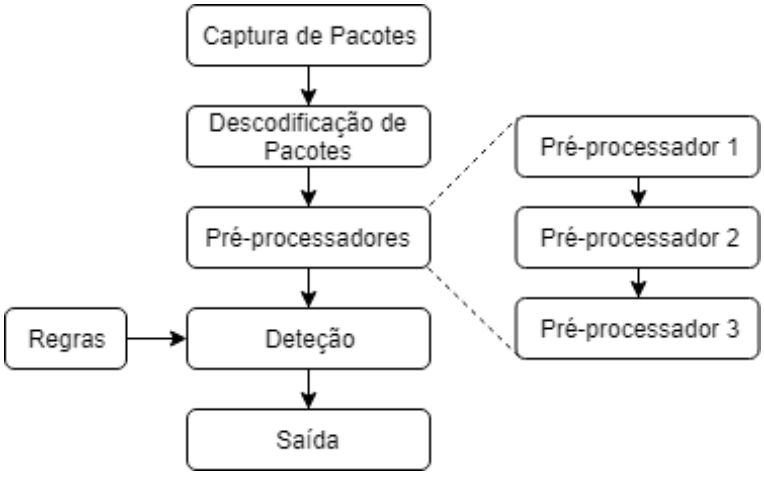


Figura 6 - Arquitetura single-threaded do NIDS Snort[10]

[10] Adaptado de Ghafir [29]

2.3.2.2. Suricata

"O Suricata é um mecanismo de deteção de ameaças de rede, gratuito e de código aberto, maduro, rápido e robusto" [27]. Este sistema possui deteção de intrusões em tempo real (IDS), prevenção de intrusão em linha (IPS), monitorização de segurança de rede (NSM) e processamento *offline* de ficheiros pcap[11]. Tal como o Snort, o método de deteção usado é baseado em assinaturas ou regras, podendo usufruir de conjuntos mais alargados e abrangentes das mesmas, desde as VRT, oficialmente criadas para o concorrente NIDS Snort, às Emerging Threats e ainda suporte a *scripts*[12] Lua[13], como auxílio na deteção de ameaças mais complexas.

Foi desenvolvido e lançado entre 2009-2010, pela Open Information Security Foundation (OISF), integrante do Navy's Space and NavalWarfare Systems Command (SPAWAR) e do Departamento de Segurança Interna dos EUA. Utiliza, igualmente, a licença GPL v.2.

Este NIDS foi produzido com o propósito de trazer novos conceitos e mecanismos ao contexto da deteção de intrusões, estabelecendo como principal diferenciação e vantagem, em relação à concorrência, o emprego de um mecanismo de deteção *multi-threaded*. Este motor de análise de tráfego, *multi-threaded,* possibilita a utilização mais eficiente dos processadores modernos, com múltiplos núcleos, o que ajuda na tentativa de desempenhar as funções de forma mais célere e eficiente. Futuramente, este mecanismo pode vir a ser auxiliado pelas unidades de processamento gráfico [5], [27].

Em termos de arquitetura, tem blocos de processamento similares ao Snort, com a diferença de não ter pré-processadores, mas, em vez destes, duas partes auxiliares: os módulos de descodificação e de deteção *multi-threaded*. Os primeiros adicionam informação adicional à representação interna dos pacotes, que por sua vez ajuda os módulos de deteção a fornecerem palavras-chave para uso nas regras [29].

Na Figura 7, pode-se observar a arquitetura do NIDS Suricata.

---

[11] Ficheiro que contém tráfego de rede, capturado e guardado por uma ferramenta, como o tcpdump (disponível em: https://wiki.wireshark.org/Development/LibpcapFileFormat; consultado a 08/01/19).

[12] Programas escritos em linguagem de *script*, que automatizam a execução de múltiplas tarefas ou comandos, sem a necessidade do utilizador as executar manual e individualmente (disponível em: https://pt.wikipedia.org/wiki/Linguagem_de_script; consultado a 08/01/19).

[13] Lua é uma linguagem de programação poderosa, eficiente e leve, projetada para estender aplicações (disponível em: http://www.lua.org/portugues.html; consultado a 09/01/19).

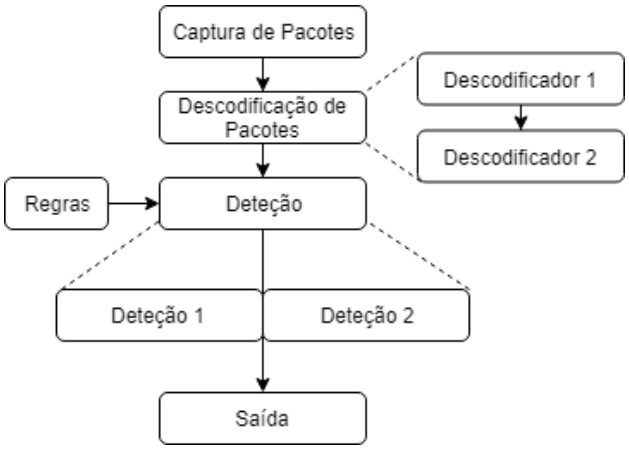


Figura 7 - Arquitetura multi-threaded do NIDS Suricata[14]

2.3.2.3. Snort vs Suricata

Na tabela[15] que se segue (Tabela 1), apresenta-se um breve resumo dos dois NIDS abordados, com alguns detalhes e funcionalidades relevantes.

Tabela 1 - Snort vs Suricata

Parâmetro	**Snort**	**Suricata**
Desenvolvedor	Sourcefire (Atualmente Cisco)	OISF
Lançamento	1998	2009 (Beta); 2010 (Estável)
Licença	GPL v.2	
Versão atual	2.9.12	4.1.2
Sistema operativo	Multiplataforma	
Mecanismo de deteção	*Single-threaded*	*Multi-threaded*
Método de deteção	Assinaturas	
Regras utilizadas	VRT (*Free*); Emerging Threats (*Free*); SO *rules* (*Shared Object*)	VRT; Emerging Threats; *Scripts* Lua
Modo IPS	Sim	
Suporte para tráfego IPv6	Sim	
Log de eventos	Arquivo simples; Base de dados; Unified2 *(Logs for* barnyard)	

[14] Adaptado de Ghafir [29].
[15] De acordo com [4], [5], [26], [27], [66] (consultado a 09/01/19).

Aceleradores de captura	Não (Uso da libpcap)	PF_RING; AF_PACKET
Análise offline (pcap)	Sim	
Ferramentas de terceiros	Sguil; SQueRT; Snorby; Aanval; BASE; FPCGUI; Snortsnarf; Telesoft CERNE Open IDS Platform	

2.3.3. Vantagens e Desvantagens dos IDS

Assim como a generalidade do software e ferramentas de segurança, os sistemas de deteção de intrusões também possuem vantagens ou pontos fortes e desvantagens ou limitações. Entre as várias características que se enquadram nesses pontos, destacam-se as seguintes [19], [30]:

- Vantagens/pontos fortes:
  - Monitorização, análise e classificação de eventos e da utilização;
  - Identificação de anomalias nas atividades dos utilizadores;
  - Definição de limites na segurança, assim como desvios aos mesmos;
  - Políticas predefinidas de segurança da informação;
  - Identificação de padrões de ataques conhecidos;
  - Registo de eventos;
  - Utilização básica relativamente facilitada, mesmo para profissionais pouco experientes.
- Desvantagens/limitações:
  - Escalabilidade difícil e complexa;
  - Limitações ao nível dos recursos solicitados, na análise de grande volume de dados;
  - Taxas significativamente altas de falsos positivos, principalmente em IDS baseado em assinaturas;
  - Perda de efetividade contra técnicas de evasão ou ataques sofisticados;
  - Dificuldade de análise em redes comutadas e dados encriptados;
  - Não compensam mecanismos ou pontos da rede mal configurados ou vulneráveis;
  - Não averiguam um ataque de forma autónoma.

2.3.4. Alertas Gerados e Desafios na Eficiência da Deteção

A principal função de um sistema IDS é a monitorização do tráfego ou eventos, em rede ou na máquina, aferindo, através dos métodos de deteção e decisão, se os dados que analisou são ou não indicadores de uma atividade maliciosa. Este tipo de decisão tem duas principais classificações: positivo, se o IDS emite um aviso, ou negativo, se por outro lado não for gerado qualquer alerta. Para além destas duas possibilidades de decisão, a mesma pode e deve ser classificada quanto à exatidão, ou seja, se um determinado alerta é verdadeiro ou falso.

Deste modo, a classificação dos alertas gerados, por um sistema de deteção de intrusões, pode ter as seguintes combinações: verdadeiro positivo, falso positivo, verdadeiro negativo e falso negativo [18].

- Verdadeiro positivo: Ocorre quando é gerado um alerta verídico, ou seja, quando o IDS deteta com exatidão um ataque ao sistema;
- Falso positivo: Ao contrário do caso anterior, trata-se de um alerta que não deveria ser emitido, ou seja, foi incorretamente identificada uma possível intrusão;
- Verdadeiro negativo: Representa o cenário ideal, quanto à inexistência de atividade maliciosa no sistema e, ao mesmo tempo, à correta ausência de qualquer alerta do IDS;
- Falso negativo: Nesta situação, por outro lado, ocorre uma intrusão, que não é detetada pelo sistema de deteção de intrusões.

Como vimos até aqui, os sistemas de deteção de intrusões desempenham um papel fundamental na segurança da informação e, mais concretamente, na deteção de tráfego e atividade maliciosa nos sistemas em rede. Contudo, e como também podemos perceber anteriormente, existem ainda algumas lacunas e limitações, entre as quais se salientam como mais pertinentes, "(...) o elevado número de falsos positivos, a falta de ferramentas e métricas disponíveis para avaliação e complexa comparação entre diferentes técnicas utilizadas em sistemas de deteção de intrusões" [10].

Por estas razões, o atual desafio, neste contexto de segurança, assenta sobretudo na procura do aumento do reconhecimento de novos ataques e na redução da taxa de falsos positivos, que, independentemente do tipo de IDS e configuração usada, continuam a representar uma larga percentagem dos alertas produzidos, traduzindo-se numa baixa eficiência de deteção e na dificuldade de gestão dos mesmos [15].

Desta forma, a avaliação e comparação de sistemas de deteção de intrusões representam uma importante investigação, nos desafios contemporâneos que estes sistemas enfrentam. No entanto, esse tipo de trabalho requer períodos de realização de testes significativamente prolongados, "(...) quer pela falta de tráfego malicioso, quer pela necessidade de avaliar a sua deteção e o número de eventos gerados pelos sistemas de deteção de intrusões" [10]. Nesse quesito, pode-se salientar as experiências que utilizam *datasets* com tráfego selecionado e armazenado, disponibilizado por diversas entidades e comunidade de segurança, e/ou que são realizadas num ambiente controlado, fazendo parte destas formas de abordagem o trabalho desenvolvido para esta dissertação.

2.4. Avaliação de Desempenho em IDS

Nesta secção do documento, abordar-se-ão alguns conceitos para uma comparação e avaliação do desempenho de IDS, os instrumentos imprescindíveis ao desenvolvimento da experiência proposta, como *frameworks* existentes e *datasets* de tráfego de rede, assim como alguns trabalhos desenvolvidos no contexto da avaliação de desempenho em sistemas de deteção de intrusões.

A avaliação de sistemas IDS é um assunto pertinente para a segurança informática. Estes sistemas possuem ainda lacunas e limitações, muita das vezes desconhecidas, que precisam ser estudadas, analisadas e comunicadas aos desenvolvedores, de forma a entregar um produto ao consumidor o mais fiável possível. Este tipo de estudo permite conhecer melhor os mecanismos de deteção dos diferentes IDS, tanto ao nível da monitorização como da deteção e possível prevenção. Ao mesmo tempo recolhem-se dados e padrões úteis dos diferentes atacantes e *malwares*, contribuindo para a robustez das técnicas de deteção [31].

2.4.1. Critérios de Avaliação

Para este tipo de estudo deve-se construir uma plataforma de trabalho que cumpra quatro requisitos fundamentais: viabilidade da mesma, orçamento o mais baixo possível, imparcial e precisa.

Uma avaliação geral de IDS deve seguir e mensurar alguns critérios, podendo os mesmos ser ajustados ou selecionados, de acordo com o foco e cenário de teste pretendido. Segundo Wei [31], esses pontos fundamentais de análise são:

- Eficácia: A capacidade que o sistema de deteção de intrusões tem para detetar o maior número de ataques possível, evitando ao mesmo tempo os falsos positivos.
- Eficiência: Deve ser capaz de efetuar o seu trabalho funcional no menor tempo possível, sem que para isso exija demasiados recursos de *hardware* e ponha em causa o restante funcionamento do sistema em que está inserido.
- Facilidade de utilização: Quanto mais fácil e intuitiva for a utilização de um IDS melhor, de forma a facilitar o seu uso para além dos especialistas em segurança.
- Segurança: Pretende-se um IDS o mais robusto e seguro possível, que proteja o sistema onde está instalado, mas também a si próprio, protegendo-se de técnicas de evasão, quebra e sobrecarga.
- Cooperação entre IDS: Possibilidade de integrar um IDS com outros, promovendo a capacidade de resposta e cobertura aos possíveis ataques. Pode, no entanto, requer mais recursos da máquina de testes.
- Colaboração: Aptidão do IDS em se integrar com outras ferramentas de segurança, como a Firewall, tendo em vista a melhoria da segurança geral do sistema.

Para além dos requisitos básicos, na construção da plataforma de trabalho, que se tentaram cumprir, destacam-se também a eficácia e eficiência, entre os principais critérios de apreciação para o foco de estudo desta dissertação.

2.4.2. Frameworks

Uma *framework* é uma estrutura ou sistema mais generalista, material ou conceptual, designada a servir de suporte ou guia para o desenvolvimento de algo mais específico.

Na área de sistemas de computadores, uma estrutura é geralmente constituída por camadas, que indicam o tipo de *softwares* que esta deve acolher e como se devem relacionar. Para além dessas características, as estruturas informáticas oferecem, geralmente, um conjunto de programas, interfaces gráficas ou de programação e ferramentas predefinidas, para o propósito pretendido [32], [33].

Abaixo, apresenta-se algumas das *frameworks* mais relevantes, de acesso gratuito e código aberto, que podem servir de base de trabalho no campo da deteção de intrusões.

2.4.2.1. Security Onion

Segundo o sítio oficial deste projeto [34], "o Security Onion é uma distribuição Linux, gratuita e de código aberto, para deteção de intrusão, monitorização de segurança e gestão de *logs*. Este inclui o Elasticsearch, o Logstash, o Kibana, o Snort, o Suricata, o Bro, o Wazuh (OSSEC), o Sguil, o Squert, o CyberChef, o NetworkMiner e muitas outras ferramentas de segurança. O assistente de configuração, fácil de usar, permite que você crie um exército de sensores distribuídos em minutos".

A *framework* Security Onion engloba e oferece três principais funcionalidades:

- Captura completa de pacotes;
- Deteção de intrusões em rede (NIDS) e em máquina (HIDS);
- Poderosas ferramentas de análise.

Esta solução assume-se, desta forma, como uma poderosa e competente estrutura e suite de *softwares*, pronta a servir de alicerce para monitorização de segurança de rede, agilizando e garantindo, em grande parte, as várias dependências, complexidades e compatibilidades das várias ferramentas integradas por predefinição.

2.4.2.2. OSSIM

O OSSIM (Open Source Security Information Management) [35] é um sistema, em código aberto, de gestão de eventos e informações de segurança (SIEM).

Integra um conjunto de ferramentas projetadas para o auxílio da administração da rede, deteção e prevenção de intrusões em computadores, tais como: PRADS, OpenVAS, Snort, Suricata, OSSEC, Nagios e Munin.

Este projeto apareceu em 2003, através de três colaboradores, tornando-se num *software* base para a empresa AlienVault, em 2008, que viria a comercializar um derivado do OSSIM, o AlienVault Unified Security Management.

Esta estrutura "(...) tem como objetivo fornecer aos gestores de segurança uma visão de todos os aspetos relacionados à segurança de seus sistemas, combinando e correlacionando administração de *log* e gestão e deteção de ativos, com dados dos controlos de segurança da informação e sistemas de deteção dedicados" [36].

2.4.2.3. Network Security Toolkit

O Network Security Toolkit [37] é um sistema "(...) baseado no Fedora 28, que fornece um fácil acesso às melhores aplicações de segurança de rede, em código aberto, podendo ser executado na maioria dos sistemas x86_64".

Grande parte dos programas e instrumentos de segurança de rede, mais amplamente cotados e utilizados, estão presentes nesta *framework*. Como complemento e forma de agilizar certas tarefas, como "(...) administração de sistema/rede, navegação, automação, monitorização de rede, localização de *hosts*, análise de rede e configuração de diversas aplicações de rede e segurança (...)", foi adicionada uma avançada interface *web* de utilizador.

Esta solução de *softwares* pode ser, ainda, "(...) empregue em servidores virtuais de empresas, que alojam máquinas virtuais, como uma ferramenta de análise, validação e monitorização de segurança de rede".

Como principais características e funcionalidades, destacam-se as seguintes:

- Captura de pacotes de rede *multi-tap;*
- Gestão, *web-based*, de ferramentas de segurança da rede;
- Monitorização de rede/sistema;
- Deteção de intrusões na rede;
- Integração, *web-based*, do NIDS Snort;
- *Scanner* de segmento de rede ARP.

2.4.3. Datasets

Os *datasets* são conjuntos de itens com dados relacionados, organizado num determinado tipo de estrutura de dados, onde se podem encontrar caraterísticas de cada elemento presente nos mesmos [38].

No contexto desta dissertação, os *datasets* que foram investigados e empregues, para a realização da experiência de avaliação e comparação de IDS, tratam-se de *pcap datasets*, ou seja, repositórios públicos com captura de pacotes de tráfego da rede. Este tipo de dados, para além de fornecer tráfego de rede previamente recolhido e armazenado, substrato essencial à execução da investigação pretendida num ambiente controlado, também, normalmente, o seleciona e categoriza devidamente, com informações imprescindíveis para uma boa avaliação de sistemas IDS.

Entre as várias fontes deste tipo de *dataset*, que se podem encontrar *online*, referenciam-se as seguintes:

- DARPA – 1998/99 DARPA Intrusion Detection Evaluation Dataset: Um dos *datasets* mais antigo e conhecido, fornecido pelo Defense Advanced Research Projects Agency. Contém "amostras do tráfego de rede e *logs* de auditoria que foram usados para avaliar sistemas. Estes dados foram primeiramente disponibilizados em fevereiro de 1998" [39]. Apesar de ser dos conjuntos de tráfego de rede mais amplamente utilizados ao longo dos anos, na área da deteção de intrusões, neste momento, encontra-se desatualizado e é pouco útil e credível para a avaliação de IDS.
- KDD – KDD Cup 1999: "Este é o conjunto de dados usado para a Terceira Competição Internacional de Ferramentas de Descoberta de Conhecimento e de Mineração de Dados, realizada em conjunto com a KDD-99 A Quinta Conferência Internacional sobre Descoberta de Conhecimento e Mineração de Dados" [40]. Este *dataset* congrega várias intrusões, simuladas num ambiente de rede militar. Como no exemplo anterior, trata-se de uma fonte de dados já ultrapassada.
- MAWI Working Group Traffic Archive – WIDE: "É um repositório de dados de tráfego, mantido pelo MAWI Working Group do projeto WIDE" [41].
- CAIDA: Trata-se de um centro de análise de dados da internet, que "coleta vários tipos diferentes de dados em locais geograficamente e topologicamente diversos, disponibilizando esses dados à comunidade de pesquisa na medida do possível, preservando a privacidade de indivíduos e organizações que doam dados ou acesso à rede" [42].
- SecRepo – Samples of Security Related Data: É um site que tenta manter uma lista credível de dados, relacionados à segurança da rede, sistema entre outros, de fontes próprias e de terceiros [43].
- Stratosphere IPS – Datasets: O projeto Stratosphere IPS visa a "(...) construção de um software gratuito com aprendizagem de máquina, baseado no sistema de prevenção de intrusões (IPS), que pode ser usado para detetar e prevenir ataques cibernéticos específicos" [44]. Este projeto cria modelos precisos para avaliações de desempenho, a partir de capturas reais de tráfego malicioso, ao usar e estudar como o *malware* se comporta na realidade. Monitorizam continuamente o

panorama das ameaças emergentes, recuperando, executando e armazenando amostras maliciosas, sendo publicados *datasets* dessas experiências, providos de informação útil e variada acerca das propriedades do tráfego incluso.

- NETRESEC – Publicly available PCAP files: A NETRESEC, fornecedor de *software* independente com foco no campo da segurança de rede, disponibiliza, no seu site, uma lista com vários repositórios públicos de captura de pacotes, entre eles, os que já aqui mencionamos [45].

2.4.4. Trabalhos Desenvolvidos

O campo da deteção de intrusões e, consequentemente, a avaliação e comparação dos sistemas desenvolvidos para este paradigma é, há vários anos, um tema bastante investigado. Entre os sistemas IDS, destacam-se os NIDS Snort e Suricata, que representam duas das mais robustas e competentes soluções, em código aberto, para o combate de intrusões. Contudo, e apesar do constante desenvolvimento dos mesmos, estes sistemas não são perfeitos e apresentam vulnerabilidades, limitações e erros na deteção de intrusões, como falsos positivos ou negativos. Apresentam arquiteturas, funcionalidades e métodos de deteção muito similares, mas também têm pequenos pontos e características que os diferenciam, como os mecanismos de deteção *single-threaded* (Snort) e *multi-threaded* (Suricata).

Deste modo, torna-se indispensável o estudo, análise e avaliação do desempenho destes sistemas, partindo, por exemplo, da comparação de resultados entre soluções idênticas e com objetivos comuns.

Neste âmbito de investigação, a ALDEID [4] realizou uma experiência de comparação entre Snort e Suricata, ainda numa fase bastante preliminar do desenvolvimento deste último, garantindo condições estruturais e de execução o mais idênticas possível. Foram realizados mais de 300 testes com estes dois NIDS, visando sobretudo as capacidades de deteção dos mesmos. Foi demonstrado, que as regras VRT e Emerging Threats são complementares e aumentam a cobertura de possíveis ataques detetados, quando combinadas, tendo o Suricata revelado incompatibilidades com algumas destas assinaturas. Em termos de documentação e maturidade, o Snort levava vantagem sobre o Suricata, embora este já apresentasse ideias e recursos promissores. Verificou-se que o Suricata apresentava maior eficiência na deteção de *malwares*, vírus e *shellcodes*. A investigação termina com a conclusão de que o Snort era, àquela altura, uma

solução mais robusta e madura, enquanto que o Suricata demonstrava ser uma alternativa emergente, com bastante potencial e capaz de revolucionar certas técnicas de deteção.

Mais recentemente, Shah e Issac [46] efetuaram um estudo com comparações do desempenho, quantitativas, rigorosas e repetitivas, dos IDS Snort e Suricata, medindo a percentagem de utilização do CPU, memória e a taxa de pacotes perdidos. "Os experimentos consistiram num banco de testes que comparou a precisão de deteção do Snort e Suricata, numa velocidade de rede de 10 Gbps e com sete tipos diferentes de tráfego malicioso". O estudo comparativo destes dois NIDS concluiu que ambos demonstraram bons índices de desempenho e eficiência, contando cada um com seus "pontos fortes e fracos". Os resultados obtidos revelaram um Snort mais moderado na solicitação de recursos físicos, para processar o tráfego selecionado, ao contrário do Suricata, que por sua vez conseguiu processar um maior número de pacotes por segundo, devido ao seu mecanismo de deteção *multi-threaded*. Em termos de alarmes classificados como falsos positivos, ambos os IDS obtiveram uma percentagem alta.

Albin e Rowe [47], executaram um teste de controlo e comparação completo ao Suricata, tendo como referência o Snort. Para tal, foram feitos testes de comparação de velocidade num ambiente similar àquele em que é geralmente instalado, testes à velocidade de deteção e possíveis melhoramentos, justificados pelo emprego do *multithreading*, e testes à precisão de deteção de *exploits* conhecidos, que confirmassem a sua cobertura. No geral, o IDS Suricata obteve um desempenho pelo menos tão bom quanto o seu concorrente Snort. Quanto aos falsos positivos e negativos gerados, o teste foi inconclusivo, devido à probabilidade de serem influenciados por falhas no conjunto de regras utilizado. Foi recomendado uma máquina de 64 bits para um melhor carregamento completo de conjuntos de regras, assim como para o pedido mais elevado de recursos durante a deteção, sobretudo pelo Suricata. A experiência mostrou que o uso agregado de CPU do Suricata era quase o dobro do uso do Snort, no caso da utilização de RAM, o Suricata usou mais do dobro da quantidade de RAM usada pelo Snort, aparentemente para lidar com o motor de deteção *multi-threading*, que pôde ajudar na redução da taxa de pacotes perdidos.

Camelo et al. [48] apresentam uma comparação dos IDS/IPS Snort e Suricata, numa topologia onde foi testado a monitorização dos pacotes que transitam entre duas redes distintas, com o sistema Kali Linux a servir de atacante, com alguns ataques previamente selecionados, e o metasploitable2 como plataforma intencionalmente vulnerável aos ataques. Após os testes realizados, com alguns ataques predefinidos, concluíram que ambos os NIDS obtiveram resultados similares. No entanto, e como critério de desempate, salientaram a falha na deteção de um ataque

de negação de serviço e de outro de ganho de acesso ao sistema, pelo Snort e Suricata respetivamente, considerando a falha na deteção do ataque de ganho de acesso ao sistema mais relevante, o que deu ligeira vantagem ao Snort neste caso.

Como vimos, a comparação e avaliação do desempenho de IDS, em particular entre os dois NIDS: Snort e Suricata, tem vindo a ser investigada com alguma frequência. Contudo, não existe um consenso sobre qual destes é melhor, mas sim um conjunto de fatores e caraterísticas em que, normalmente, um sobressai mais que o outro e vice-versa. Alguns dos testes e trabalhos sobre estes dois sistemas foram conduzidos em supercomputadores, o que não permite aferir, com certeza, acerca do verdadeiro desempenho destes NIDS numa máquina comum. Por outro lado, algumas comparações entre Snort e Suricata têm já alguns anos, fase inicial do desenvolvimento do Suricata, que em comparação ao já consolidado Snort representa, à partida, uma desigualdade.

Deste modo, atendendo aos aspetos mencionados acima entre outros, existe ainda espaço para alterações e consolidações aos paradigmas e resultados de investigação neste âmbito. Esta dissertação simulou um ambiente de testes num computador com recursos dentro do padrão de um utilizador comum. Utilizou-se as versões estáveis, mais recentes, de ambos os NIDS, assim como um conjunto de regras atualizado e compatível para ambos os sistemas de deteção de intrusões, para além do emprego de *datasets* de tráfego o mais atuais possível, no conjunto de experiências realizadas. Estas características visam uma melhor análise e aferição do desempenho geral dos NIDS, assim como das assinaturas detetadas, com maior ênfase para os falsos positivos. Procurou-se, desta forma, garantir um ambiente de execução e teste comum e compatível com o de um utilizador padrão contemporâneo, assim como o mais atual possível, tendo em vista a maior fiabilidade e atualização dos resultados no contexto pretendido, face a estudos mais antigos.

3. Testes a Realizar

Como referido no capítulo introdutório deste documento, o objetivo desta dissertação consiste na comparação e avaliação de dois sistemas NIDS semelhantes: Snort e Suricata. Para tal, realizaram-se testes aos seus desempenhos, ao nível dos recursos de *hardware* utilizados, da capacidade de processamento na análise do tráfego e dos alertas gerados, garantindo as condições e características do ambiente de trabalho o mais similares possível, para cada IDS e experiência.

Esta investigação e os testes desenvolvidos tentam dar resposta a uma questão central e alguns outros aspetos relacionados:

- Existem diferenças significativas no desempenho geral do Suricata, face ao concorrente Snort, que evidenciem as potencialidades de algumas características diferenciadoras que possui?
    - Mecanismo de deteção *multi-threaded vs single-threaded*;
    - Recursos utilizados (CPU, RAM) *vs* desempenho de deteção (processamento do tráfego, perda de pacotes e alertas gerados).

Neste capítulo, serão descritos os processos e instrumentos relevantes na preparação da plataforma de trabalho, assim como os ensaios efetuados, de acordo com as metas e questões mencionadas.

3.1. Preparação da Plataforma

Todo o trabalho desenvolvido foi realizado num ambiente de virtualização, configurado através da tecnologia QEMU-KVM[16], que permitiu a preparação de uma *framework* adequada ao âmbito de estudo. O computador utilizado para alojamento das máquinas virtuais é um *Desktop* com processador Intel Core i5-6400 e 15.6 GB de memória RAM, com o sistema operativo Linux Mint 18.3 Cinnamon 64 bit. A máquina virtual configurada tem 4 processadores, aproximadamente 14 GB de RAM, 40 GB de disco e uma interface de rede "*default*" NAT[17]. O sistema operativo instalado

[16] (KVM - Kernel-based Virtual Machine) O QEMU é um emulador/"virtualizador" de sistemas operativos, genérico e livre. Quando usado com o módulo do kernel linux KVM, o QEMU atingi desempenho similar ao nativo, executando a máquina virtual diretamente no CPU do host (disponível em: https://wiki.qemu.org/Main_Page; consultado a 09/12/18).

[17] *Network Address Translation* (NAT) é uma configuração de interface de rede, útil na conecção de máquinas virtuais em rede num ambiente de trabalho. Uma rede NAT permite que o sistema virtual tenha acesso total à rede, a comunicação entre *host* e *guest*, mas impedindo o último de ser diretamente visível na rede física (disponível em: https://wiki.qemu.org/Documentation/Networking/NAT; consultado a 23/01/19).

na máquina virtual foi o Linux Security Onion (Ubuntu 14.04.1) [34], que, como mencionado anteriormente em 2.4.2.1, apresenta uma *framework* para monitorização de segurança, mais concretamente a deteção de intrusões. A escolha desta solução como base para o trabalho a desenvolver, teve que ver com todas as potencialidades que a mesma oferece. Reúne várias ferramentas e *softwares* de segurança instalados de raiz, salientando-se os dois NIDS que se pretende estudar, Snort e Suricata, assim como interfaces de monitorização dos eventos produzidos por estes sensores, como Sguil e Squert. Possui uma interface limpa e utilização geral relativamente amigável, assim como uma atualizada e ampla documentação *online*.

3.1.1. Sistema

Após a fase de instalação da máquina virtual no gestor de máquinas QEMU-KVM, foram configuradas as melhores condições do sistema para o contexto em causa. Procedeu-se, para tal, a algumas verificações e ajustes do ambiente Security Onion, salientando-se a execução do "*setup wizard*"[18], que entre outros aspetos permitiu a configuração das interfaces de rede, das instâncias (PF_RING[19]) ativas dos mecanismos de deteção, uma por IDS no caso, e do modo de uso pretendido. Neste último caso, foi escolhido o modo de avaliação (*Evaluation Mode*)[20], ideal para o contexto do trabalho.

Para além da configuração inicial da *framework* Security Onion, foi necessário proceder a mais alguns acertos e à instalação de algumas ferramentas extra, assim como algumas dependências das mesmas, necessárias às experiências pretendidas. Neste quesito, salientam-se alguns utensílios imprescindíveis ao registo das métricas (mais à frente explicitadas) de desempenho dos IDS, como o psrecord e pidstat.

Relativamente ao tráfego de rede utilizado na comparação dos NIDS Snort e Suricata, foram escolhidos pcaps do repositório do projeto Stratosphere IPS [44], [49], apresentado previamente. Efetuou-se esta opção sobretudo pela vasta oferta de tráfego selecionado e cuidado, pela equipa

[18] Algumas dicas e ajustes de instalação e pós-instalação, presentes na documentação do Security Onion (disponível em: https://github.com/Security-Onion-Solutions/security-onion/wiki/QuickISOImage e https://github.com/Security-Onion-Solutions/security-onion/wiki/PostInstallation; consultado a 24/01/19).

[19] Numa máquina com vários núcleos CPU, o Security Onion permite a execução de instâncias paralelas dos mecanismos de deteção. (disponível em:https://github.com/Security-Onion-Solutions/security-onion/wiki/PF_RING; consultado a 27/01/19.

[20] "O modo de avaliação é ideal para ambientes de sala de aula ou de laboratório" (disponível em: https://github.com/Security-Onion-Solutions/security-onion/wiki/Use-Cases; consultado a 24/01/19).

de investigação do projeto, assim como pela garantia de *datasets* mais atuais, relativamente a outros conhecidos, tais como: DARPA e KDD.

3.1.2. Utilitários

De seguida, apresenta-se uma enumeração geral e breve explicação dos utilitários mais relevantes, aplicados no teste e recolha dos resultados de desempenho dos sistemas de deteção de intrusões.

- **Sguil:** Sguil [50] é uma interface gráfica intuitiva de análise de segurança de rede, que fornece acesso a eventos em tempo real, dados de sessão e capturas brutas de pacotes, facilitando a monitorização e análise orientada a eventos.

  Esta ferramenta foi importante na recolha, organização e análise dos alertas produzidos pelos NIDS.
- **Squert:** O projeto Squert [51] é uma aplicação Web, complementar ao Sguil, para consultar e visualizar dados de eventos armazenados na base de dados do Sguil. (normalmente, dados de alerta do IDS).

  Foi utilizada durante os testes, de forma complementar ao Sguil, por ser uma mais valia na organização e visualização gráfica dos alertas.
- **Psrecord:** O psrecord [52] é um pacote Python[21] para registo da atividade de processos, usa a biblioteca psutil[22] e matplotlib[23], para registar a atividade do processador e da memória de um processo.

  Ferramenta empregue no acompanhamento dos recursos (CPU e RAM) utilizados pelos IDS, assim como no registo dos mesmos em ficheiro de texto e graficamente.
- **Pidstat:** O pidstat [53] é um comando que faz parte do conjunto de utensílios do pacote sysstat, focado na monitorização do desempenho e atividade de um sistema. Este comando é aplicado na monitorização de tarefas individuais, geridas pelo kernel Linux.

[21] Python é uma linguagem de programação de alto nível, interpretada e com propósito geral (disponível em: https://en.wikipedia.org/wiki/Python_(programming_language); consultado a 24/01/19).
[22] O psutil é uma biblioteca em Python multiplataforma, para recolha de informações sobre processos em execução e utilização do sistema (CPU, memória, discos, rede, sensores) (disponível em: https://github.com/giampaolo/psutil; consultado a 24/01/19).
[23] O Matplotlib é uma biblioteca de *plotting* (geração de gráficos) para a linguagem de programação Python (disponível em: https://matplotlib.org/; consultado a 24/01/19).

Este utensílio foi utilizado em conjunto com o psrecord, garantindo o complemento no registo das métricas dos recursos utilizados pelo Snort e Suricata.

- **Htop:** O Htop [54] é uma aplicação gráfica em modo texto para sistemas Unix, que permite visualizar processos e recursos utilizados interactivamente.
  Recorreu-se a esta ferramenta para o acompanhamento gráfico da evolução dos recursos utilizados por cada IDS, assim como para a identificação do processo associado ao IDS a monitorizar (PID – *process identifier*), necessário para os restantes utensílios usados.
- **Netstat:** É uma ferramenta [55] de rede em linha de comandos, que imprime conexões de rede, tabelas de encaminhamento, estatísticas das interfaces de rede, conexões mascaradas e associações *multicast*.
  Aplicada no controlo da interface de rede utilizada no teste dos IDS.
- **Tcpreplay:** Trata-se de um conjunto de instrumentos, gratuitos e de código aberto, para editar e reproduzir tráfego de rede, capturado previamente [56].
  Foi utilizado na retransmissão dos ficheiros pcaps na interface de rede.

3.2. Desenvolvimento do Esquema de Teste

Os testes realizados tiveram como objetivo comum a comparação e avaliação do desempenho geral dos dois sistemas de deteção de intrusões em rede. Alguns testes, mediante a natureza do ficheiro de tráfego empregue, focaram-se mais na componente dos recursos utilizados e processamento do tráfego, enquanto que outros no aspeto mais funcional e no resultado final do IDS, ou seja, os eventos de segurança assinalados.

Relativamente ao ambiente de testes e como foi referido anteriormente, foram mantidas condições de operação o mais similares possível entre ensaios. O trabalho foi executado com as versões 2.9.9.0 do Snort e 4.0.5 do Suricata. As regras utilizadas para ambos os NIDS, dentro das escolhas disponíveis no sistema Security Onion[24], foram as Emergingthreats. É a predefinição do sistema e, possivelmente de entre as escolhas, a solução mais adequada e compatível com ambos os mecanismos de deteção. Como vimos em 2.3.2, este conjunto de regras foi projetado tendo como objetivo principal o complemento das já existentes VRT *rules*, possibilitando uma maior cobertura de ataques conhecidos, ao mesmo tempo que é garantida a maior compatibilidade

[24] Disponível em: https://github.com/Security-Onion-Solutions/security-onion/wiki/Rules; consultado a 25/01/19.

possível entre diferentes NIDS. Salienta-se, no entanto, alguma otimização extra para o Suricata. Estas regras mantiveram-se sempre atualizadas, efetuando-se a verificação e atualização das mesmas, através do PulledPork[25] com o comando *rule-update*, sempre que se mudava de sensor. Foram mantidas todas as predefinições relativas ao conjunto de regras. Para cada teste efetuou-se:

- A limpeza da base de dados com os alertas, sendo a forma mais efetiva a utilização do comando *sguil-db-purge*;
- A mudança de IDS, através dos comandos *nsm_sensor_ps-stop*, *sed -i 's|ENGINE="snort/suricata"|ENGINE="suricata/snort"|g' /etc/nsm/securityonion.conf*, *rule-update* e *nsm_sensor_ps-start*;
- O reinício da máquina virtual.

A comparação e avaliação do desempenho entre Snort e Suricata monitorizou e quantificou as seguintes métricas:

- Processamento;
- Memória;
- Disco;
- Estatísticas da interface de rede e de transmissão do ficheiro pcap (velocidade, tempo, Erros, etc);
- Alertas gerados;
- Perda de pacotes.

Foram precisas algumas ferramentas, mencionadas na subsecção 3.1.2, para a recolha dos dados pretendidos, com a particularidade de algumas delas, Pidstat, Psrecord e Netstat, necessitarem de uma execução paralela ou em segundo plano à retransmissão e processamento dos pacotes pelas ferramentas Tcpreplay e Snort/Suricata, respetivamente. Para tal, de forma a garantir dados mais fidedignos, que representem a evolução temporal do desempenho dos IDS apenas durante o processamento do tráfego, foi escrito um pequeno *script* para a execução dessas tarefas, paralelamente e em segundo plano.

[25] Gestor de regras para Snort e Suricata (disponível em: https://github.com/shirkdog/pulledpork; consultado a 25/01/19).

```
#!/bin/bash

echo
date
echo

netstat -ie

(pidstat -p $1 -rud 1) &

(psrecord $1 --log activity_pcap_$3.txt --plot plot_pcap_$3.png) &

((time sudo tcpreplay -i eth0 -M $2 $3.pcap) && (netstat -ie && date)) &

wait
```

Figura 8 - Shell Script com as tarefas a executar em segundo plano

Este *script*, ilustrado na Figura 8, foi solicitado, via *prompt* (linha de comandos) do Security Onion para cada teste efetuado, pelo seguinte comando: *./script "PID" "Velocidade máxima de transmissão" "Nome do ficheiro pcap"*, onde: "PID" representa o argumento com número identificativo do processo do IDS a monitorizar; "Velocidade máxima de transmissão" deve ser substituído pelo limite máximo de velocidade que desejamos para a transmissão do pcap, expresso em Mbps, pelo Tcpreplay e o "Nome do ficheiro pcap" deve incluir a designação completa, sem extensão, do ficheiro de tráfego que queremos utilizar. A paragem do *script* foi efetuada pelo envio de um sinal *SIGINT* ("CRTL+C"), igualmente pela linha de comandos, após o envio do ficheiro de tráfego pelo Tcpreplay para a interface de rede e do processamento do tráfego pelo NIDS em execução.

Abaixo, uma breve explicação relativamente aos comandos e à nomenclatura que se encontram no *script*:

- netstat -ie: Chama o utensílio netstat e imprime para a linha de comandos informações e estatísticas das interfaces de rede (opção "i"), com ênfase na eth0, de forma visualmente mais agradável (devido à opção "e". Foi executado imediatamente antes e depois da transmissão do tráfego, garantindo dados mais precisos;
- (pidstat -p $1 -rud 1) &: Este comando executa a ferramenta pidstat, em segundo plano ("&"), monitorizando o IDS em estudo, recebendo do comando que chamou o *script* o argumento "PID", que é passado à variável "$1". Imprime para o ecrã informações sobre processamento (opção "u"), memória (opção "r") e disco (opção "d"), de forma sequencial e em intervalos de 1 segundo ("1"), terminando com uma média final da evolução temporal desses dados;

- (psrecord $1 –log activity_pcap_$3.txt –plot plot_pcap_$3.png) &: Neste caso é executado o psrecord, também em segundo plano e ao mesmo tempo, recebendo, igualmente, o identificador do processo a monitorizar por argumento para a variável "$1". Recebe também por argumento o nome do ficheiro pcap para a variável "$3", de forma a incluir esse mesmo nome nos ficheiros que vai criar, com os dados que recolhe da evolução da utilização do processador e memória ("–log activity_pcap_$3.txt" e "–plot plot_pcap_$3.png");
- (sudo tcpreplay -i eth0 -M $2 $3.pcap) &: Comando responsável pela transmissão do ficheiro de tráfego (recebido através do terceiro argumento na chamada do script, pela variável "$3"), paralelamente à aquisição de todos os dados. Esta transmissão é feita através da interface de rede eth0 (opção "i eth0"), com um limite máximo de velocidade de transmissão (opção "M"), que lhe é fornecido igualmente por argumento para a variável "$2";
- wait: comando responsável pelo aguardo da execução completa das tarefas paralelas e em segundo plano;
- time/date/echo: comandos usados na organização ("echo"), confirmação e depuração ("time/date") da informação recolhida.

3.3. Plano de Testes

Tabela 2 - Plano de testes a realizar

Testes	Resumo
Pcap pequeno com tráfego malicioso	Comparação do desempenho Geral dos NIDS, para um pequeno pcap com tráfego malicioso.
Tráfego benigno	Teste com tráfego benigno, ideal para comparação de falsos positivos.
Pcap maior com grande volume de tráfego parcialmente malicioso	Teste com um maior volume de tráfego combinado. Avaliação das capacidades de processamento e deteção dos NIDS, num cenário mais saturado e próximo a uma situação real.
Ataque IRC DoS com pacotes UDP	Teste mais focado na capacidade de deteção. Tráfego malicioso, com ataques de negação de serviço (pacotes UDP).

Ataque IRC DoS com pacotes ICMP	Teste mais focado na capacidade de deteção. Tráfego malicioso, com ataques de negação de serviço (pacotes ICMP, menor volume de tráfego).
Ataque IRC *Port Scan*	Teste mais focado na capacidade de deteção. Tráfego malicioso, com ataques sob a forma de IRC *Port Scan* (tamanho mediano do ficheiro pcap).

3.3.1. Pcap Pequeno com Tráfego Malicioso

O primeiro teste a realizar centra-se no comportamento geral, tanto em recursos consumidos como em alertas gerados, dos dois sistemas de deteção de intrusões, para o processamento de um pequeno ficheiro de tráfego malicioso.

De seguida, apresenta-se uma tabela (Tabela 3) com algumas características relevantes do tráfego a utilizar:

Tabela 3 - Características do tráfego do teste: Pcap pequeno com tráfego malicioso

Pcap Pequeno com Tráfego Malicioso	
Dataset	CTU-Malware-Capture-Botnet-2[26]
Tamanho do pcap	Aproximadamente 3.4 MB
IP da máquina infetada (*Botnet*)	10.0.2.16
IP's relevantes	173.194.70.106; 173.194.70.94
Malware ou principais atividades	O *malware* tenta-se conectar a um grande grupo de endereços IP

3.3.2. Tráfego Benigno

No segundo teste, optar-se-á por um cenário de tráfego normal, com um tamanho consideravelmente maior. Este teste permite, de forma mais credível, comparar ambos os IDS quanto à produção de falsos negativos.

Na Tabela 4, pode-se observar algumas informações do tráfego de rede utilizado.

[26] Disponível em: https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-2/; consultado a 25/01/19.

Tabela 4 - Características do tráfego do teste: Tráfego benigno

Tráfego Benigno	
Dataset	CTU-Normal-7[27]
Tamanho do pcap	Aproximadamente 400 MB
IP da máquina normal	10.0.0.46
IP's relevantes	-
Malware ou principais atividades	Captura P2P normal (Programa P2P Deluge; Navegação web: incluindo Twitter e YouTube; Uso do jabber)

3.3.3. Pcap Maior com Grande Volume de Tráfego Parcialmente Malicioso

Neste terceiro panorama, utiliza-se o maior ficheiro de todos os testes com tráfego combinado, ou seja, tráfego que foi adquirido num computador normal, que após algum tempo foi infetado e voltou a ser "limpo". Este cenário permite testar o desempenho geral e as capacidades dos mecanismos de processamento e deteção do Snort e Suricata, numa situação mais próxima da realidade e de maior saturação de tráfego. Neste caso, atendendo ao volume de tráfego a usar, poderá haver necessidade de proceder a vários ensaios, de forma a garantir a menor perda de pacotes possível.

Relativamente às características do tráfego empregue, salientam-se as apresentadas a seguir (Tabela 5).

Tabela 5 - Características do tráfego do teste: Pcap maior com grande volume de tráfego parcialmente malicioso

Pcap Maior com Grande Volume de Tráfego Parcialmente Malicioso	
Dataset	CTU-Mixed-Capture-1[28]
Tamanho do pcap	Aproximadamente 850 MB
IP da máquina infetada (*Botnet*)	10.0.0.45
IP's relevantes	-
Malware ou principais atividades	Bubble Dock Adware; Navegação web; aplicações (Skype, Facebook, Dropbox, Gmail, etc);

[27] Disponível em: https://mcfp.felk.cvut.cz/publicDatasets/CTU-Normal-7/; consultado a 25/01/19.
[28] Disponível em: https://mcfp.felk.cvut.cz/publicDatasets/CTU-Mixed-Capture-1/; consultado a 25/01/19.

3.3.4. Ataque IRC DoS com Pacotes UDP

O quarto teste, assim como os seguintes, terão uma maior ênfase nas assinaturas detetadas, utilizando-se cenários de tráfego malicioso mais controlado e com ataques identificados.

Este cenário divide-se em dois testes, por haver duas capturas de tráfego ligeiramente diferentes, em relação ao tipo de protocolo e natureza dos pacotes utilizados no ataque. Neste primeiro teste será empregue um ficheiro de tráfego, de tamanho mediano, com ataques do tipo IRC[29] DoS, sob o protocolo UDP.

Na Tabela 6, pode-se ver um resumo do tráfego utilizado.

Tabela 6 - Características do tráfego do teste: Ataque IRC DoS com pacotes UDP

Tráfego da Ataque IRC DoS com Pacotes UDP	
Dataset	CTU-Malware-Capture-Botnet-45[30]
Tamanho do pcap	Aproximadamente 212 MB
IP da máquina infetada (*Botnet*)	147.32.84.165
IP relevante	147.32.96.69
Malware ou principais atividades	IRC DoS; DoS no endereço 147.32.96.69, porta 161, com pacotes UDP

3.3.5. Ataque IRC DoS com Pacotes ICMP

Nesta segunda parte do teste, o ficheiro de tráfego continuará a incluir o mesmo tipo de ataque, mas com pacotes ICMP e de volume substancialmente menor.

Encontra-se, na Tabela 7, as características deste tráfego:

Tabela 7 - Características do tráfego do teste: Ataque IRC DoS com pacotes ICMP

Tráfego da Ataque IRC DoS com Pacotes ICMP	
Dataset	CTU-Malware-Capture-Botnet-45[31]
Tamanho do pcap	Aproximadamente 30 MB

[29] IRC ou Internet Relay Chat é um protocolo, da camada de aplicação, que facilita a comunicação na forma de texto. As conexões de IRC, por norma, não são criptografadas e têm atividades longas, o que constitui um alvo facilitador para ataques DoS/DDoS (disponível em: https://en.wikipedia.org/wiki/Internet_Relay_Chat#Attacks; consultado a 26/01/19).

[30] Disponível em: https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-45/; consultado a 26/01/19.

[31] Disponível em: https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-45/; consultado a 26/01/19.

IP da máquina infetada (*Botnet*)	147.32.84.165
IP relevante	147.32.96.69
Malware ou principais atividades	IRC DDoS; DoS no endereço 147.32.96.69, com pacotes ICMP

3.3.6. Ataque IRC Port Scan

O último ensaio focar-se-á, uma vez mais e como salientado na descrição do teste *Ataque IRC DoS com pacotes UDP*, no tipo de ataque presente no tráfego utilizado e nos alertas gerados pelos sensores de deteção de intrusões. Este cenário de teste inclui IRC *Port Scan*, com um volume de tráfego mediano.

De seguida, pode-se observar um resumo com as particularidades mais significativas do tráfego utilizado (Tabela 8).

Tabela 8 - Características do tráfego do teste: Ataque IRC Port Scan

Ataque IRC Port Scan	
Dataset	CTU-Malware-Capture-Botnet-44[32]
Tamanho do pcap	Aproximadamente 120 MB
IP da máquina infetada (*Botnet*)	147.32.84.165
IP relevante	-
Malware ou principais atividades	IRC PS; Uso do canal IRC para recuperar informações do *bot*; varredura de portas em algumas redes.

[32] Disponível em: https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-44/; consultado a 26/01/19.

4. Resultados dos Testes

Nesta secção do documento, expõem-se os resultados finais dos ensaios anteriormente explicados. Apresenta-se com maior detalhe e como exemplo o primeiro teste – Pcap pequeno com tráfego malicioso – incluindo as estatísticas de envio dos ficheiros de tráfego para a interface de rede, até às várias métricas de desempenho mensuradas entre cada IDS. Para os restantes testes, mostra-se um resumo com os mapas finais dos resultados, podendo-se consultar os mesmos mais pormenorizadamente na secção de anexos. Na sequência, é feita uma análise e avaliação final dos testes e resultados obtidos.

Todo os testes foram realizados, de forma predefinida, com uma velocidade máxima de transmissão do ficheiro pcap de 100 mbps. Repetiram-se os ensaios para taxas de transmissão menores, consoante a necessidade de reduzir a perda de pacotes nos mecanismos de deteção (sobretudo no Snort), procurando a comparação mais equivalente possível. Considerou-se, para esses casos, o desempenho e resultados finais com a taxa de transmissão que permitiu a menor percentagem de queda de pacotes. A taxa de pacotes perdidos é uma das variáveis, no processo de deteção dos IDS, mais importantes. Esta, idealmente nula ou pelo menos equiparável, possibilita uma comparação e avaliação dos mecanismos de deteção mais fiável e equitativa, pois ambos os sistemas de deteção processarão os mesmos pacotes.

4.1. Pcap Pequeno com Tráfego Malicioso

Como mencionado acima, este primeiro teste servirá de modelo, no que toca à informação disponibilizada no corpo do texto, com a totalidade dos dados mensurados e extraídos, que serão mais à frente (capítulo 5) analisados e discutidos, para cada teste.

Neste primeiro ensaio, os resultados finais demonstram que se efetuou com sucesso o envio e processamento do tráfego, com a velocidade máxima de transmissão predefinida (100 mbps), sem perda de pacotes na interface de rede ou nos mecanismos de deteção.

4.1.1. Snort

Nas figuras seguintes (Figuras 9 e 10), observam-se as estatísticas do Tcpreplay (retransmissão do tráfego) assim como da interface de rede. Nelas, pode-se constatar alguns indicadores relevantes para esta fase inicial do teste, como a existência ou não de erros e a velocidade ou taxa de transmissão de pacotes.

```
Actual: 18523 packets (3310685 bytes) sent in 0.48 seconds.              Rated: 6897260.5 bps, 52.62 Mbps, 38589.58 pps
Statistics for network device: eth0
        Attempted packets:         18523
        Successful packets:        18522
        Failed packets:            1
        Retried packets (ENOBUFS): 0
        Retried packets (EAGAIN):  0
```

Figura 9 - Pcap pequeno com tráfego malicioso - Snort: Estatísticas de envio do ficheiro pcap (Tcpreplay)

```
Kernel Interface table
eth0      Link encap:Ethernet  HWaddr 52:54:00:f7:21:b7
          inet addr:192.168.122.25  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fef7:21b7/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:221 errors:0 dropped:0 overruns:0 frame:0
          TX packets:18709 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:21150 (21.1 KB)  TX bytes:3330065 (3.3 MB)
```

Figura 10 - Pcap pequeno com tráfego malicioso - Snort: Estatísticas da interface de rede após transmissão do tráfego (Netstat)

Nas Figuras 11 e 12 são apresentadas as informações relativas ao consumo de recursos de sistema pelos NIDS, primeiramente, sob a forma gráfica, a evolução temporal da utilização do CPU e RAM e, na segunda figura, as médias finais da requisição de todos os recursos pertinentes.

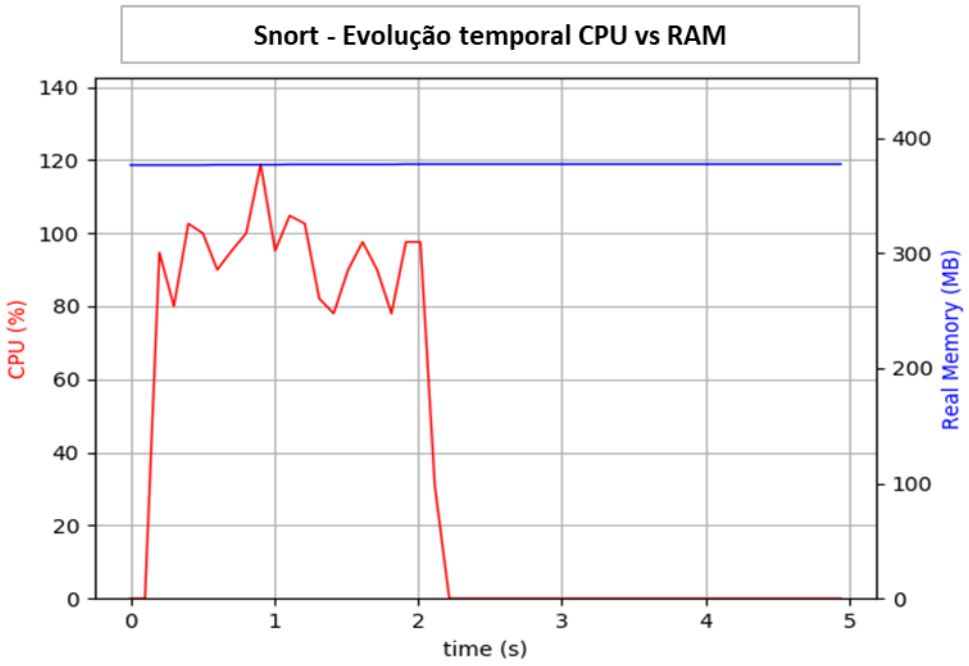


Figura 11 - Pcap pequeno com tráfego malicioso - Snort: CPU e RAM utilizados (Psrecord)

```
Average:      UID       PID    %usr %system  %guest    %CPU   CPU  Command
Average:     1001      5996   35,60    1,00    0,00   36,60     -  snort

Average:      UID       PID  minflt/s  majflt/s     VSZ    RSS   %MEM  Command
Average:     1001      5996     41,40      0,00  741276 386090   2,75  snort

Average:      UID       PID   kB_rd/s   kB_wr/s kB_ccwr/s  Command
Average:     1001      5996     -1,00     -1,00     -1,00  snort
```

Figura 12 - Pcap pequeno com tráfego malicioso - Snort: Médias dos recursos utilizados (Pidstat)

Na Figura 13, observa-se a assinatura detetada pelo NIDS, bem como a quantidade de eventos gerados para a mesma assinatura, na interface de monitorização Squert.

QUEUE	SC	DC	ACTIVITY	LAST EVENT	SIGNATURE
313	1	1		17:56:34	ET TROJAN Possible Zeus GameOver Connectivity Check

Figura 13 - Pcap pequeno com tráfego malicioso - Snort: Alertas gerados (Squert)[33]

4.1.2. Suricata

Nas figuras seguintes (Figuras 14 e 15), observam-se as estatísticas do Tcpreplay (retransmissão do tráfego) assim como da interface de rede.

```
Actual: 18523 packets (3310685 bytes) sent in 0.32 seconds.          Rated: 10345891.0 bps, 78.93 Mbps, 57884.38 pps
Statistics for network device: eth0
        Attempted packets:         18523
        Successful packets:        18522
        Failed packets:            1
        Retried packets (ENOBUFS): 0
        Retried packets (EAGAIN):  0
```

Figura 14 - Pcap pequeno com tráfego malicioso - Suricata: Estatísticas de envio do ficheiro pcap (Tcpreplay)

```
Kernel Interface table
eth0      Link encap:Ethernet  HWaddr 52:54:00:f7:21:b7
          inet addr:192.168.122.25  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fef7:21b7/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:2440 errors:0 dropped:0 overruns:0 frame:0
          TX packets:20427 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2629791 (2.6 MB)  TX bytes:3577069 (3.5 MB)
```

Figura 15 - Pcap pequeno com tráfego malicioso - Suricata: Estatísticas da interface de rede após transmissão do tráfego (Netstat)

[33] Interface Squert: QUEUE – número de eventos agrupados na fila; SC – número de IP's de origem distintos para o alerta; DC – número de IP's de destino distintos para o alerta; ACTIVITY – número de eventos por hora do alerta; LASTEVENT – horário da última ocorrência do evento; SIGNATURE – Assinatura do IDS para o alerta (disponível em: https://github.com/Security-Onion-Solutions/security-onion/wiki/Squert; consultado a 26/01/19).

Nas Figuras 16 e 17 são apresentadas as informações relativas ao consumo de recursos de sistema pelos NIDS.

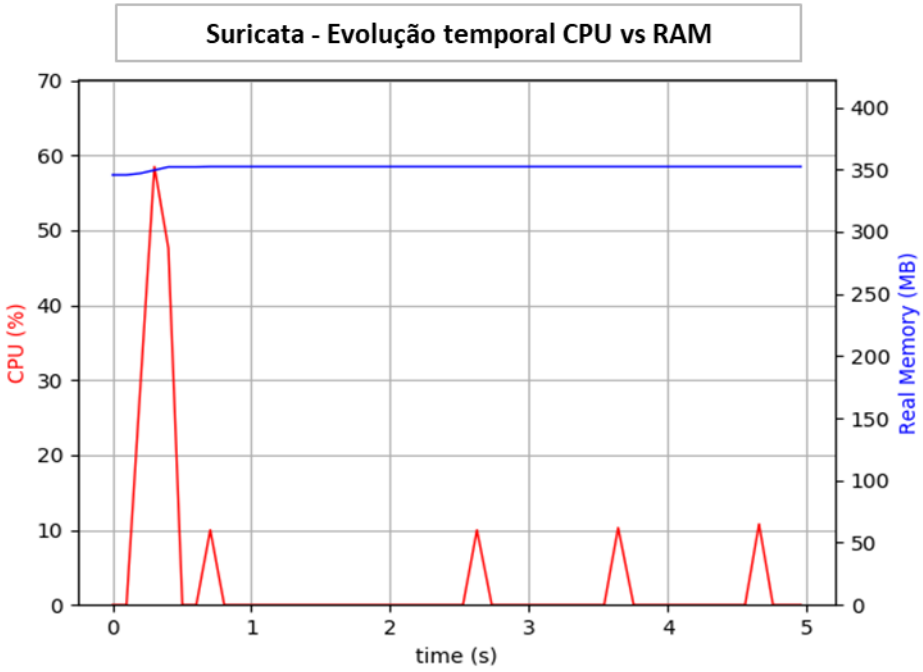


Figura 16 - Pcap pequeno com tráfego malicioso - Suricata: CPU e RAM utilizados (Psrecord)

```
Average:      UID       PID    %usr %system  %guest    %CPU   CPU  Command
Average:     1001      4335    3,40    0,20    0,00    3,60     -  Suricata-Main

Average:      UID       PID  minflt/s  majflt/s     VSZ    RSS   %MEM  Command
Average:     1001      4335    350,80      0,00  726524 360796   2,57  Suricata-Main

Average:      UID       PID   kB_rd/s   kB_wr/s kB_ccwr/s  Command
Average:     1001      4335     -1,00     -1,00     -1,00  Suricata-Main
```

Figura 17 - Pcap pequeno com tráfego malicioso - Suricata: Médias dos recursos utilizados (Pidstat)

Na Figura 18, observam-se as assinaturas detetadas pelo NIDS, bem como a quantidade de eventos gerados para a mesma assinatura.

QUEUE	SC	DC	ACTIVITY	LAST EVENT	SIGNATURE
313	1	1		19:10:51	ET TROJAN Possible Zeus GameOver Connectivity Check
2	1	1		19:10:51	SURICATA zero length padN option
1	1	1		19:11:52	SURICATA HTTP unable to match response to request

Figura 18 - Pcap pequeno com tráfego malicioso - Suricata: Alertas gerados (Squert)

4.1.3. Snort vs Suricata

As Tabelas 9 e 10 mostram de forma mais resumida e agregada os dados mais relevantes, anteriormente ilustrados. A primeira (Tabela 9), diz respeito aos valores máximos e médios da utilização de recursos de hardware, enquanto que a Tabela 10 comporta a informação relativa ao desempenho funcional de deteção.

Tabela 9 - Pcap pequeno com tráfego malicioso - Snort vs Suricata: Valores máximos e médios dos recursos de hardware consumidos

Valores Máximos e Médios	Snort	Suricata
CPU (Máximo)	118.900 %	58.500 %
RAM (Máximo)	377.195 MB	352.340 MB
CPU (Média)	36.60 %	3.60 %
RAM (Média)	2.75 %	2.57 %
Minflt/s[34] (Média)	41.40	350.80
Majflt/s[35] (Média)	0.00	0.00
Disco (Média)	-	-

Tabela 10 - Pcap pequeno com tráfego malicioso - Snort vs Suricata: Desempenho de deteção

Alertas Gerados	Snort	Suricata
ET TROJAN Possible Zeus GameOver Connectivity Check	313	313
SURICATA zero length padN option	0	2
SURICATA HTTP unable to match response to request	0	1
Total	313	316
Perda de Pacotes	0 %	0 %

4.2. Tráfego Benigno

Este teste e os seguintes são apresentados de forma resumida, podendo-se consultar a informação complementar na secção de anexos (capítulo 8).

[34] *Minflt/s: número total, por segundo, de pequenos erros/falhas da tarefa, que não obrigam a uma leitura de páginas de memória do disco.

[35] Majflt/s: número total, por segundo, de graves erros/falhas da tarefa, que obrigam a uma leitura de páginas de memória do disco.

As ferramentas Tcpreplay e Netstat revelam uma retransmissão do tráfego sem quaisquer erros ou perda de pacotes, à primeira tentativa, assim como no processo de deteção dos NIDS.

As tabelas que se seguem (Tabelas 11 e 12) mostram de forma mais resumida e agregada os dados mais relevantes, anteriormente ilustrados. A primeira (Tabela 11) diz respeito aos valores máximos e médios da utilização de recursos de hardware, enquanto que a Tabela 12 comporta a informação relativa ao desempenho funcional de deteção.

Tabela 11 - Tráfego benigno - Snort vs Suricata: Valores máximos e médios dos recursos de hardware consumidos

Valores Máximos e Médios	**Snort**	**Suricata**
CPU (Máximo)	108.100 %	32.400 %
RAM (Máximo)	379.016 MB	355.012 MB
CPU (Média)	53.90 %	12.63 %
RAM (Média)	2.76 %	2,58 %
Minflt/s (Média)	4.90	14.98
Majflt/s (Média)	0.00	0,00
Disco (Média)	-	-

Tabela 12 - Tráfego benigno - Snort vs Suricata: Desempenho de deteção

Alertas Gerados	**Snort**	**Suricata**
ET P2P BitTorrent peer sync	0	4
ET DROP Spamhaus DROP Listed Traffic Inbound group 15	0	1
ET P2P BitTorrent DHT announce_peers request	1	1
ET P2P BitTorrent DHT nodes reply	1	1
ET P2P BitTorrent DHT ping request	1	1
ET P2P Vuze BT UDP Connection (5)	1	1
Total	4	9
Perda de Pacotes	0 %	0 %

4.3. Pcap Maior com Grande Volume de Tráfego Parcialmente Malicioso

Neste terceiro cenário, salienta-se a necessidade de proceder a três ensaios, como previsto anteriormente (3.3.3), com velocidades de transmissão cada vez menores, de forma a garantir o correto envio dos pacotes para a rede pelo Tcpreplay e a redução da perda de pacotes nos IDS, sobretudo no Snort. Este, apresentou uma taxa de pacotes perdidos, para as velocidades máximas de transmissão de 100, 50 e 30 mbps, de 35.4, 10.4 e 0.0 %, respetivamente. O Suricata apresentou uma reduzida perda de pacotes, apenas na primeira tentativa, anulando-se a mesma ao segundo ensaio.

As tabelas seguintes (Tabelas 13 e 14) mostram de forma mais resumida e agregada os dados mais relevantes, anteriormente ilustrados. A primeira (Tabela 13) diz respeito aos valores máximos e médios da utilização de recursos de hardware, enquanto que a Tabela 14 comporta a informação relativa ao desempenho funcional de deteção.

Tabela 13 - Pcap maior com grande volume de tráfego parcialmente malicioso - Snort vs Suricata: Valores máximos e médios dos recursos de hardware consumidos

Valores Máximos e Médios	Snort	Suricata
CPU (Máximo)	118.900 %	55.800 %
RAM (Máximo)	395.938 MB	469.051 MB
CPU (Média)	29.91 %	8.72 %
RAM (Média)	2.81 %	3.00 %
Minflt/s (Média)	19.41	128.51
Majflt/s (Média)	0.00	0,00
Disco (Média)	-	-

Tabela 14 - Pcap maior com grande volume de tráfego parcialmente malicioso - Snort vs Suricata: Desempenho de deteção

Alertas Gerados	Snort	Suricata
ET POLICY PE EXE or DLL Windows file download HTTP	49	36
ET POLICY Executable served from Amazon S3	48	16
ET POLICY User-Agent (NSIS_Inetc (Mozilla)) - Sometimes used by hostile installers	27	27

ET CHAT Skype User-Agent detected	22	22
ET CHAT Skype VOIP Checking Version (Startup)	18	18
ET POLICY Dropbox.com Offsite File Backup in Use	15	42
ET INFO - Applet Tag In Edwards Packed JavaScript	12	12
ET POLICY SSLv3 outbound connection from client vulnerable to POODLE attack	8	0
ET POLICY Outdated Flash Version M2	5	3
ET POLICY Vulnerable Java Version 1.8.x Detected	4	0
ET POLICY Outdated Flash Version M1	1	1
ET POLICY Dropbox Client Broadcasting	1	1
SURICATA HTTP unable to match response to request	0	1679
Total	210	1857
Perda de Pacotes	0 %	0 %

4.4. Ataque IRC DoS com Pacotes UDP

Neste teste, a primeira tentativa de transmissão com velocidade máxima de 100 mbps revelou uma perda de pacotes no Snort de 25.96 %, não se registando, por outro lado, qualquer perda no Suricata. O teste foi então repetido para um limite de 70 mbps na velocidade de transmissão do ficheiro pcap, obtendo-se um processamento do tráfego pelos NIDS sem perda de pacotes.

As Tabelas 15 e 16 mostram de forma mais resumida e agregada os dados mais relevantes, anteriormente ilustrados. A primeira (Tabela 15) diz respeito aos valores máximos e médios da utilização de recursos de hardware, enquanto que a Tabela 16 comporta a informação relativa ao desempenho funcional de deteção.

Tabela 15 - Ataque IRC DoS com pacotes UDP - Snort vs Suricata: Valores máximos e médios dos recursos de hardware consumidos

Valores Máximos e Médios	Snort	Suricata
CPU (Máximo)	125.700 %	30.800 %
RAM (Máximo)	378.207 MB	374.414 MB
CPU (Média)	72.91 %	7.63 %
RAM (Média)	2.76 %	2.68 %
Minflt/s (Média)	1.11	163.70

Majflt/s (Média)	0.00	0,00
Disco (Média)	-	-

Tabela 16 - Ataque IRC DoS com pacotes UDP - Snort vs Suricata: Desempenho de deteção

Alertas Gerados	**Snort**	**Suricata**
SURICATA FRAG IPv4 Fragmentation overlap	0	8891
SURICATA UDP packet too small	0	4333
SURICATA UDP invalid header length	0	4094
ET CHAT IRC PONG response	18	102
ET CHAT IRC USER command	0	100
ET CHAT IRC NICK command	2	100
ET CHAT IRC JOIN command	2	100
ET CHAT IRC PRIVMSG command	35	100
ET CHAT IRC USER Likely bot with 0 0 colon checkin	2	100
ET CHAT IRC PING command	23	26
Total	82	17946
Perda de Pacotes	0 %	0 %

4.5. Ataque IRC DoS com Pacotes ICMP

Neste segundo cenário, complementar ao anterior, pode-se observar nos dados apresentados que se efetuou com sucesso o envio do ficheiro pcap, a velocidades muito próximas dos 100 mbps.

As Tabelas 17 e 18 mostram de forma mais resumida e agregada os dados mais relevantes, anteriormente ilustrados. A primeira (Tabela 17) diz respeito aos valores máximos e médios da utilização de recursos de hardware, enquanto que a Tabela 18 comporta a informação relativa ao desempenho funcional de deteção.

Tabela 17 - Ataque IRC DoS com pacotes ICMP- Snort vs Suricata: Valores máximos e médios dos recursos de hardware consumidos

Valores Máximos e Médios	Snort	Suricata
CPU (Máximo)	32.400 %	29.300 %
RAM (Máximo)	378.098 MB	352.184 MB
CPU (Média)	2.25 %	6.43 %
RAM (Média)	2.76 %	2.57 %
Minflt/s (Média)	10.50	5.71
Majflt/s (Média)	0.00	0,00
Disco (Média)	-	-

Tabela 18 - Ataque IRC DoS com pacotes ICMP - Snort vs Suricata: Desempenho de deteção

Alertas Gerados	Snort	Suricata
SURICATA ICMPv4 unknown type	0	26777
SURICATA ICMPv4 unknown code	0	1484
SURICATA ICMPv4 unknown version	0	14
ET CHAT IRC PONG response	2	2
ET CHAT IRC PRIVMSG command	17	0
ET CHAT IRC PING command	9	0
ET CHAT IRC NICK command	2	0
ET CHAT IRC JOIN command	2	0
ET CHAT IRC USER Likely bot with 0 0 colon checkin	2	0
Total	34	28277
Perda de Pacotes	0 %	0 %

4.6. Ataque IRC Port Scan

Neste último experimento, apesar de três tentativas de retransmissão do tráfego, a 100, 50 e 10 mbps, não foi possível anular por completo a perda de pacotes do Snort, ficando nos 36.27 %. O Suricata começou por apresentar 4.47 % de perda de pacotes, mas foi possível eliminar esta limitação logo na segunda tentativa.

Nas Tabelas 19 e 20 vê-se de forma mais resumida e agregada os dados mais relevantes, anteriormente ilustrados. A primeira (Tabela 19) diz respeito aos valores máximos e médios da

utilização de recursos de hardware, enquanto que a Tabela 20 comporta a informação relativa ao desempenho funcional de deteção.

Tabela 19 - Ataque IRC Port Scan - Snort vs Suricata: Valores máximos e médios dos recursos de hardware consumidos

Valores Máximos e Médios	Snort	Suricata
CPU (Máximo)	112.800 %	31.600 %
RAM (Máximo)	401.801 MB	386.805 MB
CPU (Média)	26.11 %	3.30 %
RAM (Média)	2.83 %	2.71 %
Minflt/s (Média)	58.27	91.05
Majflt/s (Média)	0.00	0,00
Disco (Média)	-	-

Tabela 20 - Ataque IRC Port Scan - Snort vs Suricata: Desempenho de deteção

Alertas Gerados	Snort	Suricata
ET CHAT IRC PONG response	1436	6586
ET CHAT IRC PRIVMSG command	500	2694
ET CHAT IRC PING command	1408	1657
ET CHAT IRC USER command	0	162
ET CHAT IRC NICK command	3	162
ET CHAT IRC JOIN command	3	162
ET CHAT IRC USER Likely bot with 0 0 colon checkin	3	162
ET CURRENT_EVENTS SUSPICIOUS IRC - PRIVMSG *.(exe|tar|tgz|zip) download command	40	40
Total	3393	11625
Perda de Pacotes	36.286 %	0 %

5. Análise e Discussão

5.1. Análise dos Resultados

5.1.1. Pcap Pequeno com Tráfego Malicioso

Como foi dito na secção 4.1, este primeiro teste foi efetuado com sucesso à taxa de transmissão predefinida, embora, pelas estatísticas fornecidas pelo Tcpreplay (Figuras 9 e 14), se repare que a velocidade real de transmissão ficou um pouco abaixo do limite estabelecido (não ultrapassando os 80 mbps). Tal pode ser explicado, em parte, pelo tamanho reduzido do ficheiro usado (aproximadamente 3 mbps), não havendo espaço temporal para atingir a taxa máxima de transmissão. Verifica-se, também, um único erro no envio de um pacote para ambos os NIDS (Figuras 9 e 14), por estar corrompido ou apresentar um MTU maior que o da interface de rede.

Relativamente ao desempenho, ambos os IDS processaram com rapidez o pequeno ficheiro pcap utilizado. Ainda assim, o Snort apresenta um maior consumo de recursos, tanto em CPU como RAM (Tabela 9). A memória utlizada praticamente não sofreu alteração com o processamento deste tráfego, mas, por predefinição e em repouso, o Snort ocupa maior espaço de memória do que o Suricata. Em contrapartida, o Suricata apresenta um maior número de pequenos erros de memória, durante a atividade.

Em relação aos alertas assinalados (Tabela 10), verificou-se um cenário bastante similar, com o Suricata a levar uma ligeira vantagem no número de assinaturas detetadas. Uma das regras ("ET TROJAN Possible Zeus GameOver Connectivity Check") e o número de eventos da mesma assinalados foi coincidente entre ambos os mecanismos de deteção, tratando-se, pela informação disponibilizada no *dataset* e no alerta, de verdadeiros positivos, como se evidencia nas Figuras 19 e 20.

```
Infected Machines:
Windows Name: Win2, IP: 10.0.2.16 (Label: Botnet-V1)

Malware tries to connect to a big group of a IP addresses and only a few of them answer.
From time to time (2, 7, 4 hours) tries to contact the big group of IPs again.

10th July 2013
TCP:
173.194.70.106 each 30 minutes, checking google.com
```

Figura 19 - Dataset - Pcap pequeno com tráfego malicioso: Informações relevantes

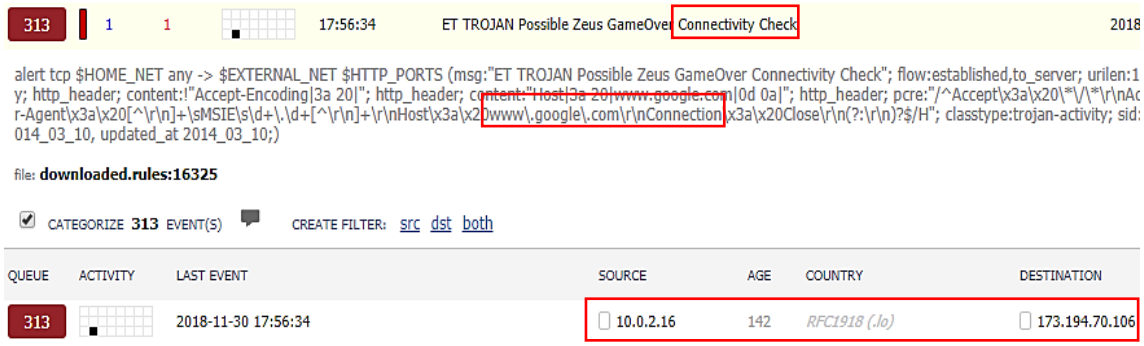


Figura 20 - Squert - Pcap pequeno com tráfego malicioso: Detalhes do alerta gerado

Por outro lado, o Suricata indicou mais duas assinaturas: uma delas ("SURICATA HTTP unable to match response to request") seria passível de ser considerada um verdadeiro positivo, ainda assim, este é um alerta geralmente reportado[36] como falso positivo aleatório em acessos a sites fidedignos, como é o caso. O último alerta reportado ("SURICATA zero length padN option") trata-se, atendendo igualmente às informações do *dataset* e a alguma documentação online[37], de um falso positivo, associado ao descodificador do mecanismo de deteção do IDS Suricata.

Conclui-se, neste primeiro teste, que o Suricata desempenhou melhor que o Snort, em termos dos recursos utilizados para processar o mesmo tráfego, contudo apresentando dois alertas provavelmente falsos positivos.

5.1.2. Tráfego Benigno

Por se tratar de um ficheiro de tráfego maior, o tempo de transmissão foi, em relação ao primeiro teste, substancialmente superior (Anexos – Tráfego Benigno).

Como esperado, verificou-se um aumento dos recursos utilizados por ambos os sistemas de deteção. Mantendo-se, ainda assim e similarmente ao primeiro teste, a vantagem para o Suricata neste quesito (Tabela 11).

Quanto a alertas gerados (Tabela 12), neste caso são todos falsos positivos por se tratar de um tráfego benigno. Ambos os NIDS assinalaram um baixo número de eventos de segurança, quatro do Snort contra nove do Suricata, contemplando duas assinaturas apenas indicadas por este último. Levando o Snort vantagem sobre o Suricata, neste aspeto.

[36] Disponível em: https://github.com/jflsakfja/suricata-rules/blob/master/list.txt; consultado a 28/01/19.
[37] Disponível em: https://github.com/OISF/suricata/blob/master/rules/decoder-events.rules; consultado a 28/01/19.

Como balanço final, salienta-se a maior facilidade do Suricata no tratamento do tráfego face aos recursos consumidos, embora na questão funcional tenha apresentado um número superior de falsos positivos.

5.1.3. Pcap Maior com Grande Volume de Tráfego Parcialmente Malicioso

Analisando os dados finais da monitorização dos recursos de *hardware* consumidos (Tabela 13), repara-se que uma das métricas mensuradas sofre uma mudança de paradigma, em relação aos anteriores experimentos, trata-se da quantidade de memória utilizada pelos IDS. Desta vez, é o Suricata que consome maior memória, para fazer face ao volume de tráfego transmitido na rede. O restante desempenho continua similar ao padrão que se vinha a observar.

Em relação aos alertas gerados (Tabela 14), pelas informações disponibilizadas pelo repositório do tráfego e pelo conteúdo do mesmo, não é possível aferir acerca da exatidão da deteção. Contudo, pode-se salientar a similaridade entre as regras assinaladas, excetuando-se a já analisada: "SURICATA HTTP unable to match response to request". Nesse caso, acaba por haver até uma ligeira vantagem no número de alertas gerados pelo Snort.

No geral, este teste comprovou a maior agilidade do mecanismo de deteção do Suricata em lidar com volumes de tráfego maiores e mais complexos, ainda que para isso tenha aumentado significativamente o consumo de recursos do sistema, nomeadamente a RAM.

5.1.4. Ataque IRC DoS com Pacotes UDP

Em termos da requisição de recursos de sistema, o cenário manteve-se equivalente à norma vista até aqui (Tabela 15).

Relativamente aos alertas gerados (Tabela 16) e atendendo ao intuito inicial deste teste e do *dataset* disponibilizado com ataques conhecidos, deve-se fazer algumas considerações. No geral, excetuando-se três exclusivas do Suricata, as assinaturas detetadas foram equivalentes entre ambos os IDS, diferindo apenas no número de vezes que foram assinaladas, onde o Suricata apresenta um número bastante mais elevado. A maior parte dos alertas revelam ser verdadeiros positivos, quando analisados e comparados os dados relativos aos mesmos com os do *dataset*, como se pode observar nalguns exemplos demonstrados abaixo (Figuras 21 e 22).

```
- Infected hosts
   - 147.32.84.165:

We are going to DoS the address 147.32.96.69 with UDP packets to port 161

We DOS with UDP some IPs.

These are the IRC commands:
NICK Pepe024268
USER ghnnza 0 0 :Pepe024268
:pepe|2!~kvirc@cmpgw-27.felk.cvut.cz PRIVMSG #zarasa48 :.login zarasa48
PRIVMSG #zarasa48 :.::[MaInFrAmE]::. Password Accettata, Welcome to x0n3-Satan.
```

Figura 21 - Dataset - Ataque IRC DoS com pacotes UDP: Informações relevantes

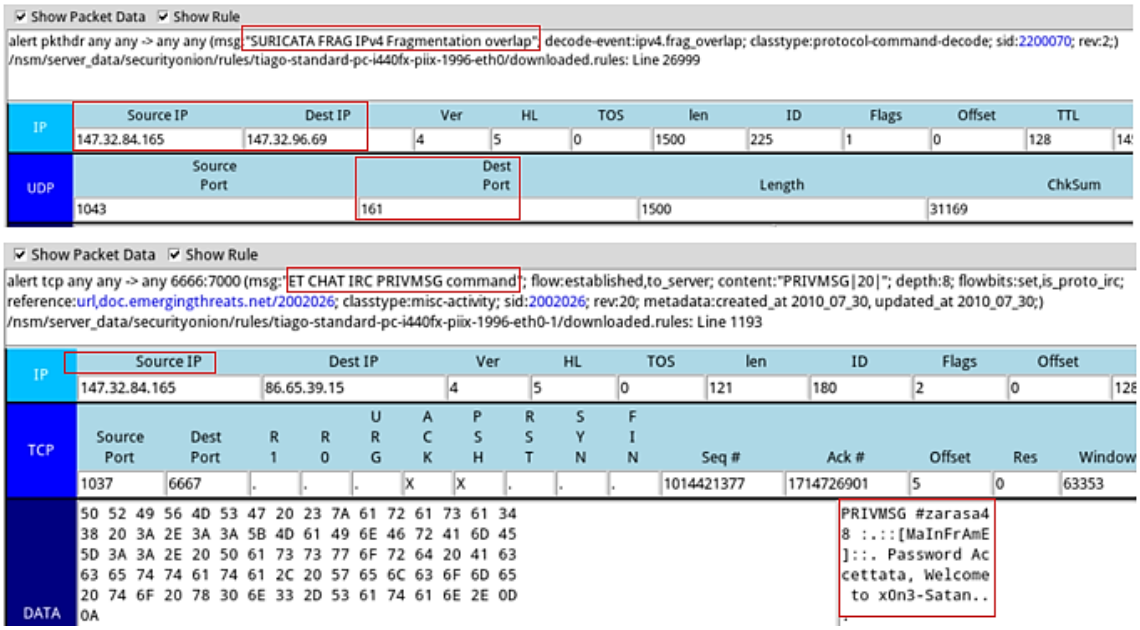


Figura 22 - Sguil - Ataque IRC DoS com pacotes UDP: Detalhes dos alertas gerados

Duas assinaturas detetadas pelo Suricata ("SURICATA UDP packet too small" e "SURICATA UDP invalid header length"), pertencem, igualmente como visto previamente, a regras específicas do mecanismo de descodificação do tráfego deste IDS, não sendo encontrado qualquer relação entre as mesmas e os ataques identificados do tráfego.

De modo geral, o Suricata desempenhou melhor o seu papel, embora com a deteção imprevista de duas regras.

5.1.5. Ataque IRC DoS com Pacotes ICMP

Relativamente aos valores de desempenho (Tabela 17), monitorizados pelo Psrecord e Pidstat, salienta-se a mudança de paradigma relativamente ao uso do processador, que foi

significativamente superior no Suricata e, por outro lado, ao maior número de erros de memória na atividade do Snort.

Estas mudanças no padrão de desempenho relacionam-se com a diferença na natureza do tráfego utilizado, em relação ao último teste, e, consequentemente, com os alertas gerados por ambos os IDS (Tabela 18). Neste caso, o Suricata detetou quatro assinaturas distintas, entre as quais três exclusivas dele próprio, com relação ao tipo de ataque efetuado (ICMP DoS), como se pode observar mais abaixo (Figuras 23 e 24), gerando um grande número de alertas para estas regras. Por outro lado, observa-se que falhou ao não assinalar regras diretamente relacionadas ao ataque de base (IRC DoS), contrariamente ao Snort.

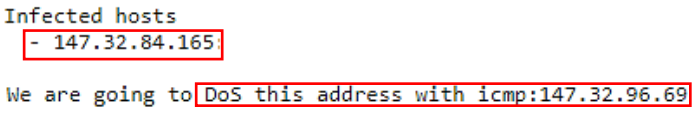


Figura 23 - Dataset - Ataque IRC DoS com pacotes ICMP: Informações relevantes

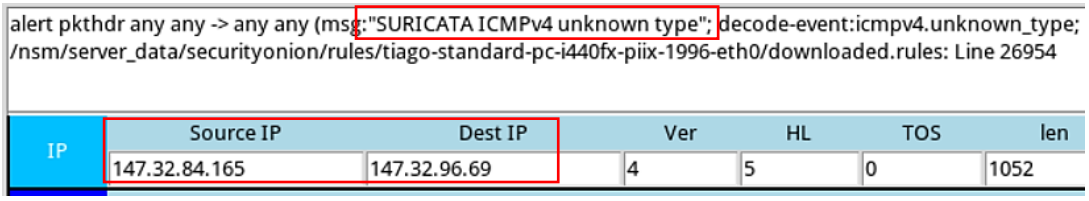


Figura 24 - Sguil - Ataque IRC DoS com pacotes ICMP: Detalhes dos alertas gerados

Este teste foi bastante equilibrado no desempenho geral dos dois NIDS, acabando ambos por se evidenciar nuns aspetos, em detrimento de outros.

5.1.6. Ataque IRC Port Scan

Como vimos em 4.4, foram necessários alguns ensaios objetivando a dissolução da perda de pacotes, possivelmente associado ao tipo de ataque e tráfego presentes, que não serão tão bem processados pelos mecanismos de deteção.

Em termos de recursos de sistema consumidos (Tabela 19), voltou-se a registar o padrão inicial, com o Suricata em melhor posição para todos os aspetos, excetuando-se as falhas da tarefa.

Em termos de alertas gerados (Tabela 20), apesar da limitação do Snort no processamento dos pacotes, observou-se um equilíbrio no tipo de assinaturas detetadas, havendo apenas discrepância numa delas, onde o Snort apresenta um cenário de falso negativo, evidenciado nas

Figuras 25 e 26. Ainda assim, e justificado em parte pela perda de pacotes do Snort, o Suricata indica para cada assinatura um número substancialmente mais elevado de intrusões, face ao homólogo.

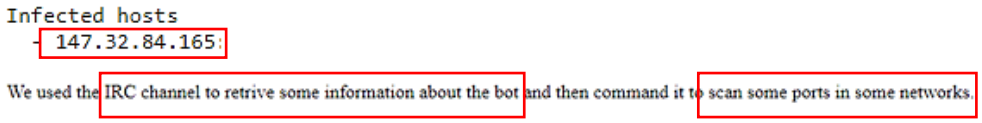


Figura 25 - Dataset - Ataque IRC Port Scan: Informações relevantes

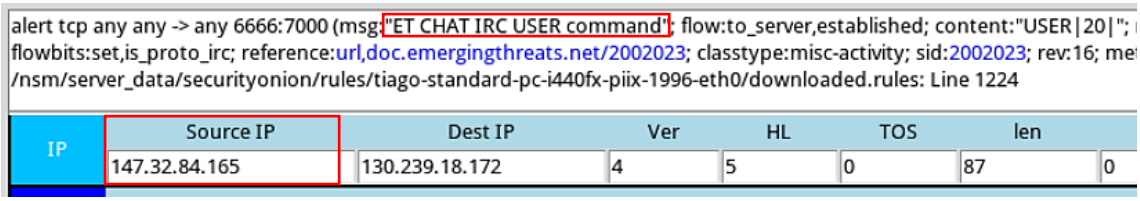


Figura 26 - Sguil - Ataque IRC Port Scan: Detalhes dos alertas gerados

5.2. Resumo e Discussão

Os testes realizados evidenciaram um conjunto de resultados maioritariamente expectáveis. No desempenho ao nível dos recursos solicitados, assistimos a uma clara tendência de um Snort mais exigente no CPU, trabalhando algumas vezes próximo ou além do limite de processamento *single-core*. Este é um ponto interessante de analise, uma vez que não seria esperado observar-se consumos de processamento acima dos 100%, tendo em conta que o mecanismo de deteção do Snort é *single-threaded* e, como referido anteriormente em 3.1.1, apenas se utilizou uma instância (PF_RING) por motor de deteção. Contudo, este sistema IDS consegue, similarmente às técnicas *multi-threaded*, tirar algum proveito dos processadores com múltiplos núcleos, através do escalonamento *multi-core* do sistema operativo, o que poderá justificar o acumular, por breves instantes, de percentagens de processamento em mais do que um núcleo do CPU [57]–[59]. Relativamente ao consumo de memória e erros na mesma, verificou-se uma clara tendência de aumento proporcional à maior complexidade e volume de tráfego a analisar, sobretudo no NIDS Suricata. Esta particularidade pode ser justificada pela maior sobrecarga na gestão do mecanismo de deteção com múltiplas *threads* [60].

Em geral, como seria de esperar, ambos os NIDS requisitaram mais recursos, tanto em CPU como em memória, e evidenciaram maior sobrecarga à medida que o volume ou complexidade do tráfego ia aumentando. No entanto, o Snort apresentou uma clara propensão nessa sobrecarga,

sobretudo na utilização do processador face ao Suricata. Este, por outro lado, destaca-se no maior consumo de memória, a partir de um certo volume e dificuldade do tráfego a analisar.

Na parte mais funcional dos dois sistemas de deteção de intrusões em rede verificou-se, em geral, um melhor desempenho do Suricata ao nível do processamento do tráfego, com perdas de pacotes quase nulas, contrariamente ao Snort, mas também nas assinaturas detetadas em cada cenário de teste. Esta maior facilidade em processar e analisar a maioria dos tráfegos de rede e respetivos ataques selecionados pode ser explicada pelas diferenças presentes em cada arquitetura de deteção. Desde logo, o mecanismo *multi-threaded* do Suricata, em vez do *single-threaded* do Snort, que permite, ao primeiro, executar múltiplas tarefas de deteção simultaneamente, ajudando na eficiência do processamento e análise do tráfego. Para além desta principal diferença, o mecanismo de deteção e fluxo de ambos os NIDS têm outras pequenas particularidades, entre as quais o uso dos pré-processadores por parte do Snort, enquanto o Suricata os substitui pelos descodificadores, motor de fluxo diferenciado, análise HTTP entre outros, podendo resultar num tratamento diferenciado do tráfego [60], [61]. Complementarmente, o conjunto de regras utilizado, embora compatível com ambos os NIDS, pode ser carregado e interpretado de forma diferente por cada motor de deteção, em particular pelo Suricata, que possui nas ET rules algumas assinaturas próprias ou melhor otimizadas. Este aspeto foi particularmente notório nos últimos ensaios realizados, onde se verificaram algumas discrepâncias nas regras assinaladas.

Relativamente aos testes onde se encontrou maiores dificuldades no tratamento do tráfego e respetivos ataques, pode-se salientar o teste *Ataque IRC Port Scan*, onde o Snort apresentou a maior sobrecarga e taxa de queda de pacotes. Este fator pode ser revelador de alguma limitação deste NIDS no tratamento do tráfego e dos ataques em causa (Canal IRC e *Port Scan*), acabando por não assinalar uma regra erradamente (falso negativo) e com uma frequência de alertas gerados para as restantes assinaturas bastante abaixo do Suricata. Por outro lado, no teste *Ataque IRC DoS com pacotes ICMP*, o Suricata deteta um elevado número de alertas para assinaturas próprias, mas, provavelmente devido a essa sobrecarga nas assinaturas próprias detetadas, não assinala outras expectáveis e presentes na deteção do Snort.

Quanto a falsos positivos ou alertas dúbios (*Grey Positives*), a tendência para um maior registo dos mesmos incide sobre o NIDS Suricata, em boa parte por algumas regras exclusivas assinaladas em larga escala. Estas precisam ser revistas, configuradas e em último caso desativadas, uma vez que algumas obtêm frequentemente correspondência com algum tipo de

atividade maliciosa, aquando por exemplo de tráfego de rede legítimo como pedidos web ou DNS [46], [62]. O teste *Tráfego benigno*, realizado com ênfase na questão dos falsos positivos, evidenciou uma falha na exatidão da análise de tráfego P2P normal, mais concretamente em *downloads* via *torrent*. Atendendo a que as regras e o número de vezes sinalizadas pelos dois NIDS foram bastante similares, pode-se deduzir que a causa, mais plausível, para esta limitação está sobretudo no conjunto de regras utilizado, precisando este de ser ajustado para este tipo de tráfego e comunicação em rede.

Como vimos anteriormente, existem dois métodos de deteção: o baseado em anomalias e o baseado em assinaturas. Este último, utilizado pelos NIDS em estudo, garante à partida uma menor percentagem de falsos positivos. Ainda assim e como podemos observar nos testes efetuados, sabemos que estes continuam a ser gerados numa quantidade muito acima do ideal, afetando significativamente o desempenho e fiabilidade de um IDS. Este tipo de alerta indevido acontece quando um determinado tráfego de rede legítimo corresponde a uma assinatura de um conjunto de regras para tráfego malicioso, levando o IDS, que trabalha com esse conjunto, a gerar um falso alerta. De forma a tentar colmatar este flagelo do universo de deteção de intrusões, foram já realizados alguns estudos e experiências com técnicas e complementos aos IDS tradicionais e isolados. Ente elas, destacam-se as técnicas de aprendizagem de máquina (*Machine learning*), a aplicação de filtros e a colaboração e correlação de vários IDS [20], [46], [63].

6. Conclusão

O tema da segurança da informação das redes e, mais concretamente, dos sistemas de informação e das possíveis ameaças que este pode estar sujeito, como as intrusões, é a algum tempo alvo de uma evolução constante.

O objetivo inicial desta dissertação foi estudar e avaliar um dos principais pilares no que toca à segurança de sistemas informáticos e de informação: o sistema de deteção de intrusões. Dentro deste universo, foram adquiridas várias noções, como os vários tipos e métodos de deteção. Foi a partir dessa investigação e tomada de conhecimento mais pormenorizada do contexto, que se decidiu efetuar o estudo comparativo de duas das principais soluções de NIDS: Snort e Suricata.

Estes dois NIDS têm vindo, há já alguns anos, a protagonizar um saudável duelo no que diz respeito à deteção de intrusões em rede. O Snort já era um IDS de renome aquando do aparecimento do Suricata, que não querendo perder terreno surgiu com ideias inovadoras e promissoras no campo da deteção de intrusões. Entre elas o seu aclamado mecanismo de deteção *multi-threaded*, que possibilitaria um incremento substancial na capacidade de processamento de maiores volumes de tráfego, aliado às capacidades dos novos processadores com múltiplos núcleos.

Desta forma, o objetivo fulcral passou a ser a comparação e avaliação do desempenho geral destes dois sistemas NIDS, através de alguns cenários de tráfego disponibilizado em *datasets*, tentando aferir das potencialidades e lacunas de ambos os IDS.

6.1. Contributos

Depois de todos os testes e trabalho efetuado com estes sistemas de deteção, pode-se concluir que estamos perante duas boas soluções para o contexto em causa, cada um com os seus pontos fortes e fracos. Em geral, verificou-se uma melhor capacidade do Suricata em lidar com os vários tráfegos de rede, desde o seu conteúdo ao volume de dados do mesmo, apresentando quase sempre um consumo de recursos mais modesto que o seu concorrente, para o mesmo tráfego, sobretudo na utilização de CPU. A perda de pacotes é, também por esse motivo, muito menos significativa no Suricata face ao Snort. No que toca aos alertas gerados, nota-se uma tendência significativa do Suricata sobressair uma vez mais, detetando quase sempre todas as assinaturas espectáveis de serem assinaladas e gerando mais alertas. No entanto, deve-se realçar novamente, que o conjunto de regras disponibilizado na *framework* para ambos os NIDS foi as Emergingthreats, conjunto esse que detém uma otimização ligeiramente superior para o Suricata, algo que acabou por ser evidente em alguns testes, sobretudo nos alertas de descodificação do

tráfego, exclusivos deste IDS. De salientar, igualmente, que apesar do maior número de alertas gerados pelo Suricata, muitos destes são passíveis de serem falsos positivos ou *Grey Positives*, o que representa um dos flagelos no contexto da deteção de intrusões.

Em suma, o objetivo fulcral para esta dissertação foi atingido, conseguindo-se responder em grande parte à pergunta de partida colocada. São evidentes algumas das potencialidades do mecanismo de deteção do Suricata, sobretudo na agilidade com que processa tráfego variado com menor requisição de recursos, reduzindo a perda de pacotes e detetando um maior número de assinaturas. Por outro lado, apesar de mais modesto em algumas das características salientadas no Suricata, o Snort apresenta-se uma solução mais madura e consistente em alguns aspetos, sobretudo nalgumas assinaturas dúbias que o Suricata ainda assinala, conduzindo a possíveis falsos positivos, podendo ser um aspeto a melhorar deste NIDS.

6.2. Limitações e Trabalho Futuro

Devido a algumas mudanças no paradigma e foco de estudo, assim como alguns critérios de escolha tomados para o contexto desta investigação, acabaram por existir algumas limitações gerais da plataforma de trabalho, dos *datasets* utilizados e das ferramentas disponíveis para análise, condicionando sobretudo a aferição mais detalhada e exata dos alertas gerados, sendo este um trabalho futuro a ter em conta.

Para tal, poder-se-á utilizar mais e melhores ferramentas ou plataformas adequadas à avaliação de desempenho de IDS, como a *framework* Pytbull [64], com um conjunto variado e detalhado de ataques e testes para os IDS. A possibilidade de criar o próprio *dataset*, com ataques únicos e menor número de pacotes infetados, também se constitui como uma vantagem, tanto pela garantia da presença de tráfego atual, como pelo melhor controlo do tipo e número de alertas expectáveis, sendo essa a principal limitação encontrada no decorrer dos testes. Por outro lado, a utilização e correlação, de mais conjuntos de assinaturas assim como da própria deteção de cada IDS, poderá servir de base para novas experiências, tendo em vista a mitigação dos falsos positivos e negativos.

7. Referências

[1] V. Santos, "Sistemas de Detecção de Intrusões (IDS – Intrusion Detection Systems) usando unicamente softwares Open Source," 2010. [Online]. Available: https://seginfo.com.br/2010/06/21/sistemas-de-deteccao-de-intrusoes-ids-intrusion-detection-systems-usando-unicamente-softwares-open-source/. [Accessed: 05-Dec-2018].

[2] J. C. L. Martins, "Framework de Segurança de um Sistema de Informação," 2008.

[3] R. Shahi Shavvon *et al.*, "Increasing phylogenetic support for explosively radiating taxa: The promise of high-throughput sequencing for Oxytropis (Fabaceae)," *J. Syst. Evol.*, vol. 55, no. 4, pp. 385–404, 2017.

[4] "Suricata-vs-snort - aldeid." [Online]. Available: https://www.aldeid.com/wiki/Suricata-vs-snort. [Accessed: 08-Jan-2019].

[5] R. J. da F. M. Ferreira, "dgs.SGA-IDS," 2013.

[6] R. Antonioli, "Detecção e Tratamento de Intrusões em Plataformas Baseadas no XEN," 2008.

[7] E. F. Cruz, "Design Science Research em Sistemas de Informação," no. June 2011, 2011.

[8] A. da S. Netto and >Marco Antonio Pinheiro da Silveira, "Gestão Da Segurança Da Informação: Fatores Que Influenciam Sua Adoção Em Pequenas E Médias Empresas/Information Security Management: Factors That Influence Its Adoption in Small and Mid-Sized Businesses," *J. Inf. Syst. Technol. Manag. JISTEM*, vol. 4, no. 3, pp. 375–397, 2007.

[9] R. Von Solms and J. Van Niekerk, "From information security to cyber security," 2013.

[10] R. Sousa, "Modelo comportamental de ataques em redes informáticas," 2014.

[11] J. Barateiro and J. Borbinha, "CONCEITOS DE SISTEMAS DE INFORMAÇÃO APLICADOS A GESTÃO DE RISCOS," 2012.

[12] H. M. D. dos Santos, "Engenharia da Segurança de Sistemas de Informação," 2013.

[13] A. F. S. do Espírito Santo, "SEGURANÇA DA INFORMAÇÃO."

[14] "Segurança em Redes de Dados/Segurança da Informação - Wikiversidade." [Online]. Available: https://pt.wikiversity.org/wiki/Segurança_em_Redes_de_Dados/Segurança_da_Informação. [Accessed: 15-Dec-2018].

[15] M. A. P. Rocha, "Modelo para definição de criticidade em eventos de segurança em redes de computadores.," 2013.

[16] M. Aurélio and S. Alencar, *Fundamentos de Redes de Computadores 2010*. 2010.

[17] "11.2.1.1 Categorias de ameaças à segurança da rede." [Online]. Available: http://deptal.estgp.pt:9090/cisco/ccna1/course/module11/11.2.1.1/11.2.1.1.html. [Accessed: 26-Dec-2018].

[18] S. R. L. Nascimento, "Mecanismos de Deteção de Intrusão – OSSEC HIDS Análise e implementação numa organização," *Univ. Fernando Pessoa Mec.*, p. 75, 2017.

[19] B. P. de C. Filho and A. M. Filho, "Detecção de Intrusão em Redes de Alta Velocidade," 2000.

[20] G. Folino and P. Sabatino, "Ensemble based collaborative and distributed intrusion detection systems: A survey," 2016.

[21] OSSEC, "Home — OSSEC," *ossec.github.io*, 2017. [Online]. Available: https://www.ossec.net/. [Accessed: 19-Feb-2019].

[22] R. Wichmann, "Samhain Labs | samhain," *la-samhna.de*, 2006. [Online]. Available: https://la-samhna.de/samhain/. [Accessed: 19-Feb-2019].

[23] "PreludeLml - PRELUDE SIEM." [Online]. Available: https://www.prelude-siem.org/projects/prelude/wiki/PreludeLml. [Accessed: 19-Feb-2019].

[24] "The Sagan Log Analysis Engine | Quadrant Information Security." [Online]. Available: https://quadrantsec.com/sagan_log_analysis_engine/. [Accessed: 19-Feb-2019].

[25] "AIDE - ArchWiki." [Online]. Available: https://wiki.archlinux.org/index.php/AIDE. [Accessed: 19-Feb-2019].

[26] Martin Roesch, "Snort - Network Intrusion Detection & Prevention System," 1998. [Online]. Available: https://www.snort.org/. [Accessed: 07-Jan-2019].

[27] Suricata, "Suricata Open Source IDS / IPS / NSM engine," *https://suricata-ids.org*, 2017. [Online]. Available: https://suricata-ids.org/. [Accessed: 08-Jan-2019].

[28] "The Zeek Network Security Monitor." [Online]. Available: https://www.zeek.org/. [Accessed: 19-Feb-2019].

[29] I. Ghafir, V. Prenosil, J. Svoboda, and M. Hammoudeh, *A survey on network security monitoring systems*. 2016.

[30] "IDS - Sistema de detecção de intrusões - Pplware." [Online]. Available: https://pplware.sapo.pt/internet/ids-sistema-de-deteccao-de-intrusoes/. [Accessed: 10-Jan-2019].

[31] L. Wei, "Evaluation of Intrusion Detection Systems," 2007.

[32] “Framework (desambiguação) – Wikipédia, a enciclopédia livre.” [Online]. Available: https://pt.wikipedia.org/wiki/Framework_(desambiguação). [Accessed: 13-Jan-2019].

[33] “O que é estrutura? - Definição de WhatIs.com.” [Online]. Available: https://whatis.techtarget.com/definition/framework. [Accessed: 13-Jan-2019].

[34] “Security Onion.” [Online]. Available: https://securityonion.net/. [Accessed: 11-Jan-2019].

[35] AlienVault, “OSSIM: The Open Source SIEM | AlienVault,” *alienvault.com*, 2017. [Online]. Available: https://www.alienvault.com/products/ossim. [Accessed: 14-Jan-2019].

[36] “OSSIM - Wikipedia.” [Online]. Available: https://en.wikipedia.org/wiki/OSSIM. [Accessed: 12-Jan-2019].

[37] “Network Security Toolkit (NST 28).” [Online]. Available: http://www.networksecuritytoolkit.org/nst/index.html. [Accessed: 12-Jan-2019].

[38] “What is data set? - Definition from WhatIs.com.” [Online]. Available: https://whatis.techtarget.com/definition/data-set. [Accessed: 14-Jan-2019].

[39] MIT, “Datasets | MIT Lincoln Laboratory.” [Online]. Available: https://www.ll.mit.edu/r-d/datasets. [Accessed: 14-Jan-2019].

[40] UCI Machine Learning Repository, “KDD Cup 1999 Data.” p. 92697, 2015.

[41] M. W. Group and others, “MAWI Working Group traffic archive,” *2012-09-20]. Http:// Mawi. Wide. Ad. Jp/Mawi*, 2012. [Online]. Available: http://mawi.wide.ad.jp/mawi/. [Accessed: 14-Jan-2019].

[42] CAIDA, “CAIDA Data - Overview of datasets, monitors and reports,” 2016. [Online]. Available: http://www.caida.org/data/overview/. [Accessed: 14-Jan-2019].

[43] M. Sconzo, “SecRepo - Security Data Samples Repository,” 2017. [Online]. Available: http://www.secrepo.com/. [Accessed: 14-Jan-2019].

[44] Stratpsphere Lab, “Datasets Overview — Stratosphere IPS.” [Online]. Available: https://www.stratosphereips.org/datasets-overview/. [Accessed: 14-Jan-2019].

[45] NETRESEC, “Public PCAP files for download.” [Online]. Available: https://www.netresec.com/?page=PcapFiles. [Accessed: 14-Jan-2019].

[46] S. A. R. Shah and B. Issac, “Performance comparison of intrusion detection systems and application of machine learning to Snort system,” *Futur. Gener. Comput. Syst.*, vol. 80, no. November 2017, pp. 157–170, 2018.

[47] E. Albin and N. C. Rowe, “A Realistic Experimental Comparison of the Suricata and Snort Intrusion-Detection Systems,” 2012.

[48] A. Camelo *et al.*, "Análise Comparativa dos Módulos de IPS/IDS do Suricata e Snort."

[49] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Comput. Secur.*, vol. 45, pp. 100–123, Sep. 2014.

[50] B. et al. Visscher, "Sguil - Open Source Network Security Monitoring," 2015. [Online]. Available: https://bammv.github.io/sguil/index.html. [Accessed: 24-Jan-2019].

[51] P. Halliday, "the squertproject," 2018. [Online]. Available: http://www.squertproject.org/. [Accessed: 24-Jan-2019].

[52] "psrecord 1.1." [Online]. Available: https://pypi.org/project/psrecord/. [Accessed: 24-Jan-2019].

[53] "SYSSTAT." [Online]. Available: http://sebastien.godard.pagesperso-orange.fr/man_pidstat.html. [Accessed: 24-Jan-2019].

[54] "htop - an interactive process viewer for Unix." [Online]. Available: https://hisham.hm/htop/. [Accessed: 24-Jan-2019].

[55] "netstat (8) - Página man do Linux." [Online]. Available: https://linux.die.net/man/8/netstat. [Accessed: 24-Jan-2019].

[56] Klassen Fred and AppNeta, "Tcpreplay - Pcap editing and replaying utilities." [Online]. Available: https://tcpreplay.appneta.com/. [Accessed: 24-Jan-2019].

[57] "Snort engine - assign multiple cores improve performance or not - Grupos do Google." [Online]. Available: https://groups.google.com/forum/#!msg/security-onion/QmR0qNzYFrg/BMZ1wckiDeMJ. [Accessed: 20-Feb-2019].

[58] "Errata Security: Multi-core scaling: it's not multi-threaded." [Online]. Available: https://blog.erratasec.com/2013/02/multi-core-scaling-its-not-multi.html#.XG0quOj7TIW. [Accessed: 20-Feb-2019].

[59] "Understanding %CPU while running top command - Unix & Linux Stack Exchange." [Online]. Available: https://unix.stackexchange.com/questions/145247/understanding-cpu-while-running-top-command. [Accessed: 20-Feb-2019].

[60] E. Albin, "A COMPARATIVE ANALYSIS OF THE SNORT AND SURICATA INTRUSION-DETECTION SYSTEMS," 2011.

[61] "Snort and suricata don't give the same amount of alerts - Grupos do Google." [Online]. Available: https://groups.google.com/forum/#!topic/security-onion/elYHcZqDNY8. [Accessed: 20-Feb-2019].

[62] "suricata-rules." [Online]. Available: https://github.com/jflsakfja/suricata-

rules/blob/master/list.txt. [Accessed: 24-Jan-2019].

[63] G. P. Spathoulas and S. K. Katsikas, "Reducing false positives in intrusion detection systems," *Comput. Secur.*, vol. 29, pp. 35–44, 2009.

[64] "pytbull - IDS/IPS Testing Framework - home." [Online]. Available: http://pytbull.sourceforge.net/. [Accessed: 29-Jan-2019].

[65] "Information security." [Online]. Available: https://en.wikipedia.org/wiki/Information_security. [Accessed: 20-Feb-2019].

[66] "Snort vs Suricata – Tactical FLEX, Inc." [Online]. Available: https://tacticalflex.zendesk.com/hc/en-us/articles/360010678893-Snort-vs-Suricata. [Accessed: 08-Jan-2019].

8. Anexos

8.1. Anexo 1 – Tráfego Benigno

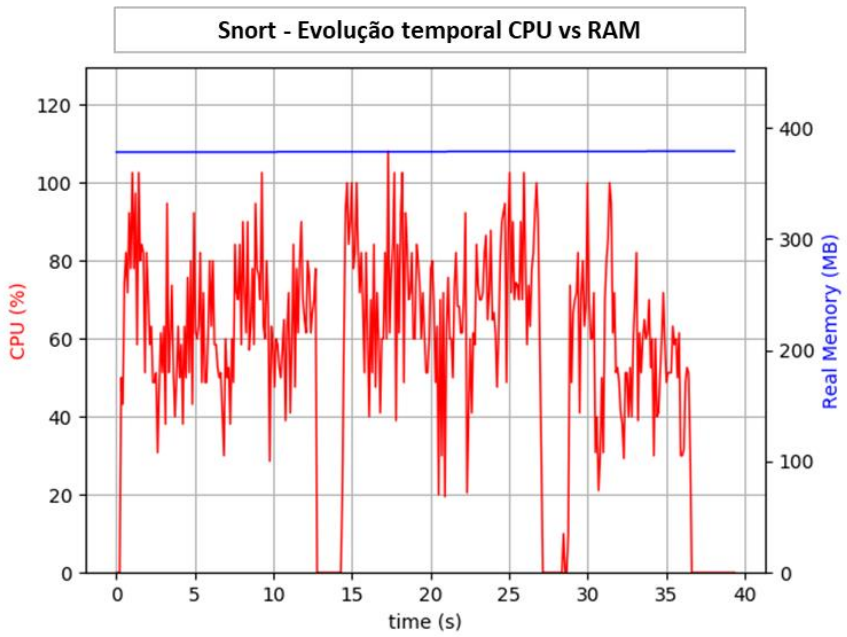
8.1.1. Snort

- **Estatísticas de Transmissão do Pcap e da Interface de Rede**

```
Actual: 471237 packets (409655375 bytes) sent in 36.34 seconds.         Rated: 11272850.0 bps, 86.01 Mbps, 12967.45 pps
Statistics for network device: eth0
       Attempted packets:         471237
       Successful packets:        471237
       Failed packets:            0
       Retried packets (ENOBUFS): 0
       Retried packets (EAGAIN):  0
```

```
Kernel Interface table
eth0      Link encap:Ethernet  HWaddr 52:54:00:f7:21:b7
          inet addr:192.168.122.25  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fef7:21b7/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:805 errors:0 dropped:0 overruns:0 frame:0
          TX packets:471829 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:285719 (285.7 KB)  TX bytes:409726812 (409.7 MB)
```

- **Recursos de Hardware Utilizados**



```
Average:      UID       PID    %usr %system  %guest    %CPU   CPU  Command
Average:     1001      4280   53,51    0,38    0,00   53,90     -  snort

Average:      UID       PID  minflt/s  majflt/s     VSZ    RSS   %MEM  Command
Average:     1001      4280      4,90      0,00  743256 387665   2,76  snort

Average:      UID       PID   kB_rd/s   kB_wr/s kB_ccwr/s  Command
Average:     1001      4280     -1,00     -1,00     -1,00  snort
```

- **Assinaturas Detetadas**

QUEUE	SC	DC	ACTIVITY	LAST EVENT	SIGNATURE
1	1	1		21:48:53	ET P2P BitTorrent DHT announce_peers request
1	1	1		21:48:46	ET P2P BitTorrent DHT nodes reply
1	1	1		21:48:43	ET P2P Vuze BT UDP Connection (5)
1	1	1		21:48:42	ET P2P BitTorrent DHT ping request

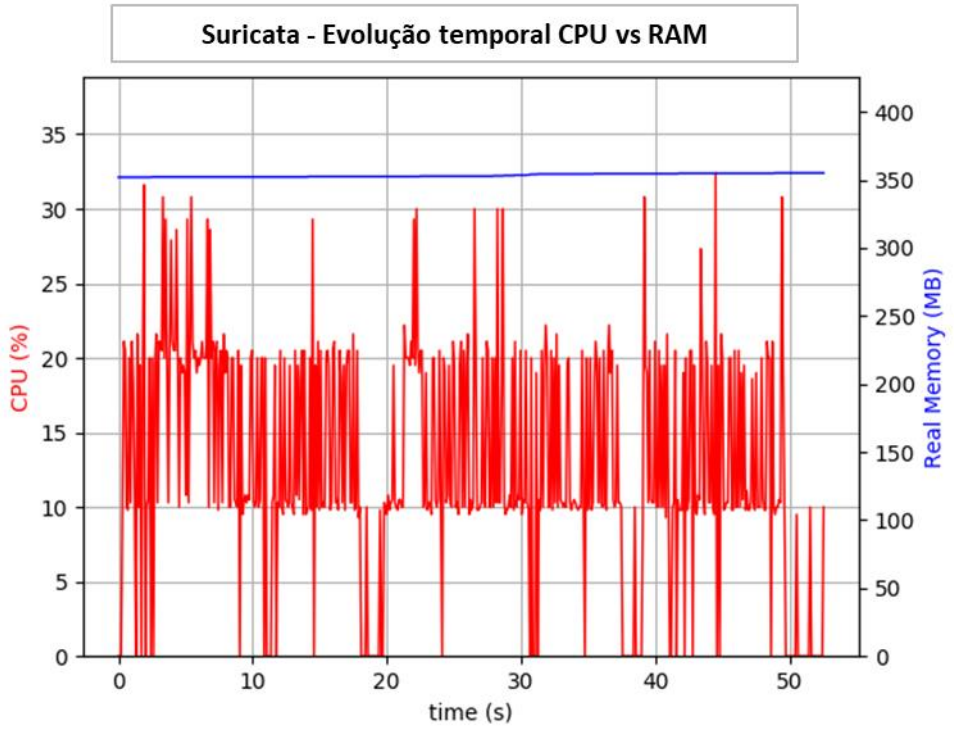
8.1.2. Suricata

- **Estatísticas de Transmissão do Pcap e da Interface de Rede**

```
Actual: 471237 packets (409655375 bytes) sent in 49.45 seconds.         Rated: 8284234.0 bps, 63.20 Mbps, 9529.57 pps
Statistics for network device: eth0
       Attempted packets:         471237
       Successful packets:        471237
       Failed packets:            0
       Retried packets (ENOBUFS): 0
       Retried packets (EAGAIN):  0
```

```
Kernel Interface table
eth0      Link encap:Ethernet  HWaddr 52:54:00:f7:21:b7
          inet addr:192.168.122.25  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fef7:21b7/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:566 errors:0 dropped:0 overruns:0 frame:0
          TX packets:471703 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:258994 (258.9 KB)  TX bytes:409714294 (409.7 MB)
```

- **Recursos de Hardware Utilizados**




```
Average:      UID       PID    %usr %system  %guest    %CPU   CPU  Command
Average:     1001      5032   11,96    0,67    0,00   12,63     -  Suricata-Main

Average:      UID       PID  minflt/s  majflt/s     VSZ    RSS   %MEM  Command
Average:     1001      5032     14,98      0,00  724692 361791   2,58  Suricata-Main

Average:      UID       PID   kB_rd/s   kB_wr/s kB_ccwr/s  Command
Average:     1001      5032     -1,00     -1,00     -1,00  Suricata-Main
```

- **Assinaturas Detetadas**

QUEUE	SC	DC	ACTIVITY	LAST EVENT	SIGNATURE
4	1	4		01:44:13	ET P2P BitTorrent peer sync
1	1	1		01:44:56	ET DROP Spamhaus DROP Listed Traffic Inbound group 15
1	1	1		01:44:29	ET P2P BitTorrent DHT announce_peers request
1	1	1		01:44:19	ET P2P BitTorrent DHT nodes reply
1	1	1		01:44:15	ET P2P Vuze BT UDP Connection (5)
1	1	1		01:44:14	ET P2P BitTorrent DHT ping request

8.2. Anexo 2 – Pcap maior com grande volume de tráfego parcialmente malicioso

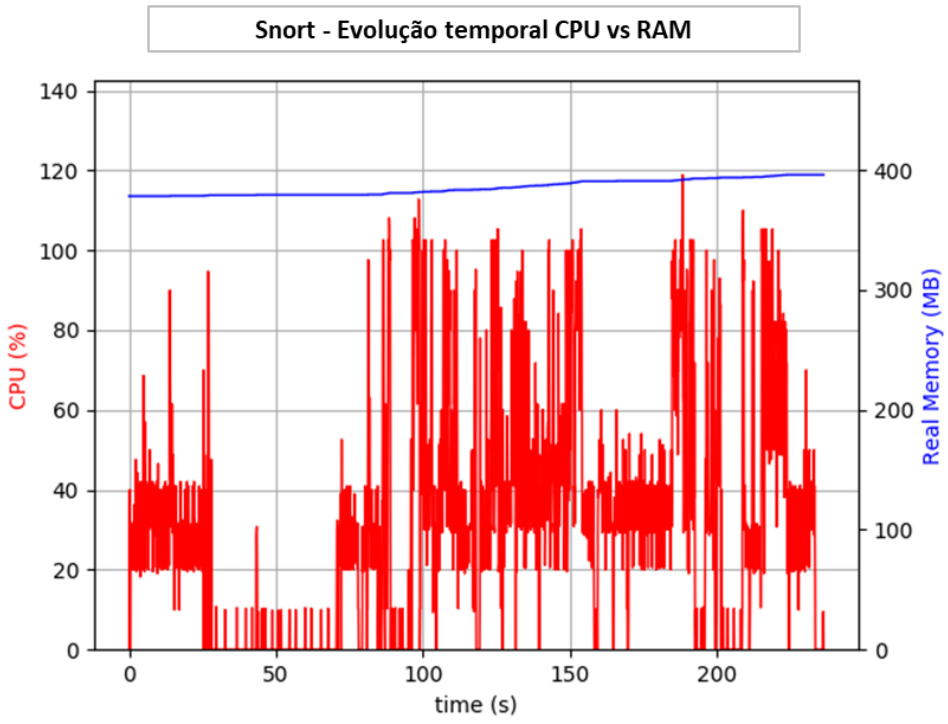
8.2.1. Snort

- **Estatísticas de Transmissão do Pcap e da Interface de Rede**

```
Actual: 1437980 packets (870710721 bytes) sent in 233.17 seconds.              Rated: 3734231.2 bps, 28.49 Mbps, 6167.09 pps
Statistics for network device: eth0
        Attempted packets:         1437980
        Successful packets:        1437980
        Failed packets:            0
        Retried packets (ENOBUFS): 0
        Retried packets (EAGAIN):  0
```

```
Kernel Interface table
eth0      Link encap:Ethernet  HWaddr 52:54:00:f7:21:b7
          inet addr:192.168.122.25  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fef7:21b7/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:893 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1438731 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:314836 (314.8 KB)  TX bytes:870814620 (870.8 MB)
```

- **Recursos de Hardware Utilizados**



```
Average:      UID       PID    %usr %system  %guest    %CPU   CPU  Command
Average:     1001      4322   28,90    1,01    0,00   29,91     -  snort

Average:      UID       PID  minflt/s  majflt/s     VSZ    RSS   %MEM  Command
Average:     1001      4322     19,41      0,00  743148 394543   2,81  snort

Average:      UID       PID   kB_rd/s   kB_wr/s kB_ccwr/s  Command
Average:     1001      4322     -1,00     -1,00     -1,00  snort
```

- **Assinaturas Detetadas**

QUEUE	SC	DC	ACTIVITY	LAST EVENT	SIGNATURE
49	4	1		22:57:48	ET POLICY PE EXE or DLL Windows file download HTTP
48	2	1		22:56:11	ET POLICY Executable served from Amazon S3
27	1	4		22:57:36	ET POLICY User-Agent (NSIS_Inetc (Mozilla)) - Sometimes used by hostile installers
22	1	4		22:58:06	ET CHAT Skype User-Agent detected
18	1	2		22:58:06	ET CHAT Skype VOIP Checking Version (Startup)
15	15	1		22:56:48	ET POLICY Dropbox.com Offsite File Backup in Use
12	1	1		22:56:26	ET INFO - Applet Tag In Edwards Packed JavaScript
8	8	1		22:57:45	ET POLICY SSLv3 outbound connection from client vulnerable to POODLE attack
5	1	3		22:57:52	ET POLICY Outdated Flash Version M2
4	1	1		22:54:16	ET POLICY Vulnerable Java Version 1.8.x Detected
1	1	1		22:56:20	ET POLICY Outdated Flash Version M1
1	1	1		22:54:59	ET POLICY Dropbox Client Broadcasting

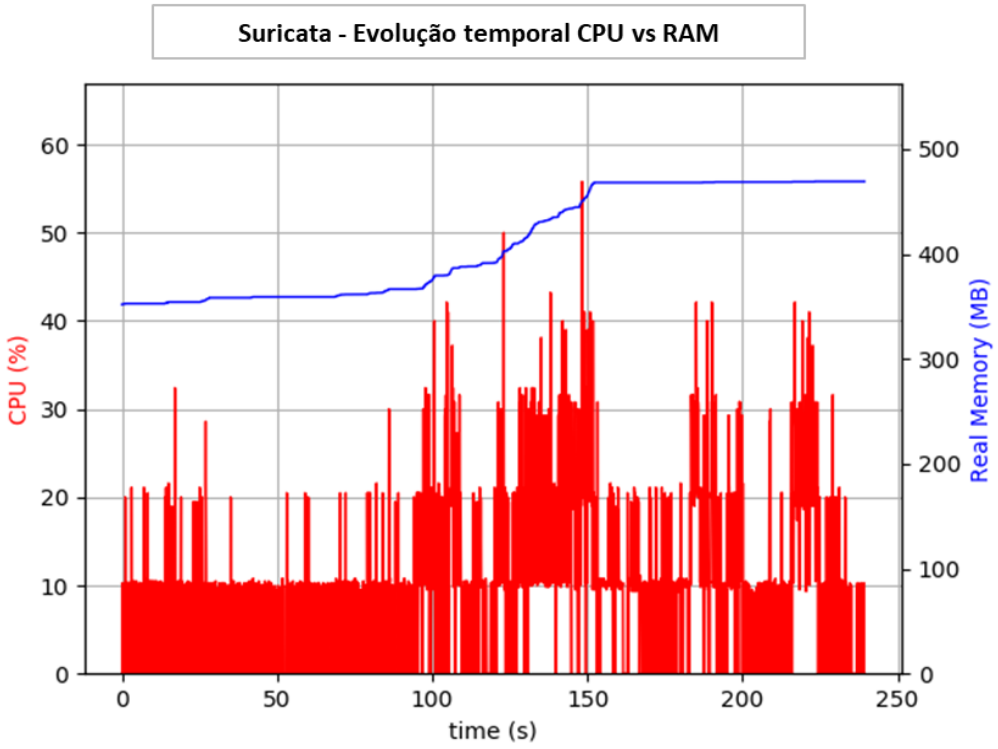
8.2.2. Suricata

- **Estatísticas de Transmissão do Pcap e da Interface de Rede**

```
Actual: 1437980 packets (870710721 bytes) sent in 233.63 seconds.              Rated: 3726879.0 bps, 28.43 Mbps, 6154.95 pps
Statistics for network device: eth0
       Attempted packets:         1437980
       Successful packets:        1437980
       Failed packets:            0
       Retried packets (ENOBUFS): 0
       Retried packets (EAGAIN):  0
```

```
Kernel Interface table
eth0      Link encap:Ethernet  HWaddr 52:54:00:f7:21:b7
          inet addr:192.168.122.25  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fef7:21b7/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:767 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1438571 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:259308 (259.3 KB)  TX bytes:870781475 (870.7 MB)
```

- **Recursos de Hardware Utilizados**




```
Average:      UID       PID    %usr %system  %guest    %CPU   CPU  Command
Average:     1001      4683    8,24    0,48    0,00    8,72     -  Suricata-Main

Average:      UID       PID  minflt/s  majflt/s     VSZ    RSS   %MEM  Command
Average:     1001      4683    128,51      0,00  755948 420365   3,00  Suricata-Main

Average:      UID       PID   kB_rd/s   kB_wr/s kB_ccwr/s  Command
Average:     1001      4683     -1,00     -1,00     -1,00  Suricata-Main
```

- **Assinaturas Detetadas**

QUEUE	SC	DC	ACTIVITY	LAST EVENT	SIGNATURE
1679	5	1		01:32:15	SURICATA HTTP unable to match response to request
42	15	1		01:31:00	ET POLICY Dropbox.com Offsite File Backup in Use
36	4	1		01:32:03	ET POLICY PE EXE or DLL Windows file download HTTP
27	1	4		01:31:50	ET POLICY User-Agent (NSIS_Inetc (Mozilla)) - Sometimes used by hostile installers
22	1	4		01:32:21	ET CHAT Skype User-Agent detected
18	1	2		01:32:21	ET CHAT Skype VOIP Checking Version (Startup)
16	2	1		01:30:24	ET POLICY Executable served from Amazon S3
12	1	1		01:30:39	ET INFO - Applet Tag In Edwards Packed JavaScript
3	1	3		01:31:53	ET POLICY Outdated Flash Version M2
1	1	1		01:30:33	ET POLICY Outdated Flash Version M1
1	1	1		01:29:12	ET POLICY Dropbox Client Broadcasting

8.3. Anexo 3 – Ataque IRC DoS com Pacotes UDP

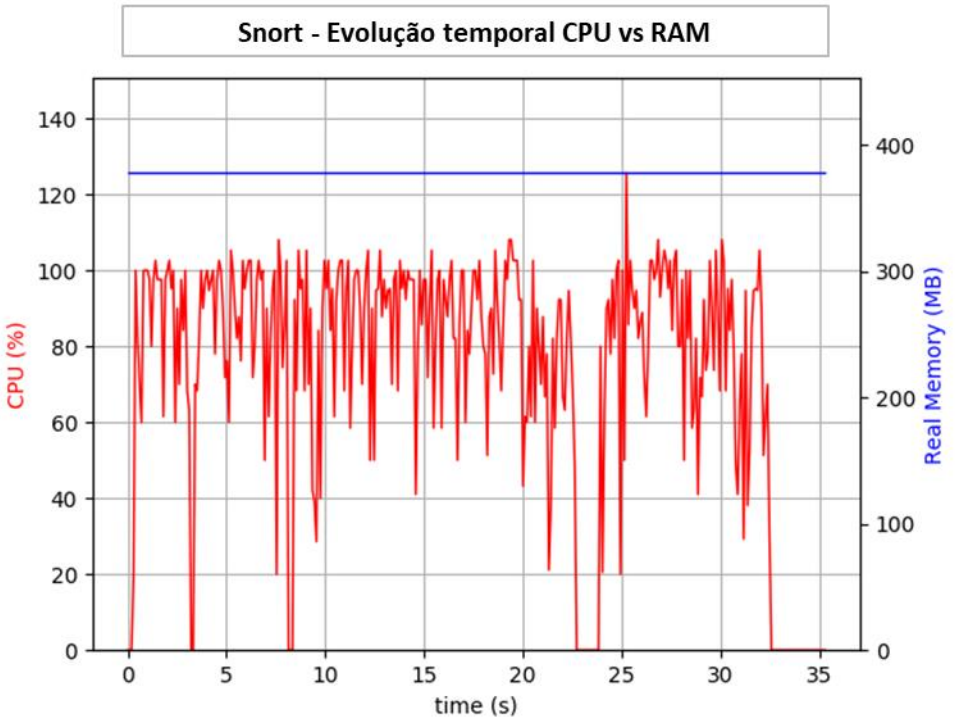
8.3.1. Snort

- **Estatísticas de Transmissão do Pcap e da Interface de Rede**

```
Actual: 256712 packets (218271666 bytes) sent in 32.22 seconds.        Rated: 6774415.5 bps, 51.68 Mbps, 7967.47 pps
Statistics for network device: eth0
       Attempted packets:         256712
       Successful packets:        256709
       Failed packets:            3
       Retried packets (ENOBUFS): 0
       Retried packets (EAGAIN):  0
```

```
Kernel Interface table
eth0      Link encap:Ethernet  HWaddr 52:54:00:f7:21:b7
          inet addr:192.168.122.25  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fef7:21b7/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:546 errors:0 dropped:0 overruns:0 frame:0
          TX packets:257185 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:247897 (247.8 KB)  TX bytes:218320476 (218.3 MB)
```

- **Recursos de Hardware Utilizados**




```
Average:      UID       PID    %usr %system  %guest    %CPU   CPU  Command
Average:     1001      5178   71,54    1,37    0,00   72,91     -  snort

Average:      UID       PID  minflt/s  majflt/s     VSZ    RSS   %MEM  Command
Average:     1001      5178      1,11      0,00  743220 387284   2,76  snort

Average:      UID       PID   kB_rd/s   kB_wr/s kB_ccwr/s  Command
Average:     1001      5178     -1,00     -1,00     -1,00  snort
```

- **Assinaturas Detetadas**

QUEUE	SC	DC	ACTIVITY	LAST EVENT	SIGNATURE
35	1	2		19:47:00	ET CHAT IRC PRIVMSG command
23	2	1		19:47:00	ET CHAT IRC PING command
18	1	2		19:47:00	ET CHAT IRC PONG response
2	1	2		19:47:00	ET CHAT IRC NICK command
2	1	2		19:47:00	ET CHAT IRC JOIN command
2	1	2		19:47:00	ET CHAT IRC USER Likely bot with 0 0 colon checkin

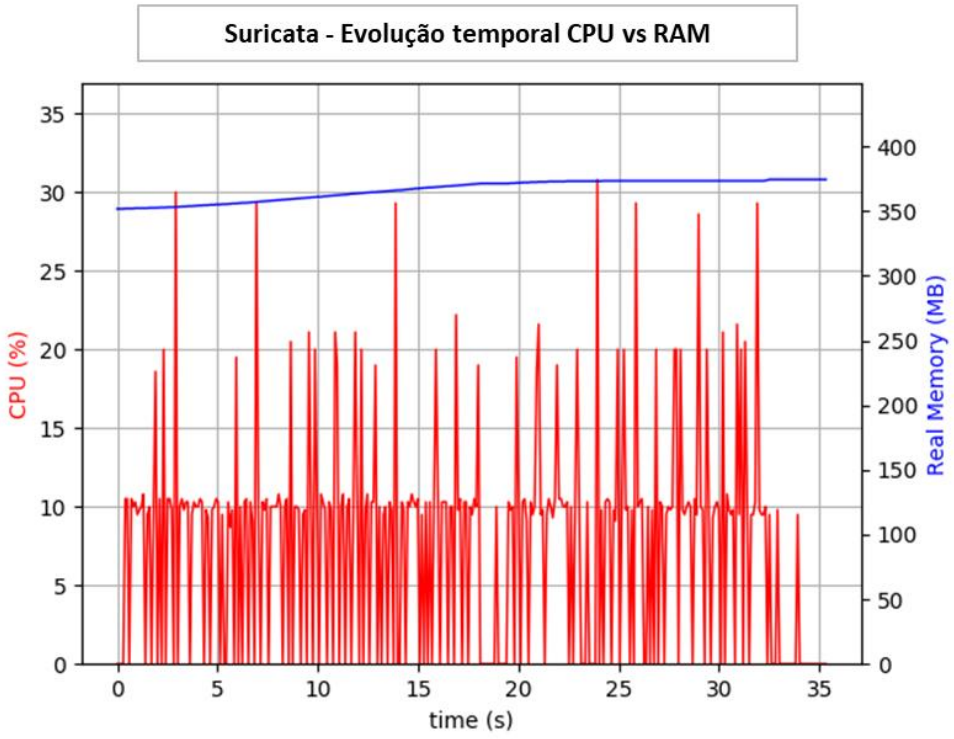
8.3.2. Suricata

- **Estatísticas de Transmissão do Pcap e da Interface de Rede**

```
Actual: 256712 packets (218271666 bytes) sent in 32.32 seconds.         Rated: 6753455.0 bps, 51.52 Mbps, 7942.82 pps
Statistics for network device: eth0
        Attempted packets:         256712
        Successful packets:        256709
        Failed packets:            3
        Retried packets (ENOBUFS): 0
        Retried packets (EAGAIN):  0
```

```
Kernel Interface table
eth0      Link encap:Ethernet  HWaddr 52:54:00:f7:21:b7
          inet addr:192.168.122.25  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fef7:21b7/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:602 errors:0 dropped:0 overruns:0 frame:0
          TX packets:257217 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:244187 (244.1 KB)  TX bytes:218323390 (218.3 MB)
```

- **Recursos de Hardware Utilizados**




```
Average:      UID       PID    %usr %system  %guest    %CPU   CPU  Command
Average:     1001      5775    6,97    0,66    0,00    7,63     -  Suricata-Main

Average:      UID       PID  minflt/s  majflt/s     VSZ    RSS   %MEM  Command
Average:     1001      5775    163,70      0,00  724692 375548   2,68  Suricata-Main

Average:      UID       PID   kB_rd/s   kB_wr/s kB_ccwr/s  Command
Average:     1001      5775     -1,00     -1,00     -1,00  Suricata-Main
```

- **Assinaturas Detetadas**

QUEUE	SC	DC	ACTIVITY	LAST EVENT	SIGNATURE
8891	1	1		00:11:54	SURICATA FRAG IPv4 Fragmentation overlap
4333	1	1		00:11:54	SURICATA UDP packet too small
4094	1	1		00:11:54	SURICATA UDP invalid header length
102	1	2		00:12:43	ET CHAT IRC PONG response
100	1	2		00:11:54	ET CHAT IRC USER command
100	1	2		00:11:54	ET CHAT IRC NICK command
100	1	2		00:11:54	ET CHAT IRC JOIN command
100	1	2		00:11:54	ET CHAT IRC PRIVMSG command
100	1	2		00:11:54	ET CHAT IRC USER Likely bot with 0 0 colon checkin
26	1	1		00:12:43	ET CHAT IRC PING command

8.4. Anexo 4 – Ataque IRC DoS com Pacotes ICMP

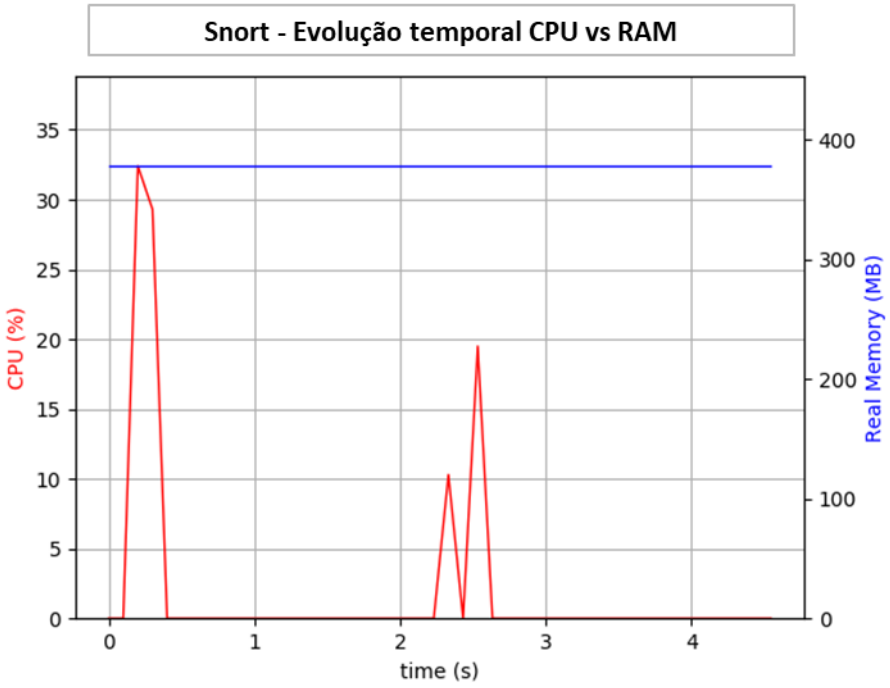
8.4.1. Snort

- **Estatísticas de Transmissão do Pcap e da Interface de Rede**

```
Actual: 28826 packets (30211468 bytes) sent in 2.40 seconds.          Rated: 12588112.0 bps, 96.04 Mbps, 12010.83 pps
Statistics for network device: eth0
        Attempted packets:         28826
        Successful packets:        28824
        Failed packets:            2
        Retried packets (ENOBUFS): 0
        Retried packets (EAGAIN):  0
```

```
Kernel Interface table
eth0      Link encap:Ethernet  HWaddr 52:54:00:f7:21:b7
          inet addr:192.168.122.25  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fef7:21b7/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:457 errors:0 dropped:0 overruns:0 frame:0
          TX packets:29245 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:231393 (231.3 KB)  TX bytes:30256064 (30.2 MB)
```

- **Recursos de Hardware Utilizados**




```
Average:      UID       PID    %usr %system  %guest    %CPU   CPU  Command
Average:     1001      5942    2,25    0,00    0,00    2,25     -  snort

Average:      UID       PID  minflt/s  majflt/s     VSZ    RSS   %MEM  Command
Average:     1001      5942     10,50      0,00  743068 387172   2,76  snort

Average:      UID       PID   kB_rd/s   kB_wr/s kB_ccwr/s  Command
Average:     1001      5942     -1,00     -1,00     -1,00  snort
```

- **Assinaturas Detetadas**

QUEUE	SC	DC	ACTIVITY	LAST EVENT	SIGNATURE
17	1	1		17:58:45	ET CHAT IRC PRIVMSG command
9	2	1		17:58:47	ET CHAT IRC PING command
2	1	2		17:58:47	ET CHAT IRC NICK command
2	1	2		17:58:47	ET CHAT IRC JOIN command
2	1	1		17:58:47	ET CHAT IRC PONG response
2	1	2		17:58:47	ET CHAT IRC USER Likely bot with 0 0 colon checkin

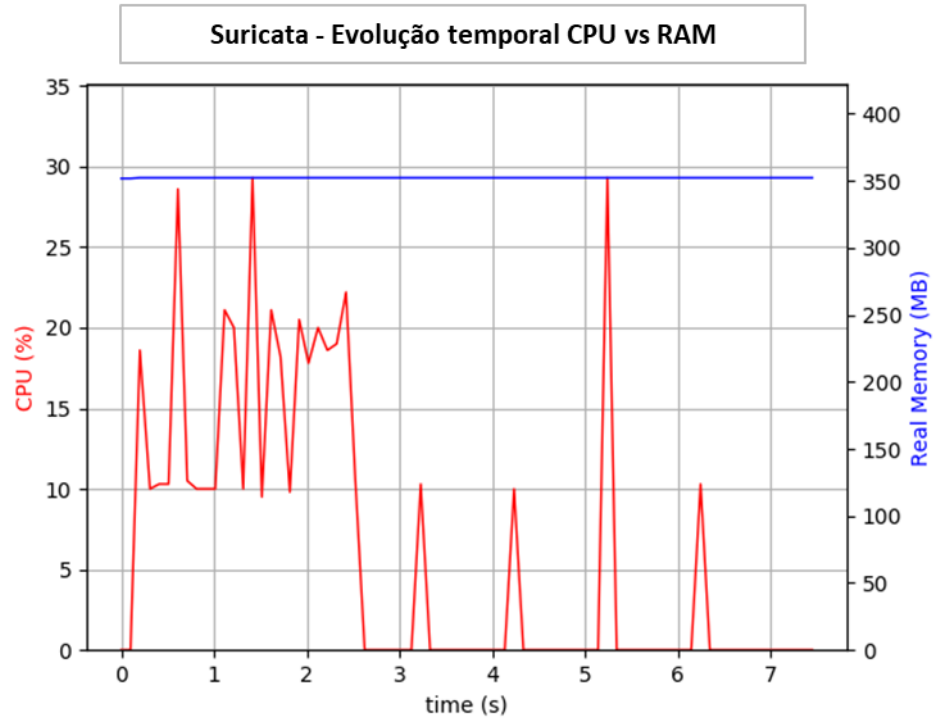
8.4.2. Suricata

- **Estatísticas de Transmissão do Pcap e da Interface de Rede**

```
Actual: 28826 packets (30211468 bytes) sent in 2.42 seconds.          Rated: 12484078.0 bps, 95.25 Mbps, 11911.57 pps
Statistics for network device: eth0
        Attempted packets:         28826
        Successful packets:        28824
        Failed packets:            2
        Retried packets (ENOBUFS): 0
        Retried packets (EAGAIN):  0
```

```
Kernel Interface table
eth0      Link encap:Ethernet  HWaddr 52:54:00:f7:21:b7
          inet addr:192.168.122.25  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fef7:21b7/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:277 errors:0 dropped:0 overruns:0 frame:0
          TX packets:29027 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:25391 (25.3 KB)  TX bytes:30224926 (30.2 MB)
```

- **Recursos de Hardware Utilizados**




```
Average:      UID       PID    %usr %system  %guest    %CPU   CPU  Command
Average:     1001      5009    4,86    1,57    0,00    6,43     -  Suricata-Main

Average:      UID       PID  minflt/s  majflt/s     VSZ    RSS   %MEM  Command
Average:     1001      5009      5,71      0,00  724692 360636   2,57  Suricata-Main

Average:      UID       PID   kB_rd/s   kB_wr/s kB_ccwr/s  Command
Average:     1001      5009     -1,00     -1,00     -1,00  Suricata-Main
```

- **Assinaturas Detetadas**

QUEUE	SC	DC	ACTIVITY	LAST EVENT	SIGNATURE
26777	1	1		23:30:55	SURICATA ICMPv4 unknown type
1484	1	1		23:30:55	SURICATA ICMPv4 unknown code
14	1	1		23:30:55	SURICATA ICMPv4 unknown version
2	1	1		23:30:55	ET CHAT IRC PONG response

8.5. Anexo 5 – Ataque IRC Port Scan

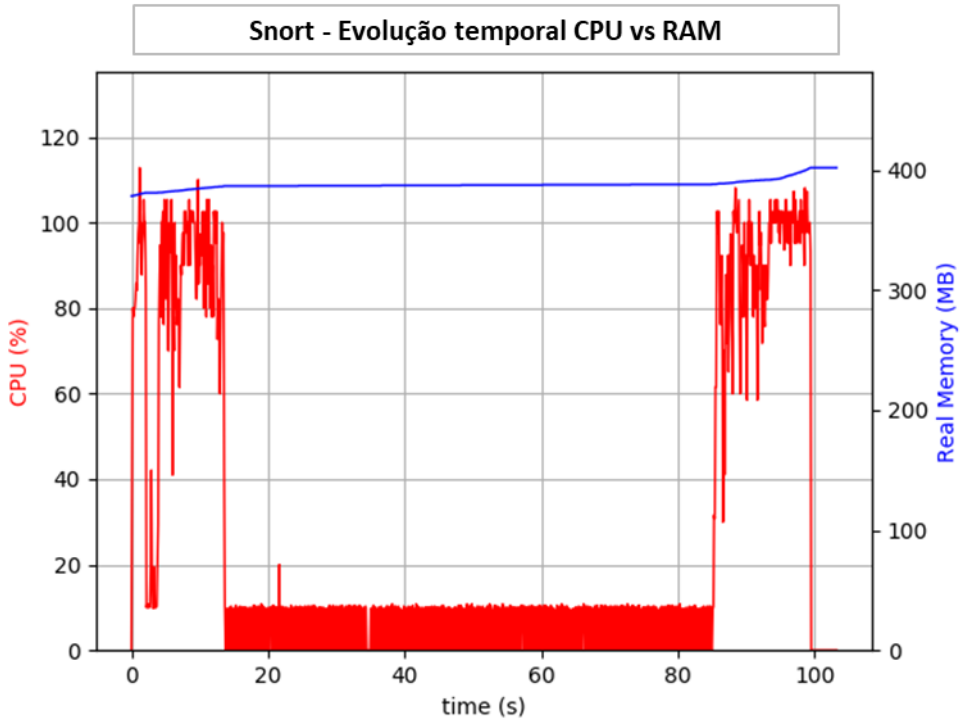
8.5.1. Snort

- **Estatísticas de Transmissão do Pcap e da Interface de Rede**

```
Actual: 495056 packets (120654271 bytes) sent in 93.64 seconds.        Rated: 1288490.8 bps, 9.83 Mbps, 5286.80 pps
Statistics for network device: eth0
       Attempted packets:         495056
       Successful packets:        495046
       Failed packets:            10
       Retried packets (ENOBUFS): 0
       Retried packets (EAGAIN):  0
```

```
Kernel Interface table
eth0      Link encap:Ethernet  HWaddr 52:54:00:f7:21:b7
          inet addr:192.168.122.25  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fef7:21b7/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:306 errors:0 dropped:0 overruns:0 frame:0
          TX packets:495265 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:28510 (28.5 KB)  TX bytes:120639430 (120.6 MB)
```

- **Recursos de Hardware Utilizados**




```
Average:      UID       PID    %usr %system  %guest    %CPU   CPU  Command
Average:     1001      5707   25,26    0,85    0,00   26,11     -  snort

Average:      UID       PID  minflt/s  majflt/s     VSZ    RSS   %MEM  Command
Average:     1001      5707     58,27      0,00  743236 397215   2,83  snort

Average:      UID       PID   kB_rd/s   kB_wr/s kB_ccwr/s  Command
Average:     1001      5707     -1,00     -1,00     -1,00  snort
```

- **Assinaturas Detetadas**

QUEUE	SC	DC	ACTIVITY	LAST EVENT	SIGNATURE
1436	1	1		20:50:50	ET CHAT IRC PONG response
1408	1	1		20:50:50	ET CHAT IRC PING command
500	1	3		20:50:50	ET CHAT IRC PRIVMSG command
40	1	1		20:49:17	ET CURRENT_EVENTS SUSPICIOUS IRC - PRIVMSG *.(exe|tar|tgz|zip) download command
3	1	3		20:49:18	ET CHAT IRC NICK command
3	1	3		20:49:18	ET CHAT IRC JOIN command
3	1	3		20:49:18	ET CHAT IRC USER Likely bot with 0 0 colon checkin

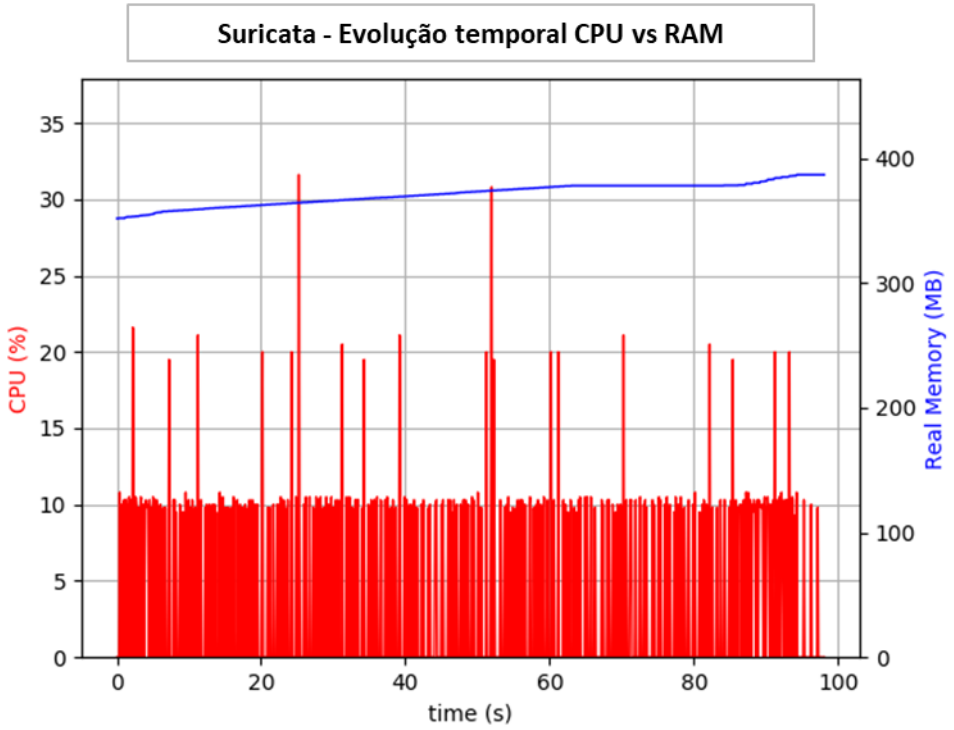
8.5.2. Suricata

- **Estatísticas de Transmissão do Pcap e da Interface de Rede**

```
Actual: 495056 packets (120654271 bytes) sent in 93.84 seconds.         Rated: 1285744.6 bps, 9.81 Mbps, 5275.53 pps
Statistics for network device: eth0
        Attempted packets:         495056
        Successful packets:        495046
        Failed packets:            10
        Retried packets (ENOBUFS): 0
        Retried packets (EAGAIN):  0
```

```
Kernel Interface table
eth0      Link encap:Ethernet  HWaddr 52:54:00:f7:21:b7
          inet addr:192.168.122.25  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fef7:21b7/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:636 errors:0 dropped:0 overruns:0 frame:0
          TX packets:495549 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:244732 (244.7 KB)  TX bytes:120675686 (120.6 MB)
```

- **Recursos de Hardware Utilizados**




```
Average:      UID       PID    %usr %system  %guest    %CPU   CPU  Command
Average:     1001      5271    2,94    0,36    0,00    3,30     -  Suricata-Main

Average:      UID       PID  minflt/s  majflt/s      VSZ    RSS   %MEM  Command
Average:     1001      5271     91,05      0,00   724688 380216   2,71  Suricata-Main

Average:      UID       PID   kB_rd/s   kB_wr/s kB_ccwr/s  Command
Average:     1001      5271     -1,00     -1,00     -1,00  Suricata-Main
```

- **Assinaturas Detetadas**

QUEUE	SC	DC	ACTIVITY	LAST EVENT	SIGNATURE
6586	1	1		00:53:59	ET CHAT IRC PONG response
2694	1	3		00:53:59	ET CHAT IRC PRIVMSG command
1657	1	1		00:53:56	ET CHAT IRC PING command
162	1	3		00:52:27	ET CHAT IRC USER command
162	1	3		00:52:27	ET CHAT IRC NICK command
162	1	3		00:52:27	ET CHAT IRC JOIN command
162	1	3		00:52:27	ET CHAT IRC USER Likely bot with 0 0 colon checkin
40	1	1		00:52:26	ET CURRENT_EVENTS SUSPICIOUS IRC - PRIVMSG *.(exe|tar|tgz|zip) download command