Pós-Graduação em Ciência da Computação

Ronierison de Souza Maciel

Avaliação do impacto de ataques DDoS e Malware: Uma abordagem baseada em Árvore de Ataque

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

Recife
2018

Ronierison de Souza Maciel

Avaliação do impacto de ataques DDoS e Malware: Uma abordagem baseada em Árvore de Ataque

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: avaliação de desempenho e dependabilidade
Orientador: Dr. Paulo Romero Martins Maciel
Coorientador: Dr. Paulo Roberto Freire Cunha

Recife
2018

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

M152a Maciel, Ronierison de Souza
Avaliação do impacto de ataques DDoS e Malware: uma abordagem baseada em árvore de ataque / Ronierison de Souza Maciel. – 2018.
90 f.: il., fig., tab.

Orientador: Paulo Romero Martins Maciel.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2018.
Inclui referências e apêndices.

1. Ciência da computação. 2. Segurança. I. Maciel, Paulo Romero Martins (orientador). II. Título.

004 CDD (23. ed.) UFPE- MEI 2018-145

Dissertação de Mestrado apresentada por **Ronierison de Souza Maciel** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "**Avaliação do impacto de ataques DDoS e Malware: Uma abordagem baseada em Árvore de Ataque**" **Orientador: Dr. Paulo Romero Martins Maciel** e aprovada pela Banca Examinadora formada pelos professores:

Prof. Dr. Stênio Flávio de Lacerda Fernandes
Centro de Informática / UFPE

Prof. Dr. Rubens de Souza Matos Júnior
Coordenadoria de Informática / IFS

Prof. **Orientador**: Dr. Paulo Romero Martins Maciel
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 31 de Agosto de 2018.

Prof. Aluizio Fausto Ribeiro Araújo
Coordenadora da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Eu dedico esta dissertação a Deus, minha família, minha noiva, amigos e professores que me deram o apoio necessário para alcançar meus objetivos.

AGRADECIMENTOS

Antes de mais nada, gostaria de agradecer ao bom Deus por ter me dado os instrumentos necessários para chegar até aqui, e por toda a persistência para não desistir diante das dificuldades encontradas no caminho.

Agradeço ao Prof. Paulo Maciel, que acreditou em meu potencial, me orientou durante toda essa jornada e ajudou em meu crescimento pessoal e profissional, ao qual tenho uma eterna dívida de gratidão.

Agradecer também à toda a minha família, que sempre estiveram comigo. Em especial a minha mãe Rosa de Lourdes e ao meu pai Manoel da Paz Maciel, amo plenamente, presente em todos os momentos da minha vida, por me ensinarem a lutar e nunca desistir dos meus objetivos.

Gostaria de agradecer também a minha noiva Melka Monique, mulher amável que esteve ao meu lado em todos os momentos e me auxiliou em todos os passos dessa jornada. Também gostaria de agradecer ao meu irmão Rener Ferraz, pelo compartilhamento de conhecimento, exemplo de perseverança e muito obrigado por toda ajuda que tens me proporcionado, agradeço também ao meu irmão, Rômulo Wallace, pelo companheirismo e ensinamentos, que levarei para toda uma vida.

Meu muito obrigado a todos os companheiros do grupo de pesquisa MoDCS. Em especial a Jean Araujo, Carlos Melo, Jamilson Dantas, Renata Pedrosa, Rubens Matos, João Ferreira e Erico Guedes, amigos para todos os desafios, pelas valiosas dicas e por estarem sempre dispostos a ajudar. E a todos os colegas que me ajudaram durante essa jornada: Danilo Mendonça, Breno Vasconcelos, Carlos Julian, Paulo Roberto, Príscila Lima, Bruno Santos, Thiago Pinheiro, Débora Souza e ao professor Terry Ingoldsby por disponibilizar a ferramenta da Amenaza a Attack Tree, de forma gratuita. De modo geral a todos membros do grupo MoDCS.

Também sou muito grato a meu irmão Josivan Martins, por me acolher em sua residência e muito obrigado pelos conselhos. Obrigado por tudo.

Enfim, aos que amo, que de forma direta ou indireta contribuíram para essa conquista. A todos, meu sincero agradecimento.

"Coincidências ruins são as cicatrizes das batalhas da vida."

(MAZUR, 2016)

RESUMO

Ataques *Distributed Denial-of-Service* (DDoS) Negação de Serviço Distribuído podem ocorrer a qualquer momento, lugar e, geralmente, ocorrem com pouco ou nenhum aviso prévio. Neste ínterim, ressalta-se que muitas empresas, de pequeno e médio porte não estão preparadas para lidar com interrupções significativas em seus sistemas. Na tentativa de prevenir ataques maliciosos e, por vezes, danosos, as organizações devem investir em largura de banda de Internet, infraestruturas redundantes, *backups* regulares e *firewall*, visando o monitoramento de ameaças, além de outros mecanismos proativos e reativos, caso contrário, o serviço poderá ser interrompido, aumentando as chances de perdas financeiras. Técnicas de modelagem podem ajudar os administradores de sistemas a avaliar protocolos de rede e identificar quais são os mais utilizados em ataques DDoS. Modelos de Árvore de Ataque permitem a avaliação e apresentação de resultados que podem ser usados para mitigar as portas de entrada para ataques. Esta dissertação visa representar o comportamento de ataques DDoS e *malware*, a fim de avaliar o impacto deles para a dependabilidade e mais especificamente, para a confidencialidade e a integridade de sistemas. Dois estudos de caso são apresentados para aplicação da solução proposta. O primeiro estudo de caso expõe uma avaliação das principais vulnerabilidades, relativas aos ataques DDoS nos serviços providos por sistemas computacionais. A avaliação tem por objetivo buscar identificar o impacto na vítima, mediante a ocorrência de um ataque DDoS. O segundo estudo de caso avalia um método de ciberataque, denominado de *malware*, onde se considerou as técnicas mais adotadas por este tipo de ataque. Desta forma, este trabalho possibilita aos analistas de redes, desenvolver meios para prevenção de ataques DDoS e malware e planejar contramedidas para mitigar o impacto dos ataques.

Palavras-chaves: Segurança. Ataque de negação de serviço distribuído. Árvore de Ataque. Ameaças. Modelagem.

ABSTRACT

DDoS attacks can occur anytime, anywhere, and usually occur with little or no advance notice. In the meantime, it should be noted that many small and medium-sized companies are not prepared to deal with significant outages in their systems. In an attempt to prevent malicious and sometimes harmful attacks, organizations should invest in Internet bandwidth, redundant infrastructures, regular backups, and texting to guard against threats, as well as other mechanisms proactive and reactive, otherwise the service may be disrupted, increasing the chances of financial losses. Modeling techniques can help system administrators evaluate network protocols and identify which ones are most commonly used in DDoS attacks. Attack Tree models allow the evaluation and presentation of results that can be used to mitigate attack ports. This dissertation aims to represent the behavior of DDoS and malware attacks, in order to assess their impact on dependability, and more specifically on system confidentiality and integrity. Two case studies are presented for the application of the proposed solution. The first case study presents an assessment of the main vulnerabilities related to attacks on services provided by computer systems. The objective of the evaluation is to identify the impact on the victim by the occurrence of a DDoS attack. The second case study evaluates a method of cyberattack, called malware, where it was considered the techniques most adopted by this type of attack. In this way, this work enables network analysts to develop means to prevent DDoS attacks and malware and to plan countermeasures to mitigate the impact of attacks.

Key-words: Security. Distributed Denial-of-Service. Malware. Attack Tree. Threats. Modeling.

LISTA DE ILUSTRAÇÕES

Figura 1 – Classificação dos ataques DDoS 21
Figura 2 – Árvore de Dependabilidade (baseado em (AVIŽIENIS et al., 2001)) 30
Figura 3 – Principais componentes de uma árvore de ataque 35
Figura 4 – Exemplo de árvore de lógica de ameaça para o sistema UNIX (WEISS, 1991) 36
Figura 5 – Árvore de Ataque: Arquitetura Básica (SCHNEIER, 1999) 37
Figura 6 – Habilidade do atacante de acordo com a probabilidade de sucesso do ataque e seu respectivo custo 39
Figura 7 – Perda de reputação e Visibilidade da vítima 40
Figura 8 – Habilidades 40
Figura 9 – Metodologia Adotada 44
Figura 10 – Indicadores do comportamento da ameaça 45
Figura 11 – Perfil de custo do ataque 46
Figura 12 – Ataque DDoS e *Malware* 48
Figura 13 – Modelo de árvore de ataque para o cenário DDoS 49
Figura 14 – Ataque DDoS e *malware* 52
Figura 15 – Árvore de Ataque para o cenário DDoS e *malware* 53
Figura 16 – Benefício com o ataque 58
Figura 17 – Facilidade do ataque 58
Figura 18 – *Percepção de Dano* 59
Figura 19 – Propensão do ataque 59
Figura 20 – Habilidade técnica do atacante 60
Figura 21 – Benefícios do ataque 62
Figura 22 – Facilidade do ataque 63
Figura 23 – *pain factor* do ataque 63
Figura 24 – Propensão do ataque 64
Figura 25 – Habilidade técnica do atacante 64
Figura 26 – Mecanismo de busca do Shodan 78
Figura 27 – Página inicial da ferramenta *Shodan* 80
Figura 28 – Página de *download* 81
Figura 29 – Lista de cidades no Brasil com protocolo *Network Time Protocol* (NTP) 82
Figura 30 – Lista de cidades no Brasil com câmeras Avtech 82
Figura 31 – Lista de roteadores TP-LINK em todo o mundo 83

LISTA DE TABELAS

Tabela 1 – Trabalhos Relacionados 19
Tabela 2 – Capacidade dos *Malware* DDoS 28
Tabela 3 – Habilidade técnica - Atacante 38
Tabela 4 – Perda de reputação - Vítima 39
Tabela 5 – Parâmetros de entrada referente a um ataque DDoS 57
Tabela 6 – Parâmetros de entrada referente a um ataque DDoS e *Malware* 61
Tabela 7 – Descrição dos métodos de busca com *Shodan* 81
Tabela 8 – Lista de portas escaneadas pelo Shodan 84

LISTA DE ABREVIATURAS E SIGLAS

ACK	*Acknowledgment*
ARM	*Advanced RISC Machine*
BA	Benefício do Ataque
Bot	*Robot*
Botnet	*Robot Network*
CA	Custo do Ataque
CERT	*Computer Emergency Readiness Team*
CGI	*Common Gateway Interface*
DDoS	*Distributed Denial-of-Service*
DNS	*Domain Name System*
DoD	*Department of Defense*
DoS	*Denial-of-Service*
EEM	*Extended Entropy Metric*
FA	Facilidade do Ataque
FEs	*Flash Event*
GRE	*Generic Routing Encapsulation*
HT	Habilidade Técnica
HTTP	*HyperText Transfer Protocol*
ICMP	*Internet Control Message Protocol*
IoT	*Internet of Things*
IP	*Internet Protocol*
IRC	*Internet Relay Chat*
MIPS	*Microprocessor without Interlocked Pipeline Stages*
MTTF	*Mean Time to Failure*
MTTR	*Mean Time to Repair*
NSA	*National Security Agency*
NTP	*Network Time Protocol*
PA	Probabilidade do Ataque

PD	Percepção de Dano
PowerPC	*Performance Optimization with Enhanced RISC – Performance Computing*
PPA	Propensão do Ataque
SDI	*Strategic Defense Initiative*
SLA	*Service Level Agreements*
SNMP	*Simple Network Management Protocol*
SPARC	*Scalable Processor Architecture*
SYN	*Synchronize*
TCP	*Transmission Control Protocol*
TI	Tecnologia da Informação
UDP	*User Datagram Protocol*
VI	Visibilidade

SUMÁRIO

Lista de tabelas **10**

1 INTRODUÇÃO **15**
1.1 OBJETIVOS 16
1.2 TRABALHOS RELACIONADOS 16

2 FUNDAMENTAÇÃO TEÓRICA **20**
2.1 *DISTRIBUTED DENIAL-OF-SERVICE (DDOS)* 20
2.1.1 Classificação do ataque DDoS **20**
2.1.2 Classificação por grau de automação **20**
2.1.3 Classificação por vulnerabilidade explorada **22**
2.1.4 Classificação por dinâmica na taxa do ataque **24**
2.1.5 Classificação por impacto **24**
2.1.6 Ataques *DDoS Protocol Exploit* **24**
2.1.7 Ataques *DDoS Flooding* **25**
2.1.8 Ataques *DDoS Amplification* **25**
2.1.9 Ataques *Exploitation of software bug* **26**
2.1.10 Ataques *Validation input* **27**
2.2 *MALWARE* 27
2.2.1 Tipos de *Malware* **27**
2.3 DEPENDABILIDADE 30
2.3.1 Ameaças à Dependabilidade **31**
2.3.2 Meios para melhoria da Dependabilidade **31**
2.3.3 Atributos de Dependabilidade **32**
2.4 ÁRVORE DE ATAQUE 35

3 MODELOS **43**
3.1 METODOLOGIA PARA MODELAGEM E AVALIAÇÃO 43
3.1.1 Avaliação **43**
3.1.2 Compreensão das principais ameaças DDoS **44**
3.1.3 Definição do modelo **45**
3.1.4 Modelo **46**
3.2 SISTEMAS AVALIADOS 47
3.2.1 Ataque DDoS **47**
3.2.2 Ataque DDoS e *Malware* **52**

4 ESTUDO DE CASO **56**

4.1 ESTUDO DE CASO I - AVALIAÇÃO DE UM ATAQUE DDOS 56
4.1.1 Parâmetros de entrada 56
4.1.2 Resultado do modelo 56
4.2 ESTUDO DE CASO II - AVALIAÇÃO DE UM ATAQUE DDOS E *MALWARE* 60
4.2.1 Resultado do modelo 61
4.3 CONSIDERAÇÕES 65

5 CONCLUSÕES 66
5.1 LIMITAÇÕES E TRABALHOS FUTUROS 67

REFERÊNCIAS 69

APÊNDICE A – FERRAMENTA *SHODAN* 78

APÊNDICE B – BANNER SSL SHODAN 79

APÊNDICE C – UTILIZAÇÃO DA FERRAMENTA *SHODAN* PARA LOCALIZAR DISPOSITIVOS VULNERÁVEIS . . 80

APÊNDICE D – LISTA DE PORTAS 84

1 INTRODUÇÃO

As infraestruturas computacionais dependem de diversos mecanismos quanto à tolerância das falhas para lidar com imprecisões de *hardware* e *software*, oferecendo disponibilidade aos usuários e recursos acessíveis em qualquer lugar e a qualquer momento (WOOD; STANKOVIC, 2002). No entanto, muitos fatores podem fazer com que um sistema fique indisponível, a exemplo dos ataques DDoS (WOOD et al., 2016) e os *malware* (SIKORSKI; HONIG, 2012). Em 2018, a plataforma de desenvolvimento de *software* online GitHub enfrentou um evento relacionado à segurança (NEWMAN, 2018), sofrendo um ataque DDoS de 1,35 terabit por segundo, que deixou a plataforma inacessível por oito minutos. Outra situação que provocou alerta foi a informação divulgada por (IBRAGIMOV et al., 2018) afirmando que especialistas em segurança cibernética detectaram uma *Robot Network* (Botnet) formada a partir de 50 mil câmeras de vigilância no Japão, estavam sendo utilizadas para realizar ataques DDoS a nível mundial.

Os métodos de ataque DDoS surgiram em meados de 1997 (CENTER, 1998), sendo ataques *flooding*, amplificação, exploração de protocolo e aplicação atualmente mais utilizados. Os ataques DDoS descaracterizam o acesso de usuários legítimos, resultando em diversos problemas que impactam a vítima, como perda de clientes, credibilidade, problemas financeiros, entre outros problemas que resultam de um ambiente computacional vulnerável.

As ameaças virtuais têm se tornado mais sofisticadas e provocando problemas cada vez mais inconvenientes aos usuários. Uma dessas ameaças está sendo representada por um novo tipo de *malware*, o *Hide-n-Seek*, atualmente o primeiro *Robot* (Bot) a suportar a reinicialização do dispositivo em que ele havia se estabelecido (BOTEZATU, 2018).

Ainda assim, estima-se que o custo médio de um ataque *malware* a uma empresa é de US $2,4 milhões, sendo o setor de saúde a indústria com o maior número de ataques por *ransonware* (ACCENTURE, 2017). Existe uma previsão de que os ataques irão se quadruplicar até 2020 (MORGAN, 2017), presume-se, ainda, que apenas 7% das empresas no mundo possam monitorar, detectar e prevenir *malware* (ISACA, 2017), as 93% restantes estão sujeitas a ataques por ameaças virtuais. Esses tipos de atividades maliciosas resultam em perdas para inúmeras empresas, gerenciamento de *data centers*, sistemas de saúde e sistemas operacionais de tempo real. Logo, os sistemas devem ser avaliados visando mitigar os impactos de possíveis falhas em sua confiabilidade.

Alguns autores utilizam árvores de ataque para solucionar esses problemas, como (ROY, 2010) que propôs técnicas de custo de ataque e de probabilidade de ocorrência para avaliar um sistema SCADA (VERBA; MILVICH, 2008). (MAUW; OOSTDIJK, 2005) mostraram uma semântica denotativa das árvores de ataque e suas características. (SCHNEIER, 1999) traz aspectos que caracterizam as árvores de ataque, como custo de ataque e probabilidade de

ocorrência. (EDGE, 2007) propôs um *framework* para modelagem, análise e mitigação de ameaças utilizando árvores de ataque. No entanto, com base em ataques DDoS, poucos estudos apresentam aspectos de como examinar várias ameaças e seus impactos nos serviços providos por sistemas computacionais. Ainda assim, a maioria dos trabalhos disponíveis na literatura não possui ampla cobertura das principais técnicas de ataque.

Em síntese, o trabalho realizado nesta dissertação propõe modelos destinados a avaliar o impacto de ataques DDoS, *malware* e atributos da dependabilidade, limitados a confidencialidade e integridade, utilizando árvore de ataque. A avaliação é dada através de um sistema sob ataque DDoS em seguida se avalia um sistema sob ataques DDoS e *malware* (ver Apêndice A). A avaliação tem por objetivo mostrar as ameaças que causam um maior impacto nos serviços providos por sistemas computacionais.

1.1 OBJETIVOS

O objetivo geral desta pesquisa é: Avaliar o impacto de ataques DDoS propondo modelos para prever ameaças de natureza criminosa nos serviços providos por sistemas computacionais e auxiliando na prevenção de ataques DDoS, a fim de alcançar objetivo geral, os seguintes objetivos específicos são detalhados:

- Elaborar modelos que representem os modos operacionais de um ciberataque;
- Identificar técnicas para reconhecer as vulnerabilidades que causam maior impacto no provimento de serviços;
- Avaliar o impacto dos ciberataques.

1.2 TRABALHOS RELACIONADOS

As pesquisas relacionadas (ver Tabela 1) com a área de interesse dessa dissertação abrangem temas sobre impacto de ataques DDoS, *malware*, custo do ataque, probabilidade do ataque, benefício do ataque, facilidade do ataque, *pain factor*, propensão e habilidade técnica, e modelos de árvore de ataque.

O trabalho desenvolvido por (WANG et al., 2010) discute técnicas utilizadas em ataques DDoS e lista alguns dos métodos de defesa contra-ataques DDoS. As técnicas de ataques *Denial-of-Service* (DoS) discutidas são, *Internet Control Message Protocol* (ICMP) *flood*, *Transmission Control Protocol* (TCP) *Synchronize* (SYN) *flood* e *User Datagram Protocol* (UDP) *flood*. O objetivo principal do trabalho foi a proposição de um algorítimo de detecção de ataques DDoS. A estratégia proposta captura os incidentes através de ataques DDoS. Não tratam ataques de amplificação como também não mostram as vulnerabilidades atualmente exploradas.

(BERTINO; ISLAM, 2017) tratam sobre os riscos de vulnerabilidades em ambientes de *Internet of Things* (IoT) e sugerem a adoção de práticas para evitar tais ameaças. Apresentam uma lista de práticas que auxiliam na diminuição e proliferação de Botnet. Apesar disso os autores não tratam problemas relativos a outras Botnet, somente a *Mirai*. Poderiam elaborar uma comparação entre Botnet.

(WANG et al., 2017) evidenciam as Botnet como uma das principais atividades maliciosas, como roubo de informações, *phishing*, *spam* e DDoS. Os autores propõem detectar Botnet a partir de um algoritmo gerador de domínio através de um único padrão de ataques DDoS. No entanto, ataques DDoS não seguem um padrão, pois a taxa de ataque pode oscilar e a Botnet pode ter um controlador *handler* e ou *Internet Relay Chat* (IRC).

O trabalho produzido por (BORATEN; KODI, 2018) aborda aspectos disjuntos do ataque DDoS, ou seja, a preocupação do aumento dos agentes maliciosos que ameaçam infringir a confiança do *hardware*, implantando *trojans* nos mesmos. É proposto um novo modelo de carga útil sequencial de *hardware trojans*, ativado, ele executa a inspeção de pacotes e injeta falhas para criar um novo tipo de ataque DDoS. As falhas injetadas são utilizadas para acionar uma resposta dos esquemas de código de correção de erros e fazer com que a retransmissão repetida prejudique os recursos de rede e crie bloqueios capazes de processar aplicativos únicos para falhas completas de chips. Para contornar a ameaça dos *hardware trojans*, foi proposto um modelo heurístico de detecção de ameaças para classificar falhas e descobrir *hardware trojans* dentro de links comprometidos. Evidenciaram aspectos relativos a ofuscadores de código, como também pode ser utilizadas esteganografia.

(MACIEL et al., 2018) elaboram uma modelagem utilizando árvore de ataque para obter resultados acerca do impacto causado por DDoS em sistemas computacionais. A abordagem utilizada mostra as vulnerabilidades exploradas para amplificar ataques DDoS. Todavia, o autor faz uma análise das probabilidades de ocorrência dessas vulnerabilidades, benefícios que o atacante pode angariar com o ataque, os custos com o ataque, entre outras métricas, como, *pain factor*, métrica associada a danos a vítima, como, perdas financeiras, reputação e dentre outras. O trabalho avalia o impacto causado pelas vulnerabilidades.

Em (EPOH, 2018) é realizado um método de análise de ataques DDoS em setores públicos e privados, podendo considerar a diminuição dos danos causados pelos ataques DDoS, através de agentes infiltrados para descobrir pessoas que tenham propensão para efetuar esse tipo de ataque dentro da empresa, onde foi realizado uma captura de tráfego da rede, no intuito de verificar anomalias na rede. Porém, a necessidade de um funcionário ser o responsável pelo ataque DDoS é algo trivial, pois o mesmo pode repassar dados de acesso sensíveis, relativo ao servidor da empresa. Tratando-se de grandes empresas, a missão de detetive não funciona, visto que cada funcionário está interligado ao seu setor, ainda assim, é possível efetuar um ataque *flooding* internamente.

(KOTENKO; SAENKO; LAUTA, 2019) aborda a exploração dos tipos de ataque DDoS,

procurando monitorar e identificar as vulnerabilidades, através de modelagem analítica de ataques cibernéticos, o mesmo utiliza redes estocásticas. A validação se deu através da varredura da rede, identificando as vulnerabilidades. O principal objetivo do autor com a utilização das redes estocásticas está associado ao uso das métricas para encontrar o tempo de probabilidade dos ataques. Entretanto, o trabalho poderia ser valorizado se o autor abordasse outros meios de modelagem, dado que o mesmo já possui os valores probabilísticos associados aos ataques, o autor não aborda um sistema de contramedida.

Além disso, (BHUYAN; BHATTACHARYYA; KALITA, 2016) utilizaram a ideia de distância da informação entre diferentes fluxos amostrais, conforme originalmente proposto por (XIANG; LI; ZHOU, 2011). Os autores calculam uma métrica *Extended Entropy Metric* (EEM) baseada em entropia generalizada estendida usando *Internet Protocol* (IP) de origem e recursos de cabeçalho de pacote de taxa de pacote de entrada para detectar ataques HR-DDoS. Eles validam sua abordagem em relação aos conjuntos de dados do MIT Lincoln, CAIDA e TUIDS. No entanto, a abordagem proposta não considerou discriminar *Flash Event* (FEs) de ataques HR-DDoS, pois ambos os tipos de tráfego compartilham muitas características comportamentais semelhantes. Além disso, o sistema proposto sofre de limitações óbvias de implantação de final de vítima.

(XIAO et al., 2015) aplicaram a ideia de similaridade de fluxo entre os fluxos gerados pelo mesmo *software* de ataque malicioso e provavelmente estão correlacionados entre si. Eles usam um algoritmo de *k-nearest* mais próximos para agrupar os fluxos gerados a partir do mesmo código malicioso ou Bot. Seu esquema proposto produz alta taxa de classificação e baixo tempo de resposta. Eles validam seu esquema proposto usando o rastreamento de tráfego do data center e o conjunto de dados do KDD'99.

Desta forma, a importância desta dissertação é observada quando em comparação aos trabalhos relacionados, justamente por reunir uma abordagem baseada em modelos de árvore de ataque para avaliar o impacto de DDoS e *malware* em serviços providos por sistemas computacionais, de antemão, avaliamos um sistema sob ataques DDoS, em seguida um sistema sob ataques DDoS e *malware*. Cujo objetivo será verificar e comparar ambos, analisando as seguintes métricas: probabilidade do ataque, custo com o ataque, benefícios com o ataque, *pain factor*, propensão do ataque e habilidade técnica do atacante.

A dissertação está organizada da seguinte forma: o Capítulo 2 apresenta a fundamentação teórica sobre os temas abordados neste trabalho: *Distributed Denial-of-Service* (DDoS), *malware*, Dependabilidade e Árvore de Ataque, discutimos sobre ataques DDoS, em particular, sobre o conceito de *malware*, essencial neste trabalho; seguida pela abordagem sobre as técnicas de modelagem utilizadas para realizar a avaliação dos atributos de dependabilidade, integridade e confidencialidade. O Capítulo 3 apresenta a metodologia aplicada e abordando os sistemas avaliados. O Capítulo 4 apresenta os estudos de caso desta pesquisa, baseados nos modelos propostos. Por fim, O Capítulo 5 apresenta as conclusões, bem como os trabalhos futuros a serem desenvolvidos com base nessa pesquisa.

Tabela 1 – Trabalhos Relacionados

	Contexto	Modelo	Custo do Ataque	Probabilidade de Ataque	Benefício do Ataque	Facilidade do Ataque	Habilidade Técnica
Esta dissertação	DDoS e *Malware*	✓	✓	✓	✓	✓	✓
(WANG et al., 2010)	DDoS	✓					
(XIANG; LI; ZHOU, 2011)	DDoS	✓					
(XIAO et al., 2015)	DDoS *Data Center*	✓					
(BHUYAN; BHATTACHARYYA; KALITA, 2016)	DDoS *flooding*	✓	✓	✓			
(BORATEN; KODI, 2018)	DDoS e *trojan*	✓					
(BERTINO; ISLAM, 2017)	Botnet e IoT	✓		✓			
(WANG et al., 2017)	DDoS			✓			
(EPOH, 2018)	DDoS			✓			
(MACIEL et al., 2018)	DDoS	✓	✓	✓	✓	✓	✓
(KOTENKO; SAENKO; LAUTA, 2019)	DDoS	✓			✓		

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos básicos das principais áreas abordadas nesta dissertação: *Distributed Denial-of-Service* (DDoS), *malware*, Dependabilidade, e Árvore de ataque. O referencial teórico aqui apresentado deve fornecer o conhecimento necessário para uma compreensão clara dos capítulos à frente.

2.1 *DISTRIBUTED DENIAL-OF-SERVICE (DDOS)*

A *Computer Emergency Readiness Team* (CERT) Equipe de Prontidão para Emergência de Computadores define DoS como, técnica pela qual um atacante emprega um equipamento conectado à rede para impossibilitar a operacionalidade de serviço, computador ou uma rede conectada à *Internet*. Quando utilizada de forma coordenada e distribuída, ou seja, quando um conjunto de equipamentos são utilizados no ataque, recebe o nome de DDoS (CERT, 2016). Os principais objetivos dos ataques DDoS são o consumo de recursos e a utilização da largura de banda, de tal forma que a vítima não pode fornecer serviços a usuários legítimos. Os invasores usufruem de uma arquitetura diferente, como *handlers*, Botnet baseado em *Internet Relay Chat* (IRC) e na *Web* para gerar ataques de DDoS (HOQUE; KASHYAP; BHATTACHARYYA, 2017).

Os ataques DDoS são executados na camada de aplicação, transporte, rede e física da estrutura TCP/IP, explorando o uso dos seus respectivos protocolos, como ICMP, TCP, *HTTP* e UDP (SINGH; DE, 2017). Desta forma, vamos abordar os ataques que utilizam a vulnerabilidade desses protocolos para praticar ataques de negação de serviço. De acordo com (MIRKOVIC; REIHER, 2004), a taxonomia dos ataques DDoS pode ser classificada conforme a Figura 1, cujos detalhes serão abordados a seguir.

2.1.1 Classificação do ataque DDoS

Como podemos observar, há quatro ramificações que envolvem a taxonomia de um ataque DDoS, a classificação dá-se da seguinte maneira. Classificação por grau de automação, por vulnerabilidade explorada, dinâmica na taxa do ataque e impacto do mesmo.

2.1.2 Classificação por grau de automação

Com base no grau de automação de um ataque DDoS pode ser classificado em ataque Manual, Semiautomático ou Automático.

Ataques manuais: o invasor faz uma análise manualmente das vulnerabilidades nos computadores de forma remota, instalando códigos maliciosos, em seguida, executa

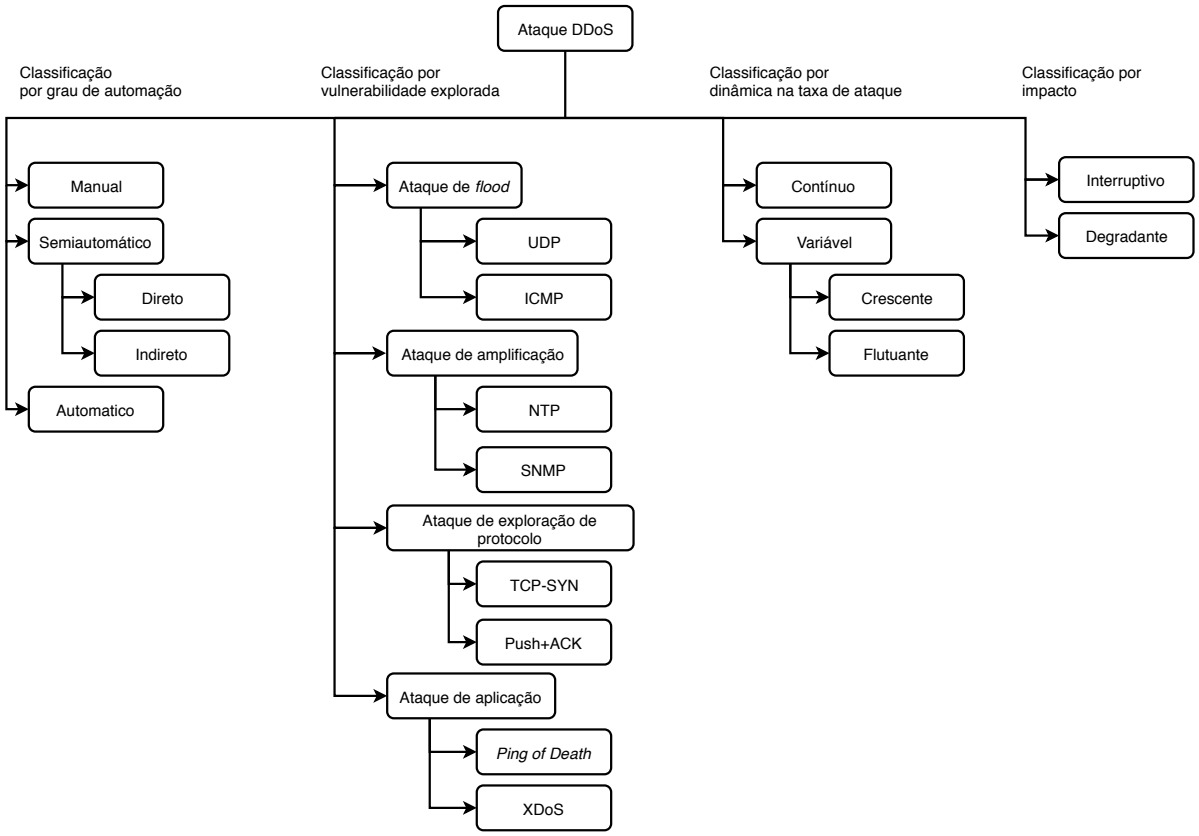


Figura 1 – Classificação dos ataques DDoS

comandos para efetuar o ataque. Somente os primeiros ataques DDoS pertenciam à categoria manual. Ao longo do tempo, todas as etapas se tornaram automatizadas.

Ataques Semiautomáticos são ataques que seguem o modelo agente-*handlers* [1]. O invasor executa uma busca à procura de *handlers* e agentes, para isso, o mesmo utiliza *scripts* maliciosos automatizados. Algumas variáveis como, o tipo de ataque, endereço da vítima e o início do ataque são especificados pelo *handler*.

Todavia, os ataques semiautomáticos, são divididos em via comunicação direta e indireta. Em ataques direto, o agente utiliza *handlers* que necessitam reconhecer a identidade um do outro, para que haja a comunicação entre ambos.

Essa comunicação é obtida através da codificação do endereço IP das máquinas *handlers*, via *scripts*, instalados nas máquinas do agente. Cada agente informa sua disponibilidade aos *handlers* que por sua vez, armazena o endereço IP para comunicações posteriores.

A desvantagem nesse método de ataque é dada pela descoberta da máquina comprometida, com isso, toda rede DDoS é evidenciada, considerando que os *handlers* e agentes escutam as conexões da rede, e são identificáveis pelos *scanners* de rede.

[1] São *hosts* comprometidos, que dispõem de um *software* especial em execução, capaz de controlar vários agentes (SPECHT; LEE, 2004).

Ataques de comunicação indireta utilizam serviços de comunicação legítima para que o agente sincronize suas ações (envio do comando para o ataque), podemos citar, por exemplo, ataques que utilizam os canais IRC.

Através da utilização dos canais IRC a função do *handler* é substituída, visto que canais IRC dispõe do anonimato concebendo aos agentes ataques DDoS sem rastreio. Considerando que o controle por parte do agente a um canal IRC pode ser análogo ao tráfego de bate papo, além do mais, o atacante pode efetuar saltos de um canal para outro, evitando ser descoberto (MIRKOVIC; REIHER, 2004).

Ataques Automáticos consistem na automatização que vai desde a fase da infecção até a execução do ataque, evitando a necessidade de comunicação mais de uma vez. O início do ataque, tipo do ataque, duração e vítima, são etapas pré-programadas no *script* do ataque, esse tipo de mecanismo oferece a mínima exposição do agente a possíveis descobertas do mesmo. Através de um único comando, todo processo será automatizado. No entanto, para esse método de ataque, o mecanismo de disseminação geralmente deixa o *backdoor* [2] das máquinas comprometidas sob fácil alteração no *script* do mesmo. Ainda assim, a maioria dos agentes, utilizam canais IRC para modificar os *scripts* existentes (POWER, 2002).

2.1.3 Classificação por vulnerabilidade explorada

Os ataques DDoS, quando classificados por vulnerabilidade explorada, são divididos em: Ataques por inundação (do inglês *flood*), Ataques de amplificação, Ataques de exploração de protocolo e Ataques de aplicação.

Ataques *Flood*: são ataques onde zumbis [3] enviam um grande volume de tráfego IP para o sistema da vítima, a fim de congestionar sua largura de banda. O impacto por esse tipo de ataque varia entre oscilante e ampliado, podendo deixar todo o sistema da vítima inoperante. Alguns dos ataques de *flood* mais conhecidos são, *flood* UDP e *flood* ICMP. Onde, os ataques *flood* UDP são caracterizados pelo envio de um extenso número de pacotes UDP ao sistema da vítima, resultando na saturação da rede e esgotamento da largura de banda disponível para solicitações de usuários legítimos.

Normalmente, os ataques *flood* UDP são elaborados para atacar as vítimas através de portas aleatórias (ver Apêndice D), o sistema da vítima recebe um pacote UDP será determinado qual aplicativo está aguardando na porta de destino.

[2] É um *script* malicioso com o único e exclusivo objetivo de oferecer acesso ao computador da vítima de forma remota (SZOR, 2005).

[3] São computadores infectados por algum tipo de código malicioso, onde o invasor consegue acessa-lo de forma remota e disseminar *spam* e vírus sem que seja descoberto. Geralmente, as Botnet como são chamadas, exercem o papel de atacante (DDoS) contra uma determinada vítima, sem que seu usuário legítimo saiba de tais atividades (COOKE; JAHANIAN; MCPHERSON, 2005).

Quando o mesmo percebe que não há nenhum aplicativo aguardando na porta, será gerado um pacote ICMP de destino inacessível (LONG; THOMAS, 2001) para o endereço de origem falsificado.

Se diversos pacotes UDP forem entregues as portas das vítimas, o sistema da mesma ficará inativo, com o auxílio de ferramentas DDoS (SPECHT; LEE, 2004) o endereço de IP de origem pode ser falsificado, com isso, a verdadeira identidade do atacante será impedida de ser exposta e os pacotes de retorno do sistema da vítima não são enviados de volta para os zumbis.

Nos ataques *flood* utilizando ICMP, o atacante envia uma vasta quantidade de pacotes com *ICMP_ECHO_REQUEST* (*"Ping"*) [4] para a vítima, assim é solicitado ao computador da mesma uma reposta, com isso ocorrendo inúmeras vezes, haverá uma saturação da largura de banda da vítima (CRISCUOLO, 2000).

Ataques de Amplificação normalmente utilizam *spoofing* [5] e o endereçamento de *broadcast* para amplificar seu ataque. De antemão, um sistema de amplificação deve ser encontrado, quer dizer, uma rede que permita comunicação com o endereço de *broadcast*, e disponha de um número elevado de *hosts* ativos.

Em seguida, o invasor envia amplos pacotes de solicitações de *echo* ICMP ao endereço de *broadcast* da rede de amplificação, com um endereço de origem que sofreu *spoofing* do sistema da vítima. O amplificador transmitirá os pacotes para todos os *hosts* da rede de amplificação e em seguida enviará pacotes ICMP *echo reply* condizentes ao endereço que sofreu *spoofing*. Essas técnicas são comumente conhecidas como ataques *smurf*, *fraggle* (ERICKSON, 2008), *BitTorrent*, NTP (Network Time Protocol) e *Simple Network Management Protocol* (SNMP) (ADAMSKY et al., 2015).

Ataques de exploração de protocolo exploram um recurso específico ou um *bug* de implementação, de algum protocolo instalado na vítima para consumir seus recursos. Um exemplo, a exploração de protocolo TCP utilizando SYN *Acknowledgment* (ACK).

Os ataques TCP SYN exploram o *handshake* de três vias. Um servidor, ao receber uma solicitação SYN inicial de um cliente, envia de volta um pacote SYN/ACK (sincronizar/confirmar) e espera que o cliente envie o ACK final (confirmação). Um atacante inicia um ataque de inundação SYN enviando um extenso número de pacotes SYN e nunca reconhece nenhuma das respostas, deixando o servidor aguardando as ACK inexistentes (BELLOVIN, 1989). Considerando que o servidor possui apenas uma fila de *buffer* limitada para novas conexões, o SYN *Flood* faz

[4] O ping é um *software* utilitário de rede para testar se um determinado *host* está acessível (NG; ZHANG, 2002).

[5] Ataque de falsificação, onde o invasor cria um contexto falso a fim de induzir a vítima a tomar uma decisão inadequada relevante à segurança (FELTEN et al., 1997).

com que o servidor não consiga processar outras conexões de entrada à medida que a fila fica sobrecarregada (SCHUBA et al., 1997). Outros exemplos de ataques de exploração de protocolo são ataques *Push*+ACK, ataques de solicitação *Common Gateway Interface* (CGI) e ataques de servidor de autenticação.

Ataques de Aplicação visam uma determinada aplicação no *host* da vítima, desabilitando assim o uso legítimo do cliente dessa aplicação. Se os recursos compartilhados da máquina *host* não forem completamente consumidos, outras aplicações e serviços ainda devem estar acessíveis aos usuários. Por exemplo, uma assinatura falsa de ataque em um servidor de autenticação vincula recursos do aplicativo de verificação de assinatura, mas a máquina de destino ainda responderá a solicitações *echo* do ICMP, e outros aplicativos que não exigem acesso autenticado ainda deverão funcionar.

2.1.4 Classificação por dinâmica na taxa do ataque

Dependendo da dinâmica na taxa do ataque, os ataques DDoS podem ser divididos em ataques de taxa contínua e taxa variável.

Contínuo: são ataques que não sofrem fragmentação ou decréscimo na taxa do ataque.

Variável ataques de taxa variável, como seu nome indica, portanto, evitam a detecção e a resposta imediata. Com base no mecanismo de mudança de taxa, distinguimos ataques com taxa crescente e flutuante. O aumento dos ataques de taxa gradualmente levará a exaustão dos recursos da vítima, atrasando a detecção do ataque. Ataques de taxa flutuante possui uma ondulação que é definida pelo comportamento da vítima, às vezes, diminuindo a taxa para evitar a detecção.

2.1.5 Classificação por impacto

Com base no impacto de um ataque DDoS, podemos dividi-los em dois tipos de ataques: interruptivos e degradantes.

Interruptivo: levam à negação completa do serviço da vítima.

Degradantes consome parte dos recursos da vítima, tendo como efeito o atraso da detecção do ataque.

2.1.6 Ataques *DDoS Protocol Exploit*

***TCP-SYN*,** o atacante envia uma sucessão de requisições SYN para o sistema da vítima visando uma sobrecarga na camada de transporte (EDDY, 2007).

***Ping of Death*,** o invasor tenta interromper uma máquina de destino enviando um pacote maior que o tamanho máximo permitido, causando o congelamento ou a falha da máquina de destino (KIM et al., 2004).

2.1.7 Ataques *DDoS Flooding*

UDP, é um tipo de ataque em que um volumoso número de pacotes UDP é enviado para um servidor de destino com o objetivo de sobrecarregar a capacidade do dispositivo de processar e responder (HUSSAIN; BEIGH, 2013).

ICMP, tem por objetivo sobrecarregar um dispositivo de destino com pacotes de solicitação de ICMP *echo-request*, fazendo com que o destino fique inacessível ao tráfego normal (CONTA; DEERING; GUPTA, 2006).

Slowloris, é uma ferramenta de ataque de DoS que permite que uma única máquina pare o servidor *Web* de outra máquina com largura de banda mínima e em serviços e portas não relacionados (DAMON et al., 2012).

Hping3, ferramenta de rede capaz de enviar pacotes *TCP/IP* personalizados e exibir respostas de destino, como o programa *ping*, com respostas de ICMP. O *hping3* manipula a fragmentação, o tamanho e o corpo dos pacotes arbitrários (MOYERS et al., 2010).

2.1.8 Ataques *DDoS Amplification*

***XDoS*,** ataque de negação de serviço *XML* (ataque *XDoS*) é um ataque de negação de serviço de conteúdo cujo objetivo é encerrar um serviço da *Web* ou sistema executando esse serviço. Um ataque *XDoS* comum ocorre quando uma mensagem *XML* é enviada com uma infinidade de assinaturas digitais e um analista ingênuo olharia para cada assinatura e usaria todos os ciclos da *CPU*, consumindo todos os recursos (YE, 2008).

***NTP*,** é um ataque de negação de serviço volumétrico distribuído baseado em reflexão. No qual um invasor explora uma funcionalidade de servidor *NTP* para sobrecarregar uma rede ou servidor alvo com uma quantidade amplificada de tráfego *UDP*, tornando o alvo e sua infraestrutura adjacente inacessíveis ao tráfego regular (RADHA et al., 2010).

***DNS*,** o invasor sobrecarrega um determinado servidor *DNS* com tráfego aparentemente válido, sobrecarregando recursos do servidor e impedindo a capacidade dos servidores de direcionar solicitações legítimas (JACKSON et al., 2009).

SNMP, são ataques que envolve a obtenção de uma elevada quantidade de respostas a um único endereço IP falsificado. Durante um ataque de reflexão *SNMP*, o autor envia um grande número de consultas *SNMP* com um endereço IP falsificado (da vítima) para vários dispositivos conectados que, por sua vez, respondem a esse endereço falsificado (SEKAR et al., 2006).

BitTorrent, o invasor utiliza uma vulnerabilidade em seu protocolo Micro *Transport Protocol* (uTP) chamado *libuTP* e utiliza arquivos popular (filmes, músicas, jogos e etc.) como um chamariz para lançar um ataque DDoS utilizando os membros do *BitTorrent*. Este ataque via *BitTorrent* não requer nenhuma modificação no *software* do lado do cliente e, por tanto, pode ser implementado a nível mundial por um único invasor (DEFRAWY; GJOKA; MARKOPOULOU, 2007).

SSDP, o agente malicioso explora os protocolos de rede UPnP (*Universal Plug and Play*) para enviar uma quantidade amplificada de tráfego a uma vítima, visando a infraestrutura do alvo e colocar o seu recurso da *Web* indisponível (LYU et al., 2017).

CharGEN, o invasor se aproveita desse protocolo ativado por padrão nas impressoras, copiadoras e etc. E utiliza para executar ataques *CharGEN* que pode ser usado para inundar um destino com pacotes *UDP* na porta 19. Quando o destino tenta entender essas solicitações, ele não conseguirá concluir a solicitação e o servidor acabará esgotando seus recursos e entrará *offline* ou será reinicializado (SANTANNA et al., 2015).

Portmap, o agente malicioso gera um elevado número de pacotes UDP com um endereço IP de origem falsificado para fazer com que pareça que os pacotes estão vindo do destino pretendido. Esses pacotes UDP são enviados para os servidores *Portmapper* (porta 111) (MANSFIELD-DEVINE, 2015).

QOTD, o atacante falsifica o IP de um pacote entre duas máquinas rodando o *QOTD*. Isso fará com que eles lancem caracteres um para o outro, reduzindo a velocidade das máquinas e saturando a rede (LYU et al., 2017).

MSSQL, o invasor obtém uma amplificação significativa através de um extenso número de servidores *MS SQL* voltados para a vítima (KRUPP; BACKES; ROSSOW, 2016).

2.1.9 Ataques *Exploitation of software bug*

Spear phishing, o atacante falsifica um *e-mail* que tem como alvo uma organização ou indivíduo específico, buscando acesso não autorizado a informações confidenciais (HONG, 2012).

Whaling, o agente efetua tentativas direcionadas a roubo de informações confidenciais a procura de ganho financeiros (GAMUNDANI; NEKARE, 2018).

***Baiting*,** o invasor conduz a vítima a acessar determinado conteúdo infectado e assim obter acesso informações e ou arquivos sigilosos (NDIBWILE et al., 2015).

***Spyware*,** geralmente são janelas *pop-up* com aparência legítima exibidas no navegador da vítima. No entanto, a infecção é feita por uma falsa instalação de uma ferramenta que irá "limpar"o sistema da vítima, mas o mesmo será infectado por algum vírus (THOMPSON, 2005).

2.1.10 Ataques *Validation input*

***SQL injection*,** o invasor aproveita a falha no sistema de compartilhamento com uma base de dados através de comandos *SQL*, onde o atacante pode inserir instruções para obter informações sobre o banco de dados (VIVINSANDAR; SHENAI, 2012).

***Cross site scripting (XSS)*,** o atacante injeta *scripts* maliciosos via *client-side* dentro da página *Web* vistas por outros usuários e obter informações sobre *cookies* de acesso e dentre outros privilégios (BARBOSA; CASTRO, 2016).

***Trojans*,** o agente malicioso através de um *e-mail* contendo um arquivo infectado com jogos, filmes e etc, têm o propósito de infectar a vítima e obter acesso ao computador da mesma (LEE et al., 2008).

***Viruses*,** o atacante infecta a vítima com um *software* malicioso que, quando executado, se réplica modificando outros programas de computador e inserindo seu próprio código (ZARGAR; JOSHI; TIPPER, 2013).

2.2 *MALWARE*

Esta seção aborda introdução sobre *malware* e suas principais características.

Malware ou *software* malicioso, exerce uma função de auxílio no acesso indevido no sistema computacional. Qualquer *software* que exerça atividade maliciosa e cause danos a um usuário de computador, pode ser considerado *malware*, maior parte se enquadra nos vírus, *backdoor*, *botnet*, *rootkit*, *scareware* e *worms* (SIKORSKI; HONIG, 2012; LE et al., 2018). Algumas desses tipos de *malware* são explicados a seguir.

2.2.1 Tipos de *Malware*

***Backdoor*,** são códigos maliciosos que se instala no computador da vítima, permitindo acesso do invasor. Os *backdoors* permitem que o invasor acesse indevidamente um computador com pouca ou nenhuma autenticação e execute comandos no sistema local.

***Botnet*,** possui semelhança ao *backdoor*, permitindo que o invasor acesse o sistema, entretanto, todos os computadores infectados com a mesma *botnet* recebem as mesmas instruções de um único servidor de comando e controle.

***Rootkit*,** foram projetados para ocultar a existência de outros códigos. Os *rootkits* normalmente são combinados com outros *malware*, como por exemplo, um *backdoor*, permitindo acesso remoto, tornando o código de difícil detecção na maioria dos antivírus.

***Scareware*,** são *malware* projetados para "alarmar"um usuário de uma possível infecção, e redirecioná-lo para um *website*, supostamente seguro, onde o mesmo será informado que seu computador está infectado com vírus, e para solucionar esse problema o mesmo terá que efetuar à compra de um *software* indicado no *website*, mas na verdade, o *software* que está sendo vendido, foi projetado para roubar os dados pessoais da vítima.

***Worm*,** possui códigos maliciosos que podem reproduzir-se e infectar outros computadores.

Ainda assim, um mesmo *malware* pode abranger diversas categorias. Por exemplo, um *software* pode conter um *keylogger* para coletar senhas e repassar para um componente de *worm* que envia *spam* (BANIN; DYRKOLBOTN, 2018).

Atualmente, uma das formas mais viável para efetuar ataques DDoS, são através dos *malware*, pela existência de uma série de dispositivos que estão conectados à *Internet*. Na Tabela 2 mostramos alguns dos *malware* utilizado para a captura desses dispositivos, para compor uma *botnet* (DONNO et al., 2017).

Tabela 2 – Capacidade dos *Malware* DDoS

Malware				**DDoS**	
Nome	Ano	Código fonte	Agente CPU	Arquitetura	Ataques viáveis
Linux.Hydra	2008	Livre	MIPS	IRC	SYN, UDP
Chuck Norris	2010	E. Reversa	MIPS	IRC	SYN, UDP, ACK
Aidra	2012	Livre	MIPS, ARM	IRC	SYN, ACK
BASHLITE	2014	Livre	MIPS, PPC	Handler	SYN, UDP, ACK
Mirai	2016	Livre	MIPS, ARM	Handler	SYN, UDP, ACK

***Linux.Hydra*,** antecessor dos *malware* utilizados em DDoS. O *Linux.Hyda* surgiu em meados de 2008 como um projeto de código aberto que visava rotear dispositivos baseados na arquitetura *Microprocessor without Interlocked Pipeline Stages* (MIPS). A fase inicial, fica dependente de um ataque de dicionário (força bruta) ou, no caso se o dispositivo se tratar de um roteador D-link, em uma vulnerabilidade de

autenticação específica (JANUS, 2011). A partir do momento em que o dispositivo tenha sido infectado, ele se tornará parte de uma rede baseada em IRC, capaz de realizar ataques SYN *flood*. A documentação do *malware* informa que é possível realizar ataques UDP *flood*, no entanto não há relatos da aplicação do mesmo (PAULI, 2016). Em suma, mesmo simples, este *malware* difundiu as bases para todos os sucessivos *malware* que visam MIPS (DONNO et al., 2017).

***Chuck Norris*,** surgiu em 2010, após da descontinuação do seu antecessor, *Psyb0t* (DURFINA; KROUSTEK; ZEMEK, 2013). Efetuaram uma engenharia reversa (E. Reversa) do *Psyb0t* para a criação do *Chuck Norris*, este *malware* é capaz de efetuar ataques *flood* utilizando UDP e ACK (JANUS, 2011), tem como alvo a arquitetura MIPS similar ao *malware Linux.Hydra*.

***Aidra/LightAidra/Zendran*,** desenvolvidos em meados de 2012, esses três *malware* possuem similaridades no código. Em comparação com os *malware* citado anteriormente, possuem um maior grau de complexidade, são capazes de ser compilados em outras arquiteturas, como, MIPS, *Advanced RISC Machine* (ARM) e *Performance Optimization with Enhanced RISC – Performance Computing* (PowerPC), ainda assim, o método de infecção depende somente de um ataque de autenticação (FAZZI, 2016). Estabelecido na arquitetura IRC, o mesmo possui capacidade para efetuar ataques utilizando *flood* SYN-ACK.

***BASHLITE*,** possui características semelhantes ao *malware Spike*. A arquitetura do *BASHLITE* está baseada em *handler* (MALWAREMUSTDIE, 2016), a variedade das arquiteturas que o *BASHLITE* infecta são, MIPS, ARM, PowerPC, *SuperH*, *Scalable Processor Architecture* (SPARC). Podem efetuar ataques *flood* utilizando SYN-ACK, UDP.

***Mirai*,** um dos *malware* predominantes dos últimos anos, conhecido por ter efetuado um dos maiores ataques DDoS já conhecidos, incluindo o abuso do serviço de *Internet* francês e provedor de hospedagem OVH em 22 de setembro de 2016 (ANGRISHI, 2017), (MILLMAN, 2016), a interrupção do serviços *DynDNS* (serviço dinâmico de resolução de nomes) em 21 de outubro de 2016 (ANGRISHI, 2017; YORK, 2016; HILTON, 2016). O *Mirai* foi projetado para infectar e controlar dispositivos IoT (como roteadores domésticos, câmeras DVRs, CCTV e etc.) depois de infectados, os dispositivos retornam o acesso ao servidor que controla as Botnet. Posteriormente, a Botnet pode ser utilizada para realizar diversos tipos de ataques DDoS, explorando diversos protocolos, como, *Generic Routing Encapsulation* (GRE), TCP, UDP, *Domain Name System* (DNS) e *HyperText Transfer Protocol* (HTTP) (MANSFIELD-DEVINE, 2016).

2.3 DEPENDABILIDADE

Em (AVIZIENIS et al., 2004) o termo dependabilidade é definido como a habilidade de um sistema computacional de entregar ou fornecer um serviço de maneira justa e confiável. Esta definição de alto nível considera a dependabilidade como uma grande caixa preta que só pode ser analisada do ponto de vista do usuário do sistema, e sua avaliação será dada com base no comportamento esperado desta caixa, e das respostas por ela fornecidas, a cada entrada realizada. Por tratar-se de um termo abrangente, apesar da curta definição, a avaliação de dependabilidade costuma se inter-relacionar ao estudo de um conjunto de **Ameaças** que impedem a entrega do serviço de maneira desejada; de **Meios** que são os responsáveis por garantir os resultados dos **Atributos** que, por sua vez, nada mais são do que métricas relacionadas à entrega do serviço ofertado da já então mencionada maneira justa e confiável. Estas propriedades podem ser esquematizadas através de uma árvore, como na Figura 2

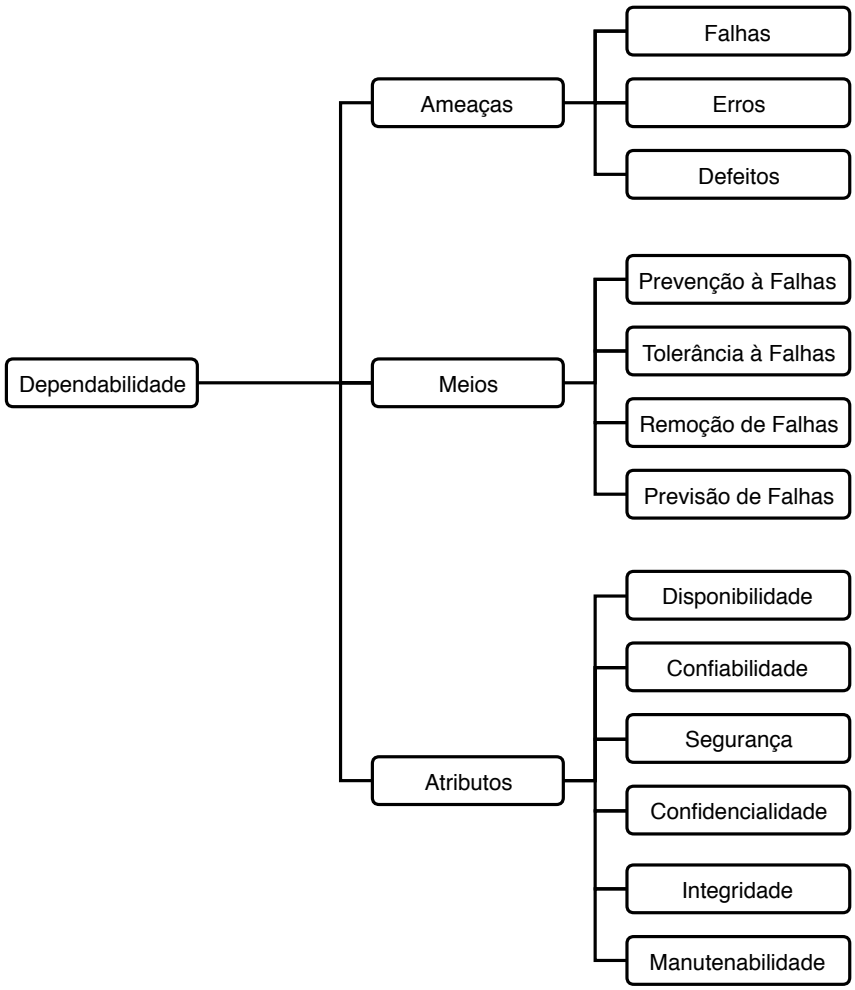


Figura 2 – Árvore de Dependabilidade (baseado em (AVIŽIENIS et al., 2001))

Os três principais ramos da árvore de dependabilidade (Ameaças, Meios e Atributos), são descritos nas subseções a seguir e as suas relações com esta dissertação são apontadas

no decorrer do texto.

2.3.1 Ameaças à Dependabilidade

Existem três grandes ameaças à dependabilidade, as folhas da grande árvore englobam as falhas, erros e defeitos que impedem a entrega do serviço da maneira desejada.

Os erros são as ameaças inclusas no próprio *software* e, geralmente, lá estão por má programação no processo de desenvolvimento, ou estão ainda mais enraizados, tendo passado como despercebidos durante toda a etapa de planejamento do sistema; no geral, erros podem acarretar um estado de defeito, que por sua vez, refere-se ao momento em que um erro atinge a interface de serviço e altera seu funcionamento, tornando-se visíveis aos usuários; enquanto que as falhas são ditas como as causas prováveis ou hipotéticas que levaram a ocorrência do erro (AVIŽIENIS et al., 2001; REGALADO, 2011).

Existem técnicas destinadas a mitigar o impacto das falhas nos sistemas computacionais, estas que estão no ramo vizinho ao das ameaças, e se intitulam Meios de Dependabilidade.

2.3.2 Meios para melhoria da Dependabilidade

Para um sistema alcançar as métricas de dependabilidade e fornecer o serviço para qual foi programado, de forma justa e confiável, é necessária a combinação de quatro técnicas que visam a redução do impacto de falhas (AVIŽIENIS et al., 2001): **Prevenção**, **Tolerância**, **Remoção** e **Previsão de Falhas**.

Prevenção a Falhas, o primeiro dos meios de dependabilidade é a prevenção às falhas, porém pode ser alcançada através da adoção de melhores técnicas para o controle de qualidade no processo de planejamento e fabricação tanto do *software* quanto do *hardware*, onde será executado (AVIŽIENIS et al., 2001).

Exercer uma etapa de levantamento e avaliação de requisitos, bem como a adoção de programação estruturada para o processo de desenvolvimento e a aplicação de testes frequentes, tende a ser o bastante para uma redução considerável da ocorrência de falhas no futuro.

Mas na ocorrência de falhas, torna-se imprescindível que algum mecanismo garanta que o sistema não fique indisponível aos seus usuários e continue entregando aquilo para que foi programado de maneira correta. Tais mecanismos são explicados a seguir.

Tolerância às Falhas consiste na entrega do serviço de maneira correta, mesmo na presença de falhas, e na capacidade que um sistema tem de se recuperar logo após a detecção de um erro (AVIŽIENIS et al., 2001). Um sistema capaz de detectar a ocor-

rência de erros de forma automática e o local de acontecimentos deste evento terá uma maior probabilidade de ser reparado antes da ocorrência de falhas catastróficas.

Depois de ser reparado, é ideal a todo e qualquer sistema a remoção da falha que reduziu a sua capacidade de operação, esta folha da árvore de dependabilidade é explicada logo abaixo.

Remoção de Falhas é um processo constante, sendo realizado durante o desenvolvimento de um sistema e no decorrer de todo o seu tempo de vida útil (AVIŽIENIS et al., 2001). Quanto mais tempo dedicarmos às análises primárias e a etapa de validação, menor tempo tende a ser gasto removendo falhas no futuro, o que pode acarretar diretamente em um aumento na disponibilidade do produto ou serviço.

Se pudermos prever a ocorrência de falhas com o máximo possível de exatidão, poderemos removê-las antes que ocorram, mitigando seu impacto sobre o fornecimento do serviço. Este que é o meio de dependabilidade explicado a seguir.

Previsão de Falhas ao se utilizar de técnicas para a análise comportamental de um sistema é possível prever, com certo índice de acurácia e confiança, quando o serviço poderá deixar de ser ofertado corretamente (AVIŽIENIS et al., 2001). Com valores estimada é possível reduzir e mitigar os impactos da ocorrência de uma falha no sistema, e através da aplicação de manutenções corretivas e preventivas é possível aumentar a continuidade no fornecimento do serviço e torná-lo mais confiável.

Confiabilidade esta, que é um dentre os seis **atributos de dependabilidade** apresentados na próxima Subseção.

2.3.3 Atributos de Dependabilidade

Os atributos de dependabilidade referem-se às metas a serem alcançadas por sistemas que almejam ser considerados confiáveis de forma justificada. Um destes atributos, a disponibilidade, recebe um destaque maior em sua explanação, por tratar-se do foco desta dissertação de mestrado.

Disponibilidade, este termo pode ser resumido como a probabilidade de um sistema estar operacional, ou seja, esteja acessível aos seus usuários dentro de um período de tempo pré-estabelecido (IEEE, 1990). Esta probabilidade pode ser dada através do tempo em que o sistema tende a ficar disponível ou *Uptime* e do seu período de indisponibilidade *Downtime*, estes valores são legalmente determinados no *Service Level Agreements* (SLA), sendo assim, a disponibilidade é uma métrica importante

para provedoras de serviços e infraestruturas. A Equação 2.1 representa o cálculo de disponibilidade (MACIEL et al., 2011) para um sistema ou serviço.

$$A = \frac{E[Uptime]}{E[Uptime] + E[Downtime]} \tag{2.1}$$

onde **A** equivale à *availability* ou disponibilidade e **E** ao valor esperado.

Outra alternativa para a representação de disponibilidade de um sistema é em termos de tempo médio entre as falhas, *Mean Time to Failure* (MTTF) (MACIEL et al., 2011), e do tempo médio entre os reparos, *Mean Time to Repair* (MTTR) (MACIEL et al., 2011), como pode ser visto na Equação 2.2 (MACIEL et al., 2011).

$$A = \frac{MTTF}{MTTF + MTTR} \tag{2.2}$$

Já a Equação 2.3 apresenta a disponibilidade em **número de noves** (NN) (MACIEL et al., 2011), e quanto maior a quantidade, maior será a sua disponibilidade, um serviço com 99,44% de disponibilidade, também pode ser interpretado como um sistema com 2,25 noves de disponibilidade, por exemplo.

$$NN = \log_{10}(UA) \tag{2.3}$$

Mas, eis que surge a seguinte pergunta: Se um sistema possuir 100% de disponibilidade, como representar isto em números de nove? É praticamente impossível a garantia desse valor, nem mesmo as maiores empresas que oferecem infraestruturas e serviços pela Internet anunciam uma disponibilidade tão alta. Em contrapartida, existe o cálculo da indisponibilidade do sistema, representado na equação anterior como **UA** (MACIEL et al., 2011), e que pode ser calculado como o inverso da disponibilidade ou 1 - A.

A Equação 2.4 (MACIEL et al., 2011) representa o cálculo para tempo médio entre as falhas do sistema.

$$MTTF = \int_0^\infty R(t)dt \tag{2.4}$$

Já o cálculo para se determinar o tempo médio entre os reparos do sistema depende diretamente do resultado proveniente da derivada anterior, e é representado pela Equação 2.5 (MACIEL et al., 2011):

$$MTTR = MTTF \times (\frac{UA}{A}), \tag{2.5}$$

O período de *downtime* do sistema pode ser entendido como o período em que o serviço esteve indisponível dentro de um intervalo de tempo. O cálculo desta métrica leva em consideração a relação entre o tempo levado para uma equipe de

manutenção locomover-se ao local da falha e a detecção de sua ocorrência, período é conhecido como *período-sem-reparo* ou *non-repair time* (NTR) e o tempo necessário para reparo da falha (TTR), deste modo, podemos calcular o *downtime* de um serviço através da expressão $Downtime = NRT + TTR$ (MACIEL et al., 2011).

Há um outro atributo de dependabilidade ligado diretamente aos tempos de falha do sistema, mas que, diferentemente da disponibilidade, não leva em consideração os valores para reparo, esta é a **confiabilidade**, a seguir apresentada.

Confiabilidade é a probabilidade do sistema ter executado sua função até um limite de tempo, previamente estabelecido e sem interrupções (AVIŽIENIS et al., 2001; BOLCH et al., 2006).

A confiabilidade de um sistema pode ser medida de maneira semelhante ao cálculo da disponibilidade, porém as taxas relacionadas ao reparo do mesmo não são consideradas, a Equação 2.6 apresenta este cálculo.

$$R(t) = P(T > t), t \geq 0 \tag{2.6}$$

Onde T é a variável aleatória que representa o tempo para falha, e que está dentro do intervalo [0, t] e R a confiabilidade (KUO; ZUO, 2003), do inglês *reliability*.

Segurança é uma métrica diretamente relacionada à confiabilidade do sistema, já que depende da inexistência de interrupções em seu fornecimento. Pode-se dizer que um sistema é seguro quando a ocorrência de falhas catastróficas é pouco provável, porém não é possível dizer o mesmo sobre a existência de perigos eminentes (AVIŽIENIS et al., 2001).

Integridade , um sistema íntegro é aquele onde alterações impróprias não serão aplicadas, sendo assim, podemos afirmar que a integridade de um sistema nada mais é do que um pré-requisito para as métricas de disponibilidade, confiabilidade e segurança (AVIŽIENIS et al., 2001).

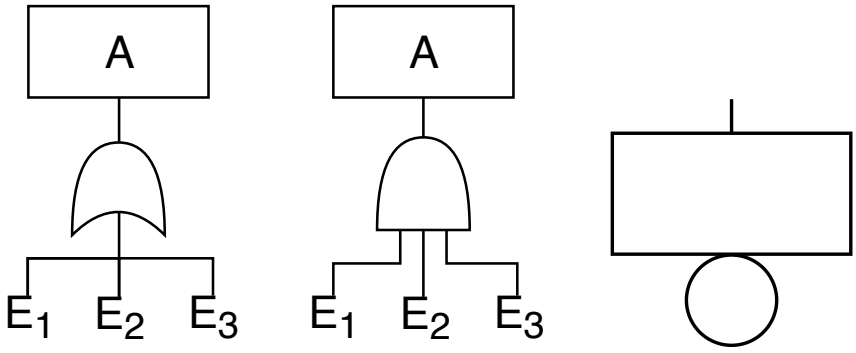
Manutenabilidade é a habilidade que um sistema tem de receber reparos e manutenções após a ocorrência de falhas (AVIŽIENIS et al., 2001), ou a probabilidade de um sistema ser reparado após a ocorrência de uma falha em algum momento (DANTAS, 2013).

Confidencialidade , um sistema é dito confiável quando não há divulgação de caráter não autorizado de dados e informações sobre o mesmo (AVIŽIENIS et al., 2001), mantendo aquilo que deve estar e continuar em segurança e sigilo.

2.4 ÁRVORE DE ATAQUE

As árvores de ataque existem há cerca de 27 anos, sob diferentes nomes (DILLON-MERRILL et al., 2008). Contudo, a aplicabilidade das mesmas para avaliações de vulnerabilidades dentro das redes de computadores é relativamente nova. As árvores de ataque têm sido pouco pesquisadas e, como tal, ainda possuem uma enorme quantidade de aplicações. Apresentamos nesta seção a narrativa que descreve o desenvolvimento e fornece o estado atual da arte, no que diz respeito às árvores de ataque, bem como as métricas necessárias para uma análise de segurança (EDGE, 2007).

Os principais componentes das árvores de ataques são as portas lógicas *OR* (Figura 3a), portas *AND* (Figura 3b) e eventos de entrada (Figura 3c). A porta *OR* indica que o evento de saída A ocorre se algum dos eventos de entrada E_i ocorrer, já a porta *AND* indica que o evento de saída A ocorre apenas quando todos os eventos de entrada E_i ocorrer ao mesmo tempo. O evento básico representa uma falha e não requer desenvolvimento adicional (ARNLJOT; RAUSAND, 2009; RAUSAND; ARNLJOT, 2004).



(a) Porta *OR* (b) Porta *AND* (c) Eventos de entrada

Figura 3 – Principais componentes de uma árvore de ataque

Nos laboratórios AT&T Bell, surgiu uma das primeiras publicações a respeito das árvores de ataque (WEISS, 1991). No documento de requisitos da engenharia de segurança de sistemas ao MIL-STD-1785, que foi utilizado durante o desenvolvimento da *Strategic Defense Initiative* (SDI) para o *Department of Defense* (DoD). As árvores de ataque desenvolvidas por (WEISS, 1991) são conhecidas até hoje e possuem portas lógicas *AND*, *OR* e eventos de entrada.

Para satisfazer as características de um nó *AND*, todos os nó filhos devem obter sucesso para atingir seu objetivo. Entretanto, um nó *OR* requer apenas que um único filho obtenha sucesso para atingir seu objetivo. Um ataque ao sistema *UNIX* [6] é apresentado na Figura 4. O nó raiz da árvore é o objetivo central do invasor, logo, obtendo privilégios de administrador. Esse nó raiz é decomposto em nós filhos com condições que levam ao nó

[6] Sistema Operacional multiusuário e multitarefa desenvolvido na *Bell Labs* no início dos anos 70.

raiz, os filhos são decompostos até que as ações externas do atacante sejam determinadas, os nós inferiores são considerados como nó folha (EDGE et al., 2006; SHAN-SHAN; YA-BIN, 2018).

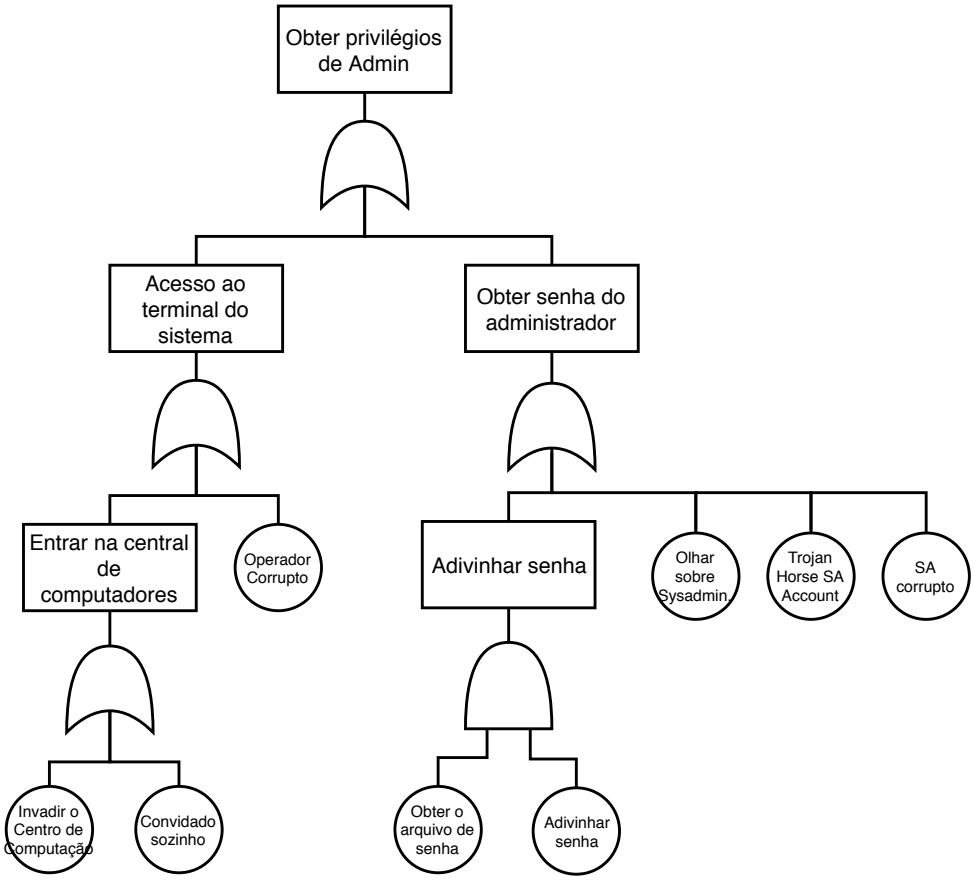


Figura 4 – Exemplo de árvore de lógica de ameaça para o sistema UNIX (WEISS, 1991)

Ainda assim, um trabalho apoiado pela *National Security Agency* (NSA) que utilizou às árvores de ataque como parte de um processo de cinco etapas de uma estrutura para desenvolver um sistema seguro. A árvore de ataque é definida como uma "ferramenta de visualização para enumerar e avaliar diferentes ataques contra o sistema"(SALTER et al., 1998).

Apesar disso, (SCHNEIER; SHOSTACK et al., 1999) deu continuidade ao trabalho da NSA acerca das árvores de ataque e apresentou-as de forma mais abrangente em um artigo decorrente, de um jeito formal e sistemático de narrar a segurança dos sistemas. O objetivo do invasor é representado através do nó raiz e os ramos de cada nó representa uma sub tarefa necessária para atingir a meta do nó pai. A Figura 5 é uma árvore de ataque constituída por um invasor tentando penetrar um cofre físico. O autor abordou diferentes métricas nos nó de ataque, desde valores *booleanos* dos ataques possíveis (P) e impossíveis (I), como também atribuiu valores em dólar no custo do ataque. As métricas são atribuídas nos nó folha e se propagam até o nó raiz.

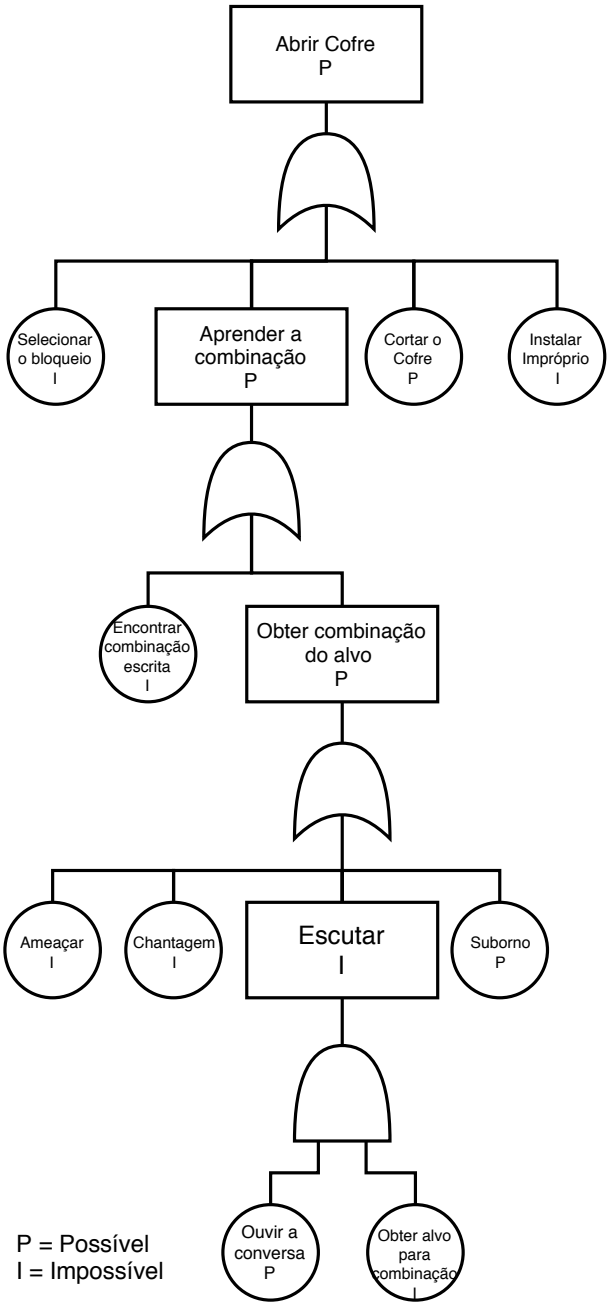


Figura 5 – Árvore de Ataque: Arquitetura Básica (SCHNEIER, 1999)

A idealização da árvore de ataque é dada por um processo iterativo e requer conhecimento diante do que irá modelar. As metas de ataques devem ser definidas e desenvolvidas em uma estrutura de árvore. Os nós individuais podem ter métricas atribuídas a eles. Algumas métricas podem mudar, na medida em que a tecnologia muda e os ataques se tornam mais fáceis. As características das árvores de ataque e a atribuições de métricas são baseadas na experiência do analista de segurança que desenvolve a árvore. (SCHNEIER, 1999) estabeleceu um conhecimento sobre às árvores de ataque, no entanto, ainda há muito para ser desenvolvimento e aprimorado.

A Amenaza, empresa desenvolvedora do produto *SecureITree* escreveu vários artigos

sobre árvore de ataque, uma de suas publicações, (INGOLDSBY, 2004) apresentou uma teoria por trás das versões atuais, através do *software SecureITree*. Para o invasor efetuar um ataque tem que existir três condições necessárias. O defensor deve possuir vulnerabilidades, o invasor ter recursos suficientes para explorar as vulnerabilidades e o invasor deverá obter algum benefício com o ataque.

A primeira condição é determinada exclusivamente pelo defensor, a segunda por uma combinação do atacante e do defensor, e a terceira particularmente pelo atacante. Essas três premissas são utilizadas para tentar prever o comportamento do atacante, e constatando qual será o impacto do defensor no caso de um ataque bem-sucedido (INGOLDSBY, 2010). Contudo, as árvores de ataque que a *Amenaza* produz, são similares àquelas elaboradas por (SCHNEIER, 1999). As árvores implementadas por ele, os nós AND/OR são utilizados para produzir uma árvore com um único nós raiz. Os nó de folha representam ataques básicos.

Apesar disso, as árvores modeladas através da *SecurITree* leva em consideração os recursos assumidos pelo atacante. Ao determinar o nível de recursos que o invasor possui, o *software* remove os ataques que estão além das capacidades declaradas do invasor. Com os ataques restantes, o *software* assume que o invasor fará algum tipo de análise de custo benefício, não necessariamente formal ou consciente, mas sim determinando, qual ataque produzirá um maior retorno pelo menor custo. Todavia, o *software SecurITree* possui funções de utilitário não linear para modelar melhor o custo, capacidade técnica e capacidade de detecção do invasor.

Cada nó na folha da árvore de ataque, representada na Figura 13, engloba indicadores dos seus respectivos componentes, que nos permite calcular métricas de interesse, como, Custo do Ataque (CA), Probabilidade do Ataque (PA), Benefício do Ataque (BA), Facilidade do Ataque (FA), Percepção de Dano (PD), Propensão do Ataque (PPA) e Habilidade Técnica (HT).

Alguns valores são necessários para a utilização das métricas em questão. A Tabela 3 descreve a habilidade técnica de um invasor, conforme seu nível de conhecimento a respeito da tecnologia (EDGE et al., 2007a). A habilidade do invasor é enumerada de forma crescente e está vinculado ao seu conhecimento acerca da área de Tecnologia da Informação (TI).

Tabela 3 – Habilidade técnica - Atacante

Habilidade	**Conhecimento do Invasor**	**Definição do Conhecimento**
20 – 30	Usuário médio	Usuário sem habilidades de programação, pouco conhecimento sobre sistemas;
30 – 40	Usuário intermediário	Poucas habilidades de programação;
40 – 60	Usuário avançado	Especialista em segurança de TI;
60 – 80	Usuário super-avançado	Alto conhecimento em segurança de TI, várias habilidades com programação.

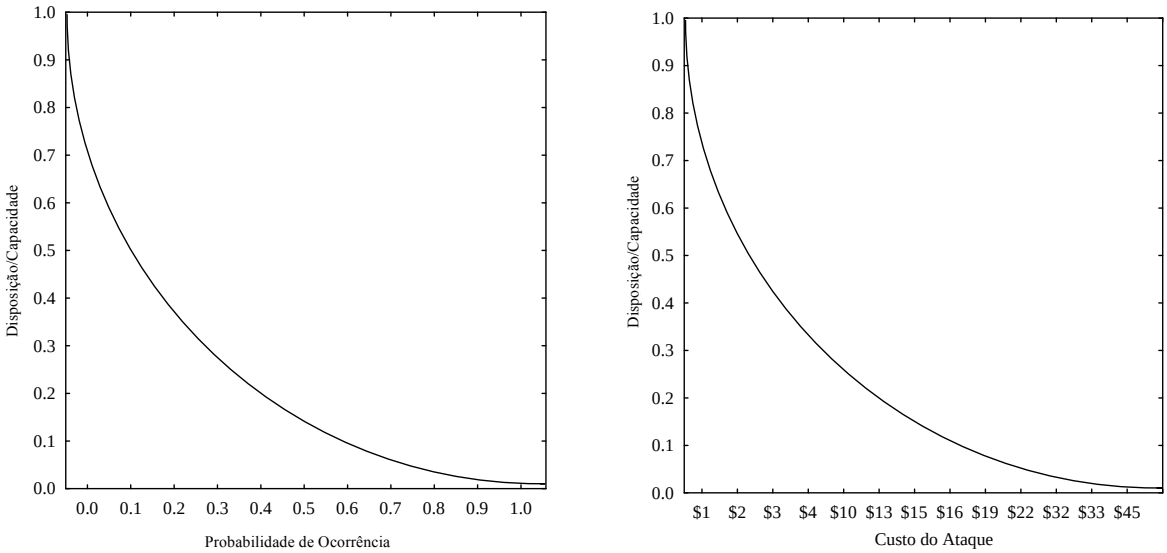
Ainda assim, a descrição da habilidade do invasor denota o seu conhecimento para efetuar um ataque, o que pode acarretar perdas para a empresa, como apresentado na Tabela 4 que exibe informações referentes à perda de reputação de uma empresa e leva em consideração o dano causado pelo ataque (INGOLDSBY, 2010).

Tabela 4 – Perda de reputação - Vítima

Alcance	**Impacto**	**Definição do impacto**
$1 \leqslant I < 4$	Negligível	Pequeno impacto,
$4 \leqslant I < 7$	Baixo	Impacto moderado,
$7 \leqslant I < 10$	Médio	longo tempo para recuperação,
10	Alto	Sistema inoperável ou destruído.

Diante dos parâmetros apresentados anteriormente, elaboramos perfis para avaliar o comportamento dos indicadores da vítima e do atacante, através deles realizamos uma avaliação do impacto de um ataque. Os valores utilizados para traçar os perfis foram retirados da Tabela 5. A Figura 6a mostra o perfil da probabilidade de ocorrência de um ataque, o eixo y mostra valores entre 0 e 1 (ou no intervalo [0,1]) do mesmo modo para o eixo x. Dessa forma quando a probabilidade do sistema ser invadido for 0,4, terá 0,2 de intenção em cometer o ataque.

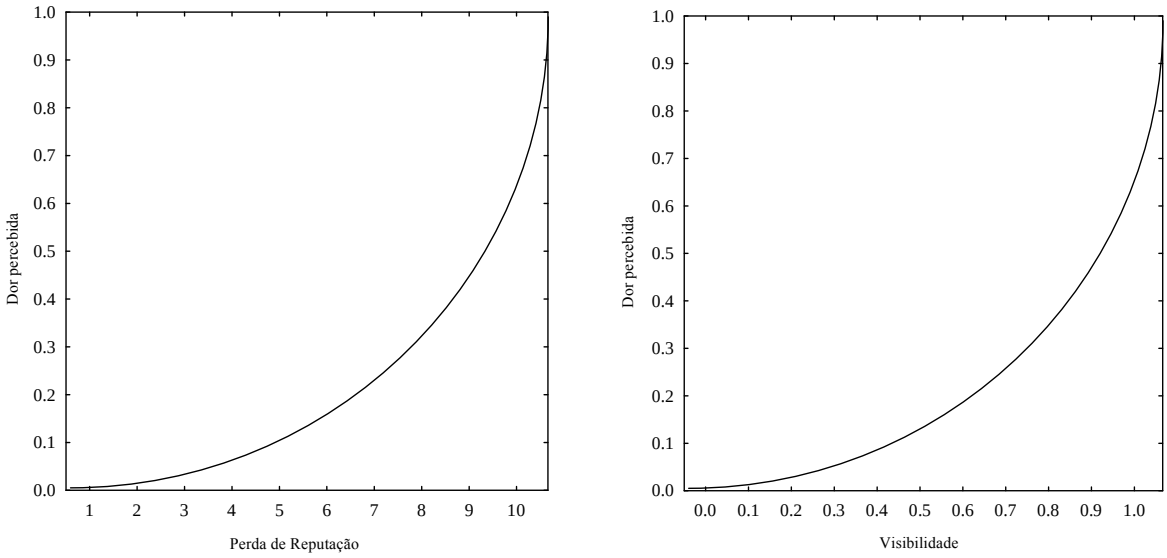
Por conseguinte, na Figura 6b o perfil se refere ao custo com o ataque, o eixo y mostra valores entre 0 e 1 e no eixo x traz valores financeiros, em que, o custo com um ataque de US $10, terá 30% de predisposição em efetuar o ataque.



(a) Probabilidade (b) Custo com Ataque ($)

Figura 6 – Habilidade do atacante de acordo com a probabilidade de sucesso do ataque e seu respectivo custo

Entretanto, a Figura 7a traz valores entre 0 e 1 no eixo y e no eixo x os valores (1,10) se refere a perda da reputação, referente a Tabela 4, caso a perda da reputação for 9, a probabilidade associada a perda será de 40%. No perfil de visibilidade da vítima (Figura 7b) exibe valores entre 0 e 1 tanto para o eixo y como para o eixo x, apresentam o quanto a vítima estar suscetível a determinada ameaça, com isso, com uma visibilidade de 80%, a chance da vítima ser acometida a um ataque, será em torno de 35%.



(a) Perda da Reputação (Numérico) (%) (b) Notoriedade da Vítima (%)

Figura 7 – Perda de reputação e Visibilidade da vítima

Por fim, a Figura 8 expõe valores no eixo y entre 0 e 1 e no eixo x mostra valores que se referem a Tabela 3. Dessa forma, um nível 30 de habilidade técnica, o atacante terá uma possibilidade de 40% de obter sucesso com o ataque.

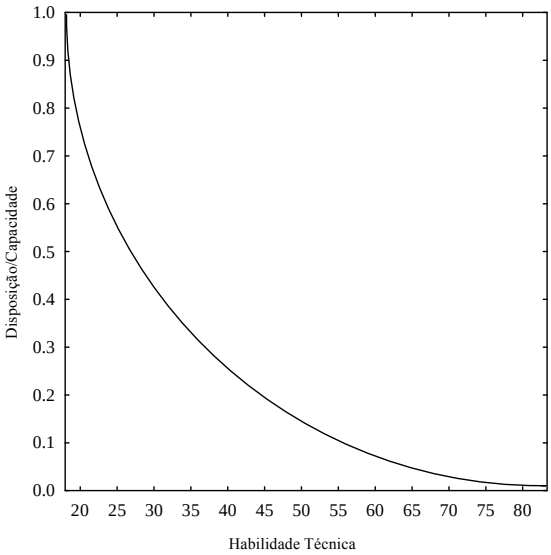


Figura 8 – Habilidades

Um ataque a um sistema computacional pode acontecer de diversas formas, maneiras, em qualquer lugar e a todo momento. Com isso, assume-se que há uma provável chance desse ataque ser bem-sucedido. A partir daí, avaliando e modelando um cenário inóspito de um possível ataque e aplicando árvores de ataques nessa avaliação, podemos encontrar entre as mais diversas probabilidades a chance desse ataque obter sucesso ou não. Diante disto, a PD pode ser calculada pela porta *AND* e vista na Equação 2.7, onde n é o número de nós folhas e $Prob_i$ é a probabilidade do ataque, que varia entre 0 e 1, os valores da probabilidade do ataque pela porta *AND* são dado pela multiplicação das probabilidades. Já o cálculo da probabilidade do ataque pela porta *OR* é feito através da Equação 2.8 onde o n é o número de nó folhas da árvore de ataque, a subtração do número um pela multiplicação das probabilidades dos ataques, é feito por estar se tratando de uma porta *OR*, a partir daí é encontrado a probabilidade de ataque (ROY; KIM; TRIVEDI, 2012; EDGE; DUBE et al., 2006).

$$PA_{AND} = \prod_{i=1}^{n} Prob_i \tag{2.7}$$

$$PA_{OR} = 1 - \prod_{i=1}^{n} (1 - Prob_i) \tag{2.8}$$

O CA envolve diversos aspectos, que vai desde o suborno de um funcionário até a aquisição de um *software* malicioso. Diante dessas circunstâncias, é possível elaborar uma avaliação por meio de uma árvore de ataque utilizando à Equação 2.9, seguindo as seguintes características. Ataques efetuados pela porta lógica *OR* é definido como a soma dos custos com o ataque, onde o n é o número de nó folhas da árvore, o $Cost_i$ está atrelado aos possíveis valores a serem gastos, o valor assumido vai de $[0, \infty]$ por não saber o valor exato que um atacante poderá gastar.

Logo, a avaliação do ataque pela porta *AND* é dado pela Equação 2.10, onde, soma-se as multiplicações das probabilidades do ataque $Prob_i$ e custo $Cost_i$ em seguida, o valor é dividido pela soma das probabilidades dos ataques (EDGE et al., 2006).

$$CA_{AND} = \sum_{i=1}^{n} Cost_i \tag{2.9}$$

$$CA_{OR} = \sum_{i=1}^{n} Prob_i \times Cost_i \tag{2.10}$$

Na maioria dos casos, ataques cibernéticos proporcionam vantagens para o atacante, por exemplo, está em ascensão, extorsão via sequestro de sistemas empresariais e liberação mediante pagamento. Diante disto, ataques computacionais têm-se intensificado, na maioria dos casos com viés financeiros. A métrica BA tange esses benefícios ao invasor, a variável n correspondo ao número de nós da árvore e o W_i *Weighting* está atrelado ao

fator de ponderação que estão entre 0 e 1 onde esses valores estão ligados ao indicador do perfil avaliado e $Prob_i$ corresponde a probabilidade associada ao ataque.

$$BA = \sum_{i=1}^{n}(W_i \times Prob_i) \tag{2.11}$$

Subverter um sistema exige conhecimento acerca das vulnerabilidades. Contudo, adentrá-lo não é simples o quanto parece, entretanto, utilizando árvore de ataque é possível obter a FA num sistema e, com isso, observar quais ameaças podem ser acometidas com maior facilidade por parte do atacante (INGOLDSBY, 2010).

A Equação 2.12 é dada através da média geométrica das funções de utilidade CA, HT e Visibilidade (VI).

$$FA = \sqrt[N]{\prod_{i=1}^{N} CA_i \times HT_i \times VI_i} \tag{2.12}$$

Os ataques cibernéticos tendem a gerar problemas para as vítimas, como perdas de clientes, credibilidade, problemas financeiros e dentre outros fatores. Contudo, através da métrica PD, é possível averiguar quais ataques causam essa inquietação à vítima (INGOLDSBY, 2010).

A Equação 2.13 apresenta as seguintes variáveis, n corresponde ao número de nó da árvore de ataque, o W_i *Weighting* está atrelado ao fator de ponderação que fica entre 0 e 1. Os valores são obtidos através Tabela 5 relativos às perdas financeiras e calculados através das funções de utilidades, apresentadas nos perfis dos indicadores, Figura 7a.

$$PD = \sum_{i=1}^{n}(W_i \times Prob_i) \tag{2.13}$$

O termo propensão é utilizado para enfatizar que o significado é semelhante à probabilidade. No entanto, há uma boa razão para acreditar que a propensão corresponde intimamente à probabilidade, ou mais precisamente, à frequência relativa da probabilidade (INGOLDSBY, 2010).

A propensão do ataque é dada pela Equação 2.14, adotamos a combinação das seguintes métricas, FA e BA para cada vulnerabilidade. O objetivo é fornecer um resultado que demonstre a frequência relativa de um ataque.

$$PPA = FA \times BA \tag{2.14}$$

3 MODELOS

Este capítulo apresenta uma metodologia de apoio para a modelagem e avaliação dos cenários de ciberataque DDoS e *malware*, propondo identificar as principais ameaças que causam maior impacto nos serviços providos por sistemas computacionais, considerando os atributos de dependabilidade como integridade e confidencialidade. No decorrer desta seção, será explicado o passo a passo da avaliação.

3.1 METODOLOGIA PARA MODELAGEM E AVALIAÇÃO

Nesta seção, é apresentada uma visão geral da metodologia de apoio utilizada nesta dissertação, que consiste em três atividades: Compreensão das principais ameaças DDoS, Definição da modelagem, Modelagem, Validação e Avaliação. A Compreensão das principais ameaças DDoS, está atrelada ao entendimento comportamental de cada método de ciberataque utilizado nas práticas de ataques DDoS em sistemas computacionais. A definição da modelagem está dividida em duas sub etapas, definição dos indicadores do atacante e da vítima, e a definição dos perfis do atacante e da vítima. Por fim, serão avaliados os resultados gerados pelo estudo.

O fluxograma da Figura 9 representa, de forma visual, a metodologia de apoio adotada no planejamento da avaliação do impacto de ataques DDoS nos serviços providos por sistemas computacionais. Os retângulos representam cada etapa da metodologia de apoio que foi seguida obedecendo à ordem de execução apontada pelas setas. Somente quando uma etapa é finalizada, o avaliador seguirá para a próxima. Já o losango representa uma etapa que pode seguir por dois caminhos diferentes, dependendo do resultado obtido. Nesse caso, a análise é encerrada se os resultados forem satisfatórios e, caso não sejam, segue para a próxima etapa. O retângulo tracejado de preto, representa as sub etapas da atividade em questão.

Ainda assim, o retângulo tracejado de vermelho, foi definido para mostrar que essa atividade não será abordada no estudo por restrições de tempo. Cada atividade da metodologia de apoio é descrita em detalhes a seguir.

3.1.1 Avaliação

Esta subseção apresenta cinco atividades: compreensão das principais ameaças DDoS, definição do modelo e construção do modelo. Tem a finalidade de expor os insumos necessários para a implementação da etapa posterior da avaliação.

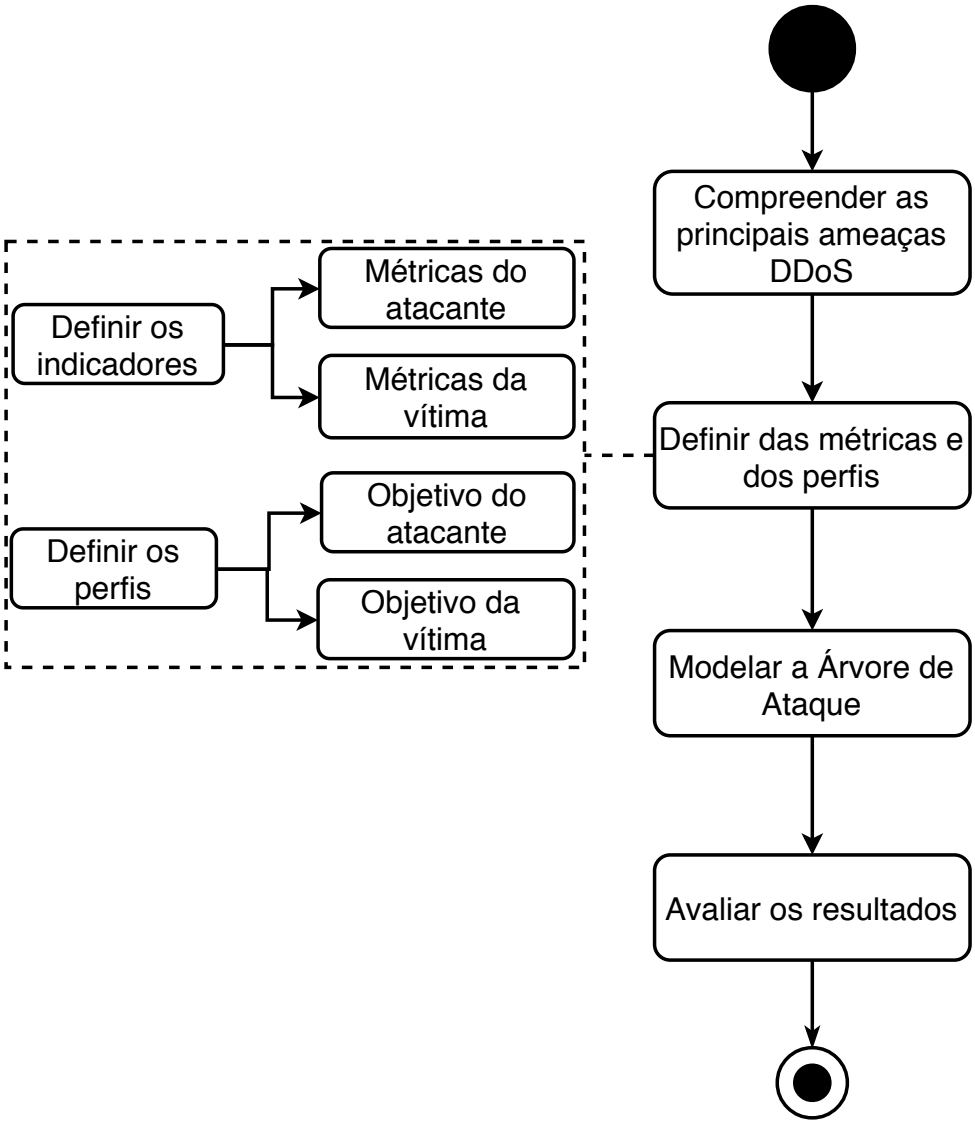


Figura 9 – Metodologia Adotada

3.1.2 Compreensão das principais ameaças DDoS

Caraterizada pelo conhecimento comportamental de cada ameaça na utilização dos ciberataque, identificando recorrências, aplicabilidade e funcionalidade, visando uma delimitação no âmbito do trabalho a ser executado, como métricas de probabilidade do ataque, custo com o ataque, benefícios do atacante, facilidade do ataque, percepção de dano, propensão do ataque e habilidade técnica do invasor.

Contudo, o entendimento das ameaças virtuais e vulnerabilidades para ataques DDoS, requer atenção e cuidado por parte do avaliador, para evitar erros de interpretação e não comprometer as demais etapas da metodologia da modelagem e avaliação dos impactos dessas vulnerabilidades em ambientes computacionais. Entendendo bem o comportamento dos ciberataque utilizando DDoS, é possível identificar as vulnerabilidades que causam maiores impactos à vítima e assim pode-se tomar uma decisão assertiva a respeito da vulnerabilidade em questão.

3.1.3 Definição do modelo

Descreve os indicadores do atacante e da vítima, bem como os perfis de ambos, conforme a descrição:

- **Definir indicadores:**
  - **Indicadores do atacante,** são métricas de interesse no qual avaliamos nesse estudo, são elas: custo do ataque, probabilidade de ocorrência, habilidade técnica e visibilidade do atacante.
  - **Indicadores da vítima,** propõem as métricas de interesse no qual avaliamos nesse estudo, são elas: visibilidade da vítima, perdas operacionais e perdas de reputação.

A Figura 10 exibe a janela da ferramenta *SecurITree* (TERRANCE, 2017) onde é possível alterar os indicadores pertencentes a modelagem em questão, baseado nisso, é possível observar que custo com ataque, probabilidade de ocorrência, habilidade técnica, perdas operacionais, reputação e visibilidade estão atreladas a cada vulnerabilidade. São parâmetros de entrada que são utilizados para encontrar o impacto ocasionado por cada vulnerabilidade avaliada.

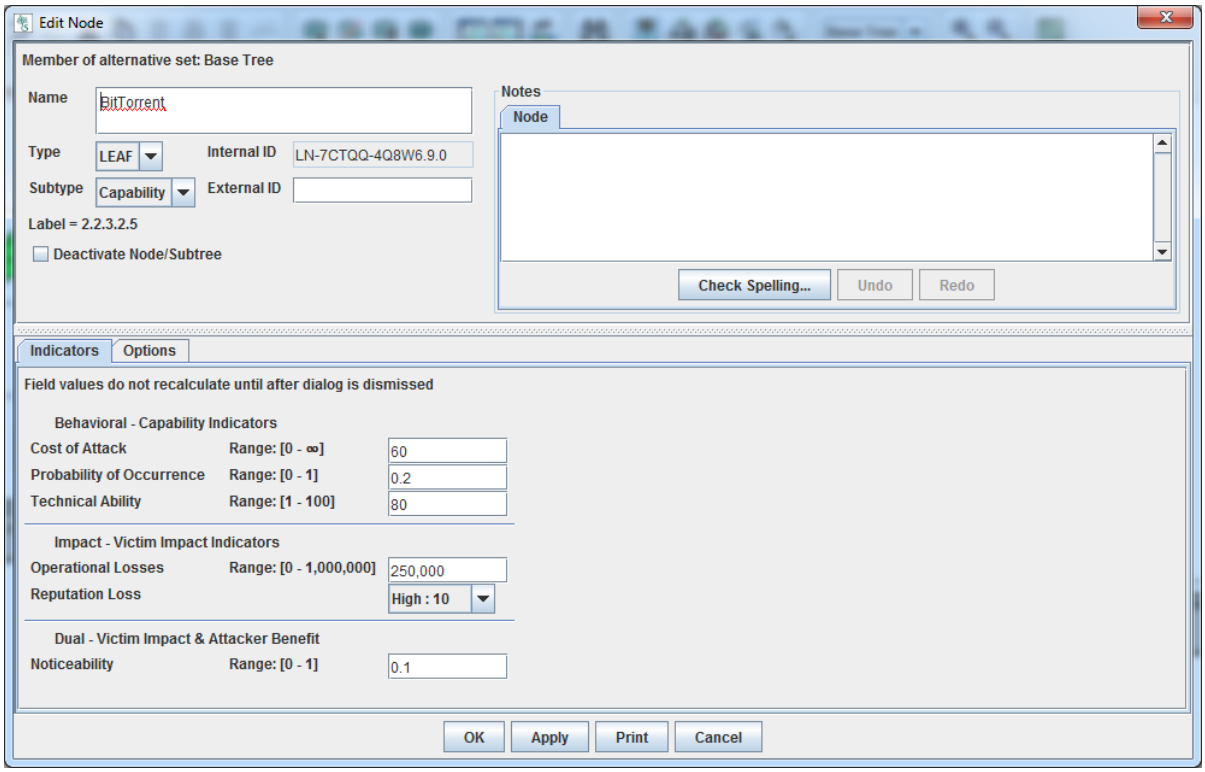


Figura 10 – Indicadores do comportamento da ameaça

- **Definir perfis:**

- **Perfis do atacante,** apresenta os objetivos de interesse no qual avaliamos nesse estudo, são eles: custo do ataque, probabilidade de ocorrência, habilidade técnica e visibilidade do atacante.
- **Perfis da vítima,** propõe os objetivos de interesse no qual avaliamos nesse estudo, são elas: visibilidade da vítima, perdas operacionais e perdas de reputação.

A Figura 11 mostra o perfil relacionado ao custo do ataque, os quadrados azul e verde, indicam a possibilidade de clicar e arrastar para alterar o perfil do atacante mediante o custo que o mesmo estará disposto a gastar para efetuar determinado ataque. No exemplo em questão, o eixo y sempre trará valores entre 0 e 1, no entanto, o eixo x varia de acordo com o perfil avaliado.

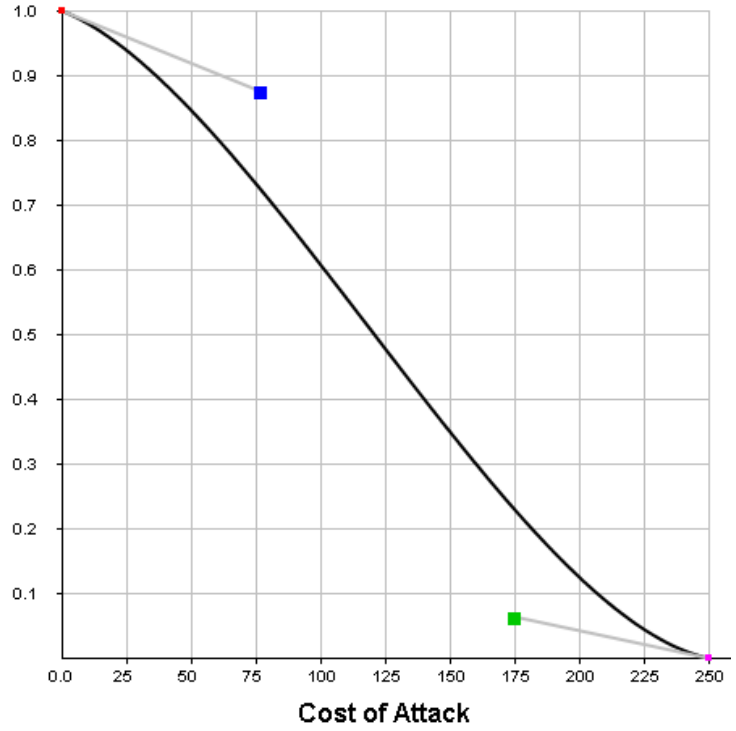


Figura 11 – Perfil de custo do ataque

3.1.4 Modelo

Na avaliação de um sistema em que deseja modelar utilizando árvore de ataque, é necessário conhecer as funcionalidades do mesmo. Nesta seção, será avaliado as principais vulnerabilidades de um sistema computacional, em seguida, viabilizaremos os primeiros modelos para representar o comportamento das principais vulnerabilidades utilizadas nos ciberataque DDoS.

A árvore de ataque deve representar de forma dedutiva as várias combinações de ataques a um sistema. A análise dedutiva inicia-se com uma conclusão geral (objetivo do

atacante) e, em seguida, tenta determinar as causas específicas da conclusão, construindo um diagrama lógico chamado de árvore de ataque.

O principal objetivo da árvore de ataque é ajudar a identificar possíveis causas de falhas do sistema, antes que as falhas realmente ocorram. Também pode ser usada para avaliar a probabilidade do evento principal usando métodos analíticos ou estatísticos.

Os símbolos básicos usados em uma árvore de ataque são chamados portas lógicas (ver Seção 2.4). A definição da porta lógica é adotada de acordo com a modelagem em questão, por exemplo, um ataque a um sistema Unix. O principal objetivo é obter privilégios de administrador do sistema, para que isso aconteça, podem ocorrer as seguintes possibilidades, acesso ao terminal do sistema ou obter senha do administrador.

Com isso, as ramificações são dadas através dessas possibilidades, que são dadas como os níveis mais altos, e a partir dos mesmos surgem outras ramificações até alcançar os níveis mais baixos, como a descoberta da senha do usuário.

3.2 SISTEMAS AVALIADOS

Nesta seção, propomos modelos para avaliar o impacto de ciberataque DDoS nos serviços providos por sistemas computacionais. Esta abordagem evidencia modelos de árvore de ataque, possibilitando identificar essas vulnerabilidades. A arquitetura geral da infraestrutura de DDoS considerada nesta dissertação é semelhante a tantas situações, tais como: sequestro de arquivos de um sistema para possíveis ganhos financeiros; disseminação de vírus; exposição de dados confidenciais sem fins lucrativos; entre outros. Na Figura 12, é exibida uma visão geral de uma infraestrutura lógica de um ataque DDoS. Entretanto, na Figura 14 é apresentada uma arquitetura, no entanto, dado ataques DDoS e *Malware*.

3.2.1 Ataque DDoS

A definição de uma arquitetura básica torna-se necessária, pois através dela somos capazes de compreender o funcionamento de um ciberataque DDoS nos serviços providos por sistemas computacionais, nos dando a possibilidade ter uma compreensão ampla e detalhada da ação do ataque. Na Figura 12 temos quatro elementos: atacante, *handlers*, Bot e vítima. Um sistema básico pode ser visto como um serviço computacional, como, por exemplo, um *website* e ou qualquer serviço *web*.

O atacante é o sujeito que efetua ciberataque a um serviço computacional, os *handlers* são utilizados para montar sua Botnet e com isso efetuar ataques DDoS no servidor da vítima. A utilização do *handler* é feita para dificultar o rastreamento do ponto de origem do ataque, caso o mesmo for descoberto, o atacante pode possuir vários *handlers* com diversos Bot em uma rede de Botnet.

Ainda assim, quando o atacante adquire usuários legítimos para montar sua Botnet, os quais são usuários desprovidos de mecanismos de defesa, como por exemplo, antivírus

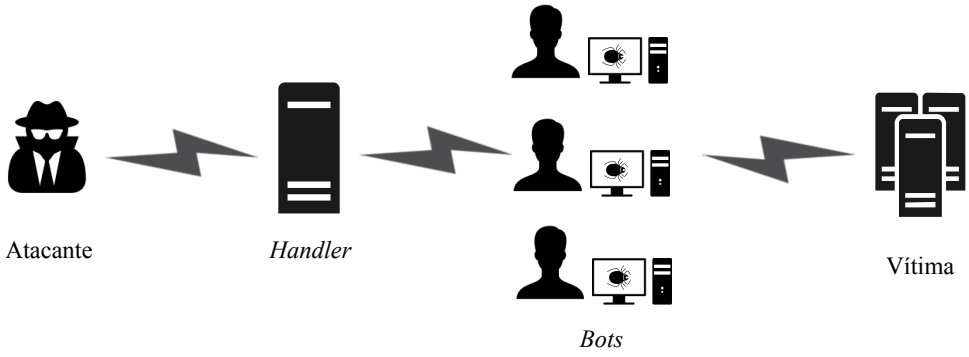


Figura 12 – Ataque DDoS e *Malware*

e etc. Em seguida, o invasor escolhe a vítima para efetuar um DDoS, sobrecarregando todo o sistema e deixando recursos inacessíveis.

O cenário proposto foi compreendido e a partir de então, utilizamos uma modelagem bastante difundida no meio acadêmico, na qual podemos ressaltar ações de um ambiente computacional, onde a utilização da árvore de ataque fez-se necessária em virtude da avaliação em tese, nos proporcionando a idealização do cenário do ciberataque utilizando DDoS. A Figura 13 apresenta o modelo proposto para o sistema computacional básico. O modelo baseia-se nas propostas de (ROY; KIM; TRIVEDI, 2012; TRIVEDI et al., 2009; MAUW; OOSTDIJK, 2005; SCHNEIER, 1999).

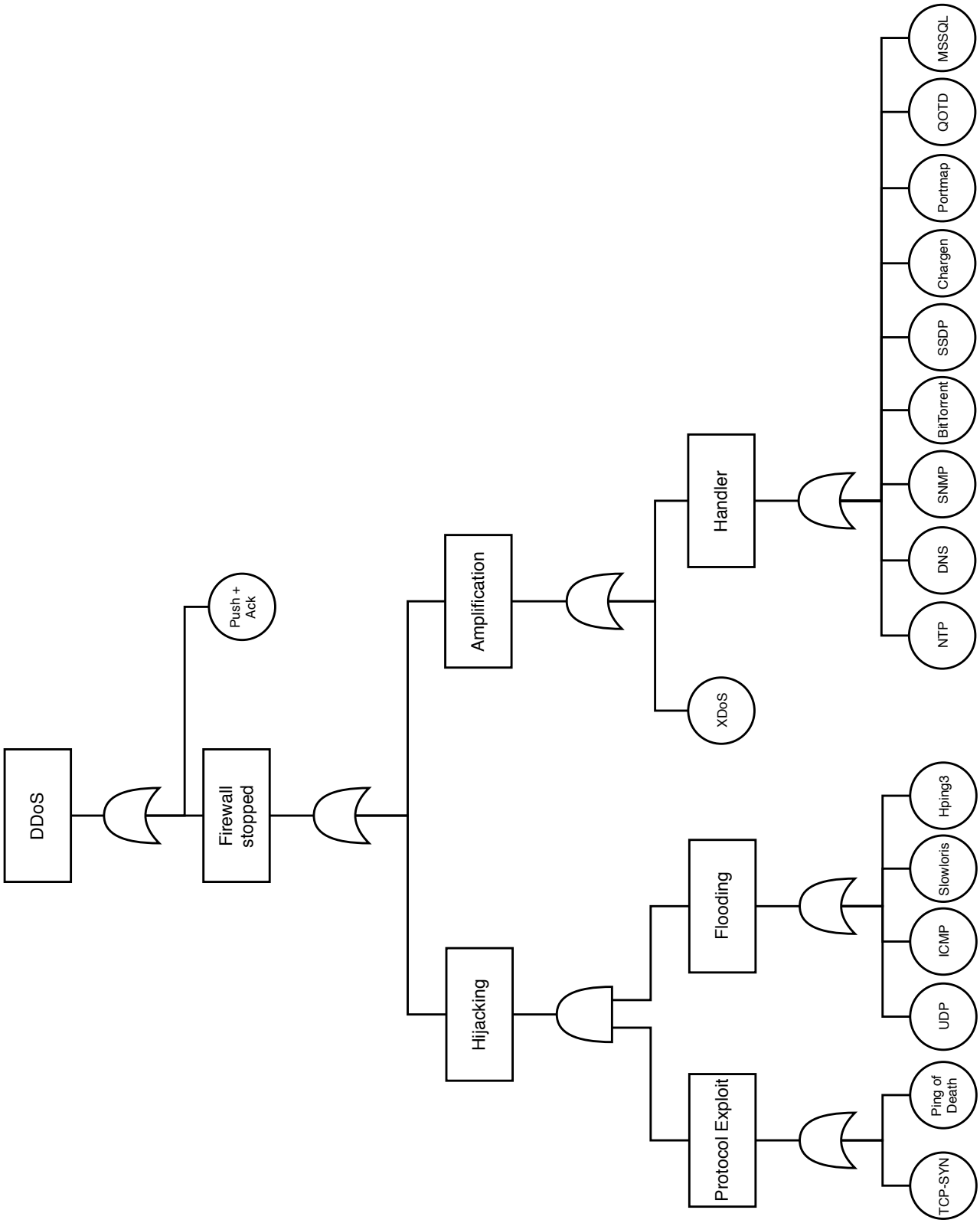


Figura 13 – Modelo de árvore de ataque para o cenário DDoS

O modelo em questão tem como objetivo demonstrar um ataque DDoS. Inicialmente adotamos uma porta lógica *OR* visto que podem ocorrer inúmeras chances de um ataque desse tipo.

Em seguida definimos os níveis dos ataques, onde, os níveis superiores na ocorrência são *Push+Ack* ou *Firewall stopped*. O *Push+Ack* é dado como um evento de entrada e não possui mais ramificações. No entanto, o *firewall* stopped possui as seguintes ramificações, *hijacking* ou *amplification*.

O ataque via *hijaking* possui ramificações com níveis mais baixos, sendo elas, *protocol exploit* e *flooding*. Dado que para ocorrer um ataque do tipo *hijaking* nesse cenário é necessário que esses dois níveis sejam satisfeitos.

Os métodos de ataque através do *protocol exploit* são definidos por uma porta ou, caso um evento seja satisfeito, o nível superior é dado como verdadeiro. O método *flooding* possui outros meios de ataque e possui a mesma porta lógica.

Ainda assim, ataques via *amplification* possui duas ramificações, ataque *XDoS* como evento de entrada ou ataque utilizando handlers que por sua vez, possui diversos eventos de entrada.

O modelo alcançado foi utilizado para análise do impacto de um ataque DDoS nos serviços providos por sistemas computacionais. Primeiramente, a partir da árvore de ataque apresentada na Figura 13, é possível obter a expressão matemática (Equação 3.1) para calcular a probabilidade do ataque. Ainda assim, é possível obter a expressão matemática (Equação 3.2) para medir custo com o ataque.

$$\epsilon_{\text{DDoS}} = (1 - (1 - \delta_{\text{FirewallStopped}}) \times (1 - P_{\text{Push+Ack}})) \tag{3.1}$$

Onde, ϵ_{DDoS} representa a probabilidade de obter sucesso com o ataque. Em que:

$$\alpha_{\text{Hijacking}} = ((1 - (1 - P_{\text{TCP-SYN}}) \times (1 - P_{\text{PingofDeath}})) \times ((1 - (1 - P_{\text{UDP}}) \times (1 - P_{\text{ICMP}}) \times (1 - P_{\text{Slowloris}}) \times (1 - P_{\text{Hping3}}))))$$
$$\beta_{\text{Handler}} = (1 - (1 - P_{\text{NTP}}) \times (1 - P_{\text{DNS}}) \times (1 - P_{\text{SNMP}}) \times (1 - P_{\text{BitTorrent}}) \times (1 - P_{\text{SSDP}}) \times (1 - P_{\text{Chargen}}) \times (1 - P_{\text{Portmap}}) \times (1 - P_{\text{QOTD}}) \times (1 - P_{\text{MSSQL}}))$$
$$\gamma_{\text{Amplification}} = (1 - (1 - P_{\text{XDoS}}) \times (1 - \beta_{\text{Handler}}))$$
$$\delta_{\text{FirewallStopped}} = (1 - (1 - \alpha_{\text{Hijacking}}) \times (1 - \gamma_{\text{Amplification}}) \times (1 - \beta_{\text{Handler}}))$$

$$\eta_{\text{DDoS}} = \left(\left(\left(P_{\zeta_{\text{FirewallStopped}}} \times C_{\zeta_{\text{FirewallStopped}}}\right) + (P_{\text{Push+Ack}} \times C_{\text{Push+Ack}})\right) / \left(P_{\zeta_{\text{FirewallStopped}}} + P_{\text{Push+Ack}}\right)\right) \tag{3.2}$$

Onde, η_{DDoS} representa o custo total para efetuar um ataque DDoS. Em que:

$$\alpha_{\text{ProtocolExploit}} = (((P_{\text{TCP-SYN}} \times C_{\text{TCP-SYN}}) + (P_{\text{PingofDeath}} \times C_{\text{PingofDeath}})) / (P_{\text{TCP-SYN}} + P_{\text{PingofDeath}}))$$
$$\beta_{\text{Flooding}} = (((P_{\text{UDP}} \times C_{\text{UDP}}) + (P_{\text{ICMP}} \times C_{\text{ICMP}}) + (P_{\text{Slowloris}} \times C_{\text{Slowloris}}) + (P_{\text{Hping3}} \times C_{\text{Hping3}})) / (P_{\text{UDP}} + P_{\text{ICMP}} + P_{\text{Slowloris}} + P_{\text{Hping3}}))$$

$$\gamma_{\text{Hijacking}} = \left(C_{\alpha_{\text{ProtocolExploit}}} + C_{\beta_{\text{Flooding}}}\right)$$
$$\delta_{\text{Handler}} = \left(\left(\left(P_{\text{NTP}} \times C_{\text{NTP}}\right) + \left(P_{\text{DNS}} \times C_{\text{DNS}}\right) + \left(P_{\text{SNMP}} \times C_{\text{SNMP}}\right) + \left(P_{\text{BitTorrent}} \times C_{\text{BitTorrent}}\right)\right.\right.$$
$$+ \left(P_{\text{SSDP}} \times C_{\text{SSDP}}\right) + \left(P_{\text{Chargen}} \times C_{\text{Chargen}}\right)$$
$$+ \left(P_{\text{Portmap}} \times C_{\text{Portmap}}\right) + \left(P_{\text{QOTD}} \times C_{\text{QOTD}}\right) + \left(P_{\text{MSSQL}} \times C_{\text{MSSQL}}\right)\right)$$
$$\left.\left./\left(P_{\text{NTP}} + P_{\text{DNS}} + P_{\text{SNMP}} + P_{\text{BitTorrent}} + P_{\text{SSDP}} + P_{\text{Chargen}} + P_{\text{Portmap}} + P_{\text{QOTD}} + P_{\text{MSSQL}}\right)\right)\right.$$
$$\epsilon_{\text{Amplification}} = \left(\left(P_{\text{XDoS}} \times C_{\text{XDoS}}\right) / \left(P_{\text{XDoS}}\right)\right)$$
$$\zeta_{\text{FirewallStopped}} = \left(\left(P_{\delta_{\text{Handler}}} \times C_{\delta_{\text{Handler}}}\right) + \left(P_{\epsilon_{\text{Amplification}}} \times C_{\epsilon_{\text{Amplification}}}\right)\right.$$
$$\left. + \left(P_{\gamma_{\text{Hijacking}}} \times C_{\gamma_{\text{Hijacking}}}\right) / \left(P_{\delta_{\text{Handler}}} + P_{\epsilon_{\text{Amplification}}} + P_{\gamma_{\text{Hijacking}}}\right)\right)$$

3.2.2 Ataque DDoS e *Malware*

Existem vários tipos de servidores. Os mais conhecidos são: Servidor de arquivos, *web*, *e-mail*. Cada um destes servidores executa uma função, por exemplo, para você visualizar uma página *web*, você está utilizando o servidor *web*, o qual é responsável pelo armazenamento das páginas de um site.

A fim de investigar o impacto de ataques DDoS e *malware* nos serviços providos por esses servidores, surgiu a necessidade de elaborar outra arquitetura que compreendesse essas características de ataque. Diante disso, a partir da Figura 14, descrevemos o passo a passo comportamental de ambos métodos de ataque.

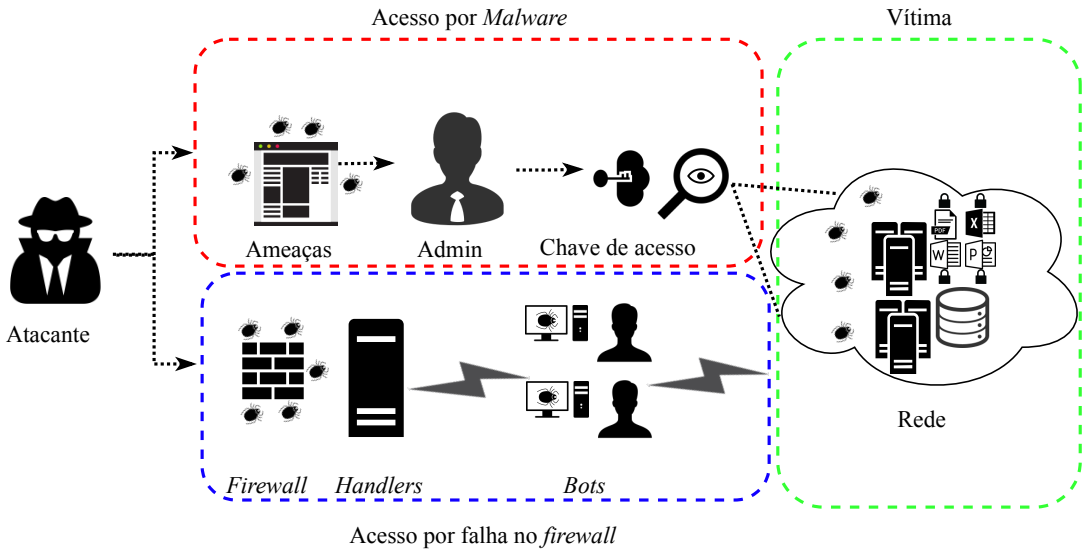


Figura 14 – Ataque DDoS e *malware*

Os ciberataque que utilizam DDoS e *malware* estão em ascensão, na Figura 14, podemos observar duas formas para efetuar esses tipos de ataques. Entretanto, a primeira viabilidade se dá através do acesso via *malware*.

Uma das formas de acesso se dá através de um ataque *Phishing*. Onde, a vítima acessa uma página web com *script* malicioso. Dessa forma, o atacante poderá efetuar instalação de um *keylogger* no computador da vítima e capturar todas as teclas que estão sendo digitadas.

Ainda assim, o mesmo poderá obter a chave de acesso e consequentemente permissão à rede interna da vítima, podendo efetuar diversas requisições como, por exemplo, acessar a base de dados e ou infectar outras vítimas.

Ataques DDoS são prejudiciais na grande maioria dos casos. Nesta arquitetura podemos observar que o atacante obtém acesso a um *handler* por falha no *firewall*. À vista disso, o invasor montará sua Botnet, por conseguinte efetuará ataques DDoS e ou obter acesso a arquivos e aplicações do servidor da vítima.

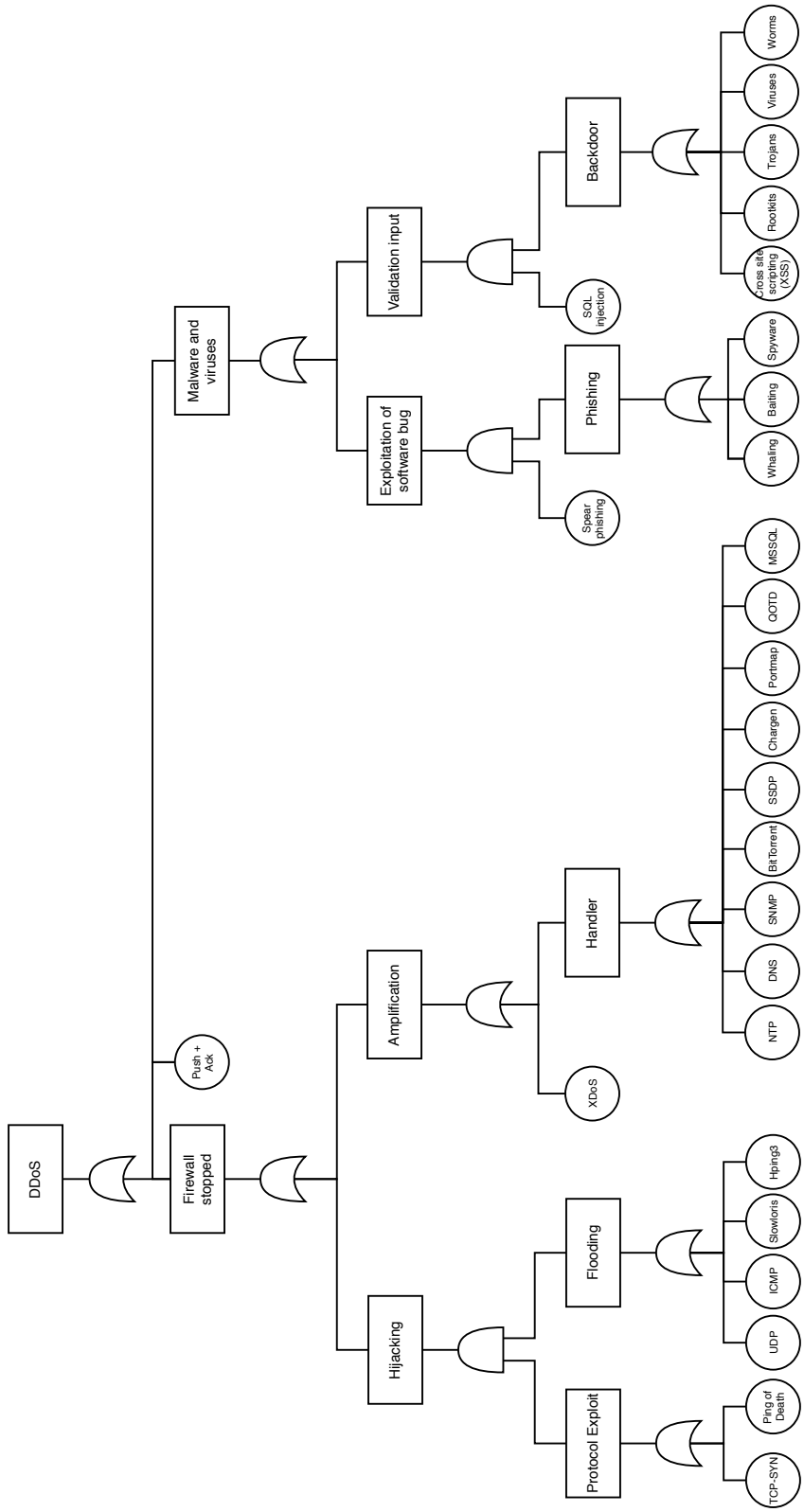


Figura 15 – Árvore de Ataque para o cenário DDoS e *malware*

O modelo alcançado foi utilizado para análise do impacto de um ataque DDoS e *malware* nos serviços providos por sistemas computacionais. Primeiramente, a partir da árvore de ataque apresentada na Figura 15, é possível obter a expressão matemática

(Equação 3.3) para calcular a probabilidade do ataque. Ainda assim, é possível obter a expressão matemática (Equação 3.4) para medir custo com o ataque.

$$\kappa_{\text{DDoS}} = (1 - (1 - \iota_{\text{FirewallStopped}}) \times (1 - P_{\text{Push+Ack}}) \times (1 - \theta_{\text{MalwareViruses}})) \tag{3.3}$$

κ_{DDoS} representa a probabilidade de sucesso com o ataque. Em que:

$$\alpha_{\text{Hijacking}} = ((1 - (1 - P_{\text{TCP-SYN}}) \times (1 - P_{\text{PingofDeath}})) \times ((1 - (1 - P_{\text{UDP}}) \times (1 - P_{\text{ICMP}}) \times (1 - P_{\text{Slowloris}}) \times (1 - P_{\text{Hping3}}))))$$
$$\beta_{\text{Handler}} = (1 - (1 - P_{\text{NTP}}) \times (1 - P_{\text{DNS}}) \times (1 - P_{\text{SNMP}}) \times (1 - P_{\text{BitTorrent}}) \times (1 - P_{\text{SSDP}}) \times (1 - P_{\text{Chargen}}) \times (1 - P_{\text{Portmap}}) \times (1 - P_{\text{QOTD}}) \times (1 - P_{\text{MSSQL}}))$$
$$\gamma_{\text{Amplification}} = (1 - (1 - P_{\text{XDoS}}) \times (1 - \beta_{\text{Handler}}))$$
$$\delta_{\text{Phishing}} = (1 - (1 - P_{\text{Whaling}}) \times (1 - P_{\text{Baiting}}) \times (1 - P_{\text{Spyware}}))$$
$$\epsilon_{\text{Backdoor}} = (1 - (1 - P_{\text{XSS}}) \times (1 - P_{\text{Rootkit}}) \times (1 - P_{\text{Trojan}}) \times (1 - P_{\text{Viruses}}) \times (1 - P_{\text{Worms}}))$$
$$\zeta_{\text{ExploitationSoftware}} = (1 - (1 - P_{\text{SpearPhishing}}) \times (1 - \delta_{\text{Phishing}}))$$
$$\eta_{\text{ValidationInput}} = (1 - (1 - P_{\text{SQLinjection}}) \times (1 - \epsilon_{\text{Backdoor}}))$$
$$\theta_{\text{MalwareViruses}} = (1 - (1 - \zeta_{\text{ExploitationSoftware}}) \times (1 - \eta_{\text{ValidationInput}}))$$
$$\iota_{\text{FirewallStopped}} = (1 - (1 - \alpha_{\text{Hijacking}}) \times (1 - \gamma_{\text{Amplification}}) \times (1 - \beta_{\text{Handler}}))$$

$$\mu_{\text{DDoS}} = (((P_{\text{FirewallStopped}} + C_{\text{FirewallStopped}}) + (P_{\text{Push+Ack}} + C_{\text{Push+Ack}}) + (P_{\kappa_{\text{MalwareViruses}}} + C_{\kappa_{\text{MalwareViruses}}})) / (P_{\text{FirewallStopped}} + P_{\text{Push+Ack}} + P_{\kappa_{\text{MalwareViruses}}})) \tag{3.4}$$

μ_{DDoS} reproduz o custo total para efetuar um ataque DDoS. Em que:

$$\alpha_{\text{ProtocolExploit}} = (((P_{\text{TCP-SYN}} \times C_{\text{TCP-SYN}}) + (P_{\text{PingofDeath}} \times C_{\text{PingofDeath}})) / (P_{\text{TCP-SYN}} + P_{\text{PingofDeath}}))$$
$$\beta_{\text{Flooding}} = (((P_{\text{UDP}} \times C_{\text{UDP}}) + (P_{\text{ICMP}} \times C_{\text{ICMP}}) + (P_{\text{Slowloris}} \times C_{\text{Slowloris}}) + (P_{\text{Hping3}} \times C_{\text{Hping3}})) / (P_{\text{UDP}} + P_{\text{ICMP}} + P_{\text{Slowloris}} + P_{\text{Hping3}}))$$
$$\gamma_{\text{Hijacking}} = \left(C_{\alpha_{\text{ProtocolExploit}}} + C_{\beta_{\text{Flooding}}}\right)$$
$$\delta_{\text{Handler}} = (((P_{\text{NTP}} \times C_{\text{NTP}}) + (P_{\text{DNS}} \times C_{\text{DNS}}) + (P_{\text{SNMP}} \times C_{\text{SNMP}}) + (P_{\text{BitTorrent}} \times C_{\text{BitTorrent}}) + (P_{\text{SSDP}} \times C_{\text{SSDP}}) + (P_{\text{Chargen}} \times C_{\text{Chargen}}) + (P_{\text{Portmap}} \times C_{\text{Portmap}}) + (P_{\text{QOTD}} \times C_{\text{QOTD}}) + (P_{\text{MSSQL}} \times C_{\text{MSSQL}})) / (P_{\text{NTP}} + P_{\text{DNS}} + P_{\text{SNMP}} + P_{\text{BitTorrent}} + P_{\text{SSDP}} + P_{\text{Chargen}} + P_{\text{Portmap}} + P_{\text{QOTD}} + P_{\text{MSSQL}}))$$
$$\epsilon_{\text{Amplification}} = (((P_{\text{XDoS}} \times C_{\text{XDoS}}) (P_{\delta_{\text{Handler}}} \times C_{\delta_{\text{Handler}}})) / (P_{\text{XDoS}} + P_{\delta_{\text{Handler}}}))$$
$$\zeta_{\text{Phishing}} = (((P_{\text{Whaling}} \times C_{\text{Whaling}}) + (P_{\text{Baiting}} \times C_{\text{Baiting}}) + (P_{\text{Spyware}} \times C_{\text{Spyware}})) / (P_{\text{Whaling}} + P_{\text{Baiting}} + P_{\text{Spyware}}))$$
$$\eta_{\text{Backdoor}} = (((P_{\text{XSS}} \times C_{\text{XSS}}) + (P_{\text{Rootkits}} \times C_{\text{Rootkits}}) + (P_{\text{Trojans}} \times C_{\text{Trojans}}) + (P_{\text{Viruses}} \times C_{\text{Viruses}}) + (P_{\text{Worms}} \times C_{\text{Worms}})) / (P_{\text{XSS}} + P_{\text{Rootkits}} + P_{\text{Trojans}} + P_{\text{Viruses}} + P_{\text{Worms}}))$$

$$\theta_{\text{ExploitationSoftware}} = \left(\left(P_{\text{SpearPhising}} \times C_{\text{SpearPhising}}\right) + \left(P_{\zeta_{\text{Phishing}}} \times C_{\zeta_{\text{Phishing}}}\right)\right.$$
$$\left.\Big/\left(P_{\text{SpearPhising}} + P_{\zeta_{\text{Phishing}}}\right)\right)$$
$$\iota_{\text{ValidationInput}} = \left(\left(\left(P_{\text{SQLinjection}} \times C_{\text{SQLinjection}}\right) + \left(P_{\eta_{\text{Backdoor}}} \times C_{\eta_{\text{Backdoor}}}\right)\right)\right.$$
$$\left.\Big/\left(P_{\text{SQLinjection}} + P_{\eta_{\text{Backdoor}}}\right)\right)$$
$$\kappa_{\text{MalwareViruses}} = \left(\left(\left(P_{\theta_{\text{ExploitationSoftware}}} \times C_{\theta_{\text{ExploitationSoftware}}}\right) + \left(P_{\iota_{\text{ValidationInput}}} \times C_{\iota_{\text{ValidationInput}}}\right)\right)\right.$$
$$\left.\Big/\left(P_{\theta_{\text{ExploitationSoftware}}} + P_{\iota_{\text{ValidationInput}}}\right)\right)$$
$$\lambda_{\text{FirewallStopped}} = \left(\left(\left(P_{\text{Hijacking}} \times C_{\text{Hijacking}}\right) + \left(P_{\text{Amplification}} \times C_{\text{Amplification}}\right)\right)\right.$$
$$\left.\Big/\left(P_{\text{Hijacking}} + P_{\text{Amplification}}\right)\right)$$

Abordamos neste capítulo modelos utilizados para a avaliação do impacto de ataques DDoS e *malware* nos serviços providos por sistemas computacionais. A aplicabilidade dos mesmos tem o propósito de mensurar a influência de um ataque de negação de serviço distribuído em um sistema computacional. Entretanto, os modelos aqui mostrados servem de apoio para constatar outros métodos de ciberataques.

Contudo, os valores obtidos para serem utilizados nos modelos podem ser obtidos através da literatura ou medição. Alguns artigos trazem valores referentes a diversos métodos de ataque, DDoS, *malware*, *phishing*, *SQL injection* entre outros. No entanto, outros métodos de ataques podem ser obtidos através de mensuração para obter a probabilidade de um ataque e o custo referente ao mesmo.

Os analistas de sistemas podem obter valores relativos a ocorrência desses métodos de ataques através do comportamento do tráfego da rede, verificando os meios mais são utilizados para efetuar ataques a mesma e, assim obter informações acerca do evento de cada ameaça. Em seguida, modela-se a árvore de ataque e insere os valores em cada nó folha para poder encontrar o impacto dessas ameaças na vítima.

4 ESTUDO DE CASO

Este capítulo apresenta estudos de caso que demostram a aplicação dos modelos apresentados no capítulo 3. O primeiro estudo de caso expõe uma avaliação das principais vulnerabilidades, relativas aos ataques DDoS nos serviços/protocolos providos por sistemas computacionais. A avaliação tem por objetivo buscar identificar o impacto na vítima mediante a ocorrência de um ataque DDoS. O segundo estudo de caso avalia ataques utilizando *malware*, onde se considerou as técnicas mais adotadas por este tipo de ataque.

4.1 ESTUDO DE CASO I - AVALIAÇÃO DE UM ATAQUE DDOS

Neste estudo de caso, evidenciam-se as avaliações dos modelos de árvore de ataque com ênfase em *DDoS*, considerando as ameaças mais relevantes.

Para a avaliação dos modelos propostos na subseção 3.2.1, que refletem o funcionamento de uma arquitetura básica, valores de entrada tornam-se necessários. Assim, para este estudo, apresentam-se os valores descritos nas Tabelas 5 e 6 que foram extraídos de trabalhos realizados por (STONEBURNER; GOGUEN; FERINGA, 2002; EDGE et al., 2006; GARVEY, 2001; EDGE et al., 2007b; ALOMARI et al., 2012; ROSSOW, 2014; SARIPALLI; WALTERS, 2010).

4.1.1 Parâmetros de entrada

Os ataques podem ser descritos por parâmetros de diferenciação. A Tabela 5 traz valores dos parâmetros do ataque. A primeira coluna se refere às técnicas que podem ser utilizadas no ataque. A segunda coluna se refere aos valores associados ao custo do ataque. A probabilidade (terceira coluna) expõe medidas relativas à ocorrência do ataque. Os valores da quarta coluna evidenciam a habilidade do atacante descrita na Tabela 3. A quinta coluna descreve o impacto do ataque que foi mencionado anteriormente (ver Tabela 4). Por fim, a sexta coluna traz a probabilidade do quanto a vítima está suscetível a determinada ameaça, valores retirados de (VERISIGN, 2018; IBRAGIMOV et al., 2018; PIETERS; DAVARYNEJAD, 2015; LÉVY-BENCHETON et al., 2015; TRIVEDI et al., 2009; HUSSAIN; HEIDEMANN; PAPADOPOULOS, 2003).

4.1.2 Resultado do modelo

Através dos modelos apresentados, foram calculadas as métricas para a arquitetura representada. As métricas utilizadas foram PA, CA, BA, FA, PD, PPA e HT (veja Seção 2.4).

Tabela 5 – Parâmetros de entrada referente a um ataque DDoS

Técnicas	Custo	Ocorrência	Habilidade	Impacto	Visibilidade
TCP-SYN	30,00	0,02	10	Negligível	0,01
Ping of Death	40,00	0,04	10	Negligível	0,03
UDP	60,00	0,40	30	Alto	0,02
ICMP (Ping)	80,00	0,20	10	Baixo	0,03
Slowloris	60,00	0,20	40	Médio	0,10
Hping3	40,00	0,30	40	Baixo	0,01
NTP	100,00	0,51	80	Alto	0,20
DNS	100,00	0,50	60	Alto	0,02
SNMP	40,00	0,03	30	Alto	0,10
BitTorrent	30,00	0,20	60	Alto	0,10
SSDP	40,00	0,37	50	Alto	0,04
Chargen	100,00	0,07	40	Médio	0,03
Portmap	60,00	0,05	40	Médio	0,04
QODT	30,00	0,03	40	Médio	0,01
MsSQL	50,00	0,10	60	Médio	0,03
XDoS	100,00	0,03	60	Médio	0,04
Push+Ack	60,00	0,04	10	Negligível	0,02

Os resultados mostraram que a PA através da (Equação 3.1) há uma chance de 92% em obter sucesso com o ataque DDoS. Ainda assim, o CA será US $87 para efetuar o ataque.

Na Figura 16, cada barra representa um tipo de ataque DDoS. Observando o gráfico, infere-se que o atacante terá cerca de 10% de chance na obtenção benefícios, com a altura da barra denotando o BA (benefício do atacante), calculado de acordo com a Equação 2.11, ou seja, menor custo com ataque e maior probabilidade de sucesso, com a utilização do ataque NTP, em outro aspecto, há chances de 5% em ainda obter benefícios utilizando os ataques *Slowloris*, SNMP e *BitTorrent*, os demais meios de ataque possui menos que 3% de chances para obtenção de benefícios, o que representa maior custo e menor chance de sucesso no ataque.

Calculou-se também a facilidade do ataque (FA), que é a relação média entre as variáveis de custo com o ataque, habilidade de técnica do invasor e a exposição da vítima. A Figura 17 mostra os tipos de ameaças e o resultado da média das probabilidades, dado que, quanto mais próximo de 1, maior a facilidade para o atacante. O resultado indica facilidade em um nível 0,8 de ataque utilizando TCP-SYN, seguido de *Hping3* com um valor de 0,7 de facilidade.

A Figura 18 exibe o resultado da percepção de dano, discutido na Seção 3.1, essa métrica apresenta o impacto de um determinado tipo de ataque, caso o mesmo venha a

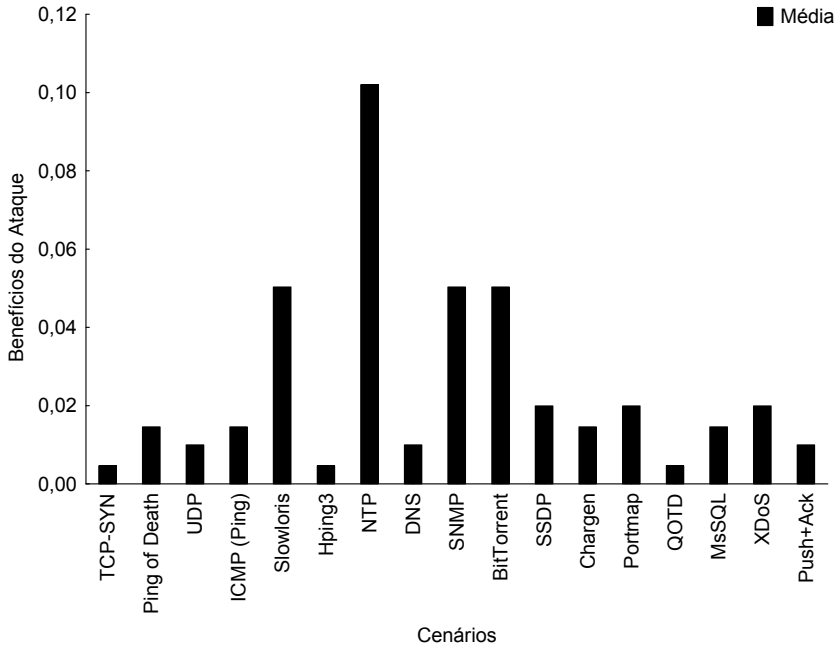


Figura 16 – Benefício com o ataque

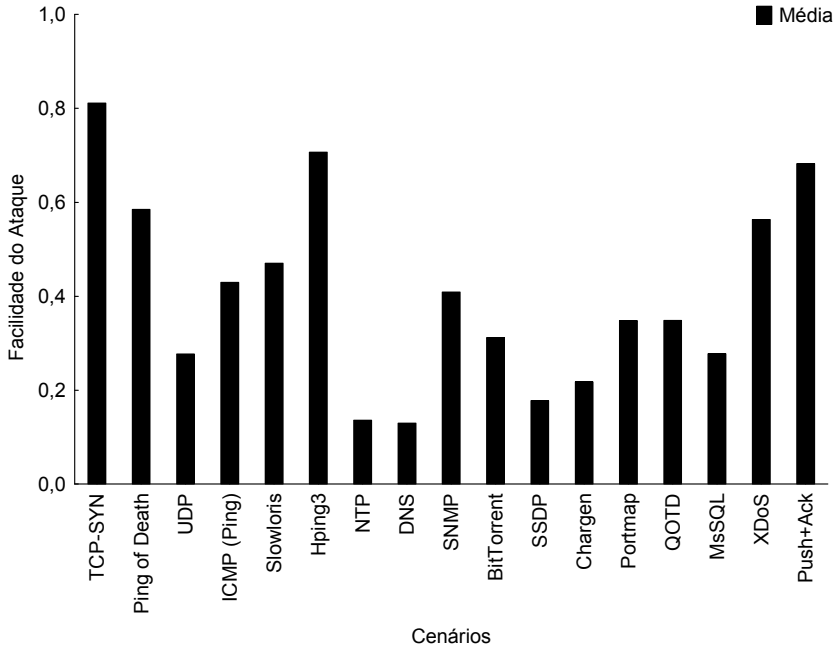


Figura 17 – Facilidade do ataque

ocorrer. Os impactos são perdas financeiras para a vítima, e também perdas operacionais. O resultado mostra que o ataque do tipo NTP pode ocasionar maior perda para as vítimas, pois é o que apresenta o maior *pain factor* (0,05). Outro ataque danoso é o do tipo *MsSQL*, que aparece com PF igual a 0,047, possibilitando perdas para o usuário.

A propensão PPA (Figura 19) de um ataque é dada através da Equação 2.14, citada na Seção 3.1. A propensão está associada à frequência relativa da ocorrência de um deter-

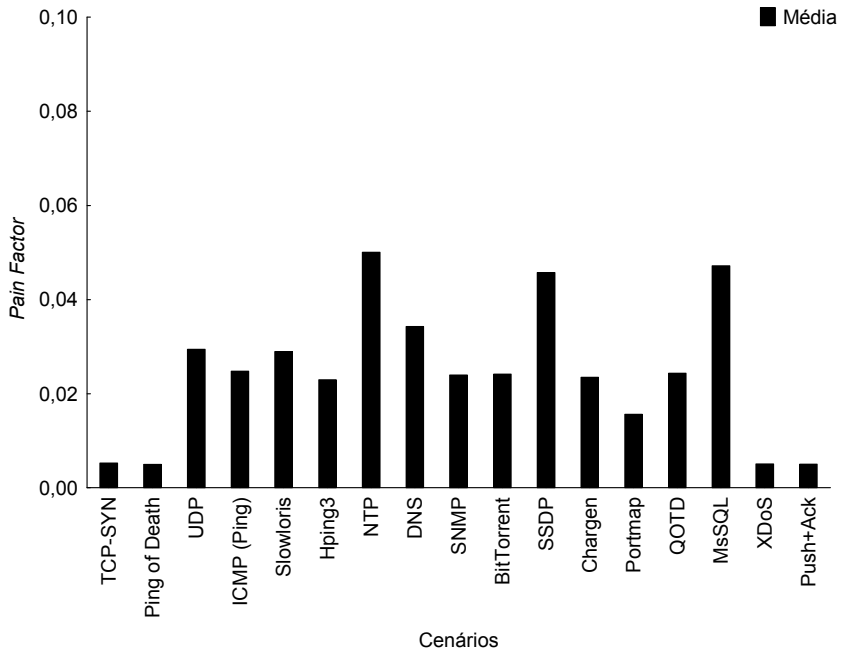


Figura 18 – *Percepção de Dano*

minado ataque, levando em consideração a facilidade do ataque e benefício com o ataque. O resultado exibe 0.023 de propensão a um ataque utilizando *Slowloris*, seguido de 0,021 de propensão a um ataque do tipo SNMP.

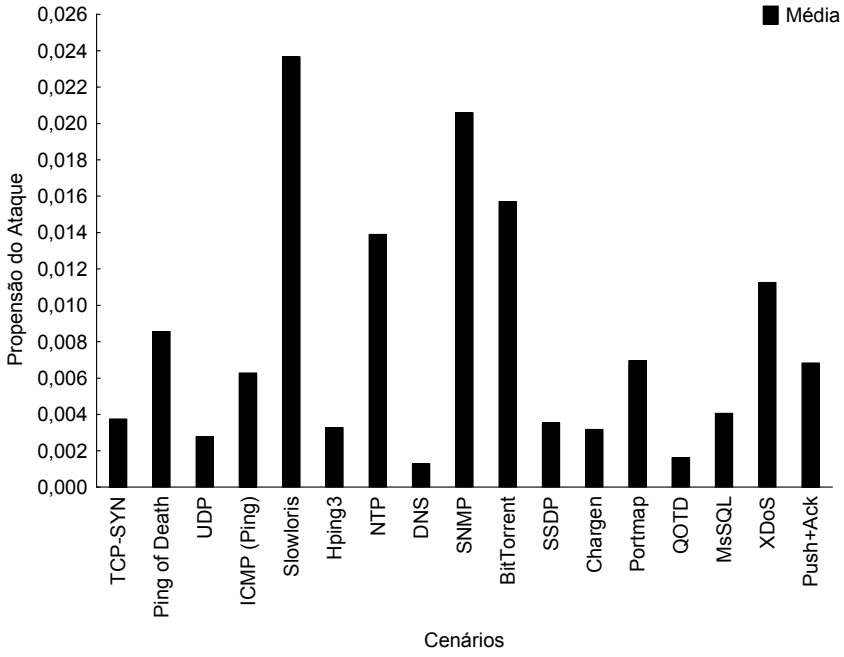


Figura 19 – Propensão do ataque

Por fim, a habilidade técnica HT define as capacidades prevalecentes de um invasor, pode ser visto na Figura 20. A avaliação revela que o invasor possui cerca de 72% de habilidade técnica com o ataque do tipo TCP-SYN e 60% de aptidão com a o ataque

push+ACK. Assim, o analista de rede deve estar atento e dispor contramedidas para antever este tipo de ameaças no seu serviço.

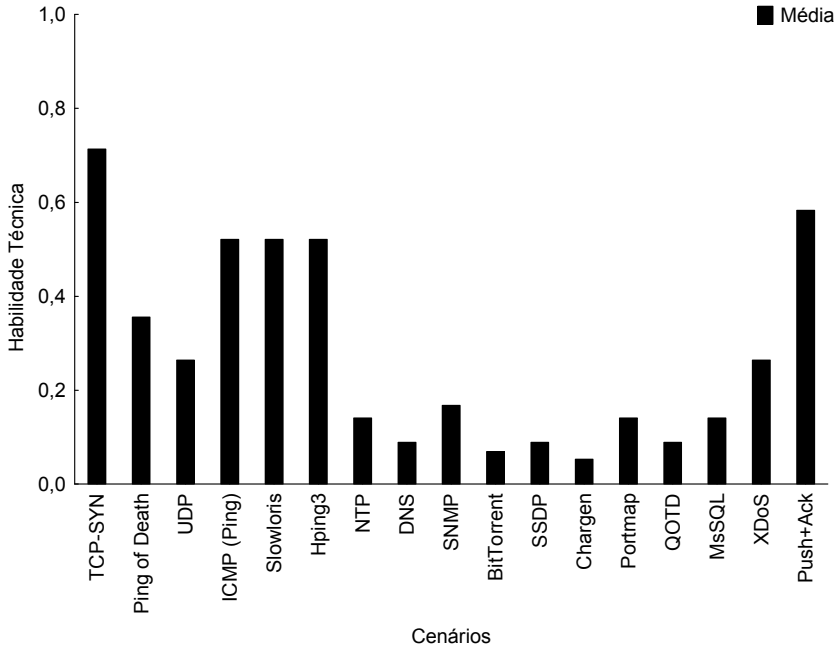


Figura 20 – Habilidade técnica do atacante

4.2 ESTUDO DE CASO II - AVALIAÇÃO DE UM ATAQUE DDOS E *MALWARE*

A fim de mostrar a versatilidade do modelo, apresentamos neste estudo de caso a adoção de outras técnicas de ciberataque na utilização de DDoS e *malware*. As principais estratégias foram subdivididas em acesso por *malware* e acesso por falha no *firewall*. Na Seção 2.1 explicamos sobre os métodos de ataque. Ainda assim, a abordagem que utilizamos visa detalhar os métodos de ataque que podem causar maiores problemas na indisponibilidade do serviço no sistema. Esta análise está vinculada ao modelo que adotamos (Seção 3.1), o mesmo considera as técnicas mais relevantes de ciberataque.

Entretanto, na avaliação do modelo proposto, levamos em considerações algumas características nos parâmetros de entrada. Tal como, habilidade técnica (Tabela 3) e perda de reputação (Tabela 4), nas quais, foram obtidas através do estudo de caso anterior, empregando novos valores na Tabela 6, utilizada para descrever as grandezas que serão utilizadas para calcular as métricas de interesse, no vigente estudo de caso.

Os valores mencionados na Tabela 6 se referem aos parâmetros de entrada do modelo em tese, a descrição de cada coluna foi mencionada na subseção 4.1.1 do estudo de caso anterior, no entanto, os valores foram ajustados ao estudo de caso em questão.

Tabela 6 – Parâmetros de entrada referente a um ataque DDoS e *Malware*

Técnicas	Custo	Ocorrência	Habilidade	Impacto	Visibilidade
TCP-SYN	30.00	0,02	10	Negligível	0,01
Ping of Death	40.00	0,04	10	Negligível	0,03
UDP	60.00	0,40	30	Alto	0,02
ICMP (Ping)	80.00	0,20	10	Baixo	0,03
Slowloris	60.00	0,20	40	Médio	0,10
Hping3	40.00	0,30	40	Baixo	0,01
NTP	100.00	0,51	80	Alto	0,20
DNS	100.00	0,50	60	Alto	0,02
SNMP	40.00	0,03	30	Alto	0,10
BitTorrent	30.00	0,20	60	Alto	0,10
SSDP	40.00	0,37	50	Alto	0,04
Chargen	100.00	0,07	40	Médio	0,03
Portmap	60.00	0,05	40	Médio	0,04
QODT	30.00	0,03	40	Médio	0,01
MsSQL	50.00	0,10	60	Médio	0,03
Whaling	100.00	0,10	60	Alto	0,10
Baiting	100.00	0,12	60	Médio	0,12
Spyware	80.00	0,10	60	Médio	0,04
XSS	60.00	0,01	40	Médio	0,03
Rootkits	40.00	0,01	50	Médio	0,03
Trojans	60.00	0,20	60	Médio	0,10
Viruses	70.00	0,10	60	Médio	0,10
Worms	80.00	0,10	40	Médio	0,10
Spear phishing	100.00	0,10	60	Alto	0,10
SQL injection	60.00	0,12	40	Médio	0,12
XDoS	100.00	0,03	60	Médio	0,04
Push+Ack	60.00	0,04	10	Negligível	0,02

4.2.1 Resultado do modelo

Dado o modelo apresentado na Seção 3.2, angariamos resultados referentes ao estudo de caso em questão. Utilizamos as seguintes métricas: PA, CA, BA, FA, PD, PPA e HT (veja Seção 2.4) para a avaliação. Os perfis utilizados pertencem aos mesmos utilizados no estudo de caso, com o objetivo de comparar ambos os resultados.

Ao calcularmos a probabilidade do ataque PA o invasor tem 97% de chance em obter sucesso com o ataque, utilizando DDoS e *malware*, notamos que analisando o estudo de caso anterior, observamos que houve um aumento de pontos percentuais de chance do invasor obter sucesso com o ataque. Ainda assim, o custo com o ataque será em torno de

US $100 (veja Equação 3.4).

O BA se refere aos benefícios do atacante dado escolhido o tipo de ataque como visto na Figura 21, onde cada barra representa um tipo de ataque DDoS. Observando o gráfico, infere-se que o atacante terá cerca de 0,20 de benefício, ou seja, menor custo com ataque e maior probabilidade de sucesso, com a utilização do ataque NTP, em outro aspecto, há chances de 0,012 em ainda obter benefícios utilizando os ataques *Bating* ou *SQL injection*, o que representa maior custo e menor chance de sucesso no ataque.

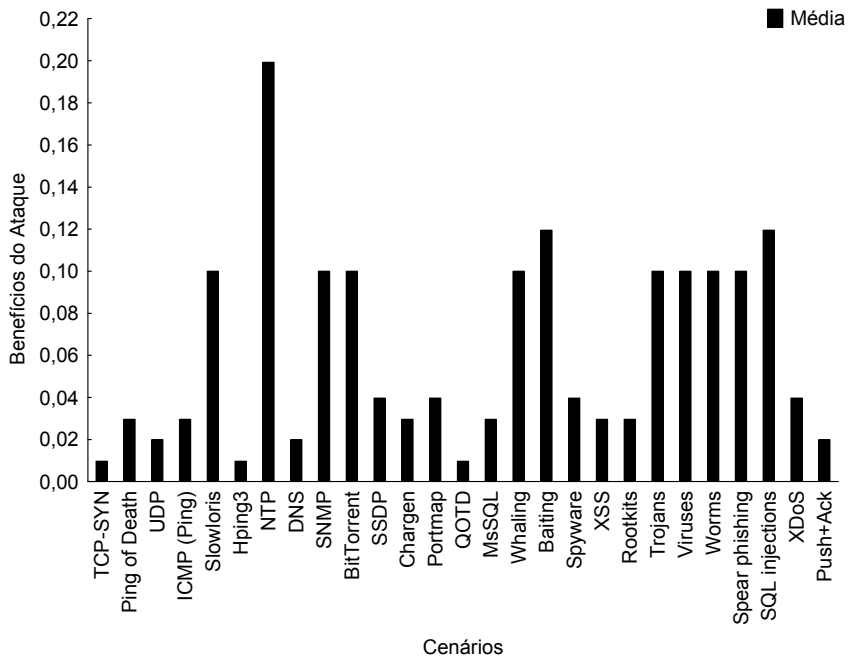


Figura 21 – Benefícios do ataque

A Figura 22 mostra os tipos de ameaças e o resultado da média das probabilidades, dado que, quanto mais próximo de 1, maior a facilidade para o atacante. O resultado indica cerca de 85% de facilidade do ataque utilizando TCP-SYN, *push*+ACK, e*Ping of Death* seguido de ICMP(Ping) com 84% de facilidade para a ocorrência do ataque.

O resultado mostra que o ataque do tipo NTP pode ocasionar maior perda para as vítimas pois é o que apresenta maior *pain factor* 61%. Outro ataque danoso é o do tipo *Spear Phishing*, que aparece com 50% possibilitando perdas para o usuário de grande magnitude.

A propensão PPA (Figura 24) corresponde a 9% utilizando *SQL injection*, seguido de 8% para um ataque do tipo SNMP.

A Figura 25 revela que o invasor possui cerca de 90% de habilidade técnica com o ataque do tipo TCP-SYN, *push*+ACK, *ping of death* e ICMP e 70% de aptidão com a o ataque SNMP. Assim, o analista de rede deve estar atento e dispor contramedidas para antever este tipo de ameaças no seu serviço.

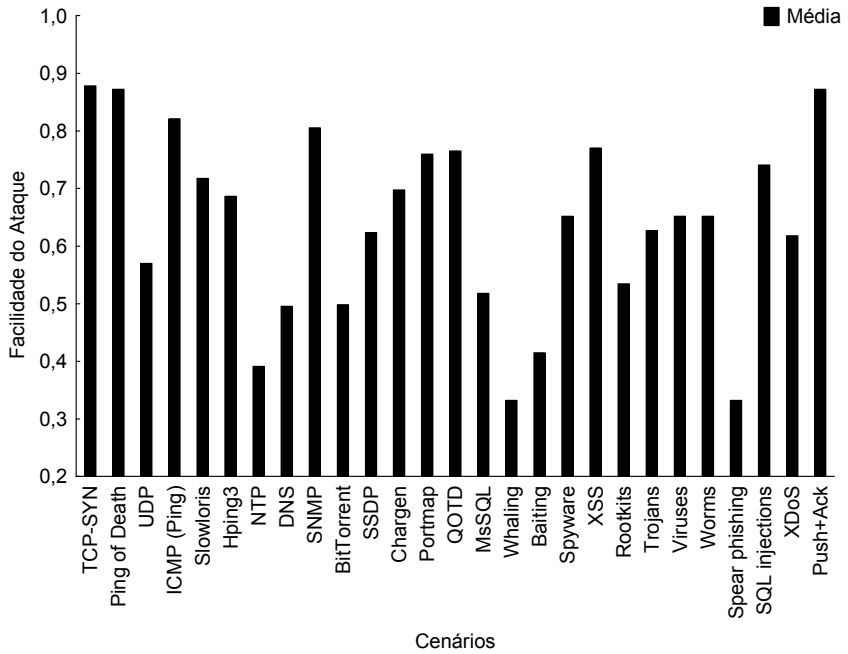


Figura 22 – Facilidade do ataque

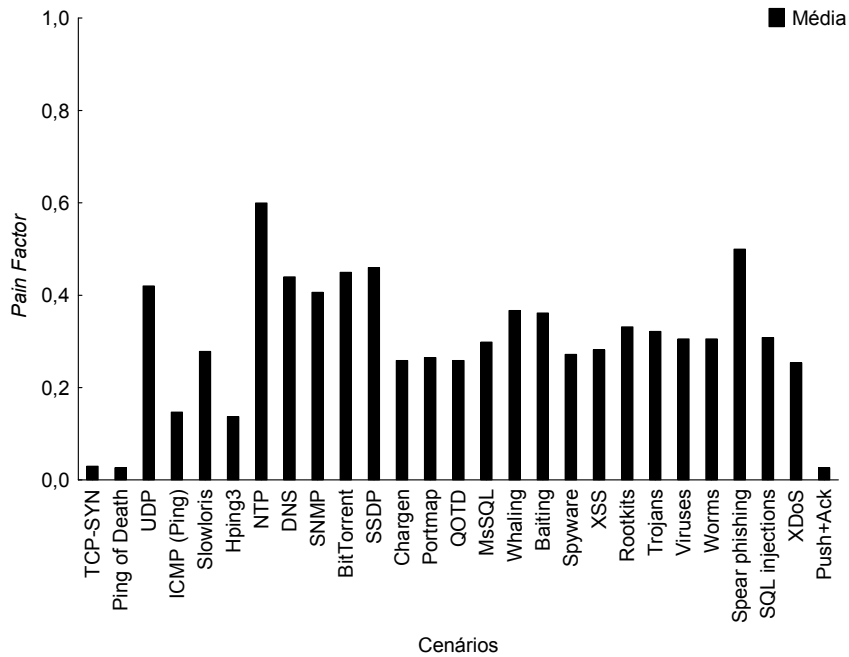


Figura 23 – *pain factor* do ataque

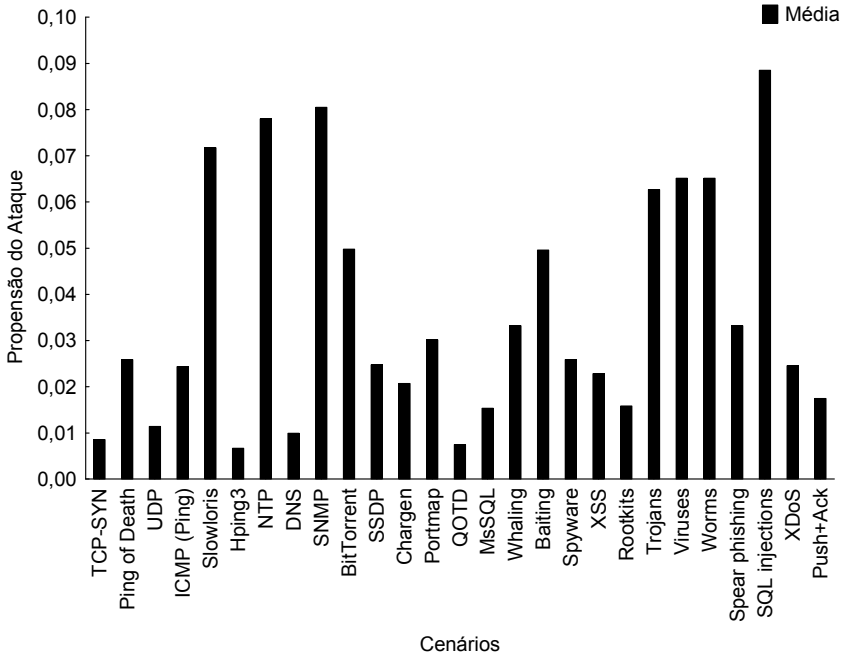


Figura 24 – Propensão do ataque

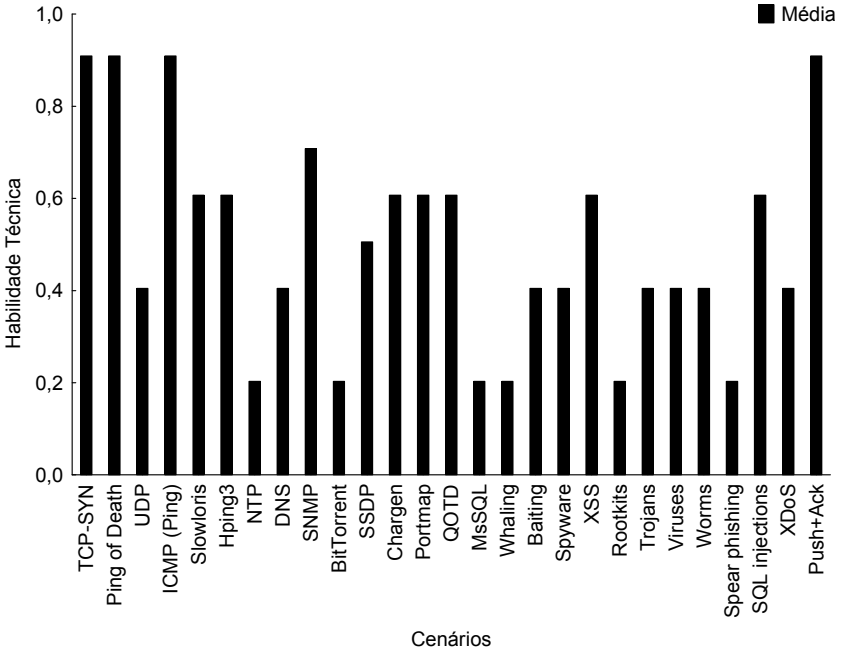


Figura 25 – Habilidade técnica do atacante

4.3 CONSIDERAÇÕES

Este Capítulo apresentou a aplicações dos modelos de árvore de ataque e estratégias propostas. Demonstrando questões no âmbito de segurança em ataques de negação de serviço distribuído que podem ser resolvidas através dessas abordagens em cenários comumente encontrados em sistemas reais. Essa abordagem tanto permite que os administradores antecipem o impacto de ciberataque através da variação individual de cada parâmetro.

5 CONCLUSÕES

Ataques DDoS estão em ascensão e preocupando empresas que dispõe de serviços na internet. Um incidente com a empresa gerenciadora de criptomoedas, Bitfinex, a deixou inoperante em junho do corrente ano, através de um ataque DDoS (CHAPARRO, 2018). Outro caso parecido, no entanto, com um desfecho desastroso deu-se pela empresa Monte A Gox, uma das maiores operadoras de *bitcoin*, sofreu um ataque DDoS em 2013 e veio a fechar em 2014, com perda de US $450 milhões (BOESLER, 2013). Ainda assim, a criptomoeda Verge no final de maio do corrente ano, sofreu um ataque DDoS e levou consigo cerca de US $1.7 milhão (MARINOFF, 2018).

O estudo realizado, bem como conceitos, estudos de caso e modelos apresentados, podem, também, serem aplicados a serviços e sistemas. É o caso de sistemas de gerenciamento bancários, saúde, escolar que precisam permanecer disponível o máximo de tempo possível para evitar que suas transações de pagamento, cadastro de usuário e ou matrícula em caso de escolas, não sejam devidamente efetivadas devido a falha de um ciberataque.

Inicialmente destacamos uma metodologia de avaliação para sistemas computacionais. A mesma foi dada através da modelagem de árvore de ataque, com o objetivo de avaliar o impacto de ataques DDoS e *malware* nos serviços providos por sistemas computacionais através da definição dos perfis da vítima e atacante, objetivando mostrar quais vulnerabilidades possui maior influência em ataques DDoS e *malware*.

Ainda assim, foi proposto o modelo da arquitetura básica de um ataque DDoS nos serviços providos por sistemas computacionais, no qual modelamos uma arquitetura básica enfatizando todos os componentes necessário em uma ocorrência de um ataque DDoS. Em seguia, um segundo modelo foi elaborado, abordando ataques DDoS e *malware*, com o objetivo de avaliar a variabilidade dos mesmos em um ambiente computacional cliente/servidor.

Entretanto, a mitigação de um ataque DDoS não é algo trivial, existem métodos para tal (OSANAIYE; CHOO; DLODLO, 2016). Ainda assim, depende de diversos fatores, como por exemplo, qual a disposição do invasor para efetuar um ciberataque, qual o impacto que irá causar na vítima. Dessa forma, foi avaliado o impacto de um ataque DDoS e *malware* nos serviços providos por sistemas computacionais. Foram elaborados dois diferentes cenários a fim de verificar a viabilidade desses ataques. O primeiro estudo de caso avaliou uma arquitetura básica de um DDoS.

Para isso, uma abordagem utilizando modelagem de árvore de ataque foi concebida para representar o comportamento de um ataque DDoS, além disso, levamos em consideração outras variáveis, como, probabilidade do ataque, custo com o ataque, benefícios com o ataque, facilidade do ataque, *pain factor*, propensão do ataque e habilidade técnica. Através dos resultados obtidos, contatamos que há uma chance de 92% do ataque ocorrer

e um custo relativo de US $87 para efetuar o ataque. Os resultados referente ao benefício com o ataque sucedeu em 10% de chance na obtenção de benefícios, com a utilização do ataque NTP e 5% de obtenção de benefícios utilizando ataques *Slowloris*, SNMP e *BitTorrent*.

Ainda assim, foi constatado uma facilidade do ataque utilizando TCP-SYN, *Hping3*, são as que mais podem impactar a vítima. Também houve um *pain factor* dado um ataque NTP MsSQL. Ataques do tipo *Slowloris* e SNMP possui uma maior propensão em ocorrer. O invasor possui habilidades técnicas com o ataque via TCP-SYN e *Push*+ACK. Essas são as ameaças que tendem a ter um maior impacto e ocorrência.

No segundo estudo de caso, foi avaliado um cenário de ataque DDoS e *malware*, onde verificamos o impacto ocasionado por ambos. A probabilidade de ataque foi de 97% de sucesso, e o custo com o ataque foi em torno de US $100. O invasor tem múltiplas chances em obter benefícios utilizando ataques via NTP, *Bating* e *SQL injection*. Os ataques via TCP-SYN, *push*+ACK e *ping of death* possuem maior facilidade, seguido de ICMP (Ping).

Constatamos que ataques via NTP possui maiores chances de acarretar problemas à vítima, seguido de *spear phishing*. Ataques *SQL injection* e SNMP são propenso a ocorrer mais que outros. Por último, o invasor possui maiores habilidades para efetuar ataques por meio TCP-SYN, push+ACK, ping of death, ICMP e SNMP.

Os resultados obtidos nesse trabalho podem nortear os administradores de rede no propósito de uma visão detalhada dos possíveis impactos que podem ocorrer em uma infraestrutura computacional através das ameaças aqui apresentadas. Este documento fornece uma visão geral e detalhada dos possíveis cenários de ataques DDoS e *malware*, auxiliando na tomada de decisões sobre implementação e políticas de prevenção. Foi possível constatar através da ferramenta *Shodan* (ver Apêndice C) a quantidade de dispositivos e protocolos vulneráveis existentes que, por sua vez, veio coincidir com o estudo em tese.

5.1 LIMITAÇÕES E TRABALHOS FUTUROS

Após a conclusão dessa pesquisa constatou-se que a mesma pode ser ampliada através de propostas futuras, como por exemplo, utilizar modelos de árvore de ataque para avaliar o impacto de ataques físicos e virtuais em sistemas de computação em nuvem, utilizando métricas de análise multivariada, bem como probabilidade de ataque, custo de ataque, benefícios do invasor, a propensão de ataque e viabilidade. Além disso, as análises de tendências, blockchain e de séries temporais podem ser usadas como meios para prevenir e mitigar o impacto de ataques DDoS.

Ainda assim, como o trabalho ora abordado considerou a árvore de ataque como modelagem, outros métodos de modelagem como, redes de petri e RBD se tornam uma abordagem importante. Objetivando também avaliar a degradação do *software* e *hardware*

através de medições MTTF e MTTR, como também verificar se há oscilação no consumo de energia dado um ataque DDoS.

REFERÊNCIAS

ACCENTURE. *Insights on the Security Investments that make a Difference*. 2017. <https://www.accenture.com/t20171006T095146Z___w___/us-en/_acnmedia/PDF-62/Accenture-2017CostCybercrime-US-FINAL.pdf>. Accessed: 2018-07-28.

ADAMSKY, F.; KHAYAM, S. A.; JÄGER, R.; RAJARAJAN, M. P2p file-sharing in hell: exploiting bittorrent vulnerabilities to launch distributed reflective dos attacks. 2015.

ALOMARI, E.; MANICKAM, S.; GUPTA, B.; KARUPPAYAH, S.; ALFARIS, R. Botnet-based distributed denial of service (ddos) attacks on web servers: classification and art. *arXiv preprint arXiv:1208.0403*, 2012.

ANGRISHI, K. Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets. *arXiv preprint arXiv:1702.03681*, 2017.

ARNLJOT, H.; RAUSAND, M. *System reliability theory: models and statistical methods*. [S.l.]: John Wiley & Sons, 2009. v. 420.

AVIŽIENIS, A.; LAPRIE, J.; RANDELL, B.; SCIENCE, U. of Newcastle upon T. C. *Fundamental Concepts of Dependability*. University of Newcastle upon Tyne, Computing Science, 2001. (Technical report series). Disponível em: <https://books.google.com.br/books?id=cDkmGwAACAAJ>.

AVIZIENIS, A.; LAPRIE, J.; RANDELL, B.; LANDWEHR, C. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, v. 1, p. 11–33, 2004.

BANIN, S.; DYRKOLBOTN, G. O. Multinomial malware classification via low-level features. *Digital Investigation*, v. 26, p. S107 – S117, 2018. ISSN 1742-2876. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1742287618301956>.

BARBOSA, E. D.; CASTRO, R. de O. Desenvolvimento de software seguro: Conhecendo e prevenindo ataques sql injection e cross-site scripting (xss). *Revista TIS*, v. 4, n. 1, 2016.

BELLOVIN, S. M. Security problems in the tcp/ip protocol suite. *ACM SIGCOMM Computer Communication Review*, ACM, v. 19, n. 2, p. 32–48, 1989.

BERTINO, E.; ISLAM, N. Botnets and internet of things security. *Computer*, IEEE, n. 2, p. 76–79, 2017.

BHUYAN, M. H.; BHATTACHARYYA, D.; KALITA, J. K. E-ldat: a lightweight system for ddos flooding attack detection and ip traceback using extended entropy metric. *Security and Communication Networks*, Wiley Online Library, v. 9, n. 16, p. 3251–3270, 2016.

BODENHEIM, R.; BUTTS, J.; DUNLAP, S.; MULLINS, B. Evaluation of the ability of the shodan search engine to identify internet-facing industrial control devices. *International Journal of Critical Infrastructure Protection*, Elsevier, v. 7, n. 2, p. 114–123, 2014.

BOESLER, M. *WORLD'S LARGEST BITCOIN EXCHANGE: We Are Suffering A Massive Attack On Our Servers. Blog.* 2013. <https://www.businessinsider.com/ddos-attack-on-mt-gox-bitcoin-servers-2013-4>.

BOLCH, G.; GREINER, S.; MEER, H. de; TRIVEDI, K. Modeling and performance evaluation with computer science application. In: *Queuing Networks and Markov Chains.* [S.l.: s.n.], 2006.

BORATEN, T.; KODI, A. Mitigation of hardware trojan based denial-of-service attack for secure nocs. *Journal of Parallel and Distributed Computing*, Elsevier, v. 111, p. 24–38, 2018.

BOTEZATU, B. *Hide and Seek IoT Botnet resurfaces with new tricks.* [S.l.]: Bitdefender LABS, 2018. <https://labs.bitdefender.com/2018/05/hide-and-seek-iot-botnet-resurfaces-with-new-tricks-persistence/>.

CENTER, C. C. *CERT advisory CA-1998-01 smurf IP denial-of-service attacks.* [S.l.]: January, 1998.

CERT, B. *Centro de Estudos, Respostas e Tratamento de Incidentes de Segurança no Brasil.* 2016. <https://www.cert.br/>.

CHAPARRO, F. *The platform is under extreme load: Cyber attack brings major cryptocurrency exchange to its knees. Blog.* 2018. <https://www.businessinsider.com/bitfinex-hit-by-cyber-attack-2018-6>.

CONTA, A.; DEERING, S.; GUPTA, M. *Internet control message protocol (icmpv6) for the internet protocol version 6 (ipv6) specification.* [S.l.], 2006.

COOKE, E.; JAHANIAN, F.; MCPHERSON, D. The zombie roundup: Understanding, detecting, and disrupting botnets. *SRUTI*, v. 5, p. 6–6, 2005.

CRISCUOLO, P. *Distributed Denial of Service Tools, Trin00, Tribe Flood Network, Tribe Flood Network 2000 and Stacheldraht.* [S.l.], 2000.

DAMON, E.; DALE, J.; LARON, E.; MACHE, J.; LAND, N.; WEISS, R. Hands-on denial of service lab exercises using slowloris and rudy. In: ACM. *proceedings of the 2012 information security curriculum development conference.* [S.l.], 2012. p. 21–29.

DANTAS, J. *Modelos para Analise de Dependabilidade de Arquiteturas de Computação em Nuvem.* Dissertação (Mestrado) — Centro de Informática - Universidade Federal de Pernambuco (Recife, Brasil), 2013.

DEFRAWY, K. E.; GJOKA, M.; MARKOPOULOU, A. Bottorrent: Misusing bittorrent to launch ddos attacks. *SRUTI*, v. 7, p. 1–6, 2007.

DILLON-MERRILL, R. L.; PARNELL, G.; BUCKSHAW, D. et al. Logic trees: Fault, success, attack, event, probability, and decision trees. *Wile y Handbook of Science and Technology for Homeland Security*, 2008.

DONNO, M. D.; DRAGONI, N.; GIARETTA, A.; SPOGNARDI, A. Analysis of ddos-capable iot malwares. In: IEEE. *Computer Science and Information Systems (FedCSIS), 2017 Federated Conference on.* [S.l.], 2017. p. 807–816.

DURFINA, L.; KROUSTEK, J.; ZEMEK, P. Psybot malware: A step-by-step decompilation case study. In: IEEE. *Reverse Engineering (WCRE), 2013 20th Working Conference on*. [S.l.], 2013. p. 449–456.

EDDY, W. *TCP SYN flooding attacks and common mitigations*. [S.l.], 2007.

EDGE, K.; DUBE, T. et al. A taxonomy of protection used in computer viruses and their application to software protection. In: *International Conference on Inforfmation Warfare (ICIW2006), University of Maryland Eastern Shore, USA*. [S.l.: s.n.], 2006.

EDGE, K.; RAINES, R.; BALDWIN, R.; GRIMAILA, M.; REUTER, C.; BENNINGTON, R. Analyzing security measures for mobile ad hoc networks using attack and protection trees. In: ACADEMIC CONFERENCES LIMITED. *ICIW2007-2nd International Conference on Information Warfare & Security: ICIW2007*. [S.l.], 2007. p. 47.

EDGE, K.; RAINES, R.; GRIMAILA, M.; BALDWIN, R.; BENNINGTON, R.; REUTER, C. The use of attack and protection trees to analyze security for an online banking system. In: IEEE. *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*. [S.l.], 2007. p. 144b–144b.

EDGE, K. S. *A framework for analyzing and mitigating the vulnerabilities of complex systems via attack and protection trees*. [S.l.], 2007.

EDGE, K. S.; DALTON, G. C.; RAINES, R. A.; MILLS, R. F. Using attack and protection trees to analyze threats and defenses to homeland security. In: IEEE. *Military Communications Conference, 2006. MILCOM 2006. IEEE*. [S.l.], 2006. p. 1–7.

EPOH, J. C. E. Techniques for detecting, preventing and mitigating distributed denial of service (ddos) attacks. In: *Information Technology-New Generations*. [S.l.]: Springer, 2018. p. 899–904.

ERICKSON, J. *Hacking: the art of exploitation*. [S.l.]: No starch press, 2008.

FAZZI, F. *Lightaidra. GitHub*. 2016. <https://github.com/eurialo/lightaidra>.

FELTEN, E. W.; BALFANZ, D.; DEAN, D.; WALLACH, D. S. Web spoofing: An internet con game. *Software World*, v. 28, n. 2, p. 6–8, 1997.

GAMUNDANI, A. M.; NEKARE, L. M. A review of new trends in cyber attacks: A zoom into distributed database systems. In: IEEE. *2018 IST-Africa Week Conference (IST-Africa)*. [S.l.], 2018. p. Page–1.

GARVEY, P. R. Track 2: Implementing a risk management process for a large scale information system upgrade—a case study. *Insight*, Wiley Online Library, v. 4, n. 1, p. 15–22, 2001.

HILTON, S. *Dyn Analysis Summary Of Friday October 21 Attack. Blog*. 2016. <https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/>.

HONG, J. The state of phishing attacks. *Communications of the ACM*, ACM, v. 55, n. 1, p. 74–81, 2012.

HOQUE, N.; KASHYAP, H.; BHATTACHARYYA, D. Real-time ddos attack detection using fpga. *Computer Communications*, Elsevier, v. 110, p. 48–58, 2017.

HUSSAIN, A.; HEIDEMANN, J.; PAPADOPOULOS, C. A framework for classifying denial of service attacks. In: ACM. *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. [S.l.], 2003. p. 99–110.

HUSSAIN, S. M.; BEIGH, G. R. Impact of ddos attack (udp flooding) on queuing models. In: IEEE. *Computer and Communication Technology (ICCCT), 2013 4th International Conference on*. [S.l.], 2013. p. 210–216.

IBRAGIMOV, T.; KUPREEV, O.; EKATERINA, B.; GUTNIKOV, A. *DDoS attacks in Q2 2018*. [S.l.]: Securelist, 2018. <https://securelist.com/ddos-report-in-q2-2018/86537/>.

IEEE. Ieee standard glossary of software engineering terminology. In: *IEEE Std*. [S.l.]: IEEE Computer Society, 1990.

INGOLDSBY, T. R. Understanding risks through attack tree analysis. *Computer Security Journal*, COMPUTER SECURITY INSTITUTE, v. 20, n. 2, p. 33–59, 2004.

INGOLDSBY, T. R. Attack tree-based threat risk analysis. *Amenaza Technologies Limited*, p. 3–9, 2010.

ISACA, C. Review manual 2017. *Information Systems Audit and Control Association Inc., ZDA*, 2017.

JACKSON, C.; BARTH, A.; BORTZ, A.; SHAO, W.; BONEH, D. Protecting browsers from dns rebinding attacks. *ACM Transactions on the Web (TWEB)*, ACM, v. 3, n. 1, p. 2, 2009.

JANUS, M. *Heads of the Hydra. Malware for Network Devices. Securelist*. 2011. <https://securelist.com/heads-of-the-hydra-malware-for-network-devices/36396/>.

KIM, M.-S.; KONG, H.-J.; HONG, S.-C.; CHUNG, S.-H.; HONG, J. W. A flow-based method for abnormal network traffic detection. In: IEEE. *Network operations and management symposium, 2004. NOMS 2004. IEEE/IFIP*. [S.l.], 2004. v. 1, p. 599–612.

KOTENKO, I.; SAENKO, I.; LAUTA, O. Modeling the impact of cyber attacks. In: *Cyber Resilience of Systems and Networks*. [S.l.]: Springer, 2019. p. 135–169.

KRUPP, J.; BACKES, M.; ROSSOW, C. Identifying the scan and attack infrastructures behind amplification ddos attacks. In: ACM. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. [S.l.], 2016. p. 1426–1437.

KUO, W.; ZUO, M. *Optimal Reliability Modeling: Principles and Applications*. Wiley, 2003. ISBN 9780471275459. Disponível em: <https://books.google.com.br/books?id=vdZ4Bm-LnHMC>.

LE, Q.; BOYDELL, O.; NAMEE, B. M.; SCANLON, M. Deep learning at the shallow end: Malware classification for non-domain experts. *Digital Investigation*, v. 26, p. S118 – S126, 2018. ISSN 1742-2876. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1742287618302032>.

LEE, K.; KIM, J.; KWON, K. H.; HAN, Y.; KIM, S. Ddos attack detection method using cluster analysis. *Expert systems with applications*, Elsevier, v. 34, n. 3, p. 1659–1665, 2008.

LÉVY-BENCHETON, C.; MARINOS, L.; MATTIOLI, R.; KING, T.; DIETZEL, C.; STUMPF, J. Threat landscape and good practice guide for internet infrastructure. *Report, European Union Agency for Network and Information Security (ENISA)*, 2015.

LONG, N.; THOMAS, R. Trends in denial of service attack technology. *CERT Coordination Center*, p. 648–651, 2001.

LYU, M.; SHERRATT, D.; SIVANATHAN, A.; GHARAKHEILI, H. H.; RADFORD, A.; SIVARAMAN, V. Quantifying the reflective ddos attack capability of household iot devices. In: ACM. *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. [S.l.], 2017. p. 46–51.

MACIEL, P.; TRIVEDI, K.; MATIAS, R.; KIM, D. Dependability modeling. In: *Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions*. [S.l.: s.n.], 2011.

MACIEL, R.; ARAUJO, J.; DANTAS, J.; MELO, C.; GUEDES, E.; MACIEL, P. Impact of a ddos attack on computer systems: An approach based on an attack tree model. In: IEEE. *Systems Conference (SysCon), 2018 Annual IEEE International*. [S.l.], 2018. p. 1–8.

MALWAREMUSTDIE. *IRC Botnet. Blog*. 2016. <http://blog.malwaremustdie.org/2016/02/mmd-0052-2016-skidddos-elf-distribution.html>.

MANSFIELD-DEVINE, S. The growth and evolution of ddos. *Network Security*, Elsevier, v. 2015, n. 10, p. 13–20, 2015.

MANSFIELD-DEVINE, S. Ddos goes mainstream: how headline-grabbing attacks could make this threat an organisation's biggest nightmare. *Network Security*, Elsevier, v. 2016, n. 11, p. 7–13, 2016.

MARINOFF, N. *Verge Cryptocurrency Suffers Its Second Hack in Less Than Two Months. Blog*. 2018. <https://bitcoinmagazine.com/articles/verge-cryptocurrency-suffers-its-second-hack-less-two-months/>.

MATHERLY, J. Complete guide to shodan. *Shodan, LLC (2016-02-25)*, 2016.

MAUW, S.; OOSTDIJK, M. Foundations of attack trees. In: SPRINGER. *International Conference on Information Security and Cryptology*. [S.l.], 2005. p. 186–198.

MAZUR, J. *Fluke: The Math and Myth of Coincidence*. [S.l.]: Basic Books, 2016.

MILLMAN, R. *OVH suffers 1.1Tbps DDoS attack. Blog*. 2016. <https://www.scmagazineuk.com/ovh-suffers-11tbps-ddos-attack/article/1476220>.

MIRKOVIC, J.; REIHER, P. A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communication Review*, ACM, v. 34, n. 2, p. 39–53, 2004.

MORGAN, S. *Ransomware damage costs predicted to hit $11.5B by 2019*. 2017. <https://www.csoonline.com/article/3237674/ransomware/ransomware-damage-costs-predicted-to-hit-115b-by-2019.html>. Accessed: 2018-07-28.

MOYERS, B. R.; DUNNING, J. P.; MARCHANY, R. C.; TRONT, J. G. Effects of wi-fi and bluetooth battery exhaustion attacks on mobile devices. In: IEEE. *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*. [S.l.], 2010. p. 1–9.

NDIBWILE, J. D.; GOVARDHAN, A.; OKADA, K.; KADOBAYASHI, Y. Web server protection against application layer ddos attacks using machine learning and traffic authentication. In: IEEE. *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*. [S.l.], 2015. v. 3, p. 261–267.

NEWMAN, L. *Github survived the biggest DDoS attack ever recorded*. 2018. <https://www.wired.com/story/github-ddos-memcached/>. Accessed: 2018-02-02.

NG, T. E.; ZHANG, H. Predicting internet network distance with coordinates-based approaches. In: IEEE. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. [S.l.], 2002. v. 1, p. 170–179.

OSANAIYE, O.; CHOO, K.-K. R.; DLODLO, M. Distributed denial of service (ddos) resilience in cloud: review and conceptual cloud ddos mitigation framework. *Journal of Network and Computer Applications*, Elsevier, v. 67, p. 147–165, 2016.

PAULI, D. *Hydra hacker bot spawns internet of things DDoS clones. The Register*. 2016. <https://www.theregister.co.uk/2016/07/01/lizardstresser_ddos/>.

PIETERS, W.; DAVARYNEJAD, M. Calculating adversarial risk from attack trees: Control strength and probabilistic attackers. In: *Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance*. [S.l.]: Springer, 2015. p. 201–215.

POWER, R. *2002 CSI/FBI computer crime and security survey*. [S.l.]: Computer Security Institute, 2002.

RADHA, S. et al. A novel method for detection and elimination of modification attack and ttl attack in ntp based routing algorithm. In: IEEE. *Recent Trends in Information, Telecommunication and Computing (ITC), 2010 International Conference on*. [S.l.], 2010. p. 60–64.

RAUSAND, M.; ARNLJOT, H. *System reliability theory: models, statistical methods, and applications*. [S.l.]: John Wiley & Sons, 2004. v. 396.

REGALADO, A. *Who Coined Cloud Computing?* 2011. <http://www.technologyreview.com/news/425970/who-coined-cloud-computing/>. [Online; accessed 21-December-2013].

ROSSOW, C. Amplification hell: Revisiting network protocols for ddos abuse. In: *NDSS*. [S.l.: s.n.], 2014.

ROY, A. *Attack countermeasure trees: A non-state-space approach towards analyzing security and finding optimal countermeasure sets*. Tese (Doutorado) — Duke University, 2010.

ROY, A.; KIM, D. S.; TRIVEDI, K. S. Attack countermeasure trees (act): towards unifying the constructs of attack and defense trees. *Security and Communication Networks*, Wiley Online Library, v. 5, n. 8, p. 929–943, 2012.

SALTER, C.; SAYDJARI, O. S.; SCHNEIER, B.; WALLNER, J. Toward a secure system engineering methodolgy. In: ACM. *Proceedings of the 1998 workshop on New security paradigms*. [S.l.], 1998. p. 2–10.

SANTANNA, J. J.; RIJSWIJK-DEIJ, R. van; HOFSTEDE, R.; SPEROTTO, A.; WIERBOSCH, M.; GRANVILLE, L. Z.; PRAS, A. Booters—an analysis of ddos-as-a-service attacks. In: IEEE. *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. [S.l.], 2015. p. 243–251.

SARIPALLI, P.; WALTERS, B. Quirc: A quantitative impact and risk assessment framework for cloud security. In: IEEE. *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*. [S.l.], 2010. p. 280–288.

SCHNEIER, B. Attack trees. *Dr. Dobb's journal*, v. 24, n. 12, p. 21–29, 1999.

SCHNEIER, B.; SHOSTACK, A. et al. Breaking up is hard to do: modeling security threats for smart cards. In: *USENIX Workshop on Smart Card Technology, Chicago, Illinois, USA, http://www. counterpane. com/smart-card-threats. html*. [S.l.: s.n.], 1999.

SCHUBA, C. L.; KRSUL, I. V.; KUHN, M. G.; SPAFFORD, E. H.; SUNDARAM, A.; ZAMBONI, D. Analysis of a denial of service attack on tcp. In: IEEE. *Security and Privacy, 1997. Proceedings., 1997 IEEE Symposium on*. [S.l.], 1997. p. 208–223.

SEKAR, V.; DUFFIELD, N. G.; SPATSCHECK, O.; MERWE, J. E. van der; ZHANG, H. Lads: Large-scale automated ddos detection system. In: *USENIX Annual Technical Conference, General Track*. [S.l.: s.n.], 2006. p. 171–184.

SHAN-SHAN, J.; YA-BIN, X. The apt detection method based on attack tree for sdn. In: ACM. *Proceedings of the 2nd International Conference on Cryptography, Security and Privacy*. [S.l.], 2018. p. 116–121.

SIKORSKI, M.; HONIG, A. *Practical malware analysis: the hands-on guide to dissecting malicious software*. [S.l.]: no starch press, 2012.

SINGH, K. J.; DE, T. Mlp-ga based algorithm to detect application layer ddos attack. *Journal of Information Security and Applications*, Elsevier, v. 36, p. 145–153, 2017.

SPECHT, S. M.; LEE, R. B. Distributed denial of service: Taxonomies of attacks, tools, and countermeasures. In: *ISCA International Conference on Parallel and Distributed Computing (and Communications) Systems*. [S.l.: s.n.], 2004. p. 543–550.

STONEBURNER, G.; GOGUEN, A. Y.; FERINGA, A. Sp 800-30. risk management guide for information technology systems. National Institute of Standards & Technology, 2002.

SZOR, P. *The art of computer virus research and defense*. [S.l.]: Pearson Education, 2005.

TERRANCE, I. R. *Amenaza Technologies*. 2017. Disponível em: <https://www.amenaza.com/>.

THOMPSON, R. Why spyware poses multiple threats to security. *Communications of the ACM*, ACM, v. 48, n. 8, p. 41–43, 2005.

TRIVEDI, K. S.; KIM, D. S.; ROY, A.; MEDHI, D. Dependability and security models. In: IEEE. *Design of Reliable Communication Networks, 2009. DRCN 2009. 7th International Workshop on*. [S.l.], 2009. p. 11–20.

TSUCHIYA, A.; FRAILE, F.; KOSHIJIMA, I.; ORTIZ, A.; POLER, R. Software defined networking firewall for industry 4.0 manufacturing systems. *Journal of Industrial Engineering and Management*, v. 11, n. 2, p. 318–333, 2018.

VERBA, J.; MILVICH, M. Idaho national laboratory supervisory control and data acquisition intrusion detection system (scada ids). In: IEEE. *Technologies for Homeland Security, 2008 IEEE Conference on*. [S.l.], 2008. p. 469–473.

VERISIGN. *Distributed Denial of Service Trends Report. Available in:*. 2018. <https://www.verisign.com/assets/report-ddos-trends-Q12018.pdf>.

VIVINSANDAR, S.; SHENAI, S. Economic denial of sustainability (edos) in cloud services using http and xml based ddos attacks. *International Journal of Computer Applications*, Foundation of Computer Science, v. 41, n. 20, 2012.

WANG, J.; PHAN, R. C.-W.; WHITLEY, J. N.; PARISH, D. J. Augmented attack tree modeling of distributed denial of services and tree based attack detection method. In: IEEE. *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*. [S.l.], 2010. p. 1009–1014.

WANG, T.-S.; LIN, H.-T.; CHENG, W.-T.; CHEN, C.-Y. Dbod: Clustering and detecting dga-based botnets using dns traffic analysis. *Computers & Security*, Elsevier, v. 64, p. 1–15, 2017.

WEISS, J. D. A system security engineering process. In: *Proceedings of the 14th National Computer Security Conference*. [S.l.: s.n.], 1991. v. 249, p. 572–581.

WOOD, A. D.; STANKOVIC, J. A. Denial of service in sensor networks. *computer*, IEEE, v. 35, n. 10, p. 54–62, 2002.

WOOD, P.; NAHORNEY, B.; CHANDRASEKAR, K.; WALLACE, S.; HALEY, K. Internet security threat report. *Symantec*, v. 17, 2016.

XIANG, Y.; LI, K.; ZHOU, W. Low-rate ddos attacks detection and traceback by using new information metrics. *IEEE transactions on information forensics and security*, IEEE, v. 6, n. 2, p. 426–437, 2011.

XIAO, P.; QU, W.; QI, H.; LI, Z. Detecting ddos attacks against data center with correlation analysis. *Computer Communications*, Elsevier, v. 67, p. 66–74, 2015.

YE, X. Countering ddos and xdos attacks against web services. In: IEEE. *Embedded and Ubiquitous Computing, 2008. EUC'08. IEEE/IFIP International Conference on*. [S.l.], 2008. v. 1, p. 346–352.

YORK, K. *Read Dyn's Statement on the 10/21/2016 DNS DDoS Attack. Blog*. 2016. <https://dyn.com/blog/dyn-statement-on-10212016-ddos-attack/>.

ZARGAR, S. T.; JOSHI, J.; TIPPER, D. A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. *IEEE communications surveys & tutorials*, IEEE, v. 15, n. 4, p. 2046–2069, 2013.

APÊNDICE A – FERRAMENTA *SHODAN*

Há diversos mecanismos de pesquisa na *Web*, *Google*, *Bing* e *Yahoo*, são excelente para encontrar sites. No entanto, existe o *Shodan* (BODENHEIM et al., 2014), o mesmo possui um mecanismo de busca para dispositivos conectados à Internet, capaz de fornece aos usuários endereços IP, nomes de *hosts*, informações de domínio, informações geográficas de um determinado IP, portas e quais os serviços funcionando na mesma, sistemas operacionais, até mesmo vulnerabilidades que acabará de ser descoberta (MATHERLY, 2016). Esses são alguns dos resultados que o mecanismo de busca traz para o usuário (TSUCHIYA et al., 2018). O *banner* de informações possuí diversos informes, senhas, nomes de usuário, vulnerabilidade associada ao sistema, localização e dentre outros (BODENHEIM et al., 2014). A Figura 26 mostra a lógica de varredura da ferramenta, primeiro o *Shodan* gera um endereço de IP aleatório e uma porta de serviço, em seguida envia um SYN. Se o SYN sinalizar com o ACK é sinalizado como sucesso, logo após, captura informações do oponente e cria um *banner*. Depois, armazena todas essas informações em um banco de dados e repete essa verificação para todos os endereços IP no intervalo fornecido, o *script* de um *banner* (ver Apêndice B).

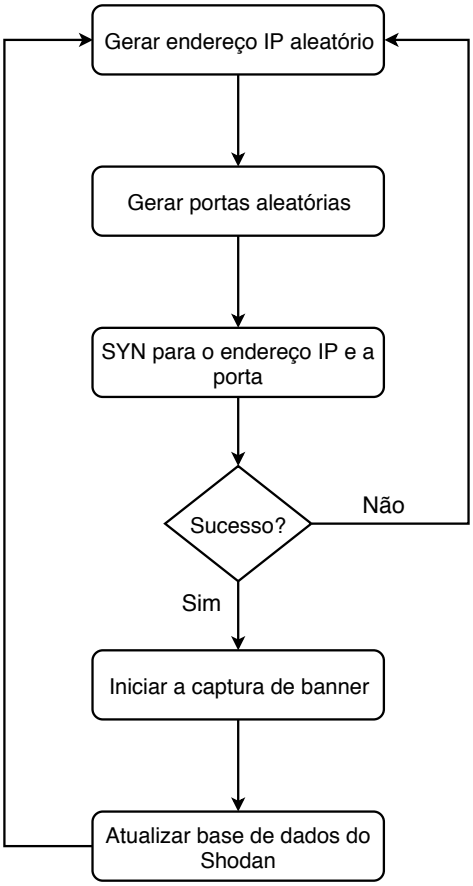


Figura 26 – Mecanismo de busca do Shodan

APÊNDICE B – BANNER SSL SHODAN

Listing B.1 – Especificação de um Banner SSL no Shodan


```
{
   "timestamp": "2014-01-16T08:37:40.081917",
   "hostnames": [
      "99-46-189-78.lightspeed.tukrga.sbcglobal.net"
   ],
   "org": "AT&T U-verse",
   "guid": "1664007502:75a821e2-7e89-11e3-8080-808080808080",
   "data": "NTP\nxxx.xxx.xxx.xxx:7546\n68.94.157.2:123\n68.94.156.17:123",
   "port": 123,
   "isp": "AT&T U-verse",
   "asn": "AS7018",
   "location": {
      "country_code3": "USA",
      "city": "Atlanta",
      "postal_code": "30328",
      "longitude": -84.3972,
      "country_code": "US",
      "latitude": 33.93350000000001,
      "country_name": "United States",
      "area_code": 404,
      "dma_code": 524,
      "region_code": null
   },
   "ip": 1664007502,
   "domains": [
      "sbcglobal.net"
   ],
   "ip_str": "99.46.189.78",
   "os": null,
   "opts": {
      "raw": "\\x97\\x00\\x03*\\x00\\x03\\x00H\\x00\\x00\\x00\\x00\\x00\\x00\\
         x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x01G\\x06\\xa7\\x8ec.\\xbdN\\x00\\
         x00\\x00\\x01\\x1dz\\x07\\x02\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00
         \\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\
         x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00
         \\x00\\x00\\x00\\x00\\x00q\\x00\\x00\\x00i\\x00\\x00\\x00\\x00\\x00\\x00\\
         x00XD^\\x9d\\x02c.\\xbdN\\x00\\x00\\x00\\x01\\x00{\\x04\\x04\\x00\\x00\\
         x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00
         \\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\
         x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00q\\x00\\x00\\
         x00o\\x00\\x00\\x00\\x00\\x00\\x00\\x00YD^\\x9c\\x11c.\\xbdN\\x00\\x00\\
         x00\\x01\\x00{\\x04\\x04\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00
         \\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\
         x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00
         \\x00",
      "ntp": {
         "more": false
      }
   }
}
```

APÊNDICE C – UTILIZAÇÃO DA FERRAMENTA *SHODAN* PARA LOCALIZAR DISPOSITIVOS VULNERÁVEIS

Este apêndice apresenta o mecanismo da ferramenta *Shodan* para busca de dispositivos que podem compor uma rede Botnet. Salientamos que o método de busca utilizado tem o único e restrito objetivo acadêmico. Observamos vários dispositivos e os respectivos protocolos por eles utilizados em nível mundial.

O *banner* é um resultado da busca no *Shodan* que descreve um serviço em um dispositivo, o conteúdo do *banner* varia dependendo do tipo de serviço. Por exemplo, abaixo está um *banner* HTTP típico:

HTTP/1.1 200 OK
Server: Router Webserver
Connection: close
Content-Type: text/html
www-authenticate: Basic realm="TP-LINK Wireless N Router WR840N"

O *banner* acima mostra que o dispositivo é um roteador TP-LINK *Wireless*, com uma autenticação básica.

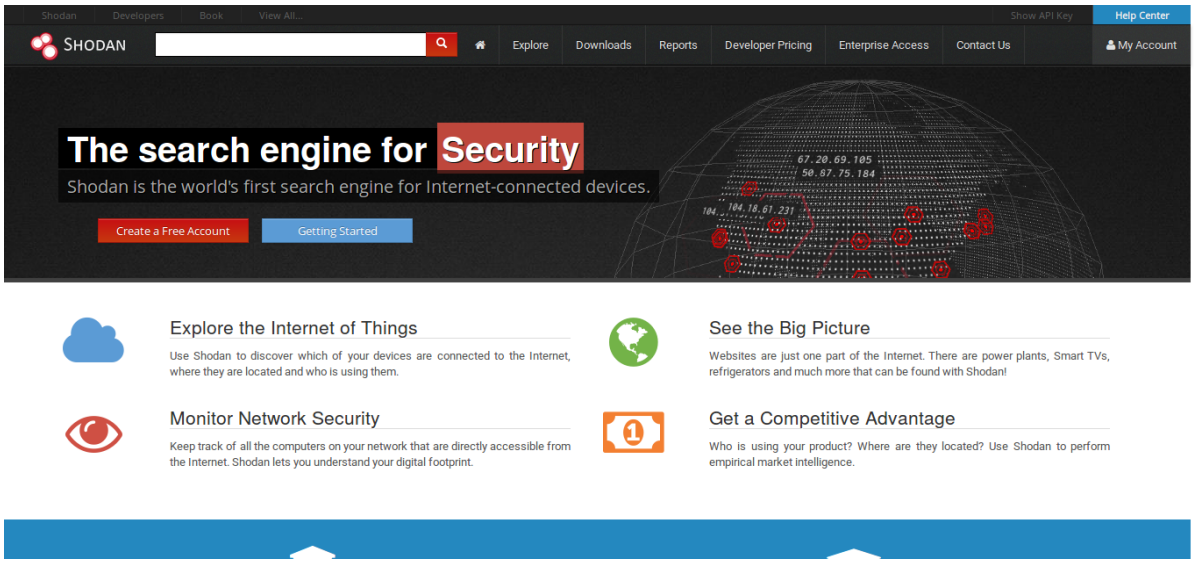


Figura 27 – Página inicial da ferramenta *Shodan*

Ainda assim, as pesquisas feitas no *Shodan* examinará os dados coletados nos últimos 30 dias. Os resultados obtidos no site fornecem uma visão precisa no momento da pesquisa. Além de pesquisar, o site também fornece as seguintes funcionalidades:

Download de dados, possível efetuar o *download* desses dados, há um botão na parte superior denominado *download* de dados. Os resultados da pesquisa nos fornecem os seguintes formatos JSON, CSV ou XML para *download*.

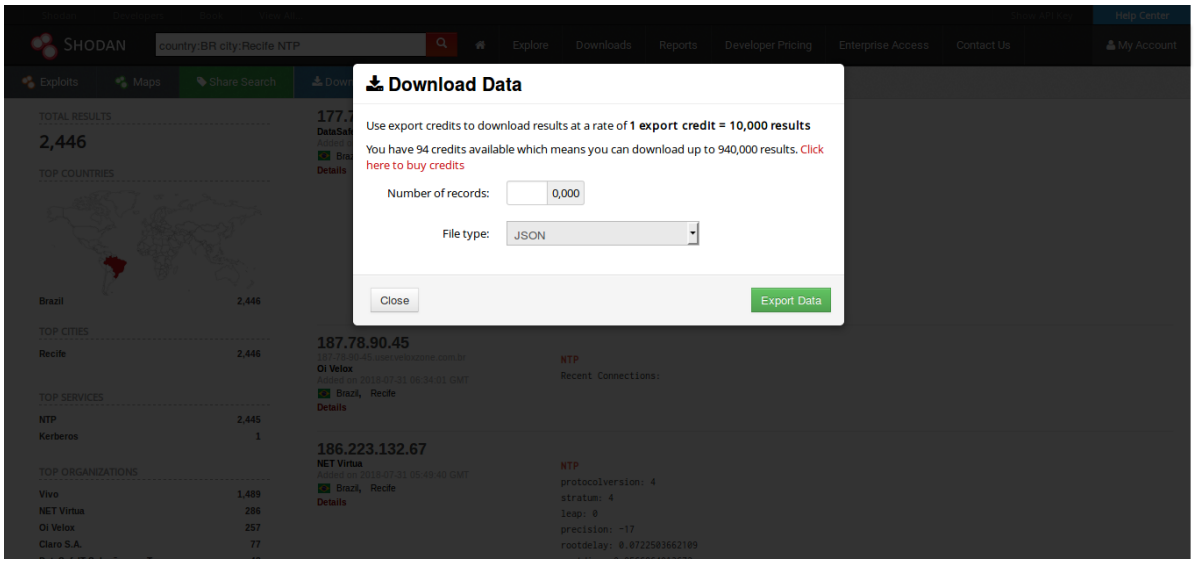


Figura 28 – Página de *download*

Agora, mostraremos como efetuar buscas de alguns dispositivos e ou protocolos na rede. Utilizamos *String* para refinar as pesquisas, algumas delas são listadas abaixo:

Tabela 7 – Descrição dos métodos de busca com *Shodan*

Nome	Descrição	Exemplo
asn	*Autonomous system number*	AS4837
ip	Endereço IP como inteiro	493427495
port	Número da porta	80
protocol	Nomes dos protocolos	NTP, HTTP
city	Nome da cidade	Ashburn
country	Nome completo do país	*United States*
latitude	Latitude dos dispositivos	21.03330
longitude	Longitude dos dispositivos	105.85000
os	Sistema operacional	Linux, Windows

country:BR city:Brasilia NTP
country:BR city:Recife DNS
country:BR city:Curitiba HTTP
country:BR linux upnp avtech
ip country:BR

Dentre as diversas combinações de busca por dispositivos conectados à rede, esses são exemplos básicos, onde é possível encontrar protocolos NTP, DNS e HTTP, como também dispositivos IoT. Abaixo listamos um exemplo da quantidade de protocolos NTP em funcionamento nas regiões Brasileiras, também, listamos as principais regiões do Brasileiras com câmeras Avtech em funcionamento.

Figura 29 – Lista de cidades no Brasil com protocolo NTP

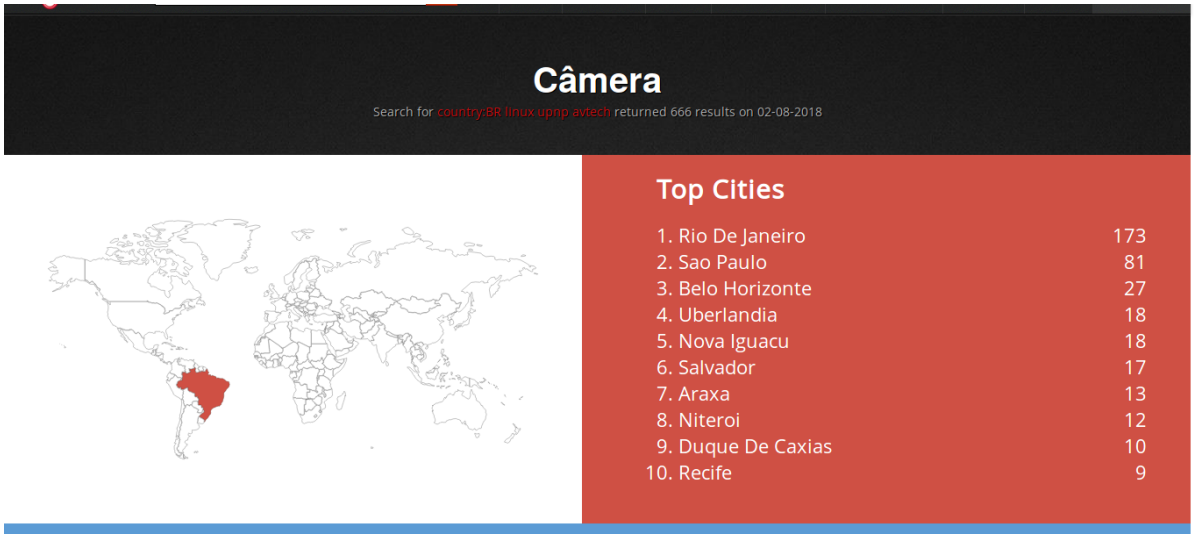


Figura 30 – Lista de cidades no Brasil com câmeras Avtech

Contudo, a *dashboard*[1] principal (ver Figura 31), exibe a quantidade de dispositivos TP-LINK a nível mundial, ainda assim, é possível observar os endereços IP atrelado a cada país e os protocolos que os mesmos operam.

[1] São painéis que mostram métricas e indicadores importantes para alcançar objetivos e metas traçadas de forma visual, facilitando a compreensão das informações.

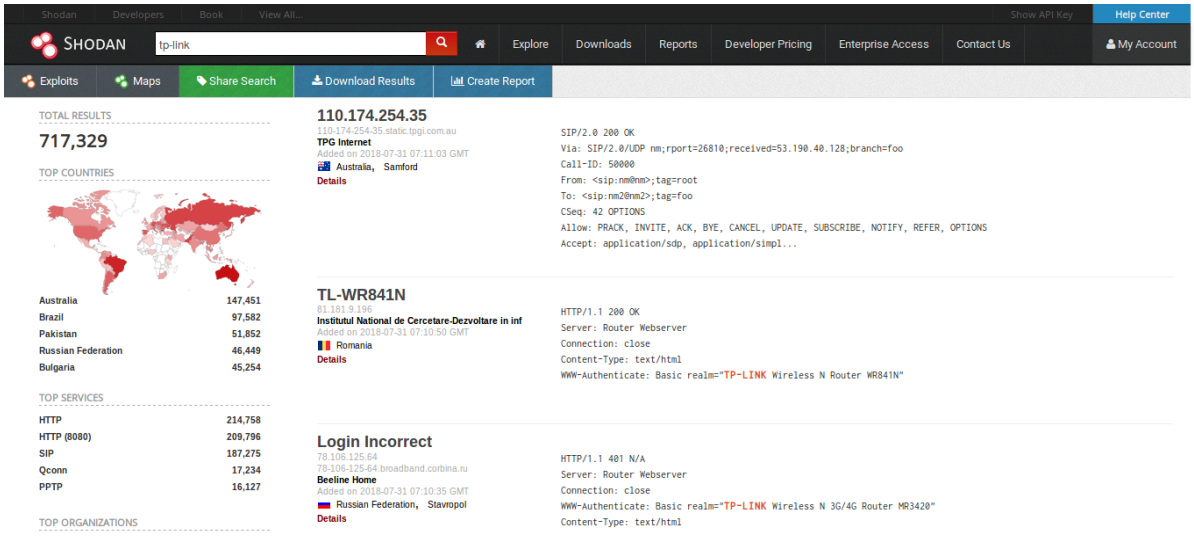


Figura 31 – Lista de roteadores TP-LINK em todo o mundo

Afinal, o objetivo do estudo de caso em tese, visou confirmar a quantidade de dispositivos e protocolos conetados a rede que possam vir a ser tornar Bot e assim serem utilizados para efetuar ataques DDoS nos sistemas computacionais.

APÊNDICE D – LISTA DE PORTAS

Tabela 8 – Lista de portas escaneadas pelo Shodan

Porta	Serviço(s)
7	Echo
11	Systat
13	Daytime
15	Netstat
17	Quote of the day
19	Character generator
21	FTP
22	SSH
23	Telnet
25	SMTP
26	SSH
37	rdate
49	TACACS+
53	DNS
67	DHCP
69	TFTP, BitTorrent
79	Finger
80	HTTP, malware
81	HTTP, malware
82	HTTP, malware
83	HTTP
84	HTTP
88	Kerberos
102	Siemens S7
110	POP3
111	Portmapper

Tabela 8 Continua: Lista de portas

119	NNTP
123	NTP
129	Password generator protocol
137	NetBIOS
143	IMAP
161	SNMP
175	IBM Network Job Entry
179	BGP
195	TA14-353a
311	OS X Server Manager
389	LDAP
443	HTTPS
444	TA14-353a, Dell SonicWALL
445	SMB
465	SMTPS
500	IKE (VPN)
502	Modbus
503	Modbus
515	Line Printer Daemon
520	RIP
523	IBM DB2
554	RTSP
587	SMTP mail submission
623	IPMI
626	OS X serialnumbered
666	Telnet
771	Realport
789	Redlion Crimson3
873	rsync
902	VMWare authentication
992	Telnet (secure)

Tabela 8 Continua: Lista de portas

993	IMAP with SSL
995	POP3 with SSL
1010	malware
1023	Telnet
1025	Kamstrup
1099	Java RMI
1177	malware
1200	Codesys
1234	udpxy
1434	MS-SQL monitor
1604	Citrix, malware
1723	PPTP
1833	MQTT
1900	UPnP
1911	Niagara Fox
1962	PCworx
1991	malware
2000	iKettle, MikroTik bandwidth test
2082	cPanel
2083	cPanel
2086	WHM
2087	WHM
2123	GTPv1
2152	GTPv1
2181	Apache Zookeeper
2222	SSH, PLC5, EtherNet/IP
2323	Telnet
2332	Sierra wireless (Telnet)
2375	Docker
2376	Docker
2404	IEC-104

Tabela 8 Continua: Lista de portas

2455	CoDeSys
2480	OrientDB
2628	Dictionary
3000	ntop
3306	MySQL
3386	GTPv1
3388	RDP
3389	RDP
3460	malware
3541	PBX GUI
3542	PBX GUI
3689	DACP
3780	Metasploit
3787	Ventrilo
4000	malware
4022	udpxy
4040	Deprecated Chef web interface
4063	ZeroC Glacier2
4064	ZeroC Glacier2 with SSL
4369	EPMD
4443	Symantec Data Center Security
4444	malware
4500	IKE NAT-T (VPN)
4567	Modem web interface
4911	Niagara Fox with SSL
4949	Munin
5006	MELSEC-Q
5007	MELSEC-Q
5008	NetMobility
5009	Apple Airport Administration
5060	SIP

Tabela 8 Continua: Lista de portas

5094	HART-IP
5222	XMPP
5269	XMPP Server-to-Server
5353	mDNS
5357	Microsoft-HTTPAPI/2.0
5432	PostgreSQL
5577	Flux LED
5632	PCAnywhere
5672	RabbitMQ
5900	VNC
5901	VNC
5984	CouchDB
6000	X11
6379	Redis
6666	Voldemort database, malware
6667	IRC
6881	BitTorrent DHT
6969	TFTP, BitTorrent
7218	Sierra wireless (Telnet)
7474	Neo4j database
7548	CWMP (HTTPS)
7777	Oracle
7779	Dell Service Tag API
8010	Intelbras DVR
8060	Roku web interface
8069	OpenERP
8087	Riak
8090	Insteon HUB
8099	Yahoo SmartTV
8112	Deluge (HTTP)
8139	Puppet agent

Tabela 8 Continua: Lista de portas

8140	Puppet master
8181	GlassFish Server (HTTPS)
8333	Bitcoin
8334	Bitcoin node dashboard (HTTP)
8443	HTTPS
8554	RTSP
8880	Websphere SOAP
8888	HTTP, Andromouse
8889	SmartThings Remote Access
9001	Tor OR
9002	Tor OR
9051	Tor Control
9100	Printer Job Language
9151	Tor Control
9160	Apache Cassandra
9191	Sierra wireless (HTTP)
9443	Sierra wireless (HTTPS)
9595	LANDesk Management Agent
9600	OMRON
10001	Automated Tank Gauge
10243	Microsoft-HTTPAPI/2.0
11211	Memcache
17185	VxWorks WDBRPC
12345	Sierra wireless (Telnet)
13579	Media player classic web interface
14147	Filezilla FTP
16010	Apache Hbase
18245	General Electric SRTP
20000	DNP3
20547	ProconOS
21025	Starbound

Tabela 8 Continua: Lista de portas

21379	Matrikon OPC
23023	Telnet
23424	Serviio
25105	Insteon Hub
25565	Minecraft
27015	Steam A2S server query, Steam RCon
27017	MongoDB
28017	MongoDB (HTTP)
30718	Lantronix Setup
32400	Plex
37777	Dahuva DVR
44818	EtherNet/IP
47808	Bacnet
49152	Supermicro (HTTP)
49153	WeMo Link
50070	HDFS Namenode
51106	Deluge (HTTP)
54138	Toshiba PoS
55553	Metasploit
55554	Metasploit
62078	Apple iDevice
64738	Mumble