UNIVERSIDADE FEDERAL DA PARAÍBA

CENTRO DE CIÊNCIAS APLICADAS A EDUCAÇÃO

DEPARTAMENTO DE CIÊNCIAS EXATAS

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Ferramenta de Medição e Identificação de Ataques (D)DoS

KELSON VICTOR PRAXEDES DE ALMEIDA

Orientador: Prof. DSc. Carlos Dias

RIO TINTO - PB

2015

KELSON VICTOR PRAXEDES DE ALMEIDA

Ferramenta de Medição e Identificação de Ataques (D)DoS

Monografia apresentada para obtenção do título de Bacharel à banca examinadora no Curso de Bacharelado em Sistemas de Informação do Centro de Ciências Aplicadas e Educação (CCAE), Campus IV da Universidade Federal da Paraíba. Orientador: Prof. Me. Carlos Dias.

RIO TINTO - PB
2015
II

Ficha catalográfica preparada pela Seção de Catalogação e Classificação da Biblioteca da UFPB

A447f Almeida, Kelson Victor Praxedes de.

Ferramenta de medição e identificação de ataques (D)DoS. / Kelson Victor Praxedes de Almeida. – Rio Tinto: [s.n.], 2015.

49 f.: il. –

Orientador: Prof. DSc. Carlos Dias.
Monografia (Graduação) – UFPB/CCAE.

1. Software - desenvolvimento. 2. Segurança digital. 3. Internet.

KELSON VICTOR PRAXEDES DE ALMEIDA

Ferramenta de Medição e Identificação de Ataques (D)DoS

Trabalho de Conclusão de Curso submetido ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal da Paraíba, Campus IV, como parte dos requisitos necessários para obtenção do grau de BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Assinatura do autor:_____

APROVADO POR:

_____

Orientador: Prof. Me. Carlos Dias
Universidade Federal da Paraíba – Campus IV

_____

Prof. Me Rafael Magalhães
Universidade Federal da Paraíba – Campus IV

_____

Prof. Me Marcus Carvalho
Universidade Federal da Paraíba – Campus IV

RIO TINTO - PB
2015
IV

“O sacrifício é o intervalo entre o seu objetivo e a sua glória.”

AGRADECIMENTOS

A Deus e ao Espírito Santo por sempre estarem comigo, guiando meus passos e caminhos durante toda a minha vida.

A minha mãe Fátima Praxedes, ao meu pai Kelson Virgilio e a minha irmã Priscilla Praxedes pelo suporte, apoio e amor incondicional que sempre demonstraram e nunca deixaram faltar e que são da mais extrema importância para minha formação como pessoa, estudante e profissional.

A minha noiva, amiga, futura esposa e companheira Erica Renally, que está presente comigo desde o início da minha graduação e com sua sabedoria e conhecimentos que me encantaram e encantam até hoje, sendo peça fundamental para que o objetivo da minha formação fosse alcançado.

Ao meu professor-orientador Carlos Hacks, que com o seu vasto conhecimento me auxiliou e me orientou com muita paciência e dedicação em todas as etapas do trabalho.

Aos amigos com os quais compartilhei e adquiri conhecimentos e amizade durante boa parte da minha graduação dividindo moradia em Rio Tinto: Danilo Formiga, Digenaldo Neto, Pablo Lima, Raphael Diniz, Rennan Felizardo, Thiago Oliveira, Lucas Cantarelli, José Paulo, Rafael Farias, entre outros companheiros de jornada.

Aos meus amigos e supervisores de estágio realizado no Canadá: Dr. Ali Ghorbani, Dra. Natalia Stakhanova, Hugo Gonzalez, Jesse English, Hossein Hadian Jazi, Andi Fitriah e Elaheh Biglar Beigi Samani. Que foram de fundamental importância para a familiaridade e escolha do tema desse trabalho.

A todos outros familiares e amigos, não mencionados, porém que sempre me apoiaram e, em especial, ao meu primo-irmão, Paulo César, por sempre demonstrar amizade e por ter sempre me motivado durante a realização desse estudo.

RESUMO

Os ataques cibernéticos estão cada vez mais assombrando o mundo da tecnologia, ataques e/ou ações que comprometam a integridade de dados, que realizem roubo de identidade, fraude com operações bancárias e até mesmo causem indisponibilidade de sites e serviços precisam de técnicas e ferramentas que ajudem a mitigar o máximo possível esses problemas. Pensando nisso esse trabalho propõe o desenvolvimento da ferramenta *Denial Capture*, que análise o impacto e identifica um dos mais poderosos tipos de ataque existente na internet, o Denial-of-service ou ataque de negação de serviço. Um conjunto de técnicas teóricas são aplicadas para prover a escolha das melhores métricas e regras com possam identificar e mensurar esse tipo de ataque. Após esses embasamento teórico é possível apresentar os passos da implementação e utilização dessa ferramenta, além de exemplificar de forma clara como os ataques e medições de testes são utilizados para o alcance do objetivo. Com isso esse trabalho será de contribuição para profissionais da área de segurança da informação, administradores de redes e desenvolvedores de software que procuram técnicas de utilização e desenvolvimento para medir e identificar esse tipo de ataque com as métricas propostas no estudo.

Palavras-Chave: Denial Capture, Ferramenta, Negação de Serviço, Ataque, Métricas.

ABSTRACT

Cyber attacks are increasingly haunting the world of technology attacks and/or actions that compromise the integrity of data, carrying out identity theft, fraud of banking operations and even cause unavailability of sites and services need techniques and tools to help mitigate these problems as much as possible. Thinking about this, this work proposes the development of the *Denial Capture* tool with the goal to analyze the impact and identifies the most powerful types of existing attack on the Internet, the Denial-of-service. A set of theoretical techniques are applied to provide the best choice of metrics and rules to identify and measure this type of attack. After these theoretical bases is possible to present the steps on the implementation and the use of this tool, and illustrate clearly how the attacks and test measurements are used to reach the goal. Thus this work will be of assistance to information security professionals, network administrators and software developers seeking to use these techniques and development to measure and identify this type of attack with the metrics proposed on the study.

Keywords: Denial Capture, Tool, Denial-of-service, Attack, Metrics.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

DoS	Denial-of-Service
DDoS	Distributed Denial-of-Service
IDS	Intrusion Detection System
NIDS	Network Intrusion Detection System
HIDS	Host Intrusion Detection System
CPU	Unidade Central de Processamento
SYN	Synchronyze
ACK	Acknowledgement

SUMÁRIO

1 INTRODUÇÃO

A rede mundial de computadores é utilizada para inúmeras atividades nos dias atuais, como lazer, trabalho ou estudos. Existem milhões de pessoas conectadas à “grande rede” e nem sempre seus usuários fazem uso de maneira correta desse recurso. Muitas pessoas mal intencionadas buscam cada vez mais se especializar em práticas maliciosas como invasões e ataques a sites e servidores da web.

Dessa forma, a segurança da informação é um ponto fundamental a ser estudado e investido pelas grandes organizações, visto que geralmente tem a Internet como a sua principal via de comunicação e/ou vendas aos seus clientes/usuários que podem estar localizados em qualquer parte do planeta.

A segurança da informação de uma organização garante a continuidade do negócio de forma estável e permite que as pessoas e os bens estejam seguros de ameaças e perigos. Roubo de informações ou identidades, acesso a dados e contas bancárias ou ataques que comprometam a disponibilidade de um serviço podem gerar graves consequências para uma empresa (ESPIRITO SANTO, 2002).

Um desses fortes ataques que prejudicam a disponibilidade de um serviço é o de Negação de Serviço ou *Denial-of-Service* que pode ser aplicado de forma centralizada ou distribuída, esta última chamada de *Distributed Denial-of-Service (DDOS)*. Os ataques de (D)DoS consistem em enviar inúmeras requisições a um servidor ou serviço até que o mesmo fique indisponível para receber e enviar novas requisições de novos clientes legítimos, ou seja, que tentam usar o sistema da maneira correta e para o qual foi projetado.Através de dados coletados pela empresa PROLEXIC, um dos maiores provedores em soluções contra ataques de negação de serviço do planeta, no segundo trimestre de 2014 houve um aumento de 46% nos números de ataques (D)DoS em servidores do mundo inteiro comparado com ao mesmo período do ano de 2013. Também quando comparado o primeiro com o segundo trimestre de 2014 notou-se que o aumento ainda continuou grande, estima-se que o ano de 2014 terá recordes de ataques de negação de serviço no mundo (PROLEXIC, 2014).

A motivação para o estudo da temática se deu após a participação em um estágio de verão em um projeto de pesquisa realizado na *University of New Brunswick* / Canadá, onde sob orientação dos professores Dra. Natalia Stakhanova e Dr. Ali Ghorbani, na qual foram desenvolvidos cenários de ataques DoS e uma ferramenta que mensurava as métricas de impacto dos ataques. Assim, com uma perspectiva prática, buscou-se analisar o uso de CPU e memória da máquina vítima de um ataque, além das conversações e transações na camada de

rede com o objetivo de medir o impacto e detectar um possível (D)DoS. A pesquisa realizada no estágio se deu até as medições das métricas de memória e CPU, para a pesquisa do TCC foi adicionada a métrica para a análises dos pacotes da rede, isso de deu devido ao baixo impacto sofrido pela memória e CPU durantes os ataques teste realizados, então para o estudo deste TCC foi vista a necessidade de também serem analisados os pacotes da rede e medir seu impacto antes e durante um ataque de negação de serviço.

Para validar o conceito de detecção e mensuração de ataques (D)DoS a partir da memória, CPU e rede foi implementada a ferramenta chamada de *Denial Capture* onde através das informações coletadas em uma máquina, tenta identificar a possibilidade de um possível ataque de negação de serviço e mede seu impacto. Foram realizadas anteriormente várias simulações em diferentes cenários para validar a abordagem utilizada na ferramenta.

1.1 OBJETIVOS GERAIS

O presente trabalho tem por objetivo mostrar e descrever um dos principais ataques existentes na *Internet*, a Negação de Serviço (*Denial of Service*) os impactos que esse ataque pode causar em um servidor ou serviço em execução e propor uma ferramenta onde através de uma série de métricas coletadas na máquina "atacada" como memória, CPU e tráfego de rede, será possível identificar e medir um ataque *(D)DoS*.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos para o estudo da temática foram:

- Análise e construção de filtros eficientes para serem usados na ferramenta *tshark*, onde são analisados os pacotes trafegados na rede. Os dados retornados servem de parâmetros para analises do impacto dos ataques na camada de rede;
- Estudo dos principais características de utilização da Memória RAM e da Unidade Central de Processamento (CPU) de uma máquina, como espaço livre, total e usado que também serviram como objetos de análise de mensuração dos ataques simulados.Definir um meio de identificar e mensurar um possível ataque de negação de serviço a partir das métricas utilizadas para o estudo.

1.3 METODOLOGIA

Para categorização da pesquisa, utiliza-se a classificação apresentada por GIL (2008), que distingue em relação a dois aspectos: quanto aos objetivos e quanto aos procedimentos técnicos.

Quanto aos objetivos, a presente pesquisa é exploratória. Exploratória por se buscar estudar uma problemática envolvendo simulações de ataques de negação de serviço e uma ferramenta que identifique e mensure os impactos desses ataques, conforme acrescenta Gil (2008), nesse tipo de pesquisa busca-se maior familiaridade com o problema estudado.

A técnica adotada no presente trabalho é a de observação direta intensiva, que para Lakatos (2003, p. 222) essa técnica "não consiste apenas em ver e ouvir, mas também em examinar fatos ou fenômenos que se deseja estudar".

O universo da pesquisa se representará por um conjunto de computadores para fins de ataques e medições. No que diz respeito à amostra, conceituada por Lakatos (2003) como a porção ou parcela, convenientemente selecionada do universo (população) que são os computadores do laboratório da UFPB/CAMPUS IV que foram disponibilizados para a realização do estudo e de simulações de ataques e medições em geral.

Em relação aos procedimentos técnicos, a pesquisa é caracterizada como pesquisa experimental, pesquisa experimental, devido investigação empírica realizada no setor em estudo, onde ocorre a manipulação dos computadores para simulações e análise de dados.

1.4 CONTRIBUIÇÕES ESPERADAS

Portanto, justifica-se esse estudo pela possível contribuição para a segurança de redes das organizações e sites da *Internet*, para a academia tendo em vista a deficiência em estudos nessa área, proporcionando assim que a pesquisa seja aprofundada e embase o surgimento de posteriores estudos a cerca de identificação e mensuração de ataques de negação de serviço. Para o pesquisador o trabalho possibilitou o aprendizado através de referenciais teóricos e do estudo prático, além de agregar conhecimentos em análise de dispositivos como Memória RAM e CPU, desenvolvimento de software e análise da *network-layer* (camada de rede) através de análise de pacotes, sendo capaz de propor uma solução para a problemática encontrada com o aporte do professor orientador.

1.5 ESTRUTURA DO TRABALHO

O trabalho está estruturado da seguinte maneira: no capítulo 2 é apresentada a fundamentação teórica dos temas quer serviram de estudo para a temática; já no capítulo 3 temos o desenvolvimento do estudo, abordando a arquitetura da ferramenta, as suas funcionalidades e as simulações realizadas e no capítulo 4 é apresentada a conclusão do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo serão apresentadas as fundamentações teóricas de todos os temas que foram estudados e que serviram como base para o estudo dessa temática, entre esses assuntos estão: Segurança em Redes, *Denial-of-service* e Sistemas de Detecção de Intrusos.

2.1 SEGURANÇA EM REDES

A segurança no mundo seja pensando em violência urbana ou até mesmo em hackers cibernéticos é bastante peculiar. O contexto é caracterizado pela evolução contínua, no qual novos tipos de ataques têm como respostas novas formas de proteção, que consequentemente levam ao desenvolvimento de inovadoras técnicas de ataques, formando assim um ciclo (NAKAMURA, 2007)

Um mesmo cenário pode ser notado na segurança em redes de computadores, que pode ser definida segundo (CANTU, 2003) como "a conexão de dois ou mais computadores para permitir o compartilhamento de recursos e a troca de informações entre as máquinas". Nas grandes empresas e organizações ter boas práticas de segurança é um dos requisitos chaves para produtividade e competitividade no mercado. Uma empresa vulnerável e que não investe em segurança digital pode ser totalmente excluída de um mercado altamente aquecido onde a segurança e integridade das informações são exigências tão requeridas por seus clientes/usuários.

2.1.1 O MODELO DE REFERÊNCIA ISO/OSI

Segundo HARINATH (2013), o modelo OSI tem o objetivo de realizar uma comunicação aberta entre os diferentes sistemas, sem exigir alterações a lógica do hardware e/ou software. O modelo OSI não é um protocolo, sendo caracterizado como um modelo para compreensão de projeção de uma arquitetura de rede flexível, robusta e interoperável.

A ISO do inglês *International Standards Organization* define o *Open Systems Interconect* (OSI) em um modelo de referência de camadas de comunicações distintas que tem como objetivo dividir e diferenciar diferentes serviços entre as entre essas camadas. No total existem 7 camadas no modelo de referência OSI que são a camada física, camada de rede, camada de transporte, camada de sessão, camada de apresentação e a camada de aplicativo. Na tabela 1 a seguir são apresentadas as 7 camadas do modelo:

Tabela 1 – Camadas do Modelo OSI

7	Camada de Aplicação
6	Camada de Apresentação
5	Camada de Sessão
4	Camada de Transporte
3	Camada de Rede
2	Camada de Enlace
1	Camada Física

Fonte: Elaboração própria (2014)

As camadas sempre complementam funções ou tarefas realizadas por cada camada anterior e são divididas em 3 grupos, o grupo da aplicação que abrange as camadas 5, 6 e 7, o grupo de transporte que é representado pela camada 4 e o grupo de rede que possui as camadas 1, 2 e 3.

Para tanto, são descritas as funções e objetivos de cada uma das camadas a seguir:

- Camada de Aplicação (Camada 7): Responsável pela interface entre a aplicação e o protocolo de comunicação;
- Camada de Apresentação (Camada 6): Fica encarregada da conversão dos dados que são recebidos da camada de aplicação para um formato que seja compreendido pelo protocolo que está sendo utilizado;
- Camada de Sessão (Camada 5): É responsável por dar permissão à dois computadores distintos estabeleçam comunicação entre si;
- Camada de Transporte (Camada 4): É a camada que recebe os dados da camada de sessão, dividi-los em pacotes e transportá-los para a camada de rede;
- Camada de Rede (Camada 3): Responsável pelo endereçamento de pacotes, convertendo endereços lógicos em físicos, para que os pacotes cheguem corretamente ao seu destino;
- Camada de Enlace (Camada 2): Captura os dados recebidos da camada de rede e os transforma em quadros que serão enviados pela rede;
- Camada Física (Camada 1): Transforma os quadros enviados pela camada de enlace em sinais compatíveis com os meios que serão transmitidos, como por exemplo sinais elétricos ou luminosos para fibra óptica.

2.1.2 A LINGUAGEM DA INTERNET: O PROTOCOLO TCP/IP

Sendo o protocolo de rede mais utilizado da atualidade para interligar instalações de computação, o TCP/IP foi desenvolvido por patrocínio da DARPA (*Defense Advanced Research Projects Agency*). (YANG, 1997).

O TCP/IP que vem do inglês *Transport Control Protocol/Internet Protocol* pode ser considerada como a "linguagem da *Internet*", pois qualquer informação que trafegue na rede mundial de computadores é compreendida e transportada através desse protocolo. Isso pode ser relacionado com as funcionalidades de rede e transporte que foram mencionados anteriormente e que estão presentes no modelo OSI, correspondendo as camadas 3 e 4 respectivamente, além de possuir relações com as camadas de aplicação e enlace/interface.

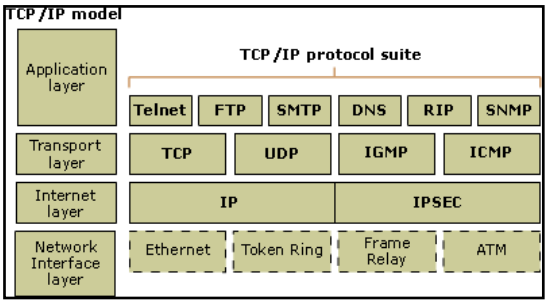

Figura 1 – Representação do Protocolo TCP/IP
Fonte: microsoft.com (2014)

Na figura 2 foram ilustradas as camadas de aplicação, transporte, internet (rede), interface (enlace) que são correspondentes ao protocolo TCP/IP. Na camada de aplicação são ilustrados os principais protocolos utilizados como Telnet, FTP, DNS, SNMP, já na camada de transporte alguns dos seus protocolos são ilustrados como o TCP e UDP, na camada de rede estão presentes os protocolos IP e IPSEC e finalmente na camada de interface ou enlace são encontrados o Ethernet, Token Ring, Frame Relay e ATM.

2.1.3 SERVIÇOS E TECNOLOGIAS PARA SEGURANÇA DE REDES

Segundo a *Cisco Systems*, muitas ameaças de segurança de redes estão espalhadas pela *Internet*. As mais comuns são:

- Vírus, *worms* e cavalos-de-troia: programas maliciosos que são instalados sem o consentimento do usuário, geralmente transmitidos pela Internet e se aproveitam de brechas nos sistemas operacionais;
- Spyware e *adwares*: programas em que sem o consentimento do usuário roubam informações ou dados confidenciais e são repassados para elementos externos

(spyware) e exibem grande quantidade de propagandas e anúncios analisando o interesse do usuário através de já sites acessados (adware).

- Ataques *zero-day*, também chamado de ataques *zero-hour* A tática de tirar proveito de vulnerabilidades de software e sites antes que os seus desenvolvedores resolvam possíveis brechas ou vulnerabilidades no sistema.
- Ataques hackers: Ataques que geralmente procuram invadir ou prejudicar a disponibilidade de sites ou sistemas.
- Ataques de negação de serviço: Tentativa de tornar um sistema ou um servidor indisponível através da grande de grandes quantidades de envio de requsições para consumir os recursos de uma vítima como memória e CPU ou "inundar" a rede com milhares de pacotes,
- Roubo e intercepção de dados: Captura de informações e dados sem o consentimento do usuário;
- Roubo de identidade: Aproveitar-se da identidade digital de usuários, como por exemplo login e senha, para fraudar acessos em sistemas.

Para evitar os tipos de ameaças listadas anteriormente devem existir métodos mais eficazes para prover segurança em redes. Esses métodos devem ser compostos por uma arquitetura que especifique e forneça os seguintes serviços listados na tabela abaixo:

Tabela 2 – Serviços para Segurança de Redes.

Serviço	**Característica**
Autenticação	Capaz de identificar usuários utilizando os serviços da rede.
Autorização	Permissão de atribuição de privilégios aos serviços.
Controle de Acesso	Permissão de atribuição de quem poderá usar os serviços e seu respectivo privilégio (restrições de acesso).
Confidenciabilidade	Garantir que os dados e informações sejam compreensíveis apenas para as partes finais da comunicação.
Não-Repúdio	Confirmas a autoria de uso dos serviços.
Auditoria	Permissão de geração de logs para verificação de eventos.

Fonte: Adaptada de MARIANO, I. S. (2007)

Portanto, visando a segurança de redes os serviços listados na tabela 2 podem ser implantados com as tecnologias listadas na tabela abaixo:

Tabela 3 – Tecnologias para Segurança de Redes

Tecnologia	Característica
Criptografia	Técnica de codificação de dados utilizando chaves públicas ou privadas, tornando assim os dados incompreensíveis. Alguns exemplos de algoritmos de criptografia são: DES, RSA e BLOWFISH. A criptografia garante a Autenticação.
Firewall	Dispositivo de controle de pacotes que podem entrar ou sair da rede. O firewall garante o controle de acesso.
IDS	Dispositivo que monitora constantemente a rede. Pode tomar ações na medida em que algum evento ocorra. O IDS pode garantir a detecção de intrusos na rede.
Integridade	Técnica que garante que os dados não foram alterados. Utiliza criptografia como SHA e MD5 para esse objetivo.
Assinatura Digital	Técnica que garante a autoria dos dados utilizando-se de criptografia.

Fonte: Adaptada de MARIANO, I. S. (2007)

2.2 DENIAL-OF-SERVICE

Denial-of-Service ou Negação de Serviço é um tipo de ataque que faz informações ou dados ficarem indisponíveis em um sistema. Existem vários métodos para executar esse tipo de ameaça onde principal objetivo é "sufocar" a rede de uma vítima com inúmeras requisições e pacotes para torná-la inacessível para outros clientes (RAO 2011).

Porém, além do tradicional meio de ataque de negação de serviço através de uma grande quantidade de pacotes e requisições, existem outras maneiras de se executar um ataque DoS, como congestionar recursos de rede, consumir memória e CPU, reduzir energia do computador, explorar *timers*, envenenamento do tradutor de domínios, entre outros (RAO 2011).

Ataques *Denial-of-Service* (DoS) e sua variação, chamada de *Distributed Denial-of-Service* apresentam um ameaça para quase todos os serviços da internet. O poder de ataque desse tipo de ameaça pode causar sérios problemas para a disponibilidade de um servidor ou serviço.

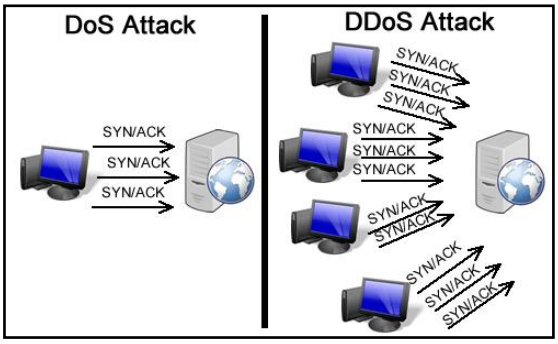

Figura 2 – Arquitetura de DoS e DDoS
Fonte: Acervo próprio

Na figura 2 é ilustrada a arquitetura de um ataque DoS e de um ataque DDoS, exemplificando em que no DoS o ataque parte geralmente de um pouco específico para uma máquina vítima, como ilustra a Figura 2 o atacante envia diversas requisições TCP SYN/ACK com o objetivo de gerar indisponibilidade e já no exemplo do DDoS o ataque pode partir de diversos pontos para uma vítima específica, assim tendo um poder de ataque bem maior, já que agora o número de requisições para uma única vítima será bem maior.

Embora ameaças como o DoS e DDoS possam causar grandes impactos, *hackers* e/ou *crackers* podem ter um trabalho significativo para preparar esses tipos de ataques, principalmente se o ataque for de forma distribuída, pois para realizar o ataque distribuído é necessário a obtenção de várias máquinas para operar como "zumbis", que podem ser descritos como vários computadores em qualquer parte do mundo que irão atuar em um ataque e geralmente sem o consentimento do usuário. A maior parte desses zumbis se criam através de malwares que se infiltram no sistema operacional de diversas vítimas com esse objetivo, assim podendo serem formados gigantescos ataques de negação de serviço de forma distribuída.

2.2.1 MECANISMO DO ATAQUE

2.2.1.1 SYN-FLOOD

O mais conhecido tipo de ataque DoS é feito por *flood* (envio intenso) de requisições TCP. O cliente manda um sinal SYN (*Synchronize*) ao servidor, e o mesmo responde com outro sinal chamado ACK (*Acknowledgement*) e o cliente responde assim com outro ACK. Em ataques dessa natureza o atacante repete esse processo centenas de vezes, ou seja, mandando vários sinais SYN seguidos até sobrecarregar o servidor, ao ponto de não aceitar

mais nenhuma conexão. Nas imagens abaixo são ilustradas uma conexão normal (Figura 3) e uma conexão com SYN-Flood (Figura 4).

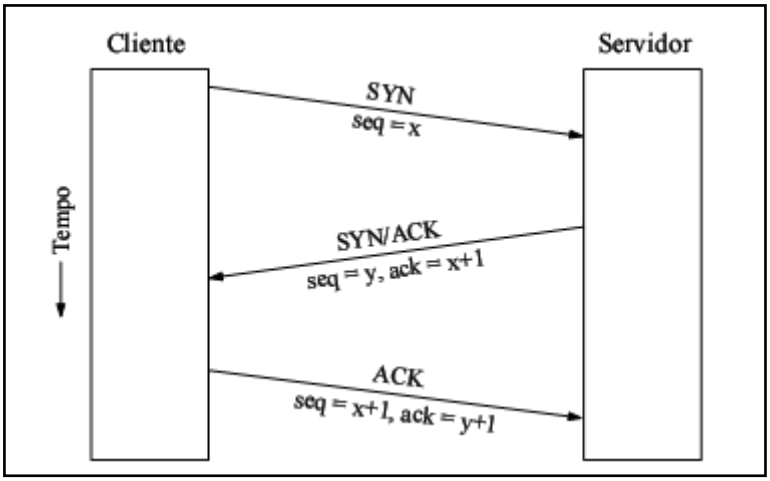

Figura 3 – Conexão TCP normal

Fonte: Amorim, R. D. (2007)

Na figura 3 é ilustrada uma conexão TCP que foi sucedida. A primeira requisição SYN de um cliente foi retornada com a resposta ACK do servidor, que também recebe outro *Acknowledgement* do cliente, assim caracterizando uma conexão normal.

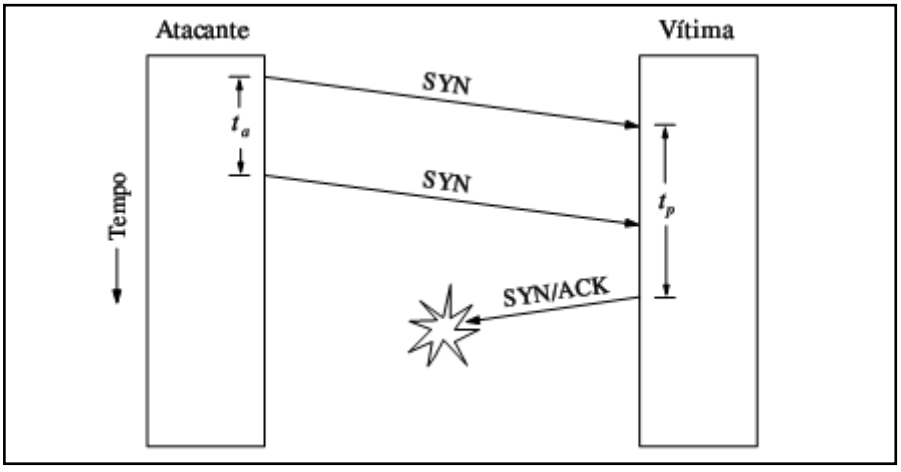

Figura 4 – Ilustração de SYN-Flooding

Fonte: Amorim, R. D. (2007)

Na figura 4 é apresentada uma conexão TCP com SYN-Flooding, o atacante envia requisições SYN simultâneas, sobrecarregando o servidor assim impossibilitando o retorno ACK.

2.2.1.2 DENIAL-OF-SERVICE DISTRUBUÍDO (DDoS)

Com o crescimento da *Internet* os ataques de negação de serviço feitos de forma distribuída obtiveram grande crescimento. A sua principal característica é utilização de "zumbis" para a tarefa de ataque. Através de malwares ou invasões várias máquinas são utilizadas simultaneamente para atacar vítimas específicas (AMORIN, 2007).

Utilizando-se dessas redes "zumbis" os atacantes podem programar grandes ataques e ainda com a possibilidade de esconder sua identidade de uma maneira mais fácil, pois dessa vez estarão vários computadores atuando como atacantes.

A figura abaixo (Figura 5) ilustra a estrutura de um ataque DDoS, onde um atacante utiliza-se de várias máquinas subordinadas (zumbis) a ele para executar ataques de negação de serviço em pontos específicos.

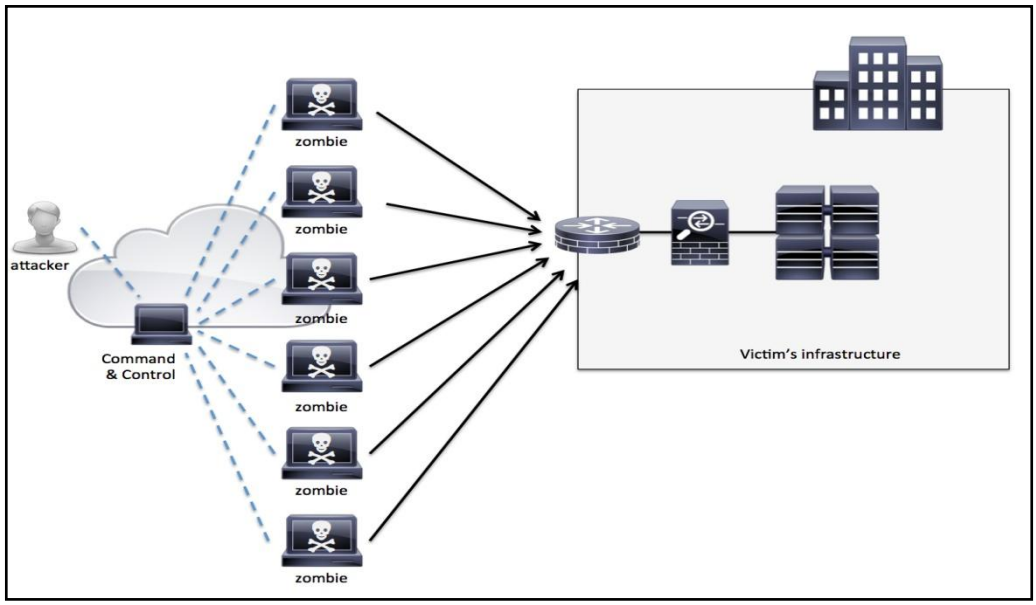

Figura 5 – Ilustração de um ataque DDoS.

Fonte: cisco.com (2014)

Na figura 5 é caracterizado um ataque de negação de serviço distribuído, onde existe um atacante que tem um controlador *handler*, no qual o mesmo é responsável por conseguir máquinas distribuídas em diferentes localizações, chamadas de "zombies", através de várias formas como por *malwares* ou invasões. Assim, os *zombies* executam a função de atacante com o objetivo de atacar a infra-estrutura de rede e/ou serviços da vítima.

2.3 SISTEMA DE DETECÇÃO DE INTRUSOS

Um sistema de detecção de intrusos é utilizado para detectar tipos de comportamentos maliciosos que possam comprometer a segurança de um sistema ou de uma rede.O IDS (do inglês *Intrusion Detection System*) auxilia os sistemas de informação a se prepararem a lidar com diversos tipos ataques. Isso é possível através da coleta de informações de uma variedade de fontes de sistemas e redes, assim sendo possível analisar possíveis problemas de segurança (ROZENBLUM, 2011).

O conceito de *Sistema de Detecção de Intrusos* surgiu na década de 80 no *Stanford Research Institute*. O Sistema de Detecção de Intrusos é um dispositivo para monitoramento de redes em tempo real. Existem alguns tipos de IDS, os principais são o baseado em rede, chamado de *Network-Based* **(NIDS)** e o baseado em host, chamado de *Host-Based* **(HIDS).**

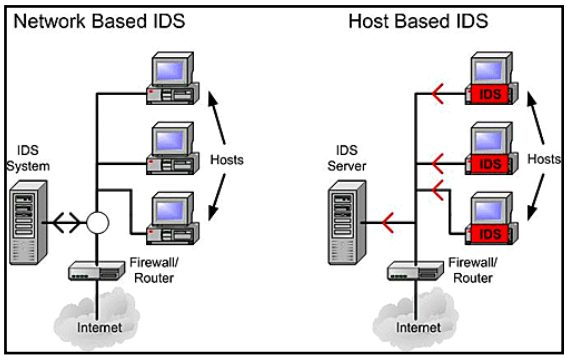

Figura 6 – Arquiteturas de HIDS e NIDS

Fonte: informit.com (2014)

Na figura 6 é demonstrada a diferença de arquitetura de um sistema de detecção de intrusos baseado em *host* e baseado em rede. Na ilustração do *Network Based IDS* existe um sistema de IDS instalado na própria rede, conectado ao *router* e ao *firewall*. Já no *Host Based IDS* os sistemas de detecção são instalados nas próprias máquinas da rede.

Segundo a Rozenblum (2011), a detecção de intrusão provê aos sistemas e redes:

- Monitoramento e análise de usuários e sistemas ativos;
- Auditoria de configurações de sistemas e vulnerabilidades;
- Avaliação da integridade de sistemas críticos e dados;
- Análise de atividades anormais;
- Auditoria do sistema operacional;
- Análise estatística de atividades baseadas no confronto de conhecidos ataques.

Nesse contexto, um IDS de rede atua unicamente nas camadas do modelo TCP/IP e/ou OSI, mas podendo detectar atividades maliciosas a partir do tráfego direcionado a qualquer *host* na rede. Por outro lado, o IDS de *host* atua especificamente em um sistema, permitindo analisar, além do tráfego de rede, outras categorias de informações como eventos nos recursos da máquina.

2.3.1 TIPOS DE SISTEMA DE DETECÇÃO DE INTRUSOS

2.3.1.1 BASEADO EM REDE (NETWORK-BASED)

O Sistema de Detecção de Intrusos baseado em rede (**NIDS**) é instalado na própria rede a ser monitorada e através de uma base de dados faz análise e decodificação dos pacotes da rede, verificando também os protocolos de rede, usuários externos, ataques de negação de serviço e proteger a integridade dos dados.

O NIDS age na camada 3 (camada de rede do modelo OSI) em modo promíscuo, analisando todo o tráfego que é passado (ROZENBLUM, 2011). Uma vez que qualquer atividade anormal for identificada é enviado um alerta para o administrador da rede, um exemplo disso seria instalar o NIDS onde fica localizado o firewall objetivando detectar possíveis intrusos na rede.

2.3.1.2 BASEADO EM HOST (HOST-BASED)

Também chamado de **HIDS**, o Sistema de Detecção de Intrusos baseado em host é instalado em uma máquina específica na rede para detectar possíveis invasões ou ataques. Analisando eventos do sistema operacional, eventos de acesso ou de aplicações, com o HIDS é possível bloquear ataques que não são detectados pelo firewall, como por exemplo, ataques protocolos criptografados. (EVANGELISTA, 200).

Existem dois tipos de aplicativos IDS baseados em host:

- **Analisadores de eventos** (ocorrências em uma rede ou num computador): Procura por conexões abertas de rede e monitoram portas do sistema;
- **Analisadores de unidades de disco do sistema**: Analisa unidade de disco e outros periféricos do sistema e cria uma base de dados. Essa base de dados é como se fosse a situação original do sistema e sempre que ocorrer uma mudança o IDS pode gerar um alerta ou registrar a mudança (EVANGELISTA, 2008)

3 DESENVOLVIMENTO

Neste capítulo serão abordados detalhes do desenvolvimento e execução da ferramenta *Denial Capture*, e como será o comportamento de medição e identificação de um ataque de negação de serviço em um servidor.

3.1 A FERRAMENTA

3.1.1 ARQUITETURA DA APLICAÇÃO NA REDE

A localização da ferramenta será na máquina vítima de um ataque *Denial-of-service*, a ilustração abaixo mostra um ataque DDoS até chegar na máquina onde está localizada a ferramenta.

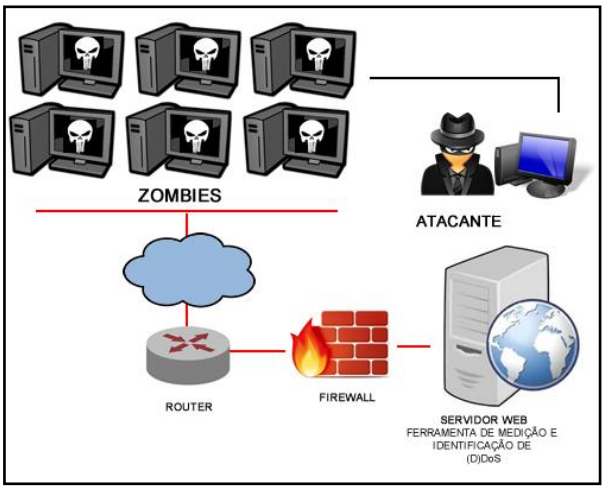

Figura 7 – Arquitetura da Ferramenta na Rede

Fonte: Acervo próprio

Na figura 7 é ilustrada uma arquitetura de rede e a localização da ferramenta nesta arquitetura. O atacante inicia o ataque de forma única ou distribuída (na figura 7 de forma distribuída) com o objetivo de chegar ao servidor web (onde está instalada a ferramenta), passando antes pelo roteador e o firewall da rede. A ferramenta para análise está instalada na máquina denominada “Servidor Web”.

3.1.2 MÉTRICAS DEFINIDAS PARA AS MEDIÇÕES

As métricas utilizadas para identificar e medir um ataque foram analisar as informações referentes a utilização da memória RAM, do CPU e dos pacotes trafegados na

rede, mais precisamente os pacotes que estão trafegando na porta 80 e 443 (portas padrões para serviços web), já que o servidor web será o alvo dos ataques testes. A seguir serão detalhadas as razões pela quais tais métricas foram escolhidas:

- **Memória RAM:** Do inglês *Random Access Memory*, é um tipo de memória que fornece a leitura e escrita dinâmica e é utilizada como memória primária em sistemas eletrônicos digitais. A importância da visualização da utilização desse componente é importante para mensurar a quantidade de leitura e/ou escrita de dados que possam estar ocorrendo durante um ataque que podem ocorrer durante um ataque (D)DoS, níveis altos de utilização podem server como alerta de atenção que será implementado na ferramenta;
- **Unidade Central de Processamento (CPU):** Do inglês *Central Processing Unit* é responsável por realizar as instruções de uma máquina. O uso elevado desse componente pode significar possíveis anomalias, assim sendo importante a verificação dessa métrica;
- **Pacotes na Rede:** Os pacotes trafegados nas portas 80 e 443 serão fundamentais para analisar possíveis anomalias em nível de rede, já que para a realização de um ataque de negação de serviço requisições terão que ser enviadas para um servidor vítima via rede.

3.1.3 FERRAMENTAS E COMANDOS UTILIZADOS PARA ANÁLISE DAS MÉTRICAS E DESENVOLVIMENTO DA APLICAÇÃO

Para o desenvolvimento e coleta de dados das métricas para as medições e identificações de possíveis ataques foram definidas as seguintes ferramentas e comandos que deram aporte ao desenvolvimento da ferramenta e análise dos dados:

- **Sistema operacional Linux e Servidor Apache**: Foi utilizado o sistema operacional Linux (distribuição Ubuntu Server 14.04 e Debian) para rodar a aplicação e por ser plataforma da maioria dos servidores web (alvo dos testes de ataque) no mundo e testes para as análises das métricas, a ferramenta de servidor web escolhido para os ataques foi o *Apache*.
- **Linguagem de programação JAVA:** Foi utilizada para desenvolver todo o *back-end* da ferramenta, sua escolha se deu pela boa interação que a linguagem possui com os comandos Shell do Linux, comandos como *free*, *mpstat* e *tshark* que retornam informações da memória RAM, CPU e pacotes na rede (serão melhor detalhados posteriormente) respectivamente;

- **Twitter Bootstrap:** Framework utilizado para o *front-end* da ferramenta, para a obtenção de um retorno organizado e bem estruturado das informações.
- **MySQL:** Banco de dados relacional utilizado para persistir e armazenar os dados coletados das métricas;
- **Sendmail:** Comando Linux de envio de emails que foi utilizado para o envio de alarmes e notificações;
- **Free:** Comando Linux que foi responsável pela coleta das informações de utilização da memória RAM, os dados ilustrados na Figura 7 exibirá o comando *free* e suas informações de retorno.

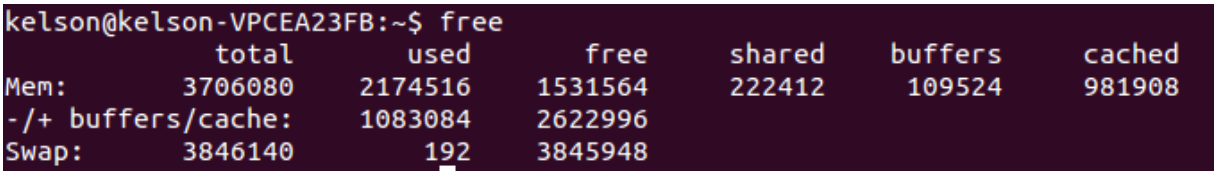

Figura 8 – Ilustração do comando free
Fonte: *print screen* do comando free.

  - Total: Espaço limite de armazenamento da memória RAM;
  - Used: Espaço utilizado na memória RAM no momento da consulta;
  - Free: Espaço disponível para armazenamento no momento da consulta;
  - Shared: Indica o espaço de memória compartilhada;
  - Buffered: Indica o total de memória carregada por diferentes aplicações;
  - Cached: Indica o espaço utilizado para *caching* de arquivos;
  - Swap: Indica o total de memória para *swap* (área de troca).
- **Mpstat:** Comando Linux que foi responsável pela coleta de informações do CPU, na Figura 8 as informações de retorno do comando são ilustradas.

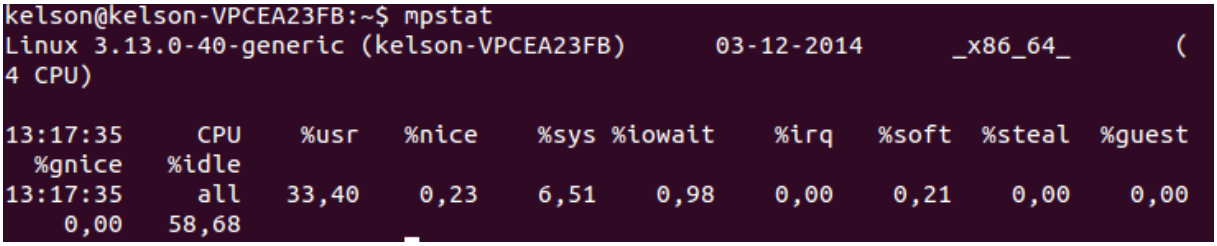

Figura 9 – Ilustração do comando mpstat.
Fonte: *print screen* do comando mpstat

Na figura 9 é ilustrado o retorno de dados do comando *mpstat*, fornecendo algumas informações em porcentagens sobre a utilização da CPU, mencionadas a seguir:

  - %usr: Utilização no momento da CPU enquanto é executado a nível de usuário;

- %nice: Utilização da CPU no momento quando ocorre execução a nível de usuário (aplicação) com prioridade para o nice;
- %sys: Utilização da CPU a nível do sistema (*kernel*);
- %iowait: Tempo em que o CPU ficou inativo durante solicitações pendentes no disco (*input* e *output*);
- %irq: Tempo gasto pela CPU em interrupções de serviços;
- %soft: Tempo gasto em interrupções de *softwares*;
- %idle: Tempo de inatividade da CPU.

- **Slowlors**: O script perl (http://ha.ckers.org/slowloris/) foi responsável pelas execuções dos ataques *DoS* ao servidor web.O *slowloris* funciona basicamente enviado centenas de requisições HTTP a um servidor web vítima, como a intenção de causar a indisponibilidade do mesmo.

3.2 ANÁLISES E SIMULAÇÕES PARA OS RESULTADOS DAS MÉTRICAS

A ferramenta irá fazer a coleta e manipulação dos dados para obter uma listagem de diferentes medições em diferentes *timestamps* (hora exata em que determinado evento ocorre) em um tempo de duração total determinado. Porém antes de iniciar o desenvolvimento da aplicação foi necessário fazer um estudo dos comportamentos das métricas em dois diferentes cenários: um sem ataques (D)DoS e outro com ataque de andamento, então foram realizados vários testes manuais para essa tarefa.

As próximas subseções mostrarão como foram analisados todos os dados das métricas escolhidas para o estudo (análise da Memória RAM, CPU e rede) e calcular o aumento percentual das informações das métricas com a ocorrência de um ataque (D)DoS.

Para realizar as medições da Memória RAM e CPU foi desenvolvido um *shellscript* para executar a cada 20 segundos os comandos da coleta de informações durante um espaço total de tempo de 3 minutos.

Esses tempos foram definidos depois de várias análises que deram a percepção que no decorrer dessa contagem já era possível capturar várias informações diferentes sobre as métricas desejadas. A Figura 10 ilustra o código *shellscript* criado para capturar das duas métricas.

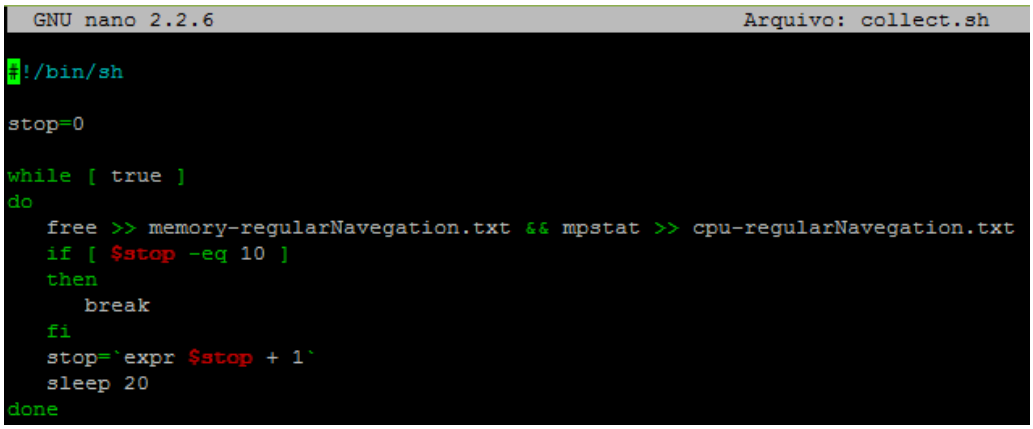

Figura 10 – Código fonte do shellscript

Fonte: *print screen* do código fonte

Na Figura 10 é ilustrado o código *shellscript* que foi responsável pela captura das informações da Memória RAM e do CPU. O código possui um laço "*while*" que executará os comandos **free (dados relativos ao uso da Memória RAM)** e **mpstat (dados relativos ao uso do CPU)** e gravará as suas saídas de dados em arquivos de texto para análises futuras (memory-regularNavegation.txt e CPU-regularNavegation.txt respectivamente) por 9 vezes em intervalos de tempo de 20 segundos (função sleep), ou seja, terá execução total de 3 minutos.

Simultaneamente em outro terminal aberto é executado o comando para a captura dos pacotes, também salvando a sua saída de dados em um arquivo de texto para as análises posteriores. A Figura 11 ilustra o comando *tshark* sendo executado.

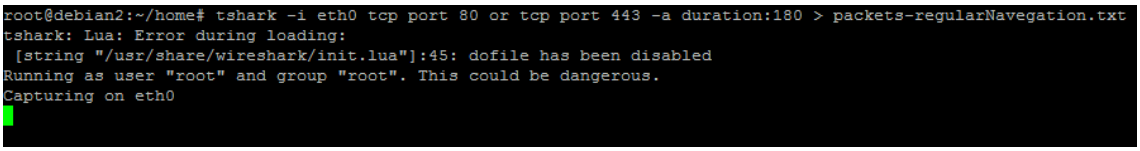

Figura 11 – Comando tshark

Fonte: *print screen* do comando tshark

Na figura 11 o comando *tshark* possui os seguintes parâmetros para filtrar apenas as informações necessárias para a coleta de pacotes da rede:

- **–i eth0:** para selecionar a interface de rede, eth0 já que o servidor está conectado a rede por essa interface;
- **tcp port 80 or tcp port 443:** para retornar apenas pacotes TCP, já que após um estudo sobre a ferramenta *slowloris*, são enviados apenas requisições HTTP, ou seja, sendo necessário somente a análise dos pacotes TCP trafegados na porta 80 e 443, portas padrões de servidores web;

- **-a duration:180:** a captura de pacotes irá durar 180 segundos (3 minutos).

Com a forma de análise manual definida, foram escolhidos dois cenários para que se chegasse a um cálculo que retornasse uma possível probabilidade de um *Denial-of-Service*.

As próximas subseções trarão todos os dados coletados nessas análises manuais em um cenário sem ataques *DoS* e em um cenário com um ataque DoS em execução.

3.2.1 TRATAMENTO E COLETA DAS INFORMAÇÕES EM UM CENÁRIO SEM ATAQUES

Ao iniciar a coleta manual dos dados, foi simulada uma conexão com o servidor web, instalado na máquina vítima de testes, para isso foi adicionada uma página HTML de exemplo no servidor. Outros 2 dispositivos na mesma rede ficaram responsáveis por acessar e atualizar o acesso à essa página diversas vezes para simular um cenário normal de movimentação dos pacotes e sem ataques. A Figura 12 ilustrará a página inserida no servidor web.

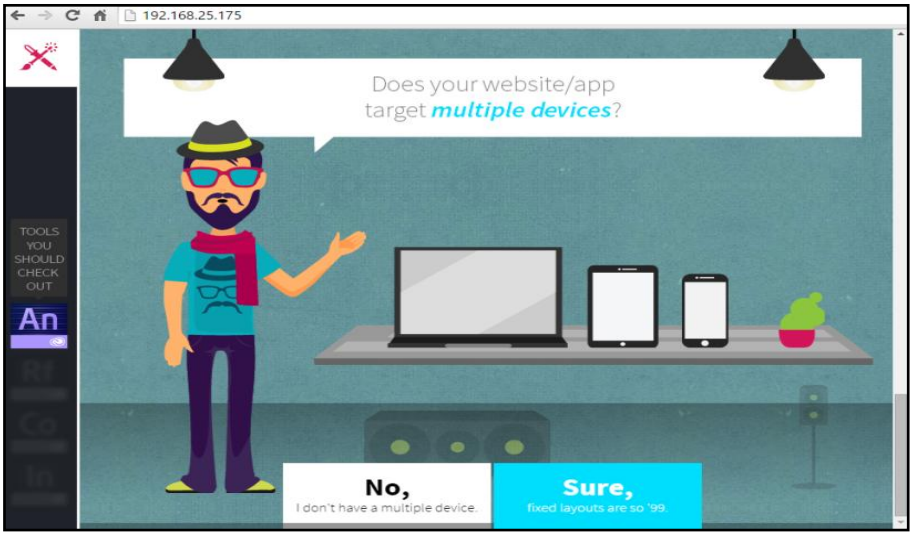

Figura 12 – Página HTML exemplo

Fonte: *print screen* da página exemplo

A página HTML ilustrada na Figura 12 foi pesquisada na internet e escolhida para ficar hospedada no servidor web de testes devido ao número significativo de componentes como imagens e *javascripts*, algo que é bem comum nos sites atuais. Com isso, foi possível ter uma simulação bem mais realista a nível dos pacotes trafegados na rede, já que existirão na página um número considerável de componentes que precisarão transitar pela rede para chegar ao destino.

Ao fim da coleta das informações que durou 3 minutos entre os *timestamps* **10:48:10** e **10:50:31**, obtivemos nos arquivos de texto as saídas de dados referentes a utilização das métricas de estudo.

As seguintes informações foram retornados nos arquivos de saída de dados referentes a utilização da **Memória RAM** e serão informadas na Tabela 4 representando as medições feitas durante 3 minutos.

Tabela 4 – Dados memória cenário sem ataque

Timestamp	Total	%used	%free	shared	buffers	cached
10:48:10	2074956 (2gb)	188920 (9%)	1886036 (91%)	0	12264	145388
10:48:10	2074956 (2gb)	213184 (10%)	1861772 (90%)	0	12280	145416
10:48:30	2074956 (2gb)	213572 (10%)	1861384 (90%)	0	12304	145428
10:48:50	2074956 (2gb)	213556 (10%)	1861400 (90%)	0	12328	145448
10:49:10	2074956 (2gb)	213556 (10%)	1861400 (90%)	0	12336	145448
10:49:30	2074956 (2gb)	213564 (10%)	1861392 (90%)	0	12368	145472
10:49:50	2074956 (2gb)	213688 (10%)	1861268 (90%)	0	12392	145476
10:50:10	2074956 (2gb)	213688 (10%)	1861268 (90%)	0	12416	145496
10:50:31	2074956 (2gb)	213680 (10%)	1861276 (90%)	0	12432	145500

Fonte: Elaboração própria (2015).

A Tabela 4 contém os dados referentes as medições realizadas na Memória RAM feitas durante 3 minutos de análises em um cenário sem ataques de (D)DoS, esses dados representam as informações salvas no arquivo de saída demonstrado anteriormente no código do *shellscript* de coleta.

Os dados referentes a utilização da memória estão presentes na coluna *%used* e indicam como está o uso em porcentagem desse dispositivo no seu respectivo *timestamp*. É perceptível que nesse cenário o uso da Memória RAM manteve-se constante e não apresentou modificações significativas.

Os outros dados presentes na Tabela 4 como *%free*, *shared*, *buffers* e *cached* também se monstraram constantes durante a coleta e servirão apenas como fonte de informação adicional para o usuário da ferramenta, ou seja, o dado mais relevante e que será utilizado nas medições posteriores é o de utilização atual *(%used*).

Os dados a seguir na Tabela 5 representam as medições realizadas na Unidade Central de Processamento simultaneamente com as medições feitas na Memória RAM demonstradas na Tabela 4.

Tabela 5 – Dados CPU cenário sem ataque

Timestamp	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%idle
09:48:10	0.18	0,00	1,99	1,19	0,00	0,14	0,00	0,00	96,51
09:48:10	0,18	0,00	2,01	1,18	0,00	0,14	0,00	0,00	96,49
09:48:30	0,18	0,00	2,01	1.18	0,00	0,14	0,00	0,00	96,49
09:48:50	0,18	0,00	2,00	1,18	0,00	0,14	0,00	0,00	96,50
09:49:10	0,18	0,00	2,00	1,17	0,00	0,14	0,00	0,00	96,51
09:49:30	0,18	0,00	1,99	1,17	0,00	0,14	0,00	0,00	96,53
09:49:50	0,18	0,00	1,99	1,17	0,00	0,14	0,00	0,00	96,53
09:50:10	0,18	0,00	1,99	1,17	0,00	0,14	0,00	0,00	96,53
09:50:31	0,18	0,0	1,97	1,16	0,00	0,13	0,00	0,00	96,56

Fonte: Elaborada própria (2015).

Também é perceptível que os dados referentes ao CPU na Tabela 5 mostraram-se constantes durante a análise. Será utilizado como dado principal o uso atual do CPU, que é representado pela coluna *%usr*. Os demais dados das outras colunas servirão como informações adicionais para o usuário da ferramenta.

Após as medições foram calculadas manualmente as médias de utilização de cada uma dessas métricas. A Tabela 6 informará o resultado da média de utilização de cada componente.

Tabela 6 – Média das utilizações da memória e CPU sem ataque

Média Utilização da Memória RAM	Média Utilização CPU
9,8%	0,18%

Fonte: Elaboração própria (2015).

Os dados representados na Tabela 6 representam o cálculo da média de utilização da Memória RAM e do CPU nas 9 diferentes medições realizadas durante os 3 minutos de análises. Esses dados servirão como base para se obter o aumento percentual das utilizações durante o teste com ataque.

Para se obter essas médias, foi utilizado o seguinte cálculo:

$$M = \frac{(uso1 + uso2 + uso3 + uso4 + uso5 + uso6 + uso7 + uso8 + uso9)}{9}$$

Então para obter a média de utilização da Memória RAM foi somada todas as utilizações da memória e resultado dividido pelo número de vezes da coleta (9 vezes). A

mesma forma foi adotada para a média do CPU, a soma das utilizações dividida pela a quantidade de vezes, que também foi 9.

Os dados que serão representados na tabela 7 representam as informações presentes no arquivo de saída do comando *tshark*, que foi ilustrado anteriormente na Figura 10.

Tabela 7 – Dados dos pacotes cenário sem ataques

Quantidade de pacotes (linhas do output):	586
Quantidade de requisições [SYN]:	20
Quantidade de respostas [ACK]:	258
IP mais freqüente no tráfego:	192.168.52.172
Quantidade de vezes que esse ip aparece:	575

Fonte: Elaboração própria (2015).

A Tabela 7 ilustra os dados coletados no arquivo de saída do comando *tshark* que também simultaneamente capturou os pacotes trafegados durante 3 minutos. Após uma consulta manual do arquivo de texto de retorno foram detectadas 586 linhas (pacotes diferentes) na captura, 20 pacotes SYN, 258 pacotes ACK, o endereço IP 192.168.25.172 como o mais presente no tráfego e sendo repetido 575 vezes.

Com isso foi possível obter os dados das métricas de estudo em um cenário sem ataques, a próxima etapa será a análise dos mesmos dados só que dessa vez em um cenário com um ataque de negação de serviço em execução.

3.2.2 TRATAMENTO E COLETA DAS INFORMAÇÕES EM UM CENÁRIO COM ATAQUE

Para obter os dados das métricas em um cenário com a simulação de um ataque único (não distribuído), foi utilizada a ferramenta *slowloris* e outra máquina na rede para atacar diretamente o servidor web na rede, após iniciado o ataque foram realizadas as medições das métricas com os mesmos procedimentos exemplificados na subseção anterior na máquina vítima (servidor web). A Figura 13 ilustra a execução de um ataque com a ferramenta *slowloris*.

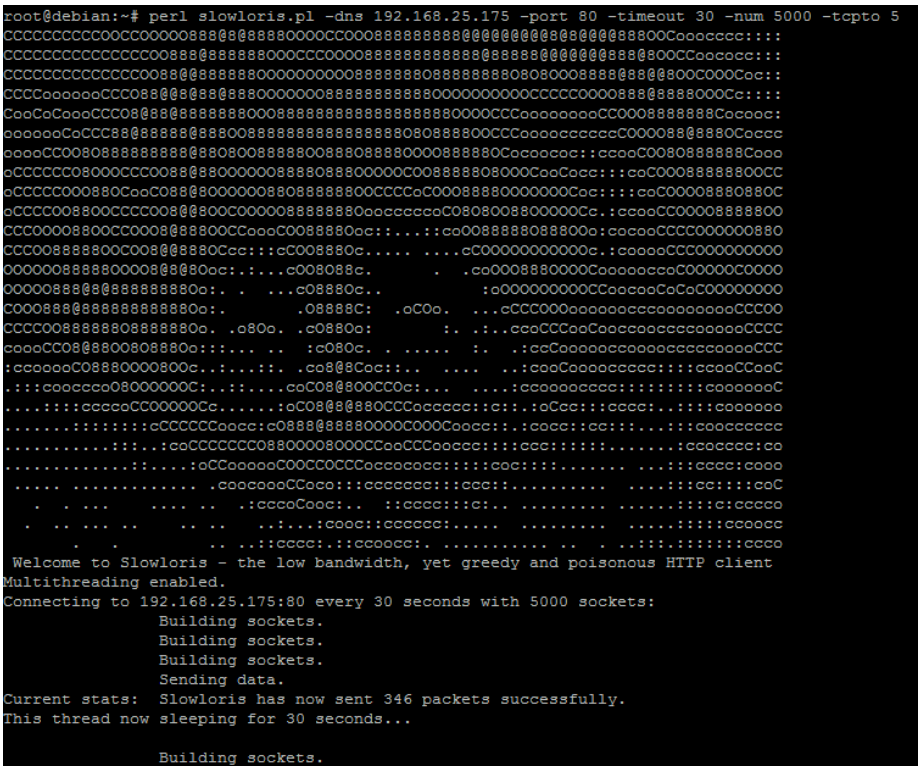

Figura 13 – Slowloris em execução

Fonte: *print screen* do slowloris em execução.

Na Figura 13 é ilustrada a ferramenta *slowloris* executando um ataque para o servidor web de teste 192.168.25.175, outros parâmetros também são especificados no comando como a porta 80 (porta do servidor web), timeout (pausa) de 30 segundos para cada conjunto de ataques e 5000 *sockets* (ponto final de um fluxo de comunicação) criados em cada conjunto de ataque. Esses parâmetros foram escolhidos para aumentar ainda mais o poder de ataque.

Após a execução do ataque, foi analisado se o servidor web já estava fora do ar para dar inicio coleta dos dados das métricas. A Figura 14 mostra o servidor web *offiline*.

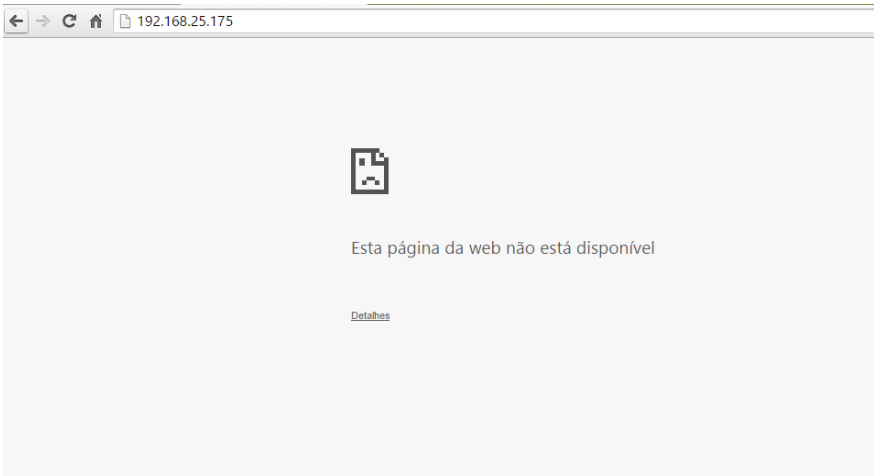

Figura 14 – Página fora do ar durante o ataque

Fonte: *print screen* da página fora do ar durante o ataque.

Com o servidor web indisponível a coleta das informações foi iniciada para a obtenção de um conjunto de dados das métricas referentes a um cenário sob um ataque DoS. A Tabela 8 mostra os resultados dessa coleta na Memória RAM.

Tabela 8 – Dados da memória cenário com ataque

Timestamp	Total	%used	%free	%shared	%buffers	%cached
09:34:36	2074956 (2gb)	224656 (10%)	1850300	0	12608	146008
09:34:56	2074956 (2gb)	226144 (10%)	1848812	0	12640	146880
09:35:16	2074956 (2gb)	227392 (10%)	1847564	0	12672	147644
09:35:36	2074956 (2gb)	228764 (11%)	1846192	0	12696	148356
09:35:56	2074956 (2gb)	229988 (11%)	1844968	0	12728	149072
09:36:16	2074956 (2gb)	230764 (11%)	1844192	0	12760	149568
09:36:36	2074956 (2gb)	231888 (11%)	1843068	0	12792	150236
09:36:56	2074956 (2gb)	233300 (11%)	1841656	0	12816	150912
09:37:16	2074956 (2gb)	234540 (11%)	1840416	0	12848	151664

Fonte: Elaboração própria (2015).

Foi perceptível que os dados coletados referentes a utilização da Memória RAM (coluna *%used*) durante 3 minutos (mesmo tempo determinado no cenário sem ataques) obteve-se bastante estável durante toda a coleta, apenas aumentando em 1% a partir do quarto *timestamp*. Isso mostra que a memória não sofreu um nível de impacto significativo com um ataque em andamento. Da mesma forma os outras informações como *%free*, *shared*, *buffers* e *cached* também se mantiveram estáveis e servirão apenas como informações extras para o usuário da ferramenta.

Na Tabela 9 serão demonstrados os dados coletados da utilização do CPU durante esses mesmos 3 minutos de coleta da Memória RAM.

Tabela 9 – Dados do CPU cenário com ataque

Timestamp	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%idle
09:48:10	0,17	0,00	1,88	1,06	0,00	0,16	0,00	0,00	96,73
09:48:10	0,17	0,00	1,91	1,05	0,00	0,16	0,00	0,00	96,71
09:48:30	0,17	0,00	1,93	1,05	0,00	0,17	0,00	0,00	96,69
09:48:50	0,17	0,00	1,94	1,05	0,00	0,17	0,00	0,00	96,67
09:49:10	0,17	0,00	1,95	1,06	0,00	0,17	0,00	0,00	96,65
09:49:30	0,18	0,00	1,96	1,08	0,00	0,18	0,00	0,00	96,61
09:49:50	0,18	0,00	1,97	1,08	0,00	0,18	0,00	0,00	96,60
09:50:10	0,18	0,00	1,98	1,08	0,00	0,18	0,00	0,00	96,58
09:50:31	0,18	0,0	2,00	1,08	0,00	0,18	0,00	0,00	96,56

Fonte: Elaboração própria (2015).

É possível notar que a utilização do CPU (coluna *%usr*) também se manteve bastante estável durante o ataque de negação de serviço, ou seja, o CPU também não teve impacto significativo durante o ataque. As outras informações (%nice, %sys, %iowait, etc) também não mostraram mudanças significativas durante o ataque.

Na Tabela 10 são informados os resultados dos cálculos da média de utilização dessas 2 métricas no cenário sob ataque.

Tabela 10 – Média das utilizações da memória e CPU com ataque

Média Utilização da Memória RAM	Média Utilização CPU
10,6%	0,17%

Fonte: Elaboração própria (2015).

Para a obtenção desses resultados também foi utilizada a mesma técnica do cenário sem ataques, ou seja, somado as utilizações e dividindo pelo número total de coletas (9).

É notório que a medição de uso dessas duas métricas (Memória RAM e CPU) não obtiveram impactos significativos de mudança durante um ataque. Isso se explica pois o ataque com a ferramenta *slowloris* se caracteriza apenas pelo envio de múltiplas requisições HTTP simultâneas, ou seja, afetando apenas o serviço web da máquina vítima, porém outros tipos de ataques de negação de serviço podem afetar o desempenho dessas duas métricas, sendo assim de extrema importância a continuidade da análise das mesmas para o funcionamento da ferramenta. A Tabela 11 mostrará o retorno de dados dos pacotes da rede.

Tabela 11 – Dados coletados dos pacotes

Quantidade de pacotes (linhas do output):	21535
Quantidade de requisições [SYN]:	5797
Quantidade de respostas [ACK]:	6095
IP mais freqüente no tráfego:	192.168.52.172
Quantidade de vezes que esse ip aparece:	21335

Fonte: Elaboração própria (2015).

Na Tabela 11 é possível perceber o grande aumento das informações, quando comparada com a Tabela 7. A captura dos pacotes que durou 3 minutos durante o ataque obteve dados muito superiores comparados com os dados da captura de pacotes do cenário sem ataque. Com isso é possível afirmar que diferentemente da Memória RAM e do CPU as informações referentes aos pacotes da rede sofrem bastante impacto durante um ataque de negação de serviço.

3.3 IMPLEMENTAÇÃO

O seguinte diagrama de classes ilustrado na figura a seguir (Figura 15) mostrará toda estrutura de classes do código **JAVA** da ferramenta:

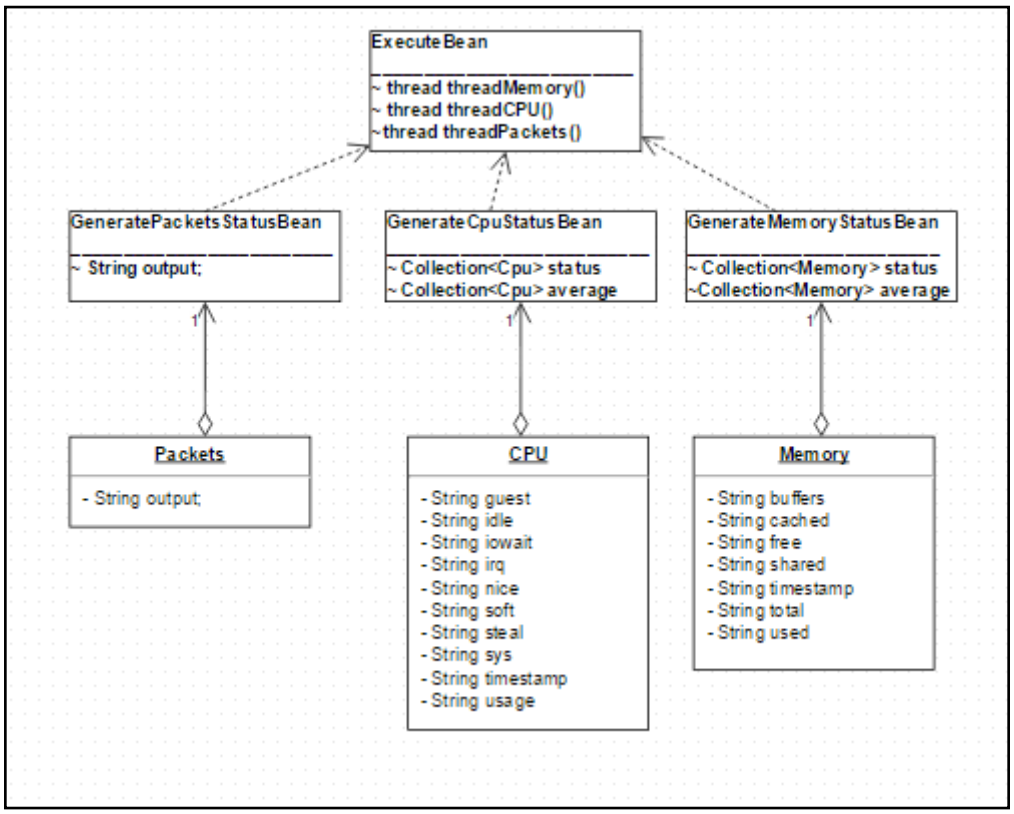

Figura 15 - Diagrama de Classes do código
Fonte: Diagrama de classes do sistema (2015)

Na Figura 15 são ilustradas todas as classes e métodos do código desenvolvidos até o momento, possuindo três entidades Memory, CPU e Packets que possuem como variáveis todos os dados retornados nos comandos *free*, *mpstat* e o retorno dos pacotes do *tshark*, respectivamente. Para o gerenciamento, execução, coleta dos dados e cálculo da média dos comandos de CPU e memória foram criadas duas classes: a *GenerateCpuStatusBeans* e a *GenerateMemoryStatusBeans*. Já para a coleta do output do tshark e atribuição das regras de tratamento da saída de dados foi criada a classe GeneratePacketsStatusBean. A execução da ferramenta e consequentemente dessas classes e métodos se darão através do acesso feito pela URL (Execução da classe ExecuteBean) da aplicação ou pelo método Main que será criado para as análises e notificações periódicas.

3.4 A FERRAMENTA EM EXECUÇÃO

Antes de ilustrar o funcionamento da ferramenta é de extrema importância demonstrar como as bases de informações dos cenários considerados sem ataques e com ataques são armazenados para a realização das consultas e comparações. Um banco de dados *MySQL* com as informações referentes as utilizações da Memória RAM, CPU e pacotes da rede darão suporte como valores de referências para identificar um possível (D)DoS. A Figura 16 irá ilustrar o banco de dados com os valores referentes aos diferentes cenários.

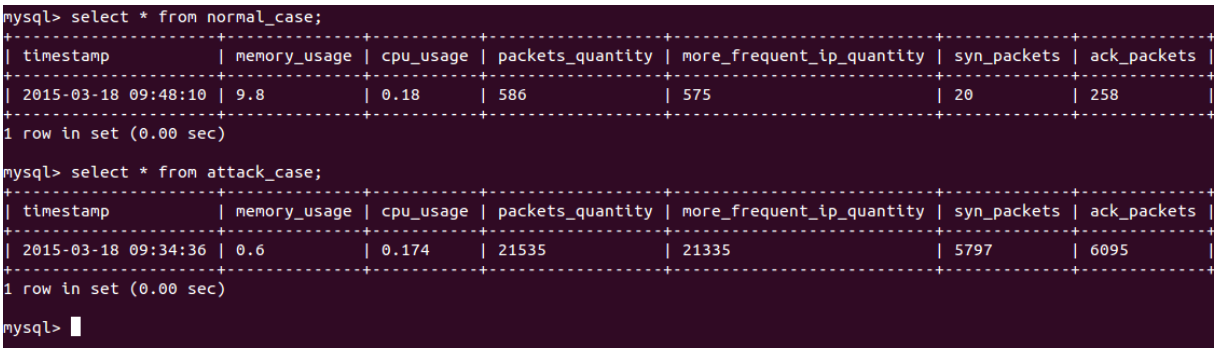

Figura 16 – Banco de dados dos dados pré-definidos

Fonte: *print screen* da página da aplicação

Através dos dados já obtidos com as medições exemplificadas nas seções 3.2.1 e 3.2.2 as tabelas do banco *normal_case* (ou caso normal) e *attack_case* (ou caso de ataque) foram preenchidas com as suas devidas informações. As colunas das tabelas de cada cenário são representadas pelas as informações de interesse das métricas, como: o uso da Memória RAM (*memory_usage*), uso do CPU (*cpu_usage*), quantidade de pacotes (*packets_quantity*), quantidade de vezes em que o ip mais freqüente aparece (*more_frequent_ip_quantity*), quantidade de pacotes [SYN] (*syn_packets*) e quantidade de pacotes [ACK] (*ack_packets*).

3.4.1 EXECUÇÃO DA FERRRAMENTA

Ao executar a ferramenta o sistema irá coletar os dados atuais das métricas de estudo e exibirá na tela as medições feitas durante 3 minutos de coleta. A Figura 17 ilustra a ferramenta após ser executada durante um *DDoS* teste realizado no laboratório do Campus IV da UFPB com 6 máquinas “atacando” o servidor web (porta 80) onde também se encontra a ferramenta.

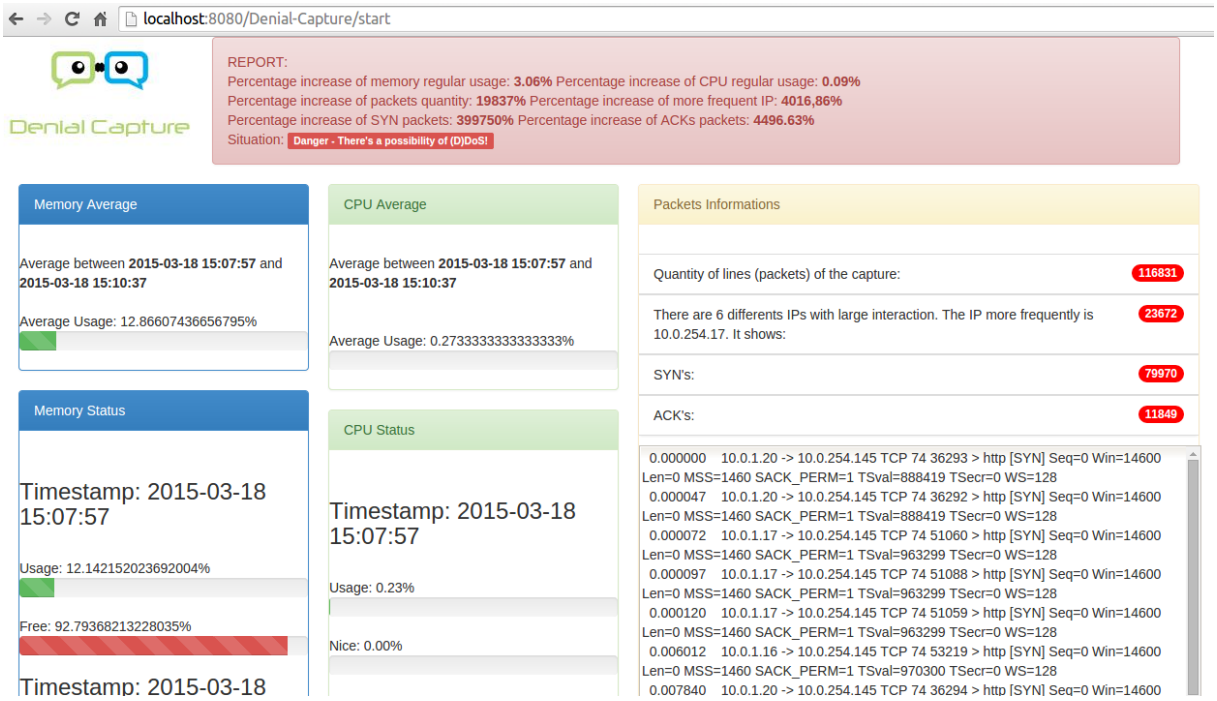

Figura 17 – Denial Capture em funcionamento

Fonte: *Print screen* da página da aplicação

Como ilustrado na Figura 17 a ferramenta retorna as informações referentes as coletas de dados das métricas. Informações como a média de utilização da Memória RAM (*Memory Average*), média de utilização do CPU (*CPU Average*) e informações dos pacotes (*Packets Informations*) foram retornadas nas caixas ilustradas na imagem, além de um relatório localizado na parte superior da tela (*REPORT*) que retorna os dados com os seus respectivos aumentos percentuais comparados ao um cenário sem ataques e também a definição se a situação é de um provável *(D)DoS* ou não. A seguir serão exemplificados como são feitos os cálculos dos aumentos percentuais e como confirmar um possível ataque.

3.4.2 CÁLCULO DO AUMENTO PERCENTUAL PARA MEDIR UM ATAQUE

Após a coleta das informações referentes a Memória RAM, CPU e rede, foi necessário analisar uma forma de comparar o impacto de um ataque *DoS* sob essas 3 métricas em um cenário sem ataque (dados já disponíveis no banco de dados) e outro com o ataque distribuído exemplificado e identificado na Figura 16. Então foi escolhida a forma de **aumento percentual** para medir o nível do ataque da medição realizada.

A Tabelas 12 mostramas médias de utilização da Memória RAM e CPU nos dois cenários sem ataque e com ataque (DDoS descrito na subseção anterior) e seu respectivo aumento percentual da diferença entre os dois.

Tabela 12– Médias dos diferentes cenários

Média de utilização da Memória RAM – Cenário sem ataques	Média de utilização da Memória RAM – Cenário com o ataque DDoS	Taxa de aumento
9,8%	12,86%	3,06%

Fonte: Elaboração própria (2015)

Já para calcular o aumento percentual dos dados referentes aos pacotes, foi necessária a utilização da formula padrão para cálculo de aumentos percentuais, já que os dados obtidos dos pacotes são retornados em quantidade e não em porcentagem. A fórmula a seguir exemplifica o cálculo da taxa de aumento das informações dos pacotes.

$$Tx(Aumento) = \left(\left(\frac{quantidade\ com\ ataque}{quantidade\ sem\ ataque}\right) - 1\right) * 100$$

Tabela 13 – Aumento percentual dos pacotes

Informações dos pacotes	**Cenário sem ataque**	**Cenário com ataque**	**Taxa de aumento**
Pacotes (linhas retornadas)	586	116831	**19837,03**%
Pacotes [SYN]	20	79970	399750%
Pacotes [ACK]	258	11849	4492,63%
IP mais frequente	575	23672	4016,86%

Fonte: Elaboação própria (2015).

Para explicar o cálculo da taxa de aumento a partir da formula citada anteriormente utilizaremos a quantidade de linhas retornadas (quantidade de pacotes). Então temos que a Taxa de Aumento será igual a (116831/ 586) – 1 que terá como resultado **198,3703071672** só que para obtemos esse número em porcentagem será necessário multiplicá-lo por 100 resultando em **19837,03%**.

3.4.3 IDENTIFICAÇÃO DE UM POSSÍVEL ATAQUE

Agora para identificar a possibilidade de um possível ataque, é preciso comparar os dados presentes na tabela *attack_case* com os dados coletados da captura das informações (não são utilizados os dados dos aumentos percentuais, mas sim os dados exatos das informações dos pacotes). As médias de utilização da Memória RAM e do CPU não serão consideradas na identificação do ataque, apenas na sua mensuração, já que em testes realizados anteriormente o desempenho desses dispositivos praticamente não são afetados com um ataque de (D)DoS com o *slowloris*.

Utilizando como exemplo os dados retornados na Figura 17, quando ocorria um ataque de negação de serviço distribuído, iremos compará-los com os dados presentes na tabela *attack_case* que representam a base de dados de um cenário em ataque.

Para realizar essa comparação foi utilizada a seguinte regra, se todas as informações referentes aos pacotes da coleta atual (coleta ilustrada na Figura 17) for **maior ou igual** do que os dados armazenados na tabela *attack_case* (base de dados de um cenário com ataque), então o sistema identificará a situação como um possível ataque (D)DoS. A Tabela X mostrará o exemplo de comparação com o ataque simulado e retornado na Figura 16.

Tabela 14 – Comparações para identificar ataque

Dados	**Dado coletado**	**Dados da base do cenário em ataque**	**Dado coletado >= Dado da base do cenário em ataque?**
Quantidade de pacotes	116831	21535	SIM
Quantidades de [SYN]	79970	5797	SIM
Quantidade de [ACK]	11849	6095	SIM
Frequência do IP que mais aparece	23672	21335	SIM

Fonte: Elaborada pelo autor (2015)

Como todas as informações atendem a regra especificada anteriormente, então essa situação pode ser considerada como um ataque (D)DoS, pois todos os dados da coleta mostraram-se superior aos dados já pré-definidos anteriormente como dados de um cenário em ataque. Assim a mensagem "Existe uma possibilidade de (D)DoS!" (ou *There's a possibility of (D)DoS*!) será retornada na tela da aplicação.

3.5 ENVIO DE ALERTAS PERIÓDICOS

Para o envio de emails periódicos ao usuário da ferramenta, foi criada posteriormente uma classe Java de nome *Execution.java* com um método *Main* que executa as mesmas medições exemplificadas anteriormente, depois de coletados os dados passam pelos mesmos passos citados antes para medir e identificar um possível ataque e no fim os dados são retornados via email (email previamente cadastrado no banco de dados da aplicação). Para isso é necessário que a aplicação seja executada periodicamente, a Figura 18 mostra a inserção de uma tarefa *cron* para executar a classe Java periodicamente.

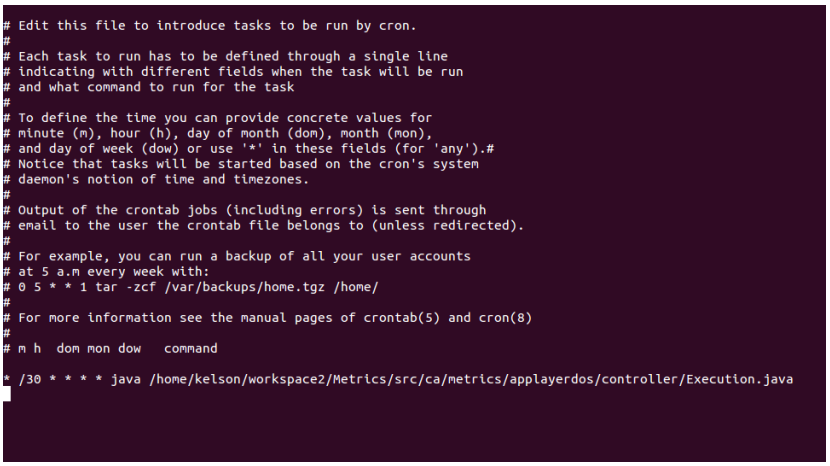

Figura 18 – Tarefa cron

A tarefa *cron* (execução automática de comandos em tempos determinados) ilustrada na Figura 18 será executada a cada 30 minutos todos os dias, com isso será possível receber os dados referentes as medições e alerta sobre a possibilidade de possíveis ataques usando como base os dados dessas medições.

3.6 DIFICULDADES ENCONTRADAS

Uma dificuldade encontrada durante o estudo e análise das métricas foi principalmente como seria a forma de definir um ataque de negação de serviço após a coleta e tratamento dos dados, então depois de várias análises foi definida a forma de comparação com uma base de dados já pré-definida.

Outra grande dificuldade foi achar um meio de que a Memória RAM e o CPU obtivessem mais impacto durante um ataque de (D)DoS, para isso foram utilizadas e analisadas outras ferramentas para efetuar o ataque, mas devido a familiaridade e facilidade com a ferramenta *slowloris*, ficou definido o seu uso e as medições de Memória RAM e CPU apenas serviriam como mensuração desses ataques no âmbito de estudo desse trabalho.

4 CONCLUSÃO

O objetivo da pesquisa se deu no desenvolvimento de uma ferramenta para medir e identificar um dos maiores tipos de ataques cibernéticos existente, o de Negação de Serviço. Através de 3 métricas pré-definas (Memória RAM, CPU e pacotes da rede) para a análise e mensuração desse tipo de ataque, foi possível a obtenção de um estudo prático e teórico sobre os temas que envolviam a problemática.

Com o desenvolvimento da aplicação os resultados esperados para obter-se uma mensuração e uma possível identificação de um ataque de Negação de Serviço foram alcançados através do instrumento que levou a pesquisa e o desenvolvimento, a ferramenta *Denial Capture*. Através disso podemos afirmar que a ferramenta se trata de um IDS, pois faz análises e detecções de ataques.

Com isso o resultado desse trabalho se deu além da investigação e análise dados, mas na obtenção final de um artefato que justificasse o tema de estudo e os objetivos que eram desejados para a sua conclusão.

A idéia é que a ferramenta *Denial Capture* possa auxilar administradores de redes e profissionais de segurança da informação a identificar e medir impacto de ataques *(D)DoS* nos serviços e/ou servidores, para isso são propostos como trabalhos futuros: (1) identificação e mensuração de ataques que afetem com mais impacto os componentes físicos de uma máquina como Memória RAM, CPU ou até mesmo Disco Rigido; (2) propor uma melhor forma de identificar mais precisamente um ataque de negação de serviço pela rede, já que para grandes servidores ou sites da web essa solução pode não ser 100% eficaz pois a grande "massa" de pacotes podem ser confundidas com ataques; (3) propor outras formas de coleta dos dados das métricas escolhidas, como por exemplo alterações do tempo dos *timestamps* de coleta na duração total das medições; (4) analisar e implementar uma melhor forma para simular um cenário "normal", com softwares específicos que realizem essas simulações no trafego da rede.

REFERÊNCIAS

AMORIN, R. D. M. **Ataques Denial-of-Service**. Recife, 2007. 12p. - Centro de Informática, Universidade Federal de Pernambuco.

CANTÚ, E. **Redes de Computadores e a Internet**, 2003. CEFET - São José/SC

ESPÍRITO SANTO, A. F. S. **SEGURANÇA DA INFORMAÇÃO**. - Departamento de Ciência da Computação, Instituto Cuiabano de Educação (ICE).

EVANGELISTA, S. V. B. **SISTEMAS DE DETECÇÃO DE INTRUSOS E SISTEMAS DE PREVENÇÃO DE INTRUSOS: PRINCÍPIOS E APLICAÇÃO DE ENTROPIA**. Petrópolis, 2008. 74p. Monografia (Tecnólogo em Tecnologia da Informação e da Comunicação) - Instituto Superior De Tecnologia em Ciência Da Computação.

GIL, A. C. **Como elaborar projetos de pesquisa.** 4. ed. São Paulo: Atlas, 2008.

HARINATH, D. **OSI Reference Model – A Seven Layered Architecture of OSI Model.** Hyderabad - India. 9p. Department of Computer Science, Osmania University.

MARCONI, M. A; LAKATOS, E.V. **Fundamentos de Metodologia Científica.** 5 ed. São Paulo: Atlas, 2003.

MARIANO, I. S. **IPSec e DDoS, ASPECTOS DE SEGURANÇA EM REDES TCP/IP**. Rio de Janeiro, 2001. 78p. Monografia (Mestrado em Engenharia de Sistemas) - Universidade Federal do Rio de Janeiro.

NAKAMURA, E. T. e GEUS, P. L. Introdução. In:___**Segurança de Redes em Ambientes Cooperativos**. 1ed. São Paulo: Novatec, 2007. p. 33-25.

PROLEXIC. **Prolexic Quartely Global DDoS Attacks Report Q2 2014**. Net, Estados Unido, 2014. Seção Attacks Reports. Disponível em: <http://www.prolexic.com/kcresources/attack-report/attack_report_q214/Prolexic-Q22014-Global-Attack-Report-A4.pdf> Acesso em: 29 nov. 2014.

RAO, S. R. S. **Denial-of-service attacks and mitigation techniques: Real time implementation with detailed analysis**. **Net**, Reino Unido, 2011. Seção Reading Room. Disponível em: <http://www.sans.org/reading-room/whitepapers/detection/denial-service-attacks-mitigation-techniques-real-time-implementation-detailed-analysi-33764> Acesso em: 29 nov. 2014.

ROZENBLUM, D. **Understanding Intrusion Detection Systems. Net**, Reino Unido, 2011. Seção Reading Room. Disponível em: <http://www.sans.org/reading-room/whitepapers/detection/understanding-intrusion-detection-systems-337> Acesso em: 27 nov. 2014.

YANG, G. **Introduction to TCP/IP Network Attacks.** Iowa - USA, 1997. 22p. - Department of Computer Science, Iowa State University.