

**UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"**

FACULDADE DE CIÊNCIAS - CAMPUS BAURU

DEPARTAMENTO DE COMPUTAÇÃO

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

MATEUS ALVES BERTONI

**UM ESTUDO SOBRE FERRAMENTA *OPEN-SOURCE* DE  
*HONEYPOT***

BAURU

2017

MATEUS ALVES BERTONI

**UM ESTUDO SOBRE FERRAMENTA *OPEN-SOURCE* DE  
*HONEYPOT***

Trabalho de Conclusão de Curso do Curso  
de Ciência da Computação da Universidade  
Estadual Paulista “Júlio de Mesquita Filho”,  
Faculdade de Ciências, Campus Bauru.  
Orientador: Prof. Dr. Kleber Rocha de Oliveira

BAURU  
2017

Mateus Alves Bertoni      UM ESTUDO SOBRE FERRAMENTA *OPEN-SOURCE* DE *HONEY-POT* / Mateus Alves Bertoni. – Bauru, 2017-      63 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Kleber Rocha de Oliveira

Trabalho de Conclusão de Curso – Universidade Estadual Paulista “Júlio de Mesquita Filho”

Faculdade de Ciências

Ciência da Computação, 2017.

1. Segurança da Informação 2. Honeypots 3. Malware 4. Problemas de invasão 5. Artillery 6. *Honeypot* de baixa-interatividade 7. *Open-Source*

Mateus Alves Bertoni

# **UM ESTUDO SOBRE FERRAMENTA *OPEN-SOURCE* DE *HONEYPOT***

Trabalho de Conclusão de Curso do Curso de  
Ciência da Computação da Universidade Esta-  
dual Paulista "Júlio de Mesquita Filho", Facul-  
dade de Ciências, Campus Bauru.

Banca Examinadora

**Prof. Dr. Kleber Rocha de Oliveira**

Orientador

Universidade Estadual Paulista "Júlio de Mesquita Filho"

Faculdade de Ciências

Departamento de Computação

**Profa. Dra. Simone Domingues Prado**

Universidade Estadual Paulista "Júlio de Mesquita Filho"

Faculdade de Ciências

Departamento de Computação

**Prof. Dr. Kelton Augusto Pontara da Costa**

Universidade Estadual Paulista "Júlio de Mesquita Filho"

Faculdade de Ciências

Departamento de Computação

Bauru, 08 de dezembro de 2017.

*Este trabalho é dedicado à ciência que, em meio a todo caos, consegue ser capaz de unir  
indivíduos totalmente distintos em prol de um bem maior*

# Agradecimentos

Em primeiro lugar, gostaria de agradecer aos meus pais, que sempre forneceram todo o suporte que precisei, me apoiaram nas decisões que tomei e sempre estiveram presentes em minha vida. Também agradeço aos meus falecidos avôs e avós, que me ensinaram a ter a humildade, honestidade, força e fé para alcançar tudo que conquistei.

Agraço aos meus companheiros de curso que estiveram comigo durante esses anos, tanto aqueles que ingressaram comigo na Universidade, como meus veteranos e meus bixos, amizades que carrego no peito. Agradeço especialmente ao Empanado, por ter sido uma referência para mim durante toda a graduação, ter me ensinado muito e ter sido de grande importância na realização deste trabalho. Agradeço também à todos meus queridos amigos 013 que me ajudaram, direta ou indiretamente, nos momentos difíceis da graduação e deste trabalho. Um abraço para as Repúblicas Alabama e Pão da Vida, locais de enorme felicidade.

Agradeço aos meus amigos de infância que, embora a frequência de nossas conversas tenha diminuído, ainda são extremamente importantes para mim. Agradeço também à todas amizades que fiz durante meu intercâmbio nos Estados Unidos e que hoje se encontram espalhadas por esse enorme Brasil. Agradeço em especial às minhas pequenas famílias 4032, Surfers e Paris 9 durante todo esse período forasteiro, que foram de enorme importância quando a saudade batia e que tanto contribuíram para meu crescimento interior, obrigado por me fazerem descobrir um novo significado de "família".

Agradeço à Naumteria, querida bateria da qual fiz parte durante todos esses anos, que intensificou minha paixão por música, que possibilitou que eu fosse à lugares que jamais imaginei ir, tocar em lugares que jamais imaginei tocar e apresentar nosso samba para multidões, sempre levando alegria por onde passamos. Agradeço à todas amizades verdadeiras que fiz na mesma, mas em especial ao Peru e Picanha, amigos que alegraram meus dias durante anos, me ensinaram muito e tenho enorme gratidão.

Agradeço também à todas amizades que fiz fora do curso, pessoas que agregaram em muito na minha vida e foram essenciais na minha formação como pessoa. Não citarei nomes para não esquecer alguém e ser injusto, mas quem esteve ao meu lado, muito obrigado pela parceria e pelas risadas. Um abraço para a República Tcheca, a original e única.

Por fim, mas não menos importante, agradeço à UNESP Bauru por ter sido o palco de tudo isso, anos intensos e especiais em minha vida. Agradeço à todos os professores que de fato buscam trazer luz aos alunos, em especial ao Marar, Humberto e Andréa, que contribuíram em muito para a minha formação como indivíduo e me ajudaram em momentos de extrema dificuldade, me incentivando a seguir em frente. Agradeço também ao meu orientador, Kleber, por ter instigado meu lado científico e curioso para conseguir produzir este trabalho.

*"A vida em seus métodos diz calma.  
Vai com calma, você vai chegar."  
(A vida em seus métodos diz calma, Di Melo)*

# Resumo

Com a popularização do computador e o acesso à Internet, o número de usuários na rede aumenta diariamente. No entanto, conforme esse número sobe, cresce também o número de possíveis alvos a serem atacados para os invasores, que buscam explorar vulnerabilidades para obter acesso não-autorizado à estas máquinas. Dentro desse panorama, a área de Segurança da Informação tem demonstrado enorme importância para garantir a proteção dos usuários e organizações, além de servir para informar a população sobre os riscos que estão presentes no cotidiano cibernético. Abrangendo outras áreas da computação, o estudo e aplicação desta área é indispensável na época atual, tendo recebido cada vez mais a atenção, que é merecida, da mídia e das organizações, sem contar no aumento de investimentos financeiros por parte das mesmas, após constatarem a necessidade de proteger seus dados. A utilização de ferramentas de segurança para assegurar a segurança interna de um ambiente é vital, e entre as existentes, o honeypot fornece um maior entendimento sobre o invasor, possibilitando estudá-lo, além de também agregar valor à segurança como um todo. Os experimentos realizados comprovam a necessidade da implementação de ferramentas de segurança em um ambiente que deseja manter seus dados confidenciais devido à grande quantidade de ataques executados diariamente, mostrando que é fundamental uma análise sobre o estado da segurança que é empregada em sistemas e redes, além de demonstrar a importância que deve ser dada à esta área da computação.

**Palavras-chave:** Segurança da informação. Malware. Ferramentas de segurança. Honeypot. Open-Source.



# Abstract

Due the low costs attached to hardware production throughout the years, computers have become more popular and therefore their access to the Internet, which has caused the number of users to grow. Because the number of users has grown, the number of possible attack targets has also grown for the invaders, which rely on known vulnerabilities to obtain unauthorized access to their machines. Within this scenario, the Information Security area of study has shown its importance to pursue the user's protection and also show to the population the risks involved in the cybernetic world. Including other areas of study of Computer Science, the study of security is essential nowadays and organizations are finally realizing that it needs attention, which has caused them to start investing heavily in order to protect their confidential data. The use of security tools to assure the internal security of a network or system is vital and, among the existing ones, not only the honeypot provides an in-depth study about the invaders but it also increments the defense in general. The experiments conducted justify the need of the deployment of security tools in order to keep the network safe and invaders-free, it also has shown the a lot of attacks are executed in daily basis and that is necessary to think about how safe a system is, furthermore the importance that this area of study needs to be given.

**Keywords:** Information Security. Malware. Security Tools. Honeypot. Open-Source.

# Lista de ilustrações

Figura 1 – Rede formada através da conexão de computadores (Topologia de rede em anel).	14
Figura 2 – Utilização da Internet em domicílios no ano de 2015. . . . .	15
Figura 3 – Os vinte países nos quais usuários enfrentam o maior risco de serem infectados online. . . . .	17
Figura 4 – Distribuição, por setor, dos ataques registrados em Agosto de 2017. . . . .	19
Figura 5 – Princípios da segurança da informação. . . . .	22
Figura 6 – Etapas da fase de prevenção . . . . .	25
Figura 7 – Prevenção, detecção e resposta . . . . .	27
Figura 8 – Fluxo da informação . . . . .	30
Figura 9 – Vírus e seus meios de propagação . . . . .	32
Figura 10 –Tipos de <i>honeypot</i> . . . . .	41
Figura 11 –Distribuição tradicional de <i>software</i> proprietário . . . . .	44
Figura 12 –Distribuição de <i>software open-source</i> . . . . .	47
Figura 13 –Invasão no laboratório . . . . .	49
Figura 14 –Elementos referentes à escolha do <i>honeypot</i> a ser implementado . . . . .	51
Figura 15 –Consumo computacional da ferramenta Artillery . . . . .	54
Figura 16 –Início do arquivo “config”, encontrado no repositório do Artillery . . . . .	55
Figura 17 –Configuração utilizada no experimento . . . . .	56
Figura 18 –Arquivo gerado pelo Artillery, “ <i>banlist.txt</i> ” . . . . .	57
Figura 19 –Geolocalização de uma amostra dos endereços IPs coletados . . . . .	59

# Lista de tabelas

Tabela 1 – Ataques e suas relações . . . . .	31
--	----

# Lista de abreviaturas e siglas

CERT	Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil
SDI	Sistema de detecção de intrusão
IBGE	Instituto Brasileiro de Geografia e Estatística
SANS	System Administration, Networking and Security

# Sumário

<b>1</b>	<b>Introdução</b>	<b>14</b>
1.1	Modelagem do Problema	17
1.1.1	O problema	17
1.2	Objetivos	18
1.2.1	Objetivo geral	18
1.2.2	Objetivos específicos	18
1.2.3	Justificativa	19
<b>2</b>	<b>Fundamentação Teórica</b>	<b>21</b>
2.1	Conceitos em Segurança	21
2.1.1	Princípios da segurança	21
2.1.2	Prevenção, detecção e resposta	23
2.2	Ameaças e invasões	28
2.2.1	Invasões	28
2.2.2	Invasores	29
2.2.3	Categorização dos Ataques	30
2.3	<i>Malware</i>	31
2.3.1	Vírus	31
2.3.2	<i>Worm</i>	32
2.3.3	<i>Backdoor</i>	33
2.3.4	Cavalo de tróia	33
2.3.5	<i>Spyware</i>	34
2.3.6	<i>Ransomware</i>	34
2.3.7	<i>Bot e botnet</i>	35
2.4	Ferramentas de Segurança	35
2.4.1	<i>Firewall</i>	36
2.4.2	Sistema de Detecção de Intrusão	36
2.4.3	Antivirus	37
2.5	<i>Honeypots</i>	37
2.5.1	Tipos de <i>honeypots</i> e níveis de interação	39
2.5.1.1	<i>Honeypots</i> de baixa interatividade	40
2.5.1.2	<i>Honeypots</i> de alta interatividade	40
2.5.2	Vantagens	41
2.5.3	Desvantagens	42
2.6	<i>Open-source</i>	43

2.6.1	História . . . . .	45
2.6.2	Definição do <i>Open-Source</i> . . . . .	45
2.6.3	Funcionamento . . . . .	46
<b>3</b>	<b>Desenvolvimento . . . . .</b>	<b>48</b>
3.1	Método de Pesquisa . . . . .	48
3.2	Abordagem do problema real . . . . .	50
3.3	Ferramenta Utilizada . . . . .	51
3.3.1	<i>Artillery</i> . . . . .	52
3.4	Experimentos . . . . .	53
3.4.1	Configuração utilizada . . . . .	55
3.5	Resultados . . . . .	56
<b>4</b>	<b>Conclusão . . . . .</b>	<b>60</b>
	<b>Referências . . . . .</b>	<b>62</b>

# 1 Introdução

Há até poucas décadas, eram raros os lugares que possuíam computadores à sua disposição, na realidade apenas algumas universidades prestigiadas ou certas empresas tinham o luxo de possuir um computador, que era responsável por processar, sozinho, o trabalho de diversos usuários. Atualmente, graças à evolução tecnológica e a queda no preço dos componentes e dispositivos eletrônicos, a antiga realidade se tornou obsoleta e existem, em um mesmo local de trabalho ou até em domicílios, diversos computadores separados fisicamente, porém conectados entre si e comunicando-se, formando uma rede de computadores.

Tanenbaum define uma rede de computadores como “um conjunto de computadores autônomos interconectados por uma única tecnologia”, e estes computadores devem ser capazes de trocar informação (TANENBAUM, 2002). Existem diversos tipos de redes de computadores, desde redes simples, como a de um domicílio que conecta alguns computadores, figura 1, até outras mais complexas, como redes de grandes empresas ou de prefeituras, que contam com uma grande quantia de computadores e intensa troca de informação, e essas redes de computadores normalmente estão conectadas entre si e formando redes maiores ainda, como a Internet.

Figura 1: Rede formada através da conexão de computadores (Topologia de rede em anel).



Fonte: Elaborada pelo autor.

A Internet é um exemplo de rede que é, na realidade, uma rede composta por inúmeras outras redes menores, formando uma rede mundial, também chamada de Wide Area Network, que possibilita que usuários troquem informações através de um protocolo comum da rede, independente de onde estejam localizados no mundo. Esta capacidade que a Internet tem de facilitar a troca de informações, em escala global, cria uma gama enorme de aplicações baseadas em computação.

Uma destas aplicações, entre inúmeras possibilidades, é a comercial, uma vez que todos negócios contam com um número substancial de computadores. Em um ambiente onde existe mais de um computador, a troca de informação é essencial e inevitável, pois um usuário pode necessitar de uma informação que se encontra em outro computador, fora de seu alcance, e

então precisará se comunicar com este computador para acessá-la. Em grandes empresas, por exemplo, que contam com funcionários e computadores espalhados em várias filiais ou até países, o fato de que um computador não se encontra à uma distância física tangível de, por exemplo, um banco de dados não pode impedi-lo de acessar sua informação.

Por contribuir de maneira positiva com a troca de informação, a Internet continua em ascensão e aumenta sua presença no cotidiano das pessoas, crescendo seu número de usuários diariamente. Além de usuários comuns fazerem uso da Internet, como pessoas que a utilizam para se informar em sites noticiários ou fazer compras on-line, ela é vital para o funcionamento de muitas organizações, negócios e é essencial para o funcionamento da sociedade moderna, pois, em qualquer ambiente que exista mais de um computador, a troca de informação é indispensável.

Figura 2: Utilização da Internet em domicílios no ano de 2015.

Situação do domicílio e Grandes Regiões	Valores relativos (%)		
	Total	Existência de utilização da Internet no domicílio	
		Havia	Não havia
<b>Brasil</b>	<b>100,0</b>	<b>62,0</b>	<b>38,0</b>
Norte	100,0	47,6	52,4
Nordeste	100,0	48,5	51,5
Sudeste	100,0	71,5	28,5
Sul	100,0	66,5	33,5
Centro-Oeste	100,0	66,5	33,5

Fonte: IBGE, Diretoria de Pesquisas, Coordenação de Trabalho e Rendimento, Pesquisa Nacional por Amostra de Domicílios 2015. (IBGE, 2015)

A figura 2 mostra que, no ano de 2015, 62% da população brasileira utilizava a Internet em seu domicílio, e com tantos usuários utilizando a rede, a quantidade de informação que trafega por ela é praticamente imensurável. Um detalhe importante é que este dado se refere somente à domicílios, e ao Brasil, logo, se levado em conta também as organizações brasileiras e, subsequentemente, o mundo como um todo, a quantidade de informação só tende a aumentar. É importante ressaltar que todo tipo de informação trafega pela rede, das mais banais, como um usuário que faz uma solicitação à um servidor desejando saber a escalação de seu time de futebol, até transações comerciais milionárias.

Evidentemente, como exemplificado acima, existem informações muito valiosas e que devem ser mantidas em total confidência, fora do alcance de qualquer um que não seja autorizado a ter acesso a mesma. Em um mundo perfeito, esse conceito seria uma verdadeiro e somente usuários autorizados conseguiriam acessar tais informações confidenciais. Porém, a realidade é muito diferente, uma vez que existem diversas pessoas má intencionadas e que buscam incessantemente maneiras de obter acesso à informações privilegiadas que não lhe são



de direito, e muitas vezes acabam conseguindo.

Os invasores utilizam muitas técnicas e *softwares* específicos para tentar adquirir acesso à sistemas e suas informações, que serão abordados em maior profundidade na Seção 2.2. Em contrapartida, é feito um extenso trabalho para prevenir invasões e melhorar as ferramentas de detecção das mesmas. Este campo de pesquisa é muito volátil, uma vez que a natureza dos ataques está em constante transformação e isso dificulta o sucesso na área de segurança, visto que as técnicas e *softwares* utilizados para realizar invasões se tornam obsoletos rapidamente. O trabalho é feito em cima de uma metamorfose constante.

Portanto, não é tarefa fácil garantir a segurança de uma rede ou um sistema, pois, além de exigir muito conhecimento técnico sobre a ciência da computação, é necessário que o profissional esteja sempre atualizado quanto às ameaças e ferramentas existentes para sua proteção, sem contar que muitos *softwares* que são utilizados para assegurar a segurança da informação são dispendiosos e consomem muitos recursos do computador, sendo essencial que o mesmo tenha alta capacidade de processamento para conseguir executar os programas, que consequentemente acarreta em mais custos.

Existem inúmeras ferramentas que são utilizadas no âmbito da segurança e cada uma tem um foco específico, diferenciando-se na sua aplicação. Algumas focam na prevenção de invasões, outras focam na detecção de invasões, já outras visam o controle de dano, quando as outras falham. Uma destas ferramentas utilizadas na área da segurança da informação é o *honeypot*, que é utilizado em diversos cenários, pois sua aplicação é extremamente flexível e contribui como um todo para a arquitetura de segurança, e será abordado detalhadamente na Seção 2.4.

Portanto, este trabalho apresenta um estudo sobre ferramentas *open-source* de *honeypot* com o intuito de aprimorar a segurança de redes e sistemas de maneira completamente gratuita, estabelecendo um fundamento sobre as ameaças existentes e buscando apresentar uma solução que fuja do modelo de mercado atual, com uma ferramenta de segurança sem custo monetário e que consome poucos recursos computacionais. O restante do documento está organizado como segue. A Seção 1.1.1 apresenta a problematização da pesquisa. A Seção 1.2.3 apresenta a justificativa em relação à resolução do problema em questão e seu ganho para a comunidade científica. A fundamentação teórica que embasa este projeto é apresentada no Capítulo 2. O desenvolvimento, metodologia empregada, experimentos e resultados obtidos são discutidos no Capítulo 3. E por fim, a conclusão se encontra no Capítulo 4.

## 1.1 Modelagem do Problema

### 1.1.1 O problema

A segurança da informação é uma área da Ciência da Computação que está em constante evolução, devido à frequente mutação das ameaças que a cercam, e, diferente de outros campos de pesquisa, conta com a existência de um adversário permanente: o invasor. Conceitos de aprendizado de máquina, arquitetura de computadores, redes, engenharia de *software*, e até psicologia, são aplicados no desenvolvimento de técnicas e ferramentas que buscam certificar a segurança de sistemas e redes.

No entanto, computadores continuam sendo invadidos diariamente no mundo inteiro e não é incomum o vazamento de informações confidenciais, que inclusive existe um mercado todo dedicado exclusivamente à venda de informações na Deep Web. Com a quantidade de computadores, e dispositivos em geral, existentes, nunca tanta informação foi gerada, armazenada e transferida como atualmente e, movidos por questões financeiras, espionagem ou simplesmente curiosidade, os invasores são muitos e põem em risco a segurança da informação, que é uma área que ainda está progredindo no Brasil, quinto país com o maior risco de um usuário ser infectado online, vide a figura 3.

Figura 3: Os vinte países nos quais usuários enfrentam o maior risco de serem infectados online.

*The TOP 20 countries where users face the greatest risk of online infection*

	Country*	% of unique users**
1	Russia	42.15
2	Kazakhstan	41.22
3	Italy	39.92
4	Ukraine	39.00
5	Brazil	38.83
6	Azerbaijan	38.81
7	Spain	38.21
8	Belarus	38.04
9	Algeria	37.11
10	Vietnam	36.77
11	China	36.53
12	Portugal	35.86
13	France	34.74
14	Armenia	33.01
15	Greece	32.99
16	Chile	32.82
17	India	32.61
18	Qatar	32.53
19	Indonesia	32.30
20	Moldova	31.42

Fonte: Kaspersky Security Bulletin, 2016. ([KASPERSKY, 2016](#))

A garantia de uma defesa sólida se dá através da implementação de ferramentas de

segurança e também de boas práticas e protocolos internos que serão abordados na Seção 2.3. No entanto, tais ferramentas nem sempre estão ao alcance de todos, pois muitas vezes são custosas, além do problema do alto consumo de recursos do computador. Nessa questão, deixando de lado aqueles que não possuem o poder de compra, o fator financeiro é algo muito negativo, visto que muitos usuários, que teriam condições financeiras de adquirir um *software* para incrementar sua proteção, abrem mão de sua segurança por julgar como um investimento desnecessário, que não lhe trará retorno imediato. Já aqueles que não possuem o poder de compra, por exclusão, estão desamparados e mais suscetíveis à falhas em sua segurança devido à falta de ferramentas que auxiliem sua defesa, novamente mostrando que o fator financeiro é algo negativo no quesito proteção.

Este projeto almeja inspirar a busca por conhecimento nessa área tão importante na sociedade moderna, refletir sobre o panorama atual da proteção dos dados em que o país está inserido e reforçar a segurança da informação de maneira ampla e gratuita, através de uma ferramenta *open-source* de *honeypot*, apresentando uma solução que esteja ao alcance de qualquer usuário, sem discriminação financeira.

## 1.2 Objetivos

### 1.2.1 Objetivo geral

Realizar um estudo que viabilize a contenção e prevenção dos vários tipos de ameaças à redes de computadores, visando gerar resultados que salientem a importância da segurança da informação e propor uma arquitetura de segurança eficaz e gratuita, através da implementação de uma ferramenta *open-source* de *honeypot* existente.

### 1.2.2 Objetivos específicos

Problemas com ataques virtuais são cada vez mais frequentes, trazendo insegurança e medo aos usuários, que receiam perder ou ter suas informações roubadas. Por isso é necessário o estudo de métodos eficientes de prevenção e contenção na área de Segurança de Redes de Computadores. Logo, este trabalho visa:

- a) Analisar e selecionar ferramentas *honeypot open-source* já existentes capazes de capturar, classificar e identificar ataques
- b) Classificar estes diferentes *honeypots* de acordo com suas características particulares, como interatividade e foco da aplicação.
- c) Selecionar, instalar e configurar o *honeypot* que melhor atender os requisitos do projeto
- d) Realizar um estudo sobre modelos e padrões de técnicas de ataques e manipulações

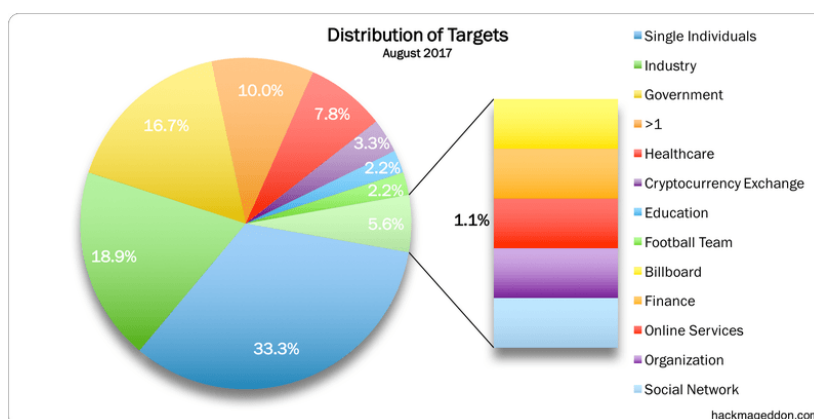
- e) Analisar os artefatos e registros coletados no *honeypot*
- f) Produzir relatório do aprendizado para execução de medidas preventivas e/ou corretivas

### 1.2.3 Justificativa

O avanço tecnológico e a queda no preço dos componentes e dispositivos eletrônicos resultaram na popularização do computador e, conseqüentemente, acesso à Internet. Diariamente, cresce o número de pessoas que estão utilizando a Internet, e como resultado cresce também o número de dispositivos conectados à rede, incluindo diversos dispositivos do cotidiano, como relógios e televisões, conceito conhecido como Internet das Coisas.

Logo, conforme prolifera-se o número de usuários e dispositivos conectados à Internet, é de se imaginar que também aumente a quantidade de informação gerada, armazenada e trocada por eles. E, embora estes números estejam em ascensão, vale ressaltar que existem diversos usuários com pouco conhecimento técnico sobre computação, ou sequer conhecimento algum, e estes não fazem ideia das ameaças existentes, nem tão quanto como proteger-se. Se grandes organizações, que contam com profissionais da área de computação, sofrem com invasões e sentem-se vulneráveis, não seria diferente com os usuários comuns, que lideram o gráfico abaixo, figura 4, como os maiores alvos de ataque em Agosto de 2017, e em seguida, organizações industriais.

Figura 4: Distribuição, por setor, dos ataques registrados em Agosto de 2017.



Fonte: hackmageddon.com ([HACKMAGEDDON, 2017](#))

Foi-se o tempo em que investimento em segurança era sinônimo de gastos somente em câmeras de vídeo, cerca-elétrica ou funcionários fortemente armados. Hoje em dia, a segurança digital é tão vital quanto a segurança física das organizações. No modelo em que está inserido o mercado atualmente, é fato que as empresas que não investem em segurança da informação estão correndo sérios riscos de serem prejudicadas, tendo suas informações como uma presa fácil. Mesmo assim, a área de segurança da informação não é amplamente estudada por muitos

profissionais, além de receber pouca atenção da grande maioria das organizações, que só lhe dará o devido valor após sofrer algum tipo de ataque e perceber sua vulnerabilidade.

A necessidade de ferramentas e técnicas que incrementem a segurança da informação é a principal motivação deste projeto, somado ao desejo de disseminar esta informação e torná-la acessível para qualquer um que queira implementá-la. Do ponto de vista comercial, o estudo de ferramentas que auxiliem na proteção da informação é de extrema importância, pois empresas são alvo de ataques constantes e devem estar sempre com sua defesa armada para garantir sua preservação.

O foco deste trabalho se dá no estudo de ferramentas *honeypot*, pois se mostraram, ao longo de anos de pesquisa e implementação, extremamente flexíveis em suas aplicações e de baixo consumo de recursos computacionais, abordado com maior detalhe na Seção 2.4, possibilitando que seja implementado em praticamente qualquer ambiente imaginável, desde um domicílio com computadores de pouco poder de processamento, até laboratórios de pesquisa de organizações renomadas com computadores poderosos. Além disso, é essencial para este trabalho que o conteúdo estudado seja *open-source*, de maneira que garanta a disseminação, gratuita, da informação e sua implementação. Sendo assim, um fator importante para a aplicação neste problema real de segurança, pois é do desejo do autor trazer visibilidade à falta de segurança e à quantidade de ataques que são realizados, mostrando que a proteção da informação é algo que deve ser valorizado.

## 2 Fundamentação Teórica

Nesta seção, é retratado todo o embasamento teórico que serve de sustentação para o desenvolvimento deste projeto. Como apontado anteriormente, o principal foco do projeto é o acréscimo da segurança da informação através da utilização de uma ferramenta *honeypot open-source*, ou seja, gratuita.

Assim sendo, é apresentada uma introdução sobre o conceito de segurança da informação e a importância da mesma no modelo comercial atual. Após apresentado este conceito, que é responsável pela motivação do projeto, o estudo será estendido, abordando o que está envolvido na invasão propriamente dita, como as técnicas e também as características do invasor, e além disso, serão abordadas as técnicas e ferramentas de segurança que são utilizadas para conter tais invasões. Então, serão introduzidos os fundamentos sobre *honeypots* e suas numerosas características, de maneira que eles possam ser validados no contexto proposto pelo projeto. Também será apresentado o conceito de *open-source*, o *software* livre, essencial para a desdobramento deste projeto.

### 2.1 Conceitos em Segurança

A segurança da informação é uma área que está sendo cada vez mais valorizada no contexto do mundo comercial. Não que algum dia ela não tenha sido importante, porém, conforme democratizou-se o acesso à computadores e à Internet, as ameaças começaram a aparecer e ampliar-se, resultando na necessidade do desenvolvimento de técnicas capazes de garantir proteção aos que necessitam.

A ciência da computação possibilita o desenvolvimento de inúmeras aplicações, e organizações que trabalham neste ramo normalmente possuem, além de seus segredos empresariais, informações confidenciais sobre seus clientes, que devem ser tão bem protegidas quanto às suas próprias informações sigilosas. Por isso, é possível notar um aumento gradual no número de organizações que estão investindo em segurança, uma vez que se torna inevitável esta preocupação com seus dados.

#### 2.1.1 Princípios da segurança

Todos negócios são sustentados por elementos denominados ativos, que são, basicamente, qualquer coisa que tenha valor para a organização, sendo responsáveis pelo suporte de toda a operação do negócio. Eles se apresentam nas mais diversas maneiras, simplificadaamente, podendo ser subdivididos em:

- a) Físicos: são aqueles que são tangíveis, como a estação de trabalho, *hardware*, funcionários, móveis e informações impressas.
- b) Lógicos: são aqueles que são intangíveis, como a marca de um produto, nome da empresa e informações armazenadas eletronicamente.

Ambos são de igual importância para o funcionamento do negócio e devem ser proporcionalmente protegidos, no entanto vale ressaltar que cada um requer uma atenção especial, pois a solução que garante a defesa física dos ativos, como alarmes e cerca-elétrica, não será a mesma solução que irá garantir a defesa lógica dos mesmos. Sendo assim, fica evidente que a informação é também um ativo, e como qualquer outro, é essencial para o desempenho do negócio e deve ser preservada, seja ela impressa em um papel ou armazenada eletronicamente em um computador (IMA-SP, 2005).

Visando os ativos, porém, especialmente, a informação, é apresentado os três objetivos principais, que são o coração, da segurança da informação, conforme exibido na figura 5.

- a) Confidencialidade: Garantia de que informações privadas ou confidenciais serão acessadas somente por pessoas devidamente autorizadas. Uma falha na confidencialidade significa divulgação não-autorizada da informação.
- b) Integridade: Garantia que alterações, supressões e adições nas informações serão realizadas somente por aqueles que são autorizados, incluindo autenticidade e não-repúdio. Uma falha na integridade significa modificação ou destruição não-autorizada da informação.
- c) Disponibilidade: Garantia de que sistemas, programas e informação estarão prontamente disponíveis para acesso daqueles que são autorizados, quando demandado, e também que estes serviços não serão negados ao mesmos. Uma falha na disponibilidade significa interrupção no acesso ou na utilização da informação ou de um sistema.

Figura 5: Princípios da segurança da informação.



Fonte: Elaborado pelo autor.

Embora estes sejam os princípios da segurança, dificilmente um sistema será completamente disponível, íntegro e confidencial, pois, apesar dos princípios aparentarem ser simples,

os mecanismos empregados para garanti-los são complexos e envolvem muito mais do que um algoritmo ou protocolo, particularmente, em sua implementação. No entanto, mesmo assim, a segurança da informação busca garantir esses princípios através da proteção da confidencialidade, garantia da integridade e manutenção da disponibilidade, e a melhor maneira da proteção ter êxito é com um bom planejamento sólido, pois assim serão reduzidos os riscos de um ataque dar certo e aumentará as capacidades de detectar e responder um ataque a tempo.

Essencialmente, segurança se resume à uma batalha entre um invasor, que procura “buracos”, ou no caso, vulnerabilidades, e um administrador que tenta fechá-los, impedindo acessos não-autorizados. No entanto, basta que o invasor consiga achar uma única vulnerabilidade para ter êxito, enquanto o administrador necessita eliminar todas vulnerabilidades para ter um sistema perfeitamente seguro.

Por isso, tendo em vista que os ativos são os responsáveis pela sustentação de toda a operação de um negócio, é possível afirmar que eles também são responsáveis pelas vulnerabilidades que podem prejudicar a operação. A fragilidade de um único ativo, ou um grupo deles, pode ser explorada por ameaças, que são as causas de incidentes inesperados, ou indesejados, resultando em possíveis danos para algum sistema ou para a organização.

Portanto, é inevitável que haja vulnerabilidades em uma organização, por isso é feito um extenso trabalho para diminuí-las ao máximo, ou pelo menos ocultá-las, e este trabalho é feito, essencialmente, através da implementação de outros princípios importantes da segurança de computadores, abordados a seguir.

### 2.1.2 Prevenção, detecção e resposta

Assim como a integridade, confidencialidade e disponibilidade são considerados o coração da segurança, e devem ser perseguidos ao máximo na implementação de qualquer arquitetura de segurança, existem outros princípios que compõem e complementam a idéia de um sistema seguro. Estes princípios trabalham de maneira a concluir o grande objetivo da segurança da informação, que é garantir os três princípios-corção da segurança, abordados na seção anterior.

Uma maneira de analisar a idéia de segurança da informação é observando-a como um processo que cresce progressivamente e se desenvolve ao longo de seu caminho, como uma bola de neve. Diversas estratégias e táticas diferentes podem ser aplicadas na implementação de um projeto de segurança da informação, e, evidentemente, todas visam proteger o ambiente em questão, logo é possível afirmar que todo este processo da implementação de um projeto de segurança da informação revela a importância da jornada propriamente dita e das decisões tomadas, não o destino em si, pois ele sempre será o mesmo. Porém, o que todos estes processos têm em comum é a aplicação de três conceitos: prevenção, detecção e resposta (SANS, 2002).



Estes três conceitos, também chamados de fases, requerem estratégias e execuções de tarefas específicas para que o projeto consiga prosseguir adiante, alcançando a próxima fase. Como mencionado anteriormente, o lado do invasor é sempre uma incerteza para quem trabalha com a defesa, pois ele está em constante mudança, o que faz com que seja exigido revisões periódicas sobre as metodologias empregadas no ciclo de prevenção, detecção e resposta. É importante ressaltar que as mudanças feitas, por exemplo, no âmbito de prevenção, irão acarretar mudanças também na detecção e na resposta, e vice-versa. Tudo faz parte de um grande ciclo, que está em constante adaptação para minimizar as vulnerabilidades do ambiente, e busca um destino final, sua proteção.

Quando o assunto é segurança da informação, é sempre mais eficaz prevenir do que remediar, mesmo que isso requeira uma análise extensa e muito planejamento. Medidas têm de ser tomadas para proteger a informação, que é um ativo que necessita de proteção proporcional ao seu valor, de qualquer modificação, destruição ou divulgação não-autorizada, seja ela intencional ou não. Portanto, na fase de prevenção, políticas de segurança, programas de conscientização e controles de acesso devem ser desenvolvidos e implementados o quanto antes possível.

Para começar a definir um planejamento de prevenção, é importante estabelecer o que deve ser protegido e documentar todos estes itens, de maneira formal. Isso é o começo de uma política de segurança, que também deve incluir as responsabilidades da organização, dos funcionários e da gerência, além de indicar os responsáveis pela implementação, execução e revisão da mesma. Essa política deve ser concisa, clara e coerente, para que possa ser bem compreendida e transmitir o menor número de dúvidas possíveis, evitando que ela seja implementada erroneamente e consequentemente tenha sua execução e revisão em vão.

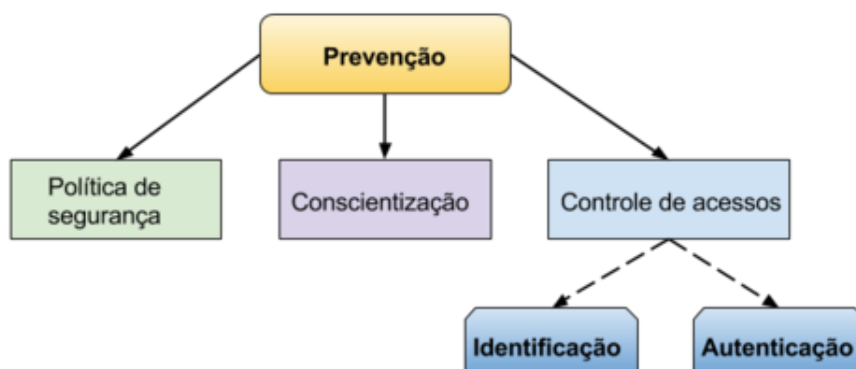
Outra medida essencial da fase de prevenção é a criação de um programa de conscientização sobre a segurança. Este tipo de programa retrata um processo, no qual ele educa os funcionários sobre a importância da segurança, a utilização das medidas de segurança e a responsabilidade dos mesmos perante a proteção da organização. É necessário que a implementação deste processo de conscientização seja contínua, para que os funcionários nunca se esqueçam de suas responsabilidades e mantenham sempre um bom nível de consciência sobre suas ações.

Não menos importante, ainda na fase de prevenção, é a implementação de um mecanismo para controle de acessos, pois, logicamente, não são todos os usuários que podem ter a capacidade de acessar qualquer sistema e as informações nele presentes. Levando em conta que acesso pode ser definido como a maneira com que usuários usam os sistemas para adquirir informações, este acesso deve ser restringido e concedido conforme a necessidade daquele usuário precisar de uma informação em questão. Para garantir e gerenciar este acesso, são utilizados mecanismos que criam contas para os usuários, contendo identificadores e métodos de autenticação, que são responsáveis pela limitação do acesso aos recursos.

- a) **Identificação:** cada usuário possui um identificador único, que é utilizado para mostrar quem é este usuário e diferenciá-lo de outros objetos. Vale ressaltar que os identificadores que são criados são únicos e não devem, de maneira alguma, ser compartilhados com outros usuários ou grupos, garantindo assim a segurança. Depois de um usuário ser identificado, o próximo passo é que ele seja autenticado, para então conseguir acesso ao recurso demandado.
- b) **Autenticação:** é um processo no qual é validada a identidade do usuário, que apresenta seu identificador e aguarda sua autenticação para que então consiga ganhar acesso. Este processo é importante pois ele verifica a identidade dos usuários que estão tentando adquirir acesso, fornecendo um maior nível de confiança e segurança ao ambiente. Existem alguns fatores de autenticação básicos que são utilizados para verificar a identidade de um usuário, como:
- Algo que o usuário é: este é o fator mais forte, pois trata de características físicas únicas, geralmente utilizando meios biométricos, como impressão digital, padrão de retina, padrão de voz, DNA, reconhecimento facial, reconhecimento de assinatura, entre outros.
  - Algo que o usuário tem: utiliza-se objetos específicos que o usuário possui, como cartões de identificação, smart cards ou tokens.
  - Algo que o usuário conhece: utiliza-se palavras-chave para identificar o usuário, como senha fixa, frases secretas ou número PIN.
  - Onde o usuário está: é limitado o acesso ao sistema à máquinas específicas, cujo acesso é restrito somente à elas.

Um processo de autenticação forte normalmente irá requerer mais de um destes fatores, ou até todos, para validar um usuário. Porém, isso varia de organização para organização e também do nível de privacidade das informações, podendo variar a quantidade de fatores de autenticação conforme as necessidades de segurança exigidas.

Figura 6: Etapas da fase de prevenção



Fonte: Elaborado pelo autor.

Após adotar uma política de segurança, criar um programa de conscientização, estipular controle de acessos e implementar estes itens abordados, uma prevenção sólida é garantida, figura 6, mas ainda é necessário implementar estratégias de detecção e resposta, pois é melhor que uma organização esteja pronta para lidar com um ataque do que subestimar uma ameaça e ser pega de surpresa, indefesa. É importante que estas estratégias estejam bem definidas, pois para ter êxito é necessário que a organização saiba o que detectar e, uma vez detectado, saiba utilizar seus recursos para responder ao ataque.

Por maior que seja o esforço investido na prevenção de ataques, não há nenhuma garantia que o ambiente será completamente seguro. Muito pelo contrário, pois devido ao crescente aumento das ameaças, não importa quão sólida a defesa seja, é provável que ela eventualmente irá ser quebrada, principalmente pelo fato de que os possíveis invasores são muitos, e dependem de sua motivação e habilidades técnicas. Por isso que a detecção de invasões é extremamente importante, e melhor ainda se a defesa for implementada em camadas, já que quando uma camada falhar, é conhecido onde ocorreu o problema, facilitando sua detecção. Embora não existe nenhuma receita de construção para uma segurança perfeita, independente da estratégia adotada, é essencial que a detecção ocorra rapidamente e uma notificação sobre a falha seja divulgada, possibilitando uma resposta eficaz.

Uma das ferramentas mais utilizadas na implementação da fase de detecção são os Sistemas de Detecção de Intrusão (do inglês, *Intrusion Detection Systems*), que são sistemas de segurança capazes de monitorar sistemas computacionais e tráfego de rede, analisando este tráfego para possíveis ataques hostis originados fora da organização e também para ataques originados de dentro da mesma. Além disso, eles detectam alterações em arquivos, configurações e atividades. Porém, monitorar uma rede muito ativa não é uma tarefa fácil, de maneira que este sistema deve ser capaz de processar a informação da rede e distinguir uma atividade normal do sistema de uma atividade maliciosa, e isso requer muitos recursos computacionais, exigindo uma máquina potente, e talvez até dedicada, para executar tal feito.

Assim que uma estratégia de detecção for definida, é necessário focar na estratégia para resposta das detecções, e documentá-la, dado que sem um plano sólido e estruturado sobre como agir em relação às futuras detecções, será um caos total e de nada terá adiantado as fases de prevenção e detecção quando um incidente ocorrer. Como citado anteriormente, prevenção, detecção e resposta caminham juntas e estão inter-relacionadas, de maneira que sem uma estratégia exata para cada uma dessas fases, as outras se tornam sem sentido e em vão.

Portanto, para que a fase de detecção tenha algum valor, é necessário que a resposta seja feita a tempo quando ocorrer a notificação de um incidente. E isso é obtido através um planejamento feito com antecedência à detecção, pois tomar decisões ou desenvolver uma estratégia de resposta sob ataque provavelmente irá resultar em falha. Este plano de resposta deve ser escrito, documentado e validado pela gerência, assim como no caso da política de

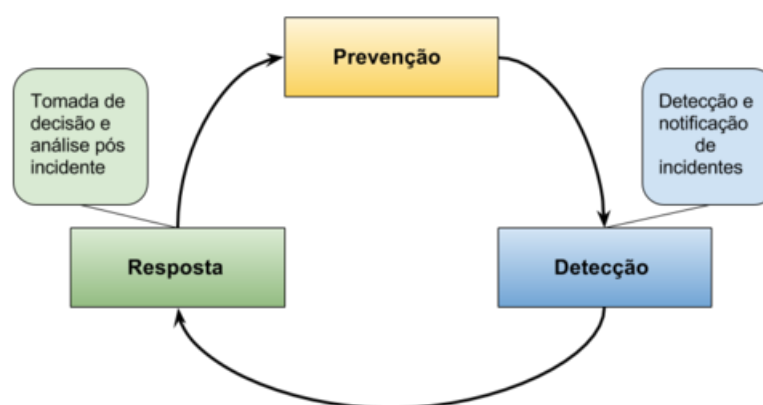
segurança.

Basicamente, existem duas filosofias quando o assunto é resposta à um ataque e quais ações devem ser tomadas. Uma delas é a organização cortar a conexão do invasor, eliminar a causa do incidente e recuperar seus sistemas afetados, essa estratégia é viável quando máquinas críticas são afetadas e é necessário uma recuperação rápida das mesmas, sendo essa a prioridade do momento. A outra abordagem se baseia em perseguir e processar o invasor, que é uma abordagem mais arriscada e exige exposição maior da organização, sendo aconselhável somente quando é de muito interesse da organização tomar medidas para investigar e identificar um invasor que comprometeu seus sistemas.

Independente da abordagem escolhida, é necessário que a metodologia empregada por ambas esteja propriamente documentada no plano de resposta. Após selecionada a abordagem a ser seguida, funcionários devem ser atribuídos a tarefas adequadas à suas competências técnicas, coletando evidências, recuperando sistemas afetados, e sempre documentando todos os processos realizados.

Então, logo após um incidente ser detectado e notificado aos funcionários responsáveis pela resposta, ele deve ser contido, o dano deve ser avaliado e o sistema deve ser cuidado e recuperado. Cada um destes processos requer funcionários com competências técnicas específicas para realizar as tarefas designadas, e cada um têm papel importante na fase de resposta. Com isso, é terminada a fase de resposta, porém o ciclo não acaba aí, pois a análise após o incidente ser contido é extremamente importante, uma vez que é com ela que se enriquece a segurança como um todo, aprendendo com as falhas e vulnerabilidades para poder consertá-las e evitar futuras invasões.

Figura 7: Prevenção, detecção e resposta



Fonte: Elaborado pelo autor.

Para vencer um invasor, seja ele interno ou externo, é necessário que a organização esteja devidamente preparada, visto que, como dito anteriormente, o processo de segurança da informação é um ciclo dinâmico entre prevenção, detecção e resposta, como representado

pela figura 7, e requer melhorias constantes na estratégia empregada em cada uma das fases citadas, já que, por exemplo, uma análise após um incidente ser contido pode revelar uma falha na fase de prevenção, que então deve ser corrigida, e esta correção pode acarretar na necessidade de alguma mudança nos mecanismos de detecção, e assim por diante.

## 2.2 Ameaças e invasões

Abordado os princípios da segurança da informação, é necessário observar como funciona o outro lado dessa batalha, e entender quem é o invasor, quais são suas características, motivações e técnicas empregadas para aproveitar as vulnerabilidades e quebrar as defesas do ambiente.

### 2.2.1 Invasões

Muitas pessoas têm a impressão errônea de que nunca serão alvos de ataques maliciosos, pois acreditam que seu sistema não possui nenhum valor ao invasor, logo ninguém irá querer atacá-lo. Essa é uma afirmação extremamente errada, pois todo sistema pode trazer algum benefício ao invasor, já que, mesmo se ele não possuir de fato nenhuma informação valiosa, o computador ainda pode ser utilizado para atacar outros sistemas ou até mesmo ser utilizado para armazenar toda a informação que o invasor roubou de outras vítimas. Tendo isso em vista, o grande problema enfrentado pelos sistemas é a invasão de usuários não-autorizados ou a utilização de *softwares* para realizar invasões, e essas invasões podem provir tanto de um ambiente externo quanto de um interno, exigindo políticas de defesa que incorporem ambos cenários.

Normalmente, invasões estão relacionadas com a segurança da rede, pois é através da rede que se conquista acesso ao sistema, sendo assim, é considerada como segura uma rede que conta com boas ferramentas preventivas, de detecção de invasão e de resposta. No entanto, mesmo assim, nem todos os ataques estão completamente ligados a rede, pois como dito acima, existem invasões que provêm de um ambiente interno, como quando um usuário legítimo, com acesso ao terminal local, tenta acessar recursos cujo seu acesso não é autorizado. Neste caso, sua conexão não utiliza nenhuma rede intermediária e acaba burlando diversas medidas preventivas empregadas visando a segurança da rede, dificultando a prevenção deste tipo de invasão.

Vale ressaltar que nem toda invasão é necessariamente realizada por um usuário, pois um *software* pode tentar violar a segurança da rede na forma de um vírus, *worm* ou cavalo-de-tróia (*trojan horse*), que será abordado detalhadamente na Seção 2.3.4.

### 2.2.2 Invasores

Embora nem todas invasões sejam realizadas por invasores, eles ainda são uma grande ameaça à segurança das redes e merecem uma atenção especial.

Anderson ([ANDERSON, 1980](#)) identificou três classes principais de invasores:

- a) Mascarado: indivíduo que não tem autorização para utilizar o computador e penetra o controle de acesso do sistema para abusar da conta de um usuário legítimo. Normalmente este tipo de invasor provém de um ambiente externo.
- b) Malfeitor: usuário legítimo que acessa dados ou recursos cujo seu acesso não é autorizado ou utiliza de maneira errônea seus privilégios. Normalmente este tipo de invasor provém de um ambiente interno.
- c) Usuário clandestino: indivíduo que aproveita controle supervisorio do sistema e utiliza este controle para não deixar rastros e acessar controles ou simplesmente para suprimir a coleta de seus rastros. Este tipo de invasor pode provir tanto de um ambiente externo quanto interno.

Já Lance Spitzner, em seu livro *Tracking Hackers* ([SPITZNER, 2002](#)), afirma que, em geral, existem dois tipos de hackers: aqueles que têm como objetivo comprometer um sistema em específico ou sistemas de alto valor e aqueles que têm como objetivo comprometer o maior número possível de sistemas. Segundo ele, este primeiro tipo de hacker normalmente é alguém já altamente experiente e motivado financeiramente, com um alvo específico de ataque, geralmente de alto valor. E embora sejam a minoria, esses hackers são altamente perigosos devido ao seu vasto conhecimento e competência técnica, capacitando-os penetrar sistemas seguros e dificultar o rastreamento de suas ações. Já o segundo tipo de hacker, ao contrário do primeiro, tem como objetivo atacar o maior número possível de sistemas, com o menor esforço possível. A maioria desses ataques são realizados através de *scripts* e *softwares*, e são menos sofisticados, embora numerosos. Esse tipo de hacker se importa com números e representa a maior parte dos ataques e golpes vistos diariamente.

Além disso, os ataques dos invasores também podem ser divididos entre benignos e sérios. No lado dos benignos, estão os invasores “curiosos”, que apenas estão explorando a Internet e analisando as consequências de suas ações e, embora tolerável, estes ataques ainda consomem recursos computacionais e podem prejudicar a performance do sistema para usuários legítimos. Já os ataques sérios são todos aqueles que se resumem a leitura de dados, modificação de dados e interrupção do sistema, feitos por usuários não-autorizados e que apresentam grande risco para a organização.

Portanto, tendo conhecimento sobre as características e motivações dos invasores, é necessário também entender como funcionam os ataques e porque eles são prejudiciais às organizações.

### 2.2.3 Categorização dos Ataques

Dentro de um sistema funcional, os computadores são responsáveis pelo fornecimento de informação e, embora nem todos ataques sejam iguais, é este o alvo principal deles. Existe um fluxo natural que a informação percorre desde sua fonte até seu destinatário, e é neste caminho, entre a fonte da informação e seu destinatário, que a maioria dos ataques acontecem.

Figura 8: Fluxo da informação



Fonte: Elaborado pelo autor.

Os ataques podem ser categorizados nas seguintes quatro formas, segundo Stallings (STALLINGS, 2010):

- a) Interrupção: quando um ativo do sistema é destruído ou se torna indisponível ou inutilizável. É um ataque na disponibilidade.
- b) Interceptação: quando alguém não-autorizado ganha acesso a um ativo. É um ataque na confidencialidade.
- c) Modificações: quando alguém não-autorizado não somente ganha acesso, mas também altera um ativo. É um ataque na integridade.
- d) Fabricação: quando alguém não-autorizado insere objetos falsificados no sistema. É um ataque na autenticidade.

Outra categorização dos ataques adotada é em termos de ataques passivos e ataques ativos, tabela 1. Ataques passivos são da natureza de monitoração ou transmissão, sendo o objetivo do invasor obter a informação que está sendo transmitida. Embora possíveis de serem evitados, estes ataques dificilmente são detectados, pois não envolvem a alteração da informação, habitualmente envolvendo apenas a leitura da informação. Exemplos de ataques passivos são a divulgação do conteúdo e a análise do tráfego da rede. Já ataques ativos são aqueles que envolvem alguma modificação da informação ou a criação de informação falsa. Este tipo de ataque é o exato oposto dos ataques passivos, uma vez que os passivos são de difícil detecção e evitáveis, enquanto estes são mais complexos de serem evitados e relativamente mais fáceis de serem detectados. Podem ser considerados ataques ativos, por exemplo, a modificação de uma mensagem e negação de serviço (do inglês, *Denial of Service*).

Abordado o foco dos ataques, é preciso compreender então as técnicas que são empregadas de fato para que eles aconteçam, e então não somente interpretar como eles funcionam, mas também possibilitar o aprendizado de como se defender dos mecanismos utilizados pelos invasores.

Tabela 1: Ataques e suas relações

Ataque	Deteccção	Prevenção
Passivo	difícil	possível
Ativo	fácil	possível

Fonte: Elaborado pelo autor.

## 2.3 *Malware*

Como citado anteriormente, embora muitos sejam os motivos que levam o invasor a realizar um ataque, eles têm em comum a utilização de códigos maliciosos (do inglês, *malware*) para realizá-los. Tais códigos são desenvolvidos por pessoas competentes e de vasto conhecimento computacional e então são propagados em massa, ou não, dependendo da finalidade do mesmo, até chegarem em seu destino final: as máquinas a serem infectadas (CERT, 2017).

Em maior detalhe, estes códigos são programas desenvolvidos com o intuito de executar ações danosas e atividades maliciosas em uma máquina, coisa que um programa ou usuário legítimo não fariam intencionalmente. Eles podem infectar a máquina através da exploração de uma vulnerabilidade existente no sistema operacional ou nos programas instalados, acessando páginas Web maliciosas por meio de navegadores vulneráveis, executando arquivos previamente infectados que foram obtidos por anexo de e-mail, mídias removíveis ou diretamente de outros computadores, ou até pela ação direta dos invasores que conseguiram burlar as defesas do ambiente e colocam arquivos que contém tais códigos na máquina.

Os códigos maliciosos podem ser divididos em diversas categorias, mas as principais serão abordados a seguir.

### 2.3.1 Vírus

Um vírus de computador é como um vírus de gripe, que procura se propagar de hospedeiro para hospedeiro e tem a habilidade de se replicar. No entanto, no âmbito da computação, os vírus possuem características específicas que são muito interessantes (SYMANTEC, 2017b).

Como citado acima, um vírus é um código malicioso e é criado com o objetivo de alterar a maneira natural que uma máquina opera, além de buscar se replicar e espalhar-se entre o maior número de máquinas possíveis, e ele faz isso inserindo cópias de si mesmo e se tornando parte de outros programas e arquivos presentes na máquina.

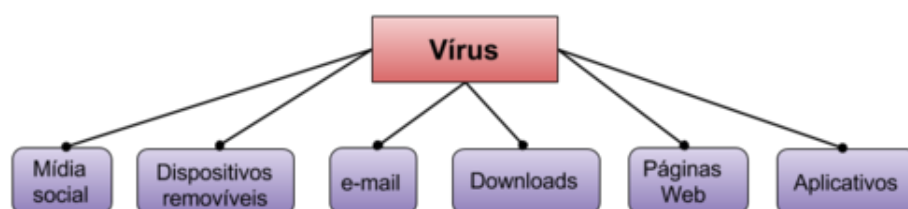
Um dos grandes problemas para detectar infecções de vírus em máquinas se dá pelo fato de que o vírus, considerando que ele já esteja presente na máquina, permanece inativo na mesma até que algum evento aconteça que faça com o que o computador, algum dispositivo, ou o usuário execute o programa ou arquivo em que ele está presente, e consequentemente



execute seu código malicioso junto. Isso significa que um vírus pode ficar no computador por muito tempo até que ele realmente seja executado, e durante todo esse tempo ocioso ele não irá demonstrar nenhum sintoma de infecção, dificultando sua detecção. No entanto, uma vez que o vírus é ativado, ele pode infectar outras máquinas da rede e se propagar rapidamente (NACHENBERG, 1997).

Cada vírus é diferente e sua finalidade depende do código que foi programado, mas as ações mais comuns são roubar senhas e informações, registrar teclas digitadas, corromper arquivos, propagar-se através de e-mails ou mídias sociais, ou apenas “brincar” com a máquina em questão, mudando as configurações de interface padrão do usuário. Vale ressaltar que vírus podem também acarretar em resultados catastróficos em certas ocasiões, como deletando arquivos, causando danos permanentes no disco rígido ou até em casos muito específicos que visam lucro financeiro.

Figura 9: Vírus e seus meios de propagação



Fonte: Elaborado pelo autor.

Como apresentado na figura 9, eles se propagam de diversas maneiras, desde o uso de *pen-drives*, que contêm arquivos infectados; e-mails com arquivos infectados de anexo, que induzem o usuário a clicar sobre este arquivo e executá-lo; acessando páginas *Web* que contêm o vírus escrito em forma de *script*, através de linguagens de programação como o *JavaScript*, e podem ser executados automaticamente, dependendo da configuração do navegador *Web*; golpes em mídias sociais, que induzem o usuário a clicar em links infectados; ou até através do *download* de aplicativos suspeitos de *smartphone*. É bom salientar também que, com a mídia social em alta atualmente, é muito comum vírus que estão disfarçados em conteúdos que são compartilháveis, como imagens engraçadas, vídeos apelativos, áudios, documentos, entre outros.

### 2.3.2 Worm

Um *worm* é um tipo de código malicioso com altíssima capacidade de replicação e propagação automática pelas redes, de maneira que ele envia cópias de si mesmo de computador para computador (KASPERSKY, 2017c).

Sua definição se assemelha a de um vírus, no entanto há uma grande diferença entre os dois, pois os vírus estão associados à algum programa ou arquivo executável, e mesmo depois que este arquivo infectado de algum modo chega até a máquina em questão, ele permanece dormente até que seja ativado, e após sua ativação que começará de fato a se replicar no sistema. Já os *worms* não necessitam de um programa para serem ativados, replicados e propagados, uma vez que eles são auto-suficientes.

Portanto, assim que o *worm* se encontra na máquina, o que normalmente acontece através de uma conexão de rede ou *download* de arquivo, ele irá fazer múltiplas cópias de si mesmo e começará a se propagar pela rede, em busca de outras máquinas vulneráveis para serem infectadas. Devido ao fato de que não há a necessidade de um intermédio para que o *worm* se auto-replique, diferente do vírus, as infecções de *worm* podem se espalhar com extrema velocidade através de redes locais ou da Internet. Vale ressaltar também que, justamente por causa da grande quantidade de cópias de si mesmo que costumam propagar, os *worms* consomem muitos recursos e podem afetar bruscamente o desempenho das redes e dos sistemas infectados.

### 2.3.3 *Backdoor*

Um *backdoor* é, basicamente, uma maneira de acessar um sistema ignorando os mecanismos de segurança habituais do mesmo. Tendo isso em mente, vale ressaltar então que um *backdoor* não é necessariamente um *malware*, uma vez que esta é uma técnica que pode ser utilizada de forma legítima por, por exemplo, programadores durante o processo de desenvolvimento de um *software*. O problema está quando estes mesmos programadores não removem este *backdoor* após o desenvolvimento, o *software* chega ao público e pessoas com más intenções descobrem essa vulnerabilidade ([TECHTARGET, 2017b](#)).

Em outros casos, *backdoors* são posicionados através de *malware*, permitindo retorno de um invasor ao computador comprometido. Normalmente, isso acontece pela ação de invasores ou códigos maliciosos que, após invadir o computador através de vulnerabilidades existentes no mesmo, incluem o *backdoor* para garantir acesso futuro ao computador, permitindo que seja possível o acesso remoto à ele, sem que seja notado e que dificilmente será detectado.

### 2.3.4 Cavalo de tróia

Um cavalo-de-tróia é um programa que executa suas funcionalidades normais, aquelas que ele realmente foi projetado para fazer, e, além disso, também executa funções maliciosas e sem o conhecimento, nem consentimento, do usuário. Vale ressaltar que, diferentemente dos vírus, esses programas não se replicam, como os vírus fazem ([SYMANTEC, 2017a](#)).

É muito comum a obtenção de cavalos-de-tróia em sites que oferecem programas gratuitos, que normalmente são pagos, como jogos, sistemas operacionais e ferramentas de

trabalho. Geralmente, estes programas consistem de um único arquivo e necessitam que sejam executados para serem instalados no computador.

Sua aplicação é vasta, pois ele pode executar basicamente qualquer coisa que foi programado para fazer, e o usuário, sem ajuda externa, provavelmente nunca irá descobrir sua existência no computador. Algumas de suas aplicações são: instalar outros códigos maliciosos; incluir *backdoors* no sistema; instalar ferramentas de negação de serviço e utilizá-las para realizar ataques; instalar programas *spyware*; entre outros.

### 2.3.5 *Spyware*

*Spyware* são programas desenvolvidos com o intuito de coletar dados de um computador, ou outro dispositivo, e enviá-los para terceiros, sem o conhecimento e consentimento do usuário. Em suma, ele repassa informações pessoais, e confidenciais, do usuário para o invasor (KASPERSKY, 2017b).

Este tipo de *malware* normalmente chega ao computador juntando-se com algum outro programa que o usuário quis de fato adquirir e instalar, assim como os cavalo-de-tróia, que inclusive são um exemplo de como um *spyware* pode chegar até o computador, além de todos os outros exemplos já citados sobre como os *malwares* apresentam-se. Todo esse processo pode acontecer discretamente e o usuário sequer saberá do acontecido, porém existem casos no qual o *software* em questão anuncia a existência do *spyware* no código, no entanto o faz com outra nomenclatura e termos técnicos, embutidos nos termos de licença do *software*. Tendo isso em mente, a melhor maneira de se proteger de um *spyware* é prevenindo que ele chegue até sua máquina, pois sua detecção é difícil.

Normalmente, um *spyware* pode ter acesso a praticamente qualquer informação do computador, como histórico de navegação, senhas armazenadas, contas de e-mail, entre outras. Com isso, se o usuário visita sites de banco *online*, o *spyware* pode capturar suas informações e vendê-las para terceiros ou o próprio invasor pode utilizá-las como bem desejar. Além disso, um dos grandes problemas acarretados por um *spyware* é o dano que ele traz ao computador e à rede, uma vez que ele consome muitos recursos e irá reduzir o desempenho, podendo causar falhas nos sistemas ou até superaquecimento do computador, causando danos permanentes.

### 2.3.6 *Ransomware*

*Ransomware* é um tipo de programa que encripta os arquivos e bloqueia a tela do computador, restringindo o acesso do usuário ao mesmo. Então, após o computador ser bloqueado, é cobrado uma quantia financeira da vítima para recuperar os dados e o acesso ao computador, normalmente em moedas virtuais, como o *Bitcoin* ou *Ethereum*.

Esse tipo de *malware* utiliza algoritmos conhecidos, ou customizados, para encriptar os arquivos e podem ir além, propagando-se para outras máquinas através da rede, aproveitando

de vulnerabilidades na mesma ([MICROSOFT, 2017](#)).

Assim como os outros *malwares*, sua infecção pode ter início através do *download* de arquivos infectados, que podem ser considerados como cavalo-de-troia, ou através de *websites* que hospedam o *malware*, que por sua vez tenta explorar vulnerabilidades do navegador utilizado, ou outros *softwares*, para instalar o *ransomware*.

### 2.3.7 Bot e botnet

Basicamente, *bot* é um tipo de *malware* que permite que o invasor tenha acesso remoto ao computador infectado, utilizando mecanismos de comunicação, como servidores *Web*, e possibilitando que o invasor envie instruções para que ações maliciosas sejam executadas. Além disso, um *bot* possui processo de infecção e propagação semelhantes ao do *worm*, podendo propagar-se automaticamente ao explorar vulnerabilidades existentes ([SYMANTEC, 2016](#)).

Os computadores infectados são referidos como “zumbis”, pois podem ser controlados remotamente e executar tudo que seu “mestre” instruir, sem o conhecimento do usuário. Logo, uma *botnet* é uma rede formada por estes computadores zumbis, podendo ser pequena ou gigantesca, contando com centenas de milhares de computadores, o que potencializa seu poder.

As *botnets* são utilizadas para desferir ataques de negação de serviço em massa (*Distributed Denial of Service*), propagação de código malicioso, coleta de informações em massa, e vale ressaltar que além do invasor poder utilizá-la como desejar, ele pode alugá-la para outras pessoas ou grupos que desejarem uma ação maliciosa específica.

Assim como o *spyware*, um possível sinal de infecção é a queda de desempenho do computador, ou falhas repentinas do mesmo, e a melhor maneira de se proteger continua sendo prevenindo que ele chegue até a máquina.

## 2.4 Ferramentas de Segurança

Abordado o lado das invasões, invasores e *malwares*, é importante apresentar as ferramentas que são utilizadas para contê-los e garantir a segurança do ambiente. Mas antes de fazê-lo, um fator significativo deve ser apontado, que é o fato de que grande parte das invasões, ou tentativas de invasões, que são executadas diariamente não são ataques complexos e arquitetados para um alvo em específico, pelo contrário, são ataques realizados por *scripts*, *bots*, *worms* e vírus, que visam alcançar a maior quantia possível de máquinas e tentam invadi-las incessantemente, uma vez que utilizam de recursos computacionais para o mesmo, coisa que um humano cansaria após algumas horas de tentativa e falha de força bruta. Tendo isso em vista, serão abordadas brevemente ferramentas de segurança e então será apresentado com maior detalhe os *honeypots*, foco deste projeto.

### 2.4.1 Firewall

Um *firewall* é uma ferramenta de segurança que monitora o tráfego da rede de entrada e saída e, baseado em um conjunto de regras de segurança definidas previamente, determina se permite ou bloqueia tráfegos específicos. Ao monitorar o tráfego da rede, cada pacote de dado é verificado para garantir que não existe nada malicioso no mesmo. Eles podem ser encontrados na forma de *software*, como um programa de segurança, ou *hardware*, como um roteador físico, no entanto ambos desempenham a mesma função (CISCO, 2017).

Os *firewalls* são uma necessidade, e é muito importante para a segurança do ambiente que ele esteja bem configurado, pois ele serve como uma barreira entre a rede interna, protegida e controlada, e as redes externas, que podem ser confiáveis ou não, como a própria Internet.

No entanto, como os *firewalls* trabalham com a análise do tráfego da rede, isso pode acarretar em lentidão e queda do desempenho computacional, especialmente se os pacotes estão sendo inteiramente analisados pelo computador. Além do fato de que alguns *firewalls* acabam bloqueando erroneamente sites legítimos, e embora corrigível, é um aborrecimento que pode acontecer habitualmente (KASPERSKY, 2017a).

### 2.4.2 Sistema de Detecção de Intrusão

Um sistema de detecção de intrusão (do inglês, *Intrusion Detection System*) é um sistema de segurança que monitora sistemas computacionais e tráfego de rede, analisando este tráfego para possíveis ataques hostis originados fora da organização e também para ataques originados de dentro da mesma (SANS, 2001).

Essa ferramenta é muitas vezes utilizada como complemento de um *firewall* para reforçar a segurança do sistema, de maneira que o *firewall* protege a rede interna de ataques maliciosos provindos da Internet e o SDI detecta caso algum destes ataques consiga burlar a segurança do *firewall* e tenta acessar qualquer sistema interno, alertando caso aconteça uma violação de segurança.

Porém, assim como no *firewall*, um dos problemas que pode acontecer ao utilizar um SDI é a lentidão e queda do desempenho da rede, uma vez que ele é responsável por estar constantemente analisando o tráfego da rede, e quanto maior for a rede, consequentemente mais pacotes de dados terão de ser analisados, requisitando mais recursos computacionais. Logo, dependendo do tamanho da rede em questão, utilizar um SDI pode ser sinônimo de gastos com um *hardware* que seja capaz de processar toda essa informação sem causar prejuízo ao desempenho geral.

### 2.4.3 Antivirus

Um antivírus é um tipo de *software* que visa prevenir, detectar e remover qualquer tipo de infecção de *malware* existente no computador, rede ou sistema em questão. Embora seu intuito seja remover qualquer tipo de *malware*, como todas aquelas citadas anteriormente e muitas outras existentes, ele foi originalmente desenvolvido com o intuito de solucionar problemas relacionados à vírus, e somente vírus ([TECHTARGET, 2017a](#)).

Eles funcionam operando em segundo plano no computador, assim como o *firewall* e o SDI, escaneando e procurando detectar qualquer tipo de atividade incomum, e então restringindo que o *malware* se propague. Um antivírus pode executar seus processos em tempo real, detectando e protegendo o alvo em questão, ou pode escanear diretórios e arquivos específicos, buscando padrões que indicam a presença de *malware*. Além disso, ele pode permitir que o usuário inicie novas buscas a qualquer momento, e quando de fato algo for encontrado, ele remove o *software* malicioso que foi detectado, podendo notificar o usuário de sua ação ou simplesmente removendo-o em segundo plano.

Assim como o *firewall* e o SDI, durante as varreduras realizadas pelo antivírus, é comum que se note uma queda no desempenho, devido à necessidade de consumo de recursos computacionais para sua execução. Além disso, outro problema é que os antivírus necessitam de acesso privilegiado ao sistema inteiro para que consigam realizar suas buscas com eficiência, e isso os torna alvo de invasores, uma vez que eles podem ter vulnerabilidades a serem aproveitadas para conquistar acesso privilegiado da máquina ao invasor.

## 2.5 Honeypots

Um *honeypot* é uma ferramenta de segurança muito potente, porém seu conceito pode aparentar ser um pouco confuso ao primeiro contato, uma vez que seu objetivo não é prevenir ataques, nem servir como uma barreira que impossibilita os invasores de ter contato com o ambiente, pelo contrário, seu objetivo é que ele seja sondado, atacado e até invadido. Eles são como iscas altamente monitoradas que servem para diversos propósitos, como distrair os invasores das máquinas que possuem valor real na rede, além de conseguir fornecer um aviso prévio sobre tendências de ataques e permitir um estudo detalhado sobre os invasores e suas técnicas de invasão ([SPITZNER, 2002](#)).

Por isso, *honeypots* funcionam de uma maneira diferente das ferramentas usuais de segurança, visto que eles não servem um único propósito, já que sua aplicação é extremamente flexível e contribui como um todo para a arquitetura de segurança. Ele pode ser utilizado para complicar invasões, como um *firewall*, pode ser utilizado para detectar invasões, como um SDI, pode ser aplicado para capturar e analisar ataques automáticos, como aqueles ataques realizados por *scripts*, *bots*, *worms* e vírus, e pode servir como um sensor de alerta, além de ter a capacidade de servir como objeto de pesquisa sobre as atividades maliciosas, capturando os

comandos ou comunicações dos invasores. A funcionalidade do *honeypot* e sua implementação dependem do administrador e de seu objetivo final.

Tendo isso em mente, vale ressaltar que eles não têm nenhum valor de produção, pois são apenas iscas que estão posicionadas para serem sondadas, logo qualquer comunicação feita com o *honeypot* se demonstra suspeita. Qualquer tráfego enviado para o *honeypot* provavelmente é sinônimo de um ataque e qualquer tráfego enviado pelo *honeypot* significa que o mesmo provavelmente foi comprometido e que o invasor está se comunicando com o exterior.

Dito isto, este tipo de ferramenta de segurança não é muito eficaz na prevenção de possíveis invasões, e isso se dá porque *honeypots* não detém de fato um invasor, pelo contrário, se mal implementado, um *honeypot* pode até ajudar um invasor e acabar sendo uma ameaça para a própria organização (NAZARIO, 2009). Por outro lado, *honeypots* podem funcionar como uma arma psicológica, aproveitando do efeito de “decepção ou dissuasão” para prevenir invasões, de maneira que, por decepção, um invasor perderia muito tempo e recursos atacando *honeypots* ao invés de sistemas reais de produção ou, por dissuasão, um invasor que descubra que há *honeypots* no ambiente fique com medo e desista de sua ação. Esse tipo de técnica, decepção ou dissuasão, somente funciona contra invasores humanos, e como os ataques mais comuns são aqueles que visam comprometer o maior número de sistemas e esses ataques não são feitos por humanos e sim ferramentas automatizadas ou *scriptadas*, essa técnica não se mostra muito eficaz na maioria dos casos, pois não há um indivíduo consciente e com psicológico por trás do ataque para ser decepcionado ou dissuadido. Felizmente, existem ataques que são feitos por humanos e então *honeypots* podem ser úteis para organizações que querem proteger seus recursos, confundindo os invasores e prevenindo seus ataques, já para aqueles casos em que os ataques não são feitos por humanos, os dados que são gerados conseguem ser muito úteis para realizar uma análise sobre a segurança.

Enquanto *honeypots* não são excepcionais na prevenção de invasões, eles se mostram de grande valor quando o assunto é detecção de invasões. Detectar invasões pode ser algo extremamente difícil, dado que uma boa ferramenta de detecção tem que ser capaz de eliminar falsos positivos, falsos negativos e produzir dados de valor. Um falso positivo podem ser considerado como qualquer comportamento normal ou esperado que é considerado como uma ameaça e falso negativo é qualquer alerta que deveria ter acontecido, como por exemplo um ataque, mas não aconteceu.

Logo, *honeypots* são ótimos para detecção de invasão porque eles lidam bem com esses três problemas fundamentais citados acima. Como *honeypots* não têm valor, e nem tráfego, de produção, não há muita atividade nele para gerar falsos positivos, que normalmente só irão ocorrer quando acontecer algum erro, mas na maioria dos casos um *honeypot* irá gerar somente alertas coerentes. Além disso, um *honeypot* funciona no conceito de que qualquer coisa que chegue até ele é considerada suspeita, então falsos negativos dificilmente irão ocorrer porque *honeypots* não são facilmente iludidos ou derrotados por novos *exploits*, diferentemente de

SDIs que têm de ser atualizados constantemente para não serem comprometidos por novas ferramentas ou *exploits* desenvolvidos pelos invasores. E como não há produção de tráfego para ser registrada, coletada e analisada em um *honeypot*, os dados gerados não são muitos, logo a maioria dos dados que são gerados são de grande valor, tornando fácil a análise da informação que é gerada por ele, que é muito diferente de outras ferramentas de detecção de intrusão, que chegam a gerar *gigabytes* de dados diariamente para serem analisados, o que dificulta, e muito, a análise desses dados e consequentemente o encontro de informação útil.

Vale ressaltar, no entanto, que um *honeypot*, apesar de simples e eficaz, não é uma ferramenta que irá solucionar todos os problemas relacionados à segurança. Ele é sim uma ótima ferramenta para detectar atividades não-autorizadas, mesmo com suas desvantagens, que serão tratadas em breve neste capítulo, e se for utilizado em complemento de outras ferramentas de segurança pode garantir uma ótima proteção ao ambiente.

Dito isto, as maneiras mais eficazes de prevenção continuam sendo as menos extravagantes, como desabilitar serviços inseguros, usar mecanismos fortes de autenticação e consertar sistemas vulneráveis, pois, se há um algum elemento vulnerável no ambiente, ele será comprometido e *honeypot* nenhum irá prevenir isso. E embora a prevenção ajude a diminuir os riscos, ela nunca irá eliminá-los, e existem diversos motivos para a prevenção vir a falhar e permitir uma invasão, como um *firewall* mal configurado ou usuários que utilizam senhas muito fáceis de serem quebradas.

Em suma, *honeypots* são ferramentas que apresentam flexibilidade e podem ser aplicadas em diferentes cenários na segurança. Eles coletam poucos dados, porém dados de grande valor, são simples e efetivos na detecção e captação de atividades não-autorizadas. Seus diferentes tipos, vantagens e desvantagens serão tratados abaixo ([PROVOS, 2003](#)).

### 2.5.1 Tipos de *honeypots* e níveis de interação

Como citado acima, os *honeypots* podem desempenhar diferentes funções dentro de uma arquitetura de segurança, e sua aplicação final será definida pelo administrador, que escolhe qual é a melhor maneira de implementar o *honeypot*, de maneira que satisfaça suas necessidades dentro do conjunto das ferramentas de segurança. Tecnicamente falando, os *honeypots* conseguem ser tão flexíveis por causa de algumas mudanças no núcleo de seu algoritmo, que refletem no conceito que será empregado e consequentemente na aplicação final de fato.

Com isso, eles podem ser divididos em diferentes categorias, que especificam qual será sua finalidade e como é seu funcionamento interno. Então, essencialmente os *honeypots* podem ser divididos em *honeypots* de baixa-interatividade e *honeypots* de alta-interatividade. Estas divisões se referem ao nível de interação que cada tipo de *honeypot* apresenta, permitindo classificá-los e compará-los de acordo com o nível de interação que um invasor tem com o



*honeypot* e com a quantidade de informações que um *honeypot* pode extrair de seu invasor.

#### 2.5.1.1 *Honeypots* de baixa interatividade

Os *honeypots* de baixa-interatividade, ou *honeypots* de produção, são aqueles onde são instaladas ferramentas para emular sistemas operacionais e serviços com os quais os atacantes irão interagir. (CERT, 2007).

Eles são fáceis de serem instalados, configurados, implementados e administrados por causa de sua simplicidade e seu funcionamento limitado. Sua principal função é a detecção de atividade não-autorizada e, como ele tem funcionalidade limitada, pode ser emulado por um programa, que simplesmente é instalado no sistema e então configurado para oferecer o que o administrador desejar.

Normalmente, eles apenas irão permitir que os invasores examinem e, potencialmente, se conectem a algumas portas de comunicação de rede. Por causa de sua simplicidade apresentam poucos riscos, pois como não há um sistema operacional ou serviços reais para o invasor interagir, o *honeypot* não pode ser utilizado, se comprometido, para danificar e comprometer outros sistemas. Logo, como a ação dos invasores é limitada, a informação que pode ser extraída do ataque e o risco ao sistema também são.

Esse tipo de *honeypot* normalmente é utilizado para proteger organizações, pois aumenta a segurança do sistema como um todo e ajuda a reduzir os riscos de invasões, mesmo ele não sendo bom para interagir com o invasor e extrair informações sobre o mesmo por causa de seu funcionamento limitado.

#### 2.5.1.2 *Honeypots* de alta interatividade

Os *honeypots* de alta-interatividade, ou *honeypots* de pesquisa, são aqueles nos quais os invasores interagem com sistemas operacionais, aplicações e serviços reais, (CERT, 2007).

Estes *honeypots* precisam de muito tempo e esforço para serem implementados e administrados devido a sua complexidade, porém, como recompensa, eles permitem adquirir um vasto conhecimento sobre o invasor. Seu objetivo é a pesquisa, coletando informações sobre o invasor, vulnerabilidades no sistema, ferramentas que são utilizadas para realizar ataques, entre outras infinitas possibilidades.

Sua capacidade de pesquisa se dá pelo fato que o nível de interação entre o invasor e o *honeypot* é altíssimo, o que permite um estudo profundo do ataque sofrido, e isso só ocorre porque esses *honeypots* permitem que o invasor interaja com um sistema ou serviço real e completamente funcional, ao invés de apenas emular seus serviços e aplicações, como os *honeypots* de baixa-interatividade. No entanto, se um *honeypot* de alta-interatividade é comprometido, o invasor tem acesso à um sistema completamente operacional e pode utilizá-lo para atacar e danificar outros sistemas ou capturar as atividades que estão acontecendo.

Então, ao trabalhar com uma ferramenta de tamanho poder, o risco atrelado à ela é imenso, e para minimizar esse risco é necessário muito empenho. Normalmente, um *honeypot* desses tem que ser colocado em um ambiente controlado e ser complementado com outras ferramentas de segurança, como um *firewall*, só que uma arquitetura dessas é complexa de ser implementada e administrada.

Logo, *honeypots* de alta-interatividade são muito mais complexos que os de produção, e têm maior funcionalidade, o que pode acabar sendo uma desvantagem, pois ele oferece maior risco ao sistema e exige maior esforço para ser implementado e administrado. A figura 10 retrata uma comparação entre os *honeypots* de baixa e alta interatividade, salientando suas principais diferenças conceituais e de aplicação.

Figura 10: Tipos de *honeypot*



Fonte: Elaborado pelo autor.

### 2.5.2 Vantagens

A flexibilidade do *honeypot* já foi apontada anteriormente como um de seus maiores atrativos, porém ele vai muito além disso. Seu conceito, desenvolvimento e implementação são tão sólidos devido à algumas de suas características intrínsecas, aqui abordadas em maior detalhe.

Organizações coletam grandes quantidades de dados diariamente, como *logs* de *firewall*, alertas de SDI e tráfego da rede, porém obter valor a partir destes dados é uma tarefa difícil devido a sua imensidão. Os *honeypots* atendem esse problema, pois, como eles não

têm nenhum valor de produção e portanto qualquer comunicação é suspeita, eles coletam poucos dados e consequentemente esses dados que são coletados possuem alto valor para a organização. Sendo assim, enquanto algumas organizações coletam *gigabytes* de dados diariamente, um *honeypot* usualmente coleta *megabytes*, e como pouca informação é coletada, é fácil separar os dados e identificar padrões que muitas organizações deixariam passar por estar ocupadas analisando uma quantia exorbitante de informação.

Outra vantagem dos *honeypots* é o fato de que eles não sofrem com o problema de falta de recursos, pois monitoram e capturam pouca atividade, diferente de outras ferramentas de segurança que são responsáveis pelo monitoramento de muita informação e constantemente sofrem com exaustão de recursos, como quando os *buffers* dos sensores dos Sistemas de Detecção de Intrusão ficam cheios e eles não conseguem mais monitorar a atividade da rede, ficando vulnerável.

Sendo assim, os *honeypots* não exigem grande investimento em *hardware* para que funcionem e cumpram sua função, diferente de *firewalls* e SDIs que podem requerer alta tecnologia e capacidade computacional do sistema, exigindo consequentemente que o *hardware* empregado seja potente o suficiente para poder realizar suas funções. Qualquer computador ultrapassado pode servir para a implementação de um *honeypot* e ainda assim conseguir ser executado sem problemas para monitorar redes que têm, por exemplo, velocidade de *gigabits* sem sofrer exaustão de recursos, e isso é uma grande vantagem.

E provavelmente uma das maiores vantagens do *honeypot* é sua simplicidade, pois ele não faz uso de algoritmos complexos, autenticações digitais, nem criptografias. Embora existam diferentes *honeypots* e alguns sejam mais complexos que outros, sua idéia e implementação continuam simples, o que o torna tão bom, pois quanto mais simples é um conceito, mais confiável e menos suscetível a erros e configurações erradas ele é.

### 2.5.3 Desvantagens

Uma desvantagem conhecida sobre os *honeypots* é o fato de que eles somente conseguem detectar ataques direcionados a eles, logo se um invasor infiltrar a rede e atacar uma variedade de sistemas, o *honeypot* não ficará ciente de nenhum destes ataques ao menos que ele seja um dos alvos. Portanto, se um invasor identifica a existência de um *honeypot* no sistema, ele pode evitá-lo e comprometer outros sistemas sem que o *honeypot* jamais perceba sua presença, e por isso é importante que sua implementação seja feita em conjunto com outras ferramentas de segurança, de maneira que uma complementa a outra, criando uma defesa sólida para quando ocorrer alguma falha, como esta citada.

Outra desvantagem é quando um invasor consegue identificar que está interagindo com um *honeypot*, o que normalmente acontece quando ele é implementado incorretamente ou possui algum comportamento ou padrão que o invasor reconhece como característico de

um *honeypot*, ao invés de um sistema real.

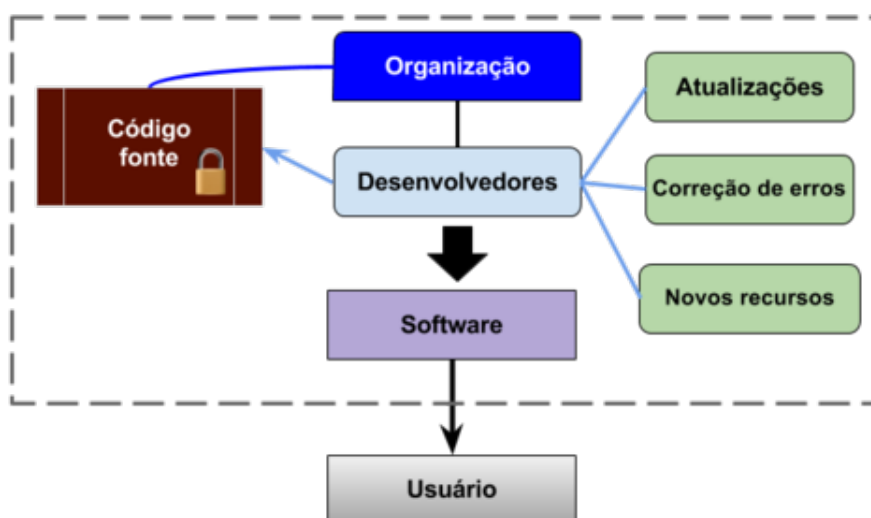
Além disso, o *honeypot* também traz risco ao ambiente, uma vez que ele se é atacado com êxito, ele pode ser utilizado para atacar ou comprometer outros sistemas. Porém, cada tipo de *honeypot* apresenta um nível de risco diferente para o ambiente, como já abordado anteriormente, de maneira que quanto mais simples for o *honeypot*, menos risco ele apresenta ao ambiente.

Portanto, um *honeypot* tem muitos pontos positivos que o tornam uma ferramenta sólida para garantir a segurança de um ambiente, porém ele não é a solução para todos os problemas que existem relacionados à segurança. E para se obter a melhor proteção do ambiente, ele deve ser implementado com outras ferramentas de segurança, além de que, como já ressaltado, a melhor prevenção continua sendo as mais simples, como salientar-se que não existem vulnerabilidades evidentes, pois se há um algum elemento vulnerável no ambiente, ele será comprometido.

## 2.6 *Open-source*

Quando se fala em propriedade, o pensamento comum é o direito de excluir uma pessoa de utilizar algo que não lhe pertence, como o autor de uma música que garante seus direitos sobre a mesma, de maneira que não seja vítima de plágio e usufrua plenamente de sua criação. Esse conceito de propriedade não se limita somente à música e se estende à praticamente todos os itens materiais de posse, e também à inúmeros itens não-tangíveis, como, por exemplo, o *software* (INITIATIVE, 2007).

Costumeiramente, um *software* de computador é comercializado somente como um produto final, que é instalado no computador do usuário através da cópia dos arquivos para os devidos diretórios e então, após instalado, pode ser utilizado normalmente. O usuário, nesse cenário, é apenas um consumidor e não interfere em nada no desenvolvimento do *software*, sua única função neste sistema é utilizar o *software*, e todo o processo de desenvolvimento do mesmo fica restrito aos desenvolvedores da organização responsável pela criação e comercialização do programa. Além do usuário não ter conhecimento exato sobre o que está por trás do que está sendo instalado, o que já é ruim por si só, quando uma organização deixa o mercado ou simplesmente pára de oferecer suporte a um produto, como o Windows ou Macintosh, os usuários deste produto ficam desamparados, especialmente pelo fato de que as correções de erros são totalmente dependentes dos fabricantes do *software*. Logo, este *software* é uma propriedade da organização que o criou, e somente ela exerce os direitos sobre o mesmo, além de serem os únicos com acesso ao código-fonte do programa, que é indispensável para a realização de atualizações, correções de erros e implementação de novos recursos ao *software*, vide figura 11.

Figura 11: Distribuição tradicional de *software* proprietário

Fonte: Elaborado pelo autor.

No meio desse conceito, décadas atrás surgia um movimento que se diferenciava e desafiava tal noção de propriedade, trazendo uma nova maneira de desenvolvimento de *software*, baseado em compartilhamento e colaboração de idéias, e também do código-fonte do programa. O movimento do *software open-source* representa um modelo diferente de distribuição de programas, o qual é fundamentado no princípio de garantir o direito de distribuição, não de exclusão.

O que começou apenas como uma hipótese acabou evoluindo em um movimento sofisticado, que resultou em alguns dos *softwares* mais estáveis, e utilizados, já produzidos. Projetos como GNU, Linux e Apache demonstraram que um sistema grande e complexo pode ser projetado, desenvolvido, mantido e expandido de maneira não-proprietária, na qual os desenvolvedores trabalharam em ambientes separados, desestruturados e de forma autônoma.

O *software open-source* é licenciado para garantir acesso gratuito ao código-fonte do programa, e isso significa que os usuários podem instalar, e utilizar, esse *software* sem nenhum pagamento, além de permitir que consigam suporte, ou até mesmo criem mecanismos de suporte para um produto cujo criador já deixou de oferecer assistência, se tornando independentes das grandes organizações que monopolizam os direitos sobre seus *softwares*, além de abrir a possibilidade de usuários e desenvolvedores autônomos tomarem as rédeas e contribuir com o crescimento do programa (GNU, 2007).

No entanto, é importante ressaltar que o movimento *open-source* não acaba com os lucros, o capitalismo ou os direitos de propriedade intelectual. Indivíduos e organizações estão criando produtos intelectuais e lucrando com projetos *open-source* ao mesmo tempo que inventam novos modelos de negócios e noções sobre propriedade.

### 2.6.1 História

Embora não faça parte do movimento *open-source*, por preferir do termo “*free-software*”, Richard Stallman foi um dos grandes responsáveis pela base do que hoje se tornou esse movimento.

Como programador de um laboratório do MIT nas décadas de 1970 e 1980, ele foi parte de uma comunidade de programadores que compartilhavam um *software*, e consequentemente seu código-fonte. Embora raro naquela época, seu laboratório era responsável pelo desenvolvimento do próprio *software* que utilizavam, e não se importavam de compartilhá-lo com ninguém. Qualquer pessoa podia analisar, copiar e fazer o que desejasse com o que era feito no laboratório. A cooperação fazia parte daquele ambiente.

Eventualmente, após frustrações com *softwares* proprietários, ele decidiu criar um sistema operacional com todas as ferramentas de *software* necessárias, como compiladores e editores, e implementar seu conceito de “*free-software*”.

Depois de anos e diversos acontecimentos, em 1998, se formava a “*Open-Source Initiative*” como uma organização educacional e administrativa, marcando um importante momento na história do desenvolvimento colaborativo e concretizando o que hoje é conhecido como o movimento *open-source*.

### 2.6.2 Definição do *Open-Source*

É importante ressaltar que *open-source* vai muito além do que simplesmente o compartilhamento do código-fonte de um *software*, pois existe uma política por trás que o define e garante seus termos de distribuição, além dos direitos de propriedade intelectual do criador do mesmo.

Sua definição é derivada do *Debian Free Software Guidelines* (DEBIAN, 2007), que por sua vez se baseia na GNU *General Public License*, por a considerar um exemplo de “*free-software*”.

- a) A licença não pode restringir ninguém de vender ou distribuir o *software* como um componente agregado à outro *software*, podendo conter programas de várias fontes diferentes. Logo, não pode haver o requerimento de *royalty* ou qualquer outro tipo de taxa referente à esta venda.
- b) O *software* deve incluir seu código-fonte e deve permitir a distribuição do mesmo, incluindo sua forma compilada. Este código-fonte não pode ser alvo de lucro e deve estar ao alcance do público, pois é através dele que os desenvolvedores modificam o *software*. Não é permitido ofuscar ou esconder o código-fonte deliberadamente.
- c) A licença deve permitir modificações e trabalhos derivados do *software*, e também deve permitir que estes sejam distribuídos sob os mesmos termos da licença original

do *software*.

- d) Não é permitido a discriminação contra qualquer pessoa ou grupo de pessoas.
- e) Não é permitido a restrição do uso do *software* em um campo específico de trabalho. Por exemplo, não se pode restringir o *software* de ser utilizado para pesquisas médicas, ou então em um negócio comercial.
- f) Os direitos atrelados ao *software* devem ser aplicados à todos aqueles que utilizam uma redistribuição do programa, sem necessidade da execução de novas licenças à esses envolvidos.
- g) Os direitos atrelados ao *software* não dependem de uma distribuição específica do mesmo. Independente da distribuição do programa, ele tem os mesmos direitos que foram garantidos em sua distribuição original.
- h) A licença não pode impor restrições no *software* que é distribuído a partir do *software* original licenciado. Logo, a licença não pode impor que distribuições provindas do *software* original devam também ser *open-source*.
- i) A licença deve ser neutra quanto à tecnologias, logo ela não pode baseada em qualquer tecnologia individual.
- j) É garantida a integridade do código-fonte do autor e sua propriedade intelectual.

### 2.6.3 Funcionamento

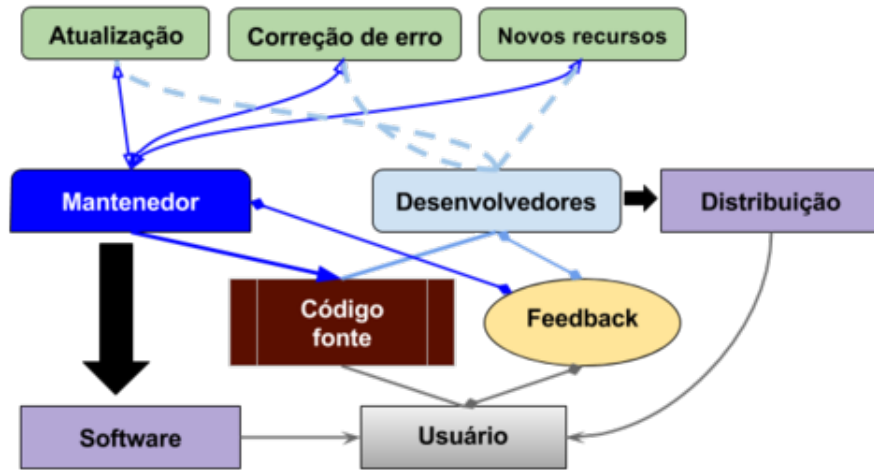
Embora sua licença garanta que qualquer um possa acessar e revisar o código-fonte do programa, apenas uma pessoa (ou um seletor grupo de pessoas) controla o *software* e incorpora nele correções, atualizações e adiciona recursos como novos lançamentos, que muitas vezes são contribuições da comunidade. Normalmente, essa pessoa é a própria criadora deste *software* ou então é um voluntário que foi escolhido pelo criador para manter o projeto adiante, sendo conhecidos como os mantenedores do projeto.

A comunidade é essencial no movimento *open-source*, e ela é composta por desenvolvedores e usuários, como visto na figura 12. Desenvolvedores, que são usuários com conhecimento técnico, contribuem diretamente com a evolução do *software*, e normalmente possuem uma área reservada para discussão dentro de um projeto *open-source*, onde discutem e contribuem com idéias, atualizações, correções e novos recursos. Já os usuários, que normalmente não possuem conhecimento técnico, contribuem relatando problemas com o *software* e pedindo auxílio em uma área de discussão reservada para os mesmos, que é monitorada por outros usuários, desenvolvedores e também pelos mantenedores do *software* (GNU, 2016).

A qualidade do *software open-source* não é proveniente de padrões rígidos de qualidade ou autocracia dos ideais do criador do mesmo. A estratégia do *open-source* é outra: ao disponibilizar atualizações e correções do *software*, o mantenedor do código consegue moldá-

lo através do *feedback* dos desenvolvedores e usuários, como se fosse uma “seleção natural” Darwinista.

Figura 12: Distribuição de *software open-source*



Fonte: Elaborado pelo autor.



## 3 Desenvolvimento

A utilização de *honeypots* em uma arquitetura de segurança agrega valor à mesma e ainda fornece preciosas informações que podem ser aplicadas à resolução dos problemas apresentados ao longo desse projeto. A flexibilidade dessa ferramenta, tanto no fator proteção, quanto no fator pesquisa, sustenta o porquê de ela ser utilizada por diversas organizações, além de auxiliar a compreender e solucionar padrões sobre os ataques sofridos.

Existem diversas organizações renomadas, como a Universidade do Texas, que utilizam *honeypots* em pesquisas, para buscar padrões e tendências de ataques globais, divulgando os resultados encontrados e empregando a informação obtida para procurar garantir a proteção do usuário. Além disso, há também àquelas que o utilizam apenas para incrementar sua própria segurança, e utilizam as informações coletadas pelo *honeypot* para assegurar a mesma.

Conforme proposto inicialmente, este projeto visa gerar resultados que salientem a importância da segurança da informação atualmente, além de propor a implementação de uma ferramenta de segurança gratuita, através da utilização de um *honeypot open-source*. Tendo isso em vista, um *honeypot* foi aplicado em um ambiente real, um laboratório da UNESP-Bauru, para incrementar sua segurança e coletar dados sobre ataques reais, após o mesmo ter sido alvo de uma invasão bem sucedida. Diversas ferramentas *open-source* foram estudadas antes da implementação final ser realizada, atentando-se às suas diferenças primordiais, e então foi escolhida a que melhor se combinou com a situação presente do laboratório.

A seguir, foram demonstrados com maiores detalhes os *honeypots* estudados no projeto, incluindo a metodologia por trás da escolha final e como ela foi utilizada para a resolução do problema enfrentado. Por fim, serão detalhados com dados e análises os resultados que foram coletados a partir da implementação do *honeypot* no laboratório.

### 3.1 Método de Pesquisa

Esta seção descreve a metodologia empregada neste trabalho de conclusão de curso. Conforme discutido ao longo do projeto, é enfrentado um problema de falta de segurança atualmente, e muitas são as ferramentas que visam garantir a proteção do usuário final, porém cada uma com características, especificidades e focos de aplicação final diferentes. É injusto apontar uma destas ferramentas como sendo melhor que a outra justamente porque suas aplicações diferem e, portanto, a solução ótima é que todas sejam implementadas em conjunto, visando abranger a maior quantidade de possíveis vulnerabilidades e assim fornecendo uma proteção completa. No entanto, para fins didáticos, este projeto foca na implementação de uma ferramenta só, o *honeypot*, pois sua capacidade de coleta de dados se mostra de

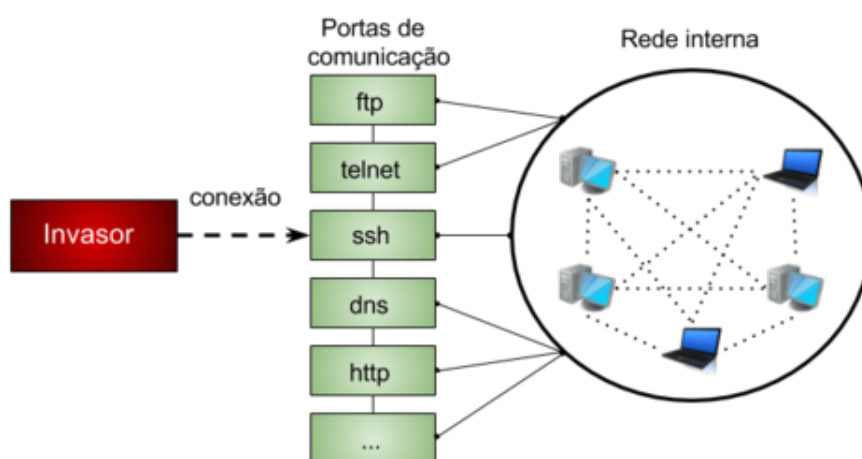
grande importância para demonstrar a veracidade do problema relacionado à segurança da informação.

Tendo isso em mente, os *honeypots* e suas características particulares foram abordados detalhadamente na Seção 2.5. Com base nas informações apresentadas, foi escolhido qual tipo de *honeypot* melhor se encaixaria na situação enfrentada no laboratório e então ele foi implementado para fornecer uma camada extra de segurança ao mesmo e coletar informações reais que servirão de base para a conclusão deste trabalho.

O laboratório em questão conta com diversos computadores que comunicam entre si através de sua rede interna e, como em qualquer outro ambiente, possuem conexão também com redes externas. A comunicação com as redes externas e outras máquinas ao redor do mundo se dá através das portas de comunicação de rede, como FTP, SSH, HTTP, entre outras. Dependendo da configuração desejada pelo administrador da rede, estas portas podem se encontrar abertas ou fechadas, e não precisam necessariamente estarem todas abertas ou todas fechadas, levando em consideração que devem ficar abertas apenas àquelas que de fato farão proveito da porta para se comunicar com o exterior e fechando as demais, para evitar vulnerabilidades e possíveis invasões.

O acontecido ocorreu quando houve uma tentativa de conexão na porta SSH, que estava aberta, e ela teve êxito. Após a conexão ter acontecido, o invasor possuía acesso à rede interna do laboratório e consequentemente à todas as máquinas lá presentes e suas informações, como representado na figura 13. Não cabe entrar em detalhes sobre o que o invasor fez após sua conexão ter sido bem sucedida, mas o que é de fato interessante é que havia uma vulnerabilidade evidente na porta SSH e ali estava a porta de entrada para outros potenciais invasores, circunstância essa que exigia que medidas fossem tomadas para evitar futuros problemas.

Figura 13: Invasão no laboratório



Fonte: Elaborado pelo autor.

## 3.2 Abordagem do problema real

Portanto, existia uma vulnerabilidade e era conhecido exatamente onde ela se encontrava, ou seja, o cenário perfeito para a implementação do *honeypot*, pois provavelmente outras tentativas de invasão ali seriam realizadas e, além de sua implementação incrementar a segurança do ambiente, dados significativos sobre ela seriam coletados.

Tendo isso em mente, foi necessário começar um estudo detalhado sobre a situação apresentada e como utilizar o *honeypot* em sua plenitude para tirar melhor proveito naquele contexto. Informações como tamanho da rede, quantidade de máquinas presentes, *hardware*, potência computacional das máquinas, portas de comunicação de rede e a atual arquitetura de segurança foram levadas em consideração para a realização da escolha do *honeypot* a ser implementado. Estes elementos são importantes pois, por exemplo, máquinas com poucos recursos computacionais não lidam bem com a implementação de um *honeypot* de alta-interatividade, logo seu desempenho seria comprometido, e como já mencionado anteriormente, o foco deste projeto é agregar valor ao ambiente como um todo sem causar prejuízos ao mesmo, como comprometendo seu desempenho ou exigindo gastos adicionais para que esse projeto seja implementado com sucesso.

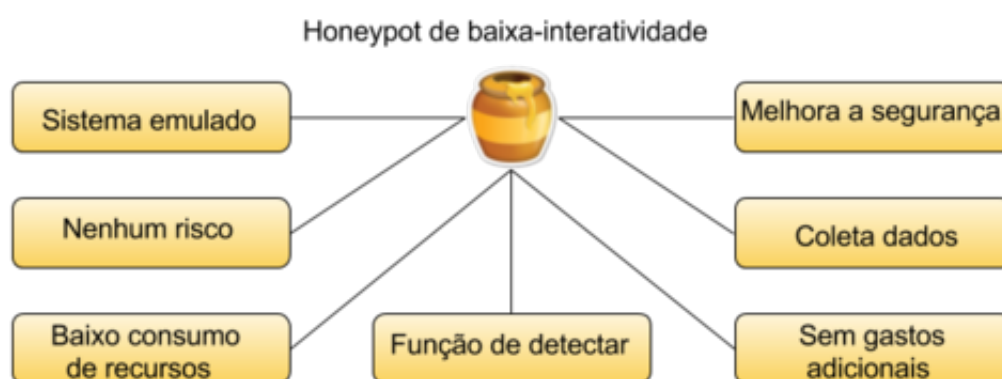
Dado a importância destes elementos, e levando em consideração as informações já apresentadas sobre *honeypots*, foi escolhido que melhor atenderia o problema enfrentado um *honeypot* de baixa-interatividade, pois, tendo em vista que já havia ocorrido uma invasão e as máquinas haviam sido comprometidas, implementar um *honeypot* de alta-interatividade, que poderia eventualmente ser corrompido e utilizado para atacar outros elementos do ambiente, estava fora de cogitação. A garantia da segurança do ambiente foi uma prioridade na escolha do *honeypot* a ser implementado. E então, como já mencionado anteriormente, por este tipo de *honeypot* ter como principal função a detecção de atividade não-autorizada e ser utilizado usualmente para assegurar a proteção do ambiente, ajudando a reduzir os riscos de invasões, ele foi o escolhido.

Além disso, o *honeypot* de baixa-interatividade apenas emula sistemas ou serviços, logo ele consome poucos recursos computacionais e isso por si só é excelente, uma vez que não é necessário que uma máquina potente fique à mercê do *honeypot* e desperdice seu potencial, que poderia estar sendo melhor aproveitado de outra maneira, portanto qualquer máquina do laboratório que fosse escolhida seria capaz de atender aos requisitos computacionais do *honeypot* tranquilamente, sem comprometer em nada seu funcionamento. E ainda, justamente por apenas emular sistemas ou serviços, a interação que o invasor tem com o *honeypot* é limitada, o que significa que não é possível que o *honeypot* seja comprometido pelo invasor e seja manipulado para atacar outras máquinas e sistemas, apresentando poucos riscos com sua implementação e ressaltando a prioridade dada à garantia de segurança interna, tratado na figura 14.

Quanto às desvantagens referentes à implementação de *honeypots*, como detalhadas anteriormente, essas eram inexistentes, pois mesmo que o invasor fosse capaz de identificar que estava interagindo com um *honeypot*, sua aplicação ainda válida e, no pior dos casos, o invasor iria interromper sua conexão, o que não comprometeria em nada a segurança do ambiente, nem traria nenhum risco ao mesmo, além de garantir que o mesmo não efetuou sua conexão, ou seja, ambiente intacto e sem invasores. Além disso, por mais que o *honeypot* só consiga detectar ataques direcionados à ele, essa detecção ainda era útil e a implementação do *honeypot* conveniente, porque, no pior dos casos, nenhum dado seria coletado e a situação do laboratório seria a mesma à antecedente a implementação desta ferramenta. Logo, a implementação do *honeypot* não traria nenhum risco adicional à segurança do ambiente e ainda acrescentaria proteção ao mesmo, além de fornecer dados relevantes, ou seja, somente pontos positivos.

Por fim, salientando um dos principais propósitos deste projeto, apresentar uma solução que esteja ao alcance de qualquer usuário, sem discriminação financeira, de maneira abrangente e gratuita, foi necessário descobrir ferramentas *open-source* que atendessem à todos os requisitos já citados e então escolher, entre os *honeypots* encontrados, aquele que melhor se encaixasse para cumprir seu papel na segurança do ambiente.

Figura 14: Elementos referentes à escolha do *honeypot* a ser implementado



Fonte: Elaborado pelo autor.

### 3.3 Ferramenta Utilizada

Após extensa pesquisa sobre *honeypots*, suas características, especificidades, vantagens, desvantagens, riscos e funcionamento, foi esboçado aquele que melhor atenderia às necessidades do problema encontrado no laboratório, já citado acima. Então, o próximo passo foi desenvolver uma ferramenta *honeypot open-source*, ou encontrar uma já existente e que atendesse aos requisitos apresentados.

Alguns *honeypots open-source* já desenvolvidos, e que cumpriam tais requisitos abordados na Seção 3.2, foram encontrados e então analisados para certificar-se que eles eram

adequados à resolução do problema enfrentado, no entanto apenas um foi escolhido para ser de fato implementado. Foi levado em consideração elementos como simplicidade de instalação, configuração e manutenção, tipo de *honeypot*, monitoramento do sistema e rede, coleta de dados e, como citado na Seção 2.6.3, estado da manutenção do código-fonte do projeto, para confirmar que o mesmo está atualizado e não possui vulnerabilidades antigas, que seriam facilmente exploradas e portanto acabariam por oferecer risco, ao invés de melhorar a segurança do ambiente.

Vale ressaltar que este trabalho somente detalha o *honeypot* que foi de fato selecionado e implementado, pois, além de ele estar atualizado, suas características encaixam-se com os requisitos almejados para a resolução do problema enfrentado no laboratório. Outros *honeypots* também foram encontrados e analisados, porém suas funcionalidades não se demonstraram tão completas quanto às oferecidas pelo *Artillery*, e muitos se encontravam desatualizados, o que por si só já é uma desvantagem, uma vez que esta ferramenta pode conter vulnerabilidades devido à sua desatualização.

### 3.3.1 *Artillery*

Esta ferramenta é um *honeypot* de baixa-interatividade desenvolvido em Python, que, conforme configurado, emula serviços semelhantes aos reais em portas de comunicação de rede que são alvos comuns de ataque, como a porta SSH, que foi a porta explorada e que permitiu que a invasão em evidência acontecesse. Além disso, ela, por padrão, monitora a porta SSH e busca tentativas de conexão por força-bruta, técnica utilizada por *scripts*, *worms*, vírus e *bots* para tentar se propagar rapidamente e infectar o maior número de máquinas possíveis. Ainda, além da porta SSH, ele pode ser configurado para monitorar diretórios específicos e detectar atividades não-autorizadas realizadas nos mesmos.

Toda essa atividade monitorada e detectada pelo *Artillery* é documentada, por padrão, em um arquivo de texto chamado “*banlist.txt*”, onde serão encontrados todos os endereços IPs das máquinas que tentaram conexão por força-bruta às portas na qual o *honeypot* se encontra posicionado e passa por um serviço real. Vale ressaltar que é configurável o número de tentativas de conexão até que o endereço IP em questão seja considerado responsável por uma tentativa de invasão por força-bruta e então banido para sempre, ou até sua remoção no arquivo “*banlist.txt*”, de tentar se conectar à máquina de que se fala.

Tendo isso em mente, sua instalação e configuração é simples, necessitando apenas que o administrador, ou o usuário, de que se fala saiba ler e escrever algumas sentenças em inglês. Fora que, apesar de ter sido implementado em uma plataforma Linux, este *honeypot* suporta tanto plataformas Windows como Linux, podendo contemplar uma gama maior de máquinas e consequentemente usuários.

Portanto, este foi o *honeypot* escolhido para ser implementado no laboratório, uma vez

que ele é de baixa-interatividade, logo não oferece riscos à segurança do ambiente e consome poucos recursos computacionais, além de emular seu serviço nas portas de comunicação e, por padrão, monitorar em tempo real a porta SSH, que é justamente aquela que foi responsável pela invasão ter ocorrido. E também, por automaticamente banir endereços IPs que tentam conexão por força-bruta, ele agrega muito valor à defesa como um todo, pois, além de detectar a tentativa de acesso por força-bruta e impedi-la de acontecer, ele ainda exclui o endereço IP responsável pela mesma, garantindo que este não irá nunca mais realizar qualquer outra tentativa de acesso à máquina, e assim diminuindo o número de potenciais invasores recorrentes. Sem contar que, ao banir tais endereços IPs, ele também os adiciona à um documento de texto, ou seja, coleta e fornece dados que são muito importantes para demonstrar a necessidade e importância do emprego de ferramentas de segurança atualmente. Por último, mas não menos importante, é uma ferramenta totalmente gratuita e de livre acesso à qualquer um que deseje utilizá-la, e até contribuir com seu desenvolvimento.

É importante, no entanto, ressaltar que o autor deste trabalho não foi o responsável pelo desenvolvimento desta ferramenta e não tem nenhum crédito quanto à criação da mesma. Todos os créditos referentes à ela são da Binary Defense Systems ([ARTILLERY, 2017](#)), organização criadora e mantenedora do Artillery. Como já abordado na Seção 2.6, essa organização é a responsável pelo *software*, porém qualquer um tem acesso ao Artillery e conseqüentemente seu código-fonte, sem a necessidade de pagamento sobre o mesmo para utilizá-lo. O autor deste trabalho foi responsável por encontrar, analisar e compreender o funcionamento da ferramenta de que se trata e perceber sua compatibilidade com o problema enfrentado, tendo então instalado e configurado o mesmo para que atendesse da melhor maneira possível o seu problema, mas não foi o responsável pela criação deste *software*.

Tendo isso em mente, este *honeypot* foi implementado com sucesso em uma das máquinas do laboratório precisamente às 22h55 do dia 28 de setembro de 2017 e permaneceu em execução até o começo de dezembro de 2017. Durante todo esse período, diversas tentativas não-autorizadas de acesso foram realizadas e devidamente documentadas. Portanto, a seguir, serão apresentados os resultados obtidos através da coleta dos dados alcançados através do Artillery, baseados na proposta deste trabalho.

### 3.4 Experimentos

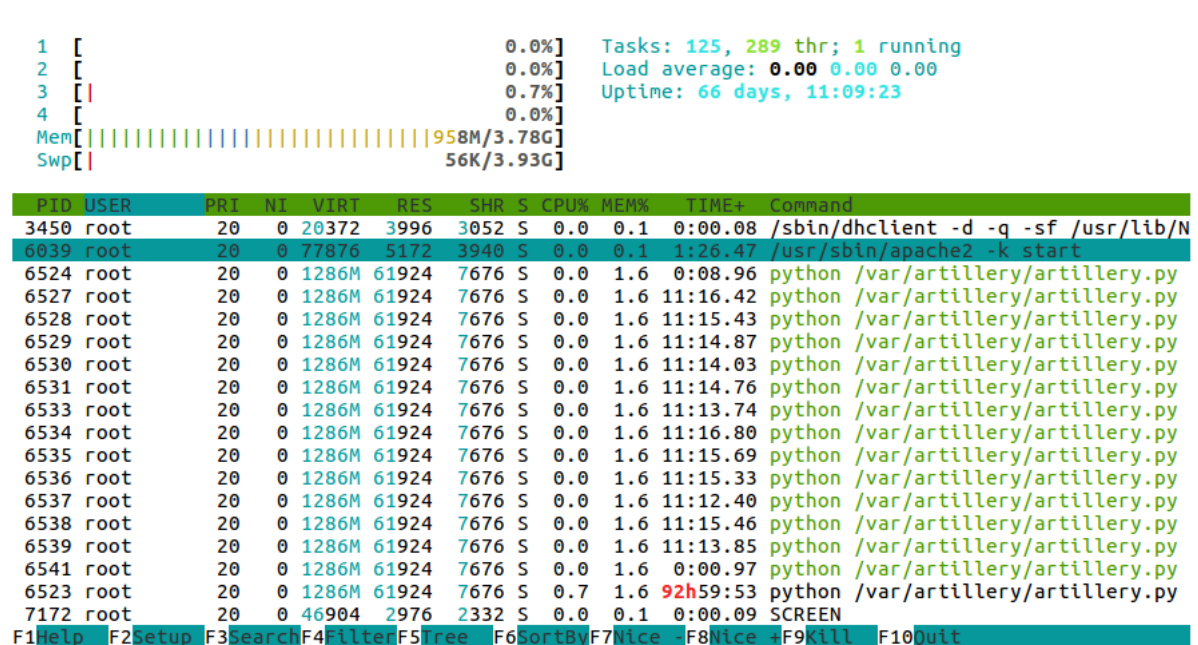
A oportunidade de implementar o *honeypot* em um ambiente real, onde uma invasão já havia ocorrido e portanto se demonstrava vulnerável, fornece à esse trabalho uma maior credibilidade, pois a ferramenta pôde ser testada em um local de risco, enfrentando uma problemática real e tendo seu desempenho analisado da maneira mais autêntica possível, aonde além disso, todos os dados coletados pelo *honeypot* são genuínos. Apesar de cada problema ter sua especificidade, e portanto diferentes soluções e abordagens, ainda é possível observar

características e certos padrões na aplicação desta ferramenta.

Os experimentos que aqui serão tratados foram realizados utilizando o *honeypot open-source* Artillery, que foi responsável por monitorar a porta de comunicação de rede 22, comumente referenciada como SSH, detectando e documentando todas atividades não-autorizadas realizadas na mesma. Além de ter sido também responsável pela exclusão dos endereços IPs que tentaram conexão por força-bruta nesta porta, adicionando-os à um arquivo de texto, que contém todos estes endereços maliciosos, e que servirá como a base de dado dos resultados que serão retratados. A configuração da máquina usada para a realização destes experimentos conta com um processador *Intel Core i5* (2 núcleos) e 4GB de memória RAM, além de utilizar o sistema operacional *Ubuntu Linux 16.04*.

A configuração dessa máquina pode ser considerada como eficiente, pois ela não conta com tecnologias obsoletas e *hardware* ultrapassado, mas mesmo assim é necessário frisar um ponto abordado anteriormente. É importante ressaltar, como foi abordado na Seção 3.2, que um *honeypot* de baixa-interatividade consome poucos recursos computacionais, e isso foi uma das razões que determinou na sua escolha para a implementação. Na figura 15, é possível observar o *honeypot* sendo executado em tempo real, precisamente encontrado na figura sob os processos abertos pelo usuário “root”, cujo comandos são “python /var/artillery/artillery.py”, os quais consomem, somados, apenas 22.4% da memória do computador.

Figura 15: Consumo computacional da ferramenta Artillery



Fonte: Elaborado pelo autor.

Como a ferramenta utilizada já vem com uma configuração básica pré-definida, as mudanças internas que o administrador ou usuário faz ao instalá-la não acarretam em grandes mudanças do funcionamento da mesma. No entanto, é claro que, por exemplo, ao modificar a



quantidade de tentativas de conexão que uma máquina pode fazer até elas sejam consideradas uma tentativa de ataque por força-bruta provavelmente irá mudar os resultados finais obtidos, havendo a possibilidade que eles sejam diferentes dos que foram obtidos neste trabalho. Mesmo assim, independente da configuração final realizada, a essência do Artillery será a mesma, o que pode mudar é a quantidade de dados coletados e, de acordo com o desejo dos responsáveis pela sua implementação, seu foco final de aplicação. Portanto, será apresentada com maiores detalhes a configuração que foi utilizada neste trabalho, empregada de maneira a solucionar esta problemática em questão, e então os resultados obtidos através da mesma.

### 3.4.1 Configuração utilizada

Como já foi dito na Seção 3.3.1, a instalação do Artillery é simples, pois basta que o usuário copie os arquivos do repositório do mesmo para sua máquina e então execute o arquivo “*setup.py*”, que instalará a ferramenta, com sua configuração padrão, na máquina. Uma vez instalado, é possível que o usuário acesse o arquivo “*config*” e mude detalhes específicos da configuração, de modo que melhor atenda seu propósito.

Figura 16: Início do arquivo “*config*”, encontrado no repositório do Artillery



```

GNU nano 2.5.3      File: config
#####$
#
# This is the Artillery configuration file. Change these variables and flags to$
# this behaves.
#
# Artillery written by: Dave Kennedy (ReL1K)
# Website: https://www.binarydefense.com
# Email: info [at] binarydefense.com
# Download: git clone https://github.com/binarydefense/artillery artillery/
# Install: python setup.py
#
#####$
#
# DETERMINE IF YOU WANT TO MONITOR OR NOT
MONITOR="ON"
#
# THESE ARE THE FOLDERS TO MONITOR, TO ADD MORE, JUST DO "/root","/var/", etc.
MONITOR_FOLDERS="/var/www","/etc/"
#
[ Read 142 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^M Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line

```

Fonte: Elaborado pelo autor.

Na figura 16, o arquivo “*config*” foi aberto utilizando o comando “*nano*”, nativo do sistema operacional Ubuntu, e o que se observa é o início do arquivo, que faz referência aos desenvolvedores do Artillery. No entanto, conforme se percorre o arquivo, vão aparecendo as configurações que o usuário pode modificar para ajustar a ferramenta à seu gosto.

As configurações padrões do Artillery que foram modificadas no caso do experimento realizado foram:

- a) A frequência na qual o Artillery monitora as portas de comunicação selecionadas, que foi modificada de “60 segundos” para “30 segundos”.



- b) O banimento automático dos endereços IPs, que foi modificado de “não” para “sim”.
- c) O monitoramento da porta de comunicação SSH para tentativas de conexão por força-bruta, que foi alterado de “não” para “sim”.
- d) O monitoramento da porta de comunicação FTP para tentativas de conexão por força-bruta, que foi alterado de “não” para “sim”.

Figura 17: Configuração utilizada no experimento

```

GNU nano 2.5.3      File: config      Modified
#####$
#
# DETERMINE IF YOU WANT TO MONITOR OR NOT
MONITOR="ON"
#
# THESE ARE THE FOLDERS TO MONITOR, TO ADD MORE, JUST DO "/root","/var/", etc.
MONITOR_FOLDERS="/var/www","/etc/"
#
# BASED ON SECONDS, 2 = 2 seconds.
MONITOR_FREQUENCY="30"
#
# PORT 22 CHECK
SSH_DEFAULT_PORT_CHECK="ON"
#
# EXCLUDE CERTAIN DIRECTORIES OR FILES. USE FOR EXAMPLE: /etc/passwd,/etc/hosts.allow
EXCLUDE=""
#
# DO YOU WANT TO AUTOMATICALLY BAN ON THE HONEYPOT
HONEYPOT_BAN="ON"
#
# WHITELIST IP ADDRESSES, SPECIFY BY COMMAS ON WHAT IP ADDRESSES YOU WANT TO WHITELIST
WHITELIST_IP="127.0.0.1,localhost"
#
# PORTS TO SPAWN HONEYPOT FOR
PORTS="22,1433,8080,21,5900,25,53,110,1723,1337,10000,5800,44443,16993"

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line

```

Fonte: Elaborado pelo autor.

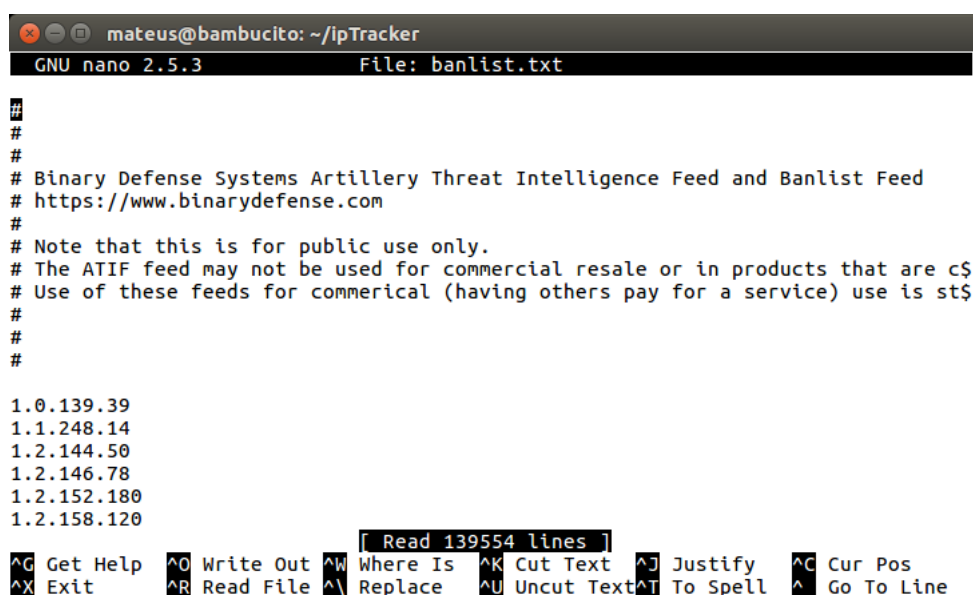
Algumas dessas modificações podem ser observadas na figura 17, mas, como já mencionado, a essência do Artillery acaba sendo a mesma independente das configurações utilizadas, porém é possível que com uma configuração diferente não se chegue aos mesmos resultados obtidos neste projeto com a configuração aqui empregada. A seguir, serão apresentados os resultados obtidos através da implementação desta ferramenta *honeypot*.

## 3.5 Resultados

Como apresentado na Seção 3.2, foi implementado um *honeypot* de baixa-interatividade, o Artillery, para apresentar uma solução ao problema de invasão enfrentado no laboratório em questão e testar sua funcionalidade e desempenho nos quesitos de proteção e coleta de dados. Vale ressaltar novamente, como declarado desde o início do projeto, que um dos propósitos do mesmo é ilustrar a importância que a segurança da informação apresenta na sociedade atual e elucidar o quão duvidoso é todo o meio que abrange a Internet.

Como já mencionado na Seção 3.3.1, essa ferramenta de segurança foi implementada com sucesso em uma das máquinas do laboratório e permaneceu em execução até este dado momento, podendo continuar assim por tempo indeterminado. Durante todo este tempo, houveram diversas tentativas de conexão por força-bruta à rede interna através da porta SSH, a mesma que havia sido a fonte da invasão anterior, e todas estas tentativas por força-bruta foram contidas pelo Artillery e devidamente documentadas pelo mesmo, que arquivou essas informações em um arquivo de texto chamado, por padrão, “*banlist.txt*”.

Figura 18: Arquivo gerado pelo Artillery, “*banlist.txt*”



```

mateus@bambucito: ~/IpTracker
GNU nano 2.5.3 File: banlist.txt

#
#
# Binary Defense Systems Artillery Threat Intelligence Feed and Banlist Feed
# https://www.binarydefense.com
#
# Note that this is for public use only.
# The ATIF feed may not be used for commercial resale or in products that are c$
# Use of these feeds for commerical (having others pay for a service) use is st$
#
#

1.0.139.39
1.1.248.14
1.2.144.50
1.2.146.78
1.2.152.180
1.2.158.120

[ Read 139554 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^_ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

```

Fonte: Elaborado pelo autor.

É possível observar na figura 18, após as primeiras linhas, que fazem referência à organização desenvolvedora da ferramenta, diversos números consecutivos que foram gerados pelo Artillery. Embora concatenados, cada um desses números faz referência à um único endereço de IP, que foi responsável por tentar se conectar através de força-bruta à porta SSH da rede interna do laboratório e teve sua tentativa detectada, impedida e documentada pelo *honeypot*.

É interessante destacar que, conforme se lê na parte inferior da figura, o arquivo contém 139554 linhas. Se excluídas as primeiras 13 linhas do arquivo, que são aquelas que não fazem referência aos endereços IPs, ainda sobram 139541 linhas, e todas essas linhas fazem referência à endereços IPs que tentaram realizar uma conexão por força-bruta na porta SSH da rede interna. Isso significa que houveram, dentro de um período menor que sessenta dias, mais de 139 mil tentativas de invasão ao laboratório.

O que intriga é pensar que o ambiente em questão se trata apenas de um laboratório qualquer dentro da UNESP Bauru, onde são realizadas algumas pesquisas, ou seja, não se trata de uma organização milionária com informações super confidenciais. Logo, em teoria, em um

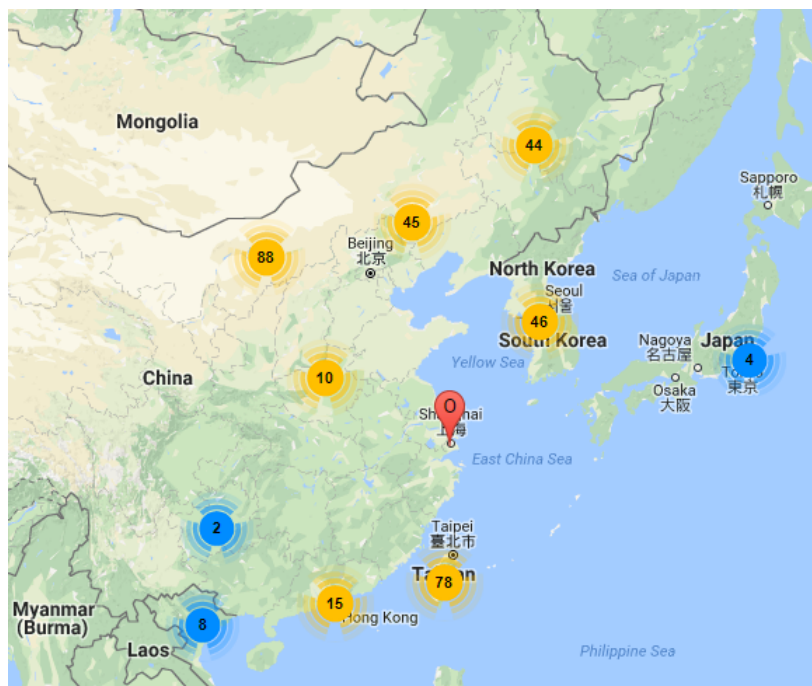
ambiente despretenso como este, a preocupação com segurança não deveria ser grande, afinal é apenas um laboratório qualquer, sem grandes atrativos para potenciais invasores, que deveriam ser poucos. E é então que se apresenta um dos grandes pontos positivos da implementação do *honeypot*, demonstrar que, independente da grandeza do ambiente em questão, existem sim tentativas de invasão e elas são inúmeras, porém se não fosse pelo *honeypot* elas nunca seriam documentadas e evidenciadas, o que cria a falsa impressão de que o ambiente é seguro, insignificante aos invasores e consequentemente livre de tentativas de invasão.

No entanto, existem informações sobre estes endereços IPs que devem ser acentuadas para que não sejam levantadas interpretações errôneas. Como citado na Seção 2.2, a grande maioria dos ataques que são realizados diariamente não são provindos de invasores com alto conhecimento técnico e com um foco de ação específico, pelo contrário, são aqueles que visam alcançar o maior número de máquinas possíveis e infectar a maior quantidade de máquinas que forem capazes. Tendo isso em mente, é válido enfatizar que o grosso destes endereços IPs coletados provavelmente são advindos de máquinas que estão infectadas, como abordado na Seção 2.3, com vírus, *worms* ou que são máquinas zumbis, ou seja, dificilmente algum desses endereços é referente à um invasor real, que de fato quis e tentou invadir explicitamente o laboratório em questão, até porque se o mesmo fosse fazê-lo, certamente que não realizaria essa ação utilizando seu endereço de IP real e o ocultaria através de VPNs ou máquinas zumbis.

Mesmo assim, como um complemento ao trabalho proposto, foi desenvolvido um *script*, utilizando Javascript, para transformar a informação coletada pelo *honeypot*, os endereços IPs, em suas respectivas coordenadas globais de latitude e longitude através da utilização de uma API pública (FREEGEOIP, 2017), e então, utilizando a API de geolocalização do Google Maps (GOOGLE, 2017), estas coordenadas foram maracadas no mapa-mundi fornecido pelo Google Maps. Este complemento serve apenas para transformar aquela massa de informação abstrata em algo visual para o leitor, e por mais que, como citado acima, as marcações que aparecem no mapa provavelmente não sejam referentes aos invasores reais, sua aplicação ainda é válida, pois demonstra a convergência de regiões que possuem máquinas que estão infectadas, ou que fazem parte da *botnet*, ou simplesmente são de fato utilizadas com o propósito de espalhar *malware* em âmbito global.

A figura 19 é referente à uma pequena amostra dos endereços IPs que foram coletados pelo *honeypot* e então tiveram sua geolocalização calculada e apresentada no mapa. Sua utilização é interessante pois permite, se processados todos os mais de 135 mil endereços obtidos, analisar focos de *botnet*, máquinas infectadas por *malware* e possíveis invasores reais espalhados pelo mundo. Porém, a análise vai além disso, uma vez que, se uma região apresenta-se com maiores números, como Taiwan (representado pelo número 78 na figura) por exemplo, significa que ali pode representar uma região que sofre com a falta de segurança e educação sobre conceitos básicos de como se proteger ao utilizar um computador conectado

Figura 19: Geolocalização de uma amostra dos endereços IPs coletados



Fonte: Elaborado pelo autor.

à Internet, e por isso há essa convergência de ataques realizados a partir daquela região. Isso pode ser um dado útil para o responsável pela segurança de rede de uma organização, pois ele pode prontamente bloquear a faixa de endereços IPs referente àquela região e não ter que se preocupar com ataques originários de lá.

A partir das análises desenvolvidas no experimento, foi possível concluir todo o estudo embasado na fundamentação teórica apresentada neste trabalho, demonstrando a necessidade da segurança da informação, uma vez que os inimigos são muitos, e tendo realizado todo esse estudo através da implementação de uma ferramenta gratuita, que está ao alcance de qualquer um que desejar utilizá-la.

## 4 Conclusão

O estudo na área de segurança da informação por si só engloba diversas outras áreas de conhecimento da própria computação, tornando-se uma área muito abrangente e responsável por fornecer diversas ferramentas e conceitos que são empregados na resolução de vários problemas encontrados atualmente, que não se limitam somente à computação propriamente dita. Essa é uma área que requer constante estudo e atualização, pois ela conta com um adversário incansável e que está em constante mutação, visando sempre burlar os mecanismos de defesa instalados e aproveitar de vulnerabilidades existentes.

Com a popularização dos computadores e do acesso à Internet, o número de usuários cresce diariamente, e com isso, cresce também o número de alvos para os invasores, que contam com a ingenuidade e ignorância de muitos usuários para se beneficiar. No entanto, enquanto cresce o número de usuários, e possivelmente de invasores também, a busca pelo conhecimento na área de segurança de informação não parece ter acompanhado os mesmos passos, seja por negligência ou simplesmente por puro desconhecimento, e isso é alarmante. Em uma era onde tudo converge à digitalização, onde bancos disponibilizam seus serviços pelo smartphone, onde fotos são compartilhadas on-line, onde inúmeras compras e vendas são realizadas diariamente ao redor do mundo, ou seja, quando tudo tende à utilização de meios digitais para a realização das atividades cotidianas, é necessário que haja uma maior preocupação com a segurança destes dados, que se mostram vulneráveis, para que eles não caiam nas mãos das pessoas erradas.

É fato que a cibercriminalidade se tornou um negócio rentável para muitos, inclusive existem mercados inteiros dedicados à compra e venda de *malwares* e dados, teoricamente confidenciais, de usuários. Simplesmente por existir tal oferta e procura, fica evidente que há um grande interesse no lado dos criminosos em obter tais informações dos usuários, além de também haver interesse em adquirir as ferramentas que são utilizadas para se aproveitar das vulnerabilidades que os levem até essas informações. E então é necessário que também haja um interesse de mesma proporção no outro lado dessa moeda, aquele responsável por montar a defesa que irá garantir a segurança da informação, caso contrário os criminosos triunfam com facilidade sobre sistemas inseguros.

Por isso se levanta a questão sobre como se proteger, quando informação sobre computadores e segurança on-line não é amplamente divulgado pela mídia e o conhecimento referente à mesma não está ao alcance de todos. Além disso, para aqueles que sabem de fato os passos que devem ser seguidos para atingir uma defesa sólida, existem barreiras financeiras no caminho, que impedem de montar uma boa arquitetura de segurança sem que haja investimentos pesados em *hardware* que aguarde altos níveis de processamento computacional ou então taxas

periódicas para a utilização de *softwares* privados de segurança.

A fundamentação teórica que constitui esse projeto almeja apresentar conceitos básicos, porém necessários, sobre a área de segurança da informação, modos de se proteger e características referentes ao invasor, de maneira que essa informação esteja ao alcance do maior público possível e dissemine esse conhecimento tão essencial atualmente. Nesse quesito, o desconhecimento é o pior inimigo do usuário, pois é através da ignorância e inocência do mesmo que os invasores se apoiam.

Os experimentos desenvolvidos por esse trabalho buscam auxiliar na visualização do que realmente acontece diariamente e ninguém de fato vê, embasando-se na metodologia estudada para comprovar a existência dos invasores e como a falta de segurança pode levar um ambiente aos escombros. Além de fazê-lo através da implementação de uma ferramenta completamente gratuita, fator que demonstra a possibilidade de obter uma defesa sólida sem a necessidade de grandes investimentos. Os resultados obtidos através dos dados coletados pelo *honeypot* são de grande importância para a análise de focos globais de *botnet*, regiões com carência de segurança computacional e até possíveis centros de propagação de *malware*.

Portanto, este projeto abrange a união de conceitos ligados à Ciência da Computação na área de segurança da informação, de maneira que contribui para o desenvolvimento do conhecimento nesta área tão importante em uma era onde tudo converge à digitalização e há a necessidade da proteção das informações dos usuários. O material apresentado por este projeto se mostra conciso e compreensível aos diversos níveis técnicos que o público pode possuir, podendo servir de apoio para ser utilizado em algumas disciplinas de segurança que são ministradas na área de computação, mais voltado à difusão do conhecimento propriamente dito e à prática do mesmo. A implementação do *honeypot* abordado neste projeto não fornece somente proteção ao ambiente no qual ele é instalado, mas também auxilia no processo de geração de informação, que pode ser analisada e a partir dela extrair dados valiosos e relevantes, além de contribuir para a formação do aluno que a estuda, uma vez que há a necessidade de combinar diversos conhecimentos computacionais e lógicos para a resolução das problemáticas enfrentadas, o que torna essa área tão interessante, pois os desafios são contínuos e estão em constante metamorfose. Os trabalhos futuros resumem-se no estudo e utilização de outras ferramentas de segurança *open-source*, com o intuito de formar uma arquitetura completa de segurança baseada somente em *software* gratuitos, e também na implementação de um *honeypot* de alta-interatividade, com o objetivo de estabelecer uma comunicação maior com o invasor e então estudar com maiores detalhes as técnicas que são utilizadas por ele ao realizar uma invasão, contribuindo para uma geração de dados maior e mais específica.

# Referências

- ANDERSON, J. P. *Computer security threat monitoring and surveillance*. Fort Washington, Pennsylvania, 1980.
- ARTILLERY. *GitHub - BinaryDefense*. 2017. <https://github.com/BinaryDefense/artillery>. Acesso em: 16 ago 2017.
- CERT. *Honeypots e Honeynets: Definições e Aplicações*. 2007. <https://www.cert.br/docs/whitepapers/honeypots-honeynets>. Acesso em: 21 abr 2017.
- CERT. *Códigos Maliciosos*. 2017. <https://cartilha.cert.br/malware/>. Acesso em: 16 out 2017.
- CISCO. *O que é um firewall?* 2017. [http://www.cisco.com/c/pt\\_br/products/security/firewalls/what-is-a-firewall.html](http://www.cisco.com/c/pt_br/products/security/firewalls/what-is-a-firewall.html). Acesso em: 18 jun 2017.
- DEBIAN. *Contrato Social Debian*. 2007. [https://www.debian.org/social\\_contract](https://www.debian.org/social_contract). Acesso em: 18 ago 2017.
- FREEGEOIP. *freegeoip.net*. 2017. <https://freegeoip.net/>. Acesso em: 17 nov 2017.
- GNU. *A licença Pública Geral GNU v3.0*. 2007. <https://www.gnu.org/copyleft/gpl.html>. Acesso em: 18 ago 2017.
- GNU. *Free Software: Freedom and Cooperation*. 2016. <https://www.gnu.org/philosophy/rms-nyu-2001-transcript.html>.
- GOOGLE. *A Google Maps Geolocation API*. 2017. <https://developers.google.com/maps/documentation/geolocation/intro?hl=pt-br>. Acesso em: 17 nov 2017.
- HACKMAGEDDON. *August 2017 Cyber Attacks Statistics*. 2017. <http://www.hackmageddon.com/2017/09/12/august-2017-cyber-attacks-statistics/>. Acesso em: 03 out 2017.
- IBGE. *IBGE: Instituto Brasileiro de Geografia e Estatística*. 2015. <https://ww2.ibge.gov.br/home/estatistica/populacao/acessointernet>. Acesso em: 23 abr 2017.
- IMA-SP. *Política de Segurança*. 2005. <https://ima.sp.gov.br/politica-de-seguranca-da-informacao>. Acesso em: 08 out 2017.
- INITIATIVE, O. S. *Open Source Initiative*. 2007. <https://opensource.org/>.
- KASPERSKY. *Kaspersky Security Bulletin*. 2016. [https://kasperskycontenthub.com/securelist/files/2016/12/Kaspersky\\_Security\\_Bulletin\\_2016\\_Statistics\\_ENG.pdf](https://kasperskycontenthub.com/securelist/files/2016/12/Kaspersky_Security_Bulletin_2016_Statistics_ENG.pdf). Acesso em: 27 abr 2017.
- KASPERSKY. *O que é um firewall?* 2017. <https://www.kaspersky.com.br/resource-center/definitions/firewall>. Acesso em: 14 nov 2017.
- KASPERSKY. *Spyware Definition and Prevention*. 2017. <https://usa.kaspersky.com/resource-center/threats/spyware>. Acesso em: 13 nov 2017.

KASPERSKY. *What is a computer virus or a computer worm?* 2017. <https://www.kaspersky.com/resource-center/threats/viruses-worms>. Acesso em: 09 nov 2017.

MICROSOFT. *Ransomware FAQ*. 2017. <https://www.microsoft.com/en-us/wdsi/threats/ransomware>. Acesso em: 13 nov 2017.

NACHENBERG, C. Computer virus-antivirus coevolution. *Commun. ACM*, ACM, New York, NY, USA, v. 40, n. 1, p. 46–51, jan. 1997. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/242857.242869>>.

NAZARIO, J. Phoneyc: A virtual client honeypot. In: *2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats, LEET '09, Boston, MA, USA, April 21, 2009*. [s.n.], 2009. Disponível em: <<https://www.usenix.org/conference/leet-09/phoneyc-virtual-client-honeypot>>.

PROVOS, N. *Honeyd: A Virtual Honeypot Daemon*. [S.l.], 2003. v. 2. Disponível em: <<http://www.citi.umich.edu/u/provos/papers/honeyd-eabstract.pdf>>.

SANS. *Intrusion Detection Systems*. 2001. <https://www.sans.org/reading-room/whitepapers/detection/intrusion-detection-systems-definition-challenges-343>. Acesso em: 18 jun 2017.

SANS. *The Information Security Process*. 2002. <https://www.giac.org/paper/gsec/501/information-security-process-prevention-detection-response/101197>. Acesso em: 08 out 2017.

SPITZNER, L. *Honeypots: Tracking Hackers*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002. ISBN 0321108957.

STALLINGS, W. *Network Security Essentials: Applications and Standards*. 4th. ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2010. ISBN 0136108059, 9780136108054.

SYMANTEC. *Bots and Botnets - A Growing Threat*. 2016. <https://us.norton.com/botnet/>. Acesso em: 13 nov 2017.

SYMANTEC. *Trojan Horse (cavalo de troia)*. 2017. [https://www.symantec.com/pt/br/security\\_response/glossary/define.jsp?letter=t&word=trojan-horse](https://www.symantec.com/pt/br/security_response/glossary/define.jsp?letter=t&word=trojan-horse). Acesso em: 13 nov 2017.

SYMANTEC. *What is A Computer Virus?* 2017. <https://us.norton.com/internetsecurity-malware-what-is-a-computer-virus.html>. Acesso em: 09 nov 2017.

TANENBAUM, A. *Computer Networks*. 4th. ed. [S.l.]: Prentice Hall Professional Technical Reference, 2002. ISBN 0130661023.

TECHTARGET. *What is antivirus software?* 2017. <http://searchsecurity.techtarget.com/definition/antivirus-software>. Acesso em: 14 nov 2017.

TECHTARGET. *What is backdoor (computing)?* 2017. <http://searchsecurity.techtarget.com/definition/back-door>. Acesso em: 13 nov 2017.