

Faculdade de Tecnologia de Americana  
Curso Superior de Tecnologia em Segurança da Informação

JUNIOR APARECIDO DOS SANTOS DE ARRUDA

**ORCHID – Sistema de segurança resiliente utilizando  
algoritmos imunoinspirados**

JUNIOR APARECIDO DOS SANTOS DE ARRUDA

**ORCHID – Sistema de segurança resiliente utilizando algoritmos  
imunoinspirados**

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Segurança da Informação, sob a orientação da Prof. Dra. Maria Cristina Aranda.

Área: Segurança da Informação

AMERICANA, SP  
2015

**FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS**  
**Dados Internacionais de Catalogação-na-fonte**

A817o	<p>Arruda, Junior Aparecido dos Santos de ORCHID: sistema de segurança resiliente utilizando algoritmos imunoinspirados. / Junior Aparecido dos Santos de Arruda. – Americana: 2015. 73f.</p> <p>Monografia (Graduação em Tecnologia de Segurança da Informação). - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza. Orientador: Prof. Dr. Maria Cristina Aranda</p> <p>1. Segurança em sistemas de informação I. Aranda, Maria Cristina II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana.</p> <p>CDU: 681.518.5</p>
-------	---

**Junior Aparecido dos Santos de Arruda**

**ORCHID – Sistema de Segurança Resiliente Utilizando  
Algoritmos Imunoinspirados**

Trabalho de graduação apresentado  
como exigência parcial para obtenção do  
título de Tecnólogo em Segurança da  
Informação pelo CEETEPS/Faculdade de  
Tecnologia – FATEC/ Americana.  
Área de concentração: Segurança da  
Informação

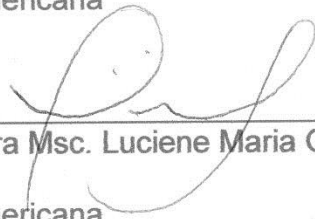
Americana, 22 de junho de 2015.

**Banca Examinadora:**



---

Professora Dra. Maria Cristina Aranda (Presidente)  
Doutora  
Fatec Americana



---

Professora Msc. Luciene Maria Garbuio Castello Branco (Membro)  
Mestre  
Fatec Americana



---

Professor Raul Paiva de Oliveira (Membro)  
Graduado  
Fatec Americana

## **AGRADECIMENTOS**

Agradeço a Deus em primeiro lugar, por sua presença constante em minha vida. Gostaria de agradecer à minha orientadora, Prof. Dra. Maria Cristina Aranda, por seu apoio. Por sua disposição em nossas conversas, pelo incentivo e principalmente por todas as incorreções apontadas. Enfim, obrigado por ter me ajudado na conclusão deste trabalho.

Agradeço imensamente aos autores, que tiveram suas teses formando a base de estudo na confecção deste trabalho, em especial Dr. Gabriel Dieterich Cavalcante e Dr. Joaquim Quinteiro Uchôa.

Aos meus amigos da turma 2009 ASTI, por me ajudarem a manter a sanidade no ambiente acadêmico, principalmente pelos momentos de descontração e café. Em especial, Carlão, Darth Vader, Ícaro, Ronaldo, Sergio, Felipão, Lester, JP.

Por fim, agradeço e peço humildes desculpas aos amigos e familiares por minha ausência durante o desenvolvimento desse trabalho.

*Ao meu pai Cicero e minha mãe Aparecida (in memoriam), minha esposa  
Fabiana e ao meu filho Vinicius, pelo apoio e compreensão.*

*“Uma inteligência artificial real seria inteligente o suficiente para não revelar que era realmente inteligente.”*

George Dyson

## RESUMO

A evolução da Internet e a contínua interconexão de computadores e outros dispositivos digitais têm criado uma série de oportunidades nas mais diversas áreas de atuação humana. Entretanto, esse processo trouxe consigo uma série de problemas, entre eles o crescente número de sistemas computacionais invadidos por intrusos. Dispositivos de segurança de rede encontram-se sob intenso ataque. Os ataques de hoje, rotineiramente contornam sistemas baseados em assinatura e, portanto, necessitam de fontes de dados adicionais além de simplesmente detectar tráfego de ataque específico. Os atacantes estão mais hábeis na sua capacidade de penetrar as redes das organizações, ao passo que novas ameaças são lançadas cada vez mais rápido, tirando vantagem de portas de comunicação e serviços comumente permitidos para garantir sua passagem através das fronteiras de segurança. O resultado tem sido uma erosão contínua da eficácia dos *firewalls* de rede e, conseqüentemente, a iluminação de falhas fundamentais no projeto inicial e posteriores modificações para estes elementos fundamentais da maioria das estratégias de segurança da empresa. Defensores precisam de sistemas inteligentes, que forneçam dados significativos para detectar ataques avançados, provendo dados para correlacionar e analisar eventos em diferentes sistemas de segurança. O trabalho aqui apresentado aborda esse problema utilizando uma arquitetura integrada que combina verificação de integridade e mecanismos de recuperação além de metodologias e algoritmos inspirados em conceitos e processos do sistema imune inato e adaptativo, solucionando uma série de problemas e falhas existentes nos dispositivos atuais.

Palavras-chave: Segurança da Informação; IDS; IPS; *Firewall*.



## **ABSTRACT**

The evolution of the internet and the continued interconnection of computers and other digital devices has created many opportunities in various fields of human activity. However, this process has brought a number of issues, including the growing number of computer systems hacked by intruders. Network security devices are under intense attack. Today's attacks routinely circumvent signature-based systems and therefore require additional data sources beyond simply detect specific attack traffic. Attackers are more skilled in their ability to penetrate corporate networks, whereas new threats are released faster and faster, taking advantage of communication ports and services commonly allowed ensuring its passage through the security boundaries. The result has been a steady erosion of the effectiveness of network firewalls and consequently the lighting of fundamental flaws in the initial design and subsequent modifications to these fundamental elements of most security strategies of the company. Defenders need intelligent systems that provide meaningful data to detect advanced attacks, providing data to correlate and analyze events in different security systems. The work presented here addresses this problem by using an integrated architecture that combines integrity checking and recovery mechanisms beyond methodologies and algorithms inspired by concepts and processes of the innate and adaptive immune system, solving a number of problems and flaws in current devices.

Keywords: Information Security, IDS, IPS, *Firewall*.

## LISTA DE FIGURAS

Figura 1 - Perfil das Empresas Participantes da Pesquisa.....	17
Figura 2 - Frequência de Incidentes.....	18
Figura 3 - Avaliação de Confiança dos Entrevistados .....	19
Figura 4 - Linhas de Pesquisa para o Desenvolvimento da Computação Natural.....	33
Figura 5 - Arquitetura Multicamada do Sistema Imune .....	35
Figura 6 - Agentes Causadores de Doenças.....	35
Figura 7 - Disposição de Camadas do Orchid.....	41
Figura 8 - IA-AIS Visão Geral.....	45
Figura 9 - DT-AIS Visão Geral.....	46
Figura 10 - Visão Geral do SELINUX.....	50
Figura 11 - Fluxo Geral de Operação do REsquared .....	53
Figura 12 - Ambiente dos Testes .....	57
Figura 13 - Resultados dos testes com o algoritmo IA-AIS na base KDDTest+ .....	60
Figura 14 - Resultados dos testes com o algoritmo IA-AIS na base KDDTest-21.....	61
Figura 15 - Resultados dos testes com o algoritmo DT-AIS na base KDDTest+ .....	62
Figura 16 - Resultados dos testes com o algoritmo DT-AIS na base KDDTest-21 ...	63

## LISTA DE TABELAS

Tabela 1 - Tipos de Firewall .....	32
Tabela 2 - Amostra de 25%.....	59
Tabela 3 - Amostra de 75%.....	59
Tabela 4 - Amostra de 100%.....	60
Tabela 5 - Resultados dos testes com o algoritmo IA-AIS na base KDDTest+ .....	60
Tabela 6 - Resultados dos testes com o algoritmo IA-AIS na base KDDTest-21 .....	61
Tabela 7 Resultados dos testes com o algoritmo DT-AIS na base KDDTest+.....	62
Tabela 8 Resultados dos testes com o algoritmo DT-AIS na base KDDTest-21.....	62

## **LISTA DE ABREVIATURAS E SÍMBOLOS**

AISF	Artificial Immune System Framework
CPU	Central Process Unit
DPI	Deep Packet Inspection
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IP	Internet Protocol
IPS	Intrusion Prevention System
MAC	Mandatory Access Control
NFS	Network File System
NGFW	Next Generation Firewall
SELINUX	Security-Enhanced Linux
SPA	Single Packet Authorization
TCP/IP	Transmission Control Protocol/Internet Protocol
TFTP	Trivial File Transfer Protocol
VCS	Version Control System
VOIP	Voice Over IP

## SUMÁRIO

<b>1</b>	<b>Introdução</b> .....	<b>12</b>
<b>2</b>	<b>Segurança Computacional</b> .....	<b>15</b>
2.1	Necessidade de Segurança .....	17
2.2	O Processo de Segurança .....	20
2.2.1	Avaliação .....	20
2.2.2	Proteção .....	20
2.2.3	Detecção .....	22
2.2.4	Resposta .....	24
<b>3</b>	<b>Sistemas de Detecção de Intrusão</b> .....	<b>25</b>
3.1	Processo de Detecção de Intrusão.....	25
3.1.1	Estratégias de Sistemas de Detecção de Intrusão .....	26
3.2	Tipos de Ataques.....	28
<b>4</b>	<b>Firewalls</b> .....	<b>31</b>
4.1	Testes de Firewalls.....	31
4.2	Firewalls de Próxima Geração .....	31
4.3	A Biologia Como Inspiração .....	33
4.3.1	O Sistema Imune Humano .....	33
4.3.2	Sistemas Imune Artificiais .....	37
4.4	Sistemas Resilientes .....	38
<b>5</b>	<b>Segurança em Camadas</b> .....	<b>40</b>
5.1	Camada de Detecção e Bloqueio.....	41
5.1.1	Base de Dados .....	42
5.1.2	Algoritmo IA-AIS.....	43
5.1.3	Algoritmo DT-AIS.....	46
5.1.4	AISF (Artificial Immune System Framework) .....	47
5.2	Camada de Segurança Interna .....	48
5.2.1	Hardening.....	48
5.3	Camada de Recuperação e Disponibilidade .....	52
5.3.1	Sistemas de Controle de Versão .....	52
5.3.2	REsquared – Recuperação com Versionamento .....	53
5.4	Camada de Monitoramento.....	54
5.4.1	Sensor .....	54

<b>6</b>	<b>Implantação .....</b>	<b>56</b>
6.1	Local e Período .....	56
6.2	Ambiente de Testes.....	56
<b>7</b>	<b>Resultados .....</b>	<b>59</b>
7.1	Resultados dos Testes de Detecção .....	59
7.2	Resultados dos Testes de Recuperação.....	63
<b>8</b>	<b>Trabalhos Futuros .....</b>	<b>65</b>
<b>9</b>	<b>Conclusões .....</b>	<b>66</b>
	<b>REFERÊNCIAS.....</b>	<b>67</b>

## 1 Introdução

Atualmente, a rede deixou de ser a atividade meio inaugurando um novo patamar, tornando-se a atividade fim. Impulsionada por necessidades de negócios, empresas e governos têm desenvolvido complexas e sofisticadas redes de informação, incorporando tecnologias diversas como sistemas de armazenamento de dados distribuídos, técnicas de criptografia, VOIP (Voice Over IP), acesso remoto e sem fio, e *web services*. Porém, junto com os benefícios gerados pela implantação desses serviços nossas redes acabam expostas e sua segurança em xeque. Novas ameaças são lançadas cada vez mais rápido e a ampla maioria tem como alvo vulnerabilidades na camada de aplicação. Para Uchôa (2009), o uso das redes se intensificou com a utilização de redes sociais e extranets, onde ameaças se aproveitam de portas e serviços de comunicação comumente permitidos para garantir a sua passagem através dos limites de segurança e para facilitar a operação no conjunto mais amplo de cenários de rede. O resultado tem sido uma constante erosão da eficácia dos *firewalls*. Segundo os trabalhos de Scambray, McClure e Kurtz (2001), para os *hackers*, esses caminhos tornam as redes mais vulneráveis do que nunca e - com relativamente pouca experiência - *hackers* têm impactado significativamente a segurança das redes das empresas e governos. *Cybercrime* também não é mais prerrogativa de *hackers* solitários ou atacantes aleatórios. Funcionários descontentes, empresas antiéticas e até mesmo organizações terroristas, olham para a Internet como um portal para coletar dados sensíveis e causar perturbações econômicas e políticas.

Com as redes mais vulneráveis e *hackers* equipados para causar estragos, não é surpresa que os ataques de rede estão em ascensão. O número de incidentes relatados está aumentando continuamente desde meados da década de 90 e governos, empresas e usuários domésticos continuam como alvos preferenciais para atividades maliciosas de hackers, ou seja, qualquer dispositivo conectado acaba em risco. Os administradores de sistema têm que garantir que seus sistemas estejam atualizados e configurados corretamente, e constantemente têm que tomar medidas para garantir computadores e redes mais seguras. No entanto, alguns hackers não são afetados por essas medidas, porque alguns deles ainda conseguem explorar com sucesso os mais variados sistemas. Sendo capazes de contornar o perímetro de

segurança de sistemas e encontrar caminhos através dos atuais sistemas de segurança.

Os efeitos de uma falha nestes sistemas podem ser catastróficos, trazendo consigo um significativo impacto financeiro, podendo custar de milhares a milhões de euros para reestabelecer a produtividade dos sistemas atingidos bem como a possibilidade de perda de vantagem competitiva frente à concorrência.

Outro aspecto que contribuiu para o agravamento das falhas de segurança foi o crescimento da inteligência dos softwares maliciosos que buscam explorar e expandir sua atuação através da infraestrutura da rede.

Os métodos de prevenção utilizados atualmente, muitas vezes sofrem com a necessidade de um banco de dados que tem de ser constantemente atualizado. Esses métodos não são suficientemente eficazes para detectar novos agentes maliciosos quando tentam infectar os sistemas de computador. Os sistemas atuais mostram deficiências na prevenção e recuperação de sistemas invadidos, levando pesquisadores e administradores a recorrer a maneiras alternativas para solução desses problemas.

A busca por soluções para essas deficiências leva diretamente aos trabalhos de Zuben e Attux (2007) relativos à Computação Natural, área da computação que utiliza a natureza como fonte de inspiração para o desenvolvimento de novas técnicas de solução de problemas, e traz consigo o melhor sistema de segurança que existe, o sistema imune humano.

O corpo humano nos mostra como um sistema de defesa robusto pode ser construído, o sistema imune humano tem a função de manter o corpo saudável e utiliza mecanismos avançados de detecção e eliminação de microrganismos infecciosos.

O sistema imune aprende a reconhecer novos patógenos que invadem o corpo e, em seguida, produz o tipo certo de resposta para combatê-los. Ele tem muitas propriedades que podem ser adaptadas para a concepção de sistemas artificiais na área de segurança da informação.

Este trabalho apresenta a implementação de um sistema de segurança de redes que é inspirado pelo sistema imune humano. Ele é capaz de realizar a detecção de uma tentativa de invasão nas cargas de pacotes que trafegam pela rede. Este sistema aplica vários algoritmos para a inspeção de pacotes, além de ferramentas para proteção e recuperação do núcleo do sistema no caso de uma eventual invasão.



Os resultados das experiências e o atual estágio de maturidade da solução são apresentados e analisados no decorrer desse trabalho.

## 2 Segurança Computacional

Se o impacto da Internet é sensível, isso ocorre tanto em seus aspectos positivos quanto negativos. Cada vez mais os crimes envolvendo o meio digital tornam-se temas de manchetes de notícias na mídia impressa ou televisiva.

Uchôa (2009) ressalta a importância da distinção, ao menos em termos práticos, dos conceitos de crime de computador e crime por computador. Essa distinção permite compreender melhor quais tipos de crime são cobertos pela lei e quais não o são.

Assim, crimes por computador são os crimes tradicionais cometidos por meios computacionais. Dessa maneira, por exemplo, tipificam-se o roubo ou o assassinato por computador. Tomando como exemplo, em um sistema comprometido a alteração da medicação de um paciente para doses fortes de substâncias a que ele tenha alergia.

É comum, portanto, a ocorrência de crimes tradicionais efetuados por computador, alguns inclusive sem que o autor desses crimes esteja atento ao fato de estar cometendo um crime já previsto na lei tradicional. O envio de determinados tipos de Spam, por exemplo, já está previsto na lei e pode render detenção de 3 meses a 1 ano ou multa, conforme o Art. 146 do Código Penal (BRASIL, 1940). Enviar e-mail com ameaça de agressão pode render pena de 1 a 6 meses de detenção ou multa, de acordo com o Art. 147. Assim, apenas modificou-se o meio, o crime continua tipificado. De forma semelhante, são tipificados crimes de invasão de privacidade, envio de vírus de computador, pedofilia ou montagem de sites com receitas de bombas ou similares.

Por outro lado, há crimes que só ocorrem no ambiente computacional, não existindo equivalente no ambiente não tecnológico: são os crimes de computador.

Nesse contexto, por exemplo, o Brasil mesmo que tardiamente reconheceu que alguns crimes podem ser cometidos no ambiente proporcionado pela rede. O Decreto-Lei 2.848 de 1940 (Código Penal Brasileiro) foi alterado pela Lei 12.737 de 2012 (Em vigor desde 03 de abril de 2013), passando a possuir tópicos sobre violação de equipamentos e sistemas - conectados ou não à internet - com intenção de destruir dados ou informações, ou instalar vulnerabilidades.

Também foram instituídas penas, que nos casos menos graves, como invasão de dispositivo informático, varia de três meses a um ano de prisão e multa.

Para casos mais sérios, como invasão para obter comunicações eletrônicas privadas, segredos comerciais ou industriais, informações sigilosas, as penas podem render de três meses a dois anos de prisão, além da multa.

Cabe ressaltar que alguns países já possuem uma legislação mais forte a respeito de crimes de computador, como é o caso dos EUA e da China. No Brasil, como citou Uchoa (2009) outrora paraíso dos invasores, a situação mudou para outro patamar, principalmente com a criação da chamada Lei Carolina Dieckman<sup>1</sup>, além da discussão de outros projetos de lei sobre o assunto. Um desses projetos exigiria o cadastro dos usuários de um provedor<sup>2</sup>. Outro projeto, ironizado em diversos *blogs*, exigiria a inserção da advertência nos monitores dos computadores, de que o uso indevido do computador pode gerar infrações que sujeitam o usuário à responsabilização administrativa, penal e civil<sup>3</sup>.

A maior polêmica atual, entretanto, recai sobre o substitutivo do Senador Eduardo Azeredo, sobre crimes contra a segurança dos sistemas informatizados<sup>4</sup>. Parte da polêmica recai sobre os implicativos dessa lei que, na opinião de vários especialistas<sup>5</sup>, fere liberdades individuais e coloca várias atividades atualmente legítimas num “limbo” legal. Além disso, acrescenta novas responsabilidades para provedores, o que inviabilizaria várias atividades de inclusão digital. Longe da polêmica sobre os projetos de lei, como informado anteriormente, ainda existem várias lacunas no que diz respeito ao direito digital.

Diante dessa situação, usuários mal-intencionados aproveitam para disparar diversos tipos de ataques, sem se preocuparem de imediato com punições formais.

Além disso, infelizmente a legislação nacional ainda não está evoluída o suficiente para tratar com propriedade casos de *cybercrimes*, principalmente por essas tecnologias ainda serem muito novas para o cenário jurídico brasileiro, e, ainda que seja razoavelmente fácil rastrear vários desses meliantes cibernéticos, eles continuam

---

<sup>1</sup> Lei Nº 12.737, de 30 de novembro de 2012

<sup>2</sup> Projeto do Senador Gerson Camata, exigindo dos provedores nome completo e número do documento de identidade do usuário, bem como identificação do terminal utilizado, data e hora de início e término de sua utilização:

[http://www.senado.gov.br/atividade/materia/detalhes.asp?p\\_cod\\_mate=86846](http://www.senado.gov.br/atividade/materia/detalhes.asp?p_cod_mate=86846)

<sup>3</sup> Projeto do Deputado Federal Carlos Bezerra

[http://www.camara.gov.br/internet/sileg/Prop\\_Detalhe.asp?id=393544](http://www.camara.gov.br/internet/sileg/Prop_Detalhe.asp?id=393544).

<sup>4</sup> Substitutivo do Senador Eduardo Azeredo:

<http://webthes.senado.gov.br/sil/Comissoes/Permanentes/CCJ/Pareceres/PLC2008061889.rtf>

<sup>5</sup> Uma avaliação mais crítica desse substitutivo pode ser verificada em várias mensagens do blogueiro Sérgio Amadeu no blog Trezentos: <http://www.trezentos.blog.br>.

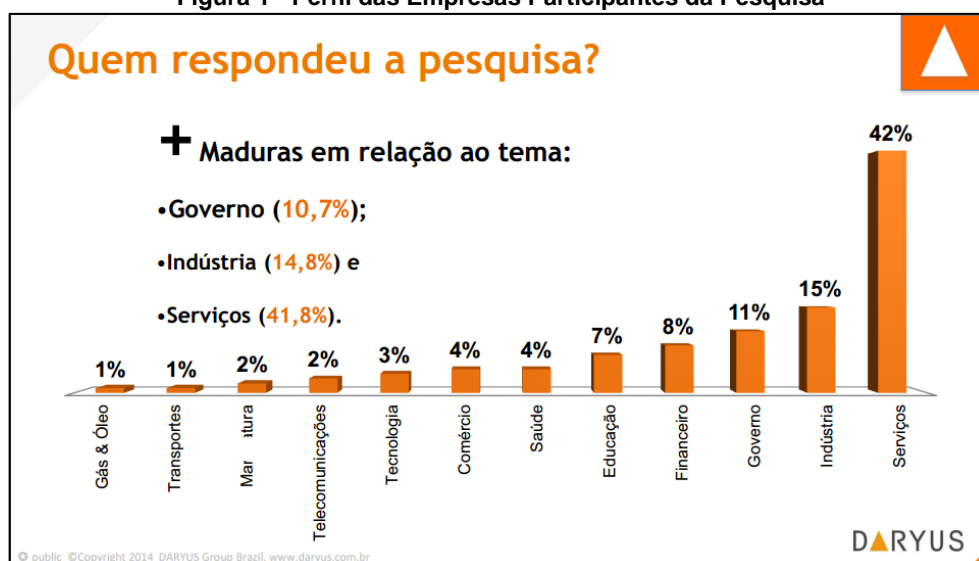
impunes graças ao pouco caso que as autoridades competentes fazem frente aos delitos virtuais.

## 2.1 Necessidade de Segurança

Segundo Nakamura e Geus (2003), nas décadas de 70 e 80 a segurança nas organizações tinha como foco o sigilo dos dados. Com o surgimento do ambiente de rede nas décadas de 80 e 90, a visão da área de segurança mudou para a integridade, com a proteção sendo feita tendo-se em mente a informação. Já na década de 90 a informática passou a ser essencial para os negócios e com o crescimento das redes o enfoque passou a ser a disponibilidade, e a proteção passou a ser sobre o conhecimento.

A falta de planejamento em segurança pode parecer uma boa situação, pois tudo funciona adequadamente, até que surgem os problemas que podem resultar em custos elevadíssimos em sua resolução, uma visão geral do perfil das empresas em relação à segurança da informação pode ser observada na figura 1. O importante não é só funcionar, mas funcionar bem e com segurança. Muitas empresas ainda deixam a segurança em segundo plano, dando-lhe a devida importância somente quando ela se torna extremamente necessária.

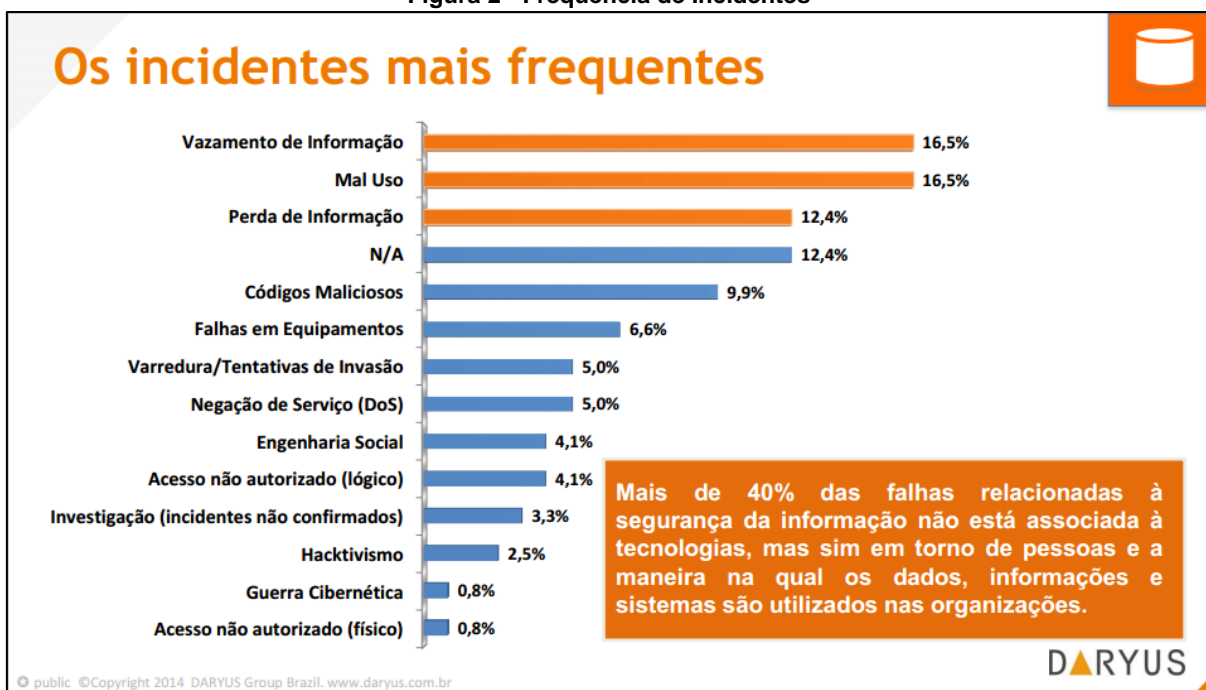
Figura 1 - Perfil das Empresas Participantes da Pesquisa



Fonte: Daryus Strategic Risk Consulting; IT Mídia; Exin Brasil (2014)

De acordo com a Pesquisa Nacional de Segurança da Informação executada pela Daryus Strategic Risk Consulting, com apoio da IT Media e Exin, das empresas entrevistadas apenas 36,36% possui um processo formal para gestão de incidentes de Segurança da Informação. Apesar disso, a frequência na qual os incidentes são relatados é baixa, sendo de 56,20%, e aproximadamente 50% dos entrevistados considera o impacto operacional dos incidentes, embora 54,55% não saiba informar como é considerado o impacto financeiro. Ainda segundo esse relatório, o maior percentual de incidentes fica por conta do vazamento de informações com 16,5% com pode ser visto na figura 1-2. E tudo isso através de redes, onde *firewalls* estavam instalados em mais de 90% dos casos como pode ser observado na figura 2.

Figura 2 - Frequência de Incidentes



Fonte: Daryus Strategic Risk Consulting; IT Mídia; Exin Brasil (2014)

Essa situação contribui para uma redução no nível de confiança dos entrevistados, tanto em leis e regulamentos realmente capazes de prevenir ou proteger contra incidentes, quanto a própria área de segurança ao não conseguir tratar ou evitar ataques cibernéticos de origem interna ou externa, como pode ser observado na figura 3.

Figura 3 - Avaliação de Confiança dos Entrevistados



Fonte: Daryus Strategic Risk Consulting; IT Mídia; Exin Brasil (2014)

Este trabalho explora uma combinação de fatores que estão expondo uma variedade de deficiências com desenhos de *firewall* e IDS (Intrusion Detection System) / IPS (Intrusion Prevention System) atuais. Esses problemas e falhas, então, serão usados para estabelecer os critérios que definem a solução ideal: um sistema de segurança denominado por “Firewall de Próxima Geração”, dispositivo que incorpora inspeção de aplicativos e recuperação automática do dispositivo.

A possibilidade de diminuir o tempo gasto para serem diagnosticados e reparados problemas de sistemas de *firewall* e suas configurações, foi a principal motivação para este trabalho, além de utilizar uma abordagem diferente, mais voltada para a computação natural. Hoje o cenário no mercado aponta que o custo de recursos humanos supera sensivelmente os custos de recursos de tecnologia, levando à diminuição no custo de produção de CPU (Central Process Unit), dispositivos de rede e armazenamento, e mudando o ambiente de tecnologia no mundo. Isso denota a necessidade de desenvolvimento de sistemas para automatizar a análise e recuperação, tendo como principal objetivo minimizar o esforço humano para este tipo de tarefa.

## **2.2 O Processo de Segurança**

De acordo com Bejtlich (2004), segurança é um processo e não um estado e envolve quatro passos, sendo estes o de avaliação, o de proteção, o de detecção e o de resposta ao ocorrido, passos esses detalhados abaixo.

### **2.2.1 Avaliação**

A avaliação de acordo com Bejtlich (2004) é o passo inicial e o mais importante por preparar para os outros passos do processo. Envolve determinar medidas que possam garantir a probabilidade de sucesso ao defender uma empresa ou instituição. Nesta etapa, devem-se definir uma política de segurança e os serviços que estarão disponíveis em uma empresa. Este passo é muito importante, pois, assim, pode-se definir como será o tráfego da instituição, tornando-se então uma tarefa mais fácil a detecção de tráfego suspeito na rede. Se a política definida for realmente rigorosa, qualquer outro tráfego não conhecido é um incidente na rede, sendo ou não um ataque.

Para Fraser (1997), os incidentes originam também de tentativas de usuários internos de escaparem das regras definidas pela instituição para acesso ou de utilização da rede. Até que ponto isso pode ser considerado como invasão ou somente violação da política é avaliação da equipe de gerência interna da rede, baseando-se na política de segurança interna adotada.

Esta definição de tráfego é muito útil e é mais visível através de um exemplo: uma empresa, ao definir que fornecerá somente o serviço de FTP aos seus clientes pode considerar que qualquer outro tráfego não relacionado, ou tentativa de acesso a uma porta diferente do serviço seja algo suspeito e que necessite ser monitorado e avaliado para verificar a possível tentativa ou real invasão.

### **2.2.2 Proteção**

De acordo com Bejtlich (2004), a proteção é a aplicação das medidas defensivas para reduzir a probabilidade de comprometimento. Em sistemas IDS ou de

monitoramento, este passo não é incluído, pois o objetivo é detectar a invasão caso a prevenção realmente falhe.

Os sistemas responsáveis pelo passo de proteção são os IPS, que podem, até mesmo, controlar um dispositivo ou um *firewall*. Mesmo que sistemas IDS não sejam uma forma de prevenção, essa etapa ajuda na eficiência de tais sistemas, tornando-os mais efetivos.

Um problema crítico na auditoria de tráfego de uma rede é a quantidade de tráfego que deve ser monitorada, ao reduzir a quantidade de tráfego, ou as formas de acesso, tem-se um tráfego menor para monitorar e analisar. Afinal, o tráfego rejeitado não pode ferir uma empresa, e sim o tráfego aceito. Tem-se somente uma exceção, referente aos ataques de negação de serviço.

Ainda segundo Bejtlich (2004), alguns itens muito úteis na proteção de um sistema são:

- Controle de Acesso: um tráfego desconhecido na rede, que foge da política do controle de acesso, pode ser uma má configuração, ao invés de uma ação maliciosa. Quando uma ferramenta de monitoramento trabalha em conjunção com uma política de segurança bem definida e com um controle de acesso reforçado e apropriado, ele oferece a forma mais pura de auditoria da rede. Desvios da política são fáceis de identificar e resolver. Com políticas apropriadas e bem implementadas, os invasores têm poucos vetores de ataque e os responsáveis pelo monitoramento e segurança da rede podem ficar intencionalmente analisando somente o tráfego em porções limitadas;
- Afinação de Tráfego: muitos IDS gastam maior parte do seu tempo analisando pacotes replicados. A afinação do tráfego que garante um conteúdo com os passos de invasão é importante para agilizar e melhorar o desempenho de sistemas de identificação de intrusão;
- Proxies: Esses sistemas, inseridos entre os cliente e servidores, são importantes por razões de segurança, monitoramento, ou desempenho, já que um cliente não tem como se conectar ao servidor sem antes se conectar ao proxy.



### 2.2.3 Detecção

Conforme Fraser (1997), a detecção é o processo de identificar os intrusos, coletando, identificando, validando e escalonando eventos suspeitos. As invasões são violações da política ou acidentes na segurança dos computadores de uma rede.

A detecção requer quatro passos:

- Coletar: envolve acessar o tráfego com intenção de inspecionar e armazenar informações úteis. Dentre os erros mais comuns de coleta, podem ser citados a má configuração ou má aplicação de filtros ou regras para evitar eventos indesejáveis, a implantação de enlaces com a capacidade superior à capacidade do sensor de detecção e a combinação de equipamentos sem o conhecimento básico de sua tecnologia;
- Identificar: uma vez que todo o tráfego é transformado em um tráfego observável, é hora de obter informações dele, reconhecendo pacotes que não são comuns à rede. O tráfego observado pode ser caracterizado em:
  - Tráfego normal: é tudo que é esperado na rede de uma empresa;
  - Tráfego suspeito: é o tráfego que traz características estranhas, incomuns, mas não causam danos aos bens da empresa;
  - Tráfego malicioso: é qualquer informação que possa trazer um impacto negativo a postura de segurança da empresa, se enquadrando os ataques de todos os tipos realizados com sucesso à empresa;
- Validar: atribuir aos eventos uma categoria de incidente preliminar. Essa categoria de incidentes classifica os eventos como advertências ou, às vezes, simplesmente como indicadores, que são evidências do ataque ou, pelo menos, algo que necessite de uma investigação adicional, sempre alertando aos analistas que houve algo malicioso ou uma falha de segurança. Os eventos podem ser assimilados em uma das categorias:

- Acesso sem autorização como administrador: ocorre quando uma ferramenta ou alguém adquire sem autorização o controle de administração de outra parte do sistema;
  - Acesso de usuário sem autorização: ocorre quando uma ferramenta ou alguém sem autorização ganha controle de uma conta de usuário que não seja de administrador;
  - Tentativas de acesso sem autorização: ocorre quando uma ferramenta ou alguém sem autorização tenta obter acesso de administrador ou usuário do computador;
  - Ataque de negação de serviço com sucesso: ocorre quando o adversário (pessoa que ainda não invadiu o sistema) consegue ter ações prejudiciais aos serviços e processos de uma máquina alvo ou de uma rede. Esses ataques podem consumir ciclos de CPU, banda, espaço em disco, tempo de usuário e vários outros recursos;
  - Prática de segurança ruim ou violação da política: ocorre quando uma operação de monitoramento detecta uma condição que expõe a todos, informações que permitem explorar os clientes, ou quando o cliente infringe a política, utilizando, por exemplo, serviços sem autorização;
  - Reconhecimento/provas/explorações: ocorre quando um adversário tenta aprender sobre uma rede ou um sistema alvo, com a intenção presumida de mais tarde comprometer este sistema ou rede;
  - Infecção por vírus: ocorre quando um sistema é infectado por um vírus ou worm. Um vírus depende da interação do humano para propagar-se e/ou pode se acoplar em um arquivo hospedeiro, como em um e-mail, um documento ou uma página da Internet. Já os worms são capazes de propagar por si só sem a capacidade de interação de homens ou de arquivos hospedeiros.
- 
- Escalonamento: é o processo de encaminhar os eventos obtidos às pessoas que tomarão decisões sobre os mesmos, sendo estes os clientes ou os supervisores de uma organização. Mesmo nessa fase, nem todas as indicações e advertências devem ser encaminhadas aos clientes, pois podem ser casos em que o incidente ainda não chegou a acontecer.

## 2.2.4 Resposta

Ainda de acordo com Fraser (1997), a resposta é o processo de validar os frutos da detecção e tomar medidas para remediar as invasões. Existem dois processos de resposta:

- **Contenção do incidente em curto prazo:** consiste em impedir que o incidente continue na rede. Geralmente os passos tomados são impedir fisicamente o acesso à máquina alvo, ou instalar uma nova regra no roteador de filtro ou no firewall para proibir o tráfego de entrar ou sair do alvo;
- **Monitoramento de emergência:** consiste em capturar todos os dados sobre o IP (Internet Protocol) invasor. Quando todo o tráfego já é armazenado, isso não é necessário.

### **3 Sistemas de Detecção de Intrusão**

Conforme Wang (2009), existem várias tecnologias que protegem o perímetro das redes de computadores contra invasores e ameaças externas. Porém, esses sistemas não podem parar invasores que obtêm acesso autenticado e usufruem do sistema como se fossem usuários legítimos. Devido a isso, torna-se importante monitorar as informações contidas nos pacotes que atravessam o firewall para analisar como os usuários, legítimos ou não, utilizam seus computadores, monitorando assim suas atividades e permitindo a detecção das ações de invasão.

De acordo com Costa (2007), os primeiros conceitos sobre sistemas capazes de detectar intrusões foram propostos no relatório técnico “Computer Security Threat Monitoring and Surveillance” escrito em 1980 por James P. Anderson. Conforme Wang (2009), os sistemas detectores de intrusão foram iniciados com Dorothy Denning e Peter Neumann em meados de 1980. Na época, eles observaram que os invasores agiam de forma diferente dos usuários normais do sistema e que as diferenças poderiam ser medidas, tornando possível a análise quantitativa do ataque. Para identificar os eventos anormais nos sistemas, torna-se necessária a criação de ferramentas automatizadas que se baseiam nas operações ocorridas nos sistemas de administração, nos protocolos de rede, nas estatísticas computacionais e na mineração de dados.

Essas ferramentas denominadas IDS, são sistemas automatizados que procuram por indícios de intrusão direta ou indireta, incluindo invasões que já aconteceram ou que estão atualmente em curso, e que notificam os administradores para que possam agir apropriadamente. Ainda de acordo com Wang (2009), essas ferramentas automatizadas para detectar invasões tem o objetivo de identificar as atividades de invasão que ocorreram ou que estejam acontecendo em uma rede. Quanto mais rápida for a detecção, menores serão os prejuízos causados pela invasão.

#### **3.1 Processo de Detecção de Intrusão**

Conforme Nakamura e Geus (2003) a detecção de intrusão é uma evolução da tradicional prática de auditoria de sistemas, onde os registros de auditoria, gerados

pelo sistema operacional e outros mecanismos de *log*, eram revisados manualmente de tempos em tempos. À medida que os computadores ficaram mais rápidos, complexos e numerosos, os registros de auditoria também aumentaram seu tamanho e complexidade, exigindo um processo automatizado de revisão.

É uma técnica relativamente recente, tendo iniciado seu desenvolvimento a partir de 1980. Entretanto, alguns conceitos e técnicas fundamentais surgiram ao longo dos últimos 20 anos de evolução na área.

Em termos gerais, os sistemas de detecção de intrusão consistem de três componentes fundamentais:

- Uma fonte de informações, que provê um fluxo de registros de auditoria, também chamados de registros de eventos, utilizados para determinar a ocorrência de uma intrusão;
- Um mecanismo de análise, que busca por sinais de intrusões no fluxo de eventos derivado da fonte de informações;
- Um componente de resposta, que gera reações baseadas na saída do mecanismo de análise.

Segundo Uchôa (2005), em um sistema medianamente seguro, uma invasão irá exigir esforço e tempo, de forma que, com um monitoramento eficiente, a invasão pode ser bloqueada em seu início.

### 3.1.1 Estratégias de Sistemas de Detecção de Intrusão

Geralmente, há duas principais estratégias para a implementação de um IDS:

- **Detecção de mau-uso** baseia-se na detecção de padrões de ataque já conhecidos. A informação é analisada por comparação com as assinaturas na base de dados. Um IDS que utiliza detecção de mau uso protege o sistema assim que instalado com uma taxa muito baixa de falsos positivos. Se o IDS gera um alarme é diretamente encaminhado para um tipo específico de atividade da rede. Mas esses sistemas não detectam novos tipos de ataques

que não estão incluídos no banco de dados. Portanto, o banco de dados tem que ser mantido atualizado.

- **Detecção de anomalias** baseia-se na detecção de desvios do comportamento do sistema, que é definido por um perfil de comportamento normal. Este perfil é aprendido pelo IDS durante um longo período de tempo de observação da rede.
  - Esta abordagem tem várias vantagens. Porque o IDS é baseado em um perfil aprendido do comportamento normal, os atacantes não sabem se sua atividade vai levar a um alarme do IDS. A detecção de intrusões não é baseada em atividades maliciosas conhecidas como em sistemas de detecção de mau-uso. Permite a detecção de anomalias do IDS para a detecção de novos ataques que provavelmente foram utilizados pela primeira vez.
  - Um IDS usando a detecção de anomalias também tem algumas desvantagens. O perfil para o comportamento normal deve ser definido e, portanto, o IDS não protege o sistema durante um período de tempo após a instalação. A definição do que é "normal" deve ser configurado no perfil. Sistemas de detecção de anomalias têm uma elevada taxa de falsos positivos e um alarme gerado é mais difícil de se associar com uma atividade específica de rede. Além disso, se um ataque específico não viola com o perfil de comportamento definido que passa despercebida.

Portanto, sistemas de detecção de anomalia não são frequentemente implementados em IDS. Na verdade, muita pesquisa é feita sobre a estratégia de detecção de anomalias. A combinação de ambas as estratégias para a detecção de intrusos irá utilizar os benefícios de ambas.

### 3.2 Tipos de Ataques

Os principais tipos de ataques, de acordo com pesquisas realizadas por Nakamura e Geus (2003), são os ataques direcionados e os oportunistas. Os direcionados são menos comuns, porém, são os mais perigosos, pois envolvem pessoas com objetivos formulados que podem ter estudado a empresa antes de iniciar o ataque. Os ataques oportunistas são mais comuns e são realizados de maneira aleatória. Diversas técnicas e ferramentas podem ser utilizadas na obtenção das informações que podem levar a um ataque de sucesso. Estas mesmas técnicas e ferramentas podem ser utilizadas também pelo próprio administrador do sistema para identificar e corrigir vulnerabilidades. Ainda de acordo com Nakamura e Geus (2003), o atacante pode utilizar engenharia social, ataques físicos, informações livres, *packet sniffing*<sup>6</sup>, *port scanning*<sup>7</sup>, *scanning de vulnerabilidades*<sup>8</sup> e *firewalking*<sup>9</sup>. O IP spoofing, técnica na qual o atacante pode disfarçar seu endereço IP (Internet Protocol) dificultando a identificação da origem do ataque, é utilizado como técnica auxiliar para os outros métodos de obtenção de informações.

De acordo com Uchôa (2005), os principais tipos de ataques são:

- **Footprinting:** Consiste em coletar informações sobre um sistema alvo. Esse processo é feito por vias tradicionais, tais como a leitura das páginas do site para obter dados.
- **Spoofing:** Consiste em fazer uma máquina se passar por outra. Geralmente é feita a tentativa de bloquear o envio de pacotes de dados de uma máquina, tentando se passar por ela.
- **Código Malicioso:** Consiste em softwares com códigos não autorizados que efetuam ações desconhecidas e não desejadas pelo usuário;
- **Exploits:** São programas criados para explorar falhas, geralmente provenientes principalmente de bugs. Entre as falhas mais exploradas, encontram-se buffer overflow, que consiste em estourar o buffer de entrada de

---

<sup>6</sup> *Packet sniffing:* ataque no qual um intruso pode ler diretamente as informações transmitidas e o conteúdo da base de dados.

<sup>7</sup> *Port scanning:* aplicativo que faz uma varredura nas portas do equipamento informando seu estado.

<sup>8</sup> *Scanning de vulnerabilidades:* aplicativo que faz uma varredura no equipamento procurando por vulnerabilidades que possibilitem um ataque.

<sup>9</sup> *Firewalking:* método de envio de pacotes ao *firewall* com o objetivo de descobrir vulnerabilidades.

um servidor, forçando-o a estourar sua memória, devolvendo um *shell* para o invasor.

- **Ataques de Senhas:** Consistem em tentar descobrir a senha de um ou mais usuários por força bruta ou usando técnicas para tornar possível a descoberta.
- **Buffer-overflow (Estouro de Pilha):** A "pilha" fornece um espaço na memória onde são armazenados diversos dados. A ideia do ataque é "estourar" a pilha para que os dados que "vazarem" sejam executados como código pelo processador.

Engenharia social também é um tipo de ataque muito utilizado e difícil de ser combatido. Através da engenharia social, o atacante pode obter informações privilegiadas enganando os usuários, utilizando identificações falsas ou conquistando a confiança da vítima. Para isso diversos meios podem ser utilizados, entre eles, o telefone, e-mail ou contato direto.

Já para Scambray, McClure e Kurtz (2001), outros ataques comuns são:

- ataques de força bruta;
- ataques dirigidos por dados;
- ataques de validação de entradas;
- ataques de telnet reverso e canais de retorno;
- ataques a serviços como TFTP (Trivial File Transfer Protocol), NFS (Network File System), sendmail, etc;
- ataques de descritor de arquivo;
- ataques de condição de corrida;
- ataques a bibliotecas compartilhadas;
- ataques a arquivos do cerne;
- ataques de falhas de kernel ;
- ataques a sistema configurados incorretamente;
- ataques de negação de serviços;
- e ataques de negação de serviços distribuídos.

Os ataques de negação de serviços DoS (Denial of Service) permitem explorar os recursos de um servidor de maneira agressiva, de modo que usuários legítimos



fiquem impossibilitados de utilizá-los. Já os ataques coordenados são os mais evoluídos, também conhecidos como ataques de negação de serviços distribuídos DDoS (Distributed Denial of Service). Este ataque faz com que diversos hosts distribuídos sejam atacados e coordenados para realização de ataques simultâneos aos alvos. Isso resulta em um ataque extremamente eficiente, no qual a vítima pode ficar praticamente indefesa, sem conseguir descobrir a origem dos ataques, já que estes procedem de hosts intermediários controlados pelo atacante. Os ataques podem ser classificados como ataques de acesso remoto e local.

Acesso remoto é definido como ganhar acesso via rede ou outro canal de comunicação. De acordo com Scambray, McClure e Kurtz (2001), acesso local, conhecido também como ataque de escalação de privilégio, é definido como ter um *login* ou *shell* de comando real no sistema. Uma vez que o atacante consegue acesso local ao sistema, este poderá coletar informações e utilizar a estação invadida como ponto de partida para ataques adicionais. Um sistema invadido pode não ser mais confiável se não houver ferramentas que permitam identificar quais as ações executadas pelo atacante no sistema durante a invasão.

## 4 Firewalls

Conforme Abdel-Aziz (2009) *firewall* é uma tecnologia antiga, do final da década de 80, o que explica suas limitações. Nos badalados anos 90 a tecnologia evoluiu e com ela a chegada do conceito de *stateful inspection*, que vinha a se tornar um padrão de mercado. Com o passar do tempo tornou-se item obrigatório, uma commodity em todas as empresas e acabou virando para leigos sinônimo de proteção. Toda essa evolução não resolveu sua característica e limitação conceitual: tomar decisões a partir de portas e protocolos. Ao mesmo tempo as técnicas de invasão utilizadas por hackers evoluíram gerando outros sistemas de segurança complementares os conhecidos “*helpers*”. O mais importante deles o IPS, é item indispensável no ambiente corporativo.

### 4.1 Testes de Firewalls

Conforme Cheswick, Belovin e Rubin (2005), testar um *firewall* é fundamentalmente diferente de testar qualquer outro sistema de hardware ou software. Por exemplo, em testes de *software* em geral, duas técnicas comuns são: o Teste da Caixa Preta e o Teste da Caixa Branca. Testes de Caixa Preta não presumem conhecimento algum sobre os detalhes internos do sistema e testam seu comportamento com relação à especificação e muitas entradas diferentes. O segundo utiliza conhecimento do código para testar como o estado interno responde a várias entradas. Estas técnicas, com algumas variações também podem ser utilizados ao testar *firewalls*.

### 4.2 Firewalls de Próxima Geração

Segundo Young et al. (2008), NGFW (Next Generation Firewall) usam inspeção profunda de pacotes DPI (Deep Packet Inspection) como tecnologia de núcleo. É importante notar que ainda não há uma definição constante e detalhada de um NGFW, embora tenham sido feitas outras tentativas para definir o que é um NGFW, uma dessas tentativas foi feita por Young e Pescatore para o Gartner Group

em 2008. No contexto deste trabalho, firewall de próxima geração é um termo usado para representar a nova geração de *firewalls stateful* que integram prevenção de intrusão, filtro de *malware*, bem como outras funções de segurança para permitir um controle mais avançado de fluxo de dados. Como indicado na Tabela 1, estes novos *firewalls* olham profundamente o *payload* do pacote antes de tomar uma decisão sobre permitir ou negar o fluxo de tráfego.

**Tabela 1 - Tipos de Firewall**

Tipo de Firewall	Packet-Filter	Stateful Packet Inspection (SPI)	Application Proxy	Deep Packet Inspection (DPI)
<b>Camada OSI</b>	Camada de Transporte	Camada de Transporte	Camada de Aplicação	Camada de Aplicação
<b>Geração</b>	1ª Geração	2ª Geração	3ª Geração	4ª Geração
<b>Características Principais</b>	Examina os endereços de origem e destino, portas e serviços solicitados. Os roteadores que usam ACLs definem o acesso aceitável para uma rede.	Examina o estado e contexto de pacotes. Mantém o controle de cada conversa usando uma tabela de estado.	Atua como um intermediário entre sistemas de comunicação por quebrar a sessão e restabelece uma nova sessão para cada sistema. Proxy diferente necessário para cada serviço permitido.	Olha profundamente nos pacotes e toma decisões de controle de acesso granular baseado em cabeçalho do pacote e carga útil. Excelente em gerenciar as ameaças de aplicativos e dados trafegados. Incorpora detecção de intrusões e recursos de tecnologia de prevenção.
<b>Requisito de Recursos</b>	Baixo	Baixo-Médio	Alto	Médio
<b>Desenho do Firewall</b>	Desenho Inicial	Desenho considerado a evolução do Packet-Filter	Desenho Alternativo	Desenho considerado a evolução do Stateful Packet Inspection

Fonte: Abdel-Aziz (2009)

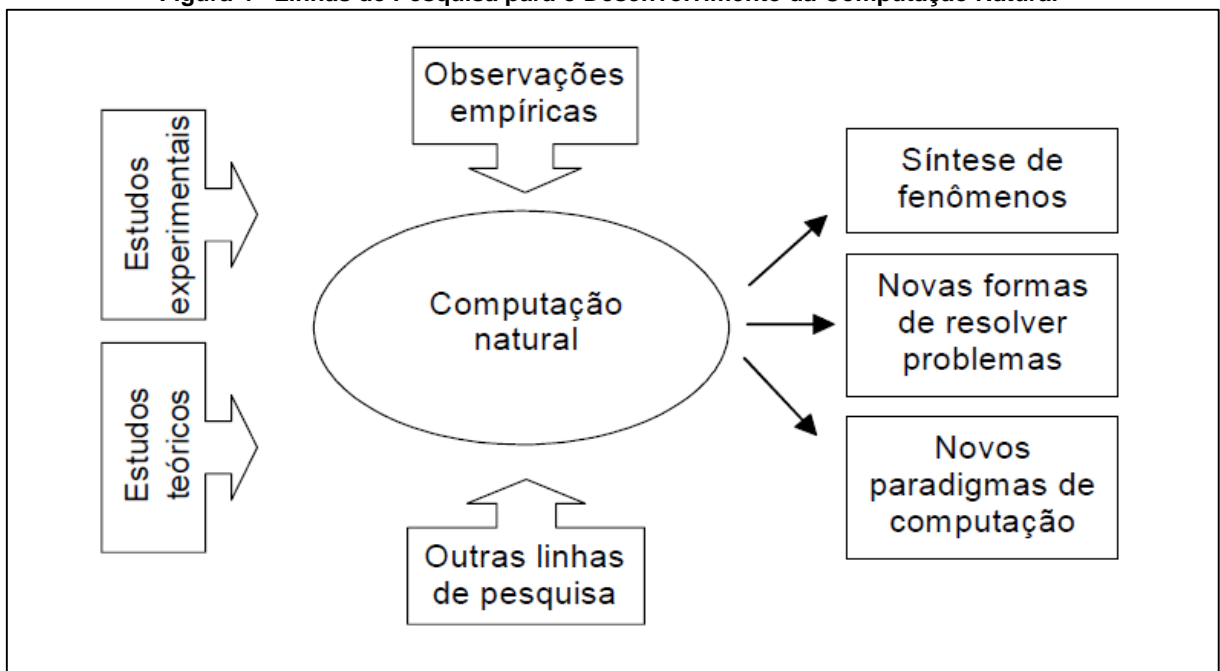
Essencialmente, eles estão realizando o papel principal de um firewall que é o de "controlar o fluxo de dados", mas de uma maneira muito mais detalhada do que era possível com *firewalls stateful*. Ainda de acordo com Young (2008), uma vez que estes *firewalls* realizam inspeção em nível de aplicação e prevenção de intrusões, e estão ganhando impulso, o Gartner prevê que o mercado NGFW vai ultrapassar o mercado de IPS/IDS no ambiente empresarial.

### 4.3 A Biologia Como Inspiração

Segundo De Castro et al. (2004), o conjunto de sistemas que são desenvolvidos com inspiração ou utilização de um ou mais mecanismos naturais ou biológicos de processamento de informações constituem a linha de pesquisa da Computação Natural, cujo esquema é ilustrado na Figura 5. Existem três subdivisões desta linha de pesquisa:

- Computação Bioinspirada - novas formas de solução de problemas;
- Simulação de eventos naturais - síntese da "vida";
- Computação com Mecanismos Naturais - novos paradigmas de computação.

Figura 4 - Linhas de Pesquisa para o Desenvolvimento da Computação Natural



Fonte: Zuben e Attux (2007)

#### 4.3.1 O Sistema Imune Humano

O sistema imune humano é um sistema de defesa muito complexo, com a tarefa de proteger o corpo contra vários tipos de ameaças. Conforme De Paula (2004) a defesa oferecida pelo sistema imunológico humano é resultado da ação de dois subsistemas: o sistema imunológico inato e o sistema imunológico adaptativo. Eles

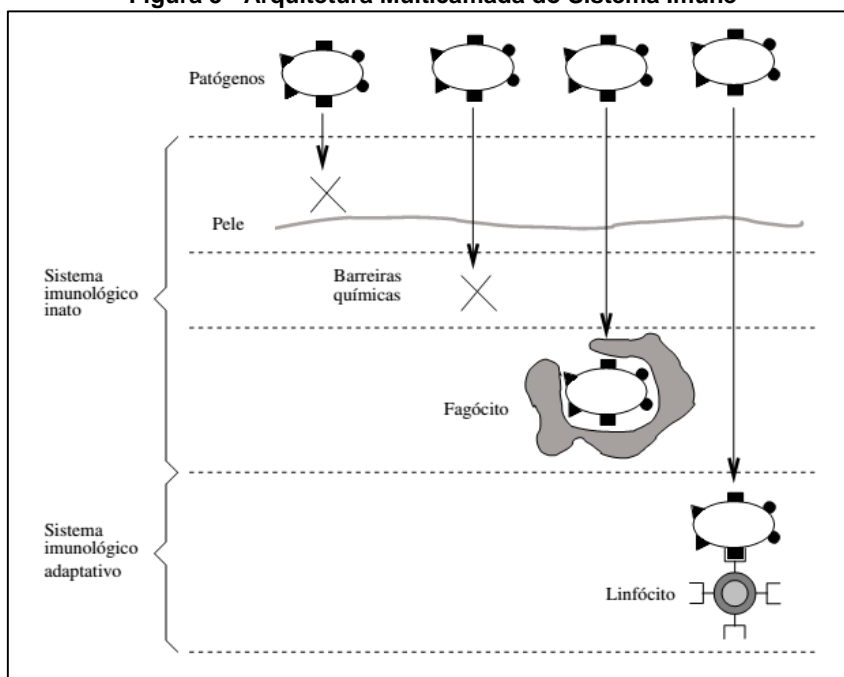
compõem o sistema imunológico global, possibilitando respostas contra agentes infecciosos conhecidos ou até então desconhecidos.

O sistema imunológico inato é caracterizado por sua natureza congênita e por sua capacidade limitada de diferenciar um agente patogênico de outro, reagindo de maneira semelhante contra a maioria dos agentes infecciosos. O sistema inato constitui a primeira linha de defesa contra a ação de micróbios, e sua resposta, por não ser específica para um determinado micróbio, é, na maioria das vezes, insuficiente. Seus principais componentes são as barreiras físicas e químicas, e as células conhecidas como fagócitos, responsáveis pela ingestão e digestão de uma grande variedade de materiais, tais como outras células, micróbios e partículas estranhas.

Em contraste com o sistema inato, o sistema imunológico adaptativo é capaz de identificar especificamente um determinado agente patogênico, permitindo uma resposta bastante eficiente. Além disso, ele é capaz de armazenar informações a respeito de um agente infeccioso, de maneira a responder mais vigorosamente em novas exposições a esse agente. Os componentes do sistema adaptativo são os linfócitos (linfócitos T e linfócitos B) e seus produtos, como os anticorpos e as linfocinas.

Os mecanismos de ambos os sistemas, inato e adaptativo, constituem um sistema de defesa multicamada em que um grande número de células e moléculas agem cooperativamente. A Figura 6 ilustra a arquitetura multicamada do sistema imune, com mecanismos de defesa em vários níveis.

**Figura 5 - Arquitetura Multicamada do Sistema Imune**



Fonte: De Paula (2004)

Durante a vida o corpo está permanentemente exposto a substâncias nocivas (*patógenos*<sup>10</sup>) como ilustra a figura 7. Se os patógenos não são detectados, levam a uma rápida deterioração da saúde do corpo.

**Figura 6 - Agentes Causadores de Doenças**



Fonte: Autor Desconhecido - Imagem disponível em: <<http://www.quora.com/What-causes-Auto-Immune-diseases>>. Acesso em 20 abr. 2015.

<sup>10</sup> Patógenos são microorganismos nocivos como bactérias ou vírus.

De acordo com Unterleitner (2008), o sistema imune tem a tarefa de combater esses patógenos para manter o corpo saudável, além de executar duas tarefas elementares.

A primeira é detectar agentes patogênicos, e assim que são detectados tais agentes, sua segunda tarefa é eliminar ou neutralizar estes agentes. Portanto, o sistema imune desenvolve uma grande quantidade de células e moléculas que são capazes de reconhecer e eliminar agentes patogênicos. As células interagem com o meio ambiente local, através de sinais químicos. Uma célula que se liga a proteínas estranhas é um evento de detecção de sinalização. Estas células, então, atraem outras células para assistência e, por consequência, inicia-se uma série de reações.

O sistema imune reage com uma resposta imune primária que conduz à destruição ou à neutralização dos agentes patogênicos. Durante a resposta primária, o sistema imune produz uma variedade de células que são capazes de detectar tipos específicos de agentes patogênicos com uma precisão cada vez maior. A adaptação dessas células leva algum tempo, mas, logo que o sistema imune reconhece os novos agentes eles são mortos. Após a eliminação bem-sucedida dos agentes patogênicos o sistema imune memoriza uma pequena fração das células adaptadas (memória imunológica). Estas células específicas permitem que o sistema imune responda muito mais rápido e de forma mais eficiente durante ocorrências futuras. Este mecanismo é denominado resposta imune secundária. Ele baseia-se na memória imunológica do sistema imune. Esta resposta secundária é muitas vezes rápida o suficiente para combater patógenos antes que eles cheguem a um número suficiente para prejudicar o corpo. A memória imunológica protege o corpo contra doenças infecciosas que o sistema imune tenha encontrado antes.

As células do sistema imune podem detectar diferenças sutis entre agentes patogênicos químicos. Elas podem distinguir entre organismos patogênicos e elementos do corpo. Assim, o sistema imune é capaz de distinguir entre *próprio* e *não-próprio*. O conjunto de *próprio* consiste em substâncias inofensivas que incluem elementos do corpo. Considerando que, o conjunto *não-próprio* contém substâncias nocivas.

### 4.3.2 Sistemas Imune Artificiais

Uma área em inteligência computacional com profunda inspiração em imunologia são os SIAs (Sistemas Imunes Artificiais), um tipo de Algoritmos Bioinspirados. Inclusive o termo “Algoritmos Imunoinspirados” possui uma maior adequação linguística, uma vez que a princípio, os SIAs consistem na aplicação de conceitos e teorias imunológicas em problemas de Engenharia (otimização e/ou aprendizado de máquina). O termo “Algoritmos Imunoinspirados” propicia uma maior aproximação com a realidade, uma vez que nem todo uso dos SIAs pode ser encarado facilmente como um “sistema imune”. Entretanto, da mesma maneira que “Redes Neurais Artificiais”, o termo “Sistemas Imunes Artificiais”, apesar de não ser totalmente adequado, encontra-se fortemente estabelecido na comunidade e será usado neste texto.

Dado esse enfoque, o pesquisador De Castro (2001) propõe, inclusive, o termo Engenharia Imunológica, como sendo “[...] uma estrutura formal para o desenvolvimento de sistemas imunológicos artificiais”. Como comentado nesse mesmo texto, os Sistemas Imunológicos Artificiais surgiram a partir das tentativas de modelar e aplicar os princípios imunológicos no desenvolvimento de novas ferramentas computacionais.

SIAs surgiram em meados da década de 1980, sendo que o trabalho de Farmer, Packard e Perelson (1986) em redes imunes é considerado pioneiro na área. Entretanto, o estabelecimento de SIAs como uma área deu-se somente em meados da década de 1990. Nessa primeira etapa foram significativas as contribuições das equipes de Forrest, Hofmeyr e Somayaji (1997) e Dasgupta (1998) com estudos em algoritmos de seleção negativa.

No final da década de 90, tornaram-se conhecidos os trabalhos envolvendo seleção clonal, destacando-se a contribuição dada por De Castro et al. (2004) além de Zuben e Attux (2007), cujos trabalhos foram pesquisados.

Originalmente SIAs foram criados apenas como abstrações de processos encontrados no sistema imune. Entretanto, a partir do ano de 1998, data da publicação do primeiro livro na área, editado por Dasgupta vários pesquisadores da área tornaram-se interessados no processo de modelagem do sistema imune e aplicação de SIAs a problemas imunológicos. Isso tem relação com o fato, ressaltado por De



Castro e Timmis (2002), que muitos pesquisadores sentem dificuldades em identificar a diferença entre SIAs e trabalhos em Imunologia Teórica. Nesse texto, SIAs são conceituados como sistemas computacionais para solução de problemas, inspirados por Imunologia teórica, bem como funções, princípios e modelos imunológicos observados. Dessa maneira, o diferencial entre essas duas áreas encontra-se na aplicabilidade.

#### **4.4 Sistemas Resilientes**

Bisset et al. (2000), caracteriza Sistema Resiliente todo sistema computacional capaz de funcionar – frequentemente com suas capacidades reduzidas – na presença de falhas e possuir capacidades de recuperação, para alcançar algum estado anterior de funcionamento pleno. Afirma ainda que, sistemas deste tipo possuem dois modos de operação, modo normal e modo de integridade/recuperação, não podendo operar em ambos simultaneamente. O sistema opera em modo normal um defeito ocorrido não afete de modo drástico a vida do usuário. O que significa que o sistema deve estar acessível quando está em modo normal.

Para Bisset et al. (2000), um sistema tem sua integridade afetada quando um defeito causa perda de dados ou o corrompimento dos mesmos. Isto significa que um sistema operando em modo integridade deve estar configurado para evitar maiores perdas nos dados, mesmo que o sistema precise ficar inacessível para conseguir isto. Existe uma característica geral que liga esse tipo de sistema a outros dois, a necessidade de possuir um componente que mantenha o sistema funcionando na presença de falhas.

Por haver tal semelhança muitas vezes eles são confundidos entre si, trata-se de sistemas tolerantes a falhas e sistemas tolerantes a desastres, ainda de acordo com o autor.

Para Weber (2001), sistemas tolerantes a falhas pregam tanto disponibilidade e integridade quando confrontados com uma única falha, e em certas circunstâncias, quando se depara com múltiplas falhas. Sistemas desse tipo requerem algoritmos especiais ou componentes específicos, que garantam seu funcionamento correto mesmo com a presença de falhas. De acordo com Bisset et al (2000), sistemas tolerantes a desastres vão além da tolerância a falhas, esses sistemas exigem que a

perda de tempo de computação ocasionado por um desastre causado pelo usuário ou que tenha ocorrido naturalmente, não afete em hipótese alguma a disponibilidade do sistema, integridade dos dados ou perda dos mesmos.

## 5 Segurança em Camadas

Baseado no trabalho de Sicht (2001), o autor deste trabalho se utilizou do modelo de segurança em quatro camadas, sendo elas, Monitoramento, Detecção e Bloqueio, Segurança Interna e Recuperação e Disponibilidade.

Para avaliar uma estratégia de defesa, é importante começar com a identificação do tipo de dados que pode ser de interesse para os atacantes, então determinar onde residem os dados e avaliar o seu nível de vulnerabilidade. Este, por sua vez, irá ajudar a aplicar os recursos adequados para os sistemas certos.

Quando se pensa em segurança em camadas, é importante considerar a exploração de caminhos tomados por atacantes e *malware*. Por exemplo, a maioria dos ataques começa com algum tipo de ataque direcionado contra um usuário. Os invasores usam essas infecções iniciais como pontos de lançamento para chegar mais fundo na organização, onde eles podem acessar dados com valor real: as senhas de administrador (as "chaves do reino"), informações da conta financeira e bancos de dados de clientes com os dados pessoais são alguns dos alvos favoritos. Os invasores também podem querer envolver computadores comprometidos em outras atividades ilícitas, incluindo DDoS e redes de phishing. Em outras palavras: Todo e qualquer usuário é um alvo em potencial.

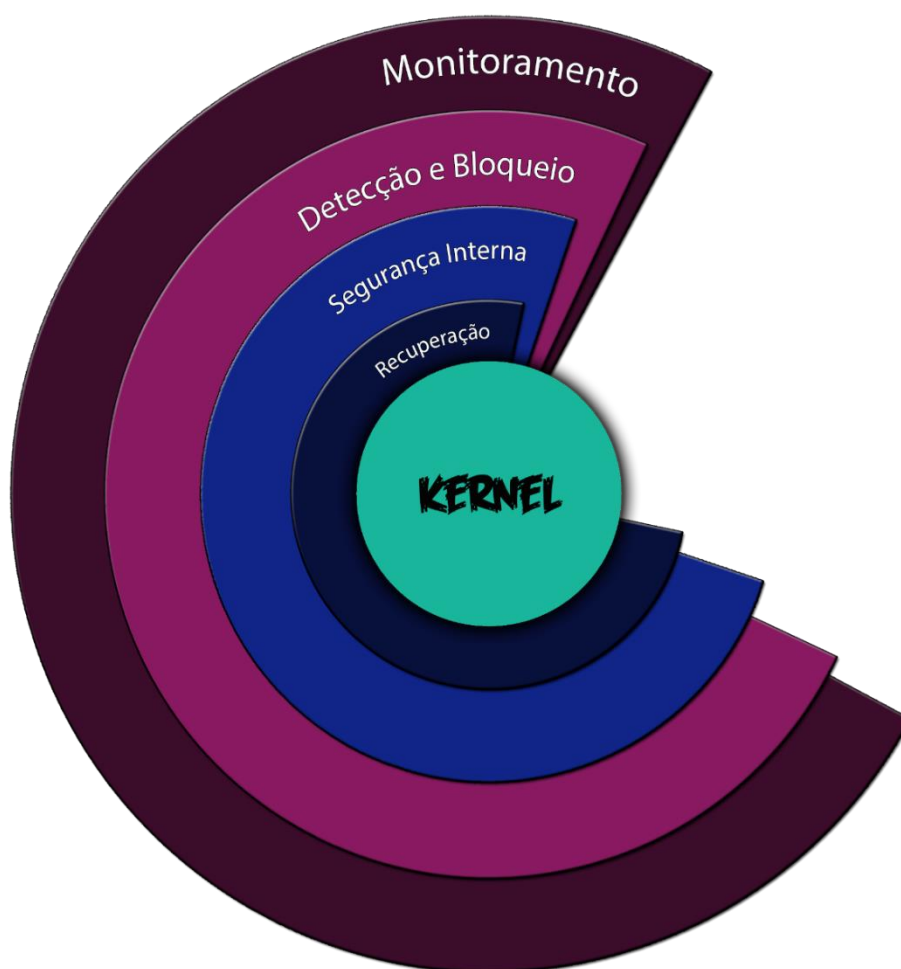
Ainda de acordo com Sicht (2001), a abordagem em camadas para a segurança pode ser implementada em qualquer nível de uma estratégia de segurança da informação. As organizações precisam ampliar suas redes de segurança para se proteger e se defender contra-ataques oportunistas e infecções.

O termo de segurança em camadas descreve uma estratégia defensiva com múltiplas camadas de defesa que são projetados para abrandar um atacante, um exemplo dessa estratégia pode ser observado na figura 8. Os militares chamam essa defesa de "defesa em profundidade". O objetivo está em abrandar um ataque causando baixas inimigas. No mundo digital, isso significa causar atrasos para os atacantes e detectá-los antes que eles possam causar sérios danos. Em alguns casos, uma camada devidamente colocada pode agir como um impedimento forte o suficiente para fazer com que o atacante tenha de olhar para um alvo mais fácil.

Se uma organização não tem conhecimento de um ataque, as camadas de segurança só podem retardar o atacante, mas um determinado atacante pode

eventualmente contornar as camadas de segurança. Monitoramento e relatórios consistentes são um requisito fundamental para o bem, da segurança em camadas. Adicionando uma camada de gerenciamento no topo de todas as camadas de segurança e vulnerabilidade é outro requisito e deve incluir fluxo de trabalho integrado para a remediação de vulnerabilidades detectadas.

Figura 7 - Disposição de Camadas do Orchid



Fonte: Elaborada pelo autor (2015)

### 5.1 Camada de Detecção e Bloqueio

É a camada responsável pela checagem do tráfego da rede encaminhado pelo sensor da camada de monitoramento e bloqueio das conexões caso alguma tentativa

de intrusão seja detectada. Neste capítulo serão apresentadas as referências teóricas e ferramentas utilizadas no desenvolvimento do protótipo do sistema.

### 5.1.1 Base de Dados

Objetivando os testes de detecção foi escolhida a base de dados NSL-KDD para comprovação dos métodos utilizados. Estes dados são uma evolução do conjunto de dados pertencentes ao KDD99, sendo esta uma versão da base inicial de dados criada pelo MIT Lincoln Labs, para o Programa de Avaliação de Detecção de Intrusão DARPA de 1998. Conforme Tavallae et al. (2009), a base KDD99 consiste de aproximadamente 4.900.000 vetores simples de conexão, cada um contendo 41 características e um rótulo indicando se é normal ou qual o tipo de ataque que pertence. De acordo com Kendall (1999), os tipos de ataques utilizados na base são mostrados abaixo:

- Ataques de Negação de Serviço (DoS): são aplicados para sobrecarregar algum recurso de memória ou recurso computacional para que usuários legítimos não consigam acessar o sistema. Alguns desses ataques abusam de uma característica específica do serviço atacado, outros já criam pacotes com formatos errados, o que atrapalha a reconstrução do pacote, enquanto outros utilizam erros de determinado serviço de rede;
- Ataques de Usuário para Administrador (U2R): ocorrem quando o invasor consegue acessar o sistema como usuário normal e então explora vulnerabilidades para conseguir acesso de administrador. Dentre os diversos tipos de ataques, o mais comum é o de overflow, que ocorre quando um programa grava muitos dados em um buffer estático sem verificação, permitindo ao usuário inserir mais dados do que os necessários, podendo então executar códigos arbitrários;
- Ataques de Remoto para Local (R2L): ocorrem quando um atacante consegue ganhar acesso em um sistema que ele não tem acesso através de envio de pacotes na rede de computadores. Existem várias formas de efetuar estes ataques, alguns podem explorar por segurança de políticas fracas, outros

podem utilizar *buffer overflow*, ou ainda podem utilizar engenharia social para infiltrar *trojans* e obter senhas de acesso;

- Ataques de Reconhecimento (Probing): são utilizados para escanear uma rede de computadores para encontrar informações sobre os sistemas instalados, ou mesmo para encontrar diretamente suas vulnerabilidades. Um atacante que possua um bom mapa de uma rede, pode obter informações sobre pontos mais fracos da rede. Alguns aplicativos podem ser utilizados para varrer uma grande área de rede procurando por vulnerabilidades conhecidas.

De acordo com Tavallae et al. (2009), a qualidade dos dados contidos no KDDCUP99 era baixa, após estudos realizados argumentaram sobre problemas na coleta dos dados, como a sobrecarga da ferramenta de coleta, diferença dos dados fornecidos com os dados costumeiramente coletados em uma rede real, e sobre a falta de explicações sobre definições específicas dos ataques. Também levantaram problemas argumentados em outros artigos, sobre diversas críticas a base KDD, como por exemplo, a quantidade desproporcional de ataques nas bases de dados. Para melhorar a qualidade dos dados, uma nova base foi proposta, o NSL-KDD, que retirou os dados redundantes e distribuiu de outra forma os dados do KDD99, separando os dados em três conjuntos:

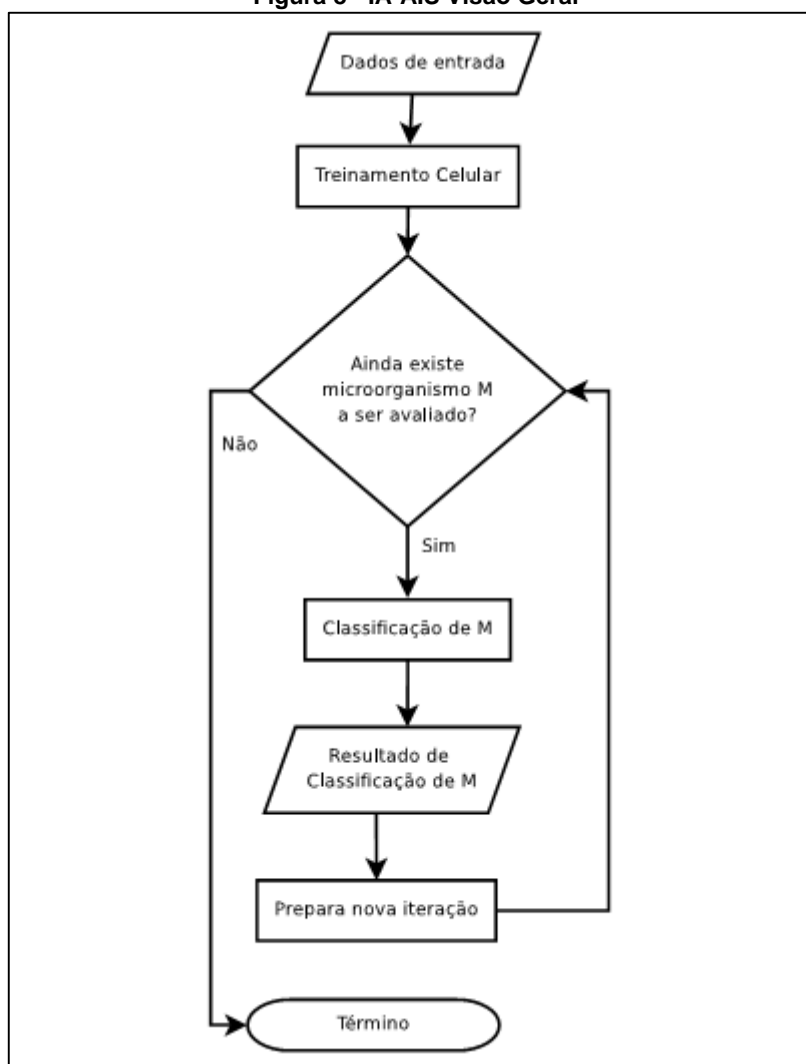
- KDDTrain+: que possui a base de treino;
- KDDTest+: que possui a base de testes;
- KDDTest-21: que possui os dados que não foram bem classificados pelos algoritmos utilizados na criação da base.

### 5.1.2 Algoritmo IA-AIS

O primeiro algoritmo utilizado por Uchôa (2009), em sua tese foi o IA-AIS (Innate and Adaptive Artificial Immune System), esse algoritmo apoia-se na distinção entre próprio e não-próprio, por questões de simplificação, e foi apresentado inicialmente em Guzella et al. (2005). Uma visão geral do algoritmo pode ser observada na Figura 9, além disso, as etapas desse algoritmo foram descritas abaixo por Uchôa (2009) como:

1. Criação de uma população inicial de macrófagos e posterior treinamento para reconhecer padrões moleculares de patógenos, mas não do próprio organismo.
2. Criação de uma população inicial de células B e células T auxiliares, ambas selecionadas por um processo de seleção negativa.
3. Criação de uma população inicial de células T regulatórias, capazes de reconhecer elementos do próprio organismo.
4. Após a etapa inicial de treinamento, para cada micro-organismo M a ser avaliado, é feita sua classificação.
5. Apresenta-se M, que pode ser ligado por macrófagos e por linfócitos, à população de macrófagos.
6. Se algum macrófago for ativado, é iniciada uma resposta imune, com a eliminação do patógeno e estimulação ou indução de linfócitos B e T.
7. Apresenta-se M, como um grupo de antígenos Ags, à população de células B. Se nenhuma célula B for estimulada, uma resposta imune não é iniciada, e a apresentação da entrada é finalizada.
8. Nessa etapa, tem-se a avaliação da entrada pelos linfócitos T, auxiliares e regulatórios. Do somatório das respostas desses linfócitos, tem-se a ativação ou supressão das células B estimuladas no passo anterior. É importante observar que apenas na presença do segundo sinal, haverá estimulação da célula B. Ou seja: caso nenhum linfócito T reconheça o antígeno, então a célula B é suprimida.
9. As células B ativadas no passo anterior são clonadas, numa proporção direta à sua afinidade ao antígeno sendo analisado: quanto maior sua afinidade, mais clones são gerados. O sistema como um todo é calibrado por um parâmetro ( $\mu$ ) que define o número máximo de clones permitidos por célula.
10. Nesse passo, os clones gerados anteriormente sofrem hipermutação somática, calibrada pelo parâmetro  $\alpha$ . Após isso, a entrada que ativou inicialmente as células B são avaliadas por esses clones, calculando-se a afinidade. Os clones são então ordenados de acordo com essa afinidade e selecionados, de acordo com o parâmetro  $N_c$ , que indica o número máximo de clones permitidos por iteração.
11. Caso a célula B receba menos sinais estimulatórios que regulatórios, ela é suprimida, ou seja: seu TTL é zerado, fazendo com que a mesma seja eliminada do sistema ao término da iteração. Nesse caso, a entrada é classificada como pertencente a um uso normal do computador.
12. Terminada a apresentação da entrada, o sistema é preparado para uma nova iteração, decrementando-se o TTL de todas as células ainda vivas, sendo eliminadas aquelas com TTL zerado. Novas células são geradas, de forma semelhante aos passos 1, 2 e 3, mas em quantidade bem menor.

Figura 8 - IA-AIS Visão Geral



Fonte: Uchôa (2009)

Uchôa (2009), inspirou-se inicialmente no CLONALG, porém em seu algoritmo existem várias diferenças com o CLONALG, além do uso de mecanismos da imunidade inata.

O CLONALG utiliza uma população específica de memória, onde todas as células presentes nessa população são, de fato, células de memória.

No IA-AIS, utiliza-se uma abordagem diferente, onde quando uma célula é criada, define-se um tempo de vida e um valor inteiro positivo representando a contagem regressiva para a morte da célula. Esse valor é decrementado após cada vez que um micro-organismo é apresentado ao sistema, com a eliminação de células com um valor nulo de tempo de vida.

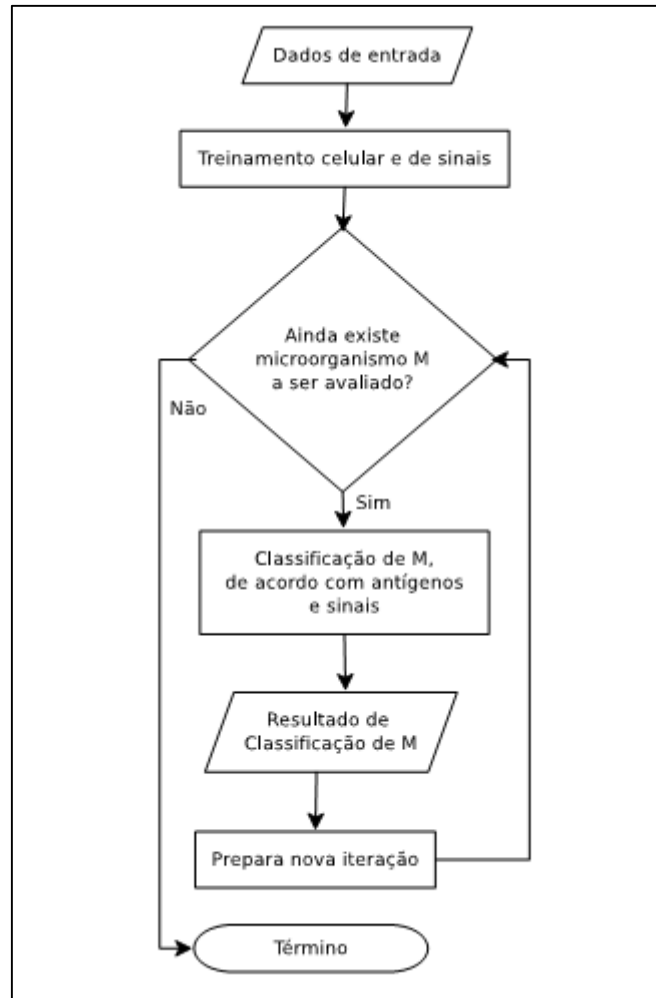


### 5.1.3 Algoritmo DT-AIS

Segundo Uchôa (2009), o algoritmo DT-AIS (Danger Theory Artificial Immune System) foi baseado na Danger Theory e além disso também é baseado no uso de mecanismos inatos e adaptativos do sistema imune, para uso em detecção de falhas. As etapas desse algoritmo são descritas abaixo, além disso uma visão geral do algoritmo pode ser verificada na Figura 10.

1. Cria-se uma população inicial de APCs e células T virgens, a partir de um conjunto de antígenos escolhidos para treinamento.
2. O tecido é treinado (evolutivamente) para emissão de sinais coestimulatórios, a partir da percepção de perigo. Esse treino é feito a partir de sinais existentes no sistema.
3. Um novo micróbio M é apresentado ao sistema e é pré-processado por APCs, que coletam dados do ambiente e definem se apresentarão ou não sinais coestimulatórios, a partir da ligação com o micróbio e dados ambientais.
4. De acordo com o tipo celular e os sinais do sistema, há um aumento ou decréscimo no reconhecimento de M como sendo ou não uma tentativa de intrusão.
5. Determina-se a existência ou não de anomalia a partir do nível de reconhecimento antigênico do sistema.
6. Diminui o tempo de vida de todas as células no sistema e gera novas células T virgens de forma aleatória, retornando ao passo 3 do algoritmo, até que não haja mais microorganismos a serem avaliados.

**Figura 9 - DT-AIS Visão Geral**



Fonte: Uchôa (2009)

Segundo Uchôa (2009), as principais diferenças do DT-AIS com o IA-AIS são o menor uso de diferentes tipos celulares e o tratamento de sinais associados a antígenos.

No DT-AIS, o processo de regulação da ativação celular é feito por uma análise do contexto, através da ativação de outras células e sinais associados ao antígeno, enquanto no IA-AIS essa tarefa é desempenhada por células T regulatórias.

#### 5.1.4 AISF (Artificial Immune System Framework)

Conforme Uchôa (2009), para a implementação dos algoritmos IA-AIS e DT-AIS, foi desenvolvido o *Artificial Immune System Framework*, biblioteca de classes e funções em Python, este framework conta com suporte a um grande número de processos e elementos do sistema imune.

## 5.2 Camada de Segurança Interna

É a camada responsável pela segurança interna do sistema, implementando *hardening* do servidor com a utilização de Tuning de Kernel e controle de acesso MAC (Mandatory Access Control) restringindo acessos indevidos e garantindo assim a integridade do sistema. Neste capítulo serão apresentadas as referências teóricas e ferramentas utilizadas no desenvolvimento do protótipo do sistema.

### 5.2.1 Hardening

Segundo Melo et al. (2006), *hardening* nada mais é do que procedimentos de segurança, ou técnicas para ajustes personalizados em um sistema. Estes procedimentos têm como principal objetivo aumentar a segurança do sistema. Quando as técnicas de *hardening* são aplicadas, existem três fatores que devem ser considerados: segurança, risco e flexibilidade, assim equilibrando-os para melhor manter a continuidade do negócio e com segurança. Não existem sistemas 100% seguros, porém quanto maior a segurança, menor será o risco e flexibilidade, e, quanto maior a flexibilidade, maior o risco e menor a segurança. Deve-se entender que não há uma regra, para equilíbrio destes fatores, cada caso deve ser analisado como único, e nem todas as normas e técnicas precisarão ser implementadas.

#### 5.2.1.1 Tuning do Kernel

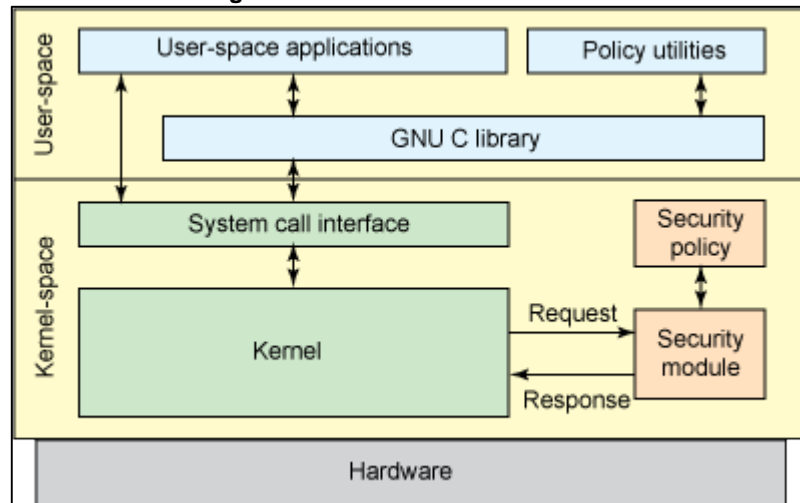
De acordo com Melo et al. (2006), o *tuning do kernel* é a realização de ajustes detalhados no sistema operacional. Como o objetivo é melhorar a segurança, o desempenho do servidor e o tratamento dos pacotes da pilha TCP/IP (Transmission Control Protocol/Internet Protocol), deve-se focar três pontos: *tuning TCP*, *tuning ICMP* (Internet Control Message Protocol) e *tuning IP*. Ainda segundo Melo et al. (2006), o protocolo TCP possui características que devem ser lembradas, entre elas que os 20 bytes que fazem parte do cabeçalho são importantes para um firewall, podendo ser verificado o estado de uma conexão. Estas informações auxiliam na defesa de possíveis ataques.

O ICMP é o controle de mensagens do protocolo da Internet, um motivo para o tuning ICMP, é que o pacote IP que carrega a mensagem ICMP pode conter informações que podem ser utilizadas com o objetivo de realizar o *fingerprint*, técnicas de extração de informações de um host ou alvo. As políticas do *firewall* devem ser equilibradas quanto às solicitações de ICMP do tipo 8 (echo request), pois os administradores de rede utilizam com frequência o *ping* como ferramenta de diagnóstico. Em um *firewall*, o *tuning* de IP define de as interfaces de rede poderão ou não trocar pacotes entre si.

### 5.2.1.2 Implantação de Controle MAC

Conforme Zucco (2008), o SELINUX (Security-Enhanced Linux) é uma implementação de uma flexível e refinada arquitetura MAC (Mandatory Access Control). SELINUX provê uma política de segurança sobre todos os processos e objetos do sistema baseando suas decisões em labels contendo uma variedade de informações relevantes à segurança. A lógica da política de tomada de decisões é encapsulada dentro de um simples componente conhecido como servidor de segurança (security server) com uma interface geral de segurança. Uma visão geral do SELINUX pode ser observada na figura 11.

Figura 10 - Visão Geral do SELINUX



Fonte: Jones (2008)

Ainda segundo Zucco (2008), SELINUX é parte integrante de algumas distribuições Linux como o Fedora Core e a Red Hat Enterprise Linux. A principal característica é a limitação das ações dos usuários e programas aplicando políticas de segurança em todo o sistema. Quando o sistema não está implementado com o SELINUX, alguns erros de software ou alterações de configuração podem tornar o sistema mais suscetível a falhas e ou vulnerabilidades. Assim, as políticas do SELINUX fornecem segurança extra contra o acesso não autorizado.

O SELINUX, mesmo se tornando um projeto público a poucos anos, se origina de trabalhos de Bell e LaPadula (1975) de várias décadas atrás. Nos anos 1980, seu trabalho influenciou vários desenvolvedores de sistemas do governo dos Estados Unidos, que acabaram criando o Orange Book Trusted Computer System Evaluation Criteria (NSA 1985), definindo 6 classes de avaliação de sistemas seguros: C1, C2, B1, B2, B3 e A1.

As classes C1 e C2 englobam sistemas que dependem de controle de acesso discricionário (DAC). Classes B1 e superiores, devem implementar controle de acesso mandatário (MAC).

O uso do SELINUX segundo Jones (2008) tem por objetivos de segurança primários:

- Isolamento das aplicações: busca o nível do menor privilégio no uso de aplicações. Com isso, um problema de segurança em uma aplicação

isolada (como bugs, vulnerabilidades e zero-days) não influencia o sistema como um todo, não comprometendo o funcionamento das outras aplicações. Desta maneira, diminui o efeito da exploração de vulnerabilidades, limitando a propagação de erros e reduzindo a necessidade de aplicação imediata de patches de segurança em aplicativos vulneráveis;

- Fluxo de informações: garantia de que a informação deve seguir caminhos predefinidos para acesso entre os processos;
- Confidencialidade: a informação não estará disponível ou será divulgada a indivíduos, entidades ou processos sem a devida autorização;
- Integridade: disponibilidade de informações confiáveis, corretas e dispostas em formato compatível com o de sua utilização. A informação manipulada mantém todas as características originais estabelecidas pelo proprietário da informação, incluindo controle de mudanças e garantia do seu ciclo de vida (nascimento, manutenção e destruição).
- Auto-proteção: como o SELINUX é aplicado diretamente sobre o kernel do Linux, além de proteger as políticas de segurança, ele também tem por objetivo proteger o próprio sistema operacional para se auto proteger;
- Menor privilégio: garante que as políticas aplicadas estão corretas e de que os processos possuem apenas o acesso necessário para realizar a sua função, e nada mais do que isso;
- Separação de papéis: definição das permissões de usuários e processos para evitar a elevação de privilégios e suas consequências.

### 5.3 Camada de Recuperação e Disponibilidade

É a camada responsável pela recuperação do sistema, utilizando sistemas de controle de versão. Sua função é trabalhar em conjunto com a camada de monitoramento e recuperar o sistema e suas configurações em caso de invasão. Neste capítulo serão apresentadas as referências teóricas e ferramentas utilizadas no desenvolvimento do protótipo do sistema.

#### 5.3.1 Sistemas de Controle de Versão

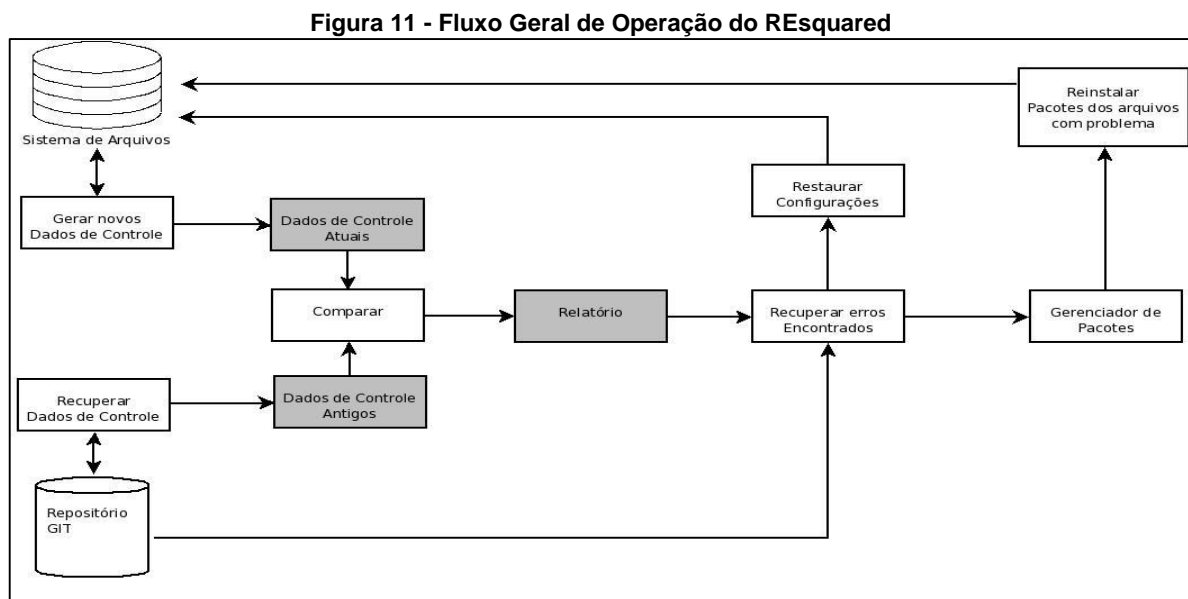
Segundo Gift e Shand (2009), VCS (Version Control System) são ferramentas utilizadas para automatizar o controle de versão de documentos, códigos fonte ou quaisquer arquivos utilizados dentro de um determinado ambiente. Com um VCS é possível gerenciar o histórico de um arquivo e obter simultaneamente duas ou mais versões do mesmo. Seu funcionamento consiste basicamente em manter um repositório com os arquivos que devem ser controlados, no repositório também são armazenados os arquivos de controle e logs. Conforme Berliner e Prisma (1990), para otimizar o espaço em disco ocupado pelas diferentes versões dos arquivos, a maior parte dos VCS utiliza o método de compressão *Delta*, que consiste em armazenar apenas a diferença entre versões sucessivas de um arquivo. De acordo com Suel e Memon (2002), o princípio geral da compressão *Delta* é aplicar sucessivamente as diferenças armazenadas sobre a versão inicial do arquivo, deste modo, partindo da versão inicial pode-se obter a mais recente ou qualquer outra intermediária.

A maioria dos VCS trabalha com o modelo *check-in/check-out*, baseado em um controle de versão individual para cada arquivo, ficando armazenados em um repositório e o usuário não pode interagir diretamente com eles. Inicialmente o usuário deve fazer um *check-out*, ou seja, fazer a retirada dos arquivos que ele deseja editar, copiando os arquivos para um diretório de trabalho. Futuramente os arquivos modificados terão novas versões a partir do momento em que o usuário efetuar seu *check-in*, ou seja, dar entrada dos arquivos no repositório.

A principal função do VCS é guardar todo o histórico de desenvolvimento do documento, desde o primeiro *check-in* até sua última versão. Isso permite que seja possível recuperar uma determinada versão de qualquer data mais antiga.

### 5.3.2 REsquared – Recuperação com Versionamento

Segundo Cavalcante (2010) REsquared é um sistema integrado, que combina a verificação de integridade e mecanismos de restauração presentes em sistemas de controle de versão. Para isto foi utilizada a detecção de intrusão baseada em integridade de arquivos e ferramentas de controle de versão de arquivos e diretórios para armazenar os diferentes estados de configuração do sistema, com objetivo final de criar um sistema operacional GNU/Linux resiliente. REsquared foi projetado de maneira inteiramente modular, o que garante alta portabilidade. A Figura 12 exibe o diagrama de fluxo geral de operação.



Fonte: Cavalcante (2010)

Ainda segundo Cavalcante (2010), inicialmente o repositório é gerado com informações coletadas do sistema de arquivos quando este é considerado íntegro, no repositório também são guardados os arquivos de configuração encontrados – arquivos de texto. Cada arquivo sob observação - binários e texto - do REsquared possui um descritor de integridade dentro do repositório, este descritor contém todas as informações sobre seus metadados e hashes. Após esta inicialização, a cada verificação são calculados novos dados de controle do sistema de arquivos, esses dados são comparados com os armazenados no repositório.



Dessa forma, podem ser encontradas evidências de alteração. O relatório possui todas essas informações de forma clara para o usuário, e serve de base para o recuperador tomar decisões sobre a correção do sistema de arquivos. No repositório ficam armazenadas somente configurações, que é a parte “customizável” dos sistemas operacionais. Todos os executáveis – programas -podem ser obtidos em fontes públicas através de um gerenciador de pacotes, que é peça fundamental em toda distribuição Linux atual.

O gerenciador de pacotes é o responsável por reinstalar os arquivos binários possivelmente comprometidos. Após a reinstalação são restauradas as configurações para que o sistema volte a sua normalidade. Além disso, o gerenciador de pacotes também permite uma rápida maneira de criação de arquivos de controle para um determinado pacote.

Desta maneira o usuário pode manter sob controle um aplicativo apenas sabendo a qual pacote ele pertence, garantindo sua integridade completa.

## **5.4 Camada de Monitoramento**

É a principal camada do sistema, responsável pela comunicação entre as demais camadas do sistema. Sua função é interligar todas as camadas e direcionar tarefas para as mesmas, monitorando tanto a rede quanto o sistema de arquivos. Neste capítulo serão apresentadas as referências teóricas e ferramentas utilizadas no desenvolvimento do protótipo do sistema.

### **5.4.1 Sensor**

Esse módulo é o sensor do sistema operacional, realizando as seguintes operações:

- Procurar com expressões regulares dentro de logs de aplicativos, por alertas;
- Monitorar o Kernel do Linux, obtendo informações vitais do sistema operacional;
- Receber e tratar logs do firewall;

- Monitorar o tráfego de rede;

Monitorar o tráfego de rede é o papel mais importante do Sensor. Ele irá testar os pacotes de rede com os algoritmos IA-AIS e DT-AIS, e avisará ao agente para que possa tomar as medidas necessárias. O Sensor utiliza a biblioteca *scapy* para obter os pacotes de rede. Estes são recebidos, normalizados e o conteúdo do *payload* testado pelos algoritmos.

## **6 Implantação**

Escolheu-se para este trabalho a implementação de um sistema com uma nova estratégia na identificação e bloqueio de intrusos em redes. O protótipo foi desenvolvido utilizando uma arquitetura integrada que combina verificação de integridade e mecanismos de recuperação além de metodologias e algoritmos inspirados em conceitos e processos do sistema imune inato e adaptativo, buscando a solução de uma série de problemas e falhas existentes nos dispositivos atuais. O objetivo a ser alcançado com a implementação do protótipo dentro do contexto da arquitetura de detecção apresentada é realizar uma prova de conceito que possibilite avaliar a viabilidade de implementação do sistema em um ambiente real.

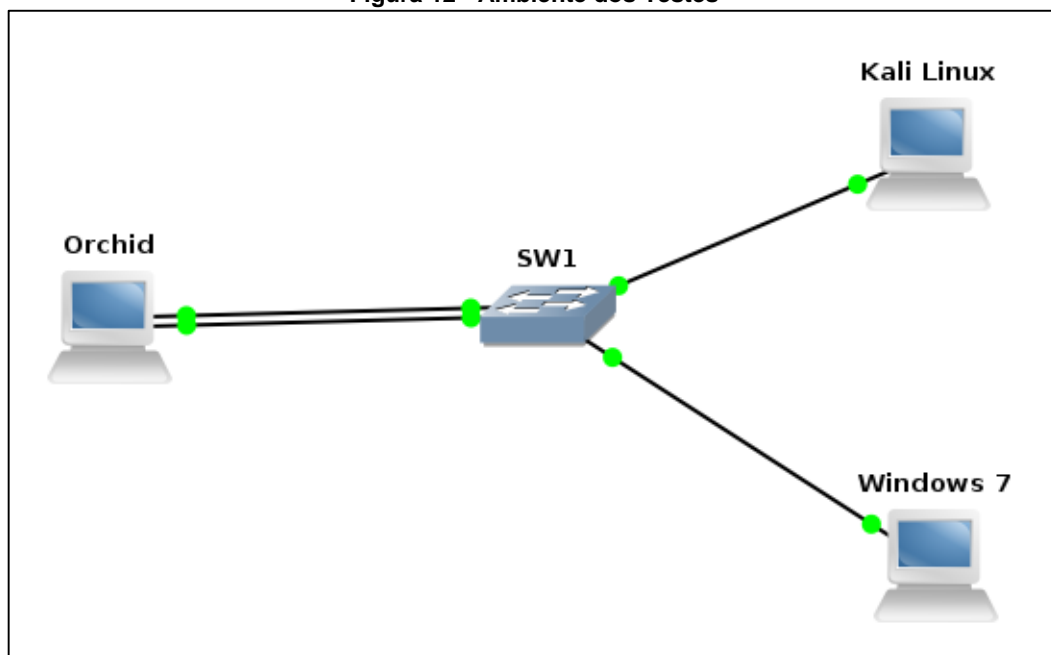
### **6.1 Local e Período**

O trabalho está sendo desenvolvido no Núcleo de Tecnologia da Informação da Delegacia de Polícia Federal em Campinas, durante o período de setembro de 2013 a junho de 2015. Sendo que o primeiro protótipo deste projeto será apresentado como trabalho de conclusão de curso de Graduação em “Tecnologia em Segurança da Informação” em junho de 2015 no CEETEPS/Faculdade de Tecnologia de São Paulo unidade Americana.

### **6.2 Ambiente de Testes**

Para efetuar os testes de prova de conceito foi montado um ambiente de testes, representado na figura 12.

Figura 12 - Ambiente dos Testes



Fonte: Elaborada pelo autor (2015)

Os recursos utilizados no ambiente de teste foram:

- **Hardware**

01 Servidor Dell PowerEdge R720 Intel Xeon E5-2600, 64 GB de memória RAM, 02 interfaces de rede Gigabit ethernet.

02 Desktops HP S5-1410br Intel Core i3-2120, 4GB, 01 interface de rede.

O sistema Orchid foi escrito para ser executado no ambiente Linux, distribuição CentOS 6.4 com o kernel 2.6.32. O programa é implementado em linguagem Python e alguns scripts na linguagem Bash.

- **Criação do Repositório de Recuperação**

Todo o repositório e o código dos módulos foi instalado em uma partição separada, cifrada e montada em modo leitura.

- **Treinamento do Sensor de Rede**

Para identificar corretamente as entradas que pudessem compor perfis de ataque foi necessário que os algoritmos recebessem o devido treinamento com um conjunto suficiente de padrões que representassem tanto comportamento normal como intrusivo.

- **Hardening do S.O.**

O hardening do sistema abordados no protótipo foram:

- Sistema e Processos
- Memória
- Sistema de Arquivos
- Controle de Acesso
- Rede
- Serviços e Aplicações

## 7 Resultados

Com o objetivo de avaliar o modelo proposto, vários experimentos foram efetuados com o protótipo implementado, os quais apresentaram resultados que demonstram as vantagens da abordagem proposta neste trabalho.

Para a elaboração dos testes de detecção foi feita a divisão da base de dados em três amostras, uma com 25% da base, uma com 75% e uma com a base em sua totalidade.

O protótipo foi submetido a 3 cargas de treinamento antes da avaliação, para isso foi utilizada a base KDDTrain+ com 125.973 conexões, sendo elas 58.630 amostras de ataque e 67.343 de tráfego normal, totalizando 377.919 amostras.

A seguir são detalhadas as fases dos experimentos realizados bem como os resultados obtidos.

### 7.1 Resultados dos Testes de Detecção

As amostras utilizadas da base NSL-KDD para realização dos testes de detecção são apresentadas abaixo:

- Para os testes com 25% dos dados: Tabela 2;
- Para os testes com 75% dos dados: Tabela 3;
- Para os testes com 100% dos dados: Tabela 4.

**Tabela 2 - Amostra de 25%**

	Ataque	Normal	Total de Conexões
<b>KDDTrain+</b>	58.630	67.343	125.973
<b>KDDTest-21</b>	2.425	538	2.963
<b>KDDTest+</b>	3.208	2.428	5.636

Fonte: Elaborada pelo autor (2015)

**Tabela 3 - Amostra de 75%**

	Ataque	Normal	Total de Conexões
<b>KDDTrain+</b>	58.630	67.343	125.973
<b>KDDTest-21</b>	7.274	1.614	8.888
<b>KDDTest+</b>	9.625	7.283	16.908

Fonte: Elaborada pelo autor (2015)

Tabela 4 - Amostra de 100%

	Ataque	Normal	Total de Conexões
KDDTrain+	58.630	67.343	125.973
KDDTest-21	9.698	2.152	11.850
KDDTest+	12.833	9.711	22.544

Fonte: Elaborada pelo autor (2015)

Os testes executados com o algoritmo IA-AIS utilizando as bases KDDTest+ e KDDTest-21 apresentaram os seguintes resultados:

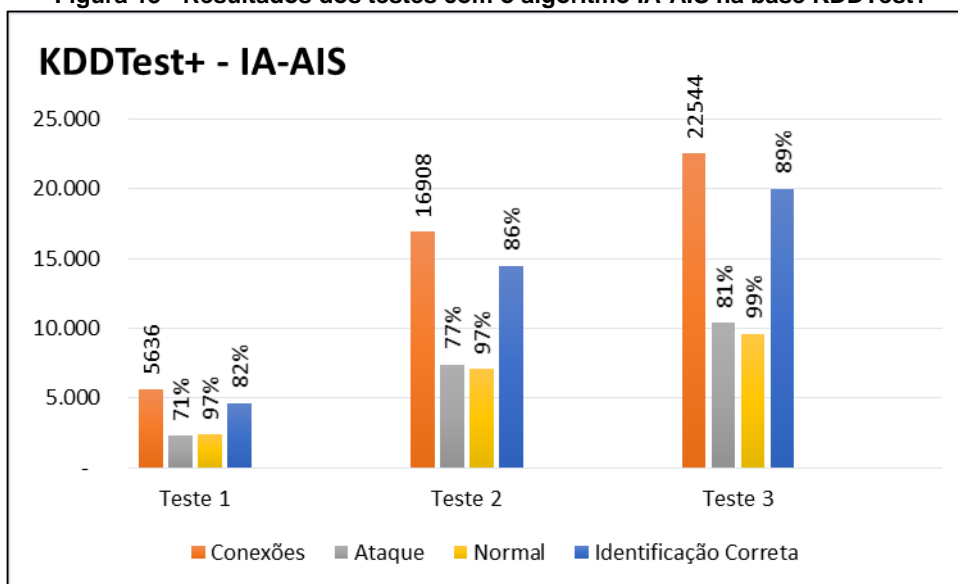
- Para os dados do KDDTest+: Tabela 5 e Figura 13;
- Para os dados do KDDTest-21: Tabela 6 e Figura 14;

Tabela 5 - Resultados dos testes com o algoritmo IA-AIS na base KDDTest+

KDDTest+	Teste 1		Teste 2		Teste 3	
<b>Conexões</b>	<b>5.636</b>		<b>16.908</b>		<b>22.544</b>	
<b>Ataque</b>	2.278	71%	7.411	77%	10.395	81%
<b>Normal</b>	2.355	97%	7.065	97%	9.614	99%
<b>Identificação Correta</b>	4.633	82%	14.476	86%	20.009	89%

Fonte: Elaborada pelo autor (2015)

Figura 13 - Resultados dos testes com o algoritmo IA-AIS na base KDDTest+



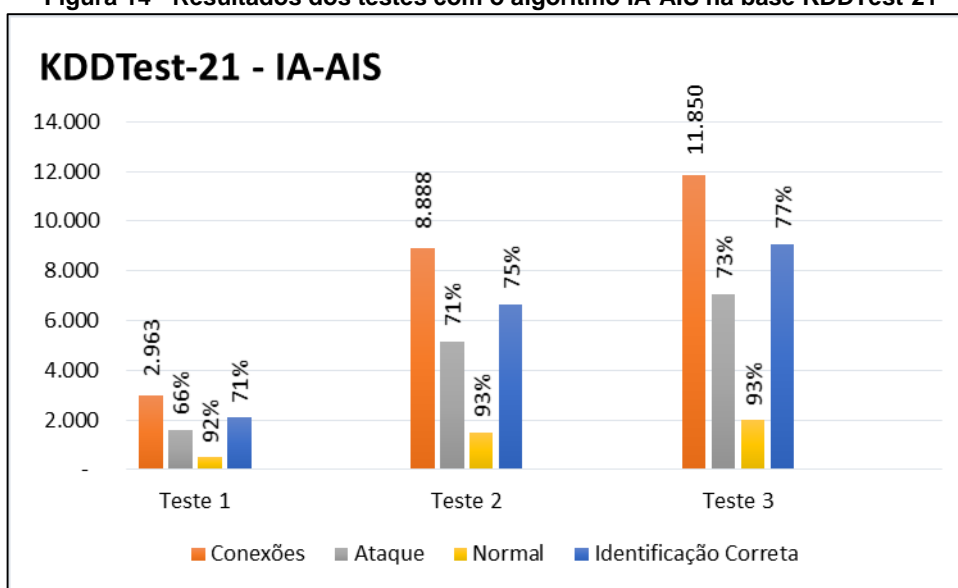
Fonte: Elaborado pelo autor (2015)

Tabela 6 - Resultados dos testes com o algoritmo IA-AIS na base KDDTest-21

KDDTest-21	Teste 1		Teste 2		Teste 3	
<b>Conexões</b>	<b>2.963</b>		<b>8.888</b>		<b>11.850</b>	
<b>Ataque</b>	1.600	66%	5.164	71%	7.080	73%
<b>Normal</b>	495	92%	1.501	93%	2.001	93%
<b>Identificação Correta</b>	2.095	71%	6.665	75%	9.081	77%

Fonte: Elaborada pelo autor (2015)

Figura 14 - Resultados dos testes com o algoritmo IA-AIS na base KDDTest-21



Fonte: Elaborado pelo autor (2015)

Os resultados demonstraram que o algoritmo IA-AIS é eficiente na detecção de tráfego, nota-se relativa melhora nos resultados mediante aumento nas amostras. E apesar dos resultados apresentados serem baixos para uma implementação comercial da ferramenta melhores resultados podem ser alcançados com uma futura evolução do algoritmo.

Os testes executados com o algoritmo DT-AIS utilizando as bases KDDTest+ e KDDTest-21 apresentaram os seguintes resultados:

- Para os dados do KDDTest+: Tabela 7 e Figura 15;
- Para os dados do KDDTest-21: Tabela 8 e Figura 16;

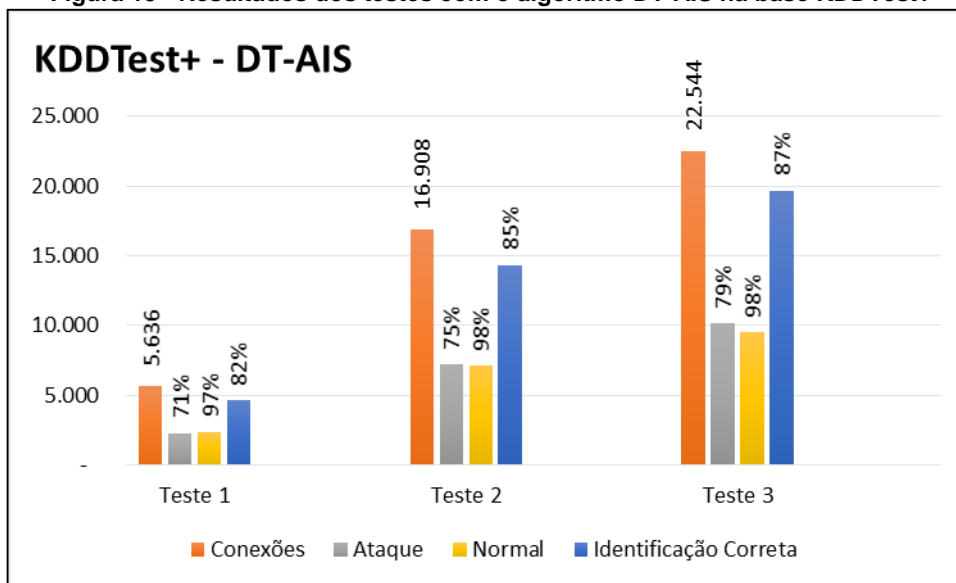


Tabela 7 Resultados dos testes com o algoritmo DT-AIS na base KDDTest+

KDDTest+	Teste 1		Teste 2		Teste 3	
<b>Conexões</b>	<b>5.636</b>		<b>16.908</b>		<b>22.544</b>	
<b>Ataque</b>	2.278	71%	7.219	75%	10.138	79%
<b>Normal</b>	2.355	97%	7.101	98%	9.517	98%
<b>Identificação Correta</b>	4.633	82%	14.320	85%	19.655	87%

Fonte: Elaborada pelo autor (2015)

Figura 15 - Resultados dos testes com o algoritmo DT-AIS na base KDDTest+



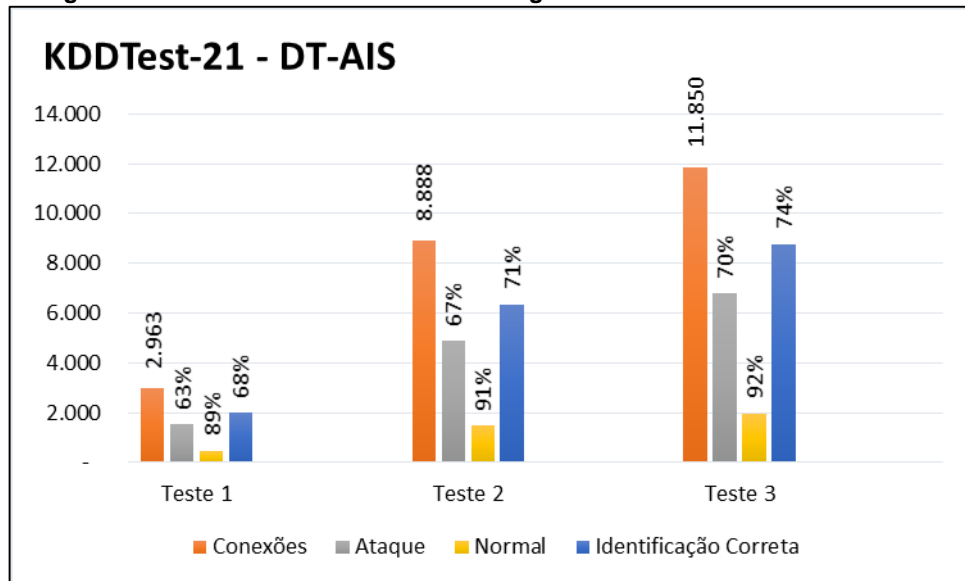
Fonte: Elaborado pelo autor (2015)

Tabela 8 Resultados dos testes com o algoritmo DT-AIS na base KDDTest-21

KDDTest-21	Teste 1		Teste 2		Teste 3	
<b>Conexões</b>	<b>2.963</b>		<b>8.888</b>		<b>11.850</b>	
<b>Ataque</b>	1.527	63%	4.873	67%	6.789	70%
<b>Normal</b>	479	89%	1.469	91%	1.969	92%
<b>Identificação Correta</b>	2.006	68%	6.342	71%	8.758	74%

Fonte: Elaborada pelo autor (2015)

Figura 16 - Resultados dos testes com o algoritmo DT-AIS na base KDDTest-21



Fonte: Elaborado pelo autor (2015)

Os resultados demonstraram que o algoritmo DT-AIS apesar de também eficiente na detecção de tráfego apresentou resultados ligeiramente inferiores aos do IA-AIS. Seus resultados na detecção também apresentaram melhora com o aumento nas amostras, porém melhores resultados podem ser alcançados com uma futura evolução do algoritmo.

## 7.2 Resultados dos Testes de Recuperação

Para efetuar os testes de recuperação foi necessário que o sensor fosse desligado pois cada alerta recebido fazia com que a conexão derrubada.

- Na primeira tentativa foi feita uma conexão remota com a conta “admin”. Logado no Orchid o usuário tenta fazer download de um *exploit* para escalar privilégios - Linux PERF\_EVENTS Local Root – para conseguir acesso de super-usuário. Com o sensor ativo a tentativa de download é detectada e a conexão é bloqueada.
- Na segunda tentativa o acesso é local e o sensor desativado para fins de teste. O download do *exploit* ocorre e sua utilização é bem-sucedida, permitindo a criação de uma nova conta de super-usuário denominada “root1”. Logo após, é criado um novo diretório e feita a cópia de 100 arquivos. Para melhorar o

cenário um *rootkit* – *t0rn rootkit* - também é instalado. Terminadas as alterações o sensor foi reativado e a primeira tarefa após sua inicialização é verificar alertas nos logs e alterações no sistema de arquivos. O sensor acusou alteração em 6 binários do sistema, nos arquivos */etc/passwd* e */etc/shadow*, além do diretório criado e arquivos copiados. Feita a identificação o diretório e os arquivos foram apagados, realizada a restauração dos arquivos */etc/passwd* e */etc/shadow* e a interface com o gerenciador de pacotes foi acionada para remoção e reinstalação dos binários alterados.

Os resultados mostraram que o mecanismo REsquared é muito eficiente nas tarefas de detecção e recuperação dos arquivos selecionados para proteção, tornando o Orchid um sistema resiliente. Porém, ainda são necessários mais testes visando melhor performance e aproveitamento de espaço do repositório para que possa ser utilizado em uma aplicação comercial.

## 8 Trabalhos Futuros

Algumas propostas surgiram no decorrer deste projeto, as quais agora são apresentadas como possíveis caminhos para continuidade do desenvolvimento do Orchid.

- É muito importante incluir um inspetor de aplicações para identificação do tipo de aplicação acessando a rede, além de realizar uma análise de dados mais profunda do código executável em um trabalho futuro.
- Uma nova abordagem de isolamento, utilizando Linux containers em detrimento ao *chroot* é uma potencial proposta de trabalho futuro, podendo ser embasada neste trabalho como ponto inicial. Para tanto é fortemente indicado o uso do Docker.
- Uma segunda proposta é a implementação da técnica SPA (Single Packet Authorization), fortalecendo a segurança em acessos remotos ao sistema. Este funcionaria como o software FWKnop, implementando SPA com uso de criptografia assimétrica e simétrica, porém com grande flexibilidade em sua configuração.
- Inviabilidade de captura de dados em ambientes criptografados, pois a biblioteca e funcionalidades implementadas não permitem decifrar o conteúdo deste tipo de seção.
- O modo promíscuo da interface de rede que é ativado pelo sensor necessita que o segmento monitorado opere com broadcast. Caso contrário, existe certa limitação na abrangência da monitoração.

## 9 Conclusões

Segurança de Informação não pode ser conseguida simplesmente pela aplicação de diversas ferramentas de segurança, por sistemas ou produtos, no entanto falhas de segurança são menos prováveis, através da implementação de políticas sólidas de segurança, processos, procedimentos e produtos.

A detecção de intrusão é um campo de pesquisa recente, e devido a isso ainda não atingiu maturidade suficiente em mecanismos de segurança. Mesmo sendo amplamente adotados, os sistemas de detecção de intrusão atuais não apresentam soluções para problemas com sua própria resiliência e a privacidade das informações trocadas entre os diferentes módulos do sistema.

O Orchid tem como objetivo alavancar o desenvolvimento de conceitos ainda pouco discutidos na literatura atual, integrando diversas soluções para os problemas ainda em estudo. Nesse protótipo várias camadas de defesa são aplicadas para conceber um sistema de segurança contra falhas. A ideia por trás de segurança de defesa em múltiplas camadas é gerenciar o risco de segurança com várias estratégias defensivas, de modo que se uma camada de defesa acaba por ser inadequada, outra camada de defesa será, de preferência, evitando uma violação completa.

Acredita-se que, no mínimo, os gestores devem aplicar uma série de defesas de perímetro de segurança de modo que seus recursos não estão expostos a ataques externos e garantir que o sistema de segurança não é limitado pelo elo mais fraco da camada de segurança, pois Segurança da Informação se faz com tecnologia, processos e pessoas, e a formação destas exige mais que uma sequência de treinamentos.

## REFERÊNCIAS

- ABDEL-AZIZ, A. **Intrusion detection & response leveraging next generation firewall technology**. GCIA Gold Certification. Practical Assignment, SANS Institute, Feb. 2009, Disponível em: < <http://www.sans.org/reading-room/whitepapers/firewalls/intrusion-detection-and-response-leveraging-next-generation-firewall-technology-33053>>. Acesso em: 02 Jan 2015.
- ANDERSON, J. P. **Computer security threat monitoring and surveillance**. Technical Report. James P. Anderson Co. Fort Washington, PA, apr. 1980.
- BEJTLICH, R. **The Tao of network security monitoring beyond intrusion detection**. Addison Wesley, 2004.
- BELL, D. E.; LAPADULA, L. J. **Secure computer systems: unified exposition and multics interpretation**. Technical Report ESD-TR-75-306, The Mitre Corporation, Bedford, Air Force Systems Command, Hanscom Field, Bedford, Março 1975.
- BERLINER, B.; PRISMA, I. CVS II: Parallelizing software development. In Proceedings of the **USENIX Winter 1990 Technical Conference**, volume 341, page 352.
- BISSETT, T. D.; FIORENTINO, R. D.; GLORIOSO, R. M.; MCCAULEY, J. D.; MCCOLLUM, D. T.; TREMBLAY, G. A. T.; TROIANI, M. **Fault resilient/fault tolerant computing**. United States Patent 6038685, May 2000.
- CAVALCANTE, G. D., Tese de Mestrado. **Detecção e recuperação de intrusão com uso de controle de versão**. IC-UNICAMP, Campinas, Junho 2010, Disponível em: <<http://www.las.ic.unicamp.br/paulo/teses/20100505-Gabriel.Dieterich.Cavalcante-Deteccao.e.recuperacao.de.intrusao.com.uso.de.controle.de.versao.pdf>>. Acesso em 22 Set 2013.

CHESWICK, W.R.; BELOVIN, S.M.; RUBIN, A.D. **Firewalls e segurança na Internet: repelindo o hacker ardiloso**. 2.ed. Porto Alegre: Bookman, 2005.

COSTA, N. S., Tese de Doutorado. **Proteção de sistemas elétricos considerando aspectos de segurança da rede de comunicação**. EESC-USP, São Carlos, Abril 2007, Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18133/tde-28082007-155730/publico/Nilson.pdf>>. Acesso em: 14 Jul 2014.

DASGUPTA, D. (Ed.). **Artificial immune systems and their applications**. London: Springer, 1998.

DE CASTRO, L. N., Tese de Doutorado. **Engenharia imunológica: desenvolvimento e aplicação de ferramentas computacionais inspiradas em sistemas imunológicos artificiais**. FEEC-UNICAMP, Campinas, Maio 2001, Disponível em: <<http://www.bibliotecadigital.unicamp.br/document/?view=vtls000220201>>. Acesso em: 22 Ago 2013.

DE CASTRO, L. N.; TIMMIS, J. Artificial immune systems: A novel paradigm to pattern recognition. In: CORCHADO, J. M.; ALONSO, L.; FYFE, C.(Ed.). **Artificial Neural Networks in Pattern Recognition**. Paisley(UK): University of Paisley, 2002. p.67–84.

DE CASTRO, L. N.; HRUSCHKA, E. R.; ROSATELLI, M. C.; CAMPELLO, R. J. G. B. **Computação natural: uma breve visão geral**. In Workshop em nanotecnologia e Computação Inspirada na Biologia, 2004.

DE PAULA, F. S., Tese de Doutorado. **Uma arquitetura de segurança computacional inspirada no sistema imunológico**. IC – UNICAMP, Campinas, Junho 2004, Disponível em: <<http://www.las.ic.unicamp.br/paulo/teses/20040713-PhD-Fabricio.Sergio.de.Paula-Uma.arquitetura.de.seguranca.computacional.inspirada.no.sistema.imunologico.pdf>>. Acesso em: 17 Out. 2014.

FARMER, J. D.; PACKARD, N.; PERELSON, A. **The immune system, adaptation and machine learning**. Physica D, v.22, p.187–204, 1986.

FORREST, S.; HOFMEYR, S. A.; SOMAYAJI, A. **Computer immunology**. Communications of the ACM, v.40, n.10, p.88–96, 1997. Disponível em: <<http://citeseer.ist.psu.edu/forrest96computer.html>>. Acesso em 16 Jan 2013.

FRASER, B. **RFC 2196 – Site security handbook**. Disponível em: <<http://www.faqs.org/rfcs/rfc2196.html>>. Setembro, 1997. Acesso em: 14 Fev 2014.

GIFT, N.; SHAND, A. **Introduction to distributed version control systems**. IBM Technical library, 2009, Disponível em: <[https://www.ibm.com/developerworks/aix/library/au-dist\\_ver\\_control/](https://www.ibm.com/developerworks/aix/library/au-dist_ver_control/)>. Acesso em: 20 Set 2013.

GUZELLA, T. S.; UCHÔA, J. Q.; SANTOS, T. A. M.; CAMINHAS, W. M. **Proposta de um Modelo de Classificação de Padrões Baseado no Sistema Imune: uma Aplicação para a Identificação de SPAM**. In: UFRN/SBRN.CBRN 2005-VII Congresso Brasileiro de Redes Neurais. Natal, 2005.v.1.

JONES, M. T. **Anatomia do security-enhanced linux (SELINUX)**, 2008, Disponível em: <<http://www.ibm.com/developerworks/br/library/l-selinux/>>. Acesso em: 20 Set 2013.

KENDALL, K. Dissertação de Mestrado. **A Database of computer attacks for the evaluation of intrusion detection systems**. Massachusetts Institute of Technology, Massachusetts, June 1999, Disponível em: <[http://wenke.gtisc.gatech.edu/ids-readings/attack\\_dbase\\_kkendall\\_thesis.pdf](http://wenke.gtisc.gatech.edu/ids-readings/attack_dbase_kkendall_thesis.pdf)>.

MELO, S.; DOMINGOS, C.; CORREIA, L.; MARUYAMA, T. **BS7799: Da tática à prática em servidores linux**. Rio de Janeiro: Editora Alta Books, 2006. 232p.

NAKAMURA, E. T.; GEUS, P. L. **Segurança de redes: em ambientes cooperativos**. 2. ed. São Paulo: Futura, 2003.



SCAMBRAY, J.; MCCLURE, S.; KURTZ, G. **Hackers expostos: segredos e soluções para a segurança de redes**. 2 ed. São Paulo: Makron Books, 2001.

SIGHT, K. **IT Infrastructure security-step by step**. InfoSec Reading Room, SANS Institute 2001, Disponível em: <<http://www.sans.org/reading-room/whitepapers/firewalls/intrusion-detection-and-response-leveraging-next-generation-firewall-technology-33053>>. Acesso em: 02 Jan 2015.

TAVALLAEE, M.; BAGHERI, E.; LU, W.; GHORBANI; A. A. **A Detailed Analysis of the KDD CUP 99 Data Set**. Proceeding of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications. (CISDA), 2009.

UCHÔA, J. Q., **Segurança computacional**, Lavras: UFLA/FAEPE, 2005, Curso de Pós Graduação “Latu Sensu” (Especialização) a Distância em Administração em Redes Linux.

UCHÔA, J. Q., Tese de Doutorado, **Algoritmos imunoinspirados aplicados em segurança computacional: utilização de algoritmos inspirados no sistema imune para detecção de intrusos em redes de computadores**, ICB-UFMG, Maio 2009, Disponível em: < <http://www.pgbioinfo.icb.ufmg.br/defesas/28D.PDF> >. Acesso em: 5 Set. 2014.

UNTERLEITNER, M. C., **Computer immune system for intrusion and virus detection** - Adaptive Detection Mechanisms and their Implementation, VDM Verlag Saarbrücken, Germany, 2008

WANG, J. **Computer network security: Theory and Practice**. Higher Education Press, Beijing and Springer-Verlag GmbH Berlin Heidelberg, 2009.

WATSON, A; BENN, P; YODER, A.G. **Multiprotocol data access: Nfs, cifs and http**. Technical report, Technical Report TR3014 Network Appliance, 1999.

WEBER, T. S. **Tolerância a falhas: conceitos e exemplos**. PPGC- UFRGS, Porto Alegre, 2001, Disponível em:  
<<http://www.inf.ufrgs.br/~taisy/disciplinas/textos/ConceitosDependabilidade.PDF>>.

YOUNG, G.; KAVANAGH, K. M.; PESCATORE, J.; HILS, A.; NICOLETT, M.; MACDONALD, N.; WHEATMAN, V.; FIRSTBROOK, P.; WHEATMAN, J.; GIRARD, J.; ORANS, L.; WAGNER, R.; FEIMAN, J; KENNEY, L. F.; PERKINS, E. **Next-Generation firewalls. hype cycle for infrastructure protection**. G00161383, Gartner Group, 2008, Disponível em: <<https://www.gartner.com/doc/761516>>. Acesso em: 12 Fev 2015.

YOUNG, G.; PESCATORE, J. **Next-Generation firewalls. magic quadrant for enterprise network firewalls**. G00162592, Gartner Group, 2008, Disponível em: <<https://www.gartner.com/doc/810612/magic-quadrant-enterprise-network-firewalls>>. Acesso em: 12 Fev 2015.

ZUBEN, F. J. V.; ATTUX, R. R. F. **Rede neural de Kohonen e aprendizado não-supervisionado**. 2007. Disponível em:  
<[ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia353\\_1s07/topico8\\_07.pdf](ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia353_1s07/topico8_07.pdf)>. Acesso em: 5 Abr. 2013.

ZUCCO, J. C., **Hardening linux usando controle de acesso mandatório**, UCS, Caxias do Sul, Fevereiro 2008, Disponível em:  
<<http://www.seer.ufrgs.br/testeCPD/issue/viewFile/487/6>>. Acesso em 14 Jul 2014.