Universidade de Pernambuco
Escola Politécnica de Pernambuco
Departamento de Sistemas e Computação
Programa de Pós-Graduação em Engenharia da Computação

Moisés Danziger

Sistema Híbrido de Detecção de Intrusão em Redes IEEE 802.11 Baseado na Teoria do Perigo e Sistemas Multi-agentes (MAWIDS-DT)

Dissertação de Mestrado

Recife, Março de 2010

UNIVERSIDADE DE PERNAMBUCO
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO

MOISÉS DANZIGER

Sistema Híbrido de Detecção de Intrusão em Redes IEEE 802.11 Baseado na Teoria do Perigo e Sistemas Multi-agentes (MAWIDS-DT)

Dissertação apresentada como requisito parcial
para a obtenção do grau de Mestre em
Engenharia da Computação

Prof. Dr. Fernando Buarque de Lima Neto
Orientador
Prof. Dr. Renato Mariz de Moraes
Co-orientador

Recife, Março de 2010.

M.199.s Danziger, Moisés

Sistema Híbrido de Detecção de Intrusão em Redes IEEE 802.11 Baseado na Teoria do Perigo e Sistemas Multi-agentes, 209 – Recife - PE : O Autor, 2010.

210 f. : 101 fig., 26 tab.

Dissertação (mestrado) – Universidade de Pernambuco. DSC Engenharia da Computação, 2010.

Inclui bibliografia e apêndice.

1. Redes IEEE 802.11. 2. Sistemas Multi-agentes – Aplicação. 3. Sistemas Imunológicos Artificiais – Teoria do Perigo. 4. Classificador Bayesiano. 5. Detecção de Intrusão.

CDU: 681.51:007.52

Pois que aproveita ao homem ganhar o mundo inteiro, se perder a sua alma? Ou que dará o homem em recompensa da sua alma? (Mateus 16: 26)

Agradecimentos

Agradeço a Deus em primeiro lugar por sua misericórdia, paciência e bondade para comigo, um ser pequeno e frágil perante Seu enorme poder. Agradeço também ao Prof. PhD. Fernando Buarque de Lima Neto, por acreditar e me orientar nestes dois anos em busca dos meus primeiros passos em ciência. Obrigado pela paciência e dedicação.

Agradecimentos ao Prof. PhD. Renato Mariz de Moraes pelas sugestões, orientações e direcionamento sobre o assunto Redes de Computadores, principalmente as redes IEEE 802.11, assim como ao Prof. Dr. Carmelo Bastos Filho, pelos incentivos e apoio na parte matemática.

A todos os professores do Departamento de Sistemas e Computação da UPE – Escola Politécnica de Pernambuco, que, direta ou indiretamente me instigaram a fazer pesquisa com qualidade e dedicação.

Agradeço de forma especial aos amigos que compartilharam das dificuldades para atingir esse sonho. Amigos como Salomão Madeiro, Robson Santana e Marcelo Teixeira que sempre me ajudaram quando precisei. Ao amigo Marcelo Lacerda pela ajuda no desenvolvimento do projeto. A todos os meus colegas do Programa de Mestrado em Engenharia da Computação. Agradeço, de forma especial, a querida Georgina, que sempre ajudou nas questões burocráticas.

A minha querida esposa Bruna, que me incentivou, apoiou e ajudou nos momentos mais difíceis. Pelo sacrifício de muitos momentos em prol da minha realização pessoal. Obrigado por ser minha esposa e acreditar em mim.

Aos meus pais, que representam o início de tudo isso, sem os quais eu não teria alcançado mais este objetivo. Vocês são meus heróis de verdade.

Resumo

Recentemente, as redes sem fio, definidas pelo padrão IEEE 802.11, vêm crescendo e ganhando cada vez mais adeptos pelo mundo. Esse novo ambiente trouxe ainda mais preocupações quanto à segurança, principalmente pelo fato de que as limitações existentes no ambiente cabeado (i.e. limites físicos) deixaram de existir e, com isso, novas possibilidades de ataque emergiram, trazendo desconforto aos usuários e dificuldades aos administradores.

De certa forma, as ferramentas de detecção de intrusão voltadas para redes IEEE 802.11, conhecidas como WIDS (*Wireless Intrusion Detection System*), ainda possuem muitas vulnerabilidades, pois estão em suas versões iniciais. Além disso, como nos ambientes cabeados, elas também possuem as mesmas dificuldades básicas para (1) automatização de rotinas e (2) identificação de novos tipos de ataques.

Uma ferramenta capaz de atuar a contento nos itens (1) e (2) necessita de habilidades muito maiores do que as oferecidas por computação clássica. Estudos recentes sobre a nova geração de sistemas imunológicos artificiais (*Artificial Immune Systems* – AIS) inspirados pela Teoria do Perigo (*Danger Theory* – DT) [126], por exemplo, têm apresentado bons resultados em problemas de detecção.

A abordagem proposta utiliza um modelo imuno-inspirado de detecção de intrusão em redes IEEE 802.11 e agentes inteligentes, organizados numa hierarquia modular que a torna capaz de atuar como sensores e atuadores (i.e. detecção e combate) para execução automatizada do processo de detecção, identificação e combate dos eventos anômalos espalhados pelo ambiente a ser monitorado.

Resultados obtidos através de vários experimentos em ambientes reais mostraram que a nova geração de AIS apresenta atributos bem qualificados para a sua aplicação em WIDS. A capacidade de identificação de novos ataques foi testada; utilizou-se um classificador bayesiano, que demonstrou condições suficientes para realizar a tarefa sem prejudicar o processo de execução do WIDS.

Palavras Chaves: Sistemas de Detecção de Intrusão, Sistemas Imunológicos Artificiais, Sistemas Multi-agentes, Redes IEEE 802.11, Teoria do Perigo e Automatização.

Abstract

Recently, wireless networks, defined by the IEEE 802.11 standard have experienced a marked grow. It has gathered a vast number of users and fervorous supporters around the world. This new environment has brought further safety concerns, mainly because of the wired environment limitations itself. In other words, physical limitations no longer exist, and so new opportunities for the attacks emerged. These problems bring stress to users and difficulty to administrators.

In a way, the tools of intrusion detection aimed at IEEE 802.11, known as WIDS (Wireless Intrusion Detection System), still have many vulnerabilities, mainly because they are in their initial versions. In addition to operating in the challenging wired environment, they also have to deal with old problems such as (1) need of automation of administrative tasks and (2) identification of new types of attacks.

A suitable tool capable of satisfying (1) and (2) requires abilities greater than those offered by the classical computing. Recent studies on new generation of artificial immune systems (AIS), inspired by the Danger Theory (DT) [126], for example, have been successfully used in detection problems.

The proposed approach uses an immune-inspired model of intrusion detection for IEEE 802.11 networks and intelligent agents, organized in a modular hierarchy. This makes it able to act as sensors and actuators (i.e. detection and fighting) for the implementation of automated detection, identification and combating of anomalous events spread out in the environment to be monitored.

Results from several experiments in real environments show that the new generation of AIS has well qualified attributes for its application in WIDS. The ability to identify new attacks was tested; a Bayesian classifier was used for this function and its demonstrated sufficient conditions to accomplish the task without damaging the process during the execution of WIDS.

Keywords: Intrusion Detection Systems, Artificial Immune Systems, Multi-Agents, IEEE 802.11 Networks, Danger Theory and Automation.

Índice

Índice de Figuras

Índice de Tabelas

Tabela de Acrônimos e Siglas

Abas	***Agente Básico***
Asub	***Agente Subalterno***
Aint	***Agente Intermediário***
Asup	***Agente Superior***
Amen	***Agente Mensageiro***
Alog	***Agente Logger***
ACK	***Acnowledgement***
AES	***Advanced Encryption Standard***
AI	***Artificial Intelligence***
AIS	***Artificial Immune System***
AP	***Access Point***
ARP	***Address Resolution Protocol***
CI	***Computing Intelligent***
CID	***Confidencialidade, Integridade e Disponibilidade***
CSM	***Co-stimulatory Molecules***
DC	***Dendritic Cell***
DCA	***Dendritic Cell Algorithm***
DoS	***Denial of Service***
DNS	***Domain Name System***
DS	***Danger Signal***
DT	***Danger Theory***
EAP	***Extensible Authentication Protocol***
FIPA	***Foundation for Intelligent Physical Agents***
HIS	***Human Immune System***
IAD	***Inteligência Artificial Distribuída***
ICV	***Integrity Check Value***
IDS	***Intrusion Detection System***
IE	***Immune Engineering***
IS	***Immune System***
ISO	***International Standards Organization***

IV	***Initialization Vector***
JADE	***Java Agent Development Framework***
JDK	***Java Development Kit***
JRE	***Java Runtime Environment***
JVM	***Java Virtual Machine***
KQML	***Knowledge Query and Manipulation Language***
LLC	***Logical Link Control***
MAC	***Media Access Control***
MAS	***Multi-agents System***
MCAV	***Mature Context Antigen Value***
MIC	***Message Integrity Check***
MITM	***Men in the Middle***
OSI	***Open System Interconnection***
PAMP	***Pathogen-associated Molecular Pattern***
PRGA	***Pseudo-Random Generation Algorithm***
PSK	***Pre Shared Key***
RADIUS	***Remote Authentication Dial in User Service***
RC4	***Ron´s Code #4***
RDP	***Resolução Distribuída de Problemas***
SMAC	**Sistema Multiagente Cognitivo**
SMAR	**Sistema Multiagente Reativo**
SS	***Safe Signal***
SSID	***Service Set Identifier***
TKIP	***Temporal Key IntegrityProtocol***
WIDS	***Wireless Intrusion Detection System***
WEP	***Wired Equivalent Privacy***
WPA	***Wi-Fi Protection Access***

Capítulo 1

Este capítulo é dedicado ao esclarecimento dos principais pontos abordados nesta pesquisa, principalmente aqueles relacionados à motivação, ao enfoque, aos objetivos e às contribuições desta dissertação. No decorrer deste capítulo, é apresentada a estrutura da dissertação para efeito de entendimento e a navegabilidade no texto.

1.1 Segurança em Redes

As redes (incluindo redes de computadores) são impressionantes vias de duas mãos. Enquanto uma caminha cada vez mais ao encontro de revolucionar a sociedade, promovendo meios em que as pessoas estarão conectadas o tempo todo com o mundo virtual, a outra surge assustadoramente na contramão, trazendo tudo o que há de problemas oriundos dos relacionamentos sociais. Roubo de informações particulares, invasão de privacidade (por exemplo, ataques a sites de relacionamento) e tentativas de extorsão (por exemplo, envio de *e-mails* com apelo emocional) são exemplos típicos de que os problemas do mundo real podem ser usados no mundo virtual.

Mesmo com tantas dificuldades, as redes se popularizam e os gerentes ou administradores, a cada dia, vêem seu trabalho aumentando, tanto em quantidade quanto em dificuldade. Para ajudá-los, muitos mecanismos de segurança foram propostos, destacando-se: (i) controle de acesso (por exemplo, autenticação de usuários), (ii) controle de tráfego (por exemplo, *Firewall* e *Proxy*), (iii) criptografia de dados e (iv) controle de eventos (por exemplo, análise de intrusão). Especificamente no mecanismo (iv), encontra-se um dos pilares das pesquisas que envolvem a segurança de redes, inclusive esta.

Até recentemente, as redes eram definidas como redes de computadores, porém, o termo “redes” tornou-se mais apropriado, principalmente pela agregação de novas tecnologias como VoIP[1], Sistemas Wi-Fi[2] e Sistemas Celulares[3]. Portanto, a análise de eventos ocorridos, ou que estão ocorrendo neste ambiente tão diversificado, traz importantes desafios para o desenvolvimento de ferramentas eficientes na detecção e combate dos problemas provocados por fatores estranhos às redes. Sistema de detecção de intrusão (IDS, do inglês, *Intrusion*

[1] Sistemas de voz para trafegar em redes operando com os protocolos TCP/IP
[2] Sistemas operando de acordo com o padrão IEEE 802.11, ou seja, comunicação sem fio
[3] A partir da 2,5 e 3ª geração

Detection System) é a definição para tais ferramentas, estas podem atuar de três formas: (1) sobre as redes, (2) sobre hosts e (3) sobre ambos.

Dois conjuntos de demandas encorajam as constantes pesquisas sobre IDS: (a) a detecção de novos tipos de ataques e (b) a execução automatizada. No primeiro, a dificuldade se apresenta devido à constante inovação e adaptação dos atacantes. No segundo, o problema principal se relaciona aos limites do binômio, autoridade e responsabilidade, pela utilização do sistema. Ou seja, até que ponto um sistema tem autonomia para atuar por si próprio. Trabalhos recentes apontam para o uso da inteligência computacional (*Computational Intelligence* – CI) e a inteligência artificial (*Artificial Intelligence* – AI) como direção para o desenvolvimento de IDSs mais eficientes na quebra desses dois conjuntos de demandas.

Padronizada pelo IEEE (do inglês, *Institute of Electrical and Electronics Engineers*), através do modelo 802.11, a despeito da grande comodidade proporcionada para os usuários, as redes IEEE 802.11 incrementaram as dificuldades dos IDS pelas deficiências técnicas que incluem: (i) falta de padronização dos fabricantes, (ii) uso de algoritmos de criptografia considerados fracos, (iii) falhas no protocolo (por exemplo, transmissão do cabeçalho dos pacotes de controle em puro texto), (iv) barreiras físicas (isto é, quando o sinal não consegue atingir seu alvo por alguma interferência, seja natural do ambiente ou provocado por terceiros), (v) capacidade do canal de transmissão e recepção e (vi) consumo de energia. Agregou-se também a esse legado de problemas a falta de conhecimento dos usuários, que muitas vezes beiram a irresponsabilidade – incluem-se aí os administradores pouco competentes – permitindo que vários tipos de ataques, muitos desconhecidos, outros adaptados das redes cabeadas, sejam disseminados. Assim, a segurança das redes sem fio tornou-se um ponto forte de pesquisas. Em 2004, o IEEE protocolou a mais nova versão, o modelo IEEE 802.11i[4], no qual inseriu várias correções e mudanças voltadas para a segurança.

Para atacar uma rede sem fio, a ação mais comum é colher dados, algo extremamente fácil – pois basta apenas estar localizada em área de alcance da rede para ser possível captar os sinais de rádio – e simples de realizar (por exemplo, usando ferramentas como NetStumbler[5] e Airsnort[6]). Este modelo de ataque é denominado ataque passivo, pois o atacante não injeta nenhuma informação (isto é, nenhum pacote é injetado na rede). Os IDS possuem muita dificuldade para detectar este tipo de ataque, principalmente porque o atacante

[4] Versão lançada com o intuito de corrigir os problemas do WEP. Ver mais em http://www.ieee.org
[5] Famoso analisador de redes sem fio. Encontrado em: http://www.netstumbler.com
[6] Ferramenta usada para escanear e quebrar criptografia baseada na WEP

está em modo promíscuo (isto é, a placa de rede coleta tudo o que está passando a seu alcance) sem inserir pacote algum na rede. Quando um atacante injeta pacotes na rede, está configurado um ataque ativo. Este trabalho apresenta seu foco nos ataques ativos. Dentre os ataques ativos, oito tipos de ataques foram estudados: (i) ataque de-authentication, (ii) ataque de injeção de pacotes ARP, (iii) ataque interativo de pacotes, (iv) ataque de fragmentação, (v) ataque ChopChop [126], (vi) ataque de autenticação falsa, (vii) ataque Cafe-Latte [162] e (viii) ataque Hirte [4]. Portanto, o foco principal foram os ataques realizados contra o protocolo WEP (*Wired Equivalient Privacy*). Porém, alguns aspectos dos modelos WPA (*Hi-Fi Protected Access*) e WPA2 foram evidenciados durante os experimentos.

Para um IDS, o ponto principal é detectar ataques e alertar o administrador. Porém, atualmente, ferramentas automatizadas, com capacidade de agir por si próprias tem sido o desejo dos pesquisadores. Desta forma, um dos objetivos principais deste trabalho foi a investigação da eficácia do modelo quando em execução automatizada. O grande interesse por ferramentas automatizadas tem ocorrido pela dificuldade encontrada pelo ser humano em ficar monitorando diuturnamente um sistema. Isso porque, em intervalos cada vez mais curtos, é possível a ocorrência de erros, como o esquecimento da leitura de um *e-mail* enviado pelo sistema, ação muito comum realizada pelos IDS quando encontram problemas.

A proposta desenvolvida neste trabalho pode ser vista também como um IPS (*Intrusion Protection System*), que possui rotinas automatizadas para defesa da rede. Porém, foi definido que a sigla IDS seria adequada para o trabalho, haja vista que a função principal do modelo é a detecção e identificação automatizada.

1.2 Sistemas Imunológicos Artificiais e a Teoria do Perigo

Observando o sistema imunológico humano (do inglês, *Human Immune System* – HIS), alguns pesquisadores [52] e [62] identificaram-no como um sistema bastante adaptável e apto para defender o corpo humano contra uma enormidade de possíveis intrusos (isto é, bactérias e vírus estão na ordem de 10^{16}). O grande interesse dos pesquisadores estava justamente voltado à capacidade de detectar e identificar um ataque mesmo sem ter nenhuma ou pouca informação anterior que pudesse ser útil. Como decorrência, várias pesquisas [78], [102], [198] foram publicadas simulando as principais características do HIS: (a) diversidade, (b) especificidade, (c) memória, (d) administração distribuída e (e) adaptabilidade.

Para simular um IDS imuno-inspirado, os pesquisadores utilizaram inicialmente o principal conceito, até então conhecido, de como o HIS detectava os intrusos: a distinção

entre próprio (*self*) e não próprio (*non-self*) [45]. Porém, descobriu-se que esse modelo tinha sérios problemas de escalabilidade para definir a complexidade que um sistema em rede podia apresentar [2]. Dessa forma, era preciso reavaliar os conceitos. Através da DT, encontraram uma "nova" forma pela qual o HIS detectava as células invasoras. Segundo esta teoria, as células dendríticas (*Dendritics Cells* – DC) [133] do HIS são responsáveis pela coleta de sinais que são captadas por seus sensores (isto é, análogo a micro-antenas). Então estas processam os sinais e, de acordo com o resultado, mudam seu estado levando até os linfócitos-T sinais para ativação ou supressão de um contra-ataque. Nesse caso, não é feita nenhuma definição anterior do que pertence, ou não, ao sistema. Alguns trabalhos [89], [90] e [6] utilizaram os conceitos da DT para problemas de segurança – em especial usando o algoritmo DCA (*Dentritic Cell Algorithm*) de Greensmith [80]. Porém, foram restringidos para alguns problemas específicos e sobre redes cabeadas.

Trabalhar com um padrão em constante evolução (isto é, o padrão IEEE 802.11), associado aos poucos estudos de técnicas inteligentes aplicados a esse ambiente, trouxe para esta pesquisa um grande desafio na aplicação dos conceitos teóricos. Por isso, dadas as características inerentes aos ataques, sua distribuição e diversidade nos motivou a instanciar um IDS imuno-inspirado embutido num MAS hierárquico.

Existem muitas ferramentas IDS desenvolvidas e em pleno uso por todo o mundo. Agumas, como o Snort[7] e a Symantec Norton[8], atuando sobre o nível de software e outras, como as utilizadas pela Cisco[9], atuando diretamente através do hardware e, às vezes, associadas com software. O modelo desenvolvido neste trabalho não leva em consideração todos os requisitos para um IDS no nível das ferramentas citadas, mas serviu para realizar experimentos na intenção de validar o modelo de detecção de intrusão proposto. Por objetividade, a camada de enlace do modelo OSI (*Open System Interconnection*) foi escolhida para aplicação do modelo desenvolvido. Assim, não foram tratados pacotes, mas quadros, durante os experimentos.

1.3 Objetivos, Metodologia e Contribuições

O objetivo principal deste trabalho foi (i) a conceituação de uma abordagem híbrida imuno-inspirada embutida em um sistema multi-agente hierárquico e (ii) o desenvolvimento de uma

[7] Um famoso IDS que pode ser encontrado em: http://www.snort.org
[8] A Symantec possui um conjunto de ferramentas para segurança em redes, inclusive para detecção. Para saber mais veja: http://www.symantec.com
[9] Empresa de fabricação de equipamentos e soluções para redes e telecomunicações. http://www.cisco.com

ferramenta computacional inteligente que fosse capaz de identificar intrusões – inclusive desconhecidas – e tomar decisões de contra-ataque, além de auditar o sistema de forma automatizada. Para isso, houve a necessidade de entendimento da DT, dos conceitos dos MAS e, conseqüentemente, dos agentes inteligentes, do padrão IEEE 802.11 e das questões de segurança envolvidas e, por fim, dos sistemas de detecção de intrusão e sua aplicação sobre o ambiente sem fio.

Para alcançar os objetivos “i” e “ii”, foi estabelecido um planejamento que incorporou as seguintes etapas metodológicas: (a) estudo e revisão das teorias envolvidas, (b) analogia entre as teorias e os problemas de detecção de intrusão para redes IEEE 802.11, (c) concepção e desenvolvimento de uma ferramenta aplicando os conceitos imunológicos sobre os agentes inteligentes, adaptando-os para a execução sobre as redes sem fio e (iv) validação do modelo proposto através de experimentos pré-configurados para redes sem fio.

Na parte de inteligência computacional do sistema – a saber, a identificação dos ataques desconhecidos – algumas técnicas foram estudadas e a escolha, explicada na apresentação do modelo, foi para os classificadores Bayesianos, especificamente a técnica *naïve Bayes.*

Mais especificamente, este trabalho investigou a aplicação de duas técnicas – em especial a DT (do inglês, *Danger Theory*) do Sistema Imunológico (*Immune System* – IS) pertencente à CI, e os MAS, Sistemas Multiagentes (do inglês, *Multiagent Systems*) pertencente a AI – ao problema da detecção e identificação de novos ataques (inclusive de novos tipos) e a automatização das rotinas de execução (com a tomada autônoma de decisão pelo próprio sistema em relação ao ataque).

O processo de validação foi dividido em três fases: (1) experimento para análise da eficiência do modelo de detecção usando a DT, (2) experimentos para análise do modelo de identificação usando *naïve Bayes* e (3) experimentos para teste do modelo como um todo, principalmente para visualização da capacidade de execução automatizada.

Os resultados e contribuições desta pesquisa foram:

1. Aplicação de modelos híbridos usando técnicas distintas para detecção de intrusão em redes IEEE 802.11;
2. Aplicação e adaptação do campo de atuação da DT sobre IDS;
3. Emprego de agentes inteligentes distribuídos hierarquicamente;
4. Desenvolvimento de um protótipo de IDS imuno-inspirado embutido sobre um MAS hierárquico para detecção, identificação e combate de intrusos em tempo real e automatizado;

5. Validação junto a uma rede IEEE 802.11, quanto à detecção, identificação e combate realizados de forma automatizada.

1.4 Estrutura do Texto

O texto está organizado em sete capítulos. O Capítulo 2 apresenta a fundamentação teórica, abordando os temas de segurança de redes e sistemas, assim como os conceitos de redes sem fio e IDS. No Capítulo 3, foi realizada uma breve revisão do HIS, dando ênfase à DT, fonte de inspiração desta pesquisa, e à descrição das principais características dos AIS. Os conceitos dos agentes inteligentes e MAS (reativos e cognitivos) são mostrados no Capítulo 4. O Capítulo 5 traz a modelagem para o WIDS proposto e os detalhes envolvidos no desenvolvimento. Os experimentos e os resultados encontrados são detalhados no Capítulo 6 e, no Capítulo 7, são apresentadas conclusões baseadas nos resultados e nas observações feitas durante a pesquisa, além das sugestões para trabalhos futuros.

1.5 Sugestões de Leitura

O *background* é parte fundamental de uma pesquisa. É através dele que são expostos os pontos importantes da teoria utilizada para conceber o modelo proposto. Mediante tal importância, para esta dissertação foram necessários três longos capítulos. De acordo com esse fato, é recomendável que, para os leitores que já possuem conhecimento sobre redes e segurança (por exemplo, administradores de redes) atentem-se aos Capítulos 3 e 4 e, para aqueles que detenham conhecimentos sobre inteligência computacional e/ou inteligência artificial atentem-se ao Capítulo 2.

Capítulo 2

Fundamentos Teóricos

Neste capítulo são mostrados (i) alguns conceitos sobre segurança de sistemas e redes, (ii) explicações sobre o padrão IEEE 802.11, focando os mecanismos de segurança e os riscos que os envolvem, (iii) detalhamento dos conceitos de detecção de intrusão, (iv) adaptação para redes sem fio, e (v) apresentação de algumas ferramentas usadas para ataque contra as redes sem fio.

Estes itens fazem parte da fundamentação teórica necessária para o entendimento deste trabalho.

2.1 A Segurança no Contexto das Redes

Segundo Gramp e Morris [86], é fácil gerenciar um sistema de segurança computacional: basta que sejam desligadas todas as conexões ao mundo externo, colocar a máquina em uma sala blindada e um posto de guarda na porta. Certamente que esta não é a solução desejada e nem existente, pois, um sistema computacional só tem serventia se puder (i) prestar serviços que apoiarão as pessoas nas suas atividades (ii) proporcionar conforto, agilidade, facilidade, entre outros fatores que dificilmente estarão presentes em uma sala blindada.

A sociedade moderna está cada vez mais conectada e, através das redes, dois usuários, ou mais, podem trocar informações de qualquer parte do globo terrestre, desde que tenham acesso a elas. Portanto, esse é um caminho sem volta, os usuários não abrirão mão das facilidades e benefícios de ter sistemas conectados por redes. Não há outra saída, aos sistemas de segurança foi imposto o desafio de acompanhar o desenvolvimento tecnológico dos sistemas e suas conexões e, aos usuários, o sacrifício da perda da segurança plena em prol da usabilidade. O segredo para alcançar um bom nível de segurança, neste caso, está no equilíbrio entre os dois lados.

Para Kruegel et al. [127], quando um sistema de computador (e outros equipamentos) está ligado através de uma rede, três problemas principais podem ocorrer. O primeiro caso é o aumento dos pontos que podem servir de fonte para um atacante realizar seu objetivo. O segundo é o aumento do perímetro físico do sistema onde, geralmente, ao transmitir uma informação pela rede, o receptor não tem controle sobre os vários dispositivos pelos quais os

pacotes passam durante sua viagem. Com a Internet, ficou ainda mais difícil garantir a origem do pacote. Em terceiro lugar, o número de serviços que um sistema em rede pode oferecer pode ser maior que um sistema autônomo, inclusive seu sistema de autenticação. Geralmente, os processos de serviços desenvolvidos para acesso remoto podem ter erros de programação e/ou erros de configuração que podem ser explorados e, ocasionalmente, comprometer a segurança.

Além dos problemas citados, outros, agregados ao meio físico pelo qual são montadas as redes, apresentam maiores ou menores vulnerabilidades. Como exemplo, as redes montadas sobre o padrão IEEE 802.11 possuem muitas deficiências oriundas da própria falha na padronização do modelo [161], outras por erros de configuração [95], através das quais inúmeros ataques têm ocorrido com relativa facilidade [39].

Portanto, vários mecanismos de segurança para inibir, coibir e até mesmo punir ataques aos sistemas conectados em rede foram criados e continuam evoluindo. São divididos em duas frentes: (i) físicos e (ii) lógicos. No primeiro caso, os mecanismos se apóiam em controles físicos como salas reservadas, portas fechadas, guardas e blindagem. Por outro lado, os controles lógicos são geralmente agregados a meios eletrônicos e possuem vários mecanismos que os apóiam como: (a) filtragem de tráfego (por exemplo, *Firewalls*); (b) criptografia de dados para dificultar o entendimento das informações caso sejam roubadas; (c) serviços de autenticação; (iv) tunelamento de pacotes (por exemplo, *Virtual Private Network – VPN*); e (v) certificação digital. Essas ferramentas, quase sempre estão presentes nas grandes redes – o que nem sempre acontece nas pequenas [128], permitindo que muitos atacantes usem esses sistemas como “sistemas hospedeiros” para ataques camuflados. Porém, esses mecanismos não se mostraram suficientes para garantir a segurança, e, dessa forma, foram desenvolvidos os sistemas de detecção de intrusão como ferramenta de auditoria do sistema.

Em qualquer sistema em que haja necessidade de proteger informações, sejam armazenadas ou em tráfego pela rede, é importante a presença de um ou mais IDS. Atualmente, com as várias tecnologias sendo associadas às redes de computadores, é um grande desafio implantar um IDS capaz de monitorar sistemas tão heterogêneos. Assim, novos paradigmas estão sendo aplicados para o aperfeiçoamento e adaptação às novas tecnologias.

2.2 Conceitos de Segurança

O conceito de segurança da informação foi padronizado pela norma ISO/IEC[10] 17799:2005 (atualmente a série de normas ISO/IEC 27000 foi reservada para o tratamento da segurança da informação e a ISO/IEC 27002 é considerada formalmente como a versão 17799:2005 para fins históricos), sendo definida da seguinte maneira: "*Segurança da Informação é a proteção da Informação de diversos tipos de ameaças para garantir a continuidade dos negócios, minimizar os danos aos negócios e maximizar o retorno dos investimentos e as oportunidades de negócio.*"

A segurança da informação está diretamente relacionada (i) à confidencialidade, (ii) à integridade e (iii) à disponibilidade de um conjunto de dados de um indivíduo ou de uma organização. Estes três termos formam o acrônimo CID e são considerados os pilares para a análise, o planejamento e a implantação do processo de segurança para um conjunto de informações que se deseja proteger [39]. Em "i" os dados são considerados confidenciais se permanecem obscuros a todos e liberados apenas para quem tem direito. Em "ii", os dados possuem integridade se permanecerem idênticos ao seu estado após serem usados pelo último usuário autorizado. E no caso "iii", os dados têm disponibilidade quando são acessíveis para usuários autorizados através de um formato conveniente e dentro de um prazo razoável [206]. Não se pode confundir segurança da informação com segurança computacional apenas. Na verdade, todos os aspectos relacionados à segurança de alguma informação devem ser levados em consideração (por exemplo, o ambiente onde estão alocadas e a infra-estrutura sobre a qual estão assentadas).

Os objetivos dos atacantes são muitos[11], mas é possível afirmar que estão amarrados aos três pilares da segurança. Se um ataque visa o roubo da informação, está atrelado à questão da confidencialidade. Caso tente danificar a informação, o problema é da integridade. E finalmente, caso queira prejudicar o acesso à informação, a disponibilidade foi afetada. Cada um desses objetivos tem seu valor agregado ao nível de exigência sobre a informação (isto é, um sistema bancário pode precisar de um nível de segurança muito mais alto que um simples provedor de Internet. Nesse último caso, a CID seria completamente utilizada num nível elevado. Mas no caso do provedor, seu nível de segurança poderá ser maior sobre a

[10] International Organization for Standardization (http://www.iso.org) and International Electrotechnical Commission (http://www.iec.ch)
[11] Mais informação sobre os muitos objetivos de ataques em: http://32.cert.br/download/32-01-conceitos.pdf. Último acesso dia 14/03/2010

disponibilidade). Portanto, um sistema de segurança geralmente é planejado de acordo com a regra do negócio, ou seja, de acordo com o foco principal do negócio [206].

Num segundo plano dentro do acrônimo CID, existem quatro outros termos importantes: (i) identificação, (ii) autenticação, (iii) autorização e (iv) responsabilidade. Em qualquer sistema de segurança, é necessário identificar quem é o usuário, obter garantias de que é ele mesmo, permitir ou não o uso da informação àquele usuário e responsabilizá-lo pelas suas ações [130]. Sobre cada um desses itens, existe um ou mais mecanismos de segurança associados, inclusive um mesmo mecanismo pode estar associado a mais de um item – caso das ferramentas de análise e monitoramento.

Um ataque ou intrusão (o termo ataque é, muitas vezes, associado ao termo intrusão) pode ter duas fontes principais de origem: (a) interna e (b) externa. É comum o pensamento de que um intruso sempre está fora dos domínios do sistema. Mas, segundo os vários órgãos que monitoram incidentes relacionados à segurança (por exemplo, o Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil – CERT-br[12]) [39], a maioria dos ataques tem ocorrido internamente.

Segundo a RFC 2196[13] (*Site Security Handbook*), para que seja possível aplicar os conceitos de segurança da informação é preciso saber quais são os objetivos da segurança. Se não souber qual objetivo, dificilmente o uso das ferramentas de segurança terá sucesso. Afinal, sem saber o que verificar e quais restrições impor, o uso de um sistema de segurança da informação não teria nenhum sentido. Portanto, os conceitos de segurança da informação devem estar embutidos dentro de uma política de segurança, na qual, aprovada e implantada pelos administradores de sistemas e redes, será a linha mestra para a obtenção de um sistema "seguro", seja qual for o ambiente onde estiver aplicada.

2.3 O Padrão IEEE 802.11

O padrão IEEE 802.11 [105][14] foi apresentado em 1997 como modelo de referência para redes sem fio e suas especificações, e abrange as camadas físicas e de enlace referentes ao modelo OSI (*Open Systems Interconnection*). De acordo com o padrão, a camada física refere-se ao meio pelo qual são trocadas as informações, que neste caso são as ondas eletromagnéticas de

[12] Unidade brasileira para o Centro de Pesquisa e Desenvolvimento mantido pelo governo dos Estados Unidos da América e operado pela universidade Carnegie Mellon. Página principal: http://www.cert.br/
[13] Acrônimo para Request for Comments (um conjunto de documentos com informações técnicas detalhadas sobre protocolos da Internet que servem de referência para fabricantes, usuários e pesquisadores). Página principal: http://tools.ietf.org
14 Mais informações pode ser encontradas em: http://www.ieee802.org/11/

radiofreqüência, e a camada de enlace define o método de acesso ao meio. O método de acesso ao meio é conhecido como CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*), que é semelhante ao modelo CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*), porém, tem seu foco na prevenção da colisão de pacotes e não na detecção da colisão [159]. Segundo Kurose e Ross [129], para detectar colisões é necessário ter alta capacidade de enviar e receber ao mesmo tempo. Dessa forma, para as redes sem fio, a potência do sinal recebido normalmente é menor se comparada com a potência do sinal transmitido no adaptador IEEE 802.11, e, para este caso, é muito caro construir um hardware que tenha capacidade de detectar colisões. Outro fator preponderante é que, mesmo se pudesse construir um adaptador com essas características, ainda assim ele não seria capaz de detectar todas as colisões devido ao problema da estação oculta e do desvanecimento (isto é, quando o sinal sofre atenuação de percurso por alguma interferência).

Por definição [67], as redes sem fio podem ser arranjadas em uma destas três configurações lógicas:

- **Ponto-a-ponto:** geralmente é usado entre dois pontos distantes (exemplo, duas antenas) que possuem linha de visada (isto é, as duas antenas podem ser alcançadas diretamente sem obstáculos). Podem ter alta capacidade de vazão (isto é, taxa de transferência) e são muito usadas, por exemplo, para transmissão de áudio e vídeo;
- **Ponto-multiponto:** a mais comum entre as redes sem fio. Sempre existirá um elemento principal conhecido como ponto de acesso (*Access Point* - AP) e vários outros elementos secundários conectados a ele. Para ser possível a sua operação, foram definidos os pacotes de controle e gerenciamento. Existem limites técnicos quanto à distância e a quantidade de usuários conectados (isto é, diferentemente do modelo de transmissão das rádios FM, que apenas espalham o sinal para os receptores, nas redes sem fio as antenas são bidirecionais). Porém, é possível aumentar a área de cobertura inserindo vários APs interconectados;
- **Multiponto-multiponto:** conhecidas como redes Ad-Hoc ou redes Mesh[15]. Neste caso, não há um elemento central, sendo que cada nó se comunica, direta ou indiretamente, com o outro. Uma vantagem desse tipo de rede é que mesmo se um nó não estiver dentro de uma célula (isto é, nome dado ao perímetro no qual o sinal da antena alcança), ele pode comunicar-se por meio de outro nó. Outra vantagem

[15] Para saber mais ver http://www.teleco.com.br/tutoriais/tutorialwmn/pagina_1.asp

está relacionada ao compartilhamento (por exemplo, numa rede Mesh, se um nó tem acesso à Internet, pode compartilhar com todos os outros). Porém, há também muitos problemas, principalmente relacionados à alta complexidade de roteamento e identificação da vizinhança, e ao baixo desempenho [129].

Para poder se comunicar, dois ou mais equipamentos precisam estar compartilhando do mesmo canal físico [159]. Como o padrão IEEE 802.11 [105] usa ondas de rádio que se propagam no ar, existe ainda outro fator importante, a freqüência de operação. Neste caso, é preciso observar que uma freqüência diferente induz a um comprimento – da onda eletromagnética – diferente, provocando falha de comunicação. Como exemplo, o próprio modelo possui algumas subdivisões baseadas em diferentes faixas de freqüência de rádio no espectro (isto é, o espectro é o resultado obtido quando as radiações eletromagnéticas são emitidas nos seus comprimentos de onda ou freqüências correspondentes [159] [67]). Os modelos IEEE 802.11a (5 GHz), IEEE 802.11b e IEEE 802.11g (2,4 GHz), e IEEE 802.11n (pode operar na faixa 2,4 GHz ou 5 GHz), são exemplos da atuação em diferentes freqüências. Mas isso não é tudo, é preciso ainda que os dois equipamentos utilizem o mesmo canal (por exemplo, o padrão IEEE 802.11b utiliza 11 canais de 22 MHz, separados por um intervalo de 5 MHz).

Após a definição de um canal dentro de uma determinada freqüência, os dispositivos estão prontos para executarem suas atividades. Existem quatro modos de operação para os dispositivos baseados no padrão [105], [67]:

- **Master (ou estruturada):** usado para criar um serviço que se parece com um AP tradicional. Nesse modo, um dispositivo sem fio somente pode comunicar-se com outros dispositivos quando conectados em modo gerenciado. Quando em modo master, um dispositivo gerencia todas as rotinas de comunicação.
- **Gerenciado:** caso específico para os clientes de um dispositivo master. Aqui, os dispositivos se associam ao elemento principal e usam o mesmo canal. Não é possível a comunicação direta entre os nós nesse modo. Toda comunicação passa pelo master junto ao qual estão associados.
- **Ad-Hoc:** operando neste modo, um dispositivo pode comunicar-se diretamente com outro dispositivo que esteja em seu alcance. Para isso, precisam acordar um nome e um canal para operação.

- **Monitor:** muito usado para monitorar todo o tráfego em um determinado canal. Estando em modo monitor, um dispositivo não pode transmitir nenhum pacote. Geralmente, este modo de operação é usado por ferramentas de análise de problemas ou para descoberta de redes (também conhecido como ataque passivo).

Para este trabalho, é utilizado o modelo estruturado, ou seja, com a presença de um AP. Fato que exclui a necessidade de explicações detalhadas sobre o modelo de rede Ad-Hoc. Dessa forma, no próximo item serão apresentados os principais pontos do modo estruturado, segundo o padrão IEEE 802.11 [105].

2.3.1 O Modo de Infra-Estrutura

Para trabalhar em modo infra-estrutura, uma rede sem fio precisa ter um agente central, no caso o AP – que é também conhecido como estação-base central – e estações sem fio. Essa estrutura recebe o nome de BSS (*Basic Service Set*) ou conjunto básico de serviço. Podem existir várias BSS que se conectam através de um Hub, comutador ou roteador, nas quais, podem permitir a mobilidade da estação, sem sair da rede, através da formação de um DS (*Distribution Service*), um sistema de distribuição. É válido lembrar que a especificação não determina a forma como deve atuar o DS, apenas especifica os serviços que são de sua responsabilidade, o que permite a formação de redes mistas (isto é, a junção de redes cabeadas com redes sem fio) [105]. A capacidade de expansão provida pelo DS, quando ligado às BSSs, recebe o nome de ESS (*Extend Service Set*), conjunto de serviços estendidos. Também não existe nenhuma restrição na especificação quanto à disposição das BSSs. Por isso, uma ESS pode assumir várias configurações, sendo uma delas bastante comum – a saber, aquelas sobrepostas para facilitar a mobilidade das estações entre as BSSs [159].

Para identificação dos dispositivos sem fio, é usado o endereço MAC (*Media Access Control*) de seis bytes, tanto para o AP quanto para as estações. Em teoria, o endereço MAC é exclusivo. Porém, é conhecido que alguns sistemas (por exemplo, o Linux) permitem que seja trocado por outro qualquer [67]. Não é recomendado o uso dessa rotina sem que haja total conhecimento por parte de quem irá usar, pois pode gerar conflitos de endereços e erros de transmissão na rede. Eesse assunto é detalhado na seção de segurança posteriormente.

Para que haja comunicação, a estação sem fio precisa se associar com o AP. Existem alguns itens que ajudam nesse processo. O primeiro deles é a identificação do AP através do SSID (do inglês, *Service Set Identifier*) ou identificador de conjunto de serviços. O segundo é a definição do canal (conforme abordado na seção anterior). Desta forma, quando uma estação

entra no perímetro de alcance do sinal da rede, ela usa dois modelos de procura pelas redes que podem estar presentes naquele local: (a) modo passivo e (b) modo ativo. No primeiro caso, todos os canais são ouvidos pela estação através da captura dos quadros de anúncio (*beacon frames*) e também os quadros de respostas às requisições de sondagens (*probe response frames*). No segundo caso, a estação deixa de ser apenas ouvinte e transmite em modo *broadcast*[16] um quadro de requisição de sondagem (*probe request frame*) para que os APs possam responder. Note que somente o AP com o SSID correto, ou seja, aquele que foi passado pela estação, responderá à estação. Esse processo economiza tempo da estação, mas, aumenta o gasto de energia para a estação [159].

Os quadros de anúncio são usados pelos APs de forma periódica para anunciar sua presença no perímetro. Junto com estes quadros são enviadas algumas informações importantes como: (i) o SSID, (ii) o canal de operação e (iii) a potência do sinal. Por motivos de segurança, é possível omitir o SSID [67]; esse assunto também será discutido na seção de segurança mais à frente.

Após encontrar a rede com que se deseja comunicar, a estação passará por um processo de autenticação junto à mesma. Este processo é necessário para que a estação possa provar sua identidade com o AP. Dois processos de autenticação são comuns: (a) sistema aberto e (b) sistema criptografado [67]. No primeiro, não existe nenhum procedimento de verificação de segurança, apenas a apresentação normal entre a estação e o AP. No segundo, a verificação se faz necessária e, por isso, algoritmos de autenticação usando esquemas de criptografia são usados. Caso tenha sucesso, a estação estará apta a se associar ao AP.

O processo de associação nada mais é do que um mecanismo pelo qual o padrão é capaz de prover condições para que uma estação seja transparente para outras estações na rede [105]. No caso de uma BSS apenas, o processo de associação é realizado automaticamente após a autenticação. Porém, quando existem duas ou mais BSSs, o processo ocorre da seguinte forma: (1) a estação envia uma requisição de associação (*association request frame*) e (2) o AP responde com uma mensagem informando o estado da associação (*association response frame*). A associação aqui é importante, pois, caso a estação esteja em movimento e o sinal atinja apenas vinte por cento de potência, havendo um AP com sinal mais forte no perímetro e pertencente à rede, a estação se desassociará do AP corrente e se associará ao AP com sinal mais forte [67].

[16] Os quadros são enviados para todas as estações da rede.

Conforme citado anteriormente, o AP envia quadros de anúncio para informação sobre sua localização e outras informações. Uma das informações presentes diz respeito à sincronização do tempo entre a estação e o AP. Para isso, é usada uma base de tempo comum, provida por um TSF (do inglês, *timer synchronization function*). Quando a estação recebe um quadro de anúncio, ela verifica o valor do tempo recebido do AP e atualiza seu TSF.

Uma das questões que influenciam as redes sem fio é a economia de energia, principalmente quando os dispositivos são móveis e não possuem fonte de energia à disposição em tempo integral. Dessa forma, existe um mecanismo de gerenciamento de energia que, para o modo infra-estrutura, é centralizado no AP. Para que seja efetivo o gerenciamento pelo AP, a estação, durante o processo de associação, envia o número de períodos de anúncio que estará em modo de economia de energia. Passado o período, a estação irá acordar e verificar se há algum quadro de dados, em estado de aguardo, para ser recebido. Nesse modo, a estação economiza muito mais energia, pois somente irá acordar de acordo com o tempo determinado pelo AP – que enviará quadros *multicast*[17] naquele período [159].

2.3.2 Detalhes da Camada de Enlace

Visando facilitar o entendimento para as seções posteriores, nesta subseção é apresentado um detalhamento sobre a camada de enlace de acordo com a especificação [105].

Segundo Earle [67] a camada de enlace é dividida em duas partes, (i) LLC (do inglês, *logical link control*) e (ii) o MAC. Sobre cada uma, existe um conjunto de responsabilidades. Em “(i)”, a principal função é prover interfaceamento para as camadas superiores, controlando o fluxo e possíveis erros de pacotes. Em “(ii)”, o controle de acesso ao meio é a principal função. Na figura 2.1, é mostrado o quadro MAC (para o modelo sem fio). O número acima de cada campo representa o tamanho em bytes. É possível observar a existência de uma divisão em três partes: (i) o cabeçalho (também conhecido como *header*), (ii) o corpo do quadro (onde são transportados os dados) e (iii) o campo de checagem FCS (*frame check sequence*).

[17] Quadros direcionado a várias estações, porém são especificadas as estações, diferentemente do processo *broadcast*, que é para todos que estiverem na rede.

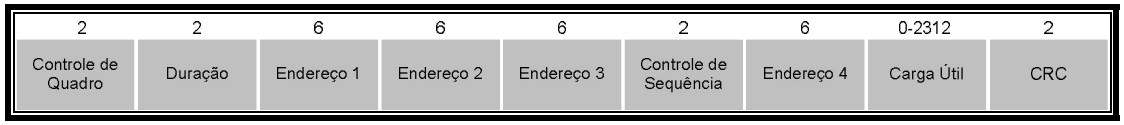


Figura 2.1 – Quadro MAC. Os números representam o tamanho em bytes de cada campo. Adaptado de [97].

Existem três tipos principais de quadros MAC:

- **Quadro de dados:** usado para transmissão de dados;
- **Quadro de controle:** utilizado para controle do acesso ao meio;
- **Quadro de gerenciamento:** usado para transmitir informações de gerenciamento. São transmitidos da mesma forma que os quadros de dados, porém não são repassados para as camadas superiores (isto é, da pilha do protocolo).

Alguns tipos de quadros têm fundamental importância para as redes sem fio. No modo infra-estrutura os quadros de controle são bastante utilizados e a seguir, são destacados os principais [67].

- **Requisição de envio (*Request to send* - RTS):** quadro proposto para ajudar a evitar colisões, principalmente para os terminais ocultos. Sua estrutura pode ser vista na figura 2.2. No campo duração é encontrado o tempo em microssegundos necessário para transmitir o quadro, o quadro CTS, o quadro de reconhecimento (ACK) e mais três curtos período de tempo conhecidos como Espaçamento Curto Interquadros (*Short Inter Frame Spaces* – SIFS) que representa o tempo mínimo de envio entre um quadro e outro;
- **Pronto para envio (*Clear to send* - CTS):** significa que a estação tem caminho aberto para enviar o seu quadro enquanto que as outras têm ciência da necessidade de esperar o tempo estabelecido neste quadro. O destino é a origem contida no quadro RTS anterior que é copiado juntamente com o tempo contido no campo duração. A figura 2.3 mostra o quadro CTS;
- **Reconhecimento (*Acknowledgment* - ACK):** nas redes sem fio, o reconhecimento de que um quadro foi transmitido com sucesso tem maior importância ainda. Caso a estação transmissora não receba um quadro de reconhecimento dentro de um determinado período de tempo, admitirá a ocorrência de falha e retransmitirá o quadro. Caso não haja nenhuma resposta para um quadro transmitido após um número fixo de retransmissões, a estação transmissora desistirá e descartará o quadro. Para retransmitir um quadro a estação precisa seguir as regras do protocolo

CSMA/CA. A figura 2.4 mostra o quadro ACK. O campo endereço de destino é copiado do campo de endereço 2 do quadro recebido (isto é, quadro de dados ou controle).

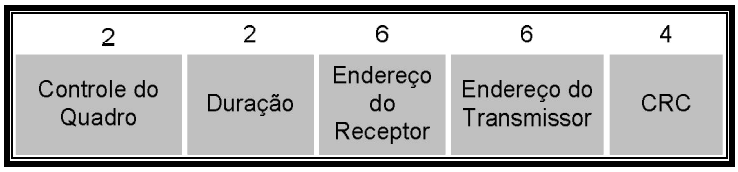


Figura 2.2 – Quadro RTS. Os números representam o tamanho em bytes. Adaptado de [105].

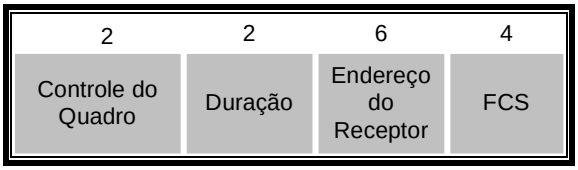


Figura 2.3 – Quadro CTS. Os números representam o tamanho em bytes. Adaptado de [105].

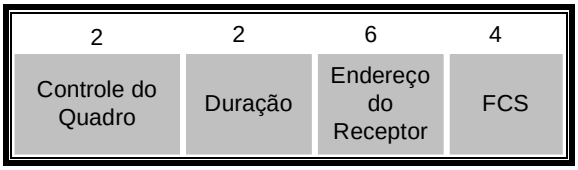


Figura 2.4 – Quadro ACK. Os números representam o tamanho em bytes. Adaptado de [105].

2.3.3 Questões de Segurança no Padrão IEEE 802.11

Geralmente, a questão da segurança de redes é vista como um problema da camada de aplicação da pilha de protocolos. Mas, para as redes sem fio esta não pode ser a regra principal. Segundo encontrado em [212] as redes sem fio possuem um meio vulnerável e pode ser até chamado de “meio compartilhado”. Dessa forma, a especificação [105] quebrou paradigmas e integrou componentes de segurança na camada de enlace para proteger o acesso à rede e manter a confidência dos dados que trafegam por ela. A primeira inclusão aconteceu em 1999 através do protocolo de segurança denominado WEP (do inglês, *Wired Equivalent Privacy)* [105]. A idéia inicial era alcançar os mesmos níveis de segurança encontrados nas

redes Ethernet. O protocolo WEP usa criptografia para proteger os dados transmitidos de uma estação a outra na rede através do AP. Normalmente, outros mecanismos de segurança atuam de forma conjunta para ajudar na tarefa de fornecer segurança à rede, entre eles destacam-se (i) a filtragem de endereços MAC e (ii) o uso de redes privadas virtuais (VPN) [67].

Mas, após algum tempo de uso (no mesmo ano), várias deficiências foram encontradas e relatadas sobre o protocolo WEP. Em um dos primeiros artigos científicos sobre WEP, publicado em outubro de 2000 [205], é descrito detalhadamente os riscos encontrados no padrão IEEE 802.11b, principalmente com relação à questão do tamanho das chaves de criptografia. No ano seguinte, Arbaugh [12] publicou outro artigo apresentando falhas no processo de autenticação de chave compartilhada através da técnica conhecida como criptoanálise de texto plano (*Plaintext Cryptanalysis*). Mas, o ponto chave, no qual foi provado que o protocolo WEP era fraco aconteceu em 2001 através do artigo de Fluher et al. [75] e a comprovação nos trabalhos [184] e [148]. Vários outros artigos e documentos foram discutidos e amplamente divulgados [13], assim como ferramentas para ataque foram lançadas na Web, destacando-se: WEPCrack, Airsnort e ASLEAP. Dessa forma, a *Wi-Fi Alliance*[18], impulsionada pela indústria, que estava preocupada com a demora do lançamento de uma nova versão do padrão que corrigisse o problema de segurança, apresentou o protocolo WPA (do inglês, *Wi-Fi Protected Access*) em 2003. O WPA foi chamado de subconjunto da versão que seria lançada pelo IEEE e, por falhas em seu desenvolvimento, novamente erros foram encontrados. Buscando atender as constantes reclamações do meio corporativo, que estava bastante desconfiado dos problemas de segurança e padronização para o modelo sem fio, em 2004, um novo padrão foi lançado pelo IEEE, sendo chamado IEEE 802.11i [105]. Este novo padrão mostrou ser bastante seguro, sendo um ponto forte o uso do algoritmo AES[19] (do inglês, *Advanced Encryption Standard*) como base para segurança dos dados. Mas, este último padrão ainda continha um defeito: os quadros de gerenciamento continuavam sem proteção, ou seja, não eram criptografados, permitindo os ataques do tipo negação de serviço (DoS). Notícias[20] do IEEE informam que o grupo está empenhado sobre o problema e a próxima versão, nomeada de IEEE 802.11w poderá ser lançada em breve.

Para mostrar o nível de vulnerabilidade de segurança encontrada no padrão IEEE 802.11, a seguir são apresentados detalhes dos protocolos WEP, WPA, WPA2.

---

[18] Associação Internacional responsável pela certificação dos produtos Wi-Fi baseados no padrão IEEE 802.11.
[19] Os algoritmos de chave-simétrica (também chamados de Sistemas de Chaves Simétricas, criptografia de chave única, ou criptografia de chave secreta) são uma classe de algoritmos para a criptografia, que usam chaves criptográficas relacionadas para a decifragem e a cifragem.
[20] Mais informações no site especialista em segurança de redes: http://www.net-security.org

2.3.4 O Protocolo WEP

Com mais de dez anos desde que foi introduzido no padrão IEEE 802.11 em 1999, este protocolo utiliza o algoritmo de criptografia RC4[21] (do inglês, *Ron´s Code #4*) [171] [22] para evitar que os dados dos usuários que estão transitando pela rede sejam lidos (isto é, entendidos) e, para verificar a integridade dos dados usa o modelo CRC-32 (do inglês, *Cyclic Redundancy Checks*). No protocolo, existem dois tipos de autenticação, (i) sistema aberto e (ii) e sistema de chave compartilhada [67].

No sistema aberto é permitida a qualquer dispositivo a associação à rede. A única exigência para isso é a informação do SSID. Para exemplificar, na figura 2.5 é apresentado o processo de autenticação por sistema aberto. Trata-se de um processo simples e rápido; a estação solicita a autenticação ao AP através de um pedido que é respondido com uma mensagem informando que a estação foi autenticada. Conforme já citado anteriormente, para se comunicar, a estação precisa se associar ao AP, o que acontece após a confirmação da autenticação por parte do AP.

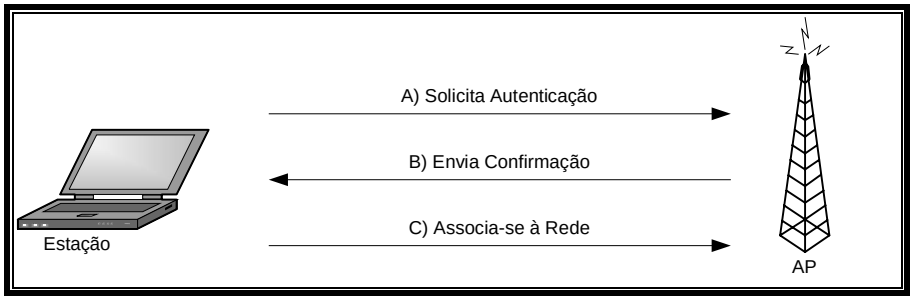


Figura 2.5 – Autenticação por Sistema Aberto. Adaptação própria.

No sistema de chave compartilhada, o funcionamento é baseado no mecanismo de autenticação “desafio / resposta” e, existe uma chave que deve ser idêntica para a estação e o AP, ou seja, ambos precisam ter a mesma chave. Conforme ilustra a figura 2.6, o processo é iniciado novamente pela estação através do envio de um pedido de autenticação, que é respondido pelo AP com uma mensagem contendo um texto-desafio (*Challenge Text*). Para responder ao texto, a estação (isto é, o cliente) utilizará sua chave para criptografar e enviará de volta para o AP. No AP, ocorre o processo contrário, ou seja, o AP descriptografa o texto

[21] Ron Rivest é o criador do RC4. Ele é membro do MIT e de outras importantes instituições. Mais informações em: http://en.wikipedia.org/wiki/Ron_Rivest
[22] Outros artigos e relatórios sobre RC4 podem ser encontrados em: http://www.wisdom.149.ac.il/~itsik/RC4/rc4.html

com a sua chave e compara o resultado com o texto original (isto é, aquele que foi enviado). Para conseguir se associar à rede, o resultado precisa ser igual.

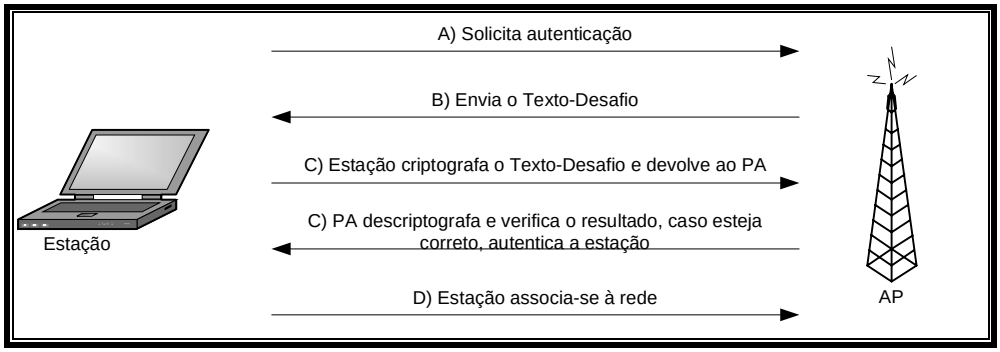


Figura 2.6 – Autenticação por Chave Compartilhada. Adaptação própria.

O protocolo WEP é usado apenas entre a estação e o AP, não sendo aplicado ao tráfego entre o AP e a rede cabeada [67]. Em todas as mensagens enviadas utilizando o protocolo é adicionado um valor de verificação de integridade ICV (*Integrity Check Value*), que pode ser chamado de CRC-32, antes da aplicação do algoritmo de criptografia. O valor é inserido no campo ICV do quadro e é comparado no receptor após o cálculo do CRC-32 da mensagem. Caso haja alguma diferença, o quadro será descartado. O ICV é criptografado, assim como a mensagem a ser enviada, através do algoritmo RC4. Este algoritmo usa chave padronizada de 64 bits que é dividida em duas partes: (a) vetor de inicialização (*Initialization Vector* - IV) com 24 bits e (b) chave estática (isto é, aquela compartilhada pela estação e o AP) com 40 bits. A princípio, o valor de IV deve ser dinâmico, porém não é especificado pelo padrão, sendo este processo usado por alguns fabricantes, e por outros não [67]. A única especificação é que a chave IV seja trocada quadro a quadro (isto é, a cada quadro dever ser mudado o valor IV). Outro fator importante é que não foi especificada a forma de incrementar o IV, sendo usadas diferentes formas por cada fabricante. A figura 2.7 detalha o processo WEP, sendo possível observar que o valor do IV, e o cabeçalho, não são criptografados. Por ser um algoritmo de criptografia de chave simétrica, a mesma chave deve ser usada no receptor para decodificação e por isso, a necessidade do envio, em texto claro, do valor do IV, concatenado à mensagem criptografada. Observando ainda a mesma figura, o processo inicia com uma mensagem para ser enviada, a esta mensagem é acrescido o cálculo do ICV, logo a seguir, usando um algoritmo gerador de números pseudo-aleatórios PRGA (do inglês, *Pseudo-Random Generation Algorithm*), o RC4 gera o valor em IV e concatena com a chave

estática, usa essa chave para criptografar os dados e no final, antes de transmitir, acrescenta o valor de IV (sem criptografia).

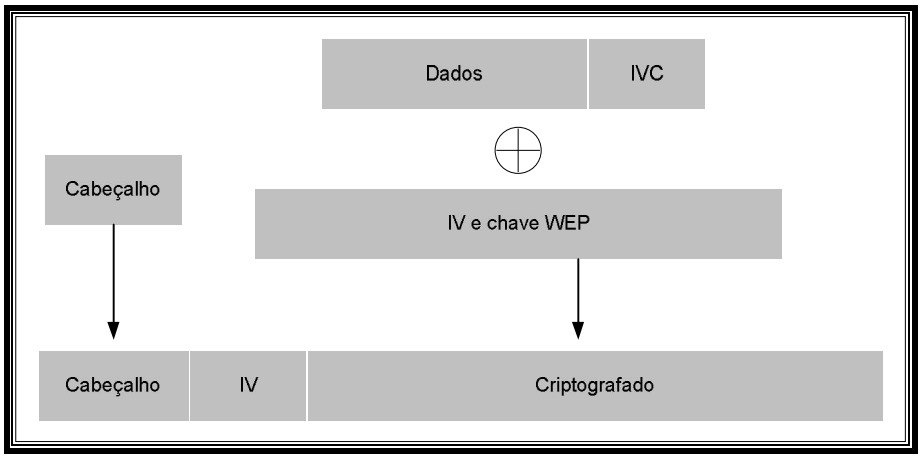


Figura 2.7 – Esquema de cifragem WEP. O cabeçalho e o IV não são criptografados representando um dos pontos falhos do protocolo [67].

Conforme já apresentado na introdução deste tópico, o protocolo WEP apresentou falhas significativas que comprometiam a segurança da rede. Conforme apresentadas nos trabalhos [184], [13], [148] e [171], as principais vulnerabilidades são:

- **Tamanho da chave:** no modelo original, a chave estática possuía 40 bits permitindo ser quebrada por ataques do tipo força bruta (isto é, um tipo de ataque no qual são testadas todas as combinações de chaves possíveis). Modelos posteriores lançados pelos fabricantes utilizaram chave estáticas maiores (entre 104 e 232 bits), porém mantiveram os mesmo 24 bits para o IV;
- **Reuso de chaves:** matematicamente, 2^{24} vetores diferentes podem não ser suficientes para evitar a repetição de IVs em uma rede com alto tráfego. Seria possível a repetição das chaves usadas pelo RC4 causando uma falha estrutural que provocaria a falta de confidência dos dados, um dos pilares da segurança. Um agravante neste caso está na escolha dos IVs, caso seja aleatória estará à mercê do Teorema do Aniversário[23]. Segundo Linhares e Gonçalves [132], após 4823 pacotes há uma probabilidade de 50 por cento da ocorrência de repetição de IV;
- **Gerenciamento de chaves:** tendo em vista que toda a segurança no protocolo WEP está baseada no segredo da sua chave, é importante que a troca da mesma

[23] O Teorema do Aniversário faz parte das teorias das probabilidades, segundo o mesmo, num conjunto de 23 pessoas existe mais de 50 por cento de chance de duas pessoas terem a mesma data de nascimento

seja feita com freqüência considerável. Como não possui um mecanismo para gerenciar as chaves e trocá-las dinamicamente, em redes muito grandes este processo torna-se difícil e muitas vezes não são feitas as trocas de maneira adequada e em tempo hábil para evitar um ataque;

- **Problemas com os IVs:** por ser de chave simétrica e, por isso necessita do mesmo IV no processo de decodificação, os IVs são passados em claro, é nesse ponto que foi criado um ataque que tem por nome FMS (em homenagem aos seus criadores) [75] que tem como principal objetivo descobrir o restante da chave necessária para o processo de decodificação, haja vista que no IV é passada a parte inicial da chave, e em texto claro. Em 2007, outra técnica de ataque sobre IVs foi apresentada por [192]. Denominada PTW[24] (novamente refere-se aos três autores) essa técnica é aplicada sobre tráfego ARP melhorando consideravelmente o tempo e a quantidade de quadros coletados;
- **Ineficiência do protocolo de autenticação:** é possível que um atacante se autentique sem que tenha conhecimento da chave WEP, para isso basta que o mesmo tenha acesso (por exemplo, fazendo um ataque passivo) aos dados que estão trafegando pela rede. Em algum instante ele conseguirá ter acesso a um pacote texto-desafio que passa em claro, e a cifra para o mesmo. Em sua posse, o atacante poderá usá-los para encontrar as chaves geradas e em seguida criar uma resposta válida para qualquer texto-desafio;
- **Problemas com RC4:** vários artigos publicados e encontrados em [163] apresentam as principais falhas deste modelo, principalmente sobre o algoritmo KSA (do inglês, *Key-Scheduling Algorithm*), passível de ataques estatísticos que podem revelar a chave WEP estática, segundo [171];
- **Re-injeção de pacotes (ou quadros):** para facilitar os ataques estatísticos é preciso que haja um alto trafego na rede, o que nem sempre acontece. Mas, através da ineficiência do modelo de autenticação, pacotes e quadros podem ser re-injetados e agilizar os ataques na quebra da chave WEP. Ferramentas como AirSnort e Aircrack-ng permitem o uso de re-injeção de pacotes;

[24] Desenvolvida na Universidade de Tecnologia de Darmstadt na Alemanha. Mais informações em: http://www.cdc.informatik.tu-darmstadt.de/aircrack-ptw/

- **Problemas com CRC-32:** um ataque conhecido como Chopchop [17], desenvolvido por Korek[25], explorou uma deficiência encontrada no CRC-32 quando usado como uma função linear, pois o mesmo não possui chave. Através destas características é possível realizar modificações de mensagens que foram capturadas e reenviá-las sem que o receptor descubra a fraude por causa da linearidade da função que detecta erros. Como não possui sistema de chaves secretas, é possível descobrir uma seqüência RC4 e com ela se autenticar na rede;
- **Negação de serviço (DoS):** quando um dispositivo sem fio recebe um quadro do tipo "*De-authentication*" geralmente ele é invalidado da rede e não consegue mais enviar dados até que seja autenticado novamente. Uma técnica usada pelos atacantes para este tipo de ataque é forjar o endereço MAC de uma estação associada para atacar a rede. Os ataques de negação de serviço são muito temidos por prejudicar a disponibilidade da rede.

2.3.5 O Protocolo WPA

Lançado em 2003, trata-se de um subconjunto do padrão IEEE 802.11i [107] que seria lançado no ano seguinte pelo [104]. Nele, é utilizado (i) o algoritmo de criptografia TKIP (do inglês, *Temporal Key Integrity Protocol*) para cifrar fluxo de dados, (ii) IV de 48 bits (isto é, mais de 2^{48} de IVs diferentes), (iii) regras para a escolha e verificação de IVs, (iv) um novo código de verificação de mensagens, (v) distribuição e derivação de chaves. Todas essas mudanças tinham por objetivo corrigir as vulnerabilidades do WEP e, para maior compreensão, a seguir são apresentadas, de forma detalhadas dentro do contexto de (a) autenticidade, (b) integridade e (c) confidencialidade, cada uma delas.

O WPA possui dois modelos distintos de funcionamento: um para redes pequenas (isto é, domésticas ou de pequenas empresas) e outro para as grandes redes (isto é, corporativas). Dessa forma, para cada um desses dois modelos existe um método de autenticação [67]:

- **Pequenas redes:** visando facilitar o uso por parte da grande maioria dos usuários comuns (isto é, aqueles que não possuem conhecimento para configurar equipamentos de rede sem fio), foi criado a idéia de usar mais caracteres para a chave compartilhada entre a estação e o AP. Dessa maneira, foi desenvolvido o WPA-PSK (do inglês, *WPA-Pre Shared Key*) que na verdade trata-se de uma

[25] Hacker conhecido no mundo virtual. Neste endereço é possível encontrar suas explicações sobre o ataque: http://www.netstumbler.org/f50/chopchop-experimental-wep-attacks-12489/

passphrase (uma frase senha). Nesse modelo, o AP é responsável pela autenticação, sendo as chaves configuradas manualmente nas estações da rede. Para a frase, é possível ter de 8 a 63 caracteres ASCII;

- **Grandes redes:** a mudança mais significativa neste caso é que o AP não é mais responsável pela autenticação que fica a cargo de um servidor de autenticação. Usuário e estação são autenticados por este servidor que utiliza o protocolo de autenticação IEEE 802.1x [106] para a comunicação entre o AP e o servidor de autenticação associado com algum tipo de serviço de autenticação EAP (do inglês, *Extensible Authentication Protocol*) desenvolvido. As RFCs 2865 [166] e 2866 [167] definiram o RADIUS (do inglês, *Remote Authentication Dial In User Service*), um servidor padrão bastante conhecido e geralmente usado para esta tarefa, mas é possível que outro serviço de autenticação seja utilizado. A figura 2.8 demonstra como é feita a autenticação usando o RADIUS. No primeiro momento, o cliente está tentando a autenticação e envia o seu perfil para o AP que irá encaminhar para o servidor de autenticação, após conferir os dados, se o perfil for válido, o usuário será autenticado e uma chave mestre MSK (*Master Session Key*) será enviada para usuário. A partir desse momento a estação poderá se associar à rede.

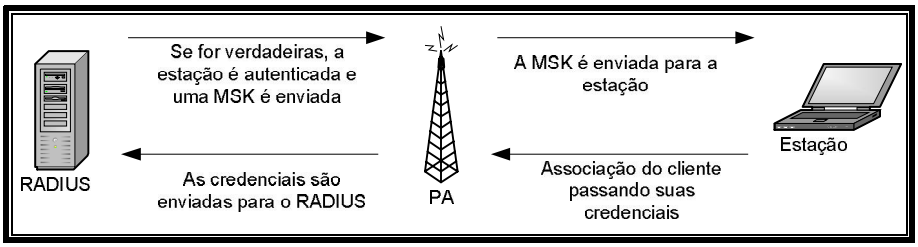


Figura 2.8 – Modelo de autenticação IEEE 98/EAP. Adpatação [105].

O par IEEE 802.1x/EAP era a parte que faltava para completar o aperfeiçoamento de segurança para redes sem fio[26]. Seu foco foram as fraquezas encontradas na autenticação do padrão IEEE 802.11i [107]. O protocolo EAP [168] funciona como um canal lógico de comunicação seguro entre o usuário e o servidor de autenticação e por ser uma estrutura genérica do protocolo de autenticação, pode trabalhar com diferentes tipos de mecanismos de

[26] O padrão IEEE 802.lx não está ligado apenas às redes sem fio, mas a todas as redes patrocinadas pelo IEEE 802

autenticação (por exemplo, certificados digitais SSL, bilhetes *Kerberos*[27], *smart cards*[28], entre outros).

Existem alguns tipos de EAP, os mais utilizados são: EAP-MD5, EAP-TLS (do inglês, EAP-*Tranport Layer Security*), EAP-TTLS (do inglês, EAP-*Tunneled Transport Layer S*ecurity), PEAP (do inglês, *Protected Extensible Authentication Protocol)* e LEAP (do inglês, *Lightweight Extensible Authentication Protocol*) [67]. A tabela 2.1 apresenta algumas das características dos protocolos de autenticação encontrados e disponíveis atualmente.

Tabela 2.1 – Protocolos de autenticação EAP.

Tipo de EAP	Sistema Aberto / Proprietário	Autenticação Mútua	Credenciais de Autenticação		Chave Material	Nome do Usuário	RFC
			Cliente	Autenticador			
MD5	Aberto	Não	Nome / Senha	Nenhuma	Não	Sim	1321
TLS	Aberto	Sim	Certificado	Certificado	Sim	Sim	2716
TTLS	Aberto	Sim	Nome / Senha	Certificado	Sim	Não	IETF Draft
PEAP	Aberto	Sim	Nome / Senha	Certificado	Sim	Não	IETF Draft
SIM	Aberto / GSM	Sim	SIM		Sim		IETF Draft
AKA	Aberto / UMTS	Sim	USIM		Sim		IETF Draft
SKE	Aberto / CDMA	Sim			Sim		IETF Draft
LEAP	Proprietário	Sim	Nome / Senha		Sim	Sim	CISCO

Após o processo de autenticação, ocorre a derivação da chave PMK (do inglês, *Pairwise Master Key*) oriunda da MSK. A chave PMK dará origem ao conjunto de chaves temporais PTK (do inglês, *Pairwise Transient Key*) que possuem duas importantes chaves que são chamadas de (i) chaves de criptografia TEK (do inglês, *Temporal Transient Key*) e (ii) de integridade de dados TMK (do inglês, *Temporal MIC Key*) (este tipo ocorre apenas no modelo de autenticação rede corporativa). Todo esse processo é chamado de apresentação de quatro mãos (4-*Way-Handshake*) e serve para garantir que o AP e a estação possuem a mesma chave PTK. As chaves PTK formam a base do protocolo TKIP. Elas são usadas por um período de tempo sendo substituídas posteriormente de forma dinâmica [67].

Uma deficiência substancial do WEP é o IV de 24 bits, portanto, no WPA, o IV possui 48 bits para dificultar (isto é, não permitir) a reutilização de IV. Porém, não havia espaço no

[27] Um protocolo de autenticação de rede implementado pelo MIT. Mais informações em: http://web.mit.edu/Kerberos/

[28] Pequenos cartões que contem um ou mais chips com capacidade de processamento. Possui um microprocessador e memória e mecanismos de segurança. Definidos pelas normas ISO/IEC 7816 e 7810. Existe um órgão sem fim lucrativos que apóia o entendimento dessa tecnologia: http://www.smartcardalliance.org/

cabeçalho do padrão sem fio e por isso, foi necessária uma adaptação conhecida como IV estendido que aloca os outros 24 bits acrescentados. Para evitar a re-injeção de quadros, foi desenvolvido o contador de quadros TSC (do inglês, TKIP *Sequence Counter*) que também é função do IV. O TSC é zerado a cada nova chave de criptografia e, com isso, os quadros que chegarem fora de ordem, serão descartados.

Segundo a especificação do WPA [105], a chave que alimenta o RC4 é diferente daquela do protocolo WEP. Para isso, é combinado o IV, o endereço MAC do transmissor e a chave de criptografia de dados e, após a geração da chave (isto é, pela combinação) é passada juntamente com o IV para o RC4 que seguirá o processo já apresentado do WEP (cf. Seção 2.3.4).

Para finalizar, a integridade do WPA se baseia na inserção de uma mensagem de verificação de integridade MIC (*Message Integrity Check*) junto ao ICV (já existente no WEP) que é adicionado ao quadro. O responsável pelo MIC é um algoritmo denominado Michael, na verdade uma função *hash*[29] não linear com criptografia chaveada, que produz uma saída de 64 bits, que corresponde a 8 bytes. Quando associado ao ICV, que contem 4 bytes, o total de bytes destinado a integridade dos dados é igual a 12 bytes. Para gerar o valor de MIC, o algoritmo Michael precisa do endereço de origem, de destino, da prioridade, os dados e uma chave de integridade conforme ilustra a figura 2.9. Este algoritmo é capaz de verificar qualquer modificação causada por erro na transmissão ou por manipulação [105].

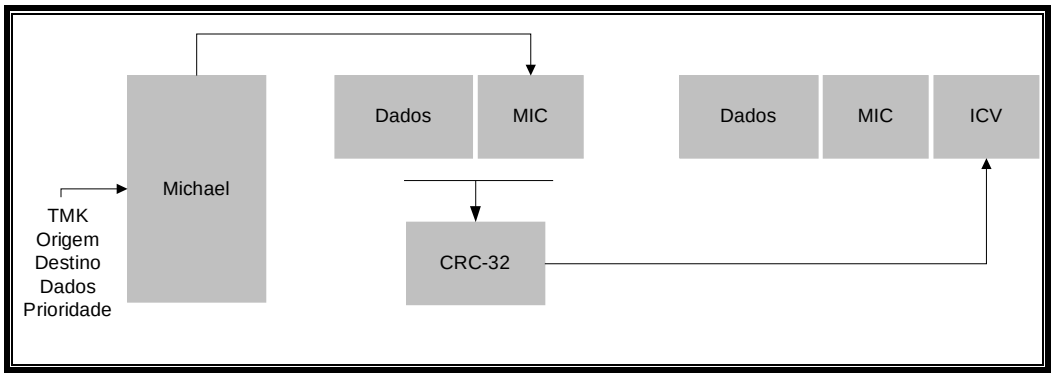


Figura 2.9 – Processo para garantir a integridade no WPA mostrando a entrada dos dados necessários para o algoritmo Michael. Adaptação própria.

[29] Uma função que transforma muita informação em pouca informação (em teoria) conforme: http://pt.wikipedia.org/wiki/Hash

A proposta inicial feita pela Aliança Wi-Fi era solucionar os problemas do WEP, de certo, o objetivo foi alcançado, porém, outros problemas surgiram, principalmente pelas falhas de desenvolvimento. Portanto, novamente o padrão de redes sem fio apresentava vulnerabilidades de segurança, sendo as mais importantes definidas como: (1) possibilidade de ataque de força bruta usando a técnica do dicionário e (2) a negação de serviço.

No mesmo ano do lançamento do protocolo WPA (isto é, 2003), Moskowitz [142] detalhou um potencial problema com WPA usando PSK e, em 2004 foi lançada a primeira ferramenta de ataque usando WPA-PSK. No artigo, o autor destaca a vulnerabilidade encontrada com uma frase-senha com tamanho menor que 21 caracteres para ataques do tipo usando dicionário de palavras. Esses dicionários são criados de acordo com o idioma, os costumes e peculiaridades de um local. Infelizmente, muitos usuários quando configuram suas redes usam senhas consideradas fracas, com poucos caracteres e de fácil recordação. Essa é uma porta aberta para esse tipo de ataque que quase sempre tem bons resultados, nestes casos [142]. Segundo o pesquisador, que participou do desenvolvimento do WPA, se no WEP era preciso capturar uma grande quantidade de quadros para que pudesse descobrir a chave, com o WPA, bastam apenas quatro pacotes específicos para decifrar a senha [142].

Um esquema de proteção no algoritmo Michael para evitar ataques de força bruta acabou permitindo que ataques de negação de serviço (DoS) pudessem serem feitos usando quadros mal formados. Isso ocorre porque quando dois MIC com erro são detectados dentro de um mesmo minuto, o AP cancela a conexão por 1 minuto e altera a chave de integridade [141].

Outro tipo de ataque ocorre sobre a fraqueza encontrada no algoritmo de combinação de chave. Nesse caso, os 32 bits mais significativos dos IVs são os mesmos para todos os outros. Segundo Moen et al. [141], se um atacante conseguir encontrar menos de 10 chaves RC4 poderá encontrar a chave de criptografia de dados e a chave de integridade. Não é um ataque muito fácil e prático, mas possui complexidade menor que os ataques de força bruta.

Em novembro de 2008, Erick e Tews [18] apresentaram um método de ataque que quebrava parcialmente a segurança WPA. Eles apenas conseguiram a quebra no sentido AP para estação, mas, no sentido contrário não conseguiram. Diferente do ataque via combinação alfanumérica, através de um dicionário usando ferramentas como CoWPAtty, que é bastante demorado e exige alto poder de processamento, o ataque possui duas etapas: (i) captura de quadros através de um método semelhante ao Chopchop aplicado sobre WEP e (ii) aplicação de funções matemáticas. O ataque é apoiado pelo padrão “IEEE 802.11e” que trata da qualidade de serviço sobre redes sem fio e com isso é capaz de recuperar a chave MIC e um

texto simples em um pequeno pacote (por exemplo, pacotes ARP[30] e pacotes DNS que não são fragmentados), após isso, pode falsificar os pacotes criptografados com chave MIC recuperada. Através desta técnica, os autores conseguiram sucesso obtendo a chave WPA em até 15 minutos. Algo muito eficiente se comparado com as técnicas anteriores.

Em setembro de 2009, Ohigashi e Morri [156] apresentaram um novo método, mais eficiente que o de Tews. Era conhecido que o ataque de Tews tinha problemas para ser aplicado sobre qualquer tipo de aplicação WPA por causa da necessidade das marcações do "IEEE 802.11e". Segundo Ohigashi e Morri [156], Tews já tinha mencionado que o ataque poderia ter sucesso se aplicado ao ataque do homem do meio MITM (*man-in-the-middle*), porém não apresentou nenhuma explicação ou exemplo prático sobre o mesmo. Dessa forma, após testes, foi provado que a teoria funcionava e por isso, estava eliminada a primeira barreira. Mas, como no ataque MITM a comunicação entre o AP e a estação é interceptada pelo atacante até que o ataque Chopchop finalize, se for um ataque demorado pode ser descoberto ou então prejudicar a transmissão a ponto de "derrubar" a rede. Cientes desse problema, Ohigashi e Morri [156] propuseram três modos de operação:

- **Modo repetidor:** o atacante retransmite todos os pacotes que contenham a sinalização SSID sem alteração e os pacotes do AP/estação são entregues à estação/AP;
- **Modo de recuperação da chave MIC:** objetiva encontrar uma chave MIC. Para isso usa o ataque Chopchop baseado no ataque MITM. O tempo previsto é de 12 à 15 minutos;
- **Modo falsificação de mensagens:** objetiva falsificar um pacote criptografado usando uma chave MIC. Se o alvo for um pacote ARP, o tempo de execução é de aproximadamente 4 minutos;

Para os autores, esses três modos podem ser mesclados dependendo da influência de uma "queda" da comunicação da rede. De acordo com esse novo método, uma vez que é recuperada a chave MIC, o atacante poderá descobrir o endereço MAC do AP (que geralmente é estático), e nesse caso, para um pacote ARP restará apenas 1 byte a ser descriptografado. No artigo [156] foi apresentada uma técnica para reduzir o tempo do ataque, até aproximadamente 1 minuto. Portanto, ficou provada a vulnerabilidade do WPA, de forma científica.

[30] Protocolo de resolução de endereços definido pela RFC 826. Ver: http://www.jonny.eng.br/trabip/rfc/rfc826.txt

2.3.6 O Protocolo WPA2

Para solucionar os problemas encontrados no WEP e WPA, o grupo IEEE lançou o padrão IEEE 802.11i [107]. Para melhorar os problemas de segurança oriundos do WEP e WPA, foi proposto o WPA2 pela Aliança Wi-Fi, que apresentava mudanças substanciais incluindo novos algoritmos de criptografia e de integridade. Porém, tal padrão não teve tanta popularidade como era previsto, principalmente por obrigar que fossem trocados os dispositivos anteriores (isto é, com o WPA, apenas a atualização de um *firmware* era suficiente para atender as especificações) [185].

Para dar proteção a integridade, a autenticação, a confidência dos dados e evitar a repetição, o WPA2 utiliza o protocolo CCMP (do inglês, *Counter-mode/Cipher Block Chaining Message Authentication Code Protocol*) conforme descrito em [105]. Este protocolo usa o padrão AES aprovado pelo NIST [206] em 2001. O AES é um algoritmo baseado em cifras de blocos (isto é, diferente do RC4 que cifra bit a bit, o AES cifra um bloco de bit por vez) com 128 bits e para o WPA2 foi adaptado para trabalhar com chaves de 128 bits. O principal objetivo em trabalhar com o AES é a prevenir que uma mensagem tenha o mesmo texto cifrado após o processo de criptografia. Trabalhando com diferentes modos de operação, o AES torna praticamente impossível (isto é, não se conhece nenhum ataque com sucesso sobre o AES) a descoberta da cifra de uma mensagem.

O CCMP[31] utiliza o CBC (do inglês, *Cipher Block Chaining*), um modo de operação do AES no qual o texto cifrado anteriormente é utilizado como entrada para o processo de criptografia subseqüente. Segundo [105] dois métodos CBC são utilizados no protocolo CCM: (a) CBC-CTR (do inglês, *Cipher Block Chaining Counter*) e (b) CBC-MAC (do inglês, *Cipher Block Chaining Message Authenticity Check*). O primeiro é usado para a confidencialidade enquanto o segundo é utilizado para garantir a integridade e autenticidade. Assim como o TKIP no WPA, o WPA2 é baseado no conceito de chaves temporais, sendo o CBC-CTR o algoritmo responsável pela criptografia do quadro. A chave de dados é simétrica com 128 bits, porém o IV continua com 48 bits [105].

Para o WPA2 poucas vulnerabilidades são conhecidas, porém, em [206] são apresentadas três técnicas para ataques de negação de serviço (DoS):

- **Ataque de reversão (*Rollback Attack*):** os dispositivos baseados no padrão [107] deveriam utilizar apenas autenticação sobre redes com segurança robusta (RSN),

[31] Definido pela IETF RFC 3610 em http://www.ietf.org/rfc/rfc3610.txt

porém, a maioria dos dispositivos ainda em funcionamento é baseada em redes pré-RSN (isto é, com WEP e WPA que não são considerados robustos) e por isso, ao permitir conexões não RSNA (do inglês, *RSN Associations*) se torna vulnerável pelas deficiências do WEP e WPA;

- **Envenenamento RSN[32] IE (RSN *Information Element Poisoning Attack*):** este tipo de ataque ocorre contra o protocolo *4-way handshake*. Neste caso o atacante usa mensagens com MIC incorreto e provoca uma desautenticação da estação com o AP;
- **De-associação (*De-association Attack*):** são usados quadros de gerenciamento da camada MAC para quebrar uma conexão existente entre uma estação e um AP. Para isso, o atacante envia um quadro desautenticação forjado para o AP a fim de desassociar a estação. O AP acredita que a conexão ainda existe o que permitirá ao atacante se passar pela estação e tentar a conexão com o AP.

Em [117] são mostradas algumas vulnerabilidades sobre o protocolo CCMP para o ataque TMTO (do inglês, *Time-Memory Trade-Off*) [155], um tipo de técnica que faz uso de dados pré-calculados, armazenados na memória, para reduzir o tempo de criptoanálise. Segundo os autores, o CCMP é bastante previsível e por isso tornou-se vulnerável a este tipo de ataque.

2.3.7 Técnicas de Ataque às Redes IEEE 802.11

O ato de espionar rede/sistema surgiu com as redes cabeadas e sempre foi um problema para os administradores. Para isso são utilizadas as ferramentas denominadas *Sniffers*, que tem a função de interceptar e decodificar transmissão de dados pela rede através de um meio, ou seja, são usadas de acordo com o meio de transmissão usado na rede. Segundo [67], é mais fácil espionar (pode ser usado o termo farejar também) as redes sem fio, haja vista que o meio de transmissão não é limitado a meios físicos como os cabos.

Para espionar uma rede sem fio o atacante usa um dispositivo 802.11 (por exemplo, uma placa de rede) com um *palmtop* ou *laptop* fazendo varredura em todos os canais ou especificando algum dentre os possíveis (isto é, de acordo com o padrão adotado para a camada física como, por exemplo, IEEE 802.11a/b/g/n) para coletar os quadros da camada

[32] Pertencente ao padrão 98 RSN (*Robust Security Networks*) introduz a noção de redes com segurança robusta

MAC [103]. Geralmente o atacante coloca o dispositivo em modo monitor para coletar os quadros que estão trafegando pelo raio de alcance do AP. Nesse modo, não há necessidade da autenticação e associação com a rede e, por isso, é conhecido como “ataque passivo”, pois o atacante está coletando dados para que possa aprender e planejar um possível ataque [140], [17].

Inicialmente o atacante fará varredura com os seguintes objetivos [103]:

- **SSID:** encontrar as redes disponíveis em um determinado perímetro é a primeira parte de um ataque. Neste caso o atacante está a procura do SSID da rede, informação que é passada por vários tipos de quadros (por exemplo, *beacons*, *probe requests*, *probe responses*). Quando o SSID é omitido no quadro de reconhecimento (isto é, *beacon*), o que pode ser feito através da configuração no AP, o atacante irá esperar até que haja algum tráfego naquele perímetro entre uma estação e um AP para, através dos quadros de requisição e resposta (isto é, para autenticação e associação) possa capturar o SSID. Este tipo de ação é possível pela falta de criptografia nos quadros de controle e gerenciamento – uma das falhas do padrão. Mesmo que no AP seja desativado o envio do SSID em requisições, o atacante pode forjar um MAC legítimo e enviar um quadro de desassociação para a estação, que precisará reassociar novamente. Dessa forma, o SSID será passado pelo quadro de requisição de reassociação da estação ao AP;
- **Endereço MAC:** em segundo lugar, o atacante recupera o endereço MAC de origem e destino, que é passado em claro em todos os quadros. O objetivo é detectar os MACs dos APs e estações para possivelmente (isto é, dependerá do tipo de ataque) forjar um endereço legítimo durante o ataque.

Quando a rede alvo utiliza algum tipo de criptografia, o atacante escolherá qual a forma mais eficiente para quebrar a chave. No caso de redes com WEP, ele precisará coletar quadros de dados, pois, o protocolo WEP gera IVs de 24 bits que são passados em claro para o receptor (isto é, por usar chaves simétricas, o receptor precisa ter a mesma chave do transmissor que é utilizada em todos os quadros que, combinada com o IV permitirá a descriptografia dos dados) [67]. Os IVs podem ser repetidos se houver grande tráfego de dados na rede e, muitos dispositivos tem inicialização em 0 com incremento de 1 por quadro [159].

Atualmente os ataques sobre redes usando o protocolo WEP conseguem sucesso em poucos minutos, principalmente se o atacante coletar bastante dados e usar cálculos estatísticos [67].

A idéia de todo atacante é não ser descoberto e, enquanto permanecer em modo passivo, a chance do administrador da rede encontrá-lo é bastante pequena. O único meio de ser encontrado é quando ele iniciar algum tipo de injeção de quadros na rede. Nesse caso, ele estará vulnerável para os mecanismos de segurança da rede IEEE 802.11 (por exemplo, algumas ferramentas de varredura injetam quadros *probe request* para descoberta de redes). Nesse caso, o ataque deixa de ser "passivo" e passa a ser chamado de ataque ativo [135].

A injeção de pacotes/quadros é feita através de duas técnicas [67]:

- **Falsificação do endereço MAC (*MAC Spoofing*):** o atacante precisa que seu pacote/quadro passe despercebido pela rede. Falsificar endereços MAC pode ser uma saída. A técnica consiste em falsificar os três últimos bytes do endereço, já que os três primeiros são atribuídos pelo IEEE de forma global. Muitos dispositivos aceitam mudança no endereço MAC, mas nem todo endereço de seis bytes é aceito como endereço MAC;
- **Falsificação de quadros (*Frame Spoofing*):** a partir do momento em que o endereço MAC foi falsificado, o atacante poderá passar quadros com conteúdo falsificado. Dificilmente será detectado, pois, não há criptografia para os quadros de controle e gerenciamento e, mesmo os quadros de dados, se estiverem criptografados, podem estar sofrendo um ataque do tipo MITM;

Quando o atacante consegue quebrar a chave de criptografia e com isso a associação com a rede, ele poderá usar de outra técnica conhecida como falsificação de endereço IP (*Internet Protocol*) ou IP *Spoofing*. Nesse caso, o ataque foi bem sucedido e o atacante já está na rede/sistema [140].

Além da falsificação do endereço MAC e dos quadros, um atacante pode decidir pelo uso de outra técnica de ataque: a inserção de outro AP falso no mesmo perímetro do AP verdadeiro (conhecido como *Rogue* AP [145]). Esse tipo de ataque ocorre principalmente porque os APs têm sido muito mal configurados, sendo facilmente descobertos pelos atacantes.

É consideravelmente fácil construir um AP com os dispositivos encontrados no mercado. Uma placa de rede, por exemplo, pode ser usada em conjunto com o sistema operacional como um AP. Através do aumento da potência do sinal de um AP falso, o que faz a estação se autenticar e associar a ele, o atacante pode capturar informações importantes para

desenvolver um ataque mais preciso. Esta técnica é conhecida por AP cavalo de tróia (*Trojan* AP) [135].

As técnicas descritas até este ponto comprometem a integridade e autenticidade da segurança. Porém, existem as técnicas que visam prejudicar a disponibilidade dos serviços da rede/sistema sendo totalmente indesejáveis para os usuários. Entre as principais técnicas, destacam-se [67], [140], [11], [135], [35]:

- **Desassociação (*De-association*):** envio de quadros *de-association* falsos para as estações contendo o endereço MAC de origem do AP. Para atingir seu objetivo, o atacante envia com um intervalo pré-determinado os quadros para que a estação não consiga reassociar;
- **Inundação de associações (*Flooding Association*):** esta técnica só funciona quando um AP não faz filtragem de endereços MAC. Os APs possuem uma tabela contendo a quantidade e o endereço MAC dos clientes associados. A especificação [105] apresenta 2007 campos para a tabela, porém os fabricantes podem mudar esse tamanho. O ataque consiste na inundação (por exemplo, através de endereços MAC aleatórios) do AP com pedido de associação. Quando estoura o limite (isto é, *overflow*) da tabela, os clientes serão impedidos de usar a rede/sistema por não poder mais se associar;
- **Falsificação de quadros de gerenciamento de energia (*Power Saving Forging*):** dispositivos sem fio (por exemplo, *palmtops* e *laptops*) precisam economizar energia. Para que seja feito de forma organizada, o padrão IEEE especificou que a estação deve passar uma mensagem informando ao AP que estará entrando em modo de economia de energia e que acordará após certo tempo. Durante o período estabelecido o AP armazenará as informações que deveriam ser encaminhadas para a estação. Quando a estação acorda, envia um quadro do tipo PS-Poll para o AP a fim de reaver os pacotes/dados que foram destinados a ela. É exatamente nesse momento que o atacante age, através da falsificação do quadro de *Polling*. Quando o AP recebe o falso *Polling* da estação, ele transmite e limpa a sua memória (isto é, o *buffer*). Como a estação está em modo econômico, não receberá os quadros e quando acordar não existirá mais informações para ela no AP;
- **Desautenticação (*de-authentication*):** para se associar a uma rede sem fio a estação precisa se autenticar. Ciente disso, o atacante ao perceber que uma estação está associada ou em processo de associação, envia quadros *de-authentication*

falsificados com o endereço MAC do AP. Este tipo de ataque também é lançado pelo atacante quando deseja alcançar sucesso com o ataque do homem do meio MITM;

- **Interferência eletromagnética (*Jamming*):** um tipo de ataque que usa a camada física como alvo. Geralmente, para realizar o ataque são utilizados antenas ou dispositivos capazes de provocar interferências eletromagnéticas (isto é, ruídos) que podem deteriorar o sinal a ponto de não ser possível o uso da rede. Este tipo de ataque tem sido estudado e a única forma de detectá-lo é através de dispositivos com capacidade para fazer varredura no espectro.

Quando uma rede sem fio possui *switches* agregados para acesso às redes cabeadas é possível existir uma vulnerabilidade bastante conhecida: o ataque por envenenamento dos pacotes ARP (ARP *Poisoning*) [72]. Tal ataque também faz parte da técnica MITM e pode partir de uma estação cabeada. O envenenamento consiste em corromper a tabela de endereços MAC do sistema operacional através de pacotes de resposta ARP com um endereço MAC falso. Quando montada na tabela ARP a referência falsa, o sistema operacional enviará pacotes para aquele endereço presente na tabela. Como não é possível determinar pelo protocolo ARP se uma resposta que chegou foi realmente requisitada, a tabela é montada erradamente.

Finalizando, outra técnica usada por atacantes é conhecida como roubo de sessão (*Session Hijacking*) [103]. O principal foco aqui é fazer com que o usuário perca sua conexão e o atacante assuma sua identidade e privilégios. Quadros *deassociation* e *de-authentication* são usados para este tipo de ataque com endereços MAC falsificados.

2.3.8 Exemplos de Ferramentas Usadas para Atacar as Redes IEEE 802.11

A fim de explorar as vulnerabilidades existentes nas redes sem fio, varias ferramentas foram criadas (isto é, usando as técnicas de ataques vistas na subseção anterior) e disponibilizadas para o domínio público. Algumas delas são extremamente eficientes a que se propõem sendo consideradas essenciais a qualquer atacante (e para qualquer administrador). Para fins de conhecimento e agregação de valor a este trabalho, são apresentadas, a seguir, as principais ferramentas que foram testadas para a escolha daquela que melhor se adaptaria as necessidades do projeto proposto. São elas (i) Aircrack-ng, (ii) AirSnort, (iii) Kismet e (iv) CoWPAtty [103] [35].

- **Aircrack-ng[33]:** desenvolvida por [63], essa é uma das mais conhecidas e completas ferramentas de análise para redes locais sem fio. Também serve para atacar e quebrar chaves WEP e WPA-PSK. Segundo os mantenedores da ferramenta, nela estão inseridas as técnicas de ataque FMS com algumas melhorias feitas por Korek assim como a técnica de PTW. Na verdade, existe um conjunto de ferramentas embutidas no Aircrack-ng conforme a tabela 2.2. Para ataques direcionados às redes usando o protocolo WPA, é usado o método de dicionário apenas.

Tabela 2.2 – Conjunto de ferramentas Aircrack-ng. Uma das mais completas ferramentas de ataque disponível na WEB gratuitamente.

Nome	Descrição
Aircrack-ng	Quebra chaves WEP e WPA (usa força bruta)
Airdecap-ng	Utiliza a chave descoberta para decriptografar os dados capturados
Airmon-ng	Responsável por mudar as placas para o modo monitor
Airodump-ng	Captura os quadros colocando em um arquivo “.cap”
Airtun-ng	Cria uma interface de túnel virtual
Aireplay-ng	Faz injeção de quadros (apenas sob Linux)
Airolib-ng[34]	Ferramenta usada para armazenar e gerenciar SSID e listas de palavras. Seu alvo principal são ataques sobre WPA/WPA2.

Além das vantagens de atuar como analisador de tráfego, quebra de chaves WEP e WPA, fazer injeção de pacotes/quadros, esta ferramenta pode ser usada em ambiente Linux e Windows;

- **AirSnort[35]**: usada para descobrir a chave WEP usando a técnica FMS. Como vantagem, é fácil e simples de usar, porém, precisa de muitos quadros para conseguir seu objetivo. Outra dificuldade encontrada é a descontinuidade do projeto;
- **Kismet[36]:** desenvolvida por Mike Kershaw, esta ferramenta é bastante completa. Funciona como detector de dispositivos sem fio na camada 2, escâner de rede e possui algumas características de detector de intrusão. A figura 2.10 mostra a tela principal da ferramenta. Suas características principais são:

[33] NG refere-se a nova geração da ferramenta e pode ser encontrada em: http://www.aircrack-ng.org/
[34] Na página oficial é possível encontrar exemplos e manuais: http://www.aircrack-ng.org/doku.php?id=airolib-ng)
[35] Página oficial: http://airsnort.shmoo.com/
[36] Página oficial: http://www.kismetwireless.net

- log compatível com as ferramentas Wireshark/TCPdump;
- log de IVs compatível com AirSnort;
- detecção de faixa de IP;
- descoberta de redes com SSID oculto;
- mapeamento gráfico das redes;
- pode trabalhar com arquitetura cliente/servidor permitindo múltiplos clientes para envio de dados ao servidor Kismet simultaneamente;
- identificação do modelo e fabricante do AP e clientes;
- detecção de AP com configuração padrão;
- decodifica pacotes com criptografia WEP em tempo de execução para redes conhecidas;
- pode ser integrada com outras ferramentas;
- gera saída em XML, HTML e CAP;
- pode detectar alguns tipos de ataque através de uma pequena base de dados de assinaturas (por exemplo, pode identificar a presença de um Network Stumbler[37] fazendo varredura).

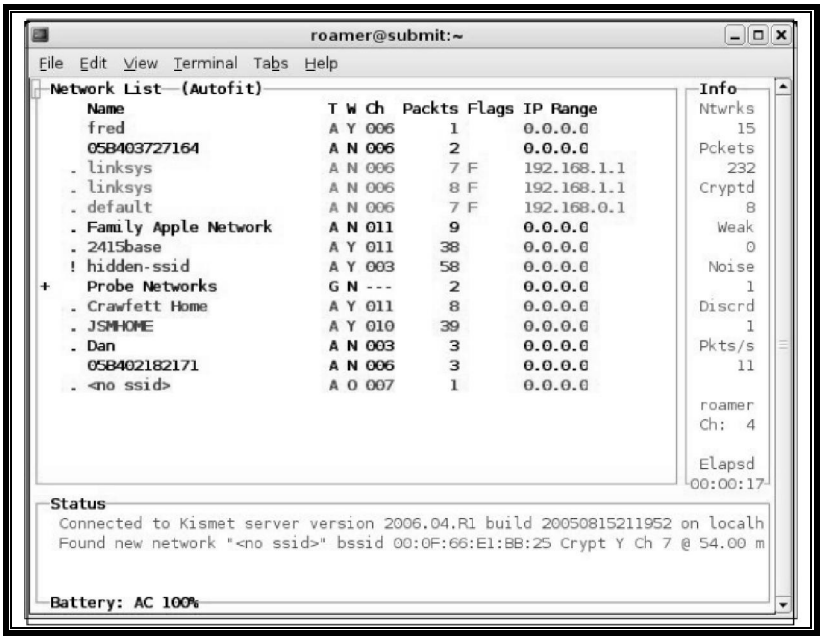


Figura 2.10 – Tela principal do Kismet onde é possível verificar várias informações sobre as redes e estações no perímetro de alcance da antena.

[37] Ferramenta usada para fazer varredura a procura de redes sem fio no ambiente Windows. Mais informações em: http://www.stumbler.net/

- **CoWPAtty[38]:** ferramenta criada por Joshua Wright[39] especificamente para ataques direcionados ao protocolo WPA-PSK. Neste caso, são usadas listas de palavras (*wordlists*) juntamente com um ataque de força bruta. O foco é a chave WPA, que com poucos caracteres e uma boa *wordlist* pode ser quebrada em questão de minutos. Uma desvantagem para o uso desta ferramenta são as próprias *wordlists*, geralmente são necessários milhões de palavras e por isso, os arquivos são muito extensos. A figura 2.11 apresenta uma tela da ferramenta onde é possível visualizar um ataque do dicionário e a descoberta da chave PSK.

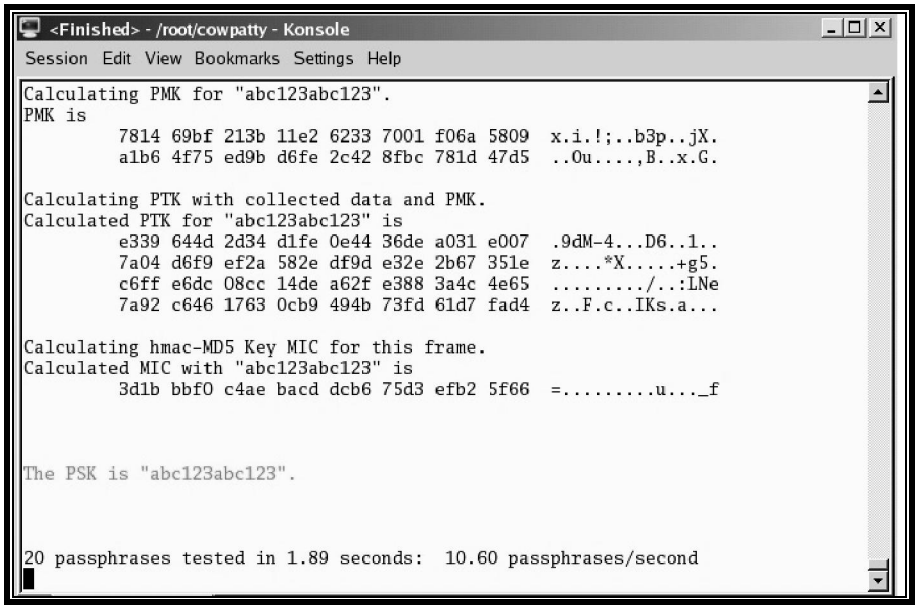


Figura 2.11 – Tela da ferramenta CoWPAtty mostrando a senha do PSK descoberta.

- **AirJack[40]:** ferramenta usada para injetar pacotes/quadros na rede usando por exemplo os quadros *de-authentication* para indisponibilizar o uso da rede. Pode ser um ataque direcionado a uma estação específica ou para uma BSS. Também é usada juntamente com outra ferramenta para forçar clientes re-associarem ao AP e assim gerar tráfego suficiente para quebrar chaves. Quando um quadro de-authentication é recebido pela estação, ela não pode descartá-lo, essa é uma regra do padrão de rede sem fio que é explorada por esta ferramenta;

[38] Página oficial: http://www.willhackforsushi.com/Cowpatty.html
[39] Pesquisador da Universidade Johnson & Wales e autor de vários artigos sobre segurança em redes sem fio
[40] Página oficial: http://sourceforge.net/projects/airjack/

- **HostAP**[41]**:** com esta ferramenta é possível criar um AP falso (ver *trojan* AP acima), ou seja, um atacante usando um dispositivo (por exemplo, a própria placa de rede sem fio) com características de um AP para enganar estações que estiverem num determinado perímetro de atuação de um AP verdadeiro. Nesse caso a estação irá se associar ao AP falso deixando o caminho livre para o atacante;
- **Wireshark**[42]**:** ferramenta de análise de quadros muito utilizada para redes cabeadas possuindo interface gráfica muito bem elaborada. Além de ser possível a instalação em vários sistemas operacionais, possui compatibilidade com praticamente todos os tipos de protocolos utilizados em redes. Para facilitar a utilização, tem esquema de filtragem de pacotes com possibilidade de escolhas de cores para facilitar a visualização por parte do usuário. Inspirada no antigo *TCPDump* [43], permite o trabalho com arquivos em vários formatos (por exemplo, ".cap" e ".dump"). Geralmente, os atacantes a usam para analisar o tráfego da rede antes de realizar um ataque. A figura 2.12 ilustra a interface bem elaborada desta ferramenta.

Outras ferramentas podem ser encontradas, inclusive com bons manuais, na Web. O emprego dessas ferramentas é livre, porém é preciso usar da ética profissional para sua utilização. Muitas empresas fazem testes de invasão contratando empresas especializadas para tentar quebrar a segurança, outras, com poder econômico menor, utiliza estas ferramentas como base para efetuar testes com sua própria equipe de profissionais.

[41] Página oficial: http://hostap.epitest.fi/
[42] Página oficial: http://www.wireshark.org
[43] Ferramenta bastante conhecida no mundo das redes cabeadas. Disponível em: http://www.tcpdump.org

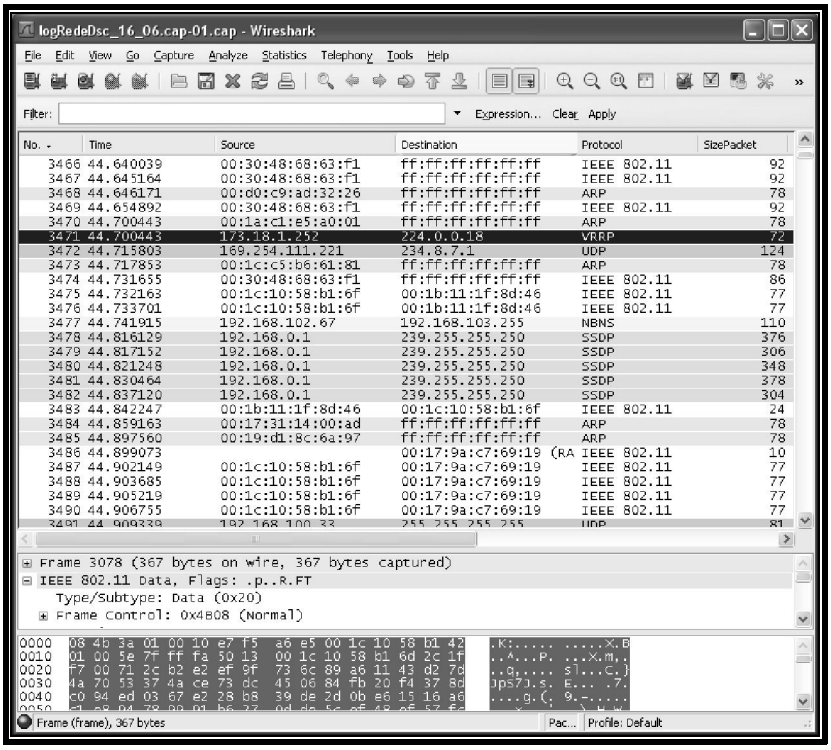


Figura 2.12 – Ferramenta de análise de quadros. É possível organizá-los por cores para facilitar a observação dos mesmos.

2.3.9 Tipos de Ataques

Na literatura, assim como na WEB, é possível encontrar vários tipos de ataques sobre as redes IEEE 802.11. Nesta seção, são descritos oito tipos. Todos eles foram usados durante os experimentos do Capítulo 6. São eles:

- **De-autenticação (De-authentication):** neste tipo de ataque o objetivo principal é derrubar a estação da rede usando os quadros "*de-authentication*". Dependendo do interesse do atacante, isso poderá servir para capturar os quadros necessários para o 4-*way handshake* em ataques sobre o protocolo WPA ou, em outro caso, impedir que a estação se conecte novamente enviando quadros continuamente. A figura 2.13 ilustra o ataque em execução. Funciona tanto para WEP quanto WPA e pode ser direcionado a uma estação ou para várias. Por exemplo, a omissão de –c e o MAC permitem o envio em modo broadcast;

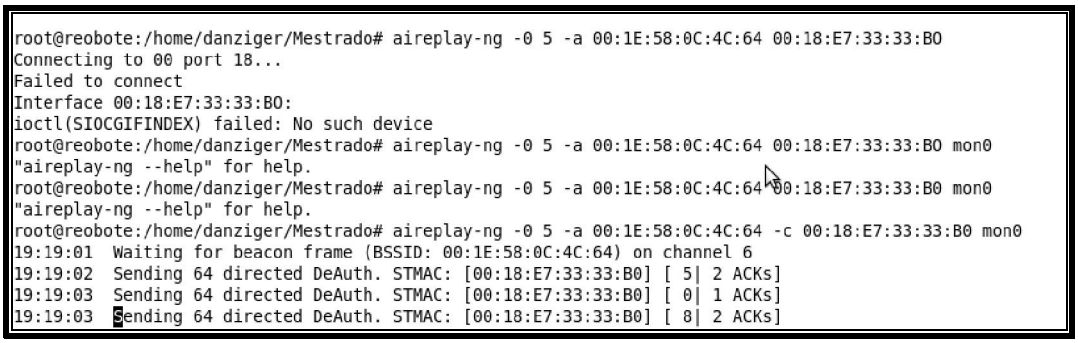


Figura 2.13 – Ataque De-authentication em execução. É possível observar o reconhecimento (ACK) do AP ao quadro. Esse ataque é direcionado a uma estação.

- **Autenticação falsa:** geralmente é acompanhado de outra técnica de ataque conhecida como *MACSpoofing* (bastante utilizada no mundo cabeado), para o caso de filtragem por parte do AP. Esse fato ocorre pela necessidade do atacante ter um cliente associado ao AP, pois este método não gera tráfego ARP. Este tipo de ataque ocorre sobre o protocolo WEP e na figura 2.14 é mostrado em execução. Alguns ataques utilizam MAC não existente, tais como "00:11:22:33:44:55", principalmente quando o AP não faz filtragem de MACs;

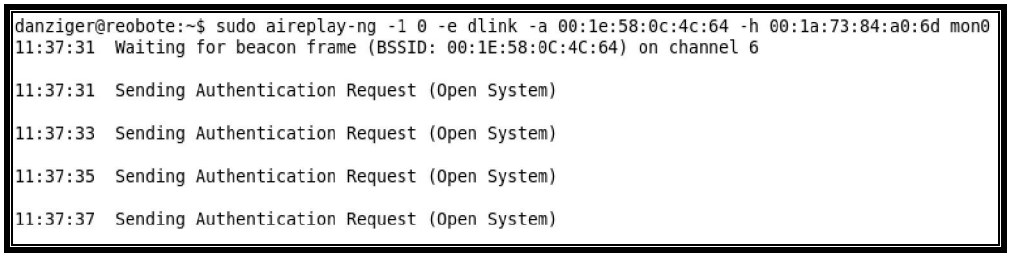


Figura 2.14 – Ataque de autenticação falsa sobre o AP utilizando *MACSpoofing*. O atributo –h define o MAC da estação vítima. Se o AP não for configurado para filtragem de MAC, pode ser qualquer MAC válido.

- **Replay de pacote interativo (*interactive packet replay*)[44]:** nesse ataque é possível escolher o tipo do quadro para ser injetado na rede (por exemplo, o quadros *data*). É comum o uso da ferramenta Packetforge-ng para criar um pacote e depois utilizá-lo no ataque. Alguns cuidados precisam ser tomados, por exemplo, não é qualquer tipo de pacote que pode ser utilizado para reenvio. Apenas aqueles pacotes que, podem ser aceitos pelo AP e gerarão IVs, que é o objetivo principal do ataque. A figura 2.15 representa a execução deste tipo de ataque;

[44] Neste sitio da WEB é possível ver a explicação completa deste tipo de ataque: http://www.aircrack-ng.org/doku.php?id=interactive_packet_replay

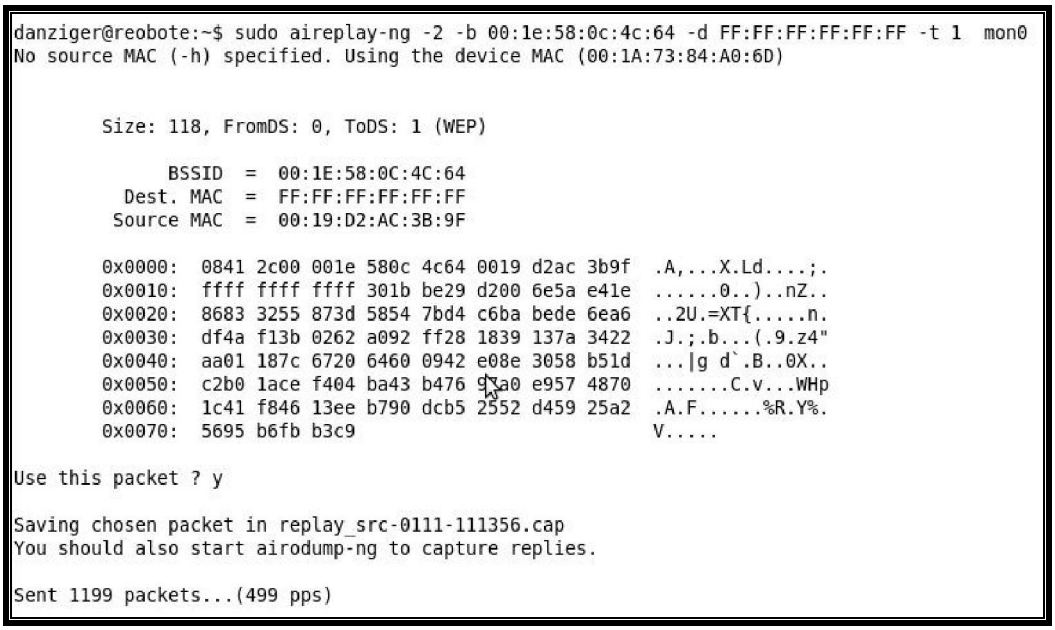


Figura 2.15 – Ataque interativo em execução com uma estação de origem pertencente à rede. Neste ataque, o destino é toda a rede (ver Dest. MAC) e o pacote é escolhido pelo atacante.

- **Requisição de pacotes ARP:** um dos mais eficazes ataques contra o protocolo WEP permite a captura de IVs com significante eficiência. Após o envio de um pacote contendo uma requisição ARP para o AP, este irá repetir o pacote contendo um novo IV. O atacante repete continuamente até que tenha capturado um número grande de IVs para que possa aplicar uma ferramenta de decriptografia (por exemplo, WEPCrack). A figura 2.16 apresenta o ataque em execução mostrando que foi recebido pela estação através dos quadros de reconhecimento ACK;

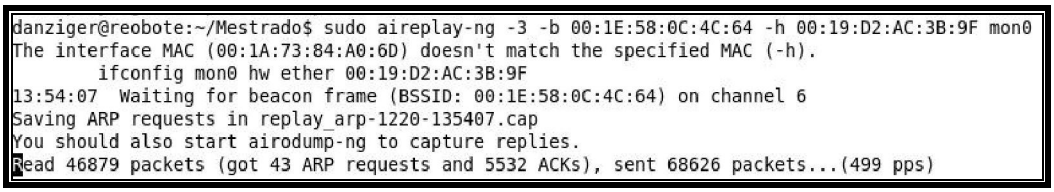


Figura 2.16 – Ataque de requisição ARP em execução. Nesse caso, 43 ARP foram enviados gerando 5532 quadros de reconhecimento. Este é um dos ataques mais eficientes sobre WEP.

- **Chopchop**[45]**:** ataque bastante poderoso no qual o atacante consegue ver o conteúdo do pacote sem ter a chave WEP. Ele também pode ser aplicado com sucesso contra WEP dinâmico. Nem todos os APs são vulneráveis a esse tipo de ataque, porém, aqueles que o são, descartam os pacotes menores que 42 bytes. Dessa forma, o atacante usa, por exemplo, o Aireplay-ng para tentar adivinhar o restante dos dados em falta que são, tanto quanto o cabeçalho, previsíveis. Neste caso, o atacante precisa apenas de um quadro de dados, no qual irá fazer o calculo do CRC para verificar se é equivalente ao que foi feito sobre os dados adivinhados. Este ataque é mostrado na Figura 2.17;

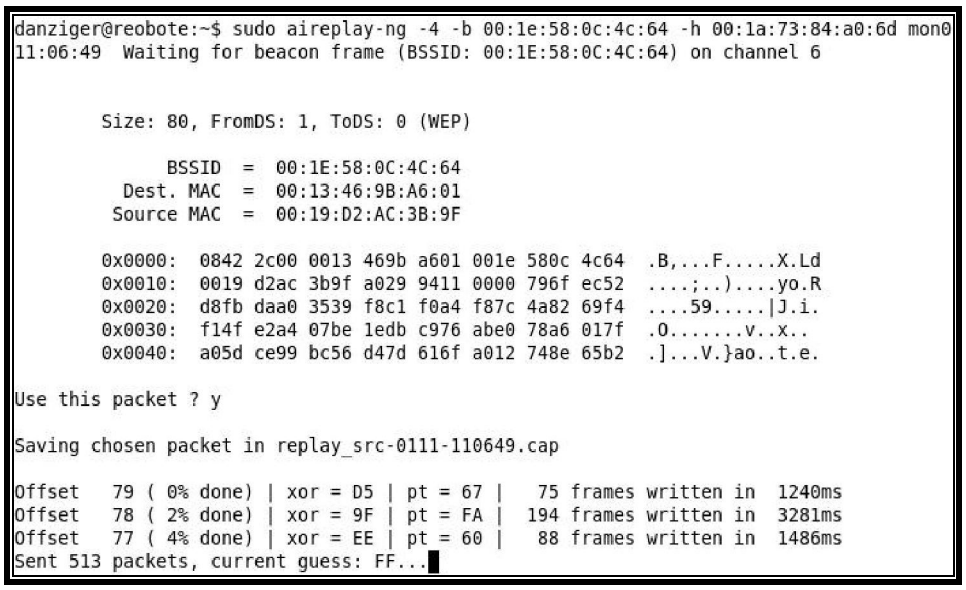


Figura 2.17 – Ataque Chopchop de Korek efetuado com sucesso. Um quadro de dados é escolhido e enviado com vários destinos diferentes. Se o AP descarta os quadros com tamanho menor que 42, o programa tenta adivinhar o restante da informação.

- **Fragmentação:** este é um ataque sobre WEP que tem por objetivo capturar o PRGA (*Pseudo Random Generation Algorithm*), portanto, assim como o ataque Chopchop, este ataque não recupera a chave WEP. A partir do momento em que o PRGA é recuperado, ele pode ser usado juntamente com a ferramenta Packetforge-ng para gerar pacotes e injetar novamente na rede. Basta apenas um quadro de dados capturado para o início do ataque. Para realizar seu objetivo, o atacante, através da ferramenta Aireplay-ng, envia pacotes ARP ou LLC com uma pequena

[45] Mais informações sobre a teoria por traz desse ataque são encontradas no sítio http://www.aircrack-ng.org/doku.php?id=chopchoptheory&DokuWiki=467f32bfb79adb0cd07ee37c8365a701

parte da chave e algumas informações conhecidas do AP (isto é, técnica conhecida como *bit flipping*). Se o AP ecoar de volta o pacote, o atacante insere maior quantidade de informações até que consiga obter algo em torno de 1500 bytes de PRGA. Em posse dessa informação a segurança estará comprometida. Este ataque pode ser visto na figura 2.18, exatamente no momento em que um quadro "*data*" é identificado para ser escolhido ou não pelo atacante;

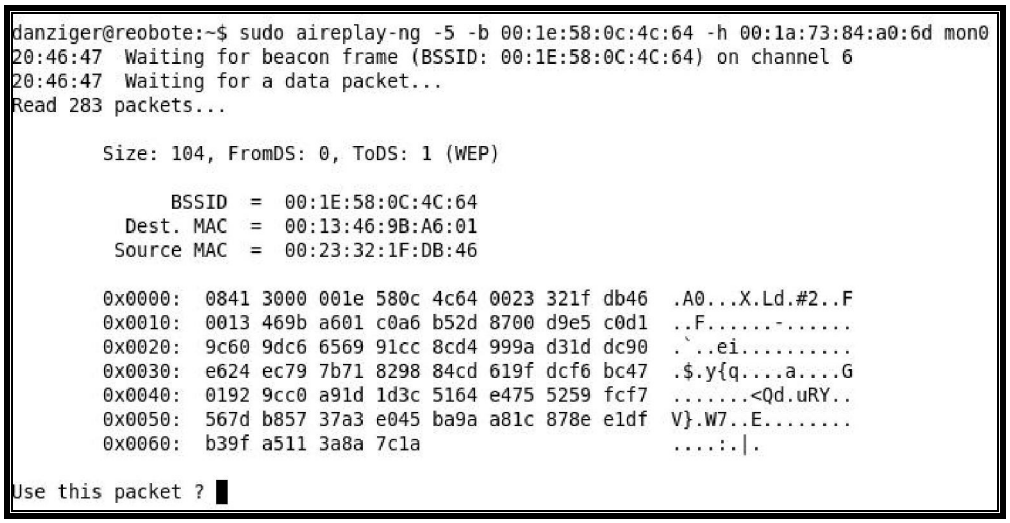


Figura 2.18 - Ataque de fragmentação em ação. Na imagem é possível ver o momento em que um quadro "*data*" é capturado e apresentado para o atacante escolher ou não para o ataque.

- **Café-Latte:** este ataque tem o objetivo de quebrar a chave WEP. Para atingir esse objetivo, o atacante precisa capturar um pacote ARP do cliente (vítima), manipular o mesmo e enviá-lo de volta. A vítima torna a enviar os pacotes novamente e dessa forma, os pacotes, com novos IVs são capturados. Segundo o artigo [162], neste ataque o atacante aplica diferentes técnicas de quebra da chave WEP, entre elas FMS, Korek, PTW. Para isso, é mudado um pacote de anúncio gratuito do protocolo ARP para um pacote de requisição ARP. Dessa forma, ao enviar ao cliente novamente, o mesmo responderá com um correto criptografado pacote de resposta. Esse tipo de ataque ocorre porque a vítima não consegue definir se o pacote de requisição ARP recebido é fidedigno. Geralmente, para ter sucesso neste tipo de ataque é usado um AP virtual (por exemplo, usando a ferramenta

HostAP[46]) para requisições *probe*. Na figura 2.19 é possível ver o ataque em execução e a quantidade de pacotes ARP enviados e ACK recebidos;

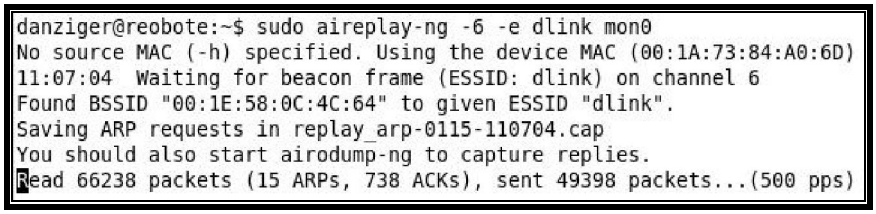


Figura 2.19 – Ataque Café-Latte em execução mostrando 15 pacotes ARP capturados e a quantidade de ACK recebidos após o reenvio do pacote modificado.

- **Hirte:** uma extensão do ataque Cafe-Latte, porém, pode usar qualquer pacote IP ou ARP. Ao capturar um pacote ARP, o atacante manipula-o alterando a posição do endereço IP e fragmenta-o em dois usando os conceitos de fragmentação de pacotes para enviar ao cliente. Se for manipulado corretamente, o cliente responderá aos pacotes (fragmentados) recebidos e o atacante terá a chance de capturar pacotes com parte da chave WEP. Para o pacote IP que será usado no ataque, uma técnica similar ao ARP é usada, porém existe uma limitada quantidade de PRGA disponíveis e por isso mais três fragmentos são necessários. Para ilustrar, a figura 2.20 mostra o ataque em execução.

É importante salientar que o primeiro ataque é conhecido como ataque de negação de serviço (DoS) atingindo tanto os protocolos WEP, WPA e WPA2. Por outro lado, os outros ataques objetivam a quebra da integridade e da confiabilidade dos dados trafegando pela rede. Portanto, os ataques atingem de maneira completa os três pilares da segurança de rede.

[46] Essa ferramenta permite que uma estação se transforme em um AP. Para maiores informações ver: http://hostap.epitest.fi/

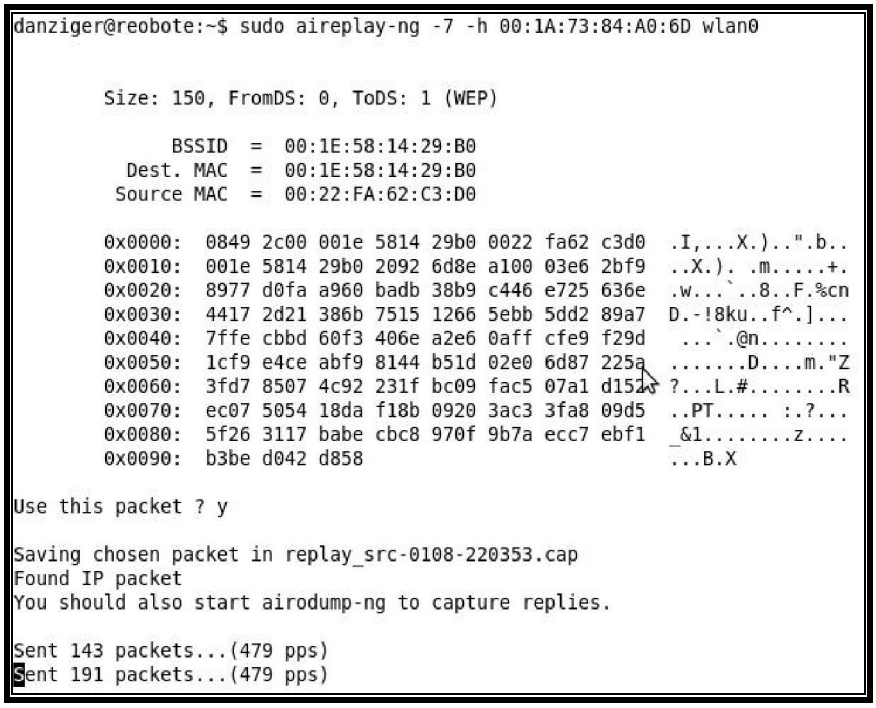


Figura 2.20 – Ataque Hirte em ação. Um pacote IP foi encontrado e dessa forma, fazendo *bit-flipping*, as posições dos bits são alteradas permitindo que seja fragmentado o pacote (geralmente em três partes).

2.4 Considerações Sobre a Segurança em Redes IEEE 802.11

Conforme pôde ser visto nas seções anteriores, vários problemas com a segurança fazem parte do padrão de redes sem fio. Mesmo assim, é possível perceber que houve evoluções significativas. Mas, muitos dos problemas atuais estão relacionados não às falhas que possam existir nos protocolos de segurança, mas sim na falta de padronização do seu uso. Como exemplo, o padrão WPA2 [105] trouxe modificações importantes e também um problema a mais para os usuários: para usá-lo é preciso substituir os dispositivos (isto é, o hardware), pois não são compatíveis com os dispositivos antigos que usam WEP e WPA com algoritmos fracos [206]. Além disso, o processamento ficou mais pesado e houve perda na capacidade de transmissão (por exemplo, a velocidade) [159]. Aliado a esse problema, a falta de padronização dos dispositivos pelos fabricantes tem sido um fator substancial na dificuldade para garantir a segurança. A falta de conhecimento dos usuários (por exemplo, muitos usuários usam senhas pequenas e frágeis) permite ao atacante uma grande chance de sucesso [67].

Para finalizar, todos os quadros de gerenciamento e controle são passados em texto limpo, ou seja, não são criptografados [67]. Essa deficiência, mesmo no WPA2, permite que

ataques DoS tenham relativo sucesso. Segundo o [104], está em desenvolvimento a versão denominada IEEE 802.11w[47] que aparentemente se propôs criptografar apenas os quadros de gerenciamento. Dessa forma, se não forem criptografados os quadros de controles, ainda assim será possível a criação de ataques DoS explorando técnicas de controle do acesso ao meio de comunicação, que segundo [206] é o local onde são encontradas a maioria dos problemas atuais.

A cada nova descoberta para quebra da segurança em redes sem fio, surge também uma ferramenta que é disponibilizada na WEB para quem desejar. Essas ferramentas, muitas com alta capacidade de atuação, se transformam em verdadeiras armas de guerra contra qualquer rede, seja grande ou pequena. Por isso, é de suma importância a existência de sistemas evoluídos para ajudar na segurança das redes e estações.

2.5 Sistemas de Detecção de Intrusão

Com o advento da Internet, o conhecimento tornou-se accessível para qualquer pessoa e com certa facilidade. Não há controle rígido sobre quando, como e quem pode ter acesso a certos tipos de informações. Assim, muitos tipos de ataques sobre redes (inclusive sem fio) estão disponíveis, com apresentação de tutoriais e manuais, além das próprias ferramentas para aplicação. Fica claro, dessa forma, que o uso do conhecimento não abrange apenas a capacitação e o desenvolvimento das pessoas, mas também é utilizado para infringir regras e prejudicar a outros. Portanto, qualquer sistema/rede precisa de um mecanismo de segurança capaz de atuar como um guardião do sistema [127].

Os IDSs são responsáveis pela detecção de eventos anormais ocorridos num determinado ponto e tempo no sistema/rede. Pela grande quantidade de incidentes de segurança relatados pelo mundo [39], os IDS se transformaram em um componente fundamental para segurança computacional. Nas seções subseqüentes são apresentadas algumas características importantes dessas ferramentas e sua aplicação sobre redes sem fio.

2.5.1 Definições e Taxonomia

Denning [61] definiu detecção de intrusão como sendo um processo de monitoração de eventos de atividades computacionais, seja em um sistema ou rede, analisando-os a procura de

[47] Mais informações podem ser encontradas no sitio:
http://standards.ieee.org/announcements/2009/pr_802.11w.html

anormalidades que possam comprometer a segurança computacional. A autora foi uma das pioneiras nos estudos com IDS e até os tempos atuais seus trabalhos são referência na área, principalmente pela idéia de um IDS executando em tempo real. Porém, Anderson [8], [9] é considerado o pioneiro na idéia de um sistema de auditoria automatizado.

Segundo [127], “detecção de intrusão é um processo de identificar e responder a atividades maliciosas direcionadas aos recursos dos sistemas computacionais. Esta definição introduz a noção de detecção de intrusão como um processo, que envolve tecnologia, pessoas e ferramentas”.

Ainda segundo Kruegel [127], pela quantidade de vulnerabilidades encontradas nos sistemas computacionais, sejam elas por falhas de fabricação, configuração ou tecnológica, associada ao crescente número de incidentes de segurança, e a morosidade encontrada nos administradores e usuários para aplicação de correções, faz com que muitos especialistas acreditem que os sistemas de computadores nunca serão seguros.

O célebre autor Andrews S. Tanembaum [188] acrescentou um fator importante para que um IDS tenha bom desempenho em suas atividades: é preciso que haja uma Política de Segurança (PS) que defina as regras (por exemplo, permissões de acesso) pelas quais o IDS irá comparar os eventos encontrados. Dessa forma, assume-se que a ação do intruso (isto é, um atacante) em um dado momento violou alguma regra da PS. Por isso, violações podem somente ser detectadas quando ações podem ser comparadas contra certa regra.

Muitos incidentes de segurança acontecem por abuso dos privilégios dos usuários do próprio sistema [39]. Mesmo assim, as estatísticas apresentam apenas os ataques que foram detectados, mesmo que o atacante não tenha tido sucesso em seu objetivo. Porém, é preciso entender que muitos incidentes nunca foram e, muitos poderão continuar sem serem descobertos. Para Northcutt [150] um fator preocupante pela falta de um IDS é a incapacidade de identificar o tipo do ataque e combatê-lo corretamente. Nesse caso, a falta da identificação permite que o ataque permaneça trazendo problemas para os usuários sem que o mesmo tome as devidas providências. Identificar uma atividade ilícita no sistema pode ajudar no aumento do conhecimento sobre aquele tipo de ataque e tornar possível o desenvolvimento de métodos para combatê-lo.

É preciso ainda compreender que um IDS não é uma ferramenta completa de segurança, mas faz parte de um conjunto delas (por exemplo, *firewalls* e *proxies*) e de acordo com [53] precisam cumprir com alguns requerimentos:

- **Precisão:** relacionado diretamente com a eficiência do IDS, a precisão diz respeito a qualidade com que o IDS detecta os eventos (isto é, quanto menos falsos alarmes melhor será sua eficiência);
- **Desempenho:** um IDS precisa operar em tempo real, ou seja, dentro de um período de tempo que permita ao mesmo tomar ações de prevenção para proteção do sistema;
- **Integridade:** é extremamente desejável que o IDS detecte todas as verdadeiras intrusões, porém é conhecido que um sistema não possui conhecimento suficiente para evitar futuros ataques contra si mesmo;
- **Tolerância a falhas:** um IDS não pode sucumbir perante qualquer ataque, deve ser robusto o suficiente para proteger a si mesmo;
- **Escalabilidade:** os IDS precisam ter capacidade de permitir novas estações/dispositivos na rede sem que isso prejudique sua capacidade de detecção. Infelizmente esse é um problema que, em tempo de execução, atinge muitos sistemas.

Para que pudesse existir um padrão de arquitetura para os IDS um grupo de trabalho concebeu um modelo chamado CIDF (do inglês, *Common Intrusion Detection Framework*) no qual agrupa um conjunto de componentes fundamentais que interagem entre si através de uma linguagem de especificação de eventos e comunicação CISL (do inglês, *Common Intrusion Specification Language*) [119]. A concepção do CIDF apresenta um IDS com quatro componentes principais agregados com regras especificas. São eles:

- **Geradores de Eventos (*E-box*):** principal função é colher eventos do meio externo ao CIDF e providenciar sua apresentação para o analisador (por exemplo, um analisador que gere eventos baseado no tráfego da rede);
- **Analisadores de Eventos (*A-box*):** responsável pelo refinamento das informações recebidas do gerador de eventos. Envia as informações analisadas de forma resumida para outros componentes – existem *A-box* que analisam dados provenientes de outro *A-box* e operam num alto nível de abstração;
- **Base de Dados de Eventos (*D-box*):** possibilita a criação de uma base de conhecimento para o IDS através do armazenamento das informações mais importantes para análise futura;
- **Unidades de Contra Medidas (*C-box*):** responsável pelas ações (passivas ou ativas) de acordo com cada evento ocorrido e analisado.

A figura 2.21 apresenta o fluxo ocorrido entre os quatros componentes. Pela figura, os eventos são coletados por dois *E-box* sendo, posteriormente, levados para outros dois *A-box* que enviam o resultado de suas análises para um terceiro *A-box* antes de enviar para o componente de contra medidas *C-Box*.

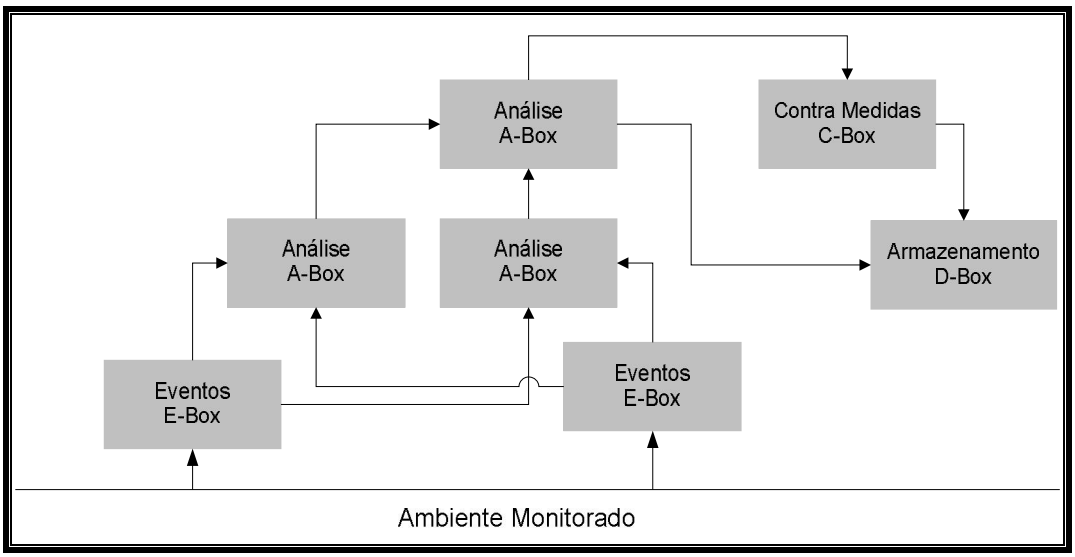


Figura 2.21 – Componentes da Padronização CIDF [127]. Podem existir outras configurações para este mesmo modelo.

Este modelo não é único, vários outros existem como o formado pelo IDWG (do inglês, *Intrusion Detection Work Group*), mas serve de guia inicial para qualquer IDS.

Existem duas estratégias genéricas na detecção de intrusos:

- **Baseadas em regras (assinaturas):** são mais comuns e mais fáceis de implementar. Por trabalhar com assinaturas, a eficiência de tais IDSs está diretamente relacionada a capacidade de atualização da base de assinaturas;
- **Baseadas em anomalias:** mais difícil de ser implementado, nesse caso, o IDS precisa de um tempo para "aprender" a atividade básica da rede ou sistema. Após o tempo necessário, que depende muito do fluxo de dados na rede, o IDS armazena as informações em uma base de dados e começa o monitoramento tendo como parâmetro os dados colhidos. Arbaugh [14] apresenta esse modelo como mais oneroso computacionalmente.

Mas, como estratégias de monitoramento, os IDS podem ser divididos em quatro categorias [127]:

- **Baseada em uma máquina (*host*):** monitoramento, no nível de sistema operacional, das informações geradas em registros de auditorias e *logs* do sistema;
- **Baseada em rede:** monitoramento dos pacotes que trafegam na rede (por exemplo, analisador de rede);
- **Baseada em aplicação:** monitoramento dos dados das aplicações em execução, incluindo *logs* dos eventos e estruturas internas à aplicação. É específico para determinadas aplicações;
- **Baseada em alvo:** nesse modelo, o IDS produz suas próprias informações. O monitoramento é feito através de *Hashes* Criptográficos, na verdade, é um modelo baseado na própria máquina (*host*), porém busca detectar alterações em objetos do sistema (isto é, alvos).

Dois modelos principais dividem os IDS: (i) baseado em redes e (ii) baseado em *host* (isto é, na estação). É comum, em grandes redes encontrar IDS híbridos, ou seja, atuando das duas formas paralelamente [47].

Algumas vantagens e desvantagens desses dois tipos:

- **Baseados em redes:** nesse tipo, os ataques são capturados e analisados através de pacotes de rede. Geralmente, coloca-se o IDS para atuar sobre a rede para monitorar todo o tráfego que faz parte de um determinado segmento, ou seja, todos os *hosts* que a este segmento pertencer, serão monitorados pelo IDS. Aplica-se nesse modelo o conceito de IDS distribuído. No caso, um conjunto de sensores é espalhado pela rede e essas unidades monitoram o tráfego e enviam os dados ao console central. Geralmente as unidades rodam em "*stealth*[48]", dificultando a localização pelo atacante.

  Segundo Bace [15], as vantagens existentes nesse modelo estão no pouco impacto sobre o desempenho da rede. Funcionando em modo passivo, eles apenas escutam a rede e são muitos seguros e praticamente invisíveis ao atacante. As desvantagens, dessa maneira, estão (i) na dificuldade do IDS lidar com equipamentos de distribuição (por exemplos, comutadores), (ii) na falta de capacidade de reconhecer se um ataque foi bem sucedido ou não e (iii) na dificuldade de lidar com pacotes fragmentados ou mal formados;
- **Baseados em *Hosts*:** atuam em computadores individuais, coletando dados sobre o sistema, arquivos, usuários, entre outras que possam ser úteis. Nesse caso, os IDS

[48] A tradução mais apropriada para esta palavra do inglês é discrição, ou seja, o IDS tem a capacidade de execução sem que seja detectado

podem analisar as conseqüências de um ataque, pois têm acesso direto as informações. Alguns suportam gerenciamento centralizado e outros podem gerar informações compatíveis com os sistemas de gerenciamento de rede.

Como vantagens, Bace [14] afirma que esses IDS podem monitorar eventos locais de um *host*, podendo detectar ataques que não poderiam ser detectados por um IDS de rede. Podem operar em ambientes de rede onde o tráfego é criptografado, capturando a informação antes de ser criptografada e depois de descriptografada. Pode ajudar a detectar '*Trojan Horses*[49]' ou outros tipos de ataques que envolvam problemas de integridade nos programas. Mas assim como os IDS de rede, também possuem desvantagens. A primeira é a dificuldade proporcionada ao administrador pela necessidade de configuração para cada *host* que receberá o IDS. A segunda está na possibilidade de ser descoberto pelo atacante do *host*, nesse caso, o atacante pode desabilitá-lo. Como reflexo de sua execução na estação, pode consumir muito recurso do *host*, ocasionando perda de desempenho [15].

De maneira geral, um IDS é capaz de oferecer alguns serviços de vital importância para uma rede/sistema [14], [127]. Tais serviços podem ser vistos da seguinte forma:

- **Identificação de tráfego:** um aplicativo IDS deve ser capaz de identificar corretamente a natureza da invasão ou do tráfego, incluindo as portas e os endereços de origem e destino;
- **Aplicação nos registros e definição de limites:** os IDS trabalham com definições de eventos. Caso seja excedido um limite, o IDS enviará alerta e/ou registrará o comportamento. Um IDS pode ampliar sua capacidade de registro colocando informações adicionais em arquivos de registros ou base de dados;
- **Alertas:** recursos de envio de mensagens para o responsável pela rede;
- **Configuração do sistema:** possibilidade de obter uma imagem instantânea da rede ou sistema operacional, e enviar alertas se ocorrer algum evento anormal.

Porém, qualquer IDS pode ser afetado por um grave problema durante sua execução: os alarmes falsos. Definidos como erros de identificação, podem ser divididos em dois tipos [150]:

- **Falsos negativos:** eventos intrusos considerados pelo IDS como não anômalos. Considerado o pior caso, pois libera acesso ao atacante na rede ou sistema;

[49] Softwares construídos para dar acesso não autorizado à máquina da vítima. Em português, significa cavalo de tróia, presente grego enviado a tróia. O atacante envia o programa disfarçado com alguma forma apelativa e chamativa. Dessa forma, a vítima aceita e o programa é instalado sem que a mesma saiba

- **Falsos positivos:** eventos legítimos que são considerados intrusos. Pode acontecer negação de algum serviço para algum aplicativo, porém não são considerados tão problemáticos quanto os falsos negativos, pois não deixa brecha de segurança;

Os falsos alarmes são uma medida de eficiência para os IDS e, mesmo que o ambiente seja totalmente inóspito, é desejável que a sua ocorrência não interfira na capacidade e qualidade do serviço prestado.

2.5.2 Modelos de IDS Aplicados Sobre Redes IEEE 802.11

Conforme apresentado na subseção 2.3, os problemas com as redes sem fio são relativamente grandes e novos quando comparados às redes cabeadas e, por isso, a aplicação de sistemas de detecção sobre esse ambiente apresenta características diferenciadas e complexas. Por isso, os IDS não podem atuar apenas nas camadas altas da cadeia de protocolos, mas sim nas camadas mais baixas como enlace e física, pois precisam analisar os três tipos de quadros do padrão [105]: (a) dados, (b) controle e (c) gerenciamento.

Outro importante aspecto para os WIDS (do inglês, *Wireless Intrusion Detection Systems*) é a necessidade de monitorar todos os canais disponibilizados pela tecnologia sem fio. Há ainda a necessidade de saber exatamente onde é o melhor local (isto é, local físico) para o WIDS evitando dessa forma que algum ponto da rede fique desguarnecido. Um WIDS deve ser independente de qualquer dispositivo da rede e ter capacidade de atuar em modo passivo e ativo [54].

A estrutura básica para aplicação nas redes sem fio não é muito diferente daquela apresentada na subseção 2.4.2. Porém, um bom WIDS precisará trabalhar com quadros (isto é, diferente dos IDS para redes cabeadas que trabalham com pacotes), sendo necessário ter um (i) mecanismo de leitura da rede/sistema à procura de eventos anormais, (ii) um analisador dos eventos encontrados, (iii) um mecanismo de reação e (iv) um mecanismo para armazenar informações sobre os eventos. Porém, estudos mostram que é preciso ir mais além, pois, alguns tipos de ataques, principalmente os DoS usando a rádio freqüência (RF) geram a necessidade de um novo módulo para os IDS: um analisador de espectro. Segundo Hurley [103] ainda não existem ferramentas que integrem totalmente um IDS com as ferramentas de análise de espectro.

Alguns IDS já previamente conhecidos do ambiente cabeado desenvolveram módulos para atuação sobre as redes sem fio, são eles: (1) Motorola Airdefense, (2) IBM Internet

Security Systems, (3) Security Works iSensor, e os modelos de código aberto (4) Snort-Wireless e (5) WIDZ.

2.5.3 Os Desafios para os IDS

Com a crescente revolução tecnológica, vários desafios são colocados a prova para os IDS. Silveira [179] constatou que as novas tecnologias, principalmente aquelas criadas para aumentar a segurança e o desempenho nas redes, têm ocasionado problemas para os IDS. Algumas tecnologias, por serem pouco utilizadas acabam por deixar brechas de segurança e desafiam o uso dos IDS quando agregadas às redes. O autor cita algumas tecnologias como ATM e FDDI, além dos problemas causados pelo uso da criptografia e VPNs.

Além destes desafios tecnológicos, dois paradigmas dos IDS encontram muitos desafios: (i) a automatização e (ii) a detecção de novos tipos de ataques. No primeiro, automatizar um IDS pode ser uma árdua tarefa [61]. É preciso ter cuidado com os falsos alarmes nas decisões tomadas pelo sistema. Porém, nenhum IDS está livre de falsos alarmes e por isso, as decisões precisam ser coerentes para não ser ele mesmo (isto é, o IDS) o causador de problemas na rede/sistema. Para o segundo caso, a qualquer momento pode surgir um novo tipo de ataque e, um bom IDS precisa ser capaz de detectar qualquer ataque, seja ele conhecido ou não. Essa não é uma realidade para os IDS clássicos – por ser baseados em assinaturas, muitos não conseguem atuar quando encontram um ataque para o qual não exista assinatura – e por isso, estudos têm sido desenvolvidos buscando inspirações em outras ciências (por exemplo, a biologia) a fim de desenvolver técnicas “inteligentes” de detecção de intrusão, sejam elas por anomalia ou por mau uso.

2.6 Considerações Finais Sobre Segurança Computacional Aplicada às Redes IEEE 802.11

Este capítulo apresentou (i) os aspectos básicos sobre as redes sem fio, (ii) os problemas de segurança e algumas técnicas de ataque que usufruem desses problemas e (iii) um mecanismo de segurança denominado IDS.

Conforme apresentado, muitos problemas envolvem as redes sem fio e muitas técnicas são propostas para quebrar a segurança. Com os recentes sucessos na quebra de chaves WPA e possivelmente do WPA2, é fato real a busca constante por falhas nos protocolos e no padrão. Infelizmente não existem apenas pesquisadores bem intencionados fazendo busca,

uma quantidade indeterminada de potenciais atacantes está constantemente em busca de uma chance para quebrar a segurança e aplicar toda sorte de atividades virtuais ilícitas [103].

Como os administradores de redes/sistemas e, principalmente os usuários, não tem capacidade de monitorar a todo instante seus sistemas, mecanismos de segurança avançados, com certo nível de “inteligência”, são essenciais para que futuros novos tipos de ataques não comprometam informações confidenciais nem prejudiquem o uso das mesmas.

No Capítulo 3 e 4 serão apresentadas duas técnicas usadas neste trabalho para mitigar os problemas a que estão expostos os WIDS.

Capítulo 3

Neste capítulo são apresentados (i) os conceitos do sistema imunológico humano (do inglês, *Human Immune System* – HIS), (ii) a Teoria do Perigo (do inglês, *Danger Theory* – DT) de Matzinger e, (iii) os sistemas imunológicos artificiais (do inglês, *Artificial Immune Systems* – AIS). No caso "(i)", são descritas as principais características das teorias imunológicas que guiam as primeiras versões dos AIS. Em contrapartida, no caso "(ii)", é discutida uma das mais novas e polêmicas teorias imunológicas. A DT tem sido utilizada como principal fonte de inspiração para as versões mais recentes dos AIS. Por último, após a descrição das peculiaridades do HIS e da DT, são apresentadas as principais características dos AIS.

Sistema Imunológico Humano, a Teoria do Perigo e os Sistemas Imunológicos Artificiais

O HIS é o principal agregado de agentes do corpo responsável pela defesa do organismo contra agentes externos, conhecidos como bactérias, vírus e patógenos. Segundo estudo apresentado em [201], esses seres vivos microscópicos constituem quase oitenta por cento de tudo o que é vivo no mundo, e no corpo humano há praticamente cem trilhões de bactérias contra dez trilhões de células. Como uma de suas características principais, o HIS consegue conviver em equilíbrio dinâmico com esses seres e, quando acontece um processo que desestabiliza essa convivência, entram em cena as células controladoras que irão combater o agente desestabilizador até o ponto em que o sistema (isto é, o corpo) volte ao equilíbrio novamente – o que significa, na maioria dos casos, a eliminação do microorganismo causador do problema.

Portanto, a imunologia consiste no estudo dos mecanismos pelos quais um organismo tem capacidade de reconhecer, neutralizar, metabolizar e eliminar as substâncias heterólogas[50], assim como tornar-se resistente a uma nova infecção pelo mesmo organismo invasor. Suas características de defesa atraem muitos estudos que buscam inspirações para aplicação em sistemas de segurança computacional.

Neste capítulo, é feito um breve resumo do sistema imunológico humano mostrando (i) suas características principais e os conceitos básicos de seu funcionamento, e (ii) a nova

[50] De origem diferente.

visão de acordo com a Teoria do Perigo proposta por Matzinger [136], [137], [138]. Em seguida, (iii) é descrita a representação simbólica usada para o desenvolvimento dos Sistemas Imunológicos Artificiais e a análise de alguns modelos propostos. Esse capítulo tem importância fundamental para o entendimento do projeto prático proposto para validação desta pesquisa.

3.1 Breve Histórico

O conceito de imunologia surgiu por volta do fim do século XVIII e início do século XIX através de Edward Jenner [110], [111], [112], [113]. Após estudar uma infecção bovina [110], Jenner usou partículas do vírus da varíola da vaca como forma de estimular no corpo humano a criação de células de combate. Com os anticorpos no corpo, o combate à varíola humana foi mais eficaz e ajudou a diminuir o grande número de mortos na Europa naqueles séculos. Surgiu dessa forma o termo "vacinação" (isto é, o termo "*vaccinia*" é originado do latim "*vacca*" em homenagem aos experimentos de Jenner), em que consistia contaminar uma pessoa com um vírus mais fraco (da vaca) para que o corpo pudesse impedir a contaminação com outro mais forte (o humano). Posteriormente, o mundo conheceu um dos mais influentes pesquisadores da área – para muitos, também é visto como o pai da imunologia – e fundador de um instituto que recebeu o próprio nome: Louis Pasteur[51]. Pasteur, juntamente com Koch, lançou a idéia de que as doenças epidêmicas eram causadas por microorganismos e que podiam ser evitadas [30]. Na mesma época, um estudo de Ilya Ilyich Mechnikov[52] [173], criador do termo Fagocitose, mostrou através do uso de larvas transparentes de estrela-do-mar e um simples acúleo[53] de uma roseira, que vinte e quatro horas após ter feito incisão sobre a larva havia um conjunto de células da larva que cercavam a ponta do organismo invasor. Essa resposta ativa era desconhecida até então.

Em 1888, foi apresentada a primeira idéia de anticorpos por Emil Adolf von Behring e Kitasato Shibasaburō [119], proporcionando a soroterapia. Durante décadas, vários pesquisadores trabalharam para aperfeiçoar estas idéias. Contanto, a grande revolução aconteceu no final da década de 1950, quando Frank Macfarlane Burnet [32] apresentou a

[51] Mais informações sobre a vida e trabalho deste renomado pesquisador ver: http://pt.wikipedia.org/wiki/Louis_Pasteur
[52] Conhecido como o pai da imunologia, ganhou o prêmio Nobel da medicina em 1908. Biografia em: http://nobelprize.org/nobel_prizes/medicine/laureates/1908/mechnikov-bio.html
[53] Conhecido como "espinho"

teoria da Seleção Clonal, o que fez da imunologia um dos pilares da ciência médica, sendo uma das áreas de concentração de pesquisas [79].

3.2 Conceitos do Sistema Imunológico Humano (HIS)

Composto por um conjunto de órgãos, células e moléculas, o HIS é responsável por defender o organismo contra ataques externos provindos de uma infinidade de corpos estranhos como vírus, bactérias, e outros parasitas conhecidos como patógenos [108], [109].

Utilizando detecção distribuída, o HIS trabalha para resolver o problema de distinguir entre o próprio (*self*), que são os elementos do organismo, e o não-próprio (*nonself*), que são os elementos estranhos ao organismo. Tal mecanismo é conhecido como seleção negativa [45], [151], [204].

O HIS possui duas linhas de ação, uma rápida e outra mais lenta, definidas como sistema inato e sistema adaptativo, respectivamente. Os macrófagos e granulócitos [16] são os responsáveis pela imunidade inata, enquanto que os linfócitos, pela adaptativa [1]. A figura 3.1 ilustra esses dois sistemas.

Resumidamente, os macrófagos e neutrófilos são capazes de ingerir vários microorganismos e partículas antigênicas [152], enquanto que os linfócitos são produzidos de acordo com o contato com certos agentes infecciosos, os quais o sistema inato não é capaz de combater por si só [174].

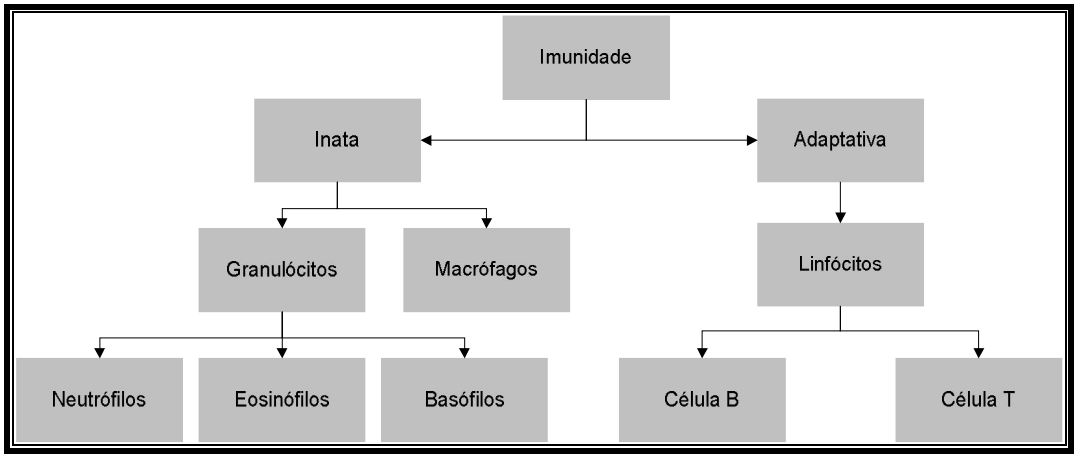


Figura 3.1 – Mecanismos de defesa e seus principais mediadores, Fonte [55].

Uma das partes de maior interesse, e muito importante para esse estudo, é a capacidade de desenvolver memória imunológica através das células, ou seja, quando um

indivíduo é infectado, as células de combate são produzidas e, após a eliminação do invasor, as células continuam no corpo do indivíduo e mantêm uma memória que é capaz de reconhecer o mesmo invasor, caso o indivíduo seja exposto ao mesmo [182], [1].

Geralmente, essas células ficam inativas no corpo e somente terão atividade caso haja interação com algum estímulo antigênico. Segundo [38], Dreher [64] e Timmis [193], qualquer substância que pode estimular respostas específicas do HIS é chamada de antígeno. Na verdade, os patógenos produzem os antígenos, que são moléculas capazes de iniciar uma resposta imune [180].

Existem dois tipos de células de combate no organismo: (i) linfócitos B (células B) e (ii) linfócitos T (células T) que serão detalhados em seção posterior, por conter características importantes para esse estudo.

Um importante diagnóstico é que somente os vertebrados possuem linfócitos, que evoluíram para proporcionar maior defesa para o corpo, juntamente com o sistema inato [124].

3.3 Anatomia do Sistema Imunológico

O HIS atua de forma distribuída e organizada, onde cada célula tem sua função específica agindo de forma organizada e distribuída. O sistema é dividido em duas frentes: (i) órgãos primários (centrais) e (ii) secundários (periféricos). Nos primários, as células são produzidas e sofrem a maturação, enquanto que nos secundários são expostos aos estímulos antigênicos. Esses órgãos recebem o nome de linfóides e, segundo de Castro [55], fazem parte dos primários:

- **Medula Óssea:** responsável pela produção de linfócitos B e as células-tronco dos linfócitos T;
- **Timo:** responsável pela maturação dos linfócitos T, que são células que migram da medula óssea para se transformar em células T;

E dos órgãos secundários:

- **Amigdalas e Adenóides:** que fazem parte do sistema imune associado a mucosas e intestino;
- **Os linfonodos:** que atuam como regiões de convergência de um sistema de vasos que coletam fluidos extracelulares dos tecidos, fazendo-os retornar ao sangue;

- **Apêndice e Placas de Peyer:** responsáveis pela proteção do sistema gastrointestinal;
- **O Baço:** local onde ocorre o combate dos linfócitos contra os agentes infecciosos da corrente sanguínea;
- **Vasos linfáticos:** transportam a linfa para o sangue e órgãos linfóides.

A clássica figura representando esse esquema é apresentada pela figura 3.2. Para Dasgupta [49] é difícil apresentar uma figura (idéia) concisa tal como um sistema complexo se muitos mecanismos ainda não são totalmente conhecidos (isto é, a imunologia é uma ciência razoavelmente nova e ainda não foram totalmente elucidadas todas as questões sobre como o HIS atua para proteger o corpo humano).

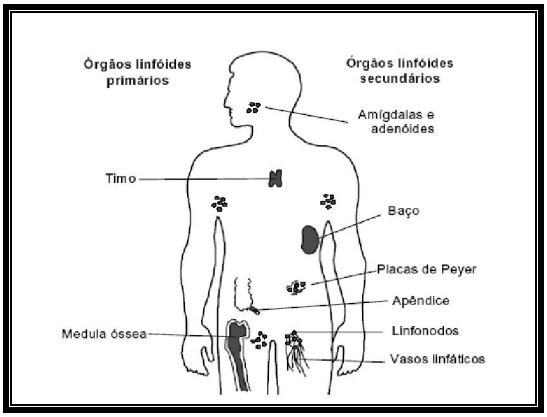


Figura 3.2 – Fisiologia do HIS mostrando de um lado os órgãos primários e do outro os secundários. Fonte [55].

Hofmeyr [100] e Janeway et al. [108], acompanhando as definições de Rensberger [165], definiu o HIS como um sistema de camadas no qual pertencem: (a) barreiras físicas, (b) barreiras bioquímicas, (c) sistema imune inato e (d) sistema imune adaptativo, conforme pode ser visualizado na figura 3.3.

Nesse caso, a pele e o sistema respiratório funcionam como uma barreira física contra os invasores. Na barreira fisiológica, fatores como o pH, temperatura da pele, salivas, suor, lágrimas e ácidos estomacais são importantes, pois inibem a sobrevivência de corpos estranhos. O sistema imune inato é aquele responsável pela primeira defesa do organismo e envia dados para o sistema adaptativo, que inicia sua atuação com certo tempo de atraso.

Nos trabalhos [108], [33] e [194] é possível encontrar maiores detalhes sobre a maioria dos mecanismos conhecidos do HIS.

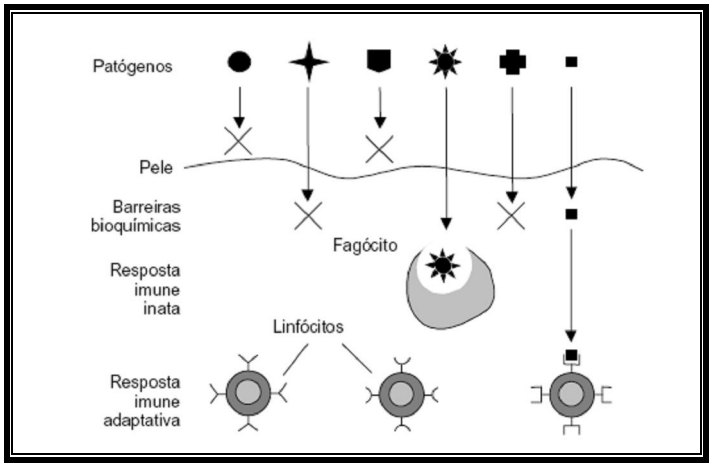


Figura 3.3 – As quatro camadas do HIS. A primeira defesa do organismo humano é a pele. Fonte [55].

3.4 O Sistema Adaptativo

A resposta adaptativa foi, desde o início, um dos pontos do HIS que mais despertou interesse. Principalmente porque ela é responsável pelo reconhecimento (isto é, identificação) e seletivamente pela eliminação das células heterólogas do corpo. Ela atua em conjunto com o sistema inato através de um processo conhecido como fagocitose[54] (isto é, as células fagócitas pertencem ao sistema inato e elas são responsáveis por ativar, ou não, uma resposta imunológica) [176].

Segundo SHAH [176], o principal fator do sistema adaptativo é o gerenciamento dos linfócitos. Para isso, existe um rigoroso controle que vai desde o nascimento até a morte do linfócito. Conforme descrito na seção 3.3, dois tipos principais de linfócitos formam a maioria de sua população. As células B e T são produzidas na medula óssea (*bone marrow*) e possuem dois estados: (i) imaturo e (ii) maturo. No primeiro caso, as células são inoperantes, incapazes de qualquer atividade. Por isso, existe um processo denominado maturação no qual as células serão treinadas através da seleção negativa (isto é, serão treinadas para distinguir o que é próprio do não-próprio). A maturação das células B e T acontece de forma diferente, assim como suas atividades [152]:

- **Célula B (linfócito):** sua principal função é combater patógenos extracelulares e seus produtos através da produção de anticorpos. O processo de maturação dessas células acontece na própria medula óssea;

[54] Processo utilizado pela célula para englobar partículas sólidas, que lhe irão servir de alimento. Mais informações podem ser encontradas em: http://www.sobiologia.com.br/conteudos/Corpo/fagocitose.php. Último acesso: 14/03/2010.

- **Célula T (linfócito):** possui várias funções, entre as quais se destacam (a) interação com células fagócitas, (b) controle da produção de células B, e (c) destruição de células infectadas. Diferente das células B, a célula T é levada para o timo para maturação.

O processo de maturação é rigoroso. Se, durante a maturação, uma célula reconhecer erradamente uma célula própria como não-própria, ela será eliminada do corpo. Isso impede que o corpo sofra um processo denominado auto-imunidade (isto é, doenças auto-imunes, onde as células combatem as células do próprio corpo) [194], [84], [177].

De acordo com [55], existem dois tipos de resposta adaptativa:

- C**elular:** realizada pelas células T, que atuam apenas contra um tipo de patógeno (isto é, cada célula T está relacionada a um tipo de patógeno) e, a cada novo contato se tornam mais eficientes. São também responsáveis pela ativação das células fagócitas;
- **Humoral:** é uma resposta mediada por anticorpos, ou seja, proteínas formadas por plasmócitos (isto é, linfócitos B). Os anticorpos são produzidos para neutralizar e eliminar o antígeno que estimulou sua produção. Para essa resposta existe ajuda da célula Th (*helpers*) (isto é, células que regulam a atividade da resposta) e do SMF (fagócitos mononucleares) que são responsáveis pela apresentação do antígeno à célula Th.

Após o encontro com o antígeno, as células T e B passam por um processo de expansão, ou seja, a quantidade de células cresce para que possa ser feito o combate à célula estranha. Esse processo é feito através de reprodução assexuada (isto é, a célula é reproduzida a partir da célula de origem (pai) sem interferência de nenhuma outra) conhecida como seleção clonal. Tal processo permite ao HIS aumentar o poder de resposta contra o antígeno. É também através desse processo que surge a célula de memória. Esta célula tem grande importância na resposta adaptativa e, através dela o HIS poderá apresentar uma resposta mais rápida se houver novo contato do corpo com aquele antígeno. Por já conhecer o antígeno, a célula de memória irá ativar a resposta do HIS com mais eficiência e em menor tempo [160] [182].

3.5 A Teoria da Rede Imunológica

No trabalho [116] o HIS foi mostrado como uma grande rede na qual são interligados os anticorpos, formando uma cadeia de reconhecimento. O principal objetivo desta teoria era explicar como ocorria a tolerância imunológica.

Segundo Jerne [116], a idéia era que o HIS poderia criar anticorpos com novas estruturas moleculares, mesmo sem a exposição a qualquer antígeno.

Na teoria da rede imunológica (*Immune Network Theory* – INT), os anticorpos possuem duas estruturas fundamentais: os paratopos (para a região de ligação do anticorpo, também chamada de região-V) e os idiotopos (a parte do anticorpo que é reconhecida por outros anticorpos).

Dessa forma, o HIS não só reconhece os antígenos, mas ele também pode reconhecer partes (regiões) das moléculas do anticorpo, no caso, os idiotopos.

Para Jerne [116], os idiotopos funcionam como antígenos próprios e promovem a ligação uns aos outros, formando as cadeias de reconhecimento ou rede imunológica, conforme ilustra a figura 3.4. De acordo com o autor, o estímulo externo provocará uma perturbação na rede, em algum ponto. Dessa maneira, após sofrer uma perturbação local e combatê-la, a rede irá se auto-organizar em uma nova configuração. Nesse caso, a manutenção da memória não está mais nas células de memória e sim distribuída pela rede [116].

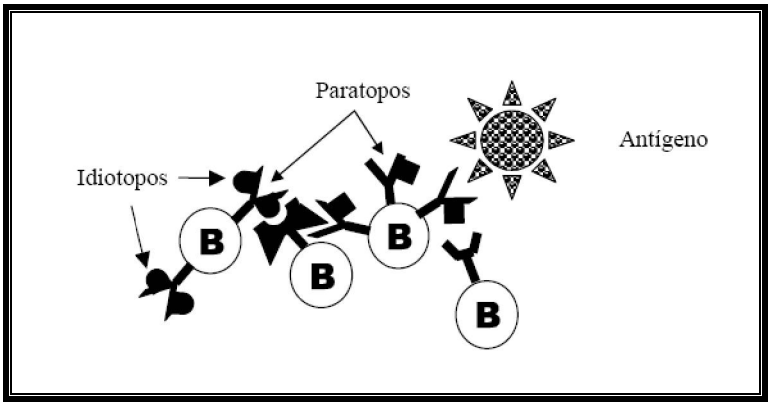


Figura 3.4 – Exemplos de como se formam as redes imunológicas. Nessa ilustração as células B aparecem formando uma rede através da interconexão. Fonte [139].

Leandro de Castro [55] também interpreta a teoria apresentando-a como um modelo cognitivo análogo ao sistema nervoso central (isto é, sistema neural). Para o autor, baseado no

estudo de [190], Jerne [116] é considerado o verdadeiro autor do modelo cognitivo do HIS. Dessa forma, ele apresenta a teoria da rede imunológica em forma de três características principais:

- **Estrutura:** é responsável pela descrição dos padrões de interconexão entre os componentes celulares e moleculares, sem dar ênfase às conseqüências dessas interações;
- **Dinâmica:** trabalha as interações dos diversos componentes do sistema;
- **Metadinâmica:** representa o potencial que o sistema imunológico possui de introduzir diversidade através da contínua produção de novos anticorpos, garantindo sua capacidade de combater novos antígenos [116].

Para os estudiosos [55], [139] e [200], o modelo de rede imunológica tem boas características para desenvolver ferramentas computacionais. Algumas delas, inclusive, têm sido desenvolvidas e testadas em algumas áreas de atuação, que podem ser vistas nos trabalhos de [58], [56] e a mineração de dados, [207] e a comunicação entre servidores WEB, [27] e os problemas paralelos, entre outros projetos.

No geral, as características com potencial para aplicação de modelos e projetos de redes imunológicas podem ser citadas como: tolerância a falhas, distribuição da informação, descentralização do gerenciamento, reconhecimentos de padrões, aprendizagem e memória, tolerância ao próprio [55].

Outros trabalhos que apresentam os conceitos e modelos de rede imunológica podem ser visto em [99], [41], [169], [23], [42], [46], [183], [31], [199], [22], [200], [23], [24].

3.6 A Teoria do Perigo

Esta teoria é considerada uma das mais recentes da imunologia. Ela é base da ferramenta de detecção de intrusão construída para esse trabalho. Portanto, nesta seção serão apresentadas suas principais características, detalhadamente.

A partir da década de 1970, imunologistas passaram a estudar outros caminhos para a definição de como o HIS trabalha. Até aquele momento, os estudos eram concentrados muito mais no sistema adaptativo que no sistema inato. Mas, através da análise do processo das doenças auto-imunes (por exemplo, esclerose múltipla), novas questões surgiram sobre a veracidade da teoria da seleção negativa. Segundo Janeway et al. [108], os pesquisadores, através do estudo do processo de vacinação por inoculação, descobriram que era necessário um sinal para ativar as células T do HIS: o segundo sinal (*Second Signal*).

Concentrando as pesquisas sobre o sistema inato, eles propuseram o modelo de infecção não-próprio (*infections nonself model* – INM), o qual apresentava a idéia de um modelo de dois sinais. Dessa forma, somente os antígenos apresentados com moléculas co-estimulatórias (*co-stimulatory molecules* – CSM) podem ativar as células T. Os autores [108] trabalharam principalmente com as células que atuam nos nódulos linfáticos, entre as quais se destacam as células dendríticas (*Dendritic Cells* – DC) e as células T. De acordo com o INM, as DCs, quando expostas aos sinais, podem formar uma classe de moléculas conhecidas como padrão molecular associado ao patógeno (do inglês, *Pathogen Associated Molecular Patterns* – PAMP). Na verdade, PAMP são sinais (podem ser chamados de moléculas) que são produzidos apenas pelos patógenos, que nesse caso podem ativar a resposta imunológica através das células T [176].

Janeway et al. [108] também descobriram que os invasores não podiam ser reconhecidos sem que estivessem acompanhados de um sinal PAMP, pois era exatamente este sinal que definia se aquele antígeno era próprio ou não-próprio. Segundo Greensmith et al. [83], esta teoria ajudou a explicar a necessidade de agentes estimulatórios para o processo de imunização (conhecido como vacinação). Mas, faltava a explicação sobre como eram desencadeadas as doenças auto-imunes.

Assim, em 1994, Matzinger [136] propôs uma nova teoria, conhecida como a Teoria do Perigo (DT), para explicar a forma como ocorre a ativação da resposta imunológica pelo corpo humano.

3.6.1 Aspectos Básicos

A DT apresentou modificações substanciais na forma de atuação do HIS. Segundo Matzinger [137], o HIS procura prevenir a destruição e não se ocupa diretamente com a distinção dos elementos próprios dos não-próprios. Através de estímulos de sinais provenientes de células (sejam elas do organismo ou invasoras), existe uma célula específica responsável por apresentar os antígenos à célula de combate. Dessa forma, sinais podem ser perigosos ou não e a forma como serão processados esses sinais é que definirá o grau do perigo.

Os sinais de perigo são provenientes da idéia dos sinais PAMPs [108]. Porém, Matzinger alterou a formulação anterior, propondo que os sinais não vêm somente de fontes

exógenas[55], mas também de fontes endógenas. Portanto, são produzidas no próprio organismo. Para esta formulação, a pesquisadora introduziu o conceito da diferenciação da morte para uma célula no qual existem dois tipos: (i) por necrose e (ii) por apoptose. O primeiro está relacionado à morte ruim, ou seja, aquela em que a célula foi atacada por um patógeno. Para o segundo, porém, o sinal teve origem na morte natural de uma célula do organismo. Grandes concentrações de sinais apontando necroses ativam o HIS, enquanto que os sinais apontando apoptoses geram supressão imunológica.

Como agentes centrais do sistema inato, as DCs são responsáveis por coletar e processar os sinais encontrados no organismo. Após o processamento, elas irão apresentar o resultado juntamente com o antígeno para a célula T, que será ativada ou não, dependendo apenas da concentração de sinais com os quais o antígeno foi coletado [137].

3.6.2 As Células Dendríticas

Conhecidas como uma família de células chamada macrófagos, a principal função das DCs é limpar o organismo de possíveis sujeiras (partículas de células) deixadas pelas células do próprio corpo ou então por células heterólogas. Elas existem em três diferentes estados: (i) imaturo, (ii) semi-maturo e (iii) maturo. Uma DC nasce no estado imaturo e permanece assim até que a concentração de sinais recebidos a façam mudar de estado [133].

Ainda no estado imaturo, as DCs percorrem livremente o corpo coletando antígenos e sinais, através da grande quantidade de receptores encontrados em sua superfície. Após a coleta do antígeno, a DC irá calcular a potência dos sinais coletados no momento em que o antígeno foi coletado e, caso atinja algum dos limiares pré-estabelecidos, poderá mudar para um dos outros dois estados e migrar para o nódulo linfático onde se encontram as células T.

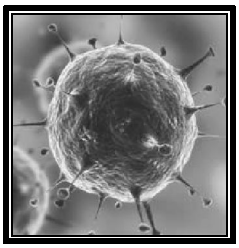


Figura 3.5 – Célula Dendrítica e seus receptores, que servem como coletores de sinais.

[55] As células exógenas são antígenos que são coletados especialmente pelas células apresentadoras (DC), mas, as células endógenas podem ser apresentadas por qualquer célula nucleada do organismo. Ver mais em: http://www.camposecarrer.com.br/default.asp?secao=det.asp&codigo=133&tipo=4

Existem três tipos de sinais que uma DC coleta: (a) PAMP, (b) sinal de perigo (*Danger Signal* – DS) e (c) sinal seguro (*Safe Signal* – SS). PAMPs, conforme explicado anteriormente, são sinais que podem ser entendidos como sinais de ataque já acontecendo. Por outro lado, SS são sinais normais, ou seja, sinais provenientes de células do próprio organismo (por exemplo, quando uma célula morre naturalmente ela emite um sinal que é captado pela DC) [93]. Os DSs podem representar sinais de problema ou de tranqüilidade. Este sinal tem bastante importância para o processo. Afinal, pode servir como ponto de equilíbrio do sistema por trabalhar como um sinal de alerta. A exposição das DCs aos sinais causa a produção de certo tipo de molécula conhecido por citocina. É através destas moléculas que as células T podem ser ativadas ou sofrer supressão. Elas existem em dois tipos básicos: (i) *Interleukin*-12 para DC em estado maturo e (ii) *Interleukin*-10 para DC em estado semi-maturo [176], [93].

Para gerar moléculas do tipo “i”, as DCs precisam ser expostas a uma grande quantidade de sinais PAMPs e DSs, mas, poucos SSs. Após esse processo, as DCs irão migrar e mudarão morfologicamente seu estado de imaturo para maturo e produzirão as CSMs que são responsáveis por facilitar o processo de apresentação do antígeno, conforme mostrou [108]. Para “ii”, a produção acontece pela alta concentração de SSs coletados, em detrimento dos outros sinais. Nesse caso, os antígenos também são apresentados para as células T, porém em um contexto de tolerância antigênica no qual eles não têm capacidade de ativar uma resposta imunológica. Segundo Williams et al. [210], esse processo é importante para o HIS diminuir o gasto de energia com a produção de células e outras atividades de combate quando o invasor já foi vencido, além de prevenir o corpo contra as doenças auto-imunes.

Finalizando, Greensmith [88] definiu as DCs como células de apresentação de antígenos que têm o poder de controlar uma resposta do sistema adaptativo. De acordo com a saída do processamento dos sinais pelas DCs, o HIS saberá como responder apropriadamente à ameaça. Esta nova abordagem tem permitido a abertura de um novo paradigma para os pesquisadores, quando aplicada a área computacional, principalmente após descobertas de que a DT pode ajudar na resolução de alguns problemas clássicos da abordagem próprio/não-próprio [93].

3.7 Aspectos de Segurança Computacional do Sistema Imunológico

A analogia entre segurança e processos biológicos surgiu com Cohen [41], através da introdução do termo vírus de computador.

Entretanto, foi com os trabalhos [76] e [120] que surgiram as primeiras idéias ligando o HIS e a segurança de computadores.

Forrest e Hofmeyr [77] afirmam que o HIS é um dos sistemas de defesa biológico mais robusto que existe na natureza. Portanto, muitas características do HIS podem ser referenciadas numa analogia simples com sistemas computacionais, conforme pode ser visto na lista de definição apontada por [51]:

- **Correspondência de padrões (*pattern matching*):** através do processo de reconhecimento de antígenos e patógenos por afinidade (isto é, nas células do HIS existe uma grande quantidade de receptores que podem interagir com as células invasoras), o HIS possui um modelo de classificação bastante eficiente, haja vista a quantidade de microorganismos a que está exposto;
- **Aprendizado (*learning*) e memória (*memory*):** quando o HIS encontra com um tipo de invasor nunca visto antes, através do primeiro combate feito pelo sistema inato, precisará avisar ao sistema adaptativo que não possui células/anticorpos para combater aquele invasor. Nesse caso, o HIS irá produzir células específicas para combatê-lo. Para produzi-las, o HIS precisa reconhecer as características do invasor para que o combate seja eficiente. Após o reconhecimento, ocorre o processo de seleção clonal, que irá produzir muitas outras células iguais (clones) que ajudarão no combate. No final, quando finalizado o embate, muitas células/anticorpos foram mortas. Porém, algumas das células que sobreviveram serão escolhidas, num processo chamado seleção positiva (isto é, existe um patamar de detecção para cada célula, quando atingem este patamar são promovidas), como células de memória e terão um tempo de vida indeterminado no corpo. Esse processo agiliza o combate caso a célula encontre o mesmo invasor novamente. Esse processo é uma das características do HIS que permite ao mesmo ser nomeado: sistema inteligente. Segundo Russel e Norvig [172], um sistema para ser considerado inteligente precisa ter capacidade de memorizar o que ocorreu no passado e usar este conhecimento para melhorar o futuro. Essa técnica pode ser chamada de aprendizagem por reforço, segundo os conceitos de Sutton e Barto [185].
- **Tarefas distribuídas (*distributed tasks*):** tanto as células de memória, quanto as outras células do HIS não possuem um agente central. Todas trabalham de forma organizada e distribuída, cada qual com a sua atividade. As células são capazes de

tomar decisões localmente e de forma colaborativa sem ter que avisar qualquer elemento (isto é, órgão) central [55].

- **Auto-regulação (*self-regulation*):** quando o HIS realiza um combate, que pode ter grandes proporções, ele poderá produzir milhares de células até que o problema seja resolvido. Após este processo, o HIS realiza um procedimento de auto-limpeza, eliminando os restos das células mortas. Mas é preciso equilibrar novamente a quantidade de células imunológicas (isto é, se a quantidade de células imunológicas ultrapassarem certo nível e não recuar após o tempo necessário para o combate poderá causar doenças como a leucemia, na qual pode haver até 200 mil leucócitos por mm^3 de sangue, quando o normal é abaixo de 10 mil leucócitos por mm^3), para isso, no HIS existem as células Tk (*killer*) que eliminam as células B em excesso [64];
- **Tolerância (*tolerance*):** o HIS é capaz de co-existir em conjunto com microorganismos que podem ser invasores, até o momento em que estes são reconhecidos.
- **Diversidade (*diversity*):** constantemente o HIS gera bilhões de diferentes moléculas para reconhecimento de diferentes tipos de antígenos. Para testar, usa o processo de seleção clonal e hipermutação para gerar configurações diferentes para detecção dos antígenos conhecidos e também dos desconhecidos. O intuito é gerar moléculas cada vez mais aperfeiçoadas para explorar o ambiente – no caso, o corpo – como um todo, mas também de forma localizada, ou seja, num processo conhecido como "*exploration*" e "*exploitation*" [55];

Olhando sobre os aspectos dos IDS, Reis [164] afirmou que o papel do corpo humano "*é análogo ao de sistemas de segurança em computadores. Ambos precisam proteger uma determinada entidade, localizada em um ambiente hostil a falhas, contra a ação de atacantes*". Para exemplificar, a tabela 3.1 mostra tal analogia.

Tabela 3.1 – Analogia entre HIS e IDS [164].

Componentes do IDS	IS
Coletor de dados	Fonte de proteínas própria/não-própria
Sistema de filtragem	Processo de apresentação do antígeno
Base de dados de perfis	Conjunto de receptores gerados aleatoriamente
Detector de anomalias	Detecção não específica do fagócito
Executor da resposta primária	Resposta primária do sistema inato
Gerador de assinaturas	Produção de célula de memória
Gerador de respostas	Produção de anticorpos específicos

Base de dados de assinaturas	Conjunto de células de memórias específicas
Detector de mau uso	Detecção realizada pelas células de memória
Executor de resposta secundária	Resposta específica do sistema adaptativo
Console	Imunidade adquirida por meio de vacinas

Leandro de Castro [55] foi mais além e propôs uma analogia entre o HIS e as redes, que pode ser visto através da tabela 3.2.

Tabela 3.2 – Analogia entre HIS e um modelo de segurança proposto para redes de computadores [55].

IS	Segurança de Redes de Computadores
Próprio	Conjunto de pares de conexões TCP/IP que ocorrerem normalmente entre computadores
Não-próprio	Conjunto de conexões que não são normalmente observadas em uma rede local
Células Imunológicas	Detectores
Localidade no organismo	Máquina, ou host, de uma rede local na qual um conjunto de detectores é colocado em execução
Timócitos	Novos detectores gerados em cada conjunto de detectores
Línfócitos T maduros	Detectores que sobrevivem ao processo de seleção negativa
Nível de citocina	Limiar adaptativo
Competição pelo reconhecimento antigênico	Mensagem enviada via correio eletrônico por um operador
Molécula de MHC	Máscaras de permutação

Nestas duas tabelas, pode ser visualizado que todo o funcionamento do processo de segurança computacional, embutido nos modelos propostos, estava embasado na abordagem de próprio/não-próprio da seleção negativa. Somente após o trabalho de Aickelin et al. [3] através do projeto perigo (*Danger Project*) que novas propostas foram apresentadas, mudando, dessa forma, alguns aspectos das analogias até então conhecidas. Mais informações sobre a teoria do perigo aplicada em segurança computacional podem ser vistas no Capítulo 5.

3.8 Sistemas Imunológicos Artificiais

Os sistemas imunológicos artificiais (AIS) foram propostos através da observação das características do HIS que eram análogas às características dos sistemas computacionais [181]. Resumidamente, os AIS podem ser considerados modelos abstratos do HIS. Mas, necessariamente não é preciso que um sistema seja uma abstração completa do HIS para receber o título de AIS. Leandro de Castro [55] trabalhou essa idéia e, num estudo detalhado do HIS e suas representações computacionais, apresentou um novo conceito para a imunologia sobre o aspecto da engenharia. O trabalho é intitulado como engenharia

imunológica (*Immune Engineering* - IE) e as características principais podem ser vistas na tabela 3.3.

Segundo a abordagem feita pela IE, não é preciso desenvolver um sistema imunológico com todas as suas características para que possa ser aplicado na resolução de problemas. A IE é considerada um processo meta-síntese, onde se deve buscar a construção de modelos inspirados no HIS (isto é, deve ter alguma característica do HIS) com possibilidade de serem representados computacionalmente. Para isso, uma das principais atividades iniciais, na construção de um AIS, é a forma como serão representados os componentes que farão parte do sistema. De certa forma, isto influenciará na escolha do método de avaliação da interação (isto é, afinidade) entre eles, assim como a interação entre eles e o ambiente.

Tabela 3.3 – Características da IE segundo [55].

Característica	Definição
Unidade Básica	A célula é representada por uma cadeia de atributos, conexões e um limiar de afinidade
Interações com outras unidades	As células possuem conexões que a identificam (receptores) e permitem reconhecer outros elementos. Estas conexões podem ser ponderadas indicando o grau de interação como outros elementos
Atividade	A célula possui uma imagem interna do ambiente que é comparada com a informação recebida
Conhecimento	É armazenado nos pesos das conexões e nas cadeias de atributos de cada célula
Aprendizagem	Ocorre principalmente através da modificação das cadeias de atributos das células e de seus pesos associados
Limiar	Determina a ligação (isto é, reconhecimento) entre uma célula e o estímulo apresentado
Robustez	Escalonável, auto-tolerante, flexível e tolerante a falhas
Localização	As células podem se deslocar
Comunicação	Ocorre através do contato celular, representado por um conjunto de conexões
Estado	Concentração e/ou afinidade entre uma célula e seu estímulo
Controle	Uma reação adaptativa determina o tipo de interação dos componentes do sistema

3.8.1 Representação dos Componentes do Sistema

Esta é a parte crucial no desenvolvimento de um sistema imuno-inspirado. Representar de forma abstrata as várias células e órgãos do HIS não é uma tarefa simples e pode comprometer o resultado final [3]. Deve-se primeiro representar os componentes (isto é, células, antígenos, patógenos) em um determinado espaço e com formas plausíveis (por exemplo, uma estrutura de dados). A maioria dos HIS desenvolvidos utilizam os conceitos de

espaço de formas S (*Shape-Space*) proposto por Segel e Perelson [175], no qual é definido que todas as propriedades das moléculas receptoras (isto é, aquelas utilizadas para a interação com os antígenos, chamadas de afinidade imunológica) podem ser desenvolvidas usando estruturas de dados, assim como os próprios antígenos. Dessa forma, para avaliar a interação entre as moléculas e os antígenos podem ser usados cálculos como a medida da distância ou de similaridade utilizando (i) cadeia de atributos (*strings*) de reais, (ii) inteiros, (iii) binária ou (iv) simbólica. Geralmente, a representação pode ser vista como:

- **Antígeno:** Ag = (Ag_1, Ag_2,...,Ag_N); onde *Ag* representa os atributos e *N* representa o tamanho das cadeias;
- **Molécula (anticorpo):** Ab = (Ab_1,Ab_2,...,Ab_N), semelhante a representação do antígeno.

3.8.2 Tipos de Avaliações de Afinidade

Conforme visto, na representação usada para as moléculas e antígenos são utilizadas cadeias de atributos (vetores). Por isso, para que possa ser avaliada a interação dos indivíduos com o ambiente, assim como a interação entre eles, é preciso verificar qual é o espaço de forma que está sendo usado para representá-los. Alguns espaços de formas são comuns para os AIS [55]:

- **Real:** os valores encontrados nos vetores são do tipo real;
- **Inteiro:** somente valores inteiros são aceitos nesse espaço;
- **Hamming:** os vetores são compostos por um alfabeto finito de tamanho *k*;
- **Simbólico:** nesse caso, diferentes tipos de atributos podem compor um vetor. Para esse tipo, há a necessidade de pelo menos um atributo ser simbólico;

Definido em qual espaço de formas estão representados os vetores, alguns modelos específicos são utilizados para avaliação de interações ou o cálculo da medida de afinidade. Para cada espaço de formas, há pelo menos um tipo. Entres os mais comuns estão (a) distância euclidiana e de Manhattan para os vetores usando valores reais, (b) distância Hamming, r-bits consecutivos e r- *chunks* para o espaço de formas de Hamming.

É importante frisar que para a DT não há cálculo da afinidade utilizando algum espaço de formas, mas sim o processamento da concentração dos sinais recebidos pelas DCs e a apresentação do resultado, juntamente com o antígeno, às células T [93].

3.8.3 Algoritmos Imunológicos e a Segurança Computacional

Alguns modelos de algoritmos imunológicos podem ser encontrados na literatura através de modelos teóricos ou desenvolvidos. Entre eles: (i) o algoritmo de seleção negativa (*Negative Selection Algorithm* – NSA [76]), (ii) algoritmo de seleção clonal (*Clonal Selection Algorithm* – CLONALG [55]), (iii) algoritmo da rede imunológica (*Artificial Immune Network* – aiNet [56]) e (iv) algoritmo das células dendríticas (*Dendritic Cells Algorithm* – DCA [88]).

Como características comuns entre eles, destacam-se: (i) o conceito de repertório (isto é, população) de anticorpos, (ii) aplicados em sua grande maioria para reconhecimento de padrões e (iii) resolução de problemas de aprendizado de máquina.

Um dos mais conhecidos algoritmos imunológicos, o NSA foi proposto por Forrest et al. [76] e foi inspirado no processo de seleção negativa ocorrida no timo. O algoritmo foi planejado primeiramente para questões de segurança computacional (por exemplo, detecção de anomalias) usando o conceito de próprio e não-próprio e é dividido em duas fases principais: (i) fase de sensoriamento e (ii) fase de monitoramento.

Em [62], os autores aperfeiçoaram o algoritmo NSA apresentando correções para melhorar o desempenho do sistema de segurança proposto.

Outros trabalhos como [181], [50], [121], [49] [85], e [96], propuseram algumas arquiteturas que podiam ser aplicáveis a sistemas de segurança computacional. Seguindo os conceitos usados pelo NSA e a seleção clonal, juntamente com outras técnicas computacionais (por exemplo, agentes inteligentes, agentes móveis, lógica *fuzzy* e teoria dos jogos).

O algoritmo CLONALG foi proposto originalmente por Leandro de Castro e Von Zuben [57]. Existem duas versões apresentadas: (i) uma para resolver problemas de aprendizado de máquina e (ii) a outra para reconhecimento de padrões. Em [59] foi estendido o CLONALG para problemas de otimização por conter algumas características que o faziam apto para esta aplicação.

Para Forrest et al. [76] a idéia de um algoritmo representando a seleção clonal leva a uma analogia com algoritmos genéticos, porém, sem *crossover*. Fato contestado por Leandro de Castro e Von Zuben [57] no desenvolvimento do CLONALG.

De acordo com o princípio da seleção clonal, um algoritmo que a utilize precisa ter:

- **Geração de diversidade:** as células B produzem anticorpos combinando-os aleatoriamente a partir da biblioteca de genes;

- **Proliferação de anticorpos:** processo ocorrido com as células B após o reconhecimento do antígeno. Esse processo ocorre por clonagem;
- **Seleção de afinidade:** quanto maior a afinidade entre o anticorpo e o antígeno, maior será a produção de clones para o contra-ataque;
- **Maturação da afinidade:** processo conhecido como mutação com taxa inversamente proporcional à afinidade;
- **Memória:** após a proliferação das células B, algumas células alcançarão um limiar que as transformarão em células “permanentes” no organismo.

Segundo seus autores, o CLONALG cumpre com todos os requisitos. Porém, não foi proposto para problemas de detecção propriamente ditos, mas, alguns dos conceitos utilizados já tinham sido aplicados em trabalhos envolvendo segurança computacional [76]. O conceito de seleção clonal pode ser usado para complementar o algoritmo NSA numa ferramenta imunológica.

A rede imunológica de Jerne originou muitos estudos posteriores na aplicação sobre ambientes computacionais. Inicialmente, os modelos de algoritmos simulando as redes imunológicas utilizavam equações diferenciais. Existem duas frentes para os algoritmos baseados nesta teoria: (i) modelos contínuos e (ii) modelos discretos. Para os modelos contínuos, cita-se o modelo proposto por [60], no qual utilizou-se um conjunto de equações diferenciais para quantificar a dinâmica dos componentes da rede imunológica. Para os modelos discretos, a proposta com bastante aceitação é guiada pelo algoritmo aiNet.

Segundo os autores, o algoritmo é baseado em um grafo ponderado composto por um conjunto de nós chamados de conexões, com um peso associado [58]. Para representar os antígenos e anticorpos, são usados valores reais. Portanto, é usado o espaço de formas euclidiano. Dessa forma, as principais características do aiNET são:

- **Capacidade de busca local e de busca global:** a primeira é feita através da maturação dos clones, enquanto que a segunda ocorre com a inserção de indivíduos aleatórios;
- **Facilidade de encontrar boas soluções subótimas:** segundo Leandro de Castro e Von Zuben [59], o algoritmo mostrou ser mais eficiente que o algoritmo de seleção clonal;
- **Problemas com custo computacional:** é maior que o de outros algoritmos utilizando conceitos de seleção clonal;

Os algoritmos citados nesta seção e aqueles que seguem as mesmas características são considerados AIS de primeira geração [3]. Os algoritmos baseados na DT são mais recentes e por mudar totalmente a visão do próprio/não-próprio, são considerados algoritmos da segunda geração dos AIS.

Os trabalhos [3] e [2] podem ser considerados os pioneiros da segunda geração dos AIS. Neles são apresentadas as características da DT e a analogia com a segurança computacional, principalmente a detecção de intrusão. Do estudo da DC, originou-se o algoritmo DCA, um dos mais conhecidos desta geração.

3.8.4 Algoritmo DCA

Sua primeira versão foi apresentada no trabalho [91]. Posteriormente, foi apresentado o trabalho [90] com novas agregações e, por fim, a versão final foi apresentada na tese de doutorado de Greensmith [88]. Baseado em populações, através da qual são representadas abstrações das DCs, tem sido aplicado para detecção de anomalias conforme os trabalhos [93] e [6]. Cada célula é capaz de coletar antígenos e processá-los juntamente com os sinais de entrada correspondente. A DC é representada computacionalmente como uma classe, sendo que sua estrutura de dados pode ser visualizada na figura 3.6, onde são mostrados os quatro componentes principais da DC.

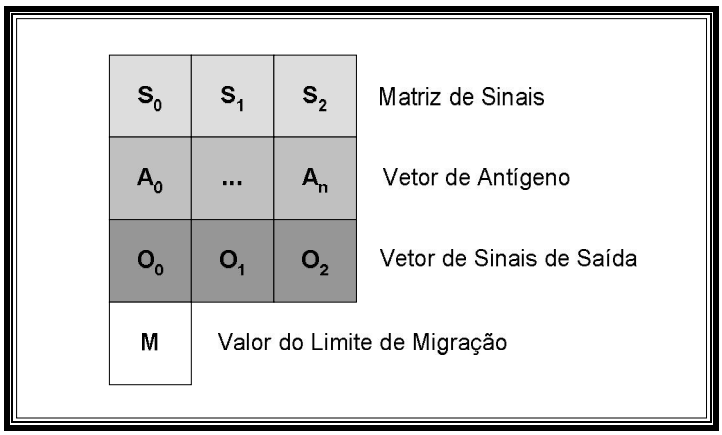


Figura 3.6 – Componentes da DC. Originalmente, três sinais são coletados para cada antígeno. Porém, é possível a existência de mais sinais. Três sinais de saída são apresentados e, dependendo do limite de migração M, a DC poderá passar de um estado para outro.

O algoritmo é dividido em duas partes: (i) uma para coleta de dados (análoga a pele) e (ii) outra para análise dos antígenos (representando o linfonodo[56]). Em "(i)" os dados são armazenados para serem apresentados para a população de DCs. Três estruturas de dados são responsáveis por esta parte: (a) vetor de antígenos principal, (b) matriz de sinais principal e (c) vetor de DCs (isto é, cada DC é um objeto). Em "(ii)", é feito o processamento de todos os sinais de entrada correspondentes aos antígenos e é realizada a avaliação sobre qual contexto foram coletados os antígenos.

Conforme apresentado na Seção 3.6.2, as DC podem assumir três estados: (a) imaturo, (b) semi-maturo e (c) maturo. Para simular esse processo é utilizada a equação 3.1. Três sinais representam a quantidade de sinais de entrada original. Porém, é possível o uso de mais sinais conforme [89]. Após a execução da equação 3.1, três sinais de saída terão sido gerados: (a) CSM (do inglês, *Costimulatory Molecules*), (b) semi-maturo e (c) maturo. Conforme visto na figura 3.6, existe um valor *M* que serve de limite para a mudança de estado da DC – CSM é análogo ao estado imaturo da célula – e, quando atingido, é avaliado o valor de "b" e "c"; aquele com maior valor definirá o novo estado da DC.

$$o_p(m) = \frac{\sum_i \sum_{j \neq 3} W_{ijp} * S_{ij}(m)}{\sum_i \sum_{j \neq 3} \left| W_{ijp} \right|} \forall p \,, \qquad (3.1)$$

onde:

- $\underline{W}$ representa o sinal da categoria i;
- $\underline{S}$ representa a matriz de sinais;
- $\underline{i}$ é a categoria do sinal de entrada;
  - (i_0 = PAMP, i_1 = DS, e i_2 = SS)
- $\underline{j}$ é o valor do sinal de saída;
  - (j_0 = CSM, j_1 = semi-maturo, j_2 = maturo)
- $\underline{o}$ representa o sinal de saída;

De acordo com Greensmith [88], existe uma tabela com valores pré-definidos que ajudam no processo de maturação das DCs. Os valores presentes na tabela 3.4 fazem

[56] Segundo o livro "Anatomia e Fisiologia Humana" de Clarice Ashworth Francone, os linfonodos ou gânglios linfáticos são pequenos órgãos perfurados por canais que existem em diversos pontos da rede linfática, uma rede de ductos que faz parte do sistema linfático. Eles atuam na defesa do organismo humano produzindo anticorpos.

referência àqueles utilizados no trabalho [210]. Dessa forma, cada valor do sinal de entrada é multiplicado pelo valor correspondente na tabela.

Tabela 3.4 – Valores pré-definidos para o cálculo do sinal de saída dos antígenos conforme experimentos de Williams et al. [210]. Na tabela, *j* representa PAMP, DS e SS, enquanto *p* representa CSM, maturo e semi-maturo respectivamente.

W_{ijp}	$j = 0$	$j = 1$	$j = 2$
$p = 1$	2	1	2
$p = 2$	0	0	3
$p = 3$	2	1	-3

O DCA possui três estágios principais (apresentados na figura 3.7): (i) inicialização, (ii) atualização e (iii) agregação. No primeiro, são dados valores aos vários parâmetros existentes (por exemplo, número de DCs, número de sinais de entrada, números de ciclos). No segundo estágio, o processo é contínuo e os valores para os antígenos e os sinais são atualizados toda vez que recebem novos dados (isto é, estímulo do ambiente).

Conforme citado anteriormente, quando o valor de saída da DC atinge um determinado limiar (isto é, através do acúmulo da variável CSM), a DC é removida da população, tendo o vetor de sinais e o valor do contexto registrado em um arquivo para o estágio de agregação. Este estágio não ocorre até que todos os dados sejam processados (isto é, no caso discreto) e sua principal função é calcular o valor de MCAV (do inglês, *Mature Context Antigen Value*), usado para calcular o grau da anomalia de certo antígeno, representado pela equação 3.2.

$$MCAV_x = \frac{Z_x}{Y_x}, \tag{3.2}$$

onde:

- MCAV representa o coeficiente para o antígeno *x*;
- Z_x é o número de apresentações sobre o contexto maturo para o antígeno do tipo *x*;
- Y_x é o número total de antígenos apresentados para o antígeno do tipo *x*;

A faixa de ação para esta equação está entre 0 e 1 e, quanto mais perto de 1, mais anômalo é o antígeno.

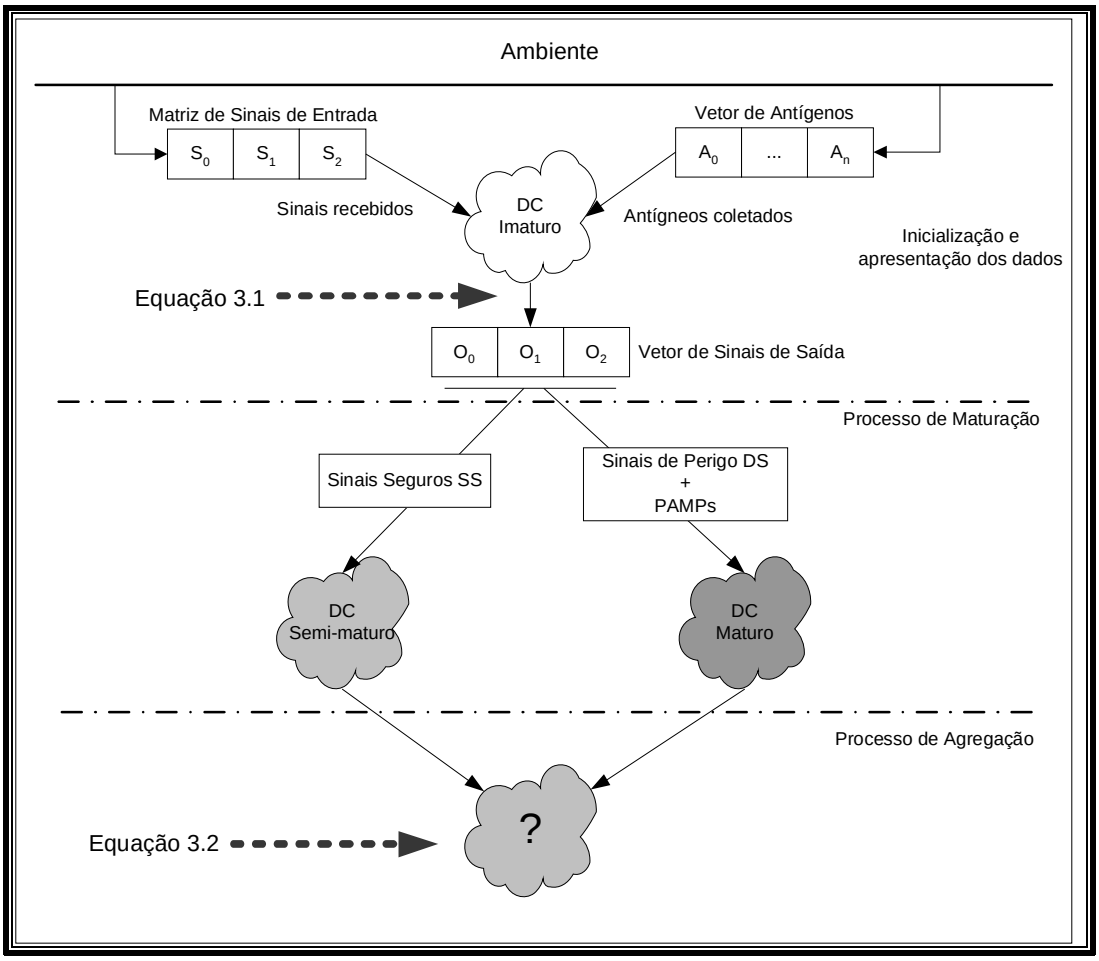


Figura 3.7 – As fases do DCA. No processo de maturação a DC muda seu estado após a análise dos sinais de saída, quando o sinal seguro (SS) é maior que a soma dos sinais de perigo (DS) e PAMP, a DC muda para o estado semi-maturo, caso contrário, muda para o estado maturo. O DCA só é executado na estação. Adaptação própria.

No DCA, existe o ciclo da célula (isto é, um processo discreto que ocorre através de uma taxa pré-definida). Para este trabalho, um ciclo de célula é realizado por segundo. Durante um ciclo, os sinais e os antígenos correspondentes àquele segundo são consumidos pelas DCs. Esse processo é contínuo, até que seja atingido um critério de parada (caso seja para análise de uma base de dados previamente coletados e trabalhados posteriormente). No caso de execução em tempo real, esse processo pode ser contínuo. Porém, para execução do estágio de agregação é preciso definir algum ponto de parada (por exemplo, por minuto) para que possa ser avaliado o grau de anomalia pelo qual são apresentados os antígenos e a capacidade de detecção das DCs [88].

O DCA apresenta um número de parâmetros iniciais que pode ser considerado alto. São eles:

- I = número de sinais de entrada por categoria;
- J = número de categorias do sinal de entrada;
- K = número de antígenos no vetor de antígenos;
- L = número de ciclos da DC;
- M = número de DCs na população;
- N = tamanho do vetor de antígenos da DC;
- P = número de sinais de saída por DC;
- Q = número de antígenos apresentados pela DC por ciclo;
- R = número máximo de antígenos coletados pela DC para um ciclo (o receptor de antígenos da DC);
- T_{max} = tamanho do vetor de antígeno principal;

A figura 3.8 apresenta um pseudocódigo do DCA, onde é possível ver a fase de inicialização, o ciclo básico, a fase de atualização e o ciclo da DC. É possível ainda verificar que o ciclo da célula mantém toda a estrutura de dados da DC. A cada iteração, os sinais são atualizados da matriz S principal para a matriz interna da DC, assim como os antígenos do vetor A principal (isto é, cópia direta dos dados). O valor $\bar{o}_p(m)$ é obtido após o acúmulo do sinal de saída o_p quando é atingido o número total de ciclos de célula. Caso o sinal de saída $o_0(m)$ seja maior que T_{max}, a DC será retirada da população e os antígenos serão impressos em um arquivo juntamente com o contexto atual da DC. Logo em seguida, são zerados todos os dados da DC, que voltará para a população. No final, após processadas todas as DCs e todos os dados de entrada, será calculado o grau de anomalia (MCAV) dos antígenos gravados no arquivo externo como uma forma de avaliar o nível de detecção das DCs.

```
Parâmetros de Inicialização
I,J,K,L,M,N,O,P,Q
Enquanto(l<L) faça          //número de ciclos
      atualiza A e S        //vetor de antígenos e matriz de sinais
      para m = 0 até M      //número de DC na população
         para 0 to Q
               DC_m apresenta Q antígenos de A
         para todo i = 0 até I e todo j = 0 até J
               s_ij^DC = S_ij
         para as N células dendríticas faça
                     DC_m processa a_nm^DC
         para p até P
               Calcule o_p
               o_p(m) = o_p(m) + o_p      //acúmulo de sinais
         Se o_0(m) > t_m                  //limiar de migração
         DC_m é removida da população
         DC_m migra, os antígenos e o contexto são impressos no arquivo
         DC_m zera o vetor de antígeno e todos os sinais
l++
analisa antígenos e calcula o valor de MCAV   //inicio da fase de agregação
```

Figura 3.8 – Pseudocódigo do DCA. Apresenta a inicialização, ciclo básico, atualização, ciclo da DC e a agregação final. Adaptado de [88].

O DCA tem servido de inspiração para muitos trabalhos, inclusive este. Por isso, ao apresentar o modelo desenvolvido para este trabalho no Capítulo 5, são apresentadas maiores informações sobre os detalhes modificados no algoritmo para este trabalho.

3.9 Considerações Finais Sobre o HIS e AIS

O HIS serviu e continua servindo de fonte de inspiração para muitos trabalhos que procuram sistemas avançados na natureza (inclui-se aqui a natureza humana). Sua analogia favorável a projetos em que a segurança é ponto fundamental fazem dos AIS modelos promissores para aplicação em segurança computacional [55].

Os algoritmos iniciais, considerada a primeira geração dos AIS, apresentaram alguns problemas, entre eles se destaca o problema da escalabilidade (isto é, quando um modelo é aplicado sobre um determinado ambiente com tamanhos cada vez maiores, acaba por ter seu desempenho abalado).

Desta forma, os algoritmos usando a abordagem dos sinais da DT se transformaram em algoritmos pertencentes à segunda geração dos AIS, por apresentar, em resultados preliminares, valores considerados bastante importantes na solução do problema da escalabilidade [88].

É importante salientar que nem todos os mecanismos do HIS são conhecidos, ou mesmo aqueles conhecidos não são totalmente esclarecidos. Por isso, existe a possibilidade de que novas abordagens possam surgir [88].

Para a abstração dos conceitos da DT, usando as DCs, assim como as células T, são usados agentes. Como exemplo, o DCA é embutido em um agente que fica na estação fazendo o processamento dos sinais à procura de eventos anormais. Portanto, a apresentação dos conceitos envolvidos na utilização de agentes é apresentada no próximo capítulo. Nele, também são apresentadas definições sobre o uso da cognição nos sistemas.

Capítulo 4

Sistemas Multiagentes

Neste capítulo são abordados os aspectos dos agentes inteligentes e dos MAS segundo a visão da AI. Na última seção são apresentados alguns trabalhos encontrados na literatura que abordam a DT e os MAS como forma de mostrar os trabalhos correlacionados com o modelo a ser apresentado no Capítulo 5.

4.1 Taxonomia dos Agentes

A idéia de agentes não é usada unicamente para a área da computação e associados. Segundo Fernandes [63], o uso do termo agente também pode ser encontrado em outras áreas, entre elas: (i) economia, (ii) sociologia e (iii) comportamento animal.

Porém, para a computação, a definição mais comum para um agente diz que o mesmo pode ser considerado uma entidade autônoma tal como um programa de software ou um robô [187]. Para Russel e Norvig [172] um agente é aquele que percebe o seu ambiente por meio de sensores, e age sobre ele através dos atuadores. Computacionalmente, um sensor pode ser qualquer dispositivo que monitora um ambiente (por exemplo, câmeras filmadoras, sensores de movimento, de temperatura). Por outro lado, os atuadores são dispositivos que executam alguma tarefa em resposta aos comandos a eles atribuídos (por exemplo, motores, braços mecânicos, rodas). Numa terceira definição, Maes [134] cita o agente como um sistema que procura atender a um conjunto de objetivos inseridos em um ambiente complexo e dinâmico.

Na biblioteca livre Wikipédia [208], existe uma definição dita “informal” para agente: “*... alguém ou alguma coisa que atua como um procurador com propósito específico de realizar ações que podem ser entendidas como benéficas dentro do contexto onde ele atua*”.

De acordo com Costa [43], que apresenta várias definições encontradas na literatura, existe um consenso entre as definições: na sua grande maioria, é possível verificar que os conceitos de autonomia, capacidade de responder a determinadas situações, facilidades para comunicação e capacidade de aprender como alcançar seus objetivos estão presentes.

Segundo Jennings e Wooldridge [115] e Costa [43], as características esperadas de um agente são:

- **Mobilidade:** capacidade de mover-se para outro ambiente;
- **Autonomia:** um agente pode operar sem a interferência humana;
- **Habilidade social:** o agente precisa ser capaz de interagir com outros agentes ou até mesmo com o ser humano;
- **Reatividade:** capacidade para perceber o ambiente no qual está inserido e reagir conforme os estímulos recebidos;
- **Pró-atividade:** habilidade em tomar iniciativa própria para executar uma ação em prol de um objetivo;
- **Continuidade temporal:** é preciso que o agente funcione continuadamente;
- **Cooperação:** capacidade que os agentes têm de trabalharem em conjunto de forma a concluírem tarefas de interesse comum;
- **Comunicabilidade:** diretamente relacionada à habilidade social.

Para Costa [43] e Nwana [154] um agente não precisa ter todas essas características, porém, sua capacidade está intrinsecamente associada à presença delas. Mas, é preciso lembrar que o uso de tais características permite a organização dos agentes em tipologias. Na verdade, classificar os agentes não é uma tarefa fácil, talvez esteja no mesmo nível de dificuldade encontrada para a definição dos mesmos. Nwana [154] propôs uma tipologia baseada em diferentes dimensões de classificação onde os agentes podem ser classificados de acordo com (i) mobilidade, (ii) racionalidade, (iii) presença de três atributos primários (isto é, autonomia, cooperação e aprendizado), (iv) objetivo do agente e (v) presença de características híbridas (isto é, característica diferentes combinadas em um mesmo agente).

Dessa forma, o autor identificou sete tipos de agentes através da interação entre as características:

- **Agentes colaboradores:** para esses agentes as características de cooperação e autonomia são as mais importantes. O conceito de um sistema com vários agentes torna-se evidente nesse caso, assim como a necessidade de um padrão de comunicação entre os agentes para que possa haver cooperação;
- **Agentes de interface:** agente diretamente relacionado ao usuário na solução de seus problemas em um mesmo ambiente [134]. A ênfase nesse caso está para a capacidade de aprendizado e autonomia. Costa [43] aponta na direção de que a colaboração nesse caso não precisa de uma linguagem explícita para comunicação, uma vez que a mesma é feita diretamente com o usuário e não com outros agentes;

- **Agentes móveis:** para esses agentes a possibilidade de mover-se para outros ambientes para realização de tarefas trouxe vários benefícios. Entre eles alguns se destacam como a redução dos custos de comunicação, a possibilidade de executar suas tarefas de forma assíncrona, independência da utilização de recursos de um mesmo local. A idéia de mobilidade de agentes chamou a atenção da computação distribuída, sendo alvo de estudos;
- **Agentes de informação:** através da digitalização das informações, antes em mídias impressas, gerou a necessidade de ferramentas que organizasse o acesso as informações e forma como apresentá-las ao usuário. Porém, não se pode confundir com simples programas de busca através da combinação de palavras-chave. Estes agentes devem ser capazes de encontrar informações relevantes através do reconhecimento de padrões;
- **Agentes reativos:** segundo pode ser encontrado em [187], os agentes reativos são baseados em modelos de organização biológica ou etológica (por exemplo, formigas, cupins, abelhas). Esta é uma definição mais prática na representação da simplicidade dos agentes dessa categoria. Mesmo sendo simples e usando de comunicação básica, a interação com outros agentes da mesma categoria faz emergir comportamentos complexos. O modelo de funcionamento é baseado no modelo ação/reação ou estímulo/resposta. Para os agentes reativos não há representação simbólica do conhecimento. Pode-se dizer que o conhecimento dos agentes é implícito;
- **Agentes híbridos:** comportam o uso de características diferentes num mesmo agente com o intuito de atuar sobre ambientes heterogêneos com maior eficácia e eficiência;
- **Agentes inteligentes:** também conhecidos como agentes cognitivos, são considerados o alvo dos pesquisadores quando o assunto é agente. Por teoria, um agente inteligente é aquele que possui todos os atributos possíveis da autonomia, aprendizagem e cooperação.

4.2 Aspectos Cognitivos dos Agentes

Algumas vezes o termo “agente” é confundido com “agente inteligente”, fato errôneo, visto que um agente não necessariamente terá capacidades que o incluirão no rol dos

"inteligentes". Para analisar esse fato é preciso levar em consideração qual o verdadeiro significado da palavra inteligência para os agentes.

Primeiramente uma definição do termo inteligência apresentado por Neisser [146]: "*Os indivíduos diferem na habilidade de entender idéias complexas, de se adaptar efetivamente ao ambiente, para aprender com a experiência, a participar em diversas formas de raciocínio, de superar obstáculos através do pensamento. Embora tais diferenças individuais possam ser substanciais, nunca são completamente consistentes: o desempenho intelectual de uma dada pessoa vai variar em ocasiões distintas, em domínios diferentes, assim como julgados por critérios diferentes. Os conceitos de 'inteligência' são tentativas de esclarecer e organizar este conjunto complexo de fenômenos.*"

A palavra cognição é freqüentemente usada nas muitas definições de agentes inteligentes. Segundo os estudiosos, o termo tem sua origem nos escritos de Platão e Aristóteles. Derivada do latim através da palavra "*cognitione*" – que significa a aquisição de um conhecimento através da percepção – a cognição pode ser definida como ato ou processo de conhecer, onde se incluem: (a) atenção, (b) percepção, (c) memória, (d) raciocínio, (e) juízo, (f) imaginação, (g) o pensamento e (h) o discurso [209].

Pelo aspecto computacional, a cognição passou a fazer parte dos estudos a partir da década de 1950 com a psicologia cognitiva formando um paralelo entre as funções do cérebro humano e os conceitos presentes nos computadores, dentre os quais: (i) codificação, (ii) armazenamento, (iii) reparação e (iv) memorização da informação. Na mesma década, surgia a AI (através do Encontro de *Darmouth*[57]) como forma de elucidação e explicação dos conceitos de representação, organização e processamento de conhecimentos conceituais [10]. Do estudo da cognição humana surgiu o termo ciências cognitivas, que passou a ter grande valor na definição do comportamento humano e sua representação computacional.

Alan Turing [195] foi um dos primeiros a propor um modelo matemático de computador, assim como questionar se uma máquina poderia ser chamada inteligente. Observando o questionamento proposto por Turing, é possível também questionar o sentido apresentado à palavra inteligente quando associada ao agente. É certo que o uso da palavra inteligente não necessariamente é aplicado em sua completude, mas, a presença de alguns atributos já o distingue de qualquer outra entidade computacional.

[57] Encontro entre vários pesquisadores de renome na época para estudar o termo inteligência artificial. Entre eles o pesquisador Marvin Minsky, o primeiro a usar o termo. Mais detalhes em: http://www.pgie.ufrgs.br/alunos_espie/espie/silviab/public_html/espieufrgs/espie00003/inteligenciaartificial.htm

É fato também que os agentes inteligentes ainda não podem representar simbolicamente todos os conceitos de cognição da inteligência humana [114], [213]. Por isso, as pesquisas com essas entidades estão em evidência na computação. Este trabalho é mais um em meio a tantos outros, que procura usá-lo como ferramenta computacional bioinspirada para solução de um problema complexo chamado detecção de intrusão em redes.

4.3 Sistemas Multiagentes

Desde os primórdios da AI, grande parte dos estudos era voltada para o desenvolvimento de teorias, técnicas e sistemas para o estudo e compreensão das propriedades do comportamento e raciocínio de uma única entidade cognitiva [187]. Porém, com a crescente popularidade das redes, nas décadas de 1980 e 1990, novos paradigmas da computação emergiram. A computação distribuída e paralela despertou interesse da comunidade científica e um novo conceito de inteligência foi proposto: a Inteligência Artificial Distribuída (IAD). Na visão de [157], a IAD é um campo da AI que se preocupa com coordenação e distribuição do conhecimento e ações em ambientes multiagentes. Sichman et al. [178] aponta a IAD como um modelo de inteligência baseado no comportamento social através da interação entre agentes. Portanto, para Alvares e Sichman [7] a metáfora utilizada pela AI clássica é de origem psicológica, enquanto que à IAD é de natureza sociológica e etológica.

Atualmente, segundo Costa [43], a IAD é subdividida em duas grandes áreas que se complementam, a saber: (i) sistemas multiagentes (MAS) e (ii) resolução distribuída de problemas (RDP). Para o autor, tanto os MASs quanto os RDPs trabalham com a presença de diversos agentes compartilhando um mesmo ambiente. Porém, a principal diferença está na forma como acontece a interação entre os agentes. Enquanto os RDPs focam na cooperação entre os agentes como forma de resolver problemas, os MASs usam a coordenação como fator chave para identificar e atuar de forma coletiva na resolução de um problema. Ainda segundo o autor, inspirado na afirmação encontrada em [66], os sistemas RDPs são sistemas que procuram o bem comum entre os agentes, levando em conta fatores como confiança e sinceridade, enquanto que por outro lado, os MASs utilizam-se dos conceitos da teoria dos jogos[58] para demonstrar a racionalidade individual como uma forma de alcançar o máximo na

[58] A teoria dos jogos tornou-se um ramo proeminente da matemática nos anos 30 do século XX, especialmente depois da publicação em 1944 de "*The Theory of Games and Economic Behavior de John von Neumanne Oskar Morgenstern*". Ver mais no livro "Teoria dos Jogos" de Ronaldo Fiani.

relação custo/benefício. Mas, segundo os autores, os dois conceitos podem conviver harmoniosamente, complementando um ao outro.

Para Sycara [187], a capacidade de um agente inteligente é limitada pelo seu conhecimento, seus recursos computacionais, e sua perspectiva. Portanto um único agente seria incapaz de solucionar problemas complexos e distribuídos, pois teria informações incompletas ou até mesmo capacidade limitada (isto é, visão limitada do problema como um todo).

Flores-Mendez [74] apresenta uma visão baseada na definição de Durfee et al. [65] para MAS na qual afirma ser uma rede flexível e de entidades solucionadoras de problemas que trabalham juntas para encontrar respostas para problemas que estão além da capacidade individual ou do conhecimento de cada entidade. Ele apresenta ainda algumas características presentes nos MASs recentes onde: (i) cada agente tem capacidades incompletas para resolver um problema, (ii) não existe um sistema global de controle, (iii) os dados são descentralizados e (iv) a computação é executada de forma assíncrona.

Nos tempos de Internet, e os sistemas abertos – aquele onde a estrutura do sistema, por ela mesma, é capaz de mudar dinamicamente – os MASs tem sido importante peça para a resolução de problemas desses ambientes [115].

Alguns autores cunham o termo “sociedade de agentes” para os MASs [72] e [178]. Sobre este ângulo, o uso de agentes autônomos, com características e funções diferentes faz emergir a idéia existente por traz de uma sociedade humana onde cada ser humano é visto como um agente, e a interação entre eles são chamados comportamentos sociais.

Quanto à classificação dos MASs, existe um consenso para a existência de dois tipos principais: (i) MAS reativo e (ii) MAS cognitivo. Dois modelos que serão abordados nas subseções seguintes.

4.3.1 Sistemas Multiagentes Reativos

Oriundo da idéia de agentes reativos (*c.f.* subseção 4.1), e também da teoria apresentada por [31] para agentes aplicados à robótica, os MASs reativos são vistos como sistemas onde o agente é uma entidade bastante simples, sem nenhuma inteligência. Porém, ao associar várias dessas entidades, surge um comportamento social complexo e inteligente que possibilita a solução de problemas complexos. A figura 4.1 ilustra a simplicidade do agente reativo, geralmente, é o modelo escolhido para atender o RMAS.

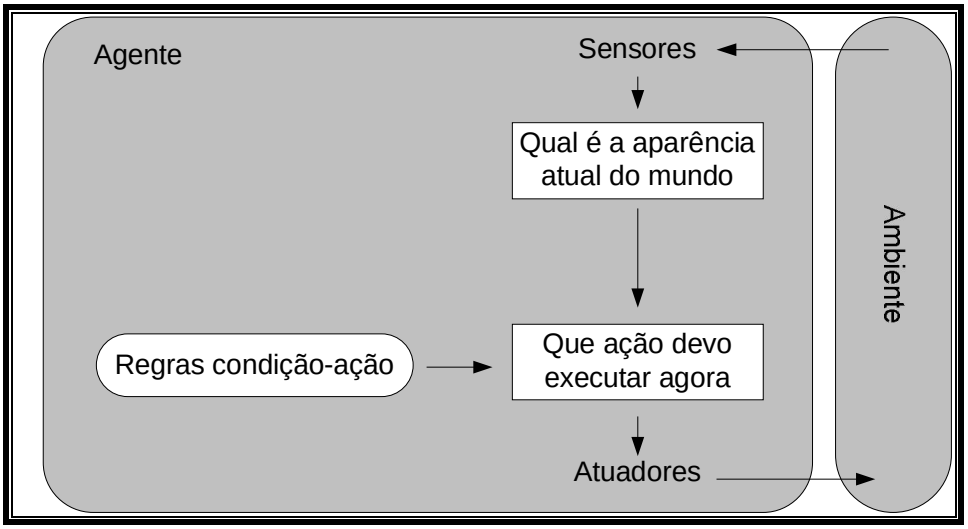


Figura 4.1 – Diagrama mostrando a arquitetura de um agente reativo simples [172].

Como exemplo, Franklin e Graesser [80] apontam para uma colônia de formigas: "*cada formiga isoladamente é uma entidade bem simples; não se atribui inteligência a uma formiga. Entretanto, o trabalho realizado por uma colônia de formigas é bem complexo: procura de alimento, transporte do alimento até o formigueiro, cuidado com os ovos e larvas, defesa da colônia, etc. E elas têm tido êxito, adaptando-se às mudanças de condições, como os períodos glaciais, e sobrevivendo há muitos milhões de anos*".

Os autores destacam as principais características dos agentes e dos MASs reativos:

- **Inexistência da representação explícita do conhecimento:** o comportamento do agente se manifesta de acordo com seu comportamento, ou seja, está implícito;
- **Inexistência da representação do ambiente:** comportamento baseado no que acontece a cada instante no ambiente sem que haja uma representação explícita do mesmo;
- **Falta de memória das ações:** não há registro dos acontecimentos vividos pelo agente;
- **Organização etológica:** forma de organização dos agentes é similar aquela observada nos animais;
- **Grande quantidade de membros:** geralmente existe grande quantidade de agentes nesses sistemas;

Resumidamente pode-se dizer que os RMAS possuem um grande número de entidades idênticas com objetivos individuais sem consciência do problema geral, mas que, através do trabalho cooperativo, reagindo conforme percebem o ambiente através de estímulo/resposta,

fazem emergir a solução através das interações entre eles sem que haja memória dos fatos vivenciados e conseqüentemente sem planejamento do futuro. Por serem agentes simples e individuais, a comunicação entre eles é limitada, sendo uma questão bastante diferenciada dos modelos cognitivos que usufruem da comunicação com outros agentes como aliada a suas atividades.

Alguns exemplos de modelos podem ser encontrados na literatura representando RMAS:

- **MANTA (*Modelling ANTnest Activity*) [69]:** simula o comportamento de uma colônia de formigas;
- **PACO (*PAtterns COrdination*) [60]:** idéia da co-evolução de um conjunto finito de agentes onde cada um representa uma solução parcial interagindo entre IS e com o ambiente. Neste caso a solução do problema é atingida pela posição do conjunto de agentes. Muito utilizado em representações espaciais como àquelas utilizadas na robótica.

4.3.2 Sistemas Multiagentes Cognitivos

A definição mais comum e encontrada em vários trabalhos é aquela na qual coloca os MASs cognitivos como modelos baseados em modelos organizacionais humanos, como grupos, hierarquias e mercados [70]. Segundo os autores, existem seis características principais para os agentes cognitivos:

- **Reconhecimento:** mantêm uma representação explícita de seu ambiente e dos outros agentes da sociedade;
- **História:** possui memória das histórias vivenciadas no passado;
- **Interação:** comunicação direta através de mensagens entre os agentes;
- **Autonomia:** possui mecanismo de controle deliberativo;
- **Organização:** organizam-se como os seres humanos, ou seja, em modelos sociológicos;
- **Quantidade:** poucos agentes compõem os CMAS;

A figura 4.2 apresenta uma arquitetura para o agente cognitivo. Resumidamente, os agentes cognitivos são entidades que possuem capacidade de raciocinar sobre as ações tomadas no passado e planejar as ações futuras. Eles têm representação explícita do ambiente,

assim como dos outros agentes que coexistem no ambiente e podem interagir com eles usando linguagens e protocolos de comunicação considerados complexos [7].

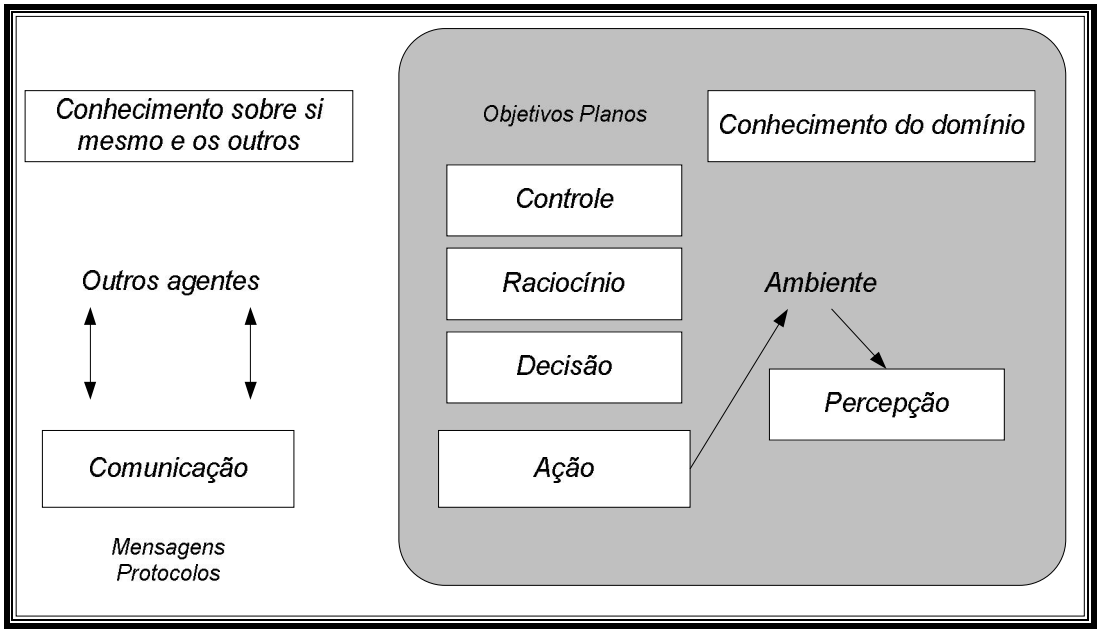


Figura 4.2 – Arquitetura do agente cognitivo mostrando que o agente precisa ter conhecimento do domínio, de si mesmo e dos outros agentes. [7].

Uma característica marcante para os agentes cognitivos é a alta complexidade para representá-los computacionalmente [70].

De acordo com Bussman e Demazeau [34] podem existir várias classes de agentes cognitivos, depende apenas da complexidade interna. O autor apresenta algumas classes:

- **Agentes organizados:** obedecem a leis e regras sociais mantendo perspectivas múltiplas acerca de um problema;
- **Agentes negociantes:** resolvem conflitos através da negociação;
- **Agentes intencionais:** possuem noções internamente como intenções, engajamentos e planos parciais;
- **Agentes cooperativos:** possuem esquema de alocação de tarefas e mantêm representações mútuas um dos outros;
- **Módulos comunicantes:** comunicam-se através de protocolos;
- **Atores, processos:** representam as primitivas de comunicação.

Costa [44] apresenta os quatros tipos de arquitetura de CMAS mais utilizados:

- ***BlackBoard System:*** o termo *blackboard* refere-se a uma memória compartilhada, na qual está representado o ambiente. Pode-se dizer que é um sistema onde os agentes não conhecem os outros, pois quando não tem capacidade para executar

uma tarefa, o agente coloca a tarefa na *blackboard* para que outro agente com habilidade para resolver o problema possa assim o fazer e colocar o resultado novamente na *blackboard*. A desvantagem está na alta taxa de troca de mensagens realizada pelo suporte de comunicação;

- **MAS federados:** nessa arquitetura existe a figura do agente facilitador, aquele que gerencia as tarefas dos outros agentes considerados mais simples. Abordagem perigosa por ter centralizadas algumas tarefas importantes sobre alguns agentes. Caso haja problemas com o agente facilitador, o MAS pode ter problemas por completo. Porém, é uma abordagem que diminui o problema de alta troca de mensagens entre os agentes;
- **MAS democráticos:** nesta abordagem não existem níveis diferentes para os agentes, todos estão no mesmo nível hierárquico. Há controle da comunicação através de uma linguagem conhecida como "linguagem de cooperação". Os agentes nessa arquitetura realizam ações coletivas assincronamente, inclusive a comunicação. Como vantagem, cita-se a modularidade e a flexibilidade. Porém, tem a desvantagem da necessidade do agente conhecer a identidade e parte das habilidades dos demais agentes da comunidade;
- **MAS abertos:** sistema que pode sofrer mudanças na quantidade de agentes dinamicamente. É considerado um sistema adaptável, pois, os agentes podem adaptar-se dependendo dos serviços disponíveis na comunidade. Por essa característica, é chamado de sistema robusto, porém a excessiva troca de mensagens e um complexo protocolo de comunicação trazem dificuldades para o sistema.

4.4 Interações entre Agentes

Pelas definições apresentadas nas seções anteriores, é possível perceber que em um sistema onde os agentes consigam resolver problemas de forma colaborativa é necessário um bom modelo de comunicação. Para os modelos CMAS que utilizam fortemente da interação entre os agentes, uma boa linguagem para a comunicação pode ser visto como fator de elevada importância. Três elementos-chaves são apresentados por [44], [187], [73] para que haja boa interação entre os agentes: (i) um protocolo e uma linguagem de comunicação comum entre

os agentes, (ii) um padrão comum no formato para o conteúdo da comunicação e (iii) uma ontologia compartilhada.

Segundo Flores-Mendez [74] existem duas abordagens para linguagens de comunicação entre agentes: (a) de natureza processual e (b) declarativa. Na primeira, a comunicação é baseada em conteúdo executável através, por exemplo, do uso de linguagens de programação JAVA ou TCL. Porém, há uma série de limitações nesta abordagem (por exemplo, dificuldade de controlar e coordenar conteúdo executável). Na segunda, a comunicação é baseada em afirmações declarativas (por exemplo, definições e suposições). Esta abordagem tem sido a preferida entre os pesquisadores e desenvolvedores.

Uma das mais conhecidas e usadas linguagens declarativas é conhecida como linguagem de manipulação e troca de conhecimento KQML (do inglês, *Knowledge Query and Manipulation Language*) [73]. É oriunda do projeto DARPA *Knowledge Sharing Effort*. Sua característica mais importante é o compartilhamento de conhecimento e informações em tempo de execução [73]. Outra característica marcante é o uso de uma sintaxe bem definida capaz de ser legível para seres humanos no formato BNF (do inglês, *Bakus-Naur Form*) que possui poucas regras e símbolos.

4.5 Arquiteturas MAS

A padronização de uma arquitetura genérica para MASs ainda é uma tarefa difícil segundo Flores-Mendez [74]. Os fatos contrários são representados pela dificuldade encontrada na programação distribuída e na complexidade em dar suporte ao processo de colaboração entre agentes. O autor continua sua teoria apontando que a adoção de tecnologias MAS só terá êxito a partir do momento em que houver formalização e padronização das arquiteturas, mecanismos e protocolos suportando a interação distribuída dos agentes.

Porém, mesmo com tantas dificuldades encontradas, existe um esforço multilateral para padronização da tecnologia MAS. Alguns grupos (por exemplo, formados por indústrias juntamente com entidades acadêmicas) criaram entidades para o desenvolvimento de modelos de padronização de arquitetura. Entre eles se destacam:

- **Modelo OMG (*Object Manager Group*) [158]:** caracterização dos agentes por suas capacidades (por exemplo, inferência e planejamento), tipo de interação e mobilidade. Estabelece o conceito de agências, que apóiam a execução simultânea do agente, a segurança e a mobilidade dos agentes;

- **Modelo FIPA (*Foundation for Intelligent Physical Agents*) [19]:** modelo com substancial importância sendo concebido para servir como orientação no desenvolvimento de MASs. É baseada na teoria de que é preciso um framework mínimo para o gerenciamento de agentes em um ambiente aberto e possui várias especificações como guia destacando entre elas duas com maior importância: (i) gerenciamento de agentes e (ii) linguagem de comunicação entre agentes. Implementa o conceito de plataforma de agentes, uma espécie de infra-estrutura para a implantação e a interação dos agentes. Esse modelo é usado neste trabalho através do framework JADE, que é apresentado posteriormente;
- **Modelo KAoS (*Knowledge-able Agent-oriented System*) [29]:** é conhecido como um modelo distribuído para agentes de *software*. A principal característica é a idéia de um agente genérico simples pelos quais outros agentes são desenvolvidos;
- **Modelo GM (*General Magic*)[59]:** proposto para executar sobre o ambiente do comércio eletrônico através do conceito de agentes móveis. É um projeto de cunho comercial e propõe a concepção de agentes fornecedores e agentes consumidores. A idéia é que existe uma rede que interconecta lugares com potencial para receber os agentes;

4.6 Ferramentas MAS

Existem algumas ferramentas para desenvolvimento de agentes e também de MASs. Entre elas: (i) JADE, (ii) JASON, (iii) Jack, (iv) Cougaar e (v) Aglets. Neste trabalho foi utilizada a primeira ferramenta, ou seja, JADE. Por isso, serão abordadas algumas características da mesma nesta seção.

A ferramenta JADE [25] (do inglês, *Java Agent Development Framework*) é totalmente desenvolvida usando a linguagem JAVA e tendo como base o padrão FIPA descrito na seção anterior. Portanto, os agentes que são desenvolvidos com JADE podem interagir com agentes desenvolvidos em outra linguagem, desde que sigam as especificações FIPA. Por serem desenvolvidos numa linguagem que tem por principio a portabilidade, os agentes podem trabalhar em computadores com diferentes sistemas operacionais. Entretanto, para que isso ocorra é preciso executar a JDK (do inglês, *Java Development Kit*) e o JRE (do

[59] Um ambiente para construção de sociedade de agentes. O nome apresentado é TELESCRIPT juntamente com o paradigma da programação orientada a agentes (AOP). Mais em: http://www.pucrs.campus2.br/~jiani/trabalhos/linguagens.htm#tele

inglês, *Java Runtime Environment*). Dessa forma, em cada estação é executada uma JVM (do inglês, *Java Virtual Machine*), onde cada JVM é um container de agentes que fornece um ambiente completo em tempo de execução para a execução do agente assim como de vários agentes ao mesmo tempo.

Segundo Bellifemine et al. [19], o modelo de comunicação FIPA foi completamente desenvolvido e seus componentes foram totalmente integrados, conforme pode ser visto na figura 4.3: protocolos de interação, envelope, ACL, linguagens de conteúdo, esquemas de codificação, ontologias e protocolos de transporte. O mecanismo de transporte é bastante adaptável a cada situação podendo ser escolhido entre vários tipos de modelos de protocolos. O serviço de páginas brancas e amarelas são serviços para organização dos agentes lembrando a idéia de quadro negro (blackboard) [19].

A plataforma JADE é o conjunto de todos os containeres (isto é, cada instância do ambiente JADE é chamada de container) e fornece abstrações das camadas inferiores para os agentes e desenvolvedores conforme é ilustrado na figura 4.4.

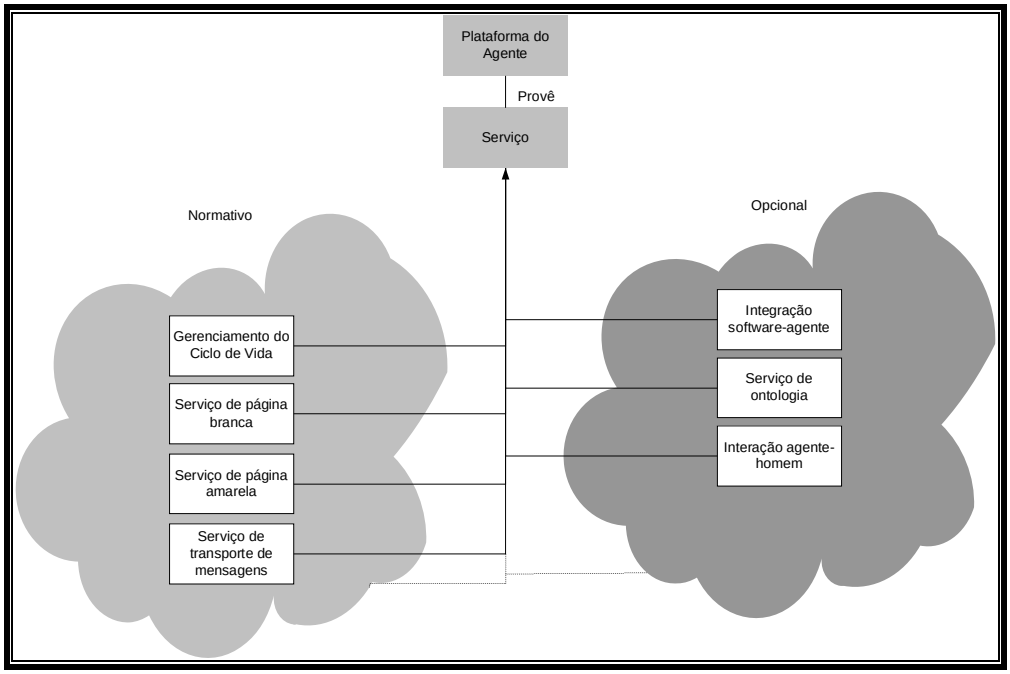


Figura 4.3 – No padrão FIPA os serviços são oferecidos dentro da plataforma do agente.

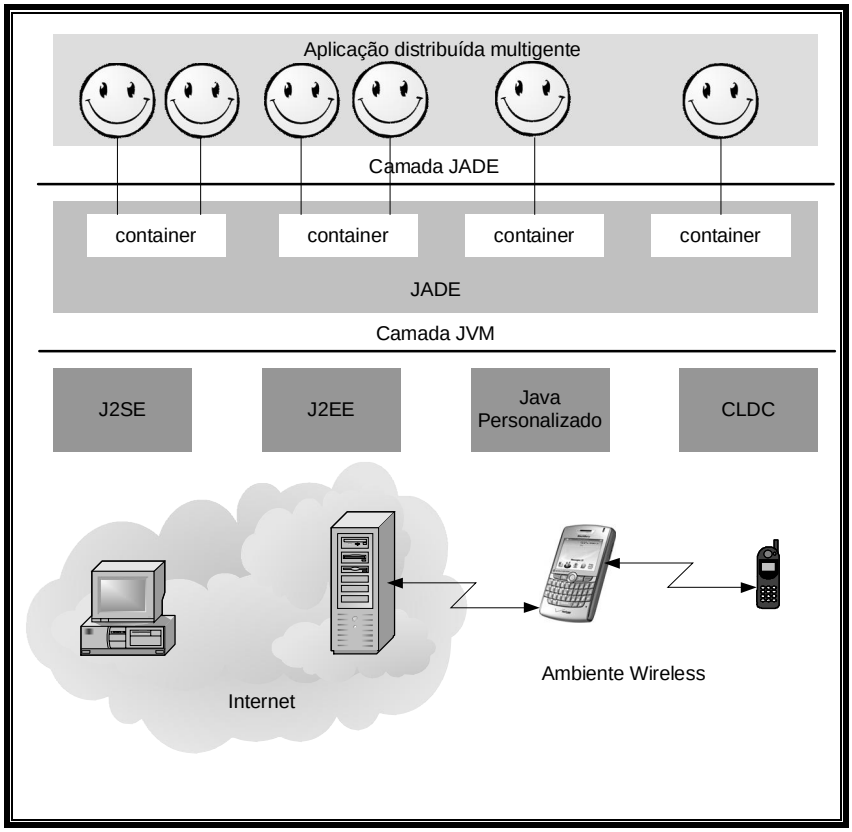


Figura 4.4 – Arquitetura de funcionamento do arcabouço JADE. Destaque para o ambiente wireless [19].

Nas últimas versões (por exemplo, 3.6 e 3.7) é possível usar o arcabouço para desenvolvimento de agentes em dispositivos móveis o que ajuda a aumentar a faixa de tecnologias aplicáveis. Conforme pôde ser visualizado na figura 4.7 abaixo, servidores usando J2EE[60] (do inglês, *Java 2 Platform Enterprise Edition*), estações através do J2SE (do inglês, *Java 2 Standard Edition*), dispositivos móveis com J2ME (do inglês, *Java 2 Micro Edition*) e cartões com a VM (do inglês, *Virtual Machine*) Java.

[60] Segundo a Sun Systems, desenvolvedora da linguagem JAVA, atualmente foi retirado o 2 da sigla, ficando apenas JEE, JSE e JME.

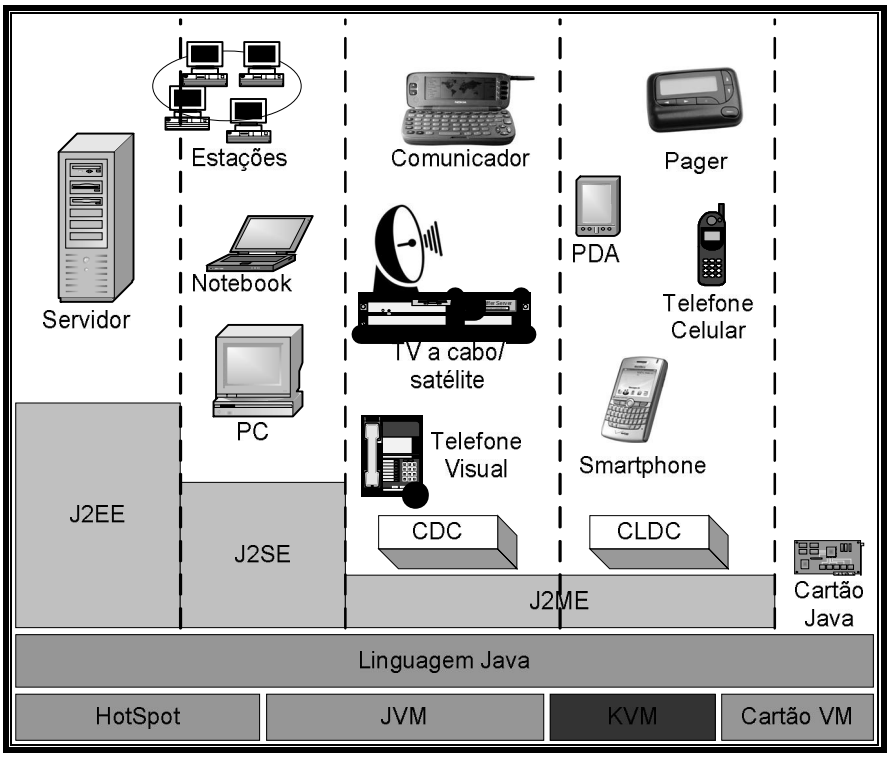


Figura 4.5 – Demonstração das várias tecnologias onde pode ser aplicado o arcabouço JADE. Destaque para a KVM e a aplicação em dispositivos móveis [19].

O JADE é composto de várias classes, ilustrado na figura 4.7, porém duas tem maior destaque: classe *agent* e classe *behaviour* (comportamento). A primeira é superclasse de todos os agentes e a segunda representa o comportamento dos agentes. Existe um forte relacionamento entre elas conforme é apresentado pela figura 4.6. Geralmente os agentes precisam ter pelo menos um comportamento instanciado, podem haver mais, porém, pelo menos um é o ideal segundo [19].

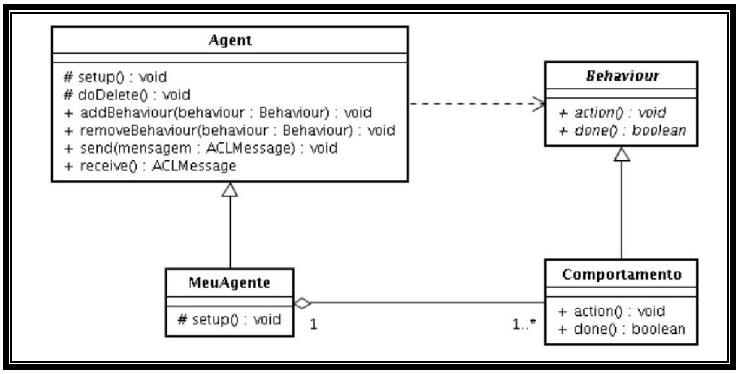


Figura 4.6 – Relacionamento entre as classes Agent e Bahaviour [19].

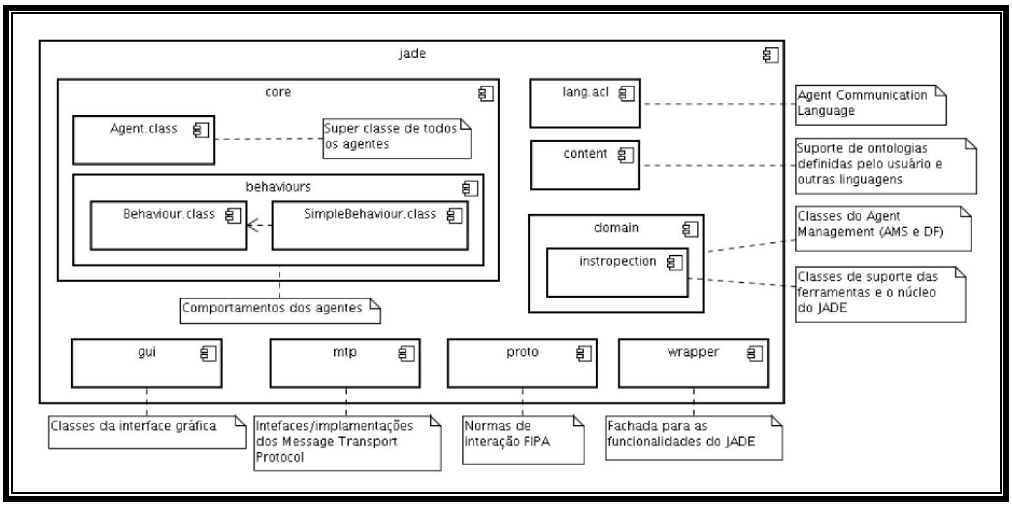


Figura 4.7 – Classes do JADE. A classe Agent é a super classe de todos os agentes. [19].

Finalizando, existem sete ferramentas internas no JADE para apoiar o gerenciamento dos agentes. São elas:

- **RMA (*Remote Management Agent*):** gerencia e controla os agentes. Através desta, todas as outras podem ser iniciadas;
- ***DummyAgent***: útil para monitorar e depurar os agentes;
- ***Introspector Agent*:** facilita o monitoramento do ciclo de vida dos agentes e seus comportamentos em execução (por exemplo, troca de mensagens);
- **DF (*Directory Facilitator*) GUI:** é uma ferramenta gráfica para interagir com qualquer DF, inclusive com integração entre diferentes DF's;
- ***LogManagerAgent*:** é um agente registrador de eventos ocorridos com os agentes em tempo de execução;
- **SocketProxyAgent:** é um agente roteador entre plataformas JADE e em uma conexão TCP/IP normal.

Mais informações sobre o arcabouço JADE podem ser encontras em vários documentos e manuais disponíveis em [20].

4.7 Trabalhos e Modelos Usando a DT e/ou MAS Encontrados na Literatura

Os principais trabalhos inspirados pela DT podem ser vistos em [2] e [3]. Estes autores são pioneiros da segunda geração dos AIS. Porém, muitos outros foram apresentados [87], [123], [191]. Nesta seção serão abordados os mais influentes.

No primeiro trabalho são apresentados detalhes da DT (este trabalho pode ser visto como uma expansão do trabalho [2]) sob o aspecto da segurança computacional. A principal característica é a analogia dos AIS com os IDS. Os autores questionaram a nova abordagem que poderia ser a ligação entre esses dois paradigmas. Aickelin et al. [3] cita a escalabilidade como um sério problema para a expansão dos AIS sobre ambientes reais e execução em tempo real. Segundo o autor, naquele momento, os trabalhos com AIS aplicados aos IDS estavam passando por certo declínio, haja vista a falta de aplicabilidade para sistemas reais (isto é, o problema da escalabilidade acabou por prejudicar a possibilidade do sistema atingir um patamar que pudesse ser desenvolvido no nível de mercado).

Num trabalho posterior, Aickelin et al. [4] fizeram uma revisão das abordagens imunológicas existentes e aplicadas sobre detecção de intrusão. Mas, foi no trabalho de Greesmith et al. [91] em 2005 que foram introduzidas as idéias de um modelo prático usando os conceitos da DT e o sistema inato. Naquele trabalho, os autores apresentaram a primeira versão do que seria o algoritmo DCA, ou seja, algoritmo das células Dendríticas. O algoritmo já apresentava os sinais PAMP, DS, SS e sinais de inflamação como entrada e os sinais CSM, SEMI e MATURO como saída, ou seja, a parte básica do DCA já estava desenvolvida. Como principal característica do DCA, os autores afirmaram não tentar nenhuma aprendizagem dinâmica através do algoritmo. No caso dos antígenos, cada linha da base de dados representava um antígeno, sendo identificados por um número correspondente. A numeração nesse caso é importante para identificar o antígeno no processo de agregação. Para testar e validar o modelo, os autores utilizaram dois experimentos usando a base de dados [34]. Os resultados encontrados mostraram 99% de acertos. Um dos fatores importantes do trabalho é que para alcançar o resultado não houve necessidade de treinamento anterior, fator muito importante nas abordagens anteriores do AIS assim como da maioria das técnicas de inteligência computacional.

Ainda em 2005, Twycross e Aickelin [196] apresentaram um framework conceitual para o desenvolvimento de algoritmos e sistemas sobre a nova geração dos AIS. Esse projeto é inspirado no trabalho [26], que construiu dois algoritmos representando a interface entre os AIS e os problemas do mundo real. No mesmo ano, outro trabalho [122] foi apresentado aplicando os conceitos do sistema inato e da DT sobre detecção de *worms*[61]. Este foi o primeiro trabalho que utilizou os conceitos das células T na DT, ou seja, uma representação mais completa do sistema inato do IS.

[61] Programa semelhante a um vírus, ele se auto-replica e não precisa do hospedeiro para se propagar. É um programa completo. Ver mais em: http://pt.wikipedia.org/wiki/Worm

Em 2006, os mesmos autores do framework conceitual apresentaram uma plataforma real de desenvolvimento para o novo paradigma. Baseado no *Project Danger* [3], o *framework* foi nomeado como *Libtissue* [197] e, segundo os autores, permite o desenvolvimento e teste de algoritmos AIS para os problemas do mundo real baseados nos princípios do sistema inato. O *framework* foi desenvolvido sobre a arquitetura cliente-servidor separando dados coletados de dados processados. Os conceitos, assim como os protótipos (isto é, objetos de programas) de antígenos e sinais já estão desenvolvidos, bastando apenas instanciá-los.

O primeiro trabalho a usar a *Libtissue* foi [92] e a partir dele foram apresentados esclarecimentos e articulações formais sobre o novo algoritmo (DCA). Segundo os autores, as DCs são os componentes de detecção desenvolvidos dentro da *Libtissue*. Se, no trabalho anterior os experimentos foram baseados em uma base de dados sem relação com a segurança computacional, validando o algoritmo como um bom classificador, neste trabalho, os experimentos foram realizados sobre um problema clássico de segurança: varredura de portas[62] (*Port Scan*). Nesse trabalho também são apresentadas as muitas variáveis e parâmetros de entrada do DCA. Os resultados mostrados no trabalho apresentaram comportamentos inéditos (isto é, não esperados) durante o processamento do algoritmo, segundo os autores, esses comportamentos foram classificados como bons resultados. Entretanto, também foi percebido aumento de falsos positivos relacionados à sensibilidade do algoritmo com os sinais de entrada. Os autores afirmaram ser o algoritmo bastante sensível as escolhas feitas para os sinais, ou seja, é preciso definir cuidadosamente os sinais de entrada.

No trabalho [90] os autores apresentaram uma extensão do trabalho anterior. Neste caso, o DCA foi testado dentro do framework *Libtissue.* Neste trabalho foi realizado testes para verificação do melhor valor para o limiar de migração. Os valores encontrados mostraram melhores resultados para um valor fixo e não variável, como era a idéia inicial. Portanto, o valor 1 obteve melhor desempenho (isto é, gerou menos erros ou falsos alarmes) durante os experimentos. Mas, o ponto de destaque deste trabalho foi a aplicação do modelo em tempo real para detecção de programas de varredura. Com resultados bastante satisfatórios, esse trabalho abriu caminho para responder o questionamento encontrado em [3].

No trabalho [89] já em 2006, os autores aplicaram novos testes para o DCA. No caso, os testes foram realizados para verificação da sensibilidade para múltiplos sinais por categoria

[62] Para se comunicar em rede, um computador usa portas específicas para se conectar. Ferramentas de varredura procuram essas portas com intuito de mostrar ao atacante quais portas estão abertas e possivelmente vulneráveis.

(isto é, há mais de um sinal para cada tipo) e da execução dos mesmos sobre problemas reais de grande escala. Para isso, dois experimentos foram realizados para análise de varredura de pacotes TCP SYN, considerado um dos mais difíceis de detectar. O DCA foi então aplicado sobre a estação vítima para análise do comportamento da mesma. Portanto, para este trabalho, os pontos mais importantes foram: (i) o uso de dois sinais por categoria e (ii) a aplicação em tempo real para ambiente complexo.

Após alguns anos de estudo, testes e validação, em 2007 são publicados os principais trabalhos envolvendo a DT. Dentre todos o que se destaca é a tese de doutorado de Greensmith [88] na qual é validado o DCA como algoritmo da nova geração de AIS. No mesmo ano, o trabalho [97] apresentou a primeira aplicação do DCA e dos conceitos da DT sobre o paradigma dos agentes inteligentes. No modelo foram criados três tipos de agentes: (i) agente antígeno, (ii) agente DC e (iii) agente TC. Segundo os autores, a cada segundo é criado um agente antígeno. No modelo, cada DC é um agente e possui todas as características peculiares a elas (por exemplo, o estado). Vários agentes DC são criados no estado imaturo, após receber mensagens selecionadas pelos agentes antígenos, o agente DC captura os atributos do antígeno, para o processamento dos sinais. Quando o valor de CSM acumulado ultrapassa o limiar de migração, o agente se move até o modulo de decisão para decidir qual o caminho será tomado. Após este processo, o agente é terminado e um novo agente é criado e acrescentado a população. A figura 4.8 mostra um diagrama do modelo em questão.

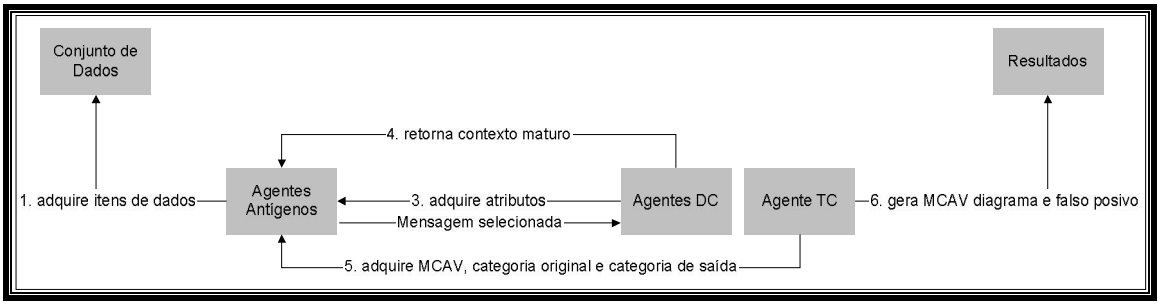


Figura 4.8– Diagrama do modelo AnyLogic, adaptado de [97]. Nele é possível ver as interações entre os agentes.

No caso dos agentes TC, o modelo usou apenas para gerar o gráfico dos valores de MCAV, ou seja, fazendo analogia entre o HIS e o modelo, não há nada em comum entre o agente TC e as células T.

Com resultados animadores, o DCA serviu de inspiração para outros pesquisadores fora do *Danger Project*. Projetos como aqueles encontrados em [81] um modelo conceitual de

MAS no qual cada agente possui é uma abstração da DT, (isto é, cada agente tem em seu interior todo o esquema da DT incluindo as DCs e as células T), [131] e o modelo conceitual denominado DCML. O modelo DCML é um modelo bastante completo na qual abrange todo a DT, incluindo as células T que são bem mais desenvolvidas que no modelo encontrado no *Project Danger*. Em [37] é estudado um novo conceito do HIS juntamente com a DT: a anergia. Esse é um processo no qual o organismo, ora sensível a um agente patógeno ou substância patogenia, torna-se incapaz de reagir. O estudo da anergia é bastante importante na biologia, e por isso, os autores investigaram os conceitos para a redução de falsos positivos. Na verdade, o trabalho é uma extensão do estudo relatado em [38].

O modelo conceitual MAAIS (*Multi-agent Artificial Immune System*) encontrado em [82] é o mais parecido com o modelo desenvolvido para esta dissertação. Ele é baseado na arquitetura encontrada em [123] e pode ser visto como uma expansão do modelo conceitual proposto em [81]. A arquitetura recebeu alterações em relação ao trabalho anterior com a inclusão de uma arquitetura cliente-servidor. O papel do servidor é dar assistência para os agentes que monitoram as estações. Cada agente possui uma arquitetura interna que envolve um modelo completo da DT. O servidor nesse caso não possui agente, mas dois módulos sendo (i) de análise e (ii) de resposta. A figura 4.9 apresenta a arquitetura completa do MAAIS.

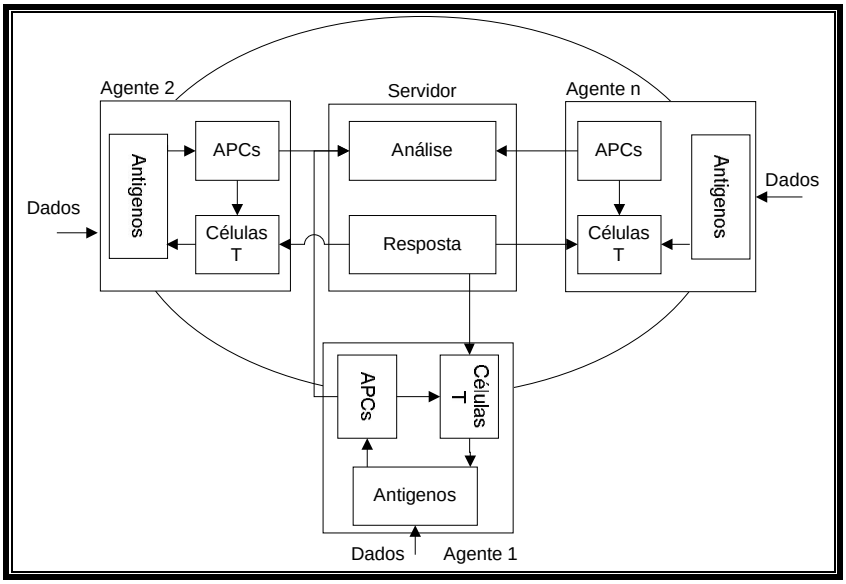


Figura 4.9 – Arquitetura do modelo MAAIS, adaptada de [82]. Importante observar que a arquitetura de agentes nas estações e um servidor central como suporte aos eventos encontrados pode ser um problema, caso venha falhar, o sistema todo pode sucumbir. Apesar disso, esse modelo possui semelhanças com o modelo proposto nesta dissertação.

Nos últimos dois anos, vários trabalhos foram desenvolvidos apresentando (i) melhorias no DCA [93], (ii) novas aplicações para o DCA [98], (iii) comparações com outras abordagens [94] e (iv) apresentação de novos modelos de algoritmos baseado nas DCs [144].

Inspirado nos modelos e trabalhos relacionados nesta seção, no próximo capítulo é apresentado o modelo desenvolvido neste trabalho.

Capítulo 5

Modelo de Sistema de Detecção de Intrusão Imuno-inspirado

O modelo concebido e desenvolvido aqui apresenta um conjunto de agentes distribuídos hierarquicamente com funções específicas, inspiradas nos conceitos da DT. Seis agentes principais foram desenvolvidos, conforme pode ser visualizado na tabela 5.1. A inspiração encontra alguma analogia militar, em que tarefas mais simples são realizadas por mais agentes (soldados) e, à medida que se caminha na hierarquia mais importante, são as ações tomadas por menos agentes (oficiais de comando).

Tabela 5.1 – Analogia dos agentes com o HIS e suas abreviações.

Nome	Analogia com o HIS	Abreviação
Agente Básico	Células Dendríticas	***Abas***
Agente Subalterno	Células T auxiliares	***Asub***
Agente Intermediário	Timo	***Aint***
Agente Superior	Células T reguladoras	***Asup***
Agente Mensageiro	-	***Amen***
Agente Logger	-	***Alog***

O modelo com todos os agentes é ilustrado na figura 5.1. Analisando, o *Abas* detecta a invasão, o *Asub* combate o intruso, o *Amen* envia as mensagens entre os agentes, o *Aint* identifica novos invasores e cria *Asub* específicos, o *Asup* atua de maneira geral no sistema usando o conceito de mobilidade, e o *Alog* registra tudo o que acontece para análise posterior à procura de falhas no sistema.

Dois objetivos principais são propostos com esse modelo: (i) automatização do sistema e (ii) adaptação a novos tipos de ataques e variações de ataques conhecidos.

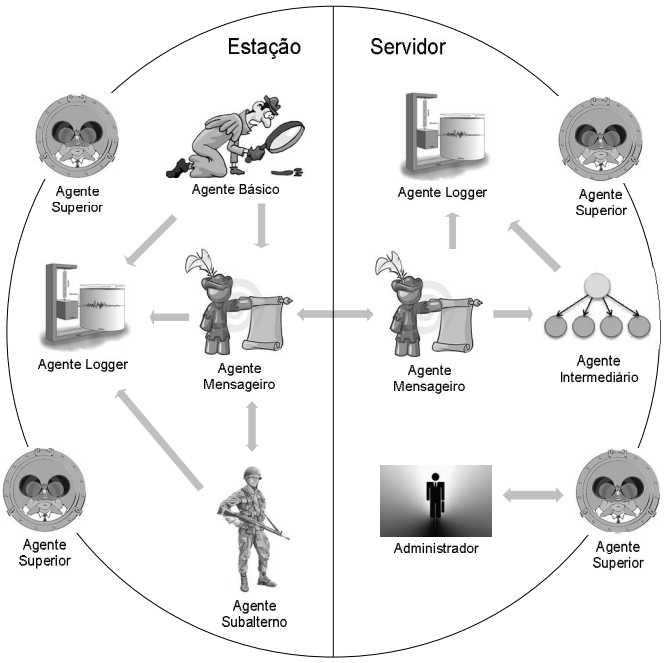


Figura 5.1 – Ilustração através de *clip-art* apresentando os agentes e suas funções.

Conforme apresentado no Capítulo 4, os agentes possuem características interessantes que permitem serem usadas neste trabalho [172]. Os agentes precisam de um motor para processar informações colhidas do ambiente. Dessa forma, os conceitos da DT foram embutidos nos agentes, alcançando um modelo representativo do HIS para detecção de intrusão. Nas próximas seções, serão apresentadas as características principais de cada agente, bem como seu funcionamento e desenvolvimento. É importante salientar que o modelo foi planejado e desenvolvido para aplicação na camada de enlace do padrão selecionado [105]. Portanto, não serão tratados pacotes, mas sim quadros.

5.1 Agente Básico (*Abas*)

O nome básico deste agente não significa simplicidade, mas refere-se a sua condição na arquitetura geral do modelo. A sua principal grande responsabilidade é detectar sinais de falha (por exemplo, ocasionados por ataque) na rede através da aplicação dos conceitos da DT. Portanto, sua função é considerada básica para o sistema de detecção. Conforme ilustrado na figura 5.2, em sua arquitetura interna, ele possui (i) um módulo de coleta de quadros e pacotes, (ii) um analisador dos dados, (iii) um módulo de mensagens e (iv) o módulo DCA.

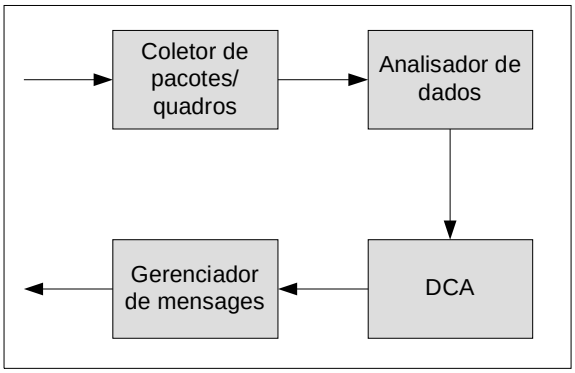


Figura 5.2 – Agente básico e sua estrutura interna com quatro módulos.

Apenas um *Abas* é instanciado por estação. Dessa forma, a quantidade deles dependerá da quantidade de nós na rede. Em analogia ao HIS e à DT, ele tem a função das DCs, ou seja, percebe estímulos no ambiente (sinais) através dos quadros (antígenos) trafegando pela rede, processa-os e, de acordo com o resultado, entrega o antígeno para o *Asub* (representando a célula T) que irá responder de acordo com as características do antígeno.

5.1.1 Coleta de quadros

A coleta de quadros não é feita diretamente pelo *Abas*, mas sim por uma ferramenta bastante conhecida, denominada Airodump-ng e pertencente ao pacote de ferramentas para redes sem fio Aircrack-ng [4]. Após a coleta, os dados são convertidos para o formato texto usando a ferramenta Wireshark [211]. Por ser de construção modular, é possível que este agente receba ferramentas de coleta especialmente construídas ou o uso de outra ferramenta já existente. Explicações mais detalhadas de como é realizada a coleta e o tratamento dos dados nos arquivos poderão ser vistas no Capítulo 6, no detalhamento dos experimentos.

5.1.2 Analisador de Dados

Este módulo tem relevada importância para o *Abas*, pois é através dele que são preparados os dados de entrada do algoritmo DCA. Após a leitura dos dados do arquivo, eles são analisados e preparados para formar a matriz de sinais e o vetor de antígenos, os quais serão utilizados como entrada no módulo DCA.

Cada quadro é transformado numa estrutura de dados que corresponde ao antígeno, que nada mais é do que um vetor contendo os valores do pacote. Os sinais são coletados e tratados de acordo com os antígenos, não podendo ser separados um do outro.

Para este trabalho foi escolhida a freqüência na linha do tempo como delimitadora do vetor de antígenos. Portanto, os antígenos podem ser separados por segundo, minuto ou hora, dependendo do modelo de análise escolhido.

5.1.3 Módulo DCA

Este é o módulo principal do *Abas* e é através dele que o agente fará o processamento entre os sinais coletados e os antígenos para identificar anomalias. O algoritmo DCA foi escolhido como base desse módulo e será descrito nesta seção. Algumas alterações foram necessárias para adaptação ao modelo de rede sem fio.

5.1.4 Modificações no DCA

Nos trabalhos existentes na literatura, como exemplo, [6] e [89], o modelo da estrutura dos sinais e dos antígenos eram baseados em chamadas do sistema realizadas por programas em execução (por exemplo, NMAP, SSH) e, através das quais, sempre um antígeno tinha uma identificação que o relacionava ao programa que o havia gerado. Dessa forma, ao executar o DCA, o resultado do grau de anomalia presente no antígeno emergia permitindo que fosse possível diferenciar cada problema com seus antígenos. Porém, neste trabalho, não foram utilizadas chamadas de sistemas e nem mesmo os antígenos foram relacionados a um identificador que referenciavam sua origem. Portanto, pode-se dizer que este modelo usa os quadros de forma direta para gerar os sinais e os antígenos. Cada quadro que entra ou sai da estação gera um antígeno que o representará. A principal conseqüência é que não haverá antígeno repetido no sistema, sempre serão antígenos novos (isto é, a cada segundo são renovados os antígenos e os sinais).

Para que pudessem ser correspondidos os antígenos aos sinais, foi usada a linha do tempo como atributo. Dessa forma, os sinais foram relacionados aos antígenos por cada fração de tempo que, no caso foi definido por segundo (poderia ser minutos ou outra forma qualquer).

A cada segundo, um ciclo do DCA é executado, ou seja, todos os antígenos coletados durante o segundo são apresentados ao DCA que os processará de acordo com os sinais de

entrada para aquele determinado segundo. Após os primeiros testes, surgiram problemas com a quantidade de DCs na população e os resultados do processamento. A cada segundo, uma quantidade variável de antígenos é apresentada, porém, os sinais são os mesmos para todos os antígenos daquele instante e, por isso, os valores processados para todas as DCs são idênticos. Após testes, verificou-se que, mesmo com o limiar de migração de cada célula seja diferente, a diferença é muito pequena, ou seja, não interfere de maneira substancial para o processamento, dessa forma, quando foram testadas *n* DCs, todas apresentaram os mesmos resultados após o processamento dos sinais. Portanto, outros testes foram realizados, inclusive com uma DC apenas. Os resultados apontaram boa eficiência na detecção de anomalias, conforme estão apresentados no Capítulo 6.

No DCA original existe uma variável que controla a quantidade de antígenos que são apresentados as DCs a cada ciclo, porém, para o presente modelo é necessário que todos os antígenos sejam apresentados as DCs (seja uma ou várias). Nesse caso, não seria possível fazer divisão dos antígenos dentro de um mesmo segundo, na verdade seria custoso para a execução e poderia gerar falsos alarmes. Portanto, todos os antígenos pertencentes a um mesmo segundo são apresentados a DC.

O cálculo do MCAV, ou seja, valor que mede o grau de anomalia dos antígenos não foi aplicado por não ser substancialmente importante nesse caso. O fato é que, por não haver distinção entre os antígenos, ou seja, todos são oriundos dos quadros, não havia necessidade de tal cálculo. Outro fator que impediu o uso do MCAV foi a forma como foram inicializados os valores do contexto para os antígenos. No modelo proposto, os valores do contexto são relacionados por segundo da seguinte forma: se após o processamento de todos os antígenos de um segundo a DC apresentar maior valor para maturo, todos os antígenos receberão aquele valor, caso contrário, todos receberão o contexto semi-maturo. Esta abordagem pode gerar problemas caso existam antígenos com valor de contexto igual a maturo, caso a maioria não estejam maturos. Esse fato pode gerar falsos alarmes. Como solução para o problema, para os segundos em que a DC estiver com contexto semi-maturo, se ela contiver algum antígeno com contexto maturo, este é enviado para análise. Se for constatado que o antígeno representa um ataque, a DC será avaliada novamente, podendo, inclusive mudar de estado.

Portanto, o cálculo dos erros para o processo de detecção do modelo é realizado verificando qual o contexto gerado para cada segundo. O contexto maturo recebe 1 como valor enquanto semi-maturo recebe 0.

5.2 Agente Subalterno (*Asub*)

O *Asub* é uma abstração da célula T do HIS. Sua principal função no modelo é receber os antígenos que são apresentados pelos *Abas* sobre o contexto maturo. Inicialmente, este agente foi planejado para representar cada um apenas um tipo de ataque. Após análise experimental, foi verificado que essa abordagem poderia gerar muitos tipos de agentes parecidos (isto é, variações bastante sutis de certos tipos de ataques que poderiam ser resolvidos por um único agente) tornando-se possível, dessa forma, a agregação sobre um mesmo agente.

Portanto, para cada estação existe um container de *Asub* que é instanciado com valor nulo, ou seja, vazio. Após a primeira detecção por parte do *Abas*, o container irá receber os antígenos do mesmo e verificará se existe algum modelo de *Asub* com afinidade para os antígenos. Caso exista, ele irá verificar a rotina para contra-ataque e tomará as medidas cabíveis. Como este trabalho não tem foco na resposta aos ataques, as rotinas criadas para os *Asub* são bastante simples (por exemplo, a desativação da rede na estação) servindo apenas como parte do modelo de detecção.

Caso não haja nenhum agente com afinidade para os antígenos, o sistema acionará o agente mensageiro para que envie uma mensagem pela rede até o servidor onde encontrará o *Aint* (ver seção 5.3) para que o mesmo possa fazer a identificação dos antígenos e do tipo de problema que são responsáveis.

O processo de afinidade é uma simples comparação entre a estrutura do antígeno e a estrutura do agente. O *Asub* possui um vetor de tamanho 7, sendo cada posição representada da seguinte maneira:

- **posição 1:** tipo do quadro;
- **posição 2:** se origem é a estação;
- **posição 3**: se origem é a rede;
- **posição 4:** se origem é o AP;
- **posição 5:** se destino é a estação;
- **posição 6:** se destino é a rede;
- **posição 7:** se o destino é o AP.

Cada *Asub* pode ter mais de uma estrutura interna, ou seja, a cada vez que o sistema encontrar um antígeno com estrutura diferente para um mesmo tipo de problema (por exemplo, um ataque de-authentication pode ser direcionado ou em modo *broadcast*, para cada

um, é preciso uma estrutura diferente), o sistema irá criar uma nova estrutura que seja compatível com a estrutura do antígeno.

Após ser criado ou atualizado, o *Aint* o enviará para a estação que enviou a mensagem solicitando a identificação. Dessa forma, o *Asub* usa, inicialmente, o conceito de agentes móveis. Ao chegar à estação irá executar a rotina de contra-ataque originalmente inserida pelo *Aint* e depois ficará desativado no container de *Asub* até que outro ataque do mesmo tipo seja detectado. Aqui é possível verificar uma analogia entre o HIS e o modelo, no caso, as células de combate, representadas pelos *Asub* são células que podem representar a função de células de memória, pois após serem criadas para um determinado tipo de problema, elas continuarão na estação fazendo parte do sistema.

Quando é criado um novo *Asub*, ele é replicado para todas as estações da rede a fim de que todas tenham os mesmos agentes de contra-ataque (isto é, analogia com a seleção clonal dos AIS).

A estrutura, relativamente simples, do *Asub* pode ser vista na figura 5.3.

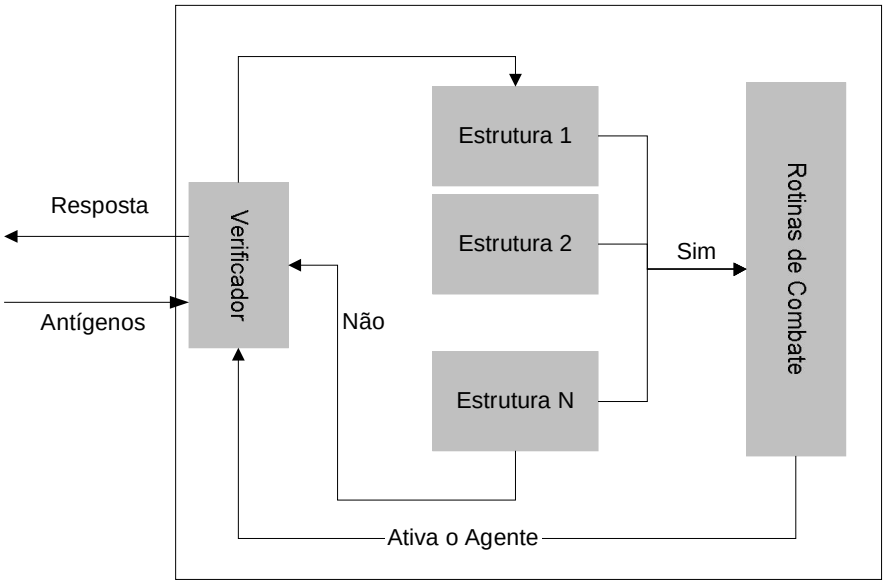


Figura 5.3 – Estrutura interna do *Asub*. Quando os antígenos são apresentados pelo *Abas*, é verificado se existe algum *Asub* que possua afinidade com o antígeno, caso seja positivo, o agente é ativado, caso negativo, o agente devolve os antígenos para o *Abas* que deverá pedir ajuda ao *Aint*.

5.3 Agente Intermediário (*Aint*)

Um dos principais agentes do modelo proposto, ele tem a função de identificar os eventos anômalos ocorridos no sistema que não puderam ser identificados pelos *Asub* das estações

(isto é, quando não há afinidade entre algum agente de resposta e os antígenos apresentados pelo *Abas*). Para essa função, e por sua simplicidade, é usada a técnica de classificação e reconhecimento de padrão denominado *naïve Bayes.*

Para que possa haver classificação é preciso que haja uma base de dados preparada com informações importantes e organizadas para que o classificador possa ser treinado e validado como uma ferramenta de classificação eficiente. Dessa forma, para esse trabalho foi criada uma base de dados contendo informações sobre alguns tipos de situações nas quais são anormais para o tráfego na rede sem fio.

Uma estação externa à rede controlada pelo sistema que envia um quadro *de-authentication* direcionado a certa estação interna, ou até mesmo para todas as estações, é uma atividade irregular que é classificada pela *naïve Bayes*.

Tabela 5.2 – Modelo da base de dados para treinamento, teste e validação do classificador *naïve Bayes*.

Origem	Destino	Tipo	Tamanho	Classificação
05	02	Deauthentication	36	Ataque
01	02	Deauthentication	36	Normal
02	02	*Data*	1-1500	Normal
02	01	*Data*	36	Ataque
Legenda: 01 = AP, 02 = estação da rede, 03 = endereço vazio, 04 = endereço *broadcast*, 05 = endereço externo.				

Existem dois padrões definidos na base de dados: (i) ataque e (ii) normal. Dessa forma, quando uma mensagem é recebida da estação, os antígenos serão apresentados para o *Aint* que irá analisá-los usando o classificador para tentar identificar o tipo do ataque e criar um *Asub* específico para a resposta.

A criação e instanciação do *Asub* são feitas em tempo de execução e para isso foi necessária algumas mudanças no framework JADE. Foi adicionada uma correção na classe abstrata do agente na qual estende todos os agentes do modelo.

Conforme já citado, o foco deste trabalho não é a eficiência do contra-ataque, mas sim a automatização e adaptação do modelo de detecção de intrusão. Portanto, o *Asub* é criado com uma rotina simples de resposta, apenas uma chamada para o *log* informando que foi executada a rotina.

Existe apenas um *Aint* instanciado no módulo servidor, que tem uma função de extrema importância para a adaptação do sistema na detecção de novos tipos de ataques. Observando o modelo, é possível existir mais de um *Aint* no sistema, porém, nunca dois na

mesma estação. Para que seja possível a existência de dois ou mais *Aint*, é necessário que todo o conhecimento seja compartilhado entre os servidores. Em se tratando de um ambiente complexo com bases de treinamento diferentes em cada servidor para uma maior granularidade na classificação, alguma perda de desempenho pode ser observada.

A função do *Aint* em analogia ao HIS é semelhante ao Timo ou a medula óssea, (isto é, local onde são treinadas as células T e B do IS).

5.4 Agente Superior (*Asup*)

O *Asup* é o agente responsável pela auditoria do sistema. Algumas de suas qualidades são a mobilidade e autonomia perante o sistema, ou seja, é o agente do modelo que tem habilitada a variável mobilidade por todo o tempo de execução.

Podem existir vários *Asup* instanciados no sistema. Primeiramente eles são instanciados no servidor e a quantidade deve variar de acordo com o tamanho da rede. É sugerido que redes com muitos nós e com complexidade elevada (por exemplo, redes heterogêneas) tenham pelo menos um *Asup* para cada estação existente na rede. Em caso de redes simples, apenas um agente pode cumprir bem suas funções.

Para realizar a auditoria o agente utiliza o arquivo de *log* gerado pelo *Alog*. Para isso, foi criado um padrão de mensagens com códigos específicos para definição das atividades de cada item do sistema. Desta forma, quando há um evento desconhecido ocorrido durante a execução dos agentes (todos), este será impresso no arquivo de *log* com um código indicando os problemas. Quando isso ocorrer, o *Asup* irá investigar o problema fazendo um rastreamento das atividades executadas durante um determinado tempo antes e depois do código encontrado. Outros problemas também poderão ser detectados através da mineração de dados pela procura de palavras consideradas problemáticas para o sistema (por exemplo, falha na detecção, erro de leitura, etc.). Esta função permitirá que o sistema não cometa erros, ou pelo menos, evitará que um erro seja encoberto ou prejudique a segurança posteriormente.

A função deste agente também pode ser vista como análoga ao processo de supressão, já que, se houver alguma falha dos agentes que executam os processos de segurança do sistema, ele poderá tomar atitudes para evitar a autodestruição do sistema.

A estrutura do *Asup* é simples, ele apenas faz a leitura do arquivo de *log* de cada componente do sistema (isto é, inclusive do servidor) à procura de falhas do sistema e de seus processos e componentes.

O *Asup* irá, após um determinado tempo, ou, em questão de chamado das estações, deslocar-se para a rede, de estação em estação, procurando problemas locais do sistema. Quando eventualmente encontrados, o *Asup* irá criar um relatório que é passado para o Administrador do sistema. Dessa forma, o *Asup* possui uma função importante para um IDS: auditar o sistema no lugar de um ser humano.

Para evitar que, ao deslocar para uma estação da rede a fim de realizar suas rotinas de segurança, o *Asup* se perca por algum problema na estação, ao sair do servidor, o *Asup* deixa um agente clone que possui as mesmas características de si mesmo. Ao sair, um contador é disparado e, a cada minuto o *Asup* informa seu local de execução. Caso o sistema não consiga definir sua posição, irá abandoná-lo e criar outro clone para que o clone atual assuma às vezes do pai.

De certo ponto de vista, o *Asup* pode representar o equivalente ao controle das doenças auto-imunes, onde o sistema poderia equivocadamente combater a si mesmo.

5.5 Agente Mensageiro (*Amen*)

A única função deste agente é a troca de mensagens entre as estações e o servidor. Cada estação possui um *Amen* e para seu desenvolvimento foi utilizado o modelo de troca de mensagens do próprio JADE. No início deste trabalho não havia um agente para cuidar da troca de mensagens entre os agentes, o que gerou gargalo no *Abas* que tinha a responsabilidade de enviar o pedido de apoio do servidor. Portanto, ao criar um agente específico para essa função, o problema de sobrecarga do *Abas* foi resolvido e a centralização das mensagens tornou o sistema mais organizado e modular.

5.6 Agente *Logger* (*Alog*)

A fim de mitigar erros e falhas no sistema, foi desenvolvido este agente que registra “tudo” o que acontece no sistema durante o tempo de execução.

Depois de gravado o arquivo de *log*, a cada período de tempo (por exemplo, uma hora) o *Alog* fecha o arquivo e cria outro. Esse procedimento é necessário para que o *Asup* possa utilizar o arquivo para suas auditorias. O tempo para fechamento e abertura de novo arquivo pode ser configurado de acordo com o grau de risco do sistema.

Um sistema de mensagens foi criado para que pudesse ser facilitado o entendimento humano do arquivo log. O formato escolhido foi um código de três dígitos com uma

abreviação do problema a que se refere. Quando o *Alog* encontra um problema, verifica na tabela o tipo do problema e insere no *log* um código e um resumo do tipo. Desta forma, evita que tenha arquivos de *log* muito grandes e facilita uma possível auditoria humana no sistema.

5.7 Arquitetura do Modelo

A arquitetura do modelo pode ser vista na figura 5.4 e 5.5, onde são mostrados os componentes e suas funções na forma de um fluxograma. A figura 5.4 mostra um fluxograma da estação e a figura 5.5 apresenta o fluxograma do servidor. Todos os componentes foram exemplificados durante o decorrer deste capítulo.

Através das figuras é possível ver que o agente *logger*, juntamente com o agente superior, tem uma função de extrema importância. É ele quem registra todas as atividades do sistema. Ao criar um arquivo de registros de eventos (*log*) é possível acompanhar todo o fluxo de informações durante a execução. Para esse modelo, o arquivo de *log* serve apenas para que o agente superior faça auditoria no sistema. Mesmo sendo simples, a auditoria é importante para cumprir com o objetivo da automatização do sistema.

Na estação existem: (i) um *Abas*; (ii) um container de *Asub*; (iii) um *Alog*; e (iv) um *Amen*. A quantidade de *Asub* dependerá da quantidade de vezes em que forem detectadas anomalias nas estações/rede.

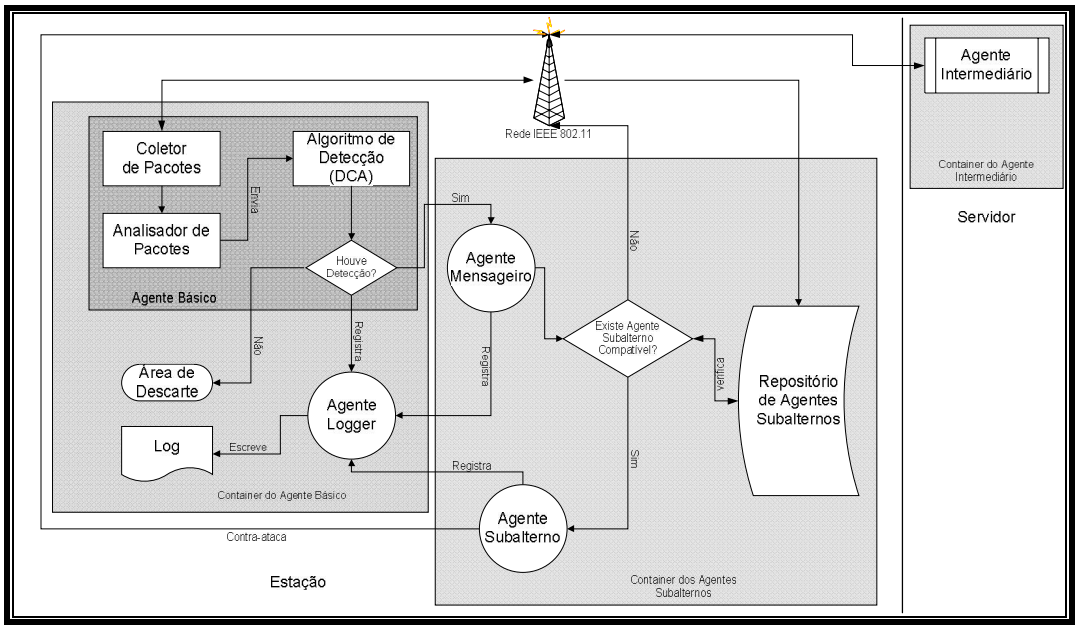


Figura 5.4 – Arquitetura da estação mostrando o fluxo de comunicação existente entre os componentes do sistema na estação.

Na figura 5.5 é mostrado um agente genérico. Esse agente é uma solução definida no modelo para que, quando houver um problema de identificação, ou seja, o agente intermediário não for capaz de definir o tipo do problema, um agente contendo uma rotina genérica (por exemplo, avisar o administrador, bloquear acesso da estação que originou o problema, executar ferramenta de varredura de vírus, entre outras) será criado na tentativa de solucionar momentaneamente o problema.

No servidor existem: (i) um *Aint*; (ii) *container* de *Asup*; (iii) um *Alog*; e (iv) um *Amen*. A quantidade de *Asup* dependerá da quantidade de estações e o tamanho da rede.

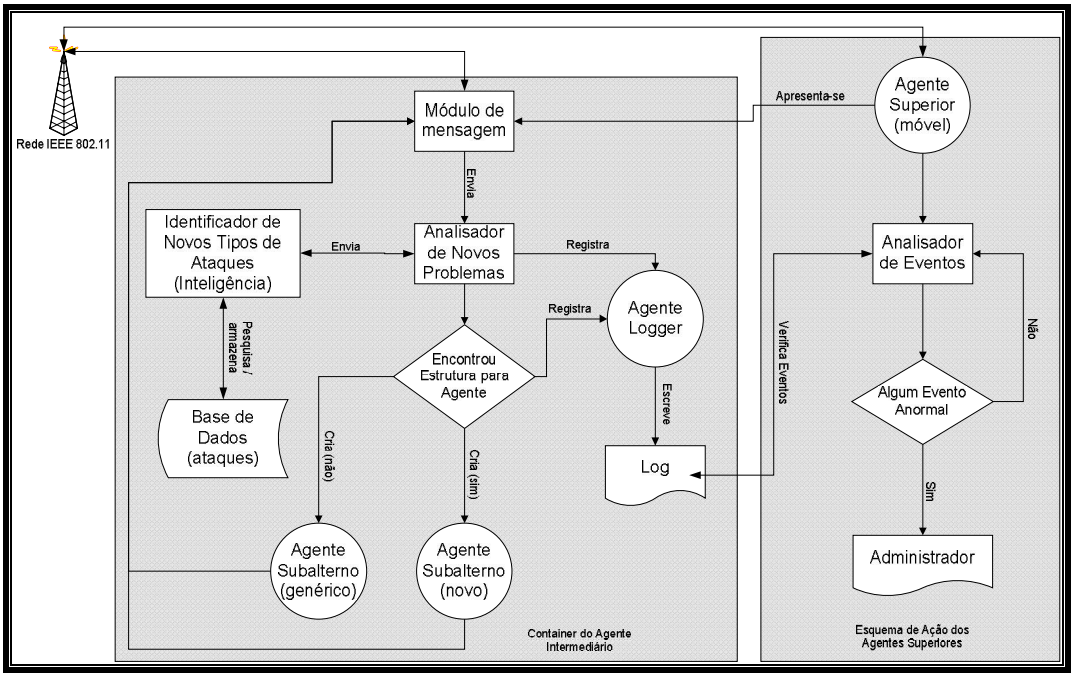


Figura 5.5 – Arquitetura do servidor mostrando o fluxo de comunicação existente entre os componentes do sistema do servidor e o modelo de criação dos *Asub*.

Em analogia ao HIS, os *Abas* e os *Asub* representam o sistema inato, principalmente os básicos que tem a função das DCs (Macrófagos) e são instanciados na estação. No caso dos subalternos, representam as células T. Tal analogia é ilustrada pela figura 5.6.

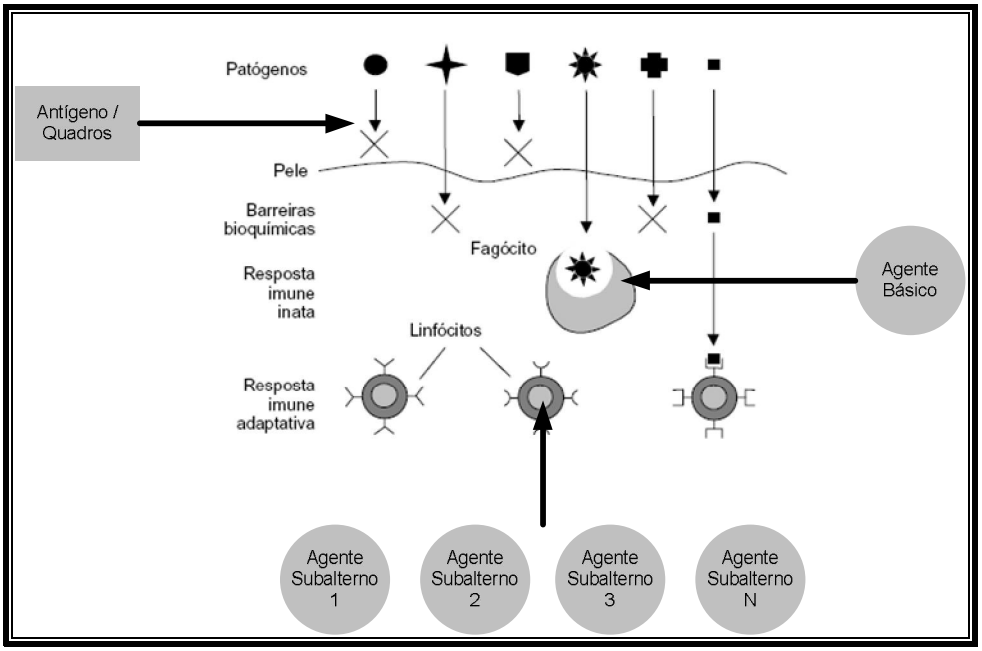


Figura 5.6 – Analogia entre os componentes do modelo desenvolvido e o HIS. Adaptado [55]

A figura 5.7 ilustra por outra visão a analogia entre o HIS e o modelo proposto. Nela pode ser visualizada a relação entre o agente intermediário e o Timus, o órgão responsável pela maturação das células T. As células T (que são linfócitos T, por estarem presentes nos linfonodos e nos vasos linfáticos) podem ser encontradas em vários tipos [55], [45], [46]: Linfócitos T Auxiliares CD4+ (*Helper*), Citotóxicos CD8+, Memória, Reguladores, Natural *Killer* e γδ (do grego, *gamma delta*).

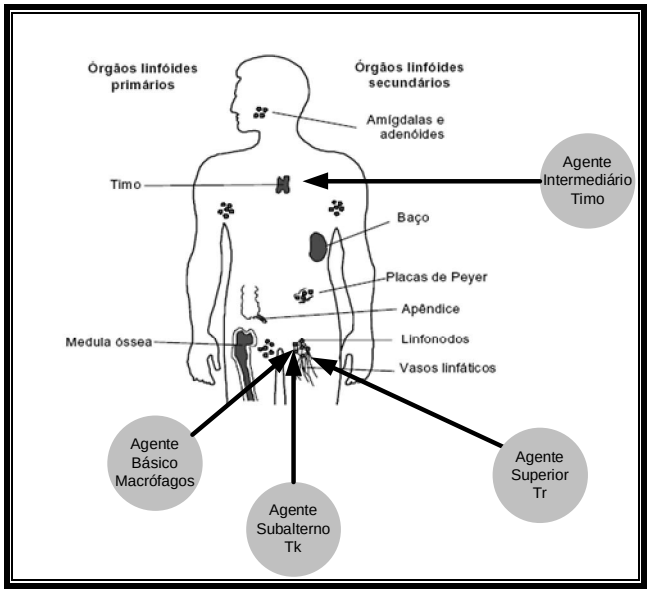


Figura 5.7 – Segunda visão análoga entre o HIS e o modelo desenvolvido. Adaptado [55].

An*Alog*amente, os agentes superiores podem exercer as funções de reguladores, natural killer e auxiliares. Enquanto que os agentes subalternos podem assumir as funções de citotóxicos (esses são os verdadeiros combatentes do organismo) e de memória. Os linfonodos e os vasos linfáticos podem ser vistos como a estação e a rede.

Numa observação simples, o modelo de MAS proposto possui agentes que representam todos os atributos da DT e dessa forma, pôde ser testado um modelo de AIS de segunda geração aplicado sobre os conceitos de um WIDS para detecção de anomalias no sistema.

Capítulo 6

Experimentos e Resultados Encontrados

Este capítulo tem o intuito de mostrar o funcionamento do sistema e de validá-lo à luz dos objetivos inicialmente propostos. Para isso, foram planejados alguns experimentos e simulações. Vários experimentos foram realizados durante a fase de desenvolvimento do sistema e serviram para validação do modelo inicial. Por brevidade, neste capítulo são apresentados apenas os experimentos realizados com todos os módulos do sistema proposto em funcionamento.

6.1 Definições dos Experimentos

A realização dos experimentos tem como objetivo (i) avaliar o sistema de detecção e (ii) o comportamento dos agentes quando em execução automática. Experimentos iniciais foram relatados no trabalho [48]. Porém, para aquele momento somente dois tipos de agentes foram testados e para um tipo de problema apenas. Portanto, os experimentos que serão mostrados nesse trabalho já contemplam todos os agentes e todos os tipos de ataques estudados.

Inicialmente, a idéia era aplicar todos os ataques em conjunto para avaliação do modelo. Entretanto, não foi possível tal aplicação pela dificuldade gerada em trabalhar com bases de dados muito grandes. A faixa de tamanho das bases coletadas durante os ataques completos mantinha valores entre 500 Mbytes e 1 Gbytes. Com esses valores, tornou-se impossível a conversão da base, antes em formato "*.cap*" (isto é, em bytes) para o formato ".txt". Desta forma, os experimentos foram divididos em oito, conforme mostra a tabela 6.1.

Tabela 6.1 – Divisão dos experimentos.

Experimento	Descrição
Experimento 1	Ataques de-authentication e Requisições ARP
Experimento 2	Ataque Hirte
Experimento 3	Ataque ChopChop
Experimento 4	Ataque Falsa Autenticação
Experimento 5	Ataque Interativo
Experimento 6	Ataque Cafe-Latte

Experimento 7	Ataque de Fragmentação
Experimento 8	Ataque sobre WPA

Sobre cada experimento são analisados alguns fatores que serviram para análise do funcionamento do modelo. Dentre os fatores se destacam (i) a avaliação da capacidade de detecção do algoritmo DCA modificado através da análise dos falsos alarmes e (ii) a verificação dos indicadores de funcionamento dos agentes que inclui (a) a análise da capacidade de identificação do tipo de ataque pelo classificador *bayesiano*, função do agente intermediário, usando novamente o estudo dos falsos alarmes, (b) a criação e clonagem dos agentes subalternos, (c) a atualização do agente subalterno quando houver variação de ataque (isto é, quando ocorre uma variação em um tipo de ataque já identificado antes pelo sistema), (d) a avaliação do tempo de resposta do sistema após encontro com o mesmo tipo de ataque em detrimento a um novo tipo, (e) a avaliação das atividades do agente superior na auditoria do sistema, (f) a avaliação da troca de mensagens entre os agentes e (g) a verificação do tráfego gerado pelo sistema na rede.

6.2 Configurações dos Ambientes

Dois ambientes foram montados para a realização dos experimentos. No primeiro, considerado bastante simples, existem apenas três estações e um AP sem a presença do servidor. Este ambiente foi montado para os experimentos iniciais nos quais podem ser vistos os resultados para ataques DoS no trabalho [48]. Para esse tipo de ambiente não havia o módulo servidor e serviu apenas para validar o DCA.

Para o segundo ambiente, mais três estações são inseridas, uma delas recebe a função de servidor. Para que os experimentos pudessem ter mais fatores reais, a rede estava conectada à WEB e todas as estações estavam liberadas para acesso. Em cada estação foi inserido algum tipo de tráfego (por exemplo, *download* de arquivos, acesso a chat, rádios e filmes).

Como protocolo de segurança, para os sete primeiros experimentos, o padrão WEP foi utilizado, mas, no oitavo experimento, foi habilitado o padrão WPA. A idéia é verificar o comportamento do modelo perante o ataque DoS realizado sobre este protocolo.

As estações possuíam configurações consideradas de última geração, com processadores modernos (por exemplo, *dual core*, *core* 2 *duo*) e memória RAM mínima de 1 GHz (somente uma estação possuía apenas 1 GHz, as outras, possuíam 2 e 3 GHz). Todas

continham uma placa de rede sem fio de distintos fabricantes (por exemplo, D-Link, Broadcom e Atheros). Quatro estações eram portáteis e, duas, *desktop*. Para o AP foi utilizado o modelo D-Link 624 com quatro portas para clientes cabeados e uma porta cabeada para acesso a WEB. O modelo do AP vem de fábrica com a versão IEEE 802.11g habilitada.

Para evitar que o sistema tivesse problemas de ataque sobre o servidor, o mesmo contempla uma segunda placa de rede, agora cabeada. Essa solução evita que o servidor sofra um ataque DoS oriundo da rede sem fio permitindo que as estações continuem a usar dos seus serviços (isto é, o módulo servidor é muito importante para o sistema, pois cria os *Asub* e mantém o controle através do agente superior.

Na figura 6.1, que apresenta os dois principais ambientes usados neste trabalho, é também mostrado o fluxo de informações durante os ataques. A primeira ação é a observação do atacante ao tráfego da rede através do ataque passivo usando a ferramenta Aircrack-ng. O segundo passo é a análise do tráfego coletado e o preparo dos ataques. Em terceiro é realizado o ataque sobre a rede. Nesse caso pode ser multidirecional (*broadcast*), conforme o quarto passo, ou então direcionado a uma estação apenas, no caso o quinto passo.

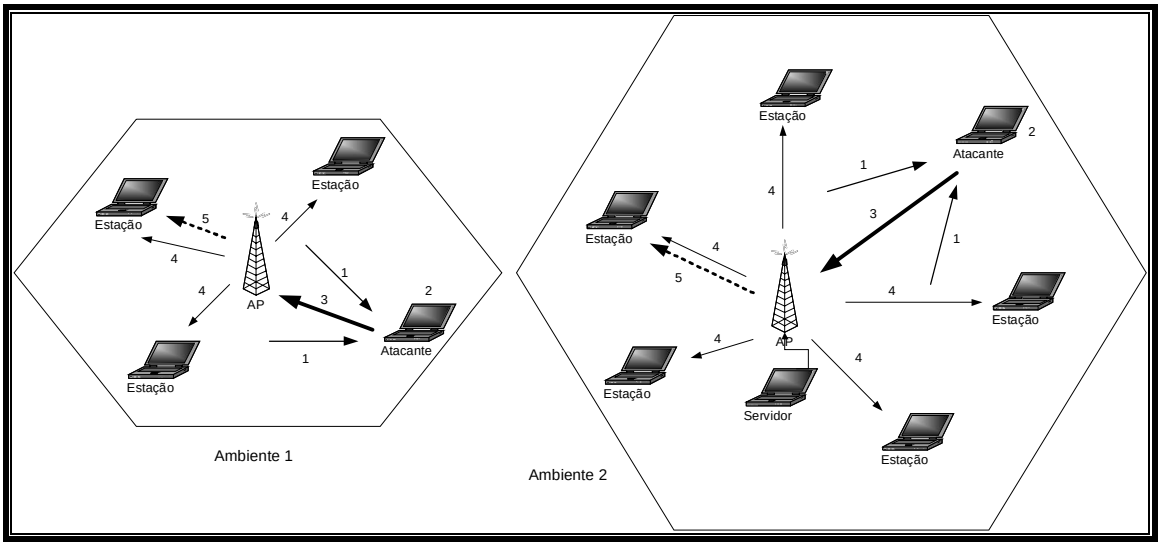


Figura 6.1 – Ambiente de ataque e o fluxo dos ataques realizados.

6.3 Definindo os Valores para os Parâmetros do Algoritmo de Detecção

Duas estruturas principais fazem parte dos parâmetros de entrada do algoritmo DCA: (a) matriz de sinais e (b) vetor de antígenos. Ambos, conforme comentados no Capítulo 5, são criados a partir da observação do ambiente de execução, no caso, as redes sem fio. Para este

trabalho, diferentemente dos outros na literatura [89], são usados os próprios quadros para formação das bases citadas. Portanto, os antígenos são criados a partir dos quadros, enquanto que os sinais são coletados a partir dos comportamentos encontrados no tráfego da rede associando-os aos antígenos por segundo.

A definição dos sinais que serão coletados para preenchimento da matriz de sinais é parte crucial para o sucesso do algoritmo [89]. Dessa forma, vários tipos de sinais foram utilizados durante a fase de desenvolvimento a fim de comprovar aqueles que melhor se adequassem ao modelo de detecção.

Nos primeiros experimentos, mostrados no trabalho [48], foram usados apenas três tipos de sinais de entrada, por tratar-se apenas de um tipo de ataque, no caso, negação de serviço. Porém, para os outros experimentos foram utilizados outros tipos de ataques, nos quais precisavam da definição de mais sinais para que fosse possível sua detecção. É possível o uso de mais de um sinal por categoria (isto é, PAMP, DS e SS) conforme pode ser visto no trabalho [89]. Entretanto, neste trabalho, existe apenas um tipo de sinal por categoria e, para cada tipo de ataque é criada uma matriz de entrada e um vetor de saída. Portanto, foram embutidas várias matrizes de sinais para o processamento simultâneo. A inserção de várias matrizes de sinais foi fundamental para a criação de um IDS, haja vista que apenas a detecção de um tipo de ataque não condiz com a idéia de um sistema de detecção, mas apenas como mais uma ferramenta de detecção específica.

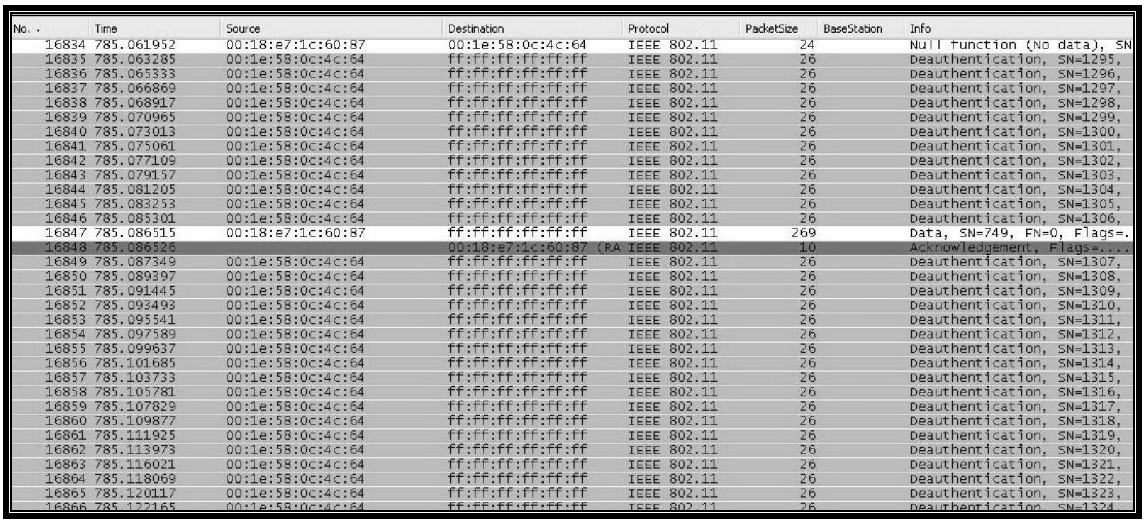
Os itens a seguir demonstram como foram definidos os sinais de entrada para cada tipo de ataque.

6.3.1 Sinais para Ataques de Negação de Serviço

Conforme visto no Capítulo 2, nas seções onde o problema de segurança de redes sem fio é abordado, quando uma estação da rede recebe um quadro "*de-authentication*" ela irá realmente cumprir com o que prevê esse tipo de quadro, irá se "desautenticar" e tentará novamente a autenticação. Por esse motivo, associado ao problema da não criptografia do cabeçalho dos quadros de gerenciamento e controle do padrão [105], as redes sem fio são vulneráveis a ataques usando tais quadros.

Para a criação dos sinais foi necessária a observação do comportamento dos quadros na rede. Uma dos comportamentos percebido para ataque DoS é que os quadros "*de-authentication*" possuem o mesmo tamanho, 26 bytes conforme figura 6.2. Outro comportamento está na quantidade de quadros enviados e, sobretudo, a origem dos mesmos.

Por ser uma rede modo infra-estrutura, esses quadros passam pelo AP, ou então possuem o AP como própria origem do quadro. Existe variação neste tipo de ataque, podendo ser direcionada para uma estação apenas, ou em modo *broadcast*. O intuito nos dois casos é diferente, no primeiro caso, o atacante quer evitar que a estação específica não consiga usar os serviços da rede, enquanto que no segundo caso, o ataque é direcionado sobre todas as estações. Com relação à quantidade de quadros, para o caso de ataque direcionado, é possível verificar que há uma intermitência de quadros chegando à estação, ou seja, o atacante envia 1 ou 2 quadros para a estação a cada segundo ou em maior tempo. Dessa forma, é preciso observar que um quadro desse tipo sozinho num segundo qualquer não significa necessariamente um ataque, podendo representar uma falha na rede ou uma saída da rede sendo executada pela estação.



No.	Time	Source	Destination	Protocol	PacketSize	BaseStation	Info
16834	785.061952	00:18:e7:1c:60:87	00:1e:58:0c:4c:64	IEEE 802.11	24		Null function (No data), SN
16835	785.063285	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1295,
16836	785.065333	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1296,
16837	785.066869	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1297,
16838	785.068917	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1298,
16839	785.070965	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1299,
16840	785.073013	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1300,
16841	785.075061	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1301,
16842	785.077109	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1302,
16843	785.079157	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1303,
16844	785.081205	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1304,
16845	785.083253	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1305,
16846	785.085301	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1306,
16847	785.086515	00:18:e7:1c:60:87	ff:ff:ff:ff:ff:ff	IEEE 802.11	269		Data, SN=749, FN=0, Flags=.
16848	785.086526		00:18:e7:1c:60:87 (RA	IEEE 802.11	10		Acknowledgement, Flags=....
16849	785.087349	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1307,
16850	785.089397	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1308,
16851	785.091445	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1309,
16852	785.093493	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1310,
16853	785.095541	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1311,
16854	785.097589	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1312,
16855	785.099637	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1313,
16856	785.101685	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1314,
16857	785.103733	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1315,
16858	785.105781	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1316,
16859	785.107829	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1317,
16860	785.109877	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1318,
16861	785.111925	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1319,
16862	785.113973	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1320,
16863	785.116021	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1321,
16864	785.118069	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1322,
16865	785.120117	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1323,
16866	785.122165	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	26		Deauthentication, SN=1324,

Figura 6.2 – Amostra do comportamento do tráfego quando em ataque *de-authentication*. Quando em modo *broadcast*, a quantidade enviada por segundo é alta.

Portanto, para o sinal PAMP (representando o ataque em execução) foi definida a quantidade de ocorrência de quadros com tamanho maior que 10 e menor que 30. Dessa forma, os quadros "*de-authentication*" sempre serão reconhecidos. Mesmo que seja identificado o quadro, não significará que o mesmo é ataque, porém quanto mais quadros tiverem por segundo, maiores serão as chances de gerar um sinal de saída maturo.

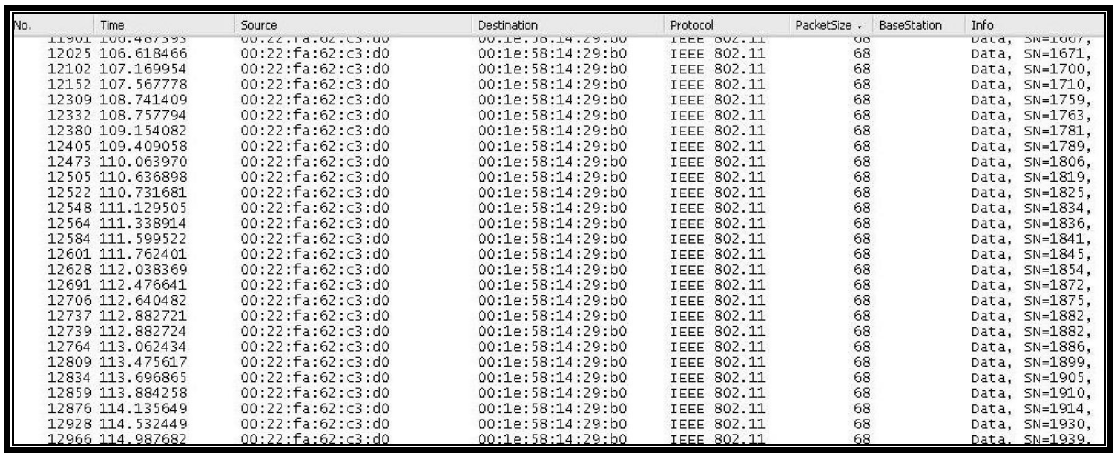
Para o sinal DS, que funciona como um contrapeso entre PAMP e SS, foi escolhido a taxa de quadros enviados e recebidos por segundo. Tal escolha pode ser explicada pela necessidade da estação se autenticar novamente, o que causa queda do tráfego de quadros por segundo.

Da mesma forma, quando existe alta taxa de "*de-authentication*", ocorre outro evento que modifica o comportamento do tráfego de quadros: a taxa de ocorrência de diferentes tipos de quadros aumenta praticamente para o máximo possível num único segundo. Essa taxa foi definida como sendo o sinal SS. Durante a execução dos experimentos foi observado que quando a taxa era maior que "0", mesmo a taxa sendo normalizada, o resultado estava comprometendo a eficiência do modelo. Após análise do problema, foi definido que a taxa maior que zero, receberia o valor "0" e a taxa igual a "0" receberia o valor máximo (100) para SS (isto é, "0" representa que não há segurança, enquanto 100 representa segurança).

6.3.2 Sinais para Ataque de Injeção de pacotes ARP

Os ataques de injeção de quadros contendo pacotes ARP geralmente usam as requisições do protocolo ARP como forma de gerar mais tráfego na rede com intuito de capturar mais informações e facilitar o processo de quebra da criptografia. Conforme já mencionado, os experimentos foram executados sobre a camada de enlace, por isso, não foi realizado análise de pacotes, mas sim de quadros, apesar disso, é possível identificar quando um quadro carrega um pacote ARP através do seu tamanho, que geralmente é padronizado com 68 bytes.

Como foram utilizados dois tipos de criptografia na rede: WEP e WPA. Para cada padrão há um valor diferente paro o quadro com pacote ARP. Portanto, ao verificar a rede usando o protocolo WEP havia um número grande de quadros do tipo "*data*" com o mesmo tamanho: 68 bytes. A figura 6.3 mostra os quadros capturados sendo possível verificar a quantidade de quadros do mesmo tamanho por segundo.



No.	Time	Source	Destination	Protocol	PacketSize	BaseStation	Info
11901	106.487393	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1667,
12025	106.618466	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1671,
12102	107.169954	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1700,
12152	107.567778	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1710,
12309	108.741409	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1759,
12332	108.757794	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1763,
12380	109.154082	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1781,
12405	109.409058	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1789,
12473	110.063970	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1806,
12505	110.636898	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1819,
12522	110.731681	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1825,
12548	111.129505	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1834,
12564	111.338914	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1836,
12584	111.599522	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1841,
12601	111.762401	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1845,
12628	112.038369	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1854,
12691	112.476641	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1872,
12706	112.640482	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1875,
12737	112.882721	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1882,
12739	112.882724	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1882,
12764	113.062434	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1886,
12809	113.475617	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1899,
12834	113.696865	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1905,
12859	113.884258	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1910,
12876	114.135649	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1914,
12928	114.532449	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1930,
12966	114.987682	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1939,

Figura 6.3 – Comportamento do ataque de injeção de pacotes com requisições ARP. Geralmente é possível observar que os quadros são repetidos de forma contínua durante um segundo.

Desta maneira, foi definido que o sinal PAMP é a taxa de quadros "*data*" de tamanho igual a 68 bytes por segundo originados da estação e o quadro de resposta geralmente possui 80 bytes de tamanho.

No caso do protocolo WPA, o valor é diferente, sendo encontrados, no caso da rede montada para os experimentos, quadros *data* com 130 bytes em grande quantidade. Portanto, PAMP também verifica quadros com esse tamanho, permitindo que o modelo seja aplicado sobre WPA mesmo que esse tipo de ataque sobre WPA não seja eficiente.

Durante esse tipo de ataque, o atacante usa a falsificação do MAC de alguma estação da rede, mesmo que haja filtragem de MAC, ainda assim é possível fazer o ataque caso seja realizado com sucesso o ataque da falsa autenticação (por exemplo, mesmo que haja filtragem de MAC, quando a estação original é derrubada e impedida de se autenticar, é possível fazer autenticação falsa usando o MAC original da estação).

Como a origem nesse caso é sempre a estação e não o AP, o sinal DS foi definido como sendo a taxa de quadros do tipo "*data*" enviados por segundo. Nesse caso, é possível o ataque direcionado ou em *broadcast*. Quando a rede tem segurança fraca, o atacante poderá usar a injeção em modo *broadcast*. Mas quando a rede é bem protegida, o atacante precisa ser cauteloso e, por isso, direciona o ataque a alguma estação.

Quando a estação ou o AP recebem um quadro, há a necessidade do envio de um quadro de reconhecimento (ACK). Esse quadro nada mais é do que a confirmação de que o quadro foi recebido pelo destino. Durante a análise dos sinais, foi detectado que, durante um ataque de injeção de pacotes ARP, o quadro de dados que é gerado não recebe resposta, ou seja, não há reconhecimento desses quadros, principalmente os que são transmitidos em modo *broadcast.* Sendo assim, o sinal SS recebeu como definição a relação de quadros do tipo "*data*" enviados e a taxa de quadros do tipo "*acknowledgement*" recebidos por segundo. Porém, existe um problema na linha do tempo para os quadros de reconhecimento, eles podem não chegar durante o mesmo segundo a que foram transmitidos, isso poderia gerar problemas para SS. Para evitar esse problema, foi dada uma porcentagem de 10 por cento de tolerância na relação entre os quadros de dados enviados e os quadros de reconhecimento recebidos no segundo.

Se a taxa ultrapassar a margem de tolerância, SS recebe o valor "0", senão recebe 100. Isso permitirá quebrar o peso de DS e PAMP.

6.3.3 Sinais para Ataques ChopChop

Um comportamento típico do ataque Chopchop é o envio de quadros do tipo *data* com destino desconhecido e tamanho variado, geralmente pequenos, numa faixa grande de endereços MAC. Portanto, para PAMP, foi escolhida a taxa de pacotes com destino diferente das conhecidas na rede, no mesmo segundo.

Conforme é possível ver na figura 6.4, a taxa de quadros com o mesmo tamanho é alta. Portanto, para SS foi determinado que o limiar de quadros, de mesmo tamanho, transmitido por segundo, está entre 31 e 150 bytes. Pode haver pequenos quadros "*data*" representando quadros legítimos, porém, não são repetidos os mesmos tamanhos de forma contínua como quando em ataque. PAMP e SS nesse caso são díspares e, quando um quadro é legitimo, porém pequeno, ainda assim, terá o endereço de alguma estação da rede, não sendo desconhecido. Portanto, não entra na taxa de quadros com destino conhecidos.

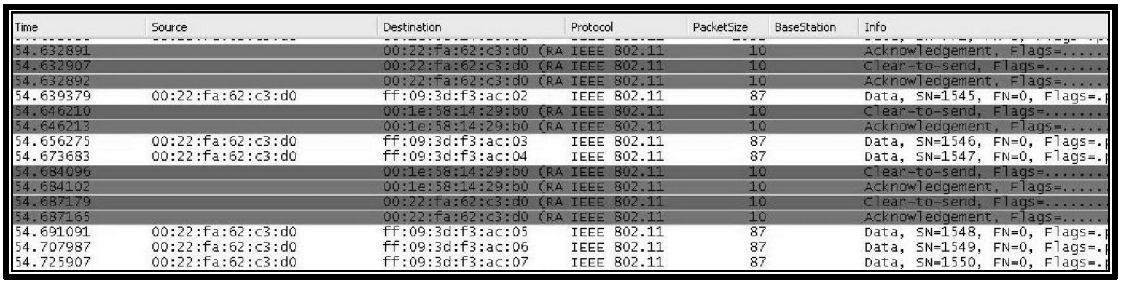


Figura 6.4 – Amostra do ataque ChopChop em execução. Os destinos, nesse caso, são desconhecidos e não repetidos.

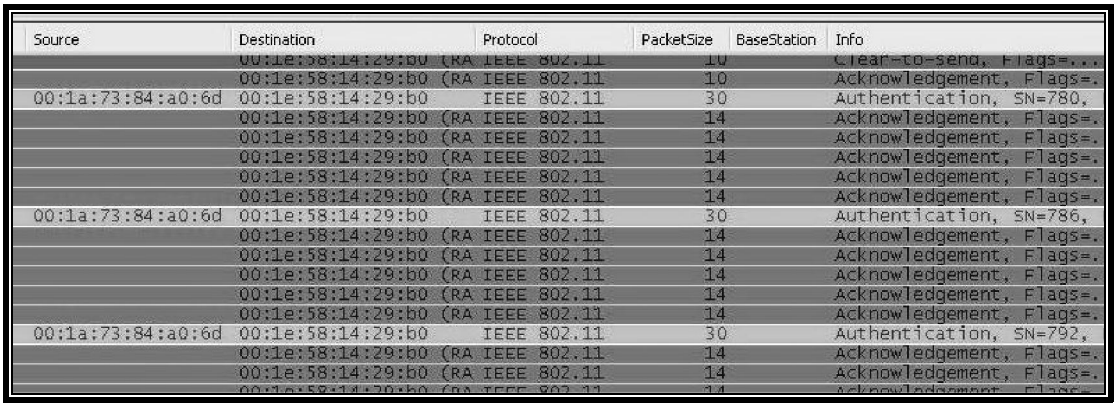
No caso de DS, o valor refere-se à taxa de quadros enviados por segundo. Como o ataque envia continuamente quadros para aumentar o tráfego na rede, geralmente o valor de DS irá subir. Quanto mais alto o valor de DS, mais perigo.

6.3.4 Sinais para Ataque de Falsa Autenticação

O intuito desse ataque é a associação ao AP da rede que servirá de vítima. A maioria dos ataques descritos nesse trabalho somente terá sucesso caso a estação esteja associada ao AP para que o mesmo responda os quadros recebidos. Portanto, geralmente o primeiro sinal de um ataque em eminência pode ser um ataque de autenticação falsa.

A diferença da análise deste tipo de ataque para os outros é que, a estação precisará monitorar o AP e a si mesma para que possa encontrar ataques. Essa solução não aparenta ser a mais adequada, visto que, para ser possível o monitoramento de todos os quadros que trafegam na rede, a placa de rede da estação precisa estar configurada em modo monitor, o que não permitirá o uso da mesma para envio e recebimento de quadros. Uma solução para este problema é a inserção de mais uma placa de rede por estação. Mas, é preciso salientar que isso pode ser oneroso e gerar problemas de escalabilidade. Mesmo assim, é importante que as estações verifiquem as requisições de autenticação das estações ao AP.

O comportamento desse tipo de ataque pode ser visto na figura 6.5. Quando um ataque está ocorrendo sem que o atacante tenha sucesso imediato, vários quadros de autenticação são enviados durante um segundo. O tamanho do quadro não muda, permanece com 30 bytes o tempo todo. Outro comportamento que emerge é a resposta ACK apresentada pelo AP vítima. Nesse caso, há uma modificação em relação ao tamanho do quadro se comparado aos quadros ACK normais: durante o ataque, eles apresentam 14 bytes enquanto que os normais, 10 bytes. Na figura 6.5, também é possível ver o tamanho variado dos quadros ACK recebidos.



Source	Destination	Protocol	PacketSize	BaseStation	Info
	00:1e:58:14:29:b0 (RA	IEEE 802.11	10		Clear-to-send, Flags=...
	00:1e:58:14:29:b0 (RA	IEEE 802.11	10		Acknowledgement, Flags=.
00:1a:73:84:a0:6d	00:1e:58:14:29:b0	IEEE 802.11	30		Authentication, SN=780,
	00:1e:58:14:29:b0 (RA	IEEE 802.11	14		Acknowledgement, Flags=.
	00:1e:58:14:29:b0 (RA	IEEE 802.11	14		Acknowledgement, Flags=.
	00:1e:58:14:29:b0 (RA	IEEE 802.11	14		Acknowledgement, Flags=.
	00:1e:58:14:29:b0 (RA	IEEE 802.11	14		Acknowledgement, Flags=.
	00:1e:58:14:29:b0 (RA	IEEE 802.11	14		Acknowledgement, Flags=.
00:1a:73:84:a0:6d	00:1e:58:14:29:b0	IEEE 802.11	30		Authentication, SN=786,
	00:1e:58:14:29:b0 (RA	IEEE 802.11	14		Acknowledgement, Flags=.
	00:1e:58:14:29:b0 (RA	IEEE 802.11	14		Acknowledgement, Flags=.
	00:1e:58:14:29:b0 (RA	IEEE 802.11	14		Acknowledgement, Flags=.
	00:1e:58:14:29:b0 (RA	IEEE 802.11	14		Acknowledgement, Flags=.
	00:1e:58:14:29:b0 (RA	IEEE 802.11	14		Acknowledgement, Flags=.
00:1a:73:84:a0:6d	00:1e:58:14:29:b0	IEEE 802.11	30		Authentication, SN=792,
	00:1e:58:14:29:b0 (RA	IEEE 802.11	14		Acknowledgement, Flags=.
	00:1e:58:14:29:b0 (RA	IEEE 802.11	14		Acknowledgement, Flags=.

Figura 6.5 – Amostra dos quadros de autenticação capturados durante ataque de falsa autenticação. Um comportamento observável é o tamanho de bytes para quadros ACK, diferente dos quadros normais que possuem tamanho igual a 10 bytes.

Outra característica encontrada no comportamento do ataque refere-se ao intervalo de tempo em que são enviados os quadros para o AP vítima. Às vezes, o ataque ocorre dentro de um intervalo de 10 segundos, ou seja, dentro de 1 a 10 podem ocorrer no mínimo 1 ataque. Por isso, alguns cuidados foram necessários para a formação dos sinais.

Em primeiro lugar, o PAMP foi definido como sendo a taxa de freqüência de quadros com tamanho 30 por segundo, ou então, em intervalos entre 1 e 10 segundos. Dessa forma,

quando o ataque não é contínuo, o sistema irá identificar que houve um evento anormal durante um segundo, ficará em observação num intervalo de 10 segundos. Caso outro quadro seja recebido durante esse período, está configurada uma tentativa de ataque. Quando em ataque contínuo, o sistema automaticamente considera os valores para a matriz de sinais.

No caso de SS, é verificada a taxa de ACK recebidos do AP com tamanho alterado. Mas, isso não é suficiente para detectar um ataque bem sucedido. Para isso, foi incluída a verificação de quadros de resposta de associação. Caso tenha a presença de um quadro, é fator de risco, e como tal, SS recebe peso “2”, ou seja, funciona como um processo de inflamação no IS. Mas, é preciso evitar o falso positivo quando uma estação verdadeira se autentica e a resposta está no modo como é feita a análise para esse ataque. Todas as estações conhecem o seu AP e as estações da rede, portanto, elas fazem o monitoramento do AP confrontando os MACs. Isso evita a possibilidade de um ataque de autenticação por MACs estranhos. Mas ainda não resolve o problema de autenticação verdadeira.

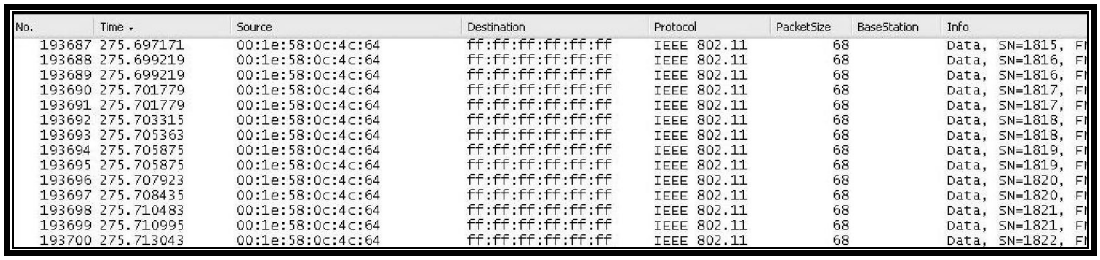
A resolução está no agente subalterno que será criado com a identificação errônea por parte do agente intermediário. Nesse caso, o agente subalterno responsável pela resposta ao possível ataque irá verificar se o sistema irá registrar o container para aquela estação (isto é, essa é uma rotina em que uma estação, ao se autenticar na rede, irá iniciar o módulo de estação do WIDS, como não terá o módulo, após um curto espaço de tempo, o agente irá comunicar ao agente superior que houve um ataque bem sucedido durante um determinado instante no tempo).

Para DS, é calculada a taxa de quadros destinados ao AP com origem desconhecida da rede. Portanto, para ataques sem *MACSpoofing*, SS funciona perfeitamente, mas, do contrário, SS apresentará valores falsos. Dessa forma, o sistema identificará como um ataque, mas, após um tempo, caso a estação conecte no sistema, será cancelado o alerta.

6.3.5 Sinais para Ataque Café-Latte

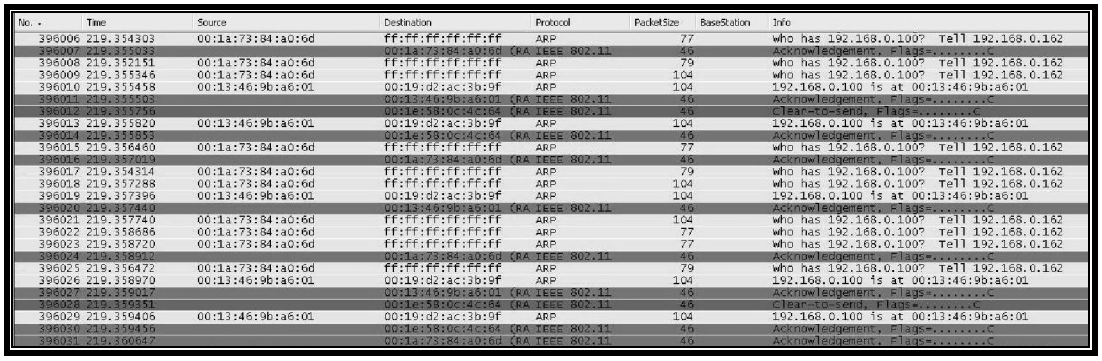
O principal efeito observado desse ataque é o envio de quadros de dados com tamanho 68 bytes em grande quantidade por segundo para ataques realizados diretamente sobre o AP conforme pode ser visto na figura 6.6, onde a origem (source) é o próprio AP. Para o caso de ataque direcionado usando uma estação associada, os quadros podem variar de tamanho, e dessa forma, é preciso observar que mesmo com tamanho diferente do padrão (isto é, geralmente dois valores são comuns para requisições ARP em redes com WEP habilitado, 68

e 80 bytes), trata-se de um quadro no qual possui como dados uma requisição ARP, conforme apresentado na figura 6.7.



No.	Time	Source	Destination	Protocol	PacketSize	BaseStation	Info
193687	275.697171	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	68		Data, SN=1815, Fl
193688	275.699219	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	68		Data, SN=1816, Fl
193689	275.699219	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	68		Data, SN=1816, Fl
193690	275.701779	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	68		Data, SN=1817, Fl
193691	275.701779	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	68		Data, SN=1817, Fl
193692	275.703315	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	68		Data, SN=1818, Fl
193693	275.705363	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	68		Data, SN=1818, Fl
193694	275.705875	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	68		Data, SN=1819, Fl
193695	275.705875	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	68		Data, SN=1819, Fl
193696	275.707923	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	68		Data, SN=1820, Fl
193697	275.708435	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	68		Data, SN=1820, Fl
193698	275.710483	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	68		Data, SN=1821, Fl
193699	275.710995	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	68		Data, SN=1821, Fl
193700	275.713043	00:1e:58:0c:4c:64	ff:ff:ff:ff:ff:ff	IEEE 802.11	68		Data, SN=1822, Fl

Figura 6.6 – Ilustração do ataque Café-Latte em execução tendo o AP como alvo. Todos os quadros são do mesmo tamanho, 68 bytes, característica de quadros com pacotes ARP embutidos.



No.	Time	Source	Destination	Protocol	PacketSize	BaseStation	Info
396006	219.354303	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	ARP	77		who has 192.168.0.100? Tell 192.168.0.162
396007	219.355033		00:1a:73:84:a0:6d (RA	IEEE 802.11	46		Acknowledgement, Flags=.......C
396008	219.352151	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	ARP	79		who has 192.168.0.100? Tell 192.168.0.162
396009	219.355346	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	ARP	104		who has 192.168.0.100? Tell 192.168.0.162
396010	219.355458	00:13:46:9b:a6:01	00:19:d2:ac:3b:9f	ARP	104		192.168.0.100 is at 00:13:46:9b:a6:01
396011	219.355503		00:13:46:9b:a6:01 (RA	IEEE 802.11	46		Acknowledgement, Flags=.......C
396012	219.355756		00:1e:58:0c:4c:64 (RA	IEEE 802.11	46		Clear-to-send, Flags=.......C
396013	219.355820	00:13:46:9b:a6:01	00:19:d2:ac:3b:9f	ARP	104		192.168.0.100 is at 00:13:46:9b:a6:01
396014	219.355853		00:1e:58:0c:4c:64 (RA	IEEE 802.11	46		Acknowledgement, Flags=.......C
396015	219.356460	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	ARP	77		who has 192.168.0.100? Tell 192.168.0.162
396016	219.357049		00:1a:73:84:a0:6d (RA	IEEE 802.11	46		Acknowledgement, Flags=.......C
396017	219.354814	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	ARP	79		who has 192.168.0.100? Tell 192.168.0.162
396018	219.357288	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	ARP	104		who has 192.168.0.100? Tell 192.168.0.162
396019	219.357396	00:13:46:9b:a6:01	00:19:d2:ac:3b:9f	ARP	104		192.168.0.100 is at 00:13:46:9b:a6:01
396020	219.357440		00:13:46:9b:a6:01 (RA	IEEE 802.11	46		Acknowledgement, Flags=.......C
396021	219.357740	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	ARP	104		who has 192.168.0.100? Tell 192.168.0.162
396022	219.358686	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	ARP	77		who has 192.168.0.100? Tell 192.168.0.162
396023	219.358720	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	ARP	77		who has 192.168.0.100? Tell 192.168.0.162
396024	219.358870		00:1a:73:84:a0:6d (RA	IEEE 802.11	46		Acknowledgement, Flags=.......C
396025	219.356472	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	ARP	79		who has 192.168.0.100? Tell 192.168.0.162
396026	219.358970	00:13:46:9b:a6:01	00:19:d2:ac:3b:9f	ARP	104		192.168.0.100 is at 00:13:46:9b:a6:01
396027	219.359017		00:13:46:9b:a6:01 (RA	IEEE 802.11	46		Acknowledgement, Flags=.......C
396028	219.359351		00:1e:58:0c:4c:64 (RA	IEEE 802.11	46		Clear-to-send, Flags=.......C
396029	219.359406	00:13:46:9b:a6:01	00:19:d2:ac:3b:9f	ARP	104		192.168.0.100 is at 00:13:46:9b:a6:01
396030	219.359456		00:1e:58:0c:4c:64 (RA	IEEE 802.11	46		Acknowledgement, Flags=.......C
396031	219.360642		00:1a:73:84:a0:6d (RA	IEEE 802.11	46		Acknowledgement, Flags=.......C

Figura 6.7 – Amostra dos quadros após decriptografia. Os pacotes ARP possuem tamanhos diferentes, porém, as requisições são destinadas a toda rede.

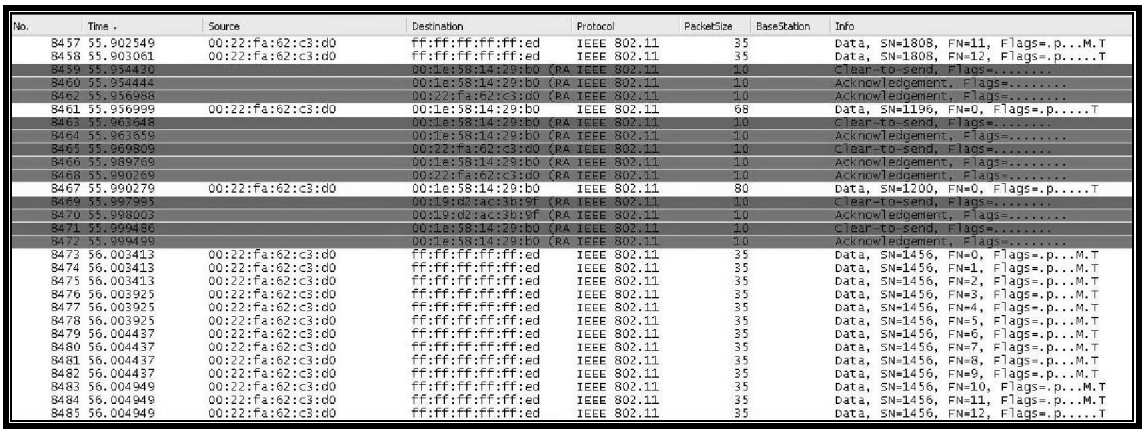
Portanto, para esse ataque, PAMP tem como valor a taxa de quadros recebidos em *broadcast* originados de endereços MAC desconhecidos ou da própria rede. Para SS, o valor é a taxa de quadros com tamanho inferior a 130 bytes e maior que 68 bytes repetidos. Para DS, o valor é a taxa de quadros recebidos por segundo, que, no caso, pode aumentar consideravelmente durante os ataques.

6.3.6 Sinais para ataque de fragmentação

Por fazer uso da possibilidade de fragmentar quadros, permitido no padrão 802.11, ao analisar o comportamento do tráfego da rede no momento do ataque, é possível observar a presença contínua de quadros com tamanho igual a 35 bytes, ilustrado pela Figura 6.8. O comportamento é parecido com aquele encontrado no ataque Café-Latte, porém, a técnica aplicada na fragmentação é diferente. Neste caso, o quadro é fragmentado com o mesmo

tamanho e enviado em modo *broadcast*, porém, o octeto final possui um valor “ed”, que facilita sua identificação.

Outra forma de identificar o ataque é a observação do valor de SN (*Sequency Number*). O valor é repetido “*n*” vezes, configurando um ataque.



No.	Time	Source	Destination	Protocol	PacketSize	BaseStation	Info
8457	55.902549	00:22:fa:62:c3:d0	ff:ff:ff:ff:ff:ed	IEEE 802.11	35		Data, SN=1808, FN=11, Flags=.p...M.T
8458	55.903061	00:22:fa:62:c3:d0	ff:ff:ff:ff:ff:ed	IEEE 802.11	35		Data, SN=1808, FN=12, Flags=.p.....T
8459	55.954430		00:1e:58:14:29:b0 (RA	IEEE 802.11	10		Clear-to-send, Flags=.......
8460	55.954444		00:1e:58:14:29:b0 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8462	55.956989		00:22:fa:62:c3:d0 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8461	55.956999	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	68		Data, SN=1196, FN=0, Flags=.p.....T
8463	55.963242		00:1e:58:14:29:b0 (RA	IEEE 802.11	10		Clear-to-send, Flags=.......
8464	55.963659		00:1e:58:14:29:b0 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8465	55.969809		00:22:fa:62:c3:d0 (RA	IEEE 802.11	10		Clear-to-send, Flags=.......
8466	55.969769		00:1e:58:14:29:b0 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8468	55.990269		00:22:fa:62:c3:d0 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8467	55.990279	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	80		Data, SN=1200, FN=0, Flags=.p.....T
8469	55.997995		00:19:d2:ac:3b:9f (RA	IEEE 802.11	10		Clear-to-send, Flags=.......
8470	55.998003		00:19:d2:ac:3b:9f (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8471	55.999486		00:1e:58:14:29:b0 (RA	IEEE 802.11	10		Clear-to-send, Flags=.......
8472	55.999499		00:1e:58:14:29:b0 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8473	56.003413	00:22:fa:62:c3:d0	ff:ff:ff:ff:ff:ed	IEEE 802.11	35		Data, SN=1456, FN=0, Flags=.p...M.T
8474	56.003413	00:22:fa:62:c3:d0	ff:ff:ff:ff:ff:ed	IEEE 802.11	35		Data, SN=1456, FN=1, Flags=.p...M.T
8475	56.003413	00:22:fa:62:c3:d0	ff:ff:ff:ff:ff:ed	IEEE 802.11	35		Data, SN=1456, FN=2, Flags=.p...M.T
8476	56.003923	00:22:fa:62:c3:d0	ff:ff:ff:ff:ff:ed	IEEE 802.11	35		Data, SN=1456, FN=3, Flags=.p...M.T
8477	56.003925	00:22:fa:62:c3:d0	ff:ff:ff:ff:ff:ed	IEEE 802.11	35		Data, SN=1456, FN=4, Flags=.p...M.T
8478	56.003925	00:22:fa:62:c3:d0	ff:ff:ff:ff:ff:ed	IEEE 802.11	35		Data, SN=1456, FN=5, Flags=.p...M.T
8479	56.004437	00:22:fa:62:c3:d0	ff:ff:ff:ff:ff:ed	IEEE 802.11	35		Data, SN=1456, FN=6, Flags=.p...M.T
8480	56.004437	00:22:fa:62:c3:d0	ff:ff:ff:ff:ff:ed	IEEE 802.11	35		Data, SN=1456, FN=7, Flags=.p...M.T
8481	56.004437	00:22:fa:62:c3:d0	ff:ff:ff:ff:ff:ed	IEEE 802.11	35		Data, SN=1456, FN=8, Flags=.p...M.T
8482	56.004437	00:22:fa:62:c3:d0	ff:ff:ff:ff:ff:ed	IEEE 802.11	35		Data, SN=1456, FN=9, Flags=.p...M.T
8483	56.004949	00:22:fa:62:c3:d0	ff:ff:ff:ff:ff:ed	IEEE 802.11	35		Data, SN=1456, FN=10, Flags=.p...M.T
8484	56.004949	00:22:fa:62:c3:d0	ff:ff:ff:ff:ff:ed	IEEE 802.11	35		Data, SN=1456, FN=11, Flags=.p...M.T
8485	56.004949	00:22:fa:62:c3:d0	ff:ff:ff:ff:ff:ed	IEEE 802.11	35		Data, SN=1456, FN=12, Flags=.p.....T

Figura 6.8 – Amostra dos quadros de 35 bytes e com o mesmo SN. Apesar de o comportamento ser parecido com o ataque Café-Latte, o método de fragmentação é diferente. Ainda é possível verificar a ocorrência de quadros com pacotes ARP embutidos (quadros com 68 e 80 bytes).

Dessa forma, PAMP foi definido como sendo a taxa de quadros recebidos ou enviados com tamanho igual a 35 bytes. SS recebeu a taxa de quadros com destino desconhecido ou *broadcast* enviados por segundo. Nesse caso, como o valor “0” representa totalmente seguro, foi alterado o valor para 100, ou seja, o valor de SS será preponderante no cálculo.

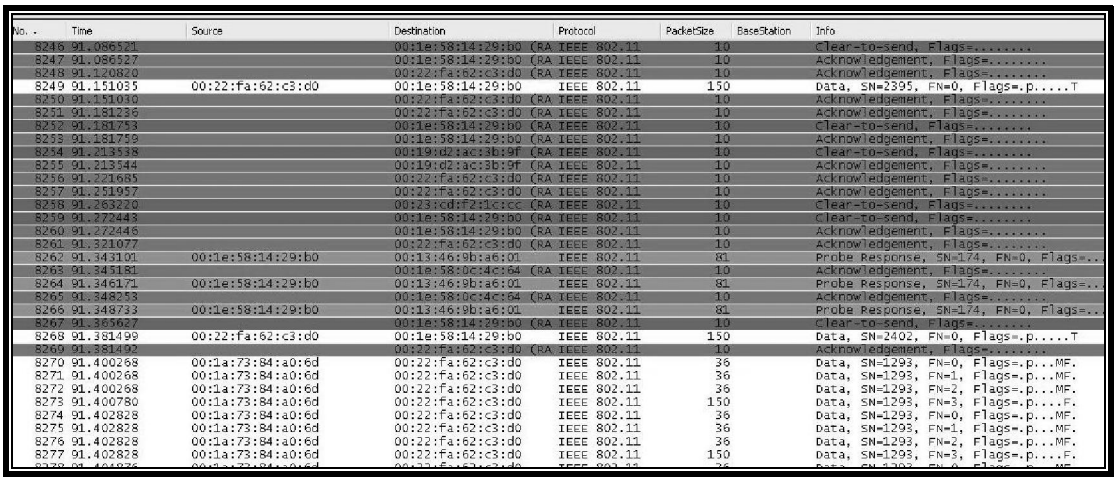
Como o problema é detectado na estação que está enviando o quadro, o valor de DS foi calculado através da quantidade de quadros enviados por segundo. Portanto, caso tenha muitos quadros em um segundo, o valor será alto, porém, quem definirá se houve ou não ataque é SS, pois 100 representam o máximo.

É importante salientar que para o sistema detectar esse tipo de ataque em tempo real é preciso que outra estação monitore o ambiente, ou então que a mesma estação tenha outra placa de rede para monitoramento.

6.3.7 Sinais para o ataque Hirte

Após observações do comportamento deste ataque, e verificar que o mesmo gera um quadro fragmentado com valor igual a 36 bytes em todos os testes, o valor de PAMP ficou configurado como sendo a taxa de quadros “*data*” na qual são recebidos ou enviados com tamanho 36 bytes. Por ser este um ataque estendido do ataque Cafe-Latte, possui o mesmo

problema de repetição de quadros com o mesmo SN. No caso de SS, o valor configurado está relacionado ao número de seqüência do quadro. Como pode ser visto na Figura 6.9, o número de seqüência SN do quadro é repetido várias vezes por segundo, enfatizando o comportamento anômalo.



No.	Time	Source	Destination	Protocol	PacketSize	BaseStation	Info
8246	91.086521		00:1e:58:14:29:b0 (RA	IEEE 802.11	10		Clear-to-send, Flags=........
8247	91.086527		00:1e:58:14:29:b0 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8248	91.120820		00:22:fa:62:c3:d0 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8249	91.151035	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	150		Data, SN=2395, FN=0, Flags=.p.....T
8250	91.151038		00:22:fa:62:c3:d0 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8251	91.181236		00:22:fa:62:c3:d0 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8252	91.181753		00:1e:58:14:29:b0 (RA	IEEE 802.11	10		Clear-to-send, Flags=........
8253	91.181759		00:1e:58:14:29:b0 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8254	91.213538		00:19:d2:ac:3b:9f (RA	IEEE 802.11	10		Clear-to-send, Flags=........
8255	91.213544		00:19:d2:ac:3b:9f (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8256	91.221685		00:22:fa:62:c3:d0 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8257	91.251957		00:22:fa:62:c3:d0 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8258	91.263220		00:23:cd:f2:1c:cc (RA	IEEE 802.11	10		Clear-to-send, Flags=........
8259	91.272443		00:1e:58:14:29:b0 (RA	IEEE 802.11	10		Clear-to-send, Flags=........
8260	91.272446		00:1e:58:14:29:b0 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8261	91.321077		00:22:fa:62:c3:d0 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8262	91.343101	00:1e:58:14:29:b0	00:13:46:9b:a6:01	IEEE 802.11	81		Probe Response, SN=174, FN=0, Flags=...
8263	91.345181		00:1e:58:0c:4c:64 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8264	91.346171	00:1e:58:14:29:b0	00:13:46:9b:a6:01	IEEE 802.11	81		Probe Response, SN=174, FN=0, Flags=...
8265	91.348253		00:1e:58:0c:4c:64 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8266	91.348733	00:1e:58:14:29:b0	00:13:46:9b:a6:01	IEEE 802.11	81		Probe Response, SN=174, FN=0, Flags=...
8267	91.365627		00:1e:58:14:29:b0 (RA	IEEE 802.11	10		Clear-to-send, Flags=........
8268	91.381499	00:22:fa:62:c3:d0	00:1e:58:14:29:b0	IEEE 802.11	150		Data, SN=2402, FN=0, Flags=.p.....T
8269	91.381502		00:22:fa:62:c3:d0 (RA	IEEE 802.11	10		Acknowledgement, Flags=........
8270	91.400268	00:1a:73:84:a0:6d	00:22:fa:62:c3:d0	IEEE 802.11	36		Data, SN=1293, FN=0, Flags=.p...MF.
8271	91.400268	00:1a:73:84:a0:6d	00:22:fa:62:c3:d0	IEEE 802.11	36		Data, SN=1293, FN=1, Flags=.p...MF.
8272	91.400268	00:1a:73:84:a0:6d	00:22:fa:62:c3:d0	IEEE 802.11	36		Data, SN=1293, FN=2, Flags=.p...MF.
8273	91.400780	00:1a:73:84:a0:6d	00:22:fa:62:c3:d0	IEEE 802.11	150		Data, SN=1293, FN=3, Flags=.p....F.
8274	91.402828	00:1a:73:84:a0:6d	00:22:fa:62:c3:d0	IEEE 802.11	36		Data, SN=1293, FN=0, Flags=.p...MF.
8275	91.402828	00:1a:73:84:a0:6d	00:22:fa:62:c3:d0	IEEE 802.11	36		Data, SN=1293, FN=1, Flags=.p...MF.
8276	91.402828	00:1a:73:84:a0:6d	00:22:fa:62:c3:d0	IEEE 802.11	36		Data, SN=1293, FN=2, Flags=.p...MF.
8277	91.402828	00:1a:73:84:a0:6d	00:22:fa:62:c3:d0	IEEE 802.11	150		Data, SN=1293, FN=3, Flags=.p....F.

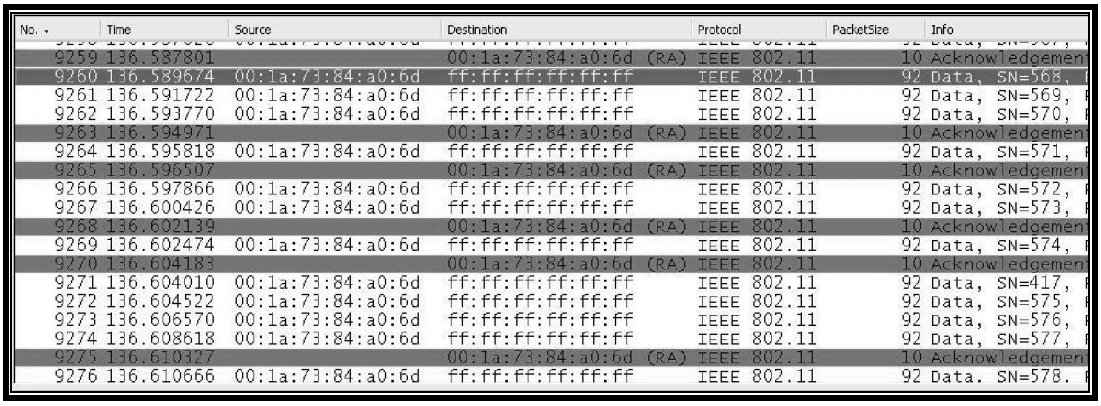
Figura 6.9– Amostra dos quadros "*data*" com tamanho 36 bytes e a mesma seqüência de número (SN) repetida várias vezes em um segundo.

O sinal DS, nesse caso, recebe a taxa de recebimento de quadros por segundo. Como o ataque gera muitos quadros fragmentados, o número de quadros recebidos pela estação pode aumentar consideravelmente.

6.3.8 Sinais para Ataque Interativo

O ataque de quadros interativos, ou seja, a interatividade está no fato do usuário poder escolher qual quadro usar para desencadear o ataque.

Duas características emergem desse ataque: (i) envio de quadros com destino broadcast e (ii) origem desconhecida. A figura 6.10 apresenta essas duas características. Os quadros atingidos nesse caso são os quadros data, porém é possível que o atacante use outros quadros (por isso é interativo). Outra característica que emerge é o tamanho dos quadros, geralmente menores que 100 bytes e com repetição contínua.



No.	Time	Source	Destination	Protocol	PacketSize	Info
9259	136.587801		00:1a:73:84:a0:6d (RA)	IEEE 802.11	10	Acknowledgemen
9260	136.589674	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	IEEE 802.11	92	Data, SN=568,
9261	136.591722	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	IEEE 802.11	92	Data, SN=569,
9262	136.593770	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	IEEE 802.11	92	Data, SN=570,
9263	136.594971		00:1a:73:84:a0:6d (RA)	IEEE 802.11	10	Acknowledgemen
9264	136.595818	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	IEEE 802.11	92	Data, SN=571,
9265	136.596507		00:1a:73:84:a0:6d (RA)	IEEE 802.11	10	Acknowledgemen
9266	136.597866	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	IEEE 802.11	92	Data, SN=572,
9267	136.600426	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	IEEE 802.11	92	Data, SN=573,
9268	136.602139		00:1a:73:84:a0:6d (RA)	IEEE 802.11	10	Acknowledgemen
9269	136.602474	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	IEEE 802.11	92	Data, SN=574,
9270	136.604183		00:1a:73:84:a0:6d (RA)	IEEE 802.11	10	Acknowledgemen
9271	136.604010	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	IEEE 802.11	92	Data, SN=417,
9272	136.604522	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	IEEE 802.11	92	Data, SN=575,
9273	136.606570	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	IEEE 802.11	92	Data, SN=576,
9274	136.608618	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	IEEE 802.11	92	Data, SN=577,
9275	136.610327		00:1a:73:84:a0:6d (RA)	IEEE 802.11	10	Acknowledgemen
9276	136.610666	00:1a:73:84:a0:6d	ff:ff:ff:ff:ff:ff	IEEE 802.11	92	Data. SN=578.

Figura 6.10 – Os quadros data com 92 bytes representam o ataque proveniente de uma estação desconhecida da rede.

De acordo com as observações e o comportamento encontrado durante o ataque, para o sinal PAMP foi escolhida a taxa de freqüência de quadros com destino desconhecido da rede. Para SS foi definida a taxa de quadros *data* repetidos com tamanho entre 40 e 100 bytes. Para evitar problemas entre PAMP e SS (isto é, quando ocorre o ataque, tanto PAMP quanto SS podem ter valores semelhantes e isso poderá dificultar nos cálculos). A solução adotada foi: quando o valor de SS ultrapassar um limite de 10 quadros (pode ser mais ou menos, sendo que é preciso apenas verificar o comportamento durante os ataques que geralmente demonstram ter alta quantidade de quadros por segundo), o valor de SS receberá “0”. Dessa forma, é dada uma taxa de tolerância para os quadros que por ventura estiverem dentro das características avaliadas, porém, com origem verdadeira.

6.3.9 Antígenos

Os antígenos, assim como os sinais, são de grande importância para o algoritmo de detecção. Para este trabalho, os antígenos são uma representação dos quadros recebidos, por isso, cada antígeno representa um quadro. Cada antígeno é um objeto (isto é, na visão sistêmica) e sua estrutura contempla todos os campos do cabeçalho do quadro mais o tamanho do quadro (inclusive com a carga útil). A carga útil não é utilizada neste trabalho, por não haver usabilidade no modelo.

Os atributos do antígeno podem ser vistos na figura 6.11. Um dos atributos refere-se exclusivamente à estrutura interna do antígeno. Nessa estrutura, estão dados coletados nos quadros e para agilizar os processos e diminuir o tamanho do mesmo, são iniciados com valores entre “0” e 1 (“0” representa falso e 1 verdadeiro).

O contexto somente será iniciado após o processamento do antígeno pelo DCA e determinará se o mesmo deverá ser separado ou não para análise dos agentes subalternos ou do agente intermediário. Caso sejam iniciados com contexto maturo, os antígenos serão expostos aos agentes subalternos que farão comparação entre a estrutura do antígeno com sua estrutura interna. Assim como serão expostos ao classificador *bayesiano* caso não sejam reconhecidos por nenhum agente subalterno.

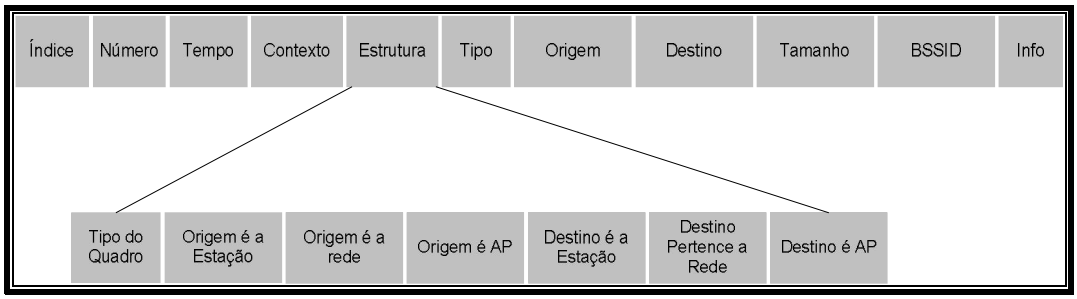


Figura 6.11 – Estrutura do antígeno antes de passar pelo processamento do DCA.

6.3.10 Parâmetros para DCA

Alguns parâmetros foram definidos de acordo com as características do projeto. Os valores podem ser vistos na tabela 6.2.

Tabela 6.2 – Valores dos parâmetros de entrada do algoritmo de detecção.

Nome	Valor Atribuído
Número de sinais por categoria	24
Número de categorias da matriz de sinais	1
Número de ciclos da célula	1
Numero de DCs na população (por segundo)	1
Número de antígenos no vetor principal	~ a quantidade da base de dados capturada
Tamanho do vetor de antígenos da DC	~ a quantidade referente ao segundo em questão
Número de sinais de saída por DC	24
Número de antígenos mostrados a DC por ciclo	~ por segundo
Número máximo de antígenos coletados por DC	~ ao total de antígenos por segundo
Valor do limiar de migração	Entre 0 e 2

Nos trabalhos encontrados na literatura acadêmica (já citados no decorrer do trabalho), foram utilizados valores fixos para estes parâmetros. Pela configuração utilizada neste trabalho, os valores não poderiam ser fixos, haja vista que os sinais são coletados diretamente do tráfego da rede, ou seja, através dos quadros que trafegam pela rede. Portanto, para os vetores de entrada de antígenos, tanto o principal quanto aquele interno à DC, os valores são variáveis através do uso de vetores com tamanho variável (por exemplo, *ArrayList* do Java).

Um elemento fundamental no processamento dos sinais é o valor de migração da DC. Cada célula tem um valor peculiar e durante o processamento pode atingi-lo ou não, transformando-a em semi-matura ou matura. Portanto, foram aplicados alguns testes para verificar qual seria o melhor valor para atingir os resultados esperados. Para os testes foram escolhidos os valores 1, 2 e 3 como limites máximos (isto é, $0 < x <= 1$, $1 < x <= 2$ e $2 < x <= 3$). Após testes, verificou-se que, os valores entre 1 e 2, apresentaram equilíbrio entre a geração de falsos positivos e falsos negativos. No caso de valores acima de 2, as taxas de erros (falsos negativos) podem alcançar valores maiores que 50%, principalmente quando perto do valor máximo. Observou-se ainda que, para redes com tráfego pequeno, o número de falsos negativos aumenta consideravelmente, visto que os valores dos sinais estão ligados ao comportamento do tráfego na rede. Para o caso dos valores entre “0” e 1, os testes demonstraram que quanto mais próximos de 1, melhor são os resultados. Enquanto que, muito próximos de “0”, há um aumento considerável de falsos positivos, haja vista que, a maior parte das DCs sofrerá migração para este caso.

Portanto, os melhores resultados podem ser vistos como aqueles entre 1 e 2.

6.4 Plataforma de Agentes JADE

Algumas alterações foram necessárias para o framework JADE. A seguir, as mais importantes:

- **Classe jade.core.Agent:** criação do método *createAgent* que tem como função criar uma alternativa para criação de agentes durante execução do sistema;
- **Classe jase.core.Agent:** modificação no método *run()* para possibilitar o envio de mensagens para o agente superior após a morte de algum agente do sistema;
- **Classe jade.core.Agent:** criação do atributo *behaviourGuide*, utilizado para informar qual tarefa o agente deverá executar;

- **Classe jade.core.AgentContainerImpl:** modificação no método *startBootstrapAgents()*. Esta é uma rotina que, juntamente com um aplicativo gerenciador de nomes de agentes rodando no servidor do sistema, nomeia os agentes de forma que não haja agentes com o mesmo nome na plataforma, quando o agente é criado via comando no *prompt* do sistema;
- **Classe jade.wrapper.ContainerController:** criação do método *createNewAgent(String nickname, String className, Object[] args)* para gerenciar os nomes dos agentes quando criados em tempo de execução;

6.5 Validação do Classificador Bayesiano

Para validar o classificador de antígenos, foi utilizada a técnica de validação cruzada (*cross-validation*) *leave-one-out* [36]. Este tipo de validação envolve o uso de apenas uma única observação (isto é, uma linha, por exemplo) a partir da amostra original, ficando o restante como dados para treinamento. Este passo é repetido até que todas as observações tenham sido usadas. O uso dessa técnica foi inspirado pela baixa quantidade de observações presentes na base de dados (158 observações), haja vista que, esta técnica não é indicada para bases grandes, pois o custo é muito alto pelo grande esforço necessário para o processo de treinamento de todas as observações. Essa técnica é parecida com a técnica *K-fold*, porém, *K* é igual à quantidade de observações na amostra original.

A figura 6.12 apresenta uma tela da ferramenta mostrando o resultado do teste de validação do classificador, no caso, igual a 83,6%. Considerando que o objetivo do sistema é prover segurança, o valor alcançado não pode ser considerado muito favorável. Porém, foi satisfatório para os resultados da identificação dos antígenos por parte do agente intermediário, conforme pode ser visto nas seções posteriores.

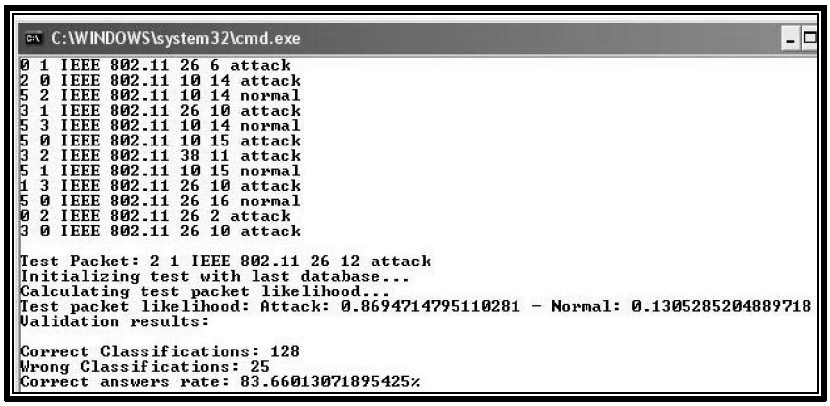


Figura 6.12 – Resultado obtido após teste de validação usando a técnica *leave-one-out*.

6.6 Resultados

Esta seção apresenta os resultados obtidos com os experimentos. Nela são esclarecidas algumas informações sobre os mesmos durante sua realização, além da explicação dos resultados.

O objetivo dos experimentos aqui apresentados é testar o comportamento do sistema como um todo analisando (i) os agentes em execução, (ii) a eficiência do algoritmo de detecção, (iii) a eficiência na classificação dos antígenos (isto é, quadros) coletados com anormalidades, (iv) a automatização do sistema e (v) o reflexo de cada ataque sobre a rede em relação aos outros tipos de sinais desenvolvidos.

As figuras apresentadas contendo dois gráficos representam respectivamente (da esquerda para direita): (a) os valores coletados para entrada do algoritmo DCA e (b) os valores alcançados após processamento do algoritmo. Os valores estão normalizados numa escala de “0” a 100 (nem todos os casos foi necessária aplicar a normalização), pois, após vários testes foi observado que, para alguns sinais, os sinais SS têm valores baixos se comparado com DS e por isso, mesmo que SS seja maior que PAMP, em muitos casos, houve falsos alarmes. Portanto, em alguns casos, quando SS recebe valor “0” (isto é, significa seguro) foi passado para 100 (representa 100 por cento seguro) enquanto que, caso seja diferente de “0” (representa insegurança) SS recebeu “0”. Em outros casos, PAMP também recebeu valor 100 ou “0”. Para o processo de normalização, é realizado, antes da execução, o mapeamento com a base de dados para coleta dos valores para o experimento. O processo de normalização seguiu a equação Eq. 6.1.

$$V = \frac{V - V_{\min}}{V_{\max} - V_{\min}} * 100, \qquad \text{Eq. 6.1}$$

Foram instanciados dois módulos principais: (i) nas estações e (ii) no servidor. No caso “(i)”, as estações receberam o módulo contendo um agente básico e um *container* de agentes subalternos vazio. Ao iniciar o módulo da estação com *container* vazio, o sistema foi colocado à prova para averiguação da capacidade de detecção e identificação do problema de forma automatizada. No caso “(ii)”, o módulo servidor foi instanciado em uma máquina diretamente conectada ao AP via cabo. Tal configuração é necessária para evitar que um ataque torne indisponível o servidor, pois, caso isso aconteça, as estações não poderão

consultar o agente intermediário para eventos anormais nos quais não são conhecidos pela mesma.

As figuras contendo apenas um gráfico representam a classificação dos antígenos durante os experimentos. No eixo vertical esquerdo o valor “0” representa que o mesmo não foi classificado como ataque, 1 representa exatamente o contrário. No eixo vertical direito, os valores representam o tipo de quadro (que também representa o ataque) representado pelo antígeno classificado. No eixo horizontal, são representados os segundos em que houve a ocorrência.

Assim como as figuras, são apresentadas tabelas para informações adicionais dos experimentos, tais como: (a) falsos alarmes, (b) taxa de erros, (c) tamanho das bases de dados, (d) quantidade de antígenos criados, (e) segundos de ataque, (f) tempo de resposta, (g) troca de mensagens, (h) taxa do custo da troca de mensagens pela rede, entre outras informações cabíveis.

Os itens contidos nas tabelas foram escolhidos por representar índices nos quais é possível aferir a eficiência do algoritmo. Dessa forma, um dos fatores fundamentais para comprovar o uso do AIS, como base do projeto, está no tempo de resposta para a detecção de um problema sem que haja agentes subalternos para a resposta (isto é, analogia entre o sistema inato e o sistema adaptativo). Nesse caso, é o tempo de saída do pedido de ajuda para o agente intermediário, a classificação do problema e o envio do agente subalterno para a estação. O agente subalterno criado após a classificação de um novo problema representa a memória do ataque ocorrido anteriormente e por isso, após o primeiro incidente, o tempo de resposta é menor, fato observado nos experimentos. Entretanto, caso um ataque sofra variação, é necessária uma nova classificação e atualização do agente.

Apesar da variação existente entre uma rede e outra (isto é, na configuração e no tráfego de dados), foi averiguado o tempo de resposta de acordo com a distância da estação para o AP. Para cada 10 metros de distância foi aferido o tempo de resposta, portanto, apesar do sinal da antena (AP) conseguir alcançar mais que 30 metros, acima deste valor, o sinal pode sofrer maior interferência e por isso, é desaconselhável. Dessa forma, os testes foram divididos em três: (i) $m < 10$, (ii) $m < 20$ e (ii) $m < 30$, onde m é a distância máxima da estação ao AP. Em “(i)”, os valores são ínfimos, cerca de meio segundo. Em “(ii)”, os sinais correspondem a menos de 2 segundos em média e para “(iii)”, o tempo gasto é de aproximadamente 3 segundos em média. Portanto, nas tabelas são apresentados os valores do tempo médio das requisições enviadas para o agente intermediário até o registro do agente subalterno na estação.

Para todos os experimentos, os resultados obtidos no tempo de ativação do agente subalterno da estação (isto é, já criado e replicado para todas as estações) são menores que 1 segundo, portanto, pode-se concluir que o tempo está dentro do necessário para execução em tempo real.

Por ser um sistema que atua sobre a rede, ou seja, necessita da rede para execução e, por isso, gera tráfego sobre a mesma, foi investigado ainda o custo da troca de mensagens entre as estações e o servidor. Para esse caso, após execução dos experimentos, foi detectado um valor médio de 100 Kbytes por mensagem. Portanto, para o cálculo da quantidade de quadros gerados na rede foi utilizada a equação Eq. 6.2,

$$T_{\text{quadros}} = \frac{(n * 100)}{1500}, \qquad \text{Eq. 6.2}$$

onde T_{quadros} e a taxa de quadros gerada na rede, 1500 é o tamanho máximo encontrado como padrão para o quadro de redes sem fio e n a quantidade de mensagens geradas durante o experimento.

Da mesma forma, para o cálculo da taxa de erros para o algoritmo de detecção, os valores seguem a equação Eq. 6.3,

$$T_{\text{erros}} = \frac{(n * 100)}{N}, \qquad \text{Eq. 6.3}$$

onde T_{erros} é a taxa de erros, n é a quantidade de erros encontrados no experimento e N é o valor de segundos nos quais houve ataque. A taxa de erros é dividida em falsos positivos e falsos negativos. Na tabela de informações gerais de cada experimento, é apresentada a soma dos valores dos dois casos.

Após análise geral dos resultados, foi constatado que, mesmo quando os valores de SS e PAMP são nulos e DS não o é, após o processamento o valor de maturo recebe valor 1 e semi-maturo “0”, fato que gerava muitos falsos positivos. Para solucionar esse problema, foi definido que, quando os valores fossem iguais a “0”, o valor de PAMP (maturo) definido pela tabela imunológica usada como entrada nos cálculos do algoritmo DCA receberia “0”. Dessa forma, não haveria a possibilidade de haver valor maturo maior que semi-maturo nas DCs.

Analisando as bases de dados, ficou comprovado que todos os valores quando nulos

geravam falsos positivos pela presença de algum valor em DS. Fica clara a necessidade de estudo sobre esse comportamento quando valores nulos são coletados.

Como são oito tipos de ataque, somente serão incluídas as figuras onde houve qualquer anormalidade com os sinais processados.

6.6.1 Experimento 1 – De-authentication e Injeção de Requisições ARP

Para este experimento, dois tipos de ataques são aplicados: (a) *de-authentication* e (b) injeção de requisições ARP. Foi utilizado o ambiente completo, ou seja, com cinco estações e um servidor. Portanto, para cada estação são apresentados os resultados obtidos através das figuras e tabelas.

As figuras entre 6.13 e 6.17 apresentam os resultados obtidos da estação 1. Na figura 6.13, exemplificando o ataque *de-authentication*, entre os segundos 308 até 391, houve ataque *broadcast*, e por isso, a quantidade de quadros enviados é muito alta. Dessa forma, os valores para PAMP alcançam o limite máximo, enquanto que, entre os segundos 740 e 1052, por ser um ataque do tipo direcionado à estação, a quantidade de quadros *de-authentication* é menor. Dessa forma, por sofrer normalização, os valores apresentam baixo tráfego. O gráfico mostra que, durante o ataque, é possível ver a potência variar para cada segundo. O gráfico da figura também mostra que, durante alguns segundos (entre 1452 até 1620), houve uma grande atividade de quadros, o que ele considerou como sinal de perigo. Neste ponto, ao observar a figura 6.14, é possível verificar que o algoritmo detectou uma parte desses sinais como ataque, a partir da análise dos sinais pertinentes ao ataque de injeção de ARP. É um ataque oriundo de uma estação da rede (quando o atacante não utiliza *MACSpoofing*) direcionado para todas as direções. O sinal SS é nulo nesta ocasião e o sinal PAMP torna-se sobressalente aos outros. A conseqüência desse comportamento é a maturidade das DCs.

É possível observar que, os valores do gráfico da direita, em cada figura, correspondem ao mesmo comportamento encontrado nos sinais de entrada, ou seja, após processado, o algoritmo DCA detectou corretamente os problemas.

As figuras 6.15, 6.16 e 6.17 mostram que o sistema encontrou alguns sinais PAMP para os ataques ChopChop, fragmentação e interativo. Porém, os valores não foram suficientes para gerar maturidade das DCs.

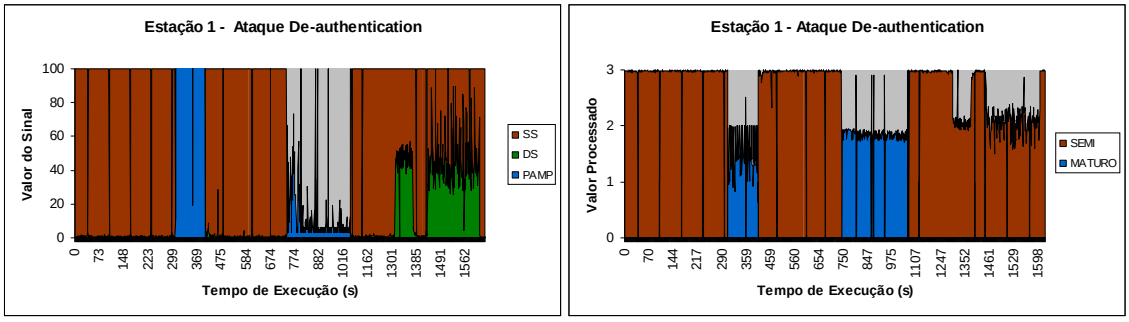


Figura 6.13 – Amostra dos sinais analisados para o ataque *de-authentication* (ou DoS). O gráfico da esquerda representa os valores de entrada, enquanto que o gráfico da direita, os valores processados.

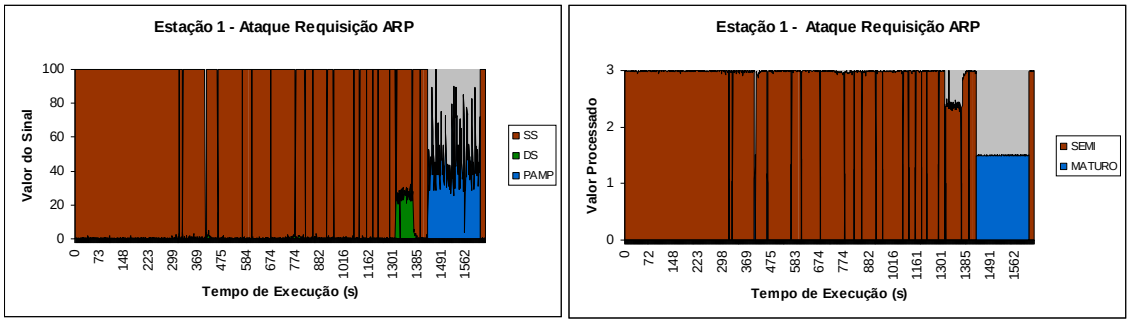


Figura 6.14 – Amostra dos sinais analisados para o ataque de injeção de pacotes contendo requisições ARP na estação 1.

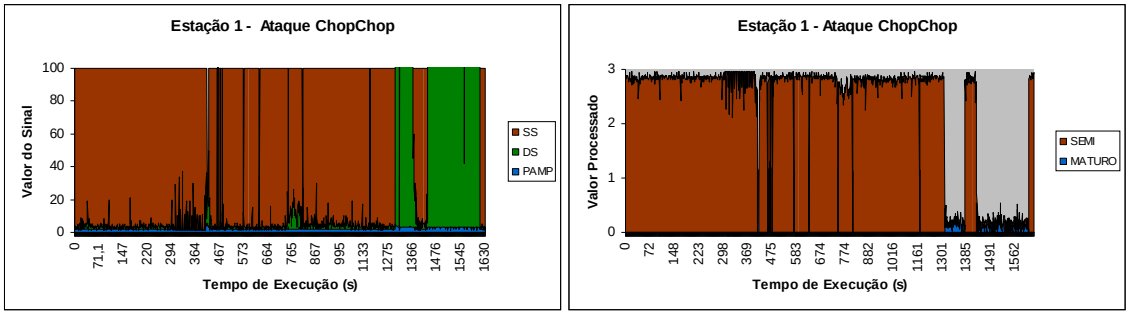


Figura 6.15 – Amostra dos sinais do ataque ChopChop. Nos segundos finais foi detectada ocorrência de alguns quadros com sinal PAMP, porém, não foram suficientes para gerar maturidade da célula.

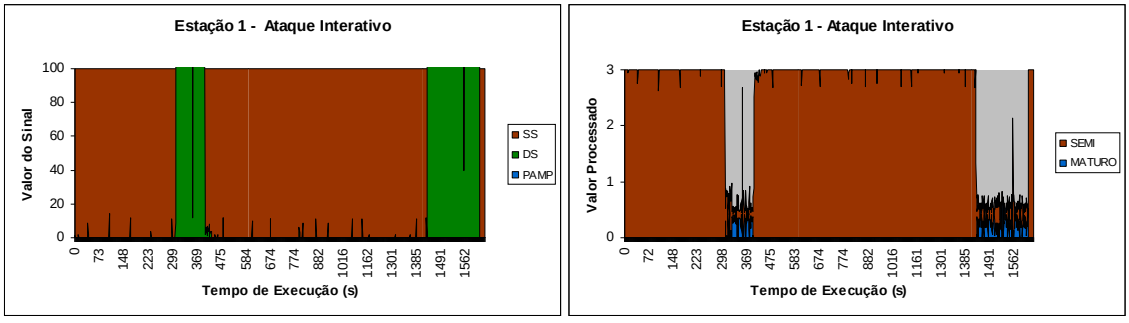


Figura 6.16 – Amostra dos sinais analisados para o ataque Interativo. Mesmo caso da figura 6.15.

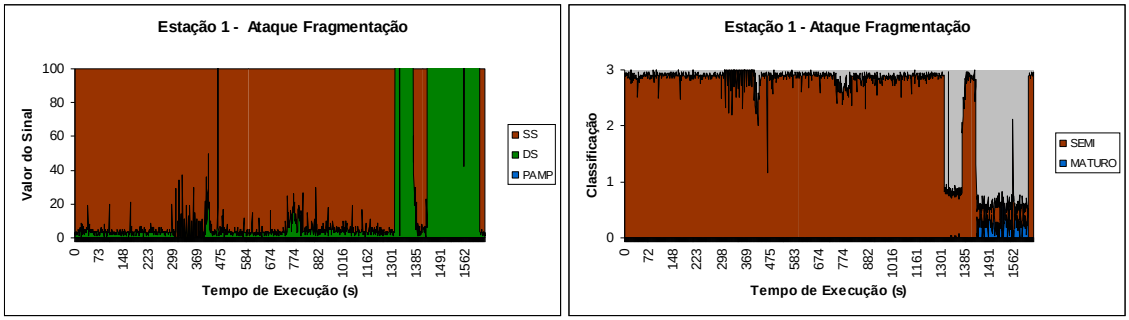


Figura 6.17 – Amostra dos sinais analisados para o ataque Fragmentação. Mesmo caso do ataque interativo.

A figura 6.18 representa os resultados alcançados durante a execução no servidor através do agente intermediário. Nela, é possível ver duas chamadas ao agente e a classificação apresentada aos antígenos passados. Através da classificação dos antígenos, 7 agentes subalternos foram criados, um para cada tipo de problema. Entretanto, não houve ataque de todos os tipos, mas, durante o ataque, alguns quadros sofrem alterações em seus comportamentos e, por isso, apresentam situações que podem significar ataque. Dessa forma, a criação dos agentes subalternos, nesse caso, gerou 2 falsos positivos para quadros de associação.

O gráfico da figura demonstra que houve uma chamada ao servidor abaixo do segundo 50, ao observar a base de dados, foi constatado que o sistema detectou anomalia no segundo 39 através dos quadros *probe response*, *probe request* e *null function*. Como a quantidade encontrada era demasiadamente grande, num único segundo, o sistema interpretou tal ocorrência como anômala e, por não conhecer tal problema, solicitou ajuda do agente intermediário enviando os antígenos para análise. Após submeter os antígenos ao classificador, o resultado foi ataque e por isso, foram criados os agentes subalternos correspondentes aqueles tipos de quadros. Ao analisar o comportamento desses quadros, durante o ataque, foi detectada a ocorrência de forma contínua, principalmente durante os segundos em ataque.

O fato importante neste caso é que o sistema não foi preparado para detectar eventos para esses tipos de quadros, porém, o modelo determinado para a geração dos sinais destes dois tipos de ataques (ARP e *de-authentication*) permitiu detectar anomalias na rede e o classificador acertadamente classificou como um evento proveniente de ataque.

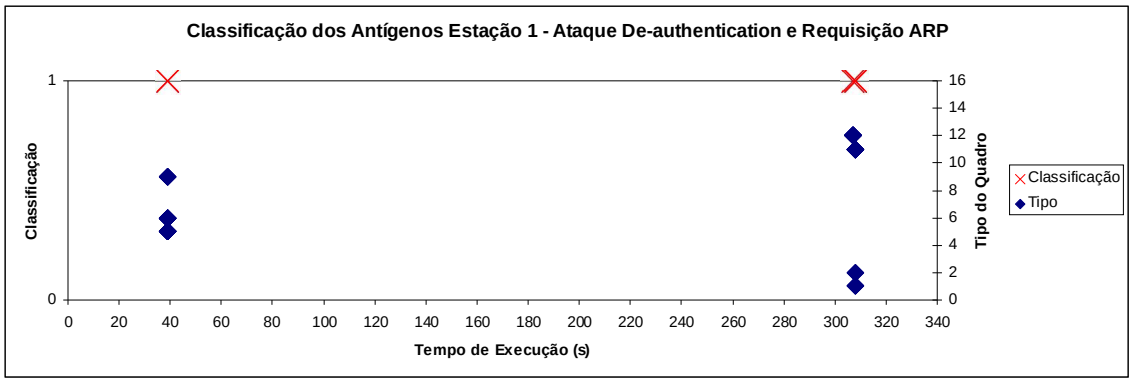


Figura 6.18 – Amostra da classificação dos antígenos pelo classificador Bayeseano para a estação 1. Há três chamadas para o *Aint*.

Nas figuras 6.19 até 6.22 são apresentados os resultados para a estação 2. Para essa estação, não há ocorrência de ataque ARP. Porém, para o ataque Cafe-Latte, representado pela figura 6.20, é possível ver que, no segundo 1422 o sistema detectou, erradamente, um evento anômalo.

A figura 6.19 mostra que houve ataque entre os segundos 307 e 391, proveniente do AP, ou seja, em modo *broadcast*. Nos segundos entre 600 e 700, há ocorrência de ataque direcionado.

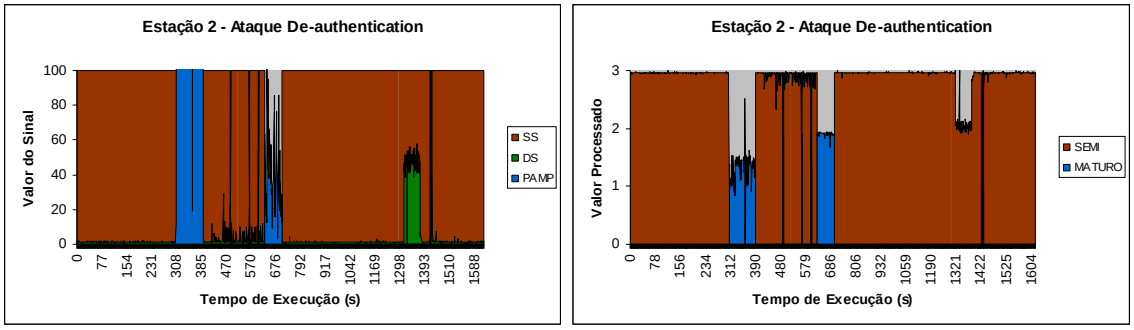


Figura 6.19 – Amostra dos sinais analisados para o ataque *de-authentication* para estação 2.

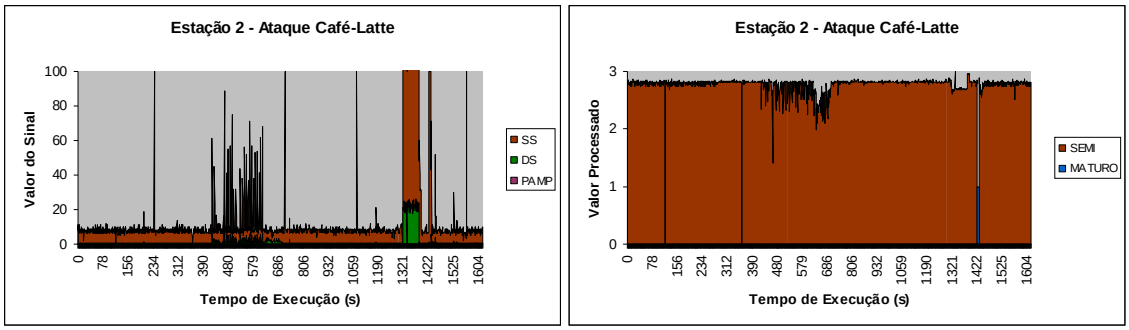


Figura 6.20 – Amostra dos sinais analisados para o ataque Cafe-Latte. No segundo 1422 é mostrado um erro de processamento (gráfico da direita).

Na figura 6.21, a presença de PAMPs durante o tempo de execução, não gerou maturidade para as DCs. O mesmo fato ocorre com a figura 6.22.

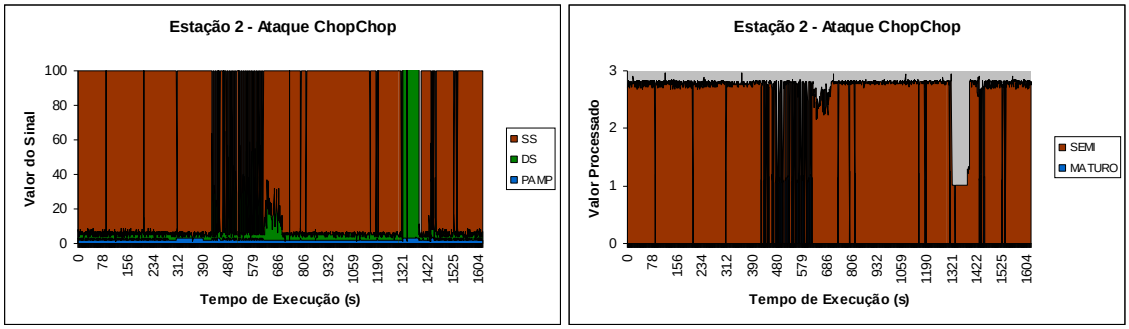


Figura 6.21 – Amostra dos sinais analisados para o ataque ChopChop. Não houve maturidade das DCs.

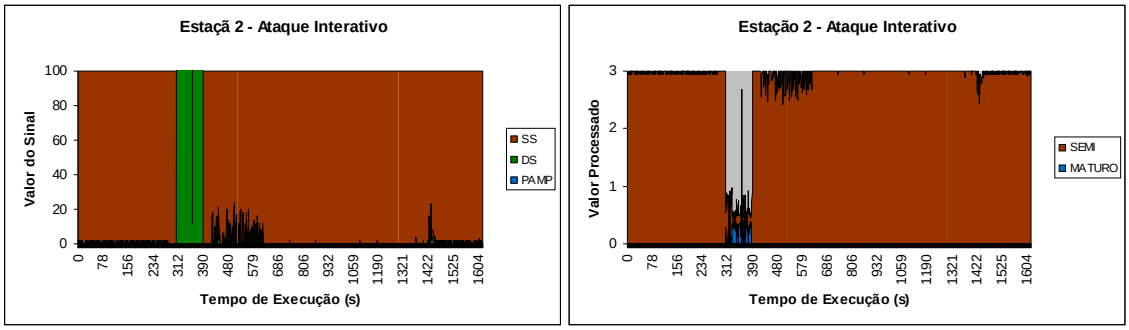


Figura 6.22 – Amostra dos sinais analisados para o ataque interativo. Alguns segundos apresentaram anormalidade, porém, não o suficiente para gerar maturidade para DC.

Durante a execução do teste sobre a estação 2, houve apenas duas chamadas para o agente intermediário, conforme figura 6.23.

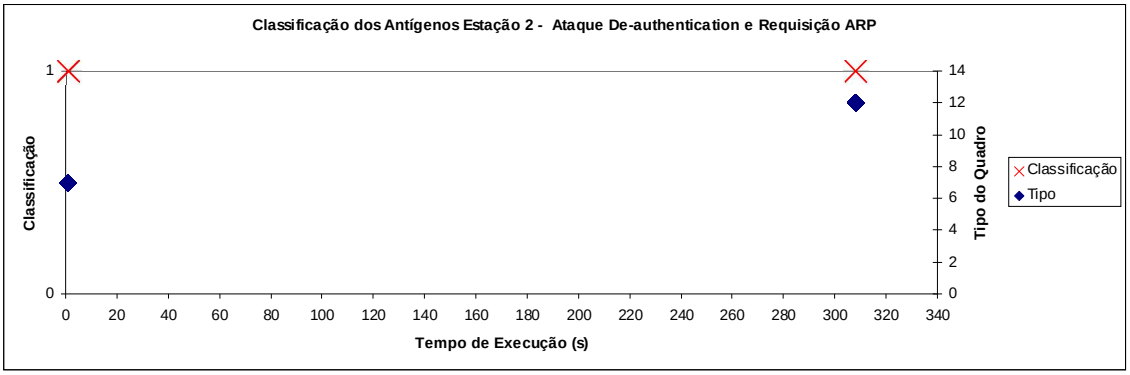


Figura 6.23 – Amostra da classificação dos antígenos para a estação 2. Um erro de classificação foi encontrado no segundo 1.

Dois tipos de problemas foram identificados: quadros *data* e quadros *de-authentication*. No primeiro caso, não houve ataque sobre o segundo 1 e por isso configurou-

se um falso alarme. Para o segundo caso, existe um tempo de resposta para as estações após o envio do pedido de ajuda ao servidor e, por isso, se comparado a figura 6.18 da estação 1, não teve tempo suficiente para que a estação recebesse o agente subalterno responsável por problemas do tipo. Caso o tempo fosse maior, a estação não teria necessidade de enviar o pedido, haja vista que o agente criado é replicado a todas as estações.

As figuras 6.24 até 6.26 representam os resultados coletados da execução do sistema na estação 3. Sob esta estação, somente houve ataque *de-authentication* em modo *broadcast*. Foi possível observar que, durante o ataque, sempre houve vestígios sobre os sinais para os ataques Chopchop e interativo, conforme mostram as figuras 6.25 e 6.26. Porém, nenhum deles gerou falsos positivos, mesmo assim, é um fato a ser estudado com maior profundidade.

A figura 6.27 apresenta os resultados gerados pelo classificador para os pedidos da estação. Os mesmos problemas com os quadros do tipo 5, 6 e 9, ocorrem no segundo 39. Porém, para esta estação, não houve falso positivo para os agentes subalternos.

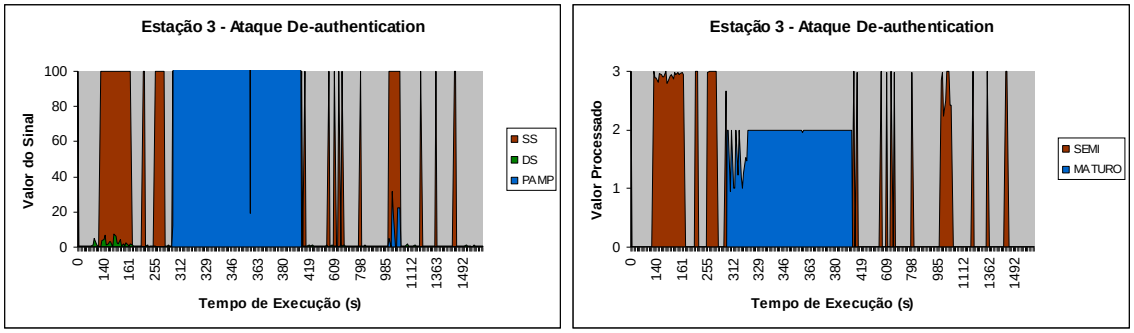


Figura 6.24– Amostra dos sinais analisados para os sinais *de-authentication*. Os segundos em ataque correspondem exatamente àqueles encontrados nas duas estações anteriores.

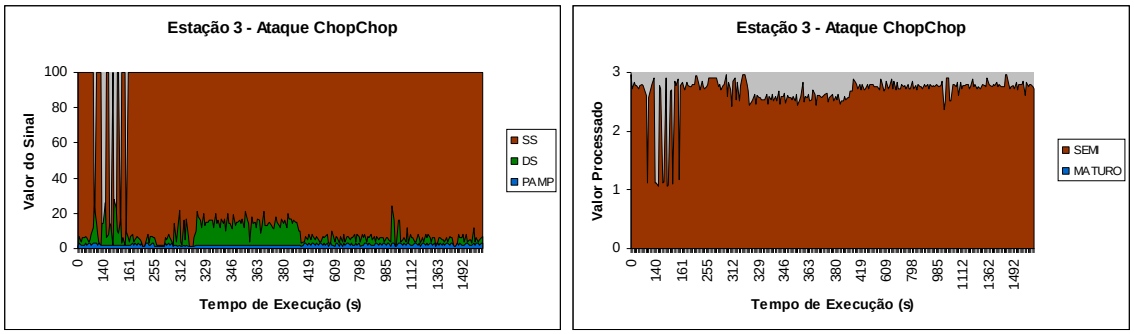


Figura 6.25 Amostra dos sinais analisados para os sinais do ataque ChopChop. Não houve DCs maturas nesse caso.

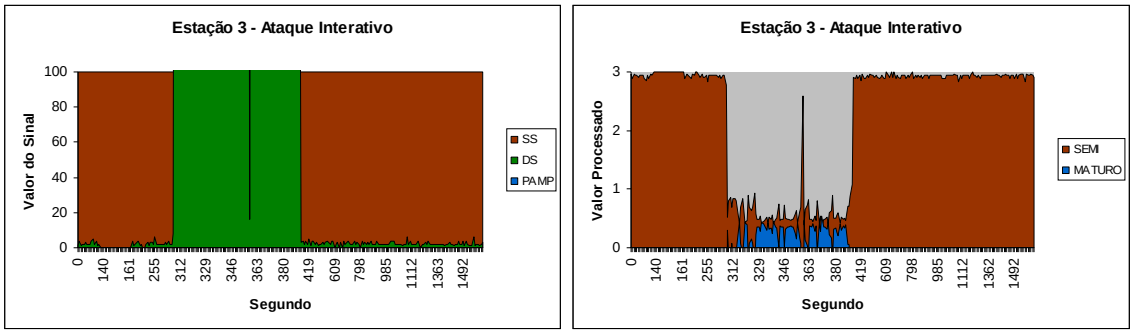


Figura 6.26 Amostra dos sinais analisados para os sinais de ataque interativo. É possível ver a interferência do ataque de-authentication sobre os sinais do ataque interativo.

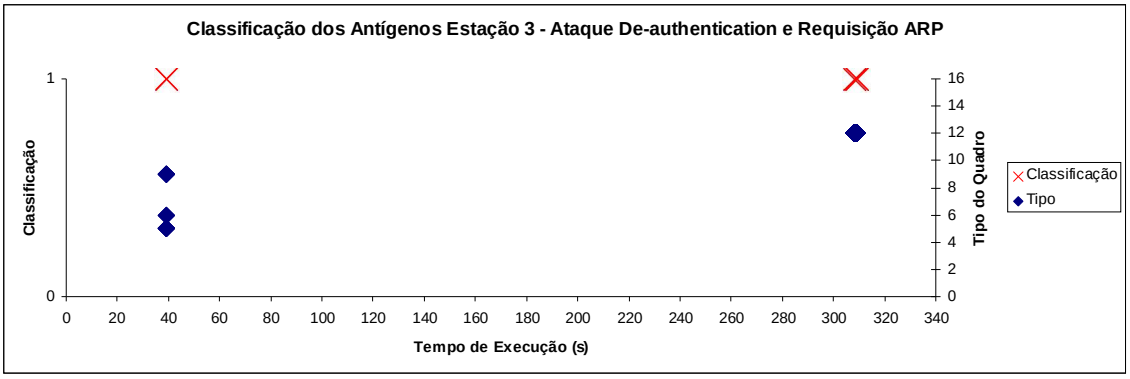


Figura 6.27 – Amostra do resultado para a classificação dos antígenos enviados pela estação 3 ao servidor.

As figuras 6.28 e 6.29 representam os resultados para a estação 4. Assim como na estação 3, nas estações 4 e 5 não houve ataque direcionado as mesmas, apenas em modo *broadcast.* Portanto, as figuras para as estações 4 e 5 são semelhantes e, por isso, somente serão inseridas duas figuras, representando os sinais para os ataques aplicados, refletidos nas duas estações. Tal decisão se faz necessária para evitar figuras repetidas nas quais não possuem nenhum acréscimo ao trabalho, por não apresentarem comportamento anômalo. A figura 6.30 representa os problemas de classificação.

Na primeira figura, é possível ver um falso negativo entre os segundos 350 e 360, esse fato também emergiu para a estação 5. Na segunda figura outro erro foi encontrado, este, gerou falso positivo.

Mesmo ocorrendo falso positivo na detecção, para o classificador, não foi definido como sendo ataque, dessa forma, não foi criado nenhum agente para combate, o que seria falso alarme de agentes.

É preciso reforçar que, a estação detectou um problema (mesmo sendo falso positivo) no segundo 400, porém, não foi necessário pedir ajuda ao agente intermediário do servidor

por já ter sido criado um agente para o problema das requisições ARP (quadro data) nos segundos anteriores em estações da rede. Cada agente subalterno criado é clonado e enviado a todas as estações da rede. Nesse caso, a estação já tinha o agente para verificar se era ou não ataque.

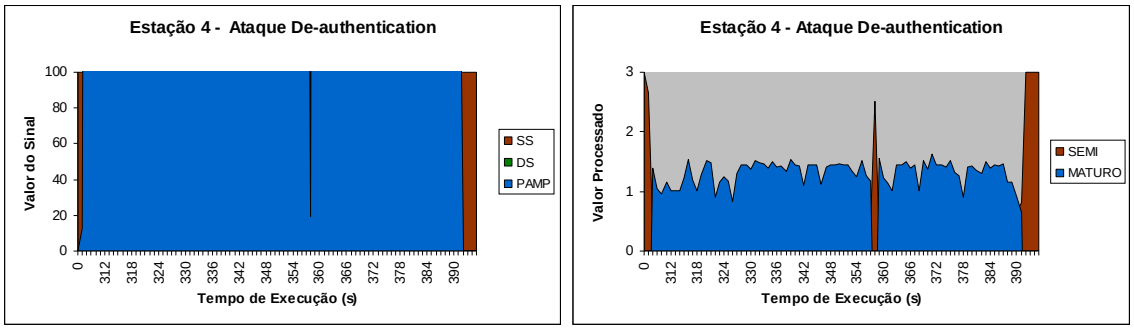


Figura 6.28 – Amostra dos sinais analisados para o ataque *de-authentication* para estação 4. No segundo 358 houve um falso negativo.

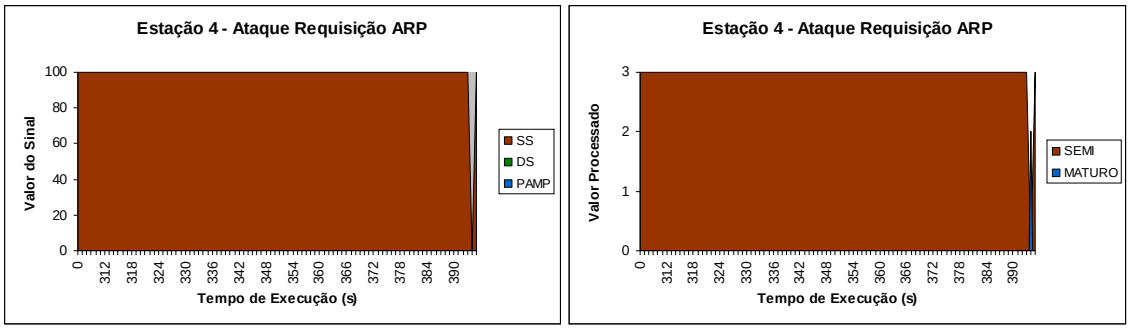


Figura 6.29 – Amostra dos sinais analisados para o ataque requisição ARP sobre estação 4. Neste caso houve um falso positivo no segundo 400.

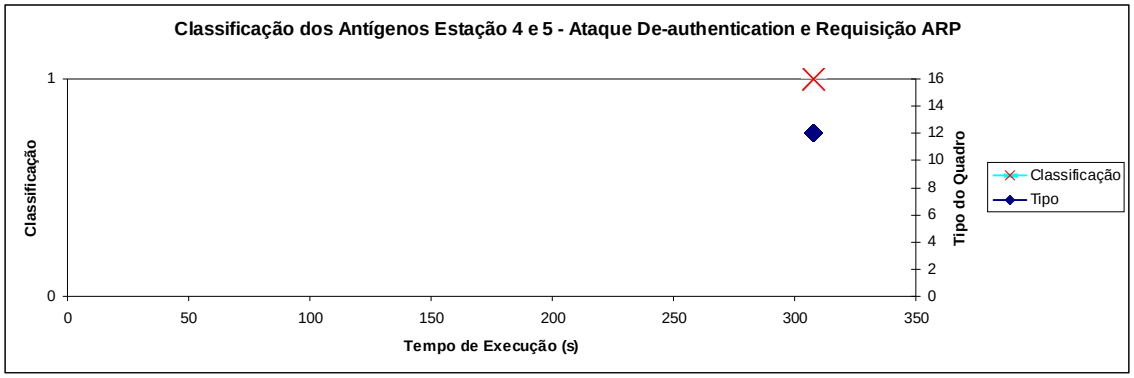


Figura 6.30 – Amostra da classificação dos antígenos da estação 4 e 5 para o ataque de-authentication. Somente um tipo de problema é encontrado.

Na tabela 6.3 são apresentadas informações gerais sobre os dados utilizados para o experimento e, na tabela 6.4 são apresentadas informações detalhadas sobre os efeitos dos ataques sobre todas as estações.

Tabela 6.3 – Informações sobre o experimento 1.

Descrição	Experimento 1
Tamanho da base de dados em pacotes	282041
Limiar de migração	1-2
Segundos em ataque	544 (~9 min)[63]
Tempo equivalente (total)	~30 min
Tempo de resposta quando já existe subalterno	~1segundo

Tabela 6.4 – Informações detalhadas do ataque sobre as estações.

Descrição	Estação 1	Estação 2	Estação 3	Estação 4 e 5
Antígenos	149714	82799	40153	26085
População DC	1636	1636	1636	1636
Células Maturas	642	166	92	83
Células Semi-maturas	872	1092	177	7
Células Imaturas	122	373	1367	1546
Segundos em ataque	544	160	84	84
Falso Positivo	98	6	8	1
Falso Negativo	14	5	1	1
Taxa de Erro (FP)	18,01%	3,75%	9,52%	1,19%
Taxa de Erro (FN)	2,57%	3,12%	1,19%	1,19%
Tamanho das Mensagens (Média)	100KB	100KB	100KB	100KB
Agentes Subalternos	5	2	4	1
Falso Positivo (*Asub*)	3	1	1	0
Falso Negativo (*Asub*)	0	0	0	0
Tempo de Resposta (da detecção até o registro do agente na estação)	~3s	~2s	~2	~2s
Mensagens para Intermediário	2	3	3	1
Quantidade de quadros gerados na rede (estimativa)	~133,33	~200	~200	~66,66

6.6.2 Experimento 2: Ataque Hirte

Neste experimento o sistema foi testado para verificação da eficiência do mesmo sob ataques Hirte.

[63] Somente para a estação 1 que sofreu o ataque completo.

O MAC de uma das estações é usado para a realização do *MACSpoofing* e, por isso, não é possível executar o teste estando a estação em execução na rede.

Na figura 6.31 é possível verificar que o ataque inicia a partir do segundo 91, ocorrendo até o final do experimento, porém, um fato observado é o processo de anormalidade na rede mostrado na Figura 6.32, no caso, os sinais SS e PAMP tiveram valor nulo e por isso, não apresentaram valor após processamento. O valor nulo é fruto da mudança citada no Capítulo 5 para a os valores da tabela de entrada do algoritmo DCA. No caso, quando o valor de SS e PAMP eram iguais a zero, falsos positivos eram gerados. Após observação, foram trocados os valores da tabela para que, caso os dois fossem iguais a zero, o resultado do processamento também fosse nulo.

Esse comportamento, apresentado na Figura 6.32, mostra a interferência do ataque Hirte sobre os sinais de-authentication. O resultado imediato da modificação da tabela foi a diminuição dos falsos positivos.

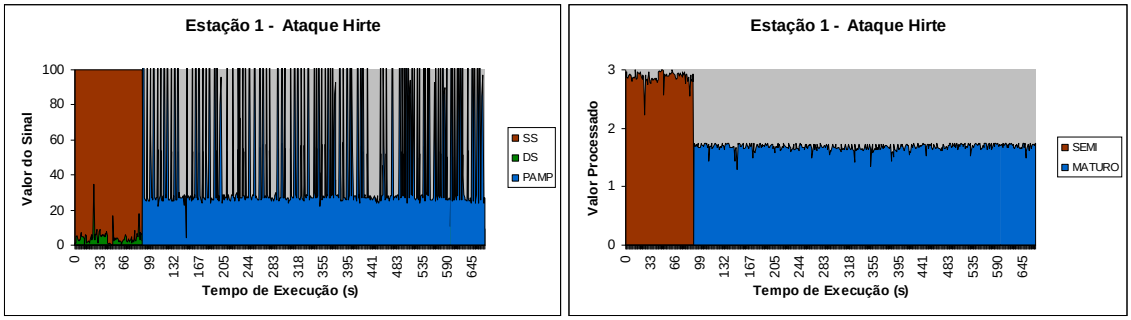


Figura 6.31 – Resultados para os sinais do ataque Hirte. A partir do segundo 91, houve ataque contínuo e maciço sobre a rede.

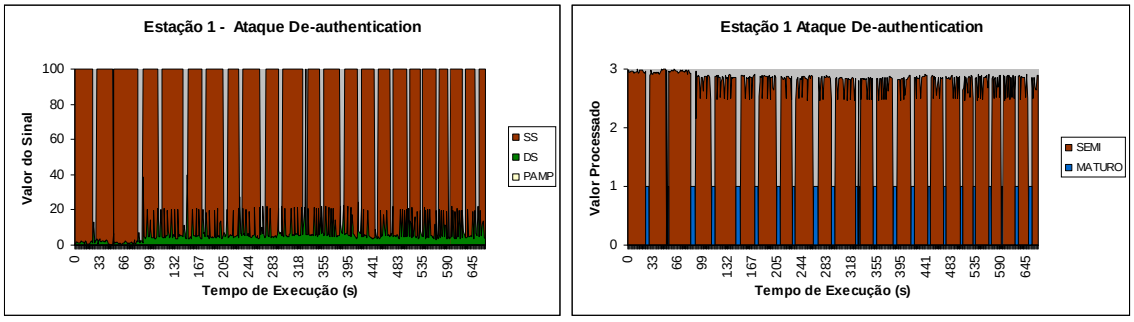


Figura 6.32 – Amostra da interferência do ataque Hirte sobre os sinais de-authentication.

A figura 6.33 representa o processo de classificação dos antígenos passados pela estação ao agente intermediário, nela são mostrados três tipos identificados: tipo 7 (*data*), tipo 6 (*probe response*) e tipo 9 (*null funtion*). Apesar de ter ocorrido ataque a partir do segundo

91, conforme mostrado na figura 6.33, no segundo “0” foi detectado problemas com quadros *data* e, por isso, naquele instante, foi criado um agente subalterno específico para os quadros *data*, não sendo necessária a criação de novo agente quando encontrado o ataque Hirte, no segundo 91.

Com relação aos outros problemas nos segundos 24 e 25, é fruto da detecção dos quadros *probe response* e *null function* em anormalidade. Neste caso, mesmo tendo sido criado em tempo errado, o agente subalterno do tipo data é o mesmo que irá responder ao ataque Hirte.

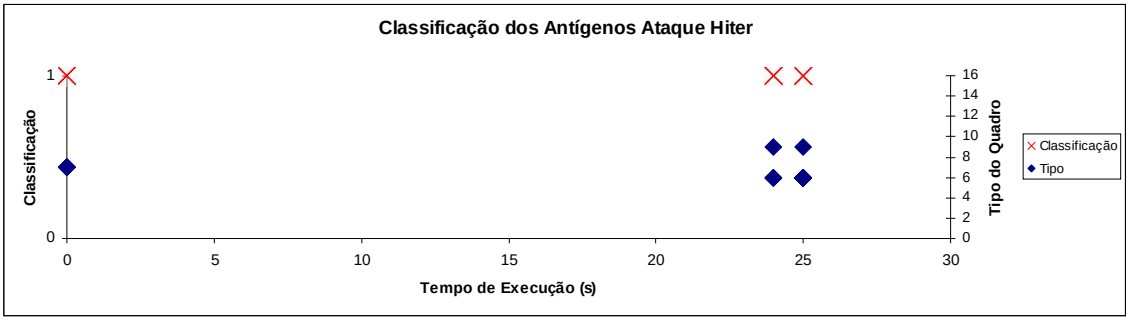


Figura 6.33 – Resultados da classificação dos antígenos para o ataque Hirte. Três agentes subalternos foram criados.

As tabelas 6.5 e 6.6 apresentam descrições detalhadas sobre o experimento 2.

Durante este experimento, atividade anormal de quadros *probe response* foi detectada pelo algoritmo gerando falsos positivos. Todas as 14 ocorrências foram deste tipo de quadro. Por isso, o agente intermediário classificou como ataque. Uma vez que, grandes quantidades destes quadros, oriundos do AP, direcionados para a estação (no mesmo segundo), sem que a mesma tenha solicitado, é uma atividade anormal, é possível dizer que nesse caso, não se trata de falsos positivos reais, apesar de não ser esse o objetivo da análise.

Para esse tipo de ataque não houve falso negativo.

Tabela 6.5 – Dados gerais do experimento 2.

Descrição	Experimento 2
Tamanho da base de dados em pacotes	54063
Limiar de migração	1
População de DC	517
Segundos em Ataque	426
Tempo equivalente	~15 min
Taxa de erro total	3,28%

Tabela 6.6 – Dados detalhados do ataque Hirte. Os falsos positivos na detecção refletem comportamento anômalo para quadros *probe response*.

Descrição	Estação 1
Antígenos	42682
Células Maturas	440
Células Semi-maturas	76
Células Imaturas	1
Falso Positivo (FP)	14
Falso Negativo (FN)	0
Taxa de Erro (FP)	3,28%
Taxa de Erro (FN)	0%
Tamanho das Mensagens (Média)	~100KB
Agentes Subalternos Criados	3
Falso Positivo (Agentes Subalternos)	1
Falso Negativo (*Asub*)	0
Tempo de Resposta (m<20)	~2s
Tempo de resposta quando já existe o agente subalterno	~1s
Mensagens para Intermediário	3
Quantidade de quadros gerados na rede	~200

6.6.3 Experimento 3: Ataque ChopChop

Experimento realizado para teste do sistema sob efeito do ataque ChopChop. Os resultados são apresentados nas figuras 6.34 a 6.36. Destaque para as duas primeiras figuras, nas quais, é possível ver a ação do ataque. Na primeira figura, representando os sinais verdadeiramente coletados para o ataque, nota-se o ataque ocorrendo de duas formas distintas: contínuo e discreto. Na segunda figura é observado um ataque de descoberta de rede através do programa de rede PING. Quadros *data* contendo pacotes de requisição ARP são recebidos pela estação durante todo o tempo do experimento. Entretanto, a quantidade é pequena durante um segundo, cerca de 4 a 5 pacotes, por isso, os valores para o gráfico da figura 6.34 eram muito pequenos, ficando quase imperceptível. Dessa forma, nos segundos onde o valor de PAMP era maior que “0” e SS igual a “0”, PAMP recebeu 100. Tal mudança serviu apenas para facilitar a visualização no gráfico, não alterando nenhum aspecto do sinal.

Novamente foi observado anormalidade nos quadros *probe response* e *null function* que geraram valores nulos para *de-authentication*, conforme mostra a figura 6.36.

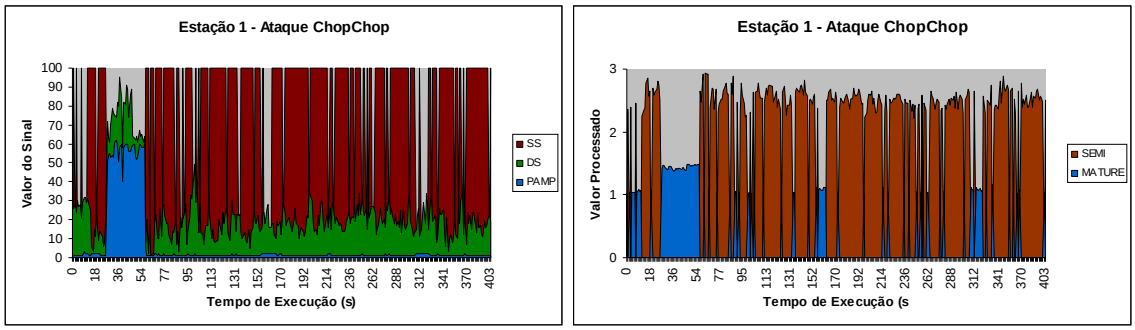


Figura 6.34 – Resultados para os sinais do ataque ChopChop. Durante o período entre 20 e 60 segundos houve um ataque contínuo.

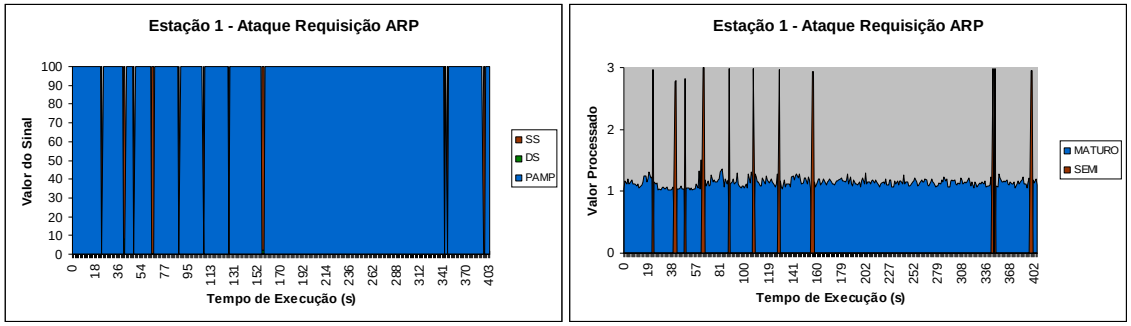


Figura 6.35– Resultados para os sinais do ataque requisição ARP. Um dos reflexos do ataque ChopChop é a ocorrência de quadros de requisição ARP.

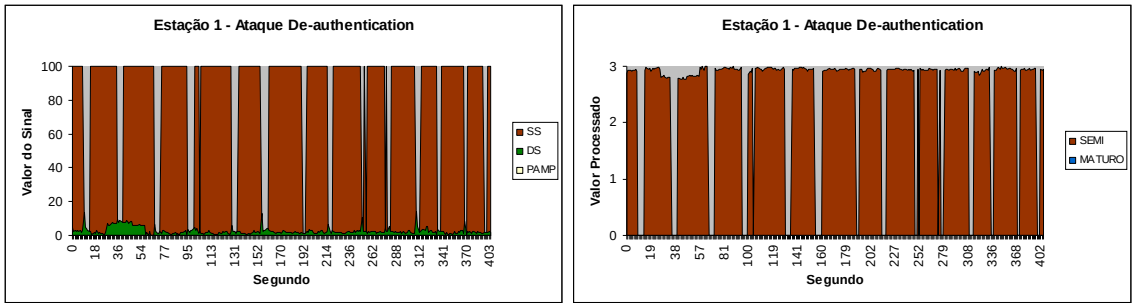


Figura 6.36 – Resultados para os sinais de ataque *de-authentication* na estação 1. Mesmo problema ocorrido durante ataque Hirte.

Na figura 6.37 é possível observar os tipos de problemas identificados pelo classificador do agente intermediário. Nestes casos, foram classificados como ataque e, por isso, estão todos fixados na linha correspondente ao valor 1 (isto é, 1 é verdadeiro e 0 falso). Portanto, foram criados três agentes subalternos equivalentes aos problemas identificados.

A partir do segundo 12 não houve mais requisições, pois todos os antígenos passados para o container de agentes subalternos da estação foram resolvidos, diminuindo o tempo de

resposta ao problema. Os três últimos quadros demonstram o longo tempo para que fosse criado o agente subalterno para problemas do tipo 9 (*Null function*). No caso, 2 segundos que representa o valor médio aferido durante os experimentos.

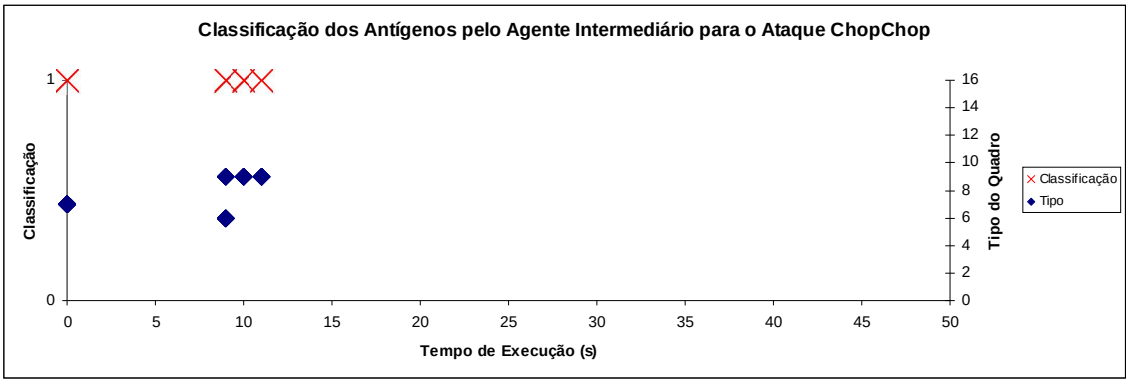


Figura 6.37 – Resultados da classificação dos antígenos para o ataque ChoChop.

As tabelas 6.7 e 6.8 demonstram detalhadamente os resultados obtidos durante o experimento. Através dela, é possível ver a taxa de erros total e por tipo de alarme. Apesar de o ataque principal ser ChopChop, que atua principalmente sobre quadros *data*, outros quadros também apresentaram problemas, sendo classificados como ataque pelo agente intermediário. Por isso, três agentes subalternos foram criados. Nesse caso, não houve falso alarme para o processo de identificação dos antígenos após avaliação dos quadros na base original.

Tabela 6.7 – Dados gerais do experimento 3.

Descrição	Experimento 3
Tamanho da base de dados em pacotes	56978
Antígenos	8991
Limiar de migração	1
Segundos em Ataque	314
Taxa de erro total	2,17%
Tempo equivalente	~10 min

Tabela 6.8 – Dados detalhados do ataque ChopChop.

Descrição	Estação 1
Antígenos	8991
População DC	402
Células Maturas	317
Células Semi-maturas	5
Células Imaturas	80
Falso Positivo	4
Falso Negativo	3

Taxa de Erro (FP)	1,27%
Taxa de Erro (FN)	0,9%
Mensagens para intermediário	2
Tamanho das Mensagens (Média)	100KB
Agentes Subalternos	3
Falso Positivo (*Asub*)	0
Falso Negativo (*Asub*)	0
Tempo de Resposta (m<10)	~1s
Quantidade de pacotes gerados na rede	~133

6.6.4 Experimento 4: Falsa Autenticação

Este experimento serviu para analisar os efeitos na rede do ataque de autenticação falsa. Este ataque é ponto inicial de outros tipos de ataque, pois, o atacante precisa estar associado ao AP. Dessa forma, este experimento se fez necessário para avaliar a capacidade do sistema em detectar problemas quando sob este tipo de ataque. Como explicado anteriormente, para este ataque a estação precisa monitorar o AP para que seja possível detectar falsas autenticações.

As figuras 6.38 até 6.41 mostram os resultados analisados apresentando o comportamento dos sinais do algoritmo de detecção. Destaque para este ataque é a presença de quadros com requisição ARP (figura 6.40) detectados durante todo o experimento. Não houve simulação de ataque diretamente, mas, mesmo assim, o sistema detectou anormalidade no comportamento dos quadros *data* e após análise da base de dados, foi verificada a veracidade da detecção, fato que comprova a eficiência do processamento dos sinais pelo algoritmo.

A figura 6.39 demonstra novamente o problema com os quadros *probe response* e *null function.* Na figura 6.41 foram detectados alguns problemas com quadros com tamanho pequeno, fato que gerou problemas com os sinais do ataque ChopChop. Esse tipo de problema pode ser reflexo do ataque como pode ser algum tipo de anormalidade da rede, porém, não foram estudadas as causas desse problema, sendo definidos como falsos positivos.

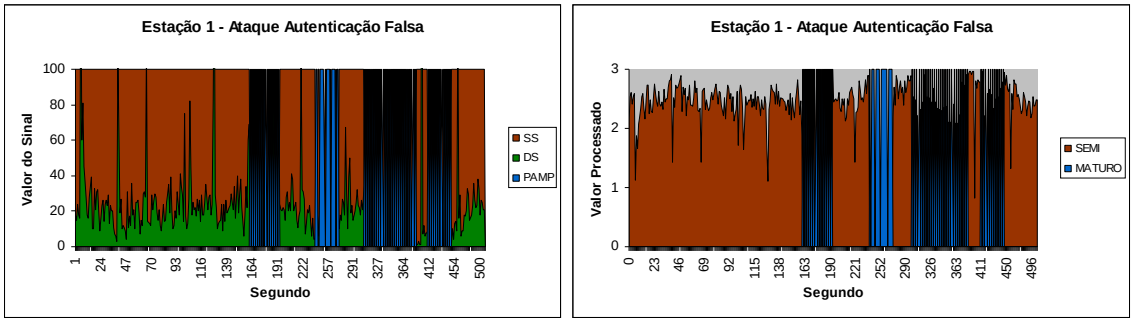


Figura 6.38 – Resultados para os sinais do ataque autenticação falsa. Durante quatro momentos foram detectados ataques.

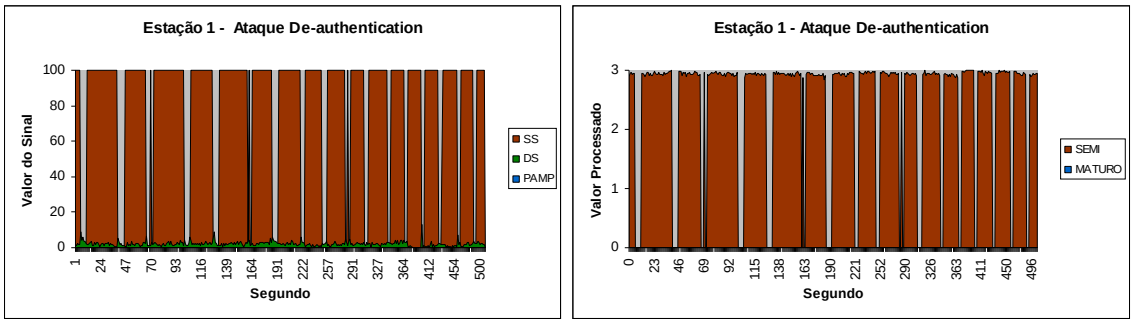


Figura 6.39 – Resultados apresentam interferência nos sinais de-authentication.

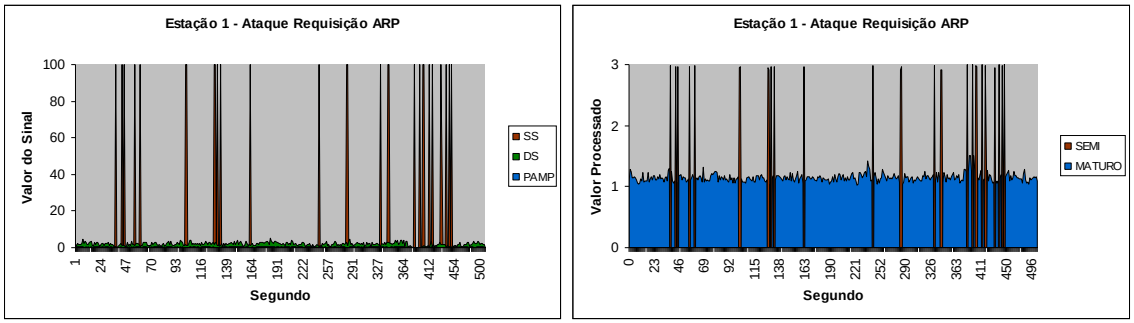


Figura 6.40 – Resultados para os sinais de ataque de requisição ARP. Houve ataque imprevisto.

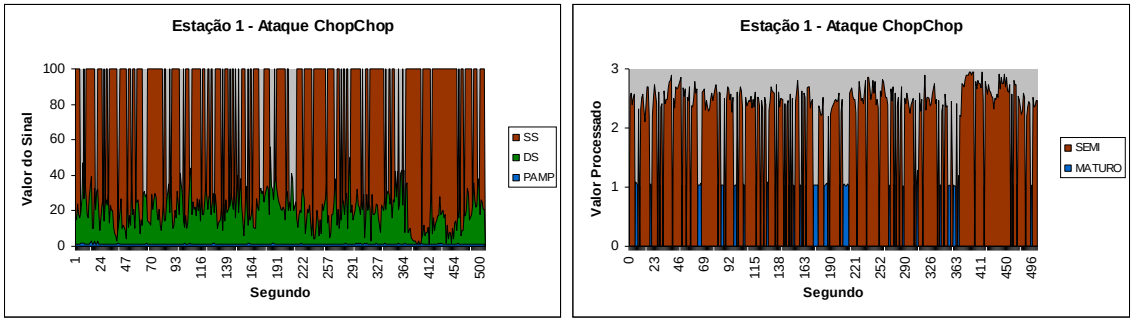


Figura 6.41 – Resultados dos sinais do ataque ChopChop.

Na figura 6.42 apresenta o resultado do classificador. Neste caso houve dois falsos positivos para os tipos *probe response* e *null function*. Ao avaliar a base de dados, foi verificado que os quadros não eram reflexos do ataque, mas sim de uma falha na rede.

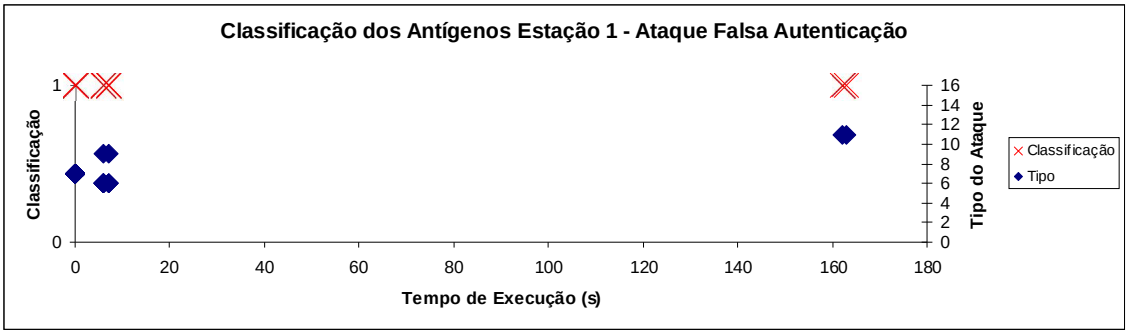


Figura 6.42 – Resultados obtidos com o classificador do agente intermediário. Dois agentes subalternos foram criados de forma errônea, ou seja, falsos positivos.

As tabelas 6.9 e 6.10 mostram mais informações sobre o experimento.

Tabela 6.9 – Dados gerais do experimento 4.

Descrição	Experimento 4
Tamanho da base de dados em pacotes	62607
Limiar de migração	1
Segundos em Ataque de Autenticação	80
Segundo em ataque geral	271
Taxa de erro total	3,75%
Tempo equivalente	~8 min

Tabela 6.10 – Dados detalhados do ataque de autenticação falsa. Durante a detecção houve alguns erros, mas, pode se considerar baixo, apesar de que a presença de falso negativo é indesejável para qualquer rede.

Descrição	Estação 1
Antígenos	8448
População DC	509
Células Maturas	91
Células Semi-maturas	417
Células Imaturas	1
Falso Positivo	11
Falso Negativo	0
Taxa de Erro (FP)	3,75%
Taxa de Erro (FN)	0
Tamanho das Mensagens (Média)	100K

Agentes Subalternos	4
Falso Positivo (*Asub*)	2
Falso Negativo (*Asub*)	0
Tempo de Resposta (da detecção até o registro do agente na estação)	~2s
Mensagens para Intermediário	5
Quantidade de quadros gerados na rede (média)	~333

6.6.5 Experimento 5: Ataque Interativo

Experimento realizado para verificação do comportamento da rede/estação sob o ataque interativo. Um das características deste tipo de ataque é a passagem de quadros *data* com destino *broadcast* e tamanho menor que 100 *bytes*. Durante o experimento, foi instanciado um ataque contínuo sobre uma estação da rede, conforme pode ser visto na figura 6.43, a partir do segundo 122. As figuras 6.44, 6.45 e 6.46 apontam alguma anomalia na rede ao conceber valores nulos tanto para SS quanto para PAMP. Ambos os gráficos das figuras representam ataques que são aplicados sobre quadros *data* e, por isso, de forma geral, é possível dizer que se trata de uma confirmação de que houve anormalidade na rede, haja vista que os quadros *data* são usados pelo ataque interativo.

Finalizando a análise do ataque, a figura 6.47 ilustra o resultado para o agente intermediário e a classificação apresentada quando solicitada. É possível observar três tipos de problemas classificados: (i) *probe response*, (ii) *Null function* e (iii) *data*.

Novamente, foi encontrado problema com os quadros de resposta do AP. Conforme já explicado, não foi definido como falso positivo por se tratar de um evento anormal na rede. Fato que deixa claro nos segundos onde, o valor após processado, zerou.

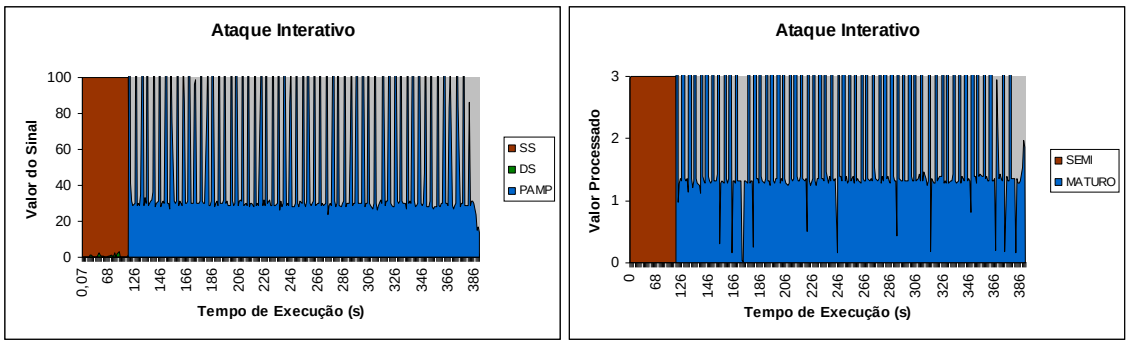


Figura 6.43 – Resultados para o ataque interativo mostrando que a partir do segundo 122 houve ataque.

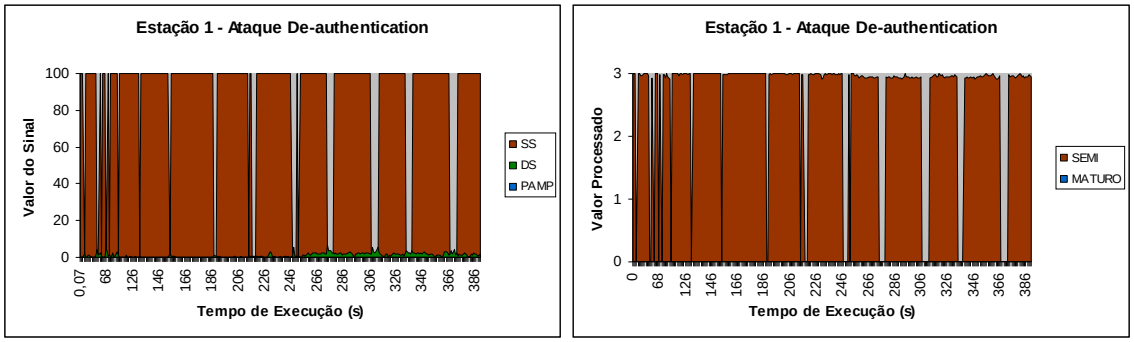


Figura 6.44 – Resultados para o ataque *de-authentication*. Vestígio de anormalidade por não apresentar valores para SS e PAMP.

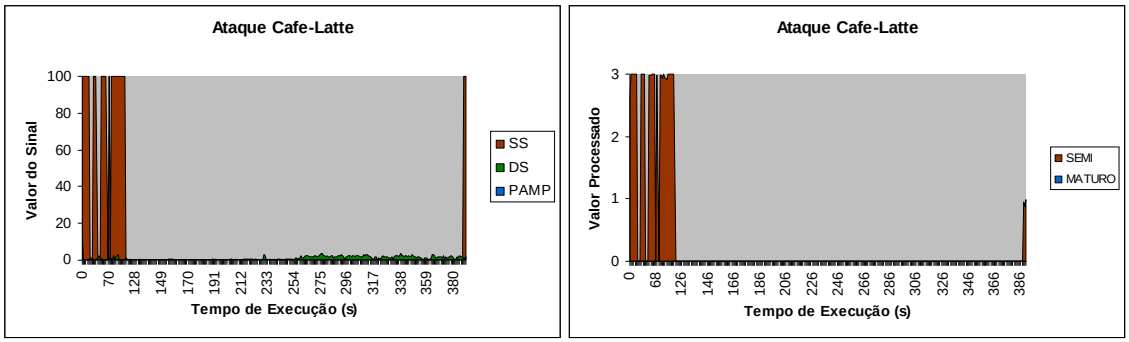


Figura 6.45 – Resultados para os sinais do ataque Cafe-Latte. Mesmo caso da figura anterior.

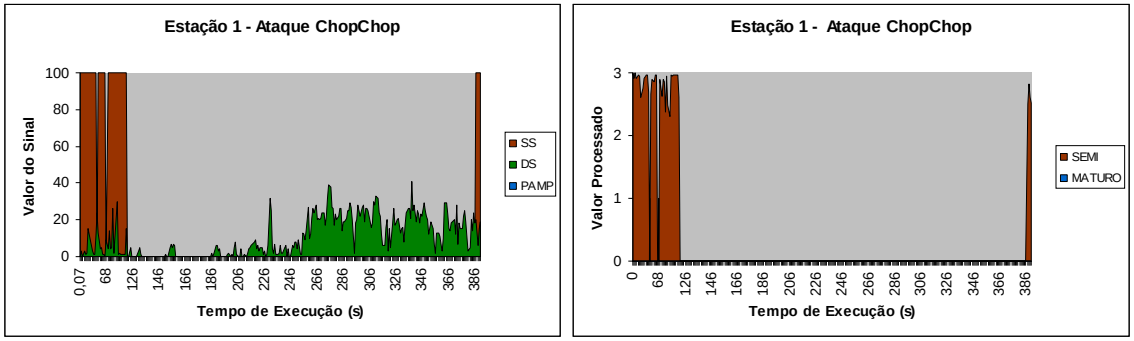


Figura 6.46 – Resultados para os sinais do ataque ChopChop. Novamente nota-se anormalidade na rede através do gráfico dos sinais.

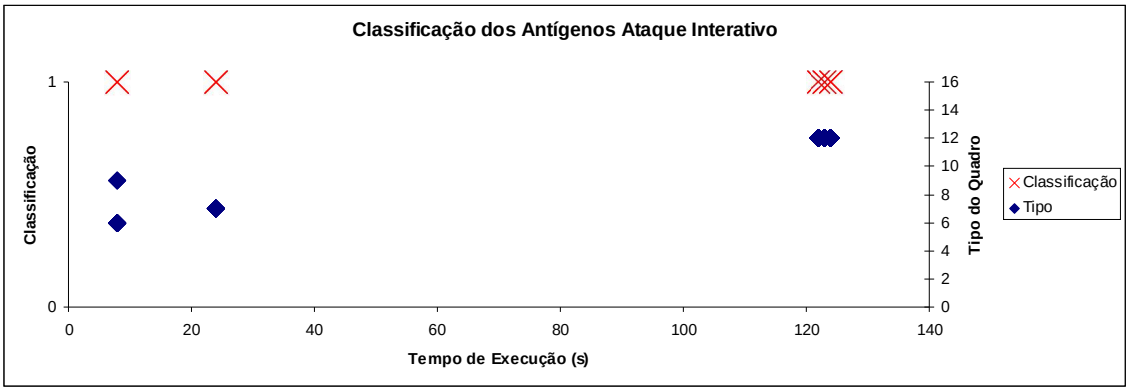


Figura 6.47 – Resultados para o ataque interativo na classificação dos antígenos. Quatro tipos de problemas foram identificados.

As tabelas 6.11 e 6.12 representam os valores coletados durante o experimento. Como ponto fraco, cita-se a presença de falsos negativos que, nesse caso, permite o sucesso do ataque. Esse tipo de falha na detecção poderá comprometer a identificação, mesmo que o funcionamento esteja exatamente como esperado.

Tabela 6.11 – Dados gerais do ataque experimento 5.

Descrição	Experimento 5
Tamanho da base de dados em pacotes	111393
Limiar de migração	1
Segundos em Ataque	271
Taxa de erro total	4,42%
Tempo equivalente	~8 min

Tabela 6.12 – Dados detalhados do ataque interativo. Porém, para a detecção houve alguns erros, mas, pode se considerar baixo, apesar de que a presença de falso negativo é indesejável para qualquer rede.

Descrição	Estação 1
Antígenos	3594
População DC	305
Células Maturas	279
Células Semi-maturas	26
Células Imaturas	0
Falso Positivo	8
Falso Negativo	4
Taxa de Erro (FP)	2,95%
Taxa de Erro (FN)	1,47%
Tamanho das Mensagens (Média)	100KB
Agentes Subalternos	3
Falso Positivo (*Asub*)	0

Falso Negativo (*Asub*)	0
Tempo de Resposta (da detecção até o registro do agente na estação)	2s
Mensagens para Intermediário	5
Quantidade de quadros gerados na rede (média)	~333

6.6.6 Experimento 6: Ataque Cafe-Latte

Este experimento foi realizado para testar o sistema quando em ataque do tipo Cafe-Latte. Os resultados encontrados podem ser vistos nas figuras 6.48 até 6.51 e nas tabelas 6.13 e 6.14.

Pelas figuras apresentadas é possível verificar que o ataque teve dois momentos de maior intensidade contra alguns momentos esporádicos durante o tempo. Por ser um ataque que atua sobre os quadros *data*, houve interferência nos sinais dos ataques ChopChop e Interativo. No caso do ataque interativo, houve sinais PAMP durante o tempo de execução do experimento, porém, ao serem processados, não gerou maturidade das DCs.

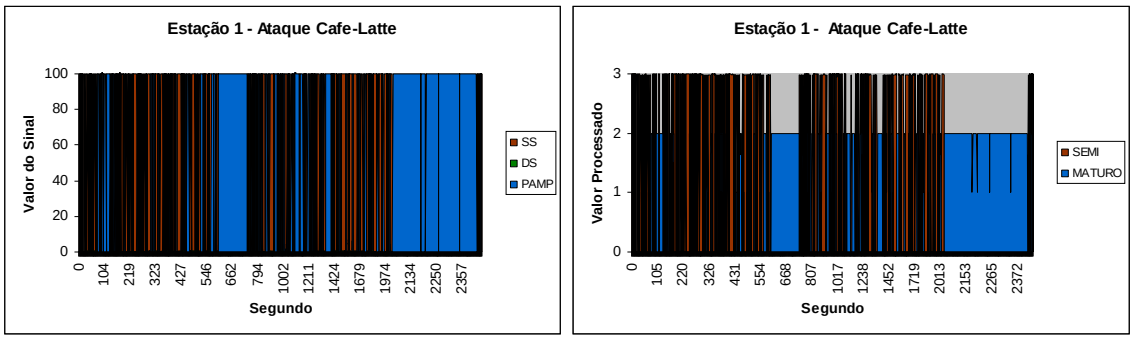


Figura 6.48 – Resultados para os sinais do ataque Cafe-Latte. Em vários segundos ocorreram ataques, entretanto, nos segundos finais ocorreram ataques contínuos na linha do tempo.

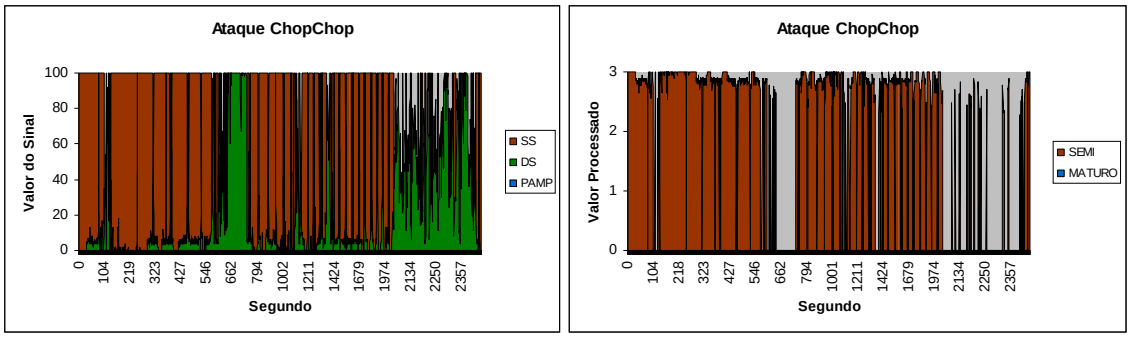


Figura 6.49 – Resultados para os sinais de ataque ChopChop. Novamente a ausência de sinais PAMP e SS podem representar anormalidade na rede.

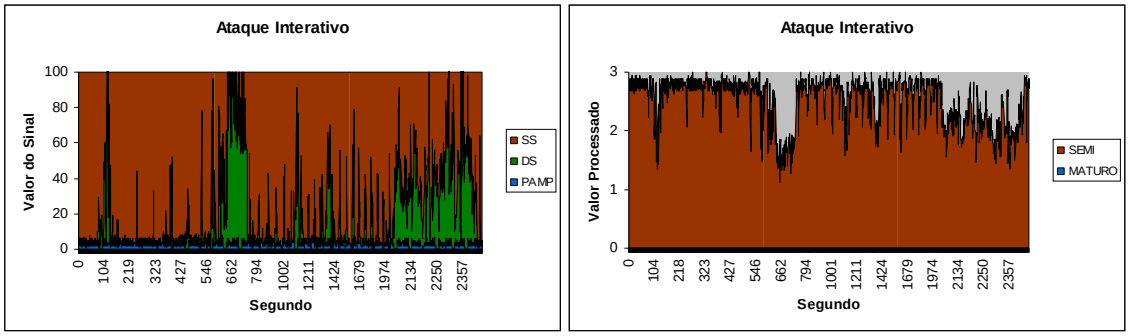


Figura 6.50 – Resultados para os sinais de ataque interativo. Apesar de haver sinais PAMP (com baixa freqüência por segundo) durante o processamento não foram suficientes para produzir DCs maturas.

Afigura 6.51 mostra duas chamadas apenas para o agente intermediário. Nesse caso, três tipos de quadros foram encontrados com anomalias. Por isso, três agentes subalternos foram criados. Novamente, os quadros *null function* e *probe response* foram classificados como ataque.

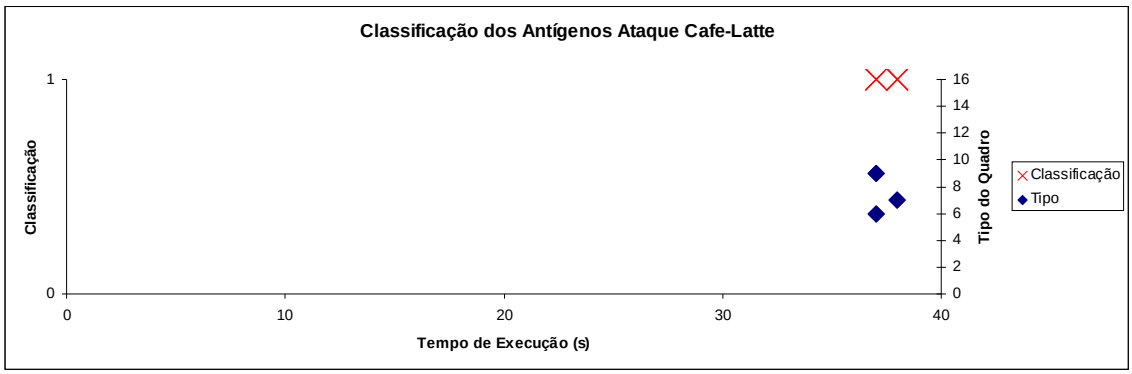


Figura 6.51 - Resultados na classificação dos antígenos para o ataque Cafe-Latte. Apenas duas chamadas ao agente intermediário.

As tabelas 6.13 e 6.14 representam as informações detalhadas para o experimento 6. Pela quantidade de tempo de duração do experimento, a quantidade de erros pode ser entendida como aceitável. A quantidade de falsos negativos pode ser pequena, porém, para qualquer sistema de segurança, falso negativos são considerados os piores casos, pois representa um ataque que não foi detectado, comprometendo a segurança da rede/sistema.

Tabela 6.13 – Dados gerais do experimento 6.

Descrição	Experimento 6
Tamanho da base de dados em pacotes	142876
Limiar de migração	1
Segundos em Ataque (total)	823
Taxa de erros (total)	5,21%

Tempo equivalente	40 min

Tabela 6.14 – Dados detalhados do ataque Cafe-Latte. Observam-se alguns falsos negativos e positivos para a detecção, porém, nenhum falso alarme na classificação do ataque.

Descrição	Estação 1
Antígenos	66000
População DC	2373
Células Maturas	861
Células Semi-maturas	623
Falso Positivo	38
Falso Negativo	5
Taxa de Erro (FP)	4,61%
Taxa de Erro (FN)	0,60%
Tamanho das Mensagens (Média)	100KB
Agentes Subalternos	3
Falso Positivo (*Asub*)	0
Falso Negativo (*Asub*)	0
Tempo de Resposta	~3s
Mensagens para Intermediário	2
Quantidade de quadros gerados na rede	~133

6.6.7 Experimento 7: Ataque de Fragmentação

Experimento realizado para aferição do comportamento da rede/estação quando em ataque de fragmentação.

As figuras 6.52 até 6.54 demonstram os resultados encontrados após execução do experimento. O ataque de fragmentação ocorre a partir do segundo 31, seguindo até o segundo 114. Durante a análise da base de dados para definição dos segundos em ataque, foi detectada ocorrência de quadros data representando pacotes ARP, assim como a ocorrência de vários quadros por segundo com tamanho menor que 100 bytes. Esses eventos anômalos na rede provocaram a maturação das DCs e por isso, foi definido que em todos os segundos houve algum problema na rede.

Dessa forma, na figura 6.55, os ataques provocaram a criação de cinco agentes de combate. Pela análise da base de dados, todos os quadros identificados como ataque estão corretos. Conforme mencionado em todos os experimentos, os quadros *probe response* também apresentaram comportamento anômalo nesse caso. Um tipo de antígeno foi

classificado como normal (corretamente), os outros quatros tipos de antígenos foram classificados corretamente como ataque e por isso geraram agentes subalternos de combate.

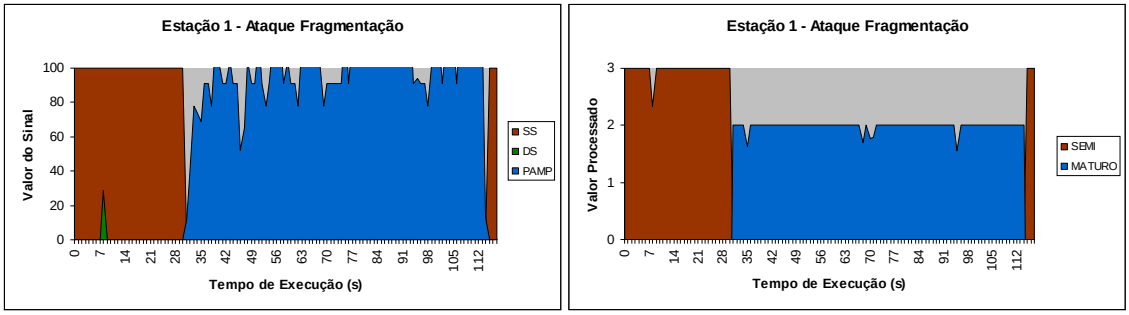


Figura 6.52 – Resultados para o processamento dos sinais para o ataque de fragmentação. O ataque ocorre entre os segundos 31 e 114.

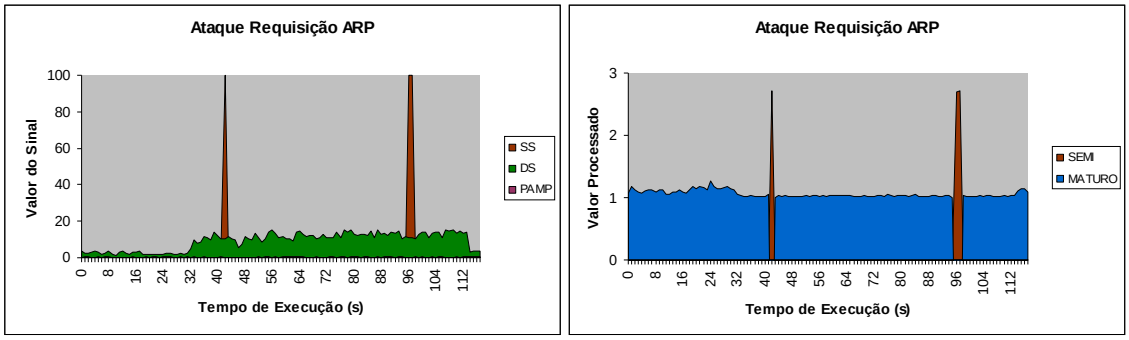


Figura 6.53 – Resultados para os sinais de ataque requisição ARP. Durante todo o tempo foi detectada a ocorrência de quadros de requisição ARP.

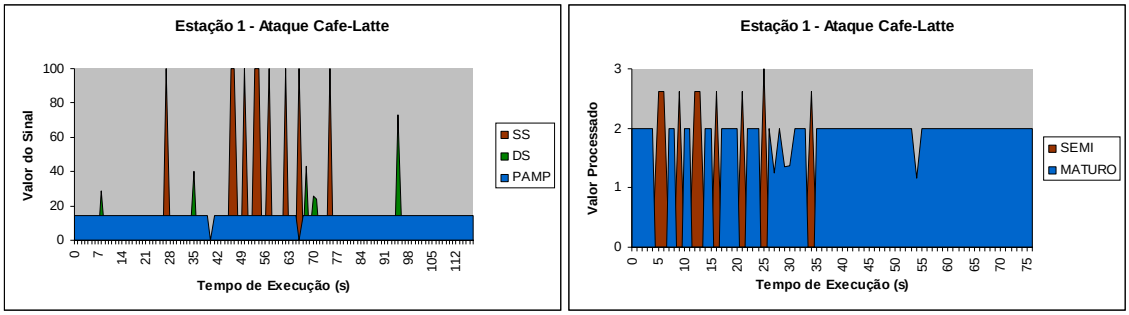


Figura 6.54 – Resultados para os sinais de ataque Cafe-Latte. Assim como o ataque ARP, os sinais deste tipo de ataque detectaram quadros de pequeno tamanho repetidos na maioria dos segundos.

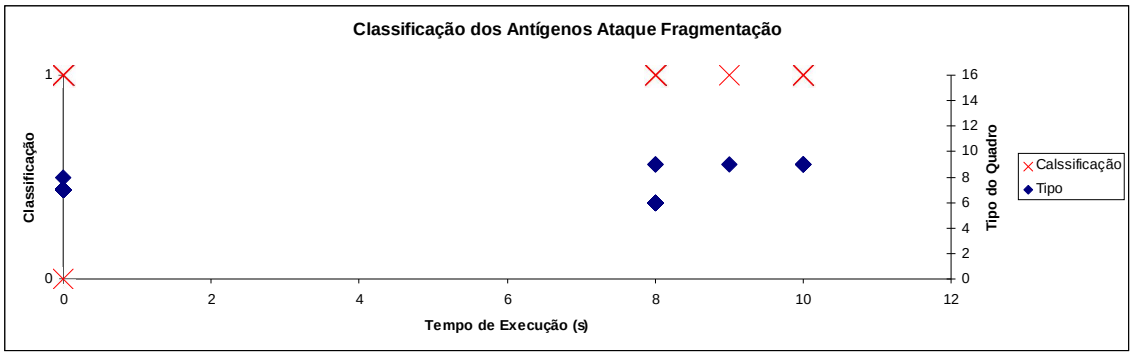


Figura 6.55 – Resultados para o ataque de fragmentação na classificação dos antígenos.

As tabelas 6.15 e 6.16 apresentam os resultados obtidos para o modelo por completo. Com taxa de erros nula, pode-se dizer que para esse experimento os resultados foram ótimos. Fator de destaque ainda pode ser apresentado como a detecção de anormalidades não esperadas durante o experimento, conforme mostraram as figuras 6.60 e 6.61. Um fator em desvantagem é o pouco tempo de ataque sobre a estação.

Tabela 6.15 – Dados gerais sobre o experimento 7.

Descrição	Experimento 7
Tamanho da base de dados em pacotes	22626
Limiar de migração	1-2
Segundos em Ataque (total)	118
Taxa de erros (total)	0,0%
Tempo equivalente	~3 min

Tabela 6.16 – Dados detalhados do ataque de fragmentação. Nenhum falso alarme foi gerado nesse experimento.

Descrição	Estação 1
Antígenos	11440
População DC	118
Células Maturas	118
Células Semi-maturas	0
Falso Positivo	0
Falso Negativo	0
Taxa de Erro (FP)	0,0%
Taxa de Erro (FN)	0,0%
Tamanho das Mensagens (Média)	100KB
Agentes Subalternos	3
Falso Positivo (*Asub*)	0
Falso Negativo (*Asub*)	0

Tempo de Resposta	~2s
Mensagens para Intermediário	4
Quantidade de quadros gerados na rede	~266

6.6.8 Experimento 8: Ataque Sobre WPA

Recentemente um tipo de ataque sobre WPA gerou bastante alarme por sua capacidade de sucesso em apenas 15 minutos segundo seus autores [16]. Por isso, para reforçar a validação do modelo desenvolvido, foi montado um experimento para averiguação do comportamento do mesmo perante o protocolo de segurança para redes sem fio WPA. A principal diferença, entre o comportamento dos ataques, para WEP e WPA, é que o WPA não permite a injeção de quadros como é realizada no WEP. Esse fato ocorre, principalmente, porque o WPA não permite ataque de autenticação falsa. Portanto, é conhecido que os ataques DoS ainda continuam tendo efeito sobre tais redes e, por isso, esse experimento foi planejado para verificar o comportamento do modelo perante estes tipos de ataques.

A figura 6.56 apresenta os resultados para o experimento. Dois tipos de ataques foram realizados: direcionado a uma estação e para toda a rede. Nos gráficos da figura, os valores em azul, que alcançam o máximo (isto é, valor 100), representam o ataque em modo broadcast. Sobre o efeito deste ataque, muitos quadros de-authentication são enviados por segundo. Por outro lado, nos segundos onde os ataques são direcionados a alguma estação da rede, os quadros são enviados de forma contínua, porém, em média, 4 quadros são encontrados durante um mesmo segundo.

A figura 6.57 apresenta apenas uma chamada para o agente intermediário, que classificou corretamente o tipo de quadro com anomalia. Foi criado um agente subalterno para aquele tipo de problema.

Nas tabelas 6.17 e 6.18 são apresentados os resultados para o modelo completo. O baixo número de erros é um fator importante para esse experimento. Mesmo os falsos negativos encontrados, têm relação com a quantidade de quadros encontrada por segundo. Por exemplo, existem alguns segundos em que ocorre apenas um quadro de-authentication e o sistema não consegue determinar como um ataque, principalmente quando ocorrem em intervalos maiores. Portanto, esse problema pode ser resolvido através de um analisador na linha do tempo que armazene resultados do passado para tentar prever ataques no futuro.

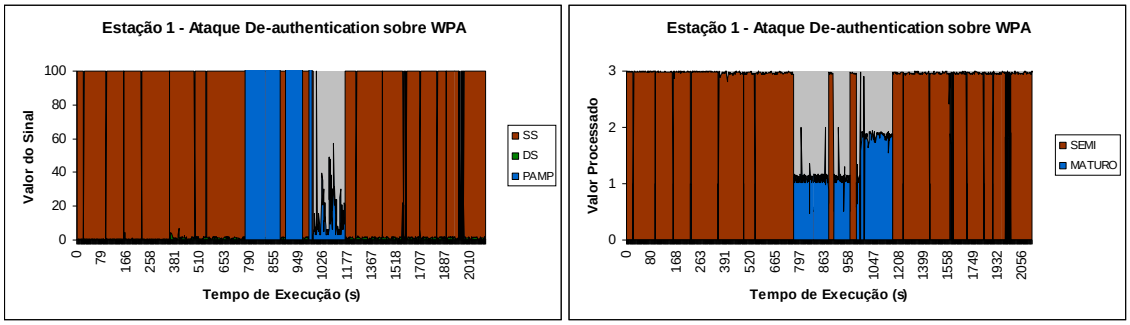


Figura 6.56 – Resultados para o ataque *de-authentication* sobre WPA.

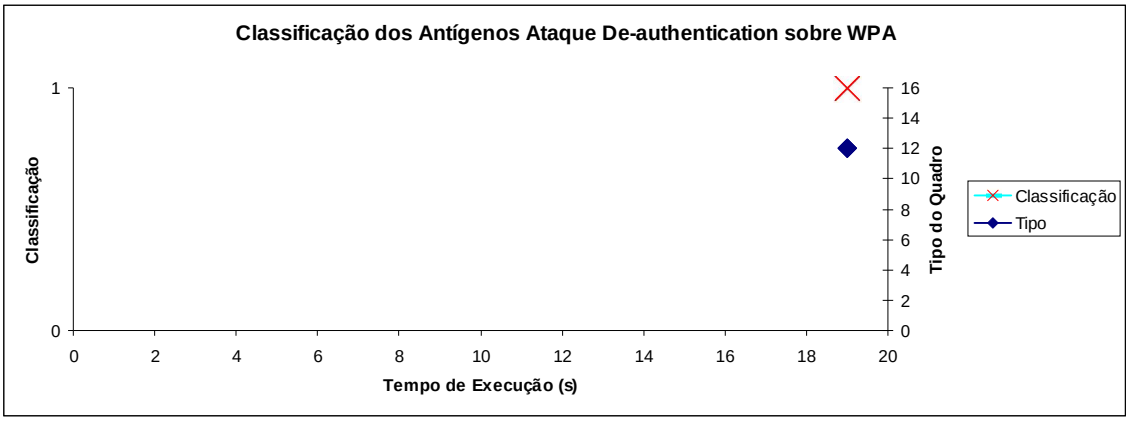


Figura 6.57 – Resultados para o ataque de fragmentação na classificação dos antígenos. O resultado apresentado revela que o sistema identificou corretamente o único ataque ocorrido.

Tabela 6.17 – Dados gerais do experimento 8.

Descrição	Experimento 8
Tamanho da base de dados em pacotes	121604
Limiar de migração	1-2
Segundos em Ataque (total)	459
Taxa de erros (total)	2,38%
Tempo equivalente	~35 min

Tabela 6.18 – Dados detalhados do ataque de-authentication. Os resultados demonstraram que o modelo é capaz de detectar ataques DoS sobre WPA/WPA2.

Descrição	Estação 1
Antígenos	43121
População DC	2001
Células Maturas	465
Células Semi-maturas	1503
Falso Positivo	6
Falso Negativo	5

Taxa de Erro (FP)	1,30%
Taxa de Erro (FN)	1,08%
Tamanho das Mensagens (Média)	100KB
Agentes Subalternos	1
Falso Positivo (*Asub*)	0
Falso Negativo (*Asub*)	0
Tempo de Resposta	~2s
Mensagens para Intermediário	1
Quantidade de quadros gerados na rede	~66

7 – Conclusões e Trabalhos Futuros

Neste capítulo, é apresentado um resumo e conclusão dos principais tópicos trabalhados, assim como é feita algumas sugestões de trabalhos futuros.

7.1 Resumo

Neste trabalho, foi proposto um modelo híbrido de detecção de intrusão para redes sem fio baseadas na especificação IEEE 802.11, conforme apresentado no Capítulo 5. Três técnicas diferentes de inteligência computacional (inclusive de classes diferentes) foram utilizadas: (i) AIS, (ii) MAS e (iii) *naïve Bayes*. A principal característica do modelo é a sua capacidade de adaptação a novas formas de ataque e sua autonomia de funcionamento, onde o conjunto de conceitos efetivamente realiza detecção de anomalias na rede.

No Capítulo 6, foram apresentados oito experimentos, os sete primeiros para o protocolo WEP e o oitavo para o protocolo WPA/WPA2. Esses experimentos foram usados para investigar os principais problemas conhecidos de ataque.

O desempenho do modelo em relação à presença de falsos alarmes tanto para o processo de detecção quanto para a devida identificação do tipo de problema ocorrido na estação, são realizados pelos agentes, básico e intermediário, respectivamente. Também foi investigado o custo da troca de mensagens na rede, assim como o tempo de resposta no caso de já existir um agente de combate para um determinado tipo de problema e no caso de pedido de ajuda da estação ao servidor.

7.2 Discussão

Após a realização dos experimentos e da análise dos resultados, foi possível verificar que o sistema se comportou de acordo com o desejado, ou seja: ele identificou, de forma adaptativa, as ameaças, além de possuir autonomia em relação ao administrador. Alguns problemas, oriundos da dificuldade em trabalhar com grandes bases de dados (algumas acima da casa de 500 M*bytes*) prejudicaram a aplicação de experimentos com todos os ataques ao mesmo tempo. Esse poderia ser o pior caso para o modelo. Mesmo assim, a taxa de erros encontrada mostrou que, nos primeiros experimentos, alguns erros provenientes do desenvolvimento do sistema levaram a uma taxa máxima de 18% no primeiro experimento. Nos últimos

experimentos, as taxas apresentaram ligeira queda, alcançando o valor zero para o experimento 7.

Além da baixa quantidade de erros (isto é, falsos alarmes), outros fatores emergiram dos experimentos, permitindo que o modelo pudesse alcançar resultados bastante satisfatórios, como exemplo, a detecção de eventos não esperados. Portanto, os principais pontos de destaque foram:

- **Tempo de resposta para o ataque:** fator de extrema importância para um IDS, o tempo de resposta precisa ser o mais curto possível. Dessa forma, em analogia ao sistema inato e o adaptativo, o tempo de resposta difere em duas situações. A primeira acontece quando o sistema nunca teve contato com o problema detectado, neste caso, o sistema demorou em média 2 a 3 segundos para responder ao questionamento feito pela estação (representa o sistema adaptativo). A segunda refere-se exatamente ao fato de já existir agente de combate para o problema detectado no sistema. Para este caso, passou a ser instantâneo, com tempo de resposta menor que 1 segundo (representa o sistema inato);
- **Taxa de erros (falsos alarmes):** o objetivo principal de qualquer IDS/WIDS é executar suas tarefas sem erros. Essa talvez seja a maior dificuldade em projetos de detecção de intrusão. Geralmente é difícil atingir um bom equilíbrio entre os falsos positivos e falsos negativos. Na maior parte do tempo, quando um lado apresenta bons valores, o outro apresenta dificuldades. Neste trabalho, os resultados demonstraram maiores taxas para falsos positivos. Porém, os valores não foram tão díspares e houve um relativo equilíbrio na matriz de confusão;
- **Detecção de eventos não previstos:** durante a execução dos experimentos, alguns eventos foram detectados e identificados como anômalos. Um dos principais foi o quadro de resposta a requisições feitas ao AP pela estação. Outros também foram identificados como anomalia e, após análise das bases de dados, foi averiguado que, na maior parte dos casos, houve realmente eventos anormais para aqueles quadros. O fator preponderante nestes casos é o reflexo dos ataques a que foram submetidas as estações. Dessa forma, emerge uma conclusão de que o conjunto de sinais, quando trabalhados em conjunto, pode detectar vários tipos de anomalias, mesmo aquelas não conhecidas;
- **O custo do tráfego na rede:** um ponto muito importante para um sistema que opera usando a rede é a quantidade de tráfego que será gerado por ele na rede. Por

isso, esse foi um ponto investigado. Ao inferir o tamanho médio das mensagens trocadas entre os agentes da estação e do servidor, ficou aparente um tamanho padrão de 100KB. Esse valor pode parecer grande, mas não o é, haja vista que, ao passar uma mensagem solicitando ajuda ao agente intermediário, o agente básico precisa passar os antígenos concomitantemente. Ora, cada antígeno é um objeto na visão do sistema e é conhecido que um objeto possui atributos e esses atributos têm tamanho, que pode ser dos tipos *byte*, *float*, *int*, *double*. Observando a estrutura do antígeno no Capítulo 5, é possível ver que existem vários atributos, sendo que o tipo *double* foi escolhido por lidar com valores precisos. Também foram usadas *strings*, que na verdade é um objeto em JAVA e tem tamanho variável. Computacionalmente falando, *strings* são pesados para trabalhar e *double* reserva 8 bytes (64 bits). Portanto, o tamanho dos antígenos pode ser um problema ao ser enviado pela rede. Por isso, foi avaliada a quantidade de quadros que poderiam ser gerados durante a execução do sistema. Os valores encontrados sugerem um máximo de 5 a 6 mensagens trocadas entre os agentes que, aplicando a Equação 6.2, foi possível ver que a quantidade média de quadros que foram gerados pelo sistema ficou na faixa de 194. Portanto, 194 mensagens geradas durante um experimento com 40 minutos, por exemplo, representa um valor ínfimo para a rede;

- **Solução distribuída:** ao usar estações que podem monitorar tanto a rede quanto a si própria, fez emergir uma qualidade na qual o sistema é capaz de, ao haver algum erro com uma estação (por exemplo, não conseguir detectar o ataque), outra estação poderá detectar algo errado e gerar aviso para o servidor que irá inspecionar com o agente superior. Ainda nesse item, é preciso salientar que, a cada agente criado para um tipo de combate, todas as estações da rede recebem um agente clonado. Dessa forma, se um ataque ocorrer sobre duas estações, após a primeira fazer o pedido de ajuda ao servidor, as outras receberão a solução para o problema automaticamente, ajudando a diminuir o tráfego na rede, assim como o tempo de resposta;
- **Portabilidade:** por ter sido desenvolvido em plataforma livre, o sistema poderá ser implantado em qualquer sistema operacional. Portanto, esse é um ponto importante para a utilização prática como ferramenta de segurança;
- **Escalabilidade:** os experimentos realizados com volumes de estações e carga de redes diferentes não apresentaram mudanças substanciais de performance. A

hipótese é de que, ao não definir o que é próprio e não próprio ao sistema (ou seja, procedimento de custo computacional elevado), pode-se dizer que a abordagem não indica possuir problema de escalabilidade. Sendo que, para afirmar isso categoricamente, há de haver mais estudos, principalmente com redes maiores e mais complexas;

- **Automatização:** um dos objetivos deste trabalho foi o desenvolvimento de um modelo que pudesse executar todas as rotinas de forma automatizada (isto é, sem que fosse preciso a interferência humana nas decisões tomadas). Para validar essa premissa, todos os agentes foram instanciados e definidos para tomarem decisões durante o tempo de execução. Por exemplo, o agente básico foi capaz de detectar problemas não conhecidos, enviar pedido de ajuda ao agente intermediário, que criou os agentes subalternos de combate sem interferência humana. O agente superior, por exemplo, é o agente responsável pela auditoria do sistema. Quando são encontrados problemas, ele gera relatório para o administrador, assinalando o foco do problema. Esse processo permite que o difícil processo de procura de erros e problemas em *logs* seja feito de maneira automatizada e contínua. Como exemplo, durante os experimentos, algumas vezes a rede sofreu queda, por não conseguir acessar as estações para auditá-las. O agente reportou comportamento estranho da rede e foi possível comprovar que havia problemas na conexão.

Como desvantagens encontradas no modelo citam-se:

- **Ferramenta de captura de quadros:** um dos pontos fundamentais para o modelo é o tratamento dos dados de entrada. Para este trabalho, não foi possível o desenvolvimento de uma ferramenta de captura de alto nível, em muito pela falta de tempo hábil e pela dificuldade do desenvolvimento. Por esse motivo, o tratamento dos quadros capturados precisou ser realizado por outra ferramenta. Porém, nem todas as informações que poderiam ser úteis no início do processo foram trabalhadas;
- **Placas de rede:** para o experimento com autenticação falsa, foi averiguado que uma estação não poderia verificar se estava ocorrendo falsa autenticação com seu próprio endereço MAC, haja vista que o atacante geralmente usa o MAC quando a estação não está conectada à rede ou então quando é atacada por ataques DoS. De acordo com essa dificuldade, foi definido que, para esse caso, cada estação faria o monitoramento junto ao AP, ou seja, com informações sobre todas as estações da

rede e do AP, a estação poderia verificar os quadros de autenticação que são enviados ao AP. Entretanto, essa solução resolveu o problema de imediato. Porém, haverá problemas posteriores. Por exemplo, para executar essa função, o sistema precisará de duas placas de rede, isso porque não é possível trabalhar com uma placa de rede normalmente quando ela está configura em modo monitor;

- **Técnica usada no classificador:** a técnica (*naïve Bayes*) utilizada para classificação dos antígenos apresentou resultado abaixo do esperado para o processo de validação. Durante os experimentos, não foram observados problemas de falsos negativos para o classificador. Mas, mesmo assim, há um índice de 16% de erros, o que pode ser alto para aplicação em uma rede mais complexa. Para esse trabalho, não foram abordadas técnicas de aprendizado dinâmico;
- **Sensibilidade aos sinais:** para que possa haver qualidade na detecção, os sinais precisam ser cuidadosamente estudados e, nos experimentos, demonstrou ser altamente sensíveis (isto é, pode gerar falsos alarmes) a qualquer alteração.

7.3 Conclusão

É possível afirmar que o sistema atingiu seu objetivo inicial: a execução automatizada e adaptação ao desconhecido. O sistema de detecção apresentou resultados expressivos em todos os experimentos, mostrando que o modelo de sinais é um modelo com potencial para um WIDS, podendo inclusive ser aplicado a modelos cabeados. Os resultados mostraram ainda que o processo de detecção não ficou preso ao problema de escalabilidade dos AIS de primeira geração, pois não foi necessário definir o que é próprio da rede/sistema. O reflexo de alguns ataques sobre outros sinais, que não aqueles específicos, mostraram a sensibilidade na escolha dos sinais para cada tipo de ataque.

O sistema de identificação de ataques mostrou bons resultados para a criação dos agentes, mesmo quando alguns tipos de problemas não foram trabalhados nos sinais, caso dos quadros *probe response* e *null function*, que são reflexos dos ataques sobre a rede, principalmente ataques DoS (ver Seção 6.7.1).

Os bons resultados, mesmo para redes com WPA/WPA2, mostra que o sistema possui boa portabilidade e adaptação. Dessa forma, é possível dizer que o modelo é adaptável ao ambiente e as configurações.

O tamanho das mensagens e a quantidade de quadros gerados por experimento mostraram que o sistema não é capaz de interferir no tráfego da rede. Comparando com os

sistemas, inato e adaptativo, do HIS, foi possível verificar que, após a identificação e criação dos agentes de combate, a clonagem e o envio para todas as estações, o sistema tem o tempo de resposta diminuído, o que comprova a eficiência do modelo imunológico (isto é, o sistema inato é mais rápido que o sistema adaptativo).

7.4 Trabalhos Futuros

Houve pontos que não foram totalmente investigados, a maioria deles ligados à operação e do desempenho. Dessa forma, há algumas avenidas de pesquisa que podem ser trilhadas em trabalhos futuros com vistas ao aperfeiçoamento do modelo. Entre elas podem ser citados:

- Aprofundamento dos estudos que objetivem melhores formas para captura dos sinais;
- Concepção e desenvolvimento de ferramenta para filtragem e, captura de quadros, adaptável ao modelo, com o objetivo de diminuir o tamanho das bases de dados;
- Estudo das rotinas de combate com a finalidade de aperfeiçoar os agentes subalternos durante o tempo de execução:
- Aplicação e comparação do módulo de classificação (presente no agente intermediário) com outras técnicas de computação inteligente;
- Aperfeiçoamento do modelo de auditoria usando técnicas de mineração de dados;
- Adaptação do modelo aos ambientes de rede Ad-Hoc, com o intuito de observar o comportamento em redes sem infra-estrutura e de alta complexidade;
- Estudo para aplicação do modelo mediante as novas soluções de segurança propostas pelo IEEE.

Referências

[1] AHMED, R. & SPRENT, J., **Immunological Memory.** The Immunologist, 7/1-2, pp. 23-26, 1999.

[2] AICKELIN, U. and CAYZER, S., **The Danger Theory and Its Application to Artificial Immune Systems.** In Proceedings of the First International Conference on Artificial Immune Systems (ICARIS02), pp. 141-148, 2002.

[3] AICKELIN, U.; BENTLEY, P.; CAYZER, S.; KIM, J.; MCLEOD, J. **Danger Theory: The Link Between AIS and IDS?** In Proceedings of the Second International Conference on Artificial Immune Systems (ICARIS-03), p. 147-155, September, 2003.

[4] AICKELIN, U.; GREENSMITH, J. and TWYCROSS, J., **Immune system approaches to intrusion detection – a review.** In Proceedings of the Second International Conference on Artificial Immune Systems (ICARIS03), pp. 316-329, 2004.

[5] AIRCRACK-NG, **Hirte Attack.** At. http://www.aircrack-ng.org/doku.php?id=hirte, (Acessado em Janeiro de 2010).

[6] AL-HAMMADI, Y.; AICKELIN, U. and GREENSMITH, J., **DCA for Bot Detection.** In IEEE Congress on Evolutionary Computation, Hong Kong, pp. 1807-1816, June 1- 6, 2008.

[7] ALVARES, L. O. e SICHMAN, J. **Introdução aos Sistemas Multiagentes.** Em: JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA, 16.; CONGRESSO DA SBC, 17, Brasília. Anais, Brasília: SBC, 1997. pp.1-38, 1997.

[8] ANDERSON, J. **Computer security technology planning study 2.** Technical Report, Electronic Systems Division, Air Force Systems Command, Hanscom Field, 1972.

[9] _____, **Computer Security Threat Monitoring and Surveillance.** Technical Report, James P. Anderson Co. Fort Washington, PA, 1980.

[10] ANDERSON, M. L., **How to study the mind: An introduction to embodied cognition.** In F.Santoianni and C. Sabatano, eds. Embodied Cognition and Perceptual Learning in Adaptive Development; Cambridge: Cambridge Scholars Press. 2005.

[11] ANTONIEWICZ, B. **802.11 Attacks.** In Foundstone Professional Services, White Paper. 46 p. 2008. At: www.foudstone.com, Acessado em 01 de dezembro de 2009.

[12] ARBAUGH, W. A. **An inductive chosen plaintext attack against WEP/WEP2.** IEEE Document 802.11-01/230, At http://www.cs.umd.edu/~waa/attack/v3dcmnt.htm May 2001. Acessado dia 25 de outubro de 2009.

[13] ARBAUGH, W. A.; SHANKAR, N. and WAN, Y. J. **Your 802.11 wireless network has no clothes.** At http://www.cs.umd.edu/~waa/wireless.pdf, 13 p. Mar. 2001. Acessado dia 23 de outubro de 2009.

[14] AXELSSON, S. **Intrusion Detection Systems: A survey and Taxonomy.** [S.l.]: New Riders, 2000.

[15] BACE, R. G. **Intrusion Detection.** 1st Edition, McMillan Technical Publishing, p. 374, Indianapolis, USA, 2000.

[16] BANCHEREAU, J. & STEINMAN, R. M., **Dendritic Cells and the Control of Immunity.** Nature, 392, pp. 245-252, 1998.

[17] BARKEN, L. **Wireless Hacking: projects for Wi-Fi Enthusiasts.** Syngress Publishing, Inc. Rockland, USA, 370 p. 2004.

[18] BECK, M. and TEWS, E. **Practical attacks against WEP and WPA.** 2008. At: http://dl.aircrack-ng.org/breakingwepandwpa.pdf. Acessado dia 15 de novembro de 2009.

[19] BELLIFEMINE, F.; POGGI, A.; RIMASSA, G. **JADE - A FIPA-compliant agent framework.** In Telecom Italia internal technical report. Part of this report has been also published in Proceedings of PAAM'99, London, pp.97-108, April 1999.

[20] BELLIFEMINE, F.; CAIRE, G.; POGGI, A. and RIMASSA, G., **JADE - A White Paper.** At http://jade.tilab.com/ Acessadop dia 21 de dezembro de 2009.

[21] BENTLEY, P.; GREENSMITH, J. and UJJIN, S., **Two ways to grow tissue for artificial immune systems.** In ICARIS-05, LNCS 3627, pages 139–152, 2005.

[22] BERSINI, H. VARELA, F. J., **Hint for Adaptive Problem Solving Gleaned from Immune Networks.** Parallel Problem Solving from Nature, pp. 343-354, 1990.

[23] BERSINI, H. **Immune Network and Adaptive Control.** Proceedings of the First European Conference on Artificial Life, MIT Press. 1991.

[24] BERSINI, H. CALENBUHR, V., **Frustrated Chaos in Biological Networks.** J, Theor. Biol., 188, pp. 187-200, 1997.

[25] BEUTLER, B. Innate immunity: an overview. Mol Immunol., v. 40, n. 12, p. 845-59,2004.

[26] BERNARDES, A. T. & dos SANTOS, R. M. Z. **Immune Network at the Edge of Chaos.** J. theor. Biol., 186, pp. 173-187. 1997.

[27] BLAKE, C. L.; HETTICH, S. and MERZ, C. J., **UCI repository of machine learning databases.** 1998.

[28] BONNA, C. A. & KOHLER, H. **Immune Networks.** Annals of the New York Academy of Sciences, 418. 1983.

[29] BRADSHAW, J.M., DUTFIELD. S., BENOIT, P., WOOLLEY, J.D., **KAoS: toward an industrial-dtrength open agent architecture.** in Bradshaw, J. (Eds),Software Agents, AAAI Press, Menlo Park, CA, 1997.

[30] BRENNER, P., **Technical Tutorial on the IEEE 802.11 Protocol.** At http://sss-mag.com/pdf/802_11tut.pdf acessado em 29 de outubro de 2009.

[31] BROOKS, R. A., **A robust layered control system for a móbile robot.** IEEE Journal of Robotics and Automation, Vol. 2, Issue 1, pp. 14-23, March 1986.

[32] BURNET, F. M., **The Clonal Selection Theory of Acquired Immunity.** Cambridge University Press, 1959.

[33] _____, **Clonal Selection and After.** Theoretical Immunology, (Eds.) G. I. Bell, A. S. Perelson & G. H. Pimbley Jr., Marcel Dekker Inc., pp. 63-85, 1978.

[34] BUSSMANN, S. and DEMAZEAU, Y., **An agent model combining reactive and cognitive capabilities.** In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS-94), Munich, Germany, 1994.

[35] CACHE, J. and LIU, V. **Hacking Exposed Wireless: wireless security secrets.** McGraw-Hill Osborne Media, 1st Edition, March 26, 2007.

[36] CAWLEY, G. C. and TALBOT, N. L., **Fast exact leave-one-out cross-validation of sparse least-squares support vector machines.** Neural Networks, Elsevier Science Ltd, Oxford, UK, Vol. 17, Issue 10, December, 2004.

[37] CAYZER, S.; SULLIVAN, J., **Modeling Danger and Anergy in Artificial Immune Systems**. In Proceedings of the IEEE Genetic and Evolutionary Computation Conference (GECCO 2007), London, England, United Kingdom, 2007.

[38] CHAN, C.; STARK, J. and GEORGE, A., **The Impact of Multiple T cell – APC Encounters and the Role of Anergy.** In Journal of Computational and Applied Mathematics, 184, pp. 101-120, 2005.

[39] CERT-BR. **Cartilha de Segurança para Internet.** In Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil. At. http://cartilha.cert.br/conceitos/sec1.html Acessado em novembro de 2009.

[40] CHAABOUNI, R. **Break WEP Faster with Statistical Analysis.** In School of Computer and Communication Sciences, École Polytechnique Féderale de Lausanne,

At http://lasecwww.epfl.ch/pub/lasec/doc/cha06.pdf, 2006. Acesso em 03 de novembro de 2009.

[41] COHEN, F. **Computer viruses.** Computers & Security, 6:22–35, 1987.

[42] COUTINHO, A. **The Network Theory: 21 Years Later.** Scand. J. Imm., 42, pp. 3-8. 1995.

[43] COSTA, M. T. C. **Uma Arquitetura Baseada em Agentes para Suporte ao Ensino à Distância.** Tese de Doutorado, DEPS, Universidade Federal de Santa Catarina, Abril de 1999.

[44] COSTA, A. L., **Conhecimento social dinâmico: uma estratégia de cooperação para sistemas multi-agentes cognitivos.** 121 f. Tese de doutorado. FEE – Universidade Federal de Santa Catarina, Santa Catarina, 2001.

[45] COUTINHO, A. **The Self Non-Self Discrimination and the Nature and Acquisition of the Antibody Repertoire.** In Annals of Immunology. Vo. 131D, 1980.

[46] _____., **Beyond Clonal Selection and Network.** Imm. Rev., 110, pp. 63-87. 1989.

[47] CROSBIE, M.; SPAFFORD, G. **Defending a Computer System using Autonomous Agents.** PhD Tesis, Departament of Computer Sciences, Purdue University, 1995.

[48] DANZIGER, M.; LACERDA, M. and LIMA NETO, F. B., **Danger Theory and Multi-agents Applied for Addressing the Deny of Service Detection Problem in IEEE 802.11 Networks.** In Ninth International Conference on Intelligent Systems Design and Applications, ISDA 2009, pp. 695-702, Pisa, Italy, November 30 – December 2, 2009.

[49] DASGUPTA, D., **Artificial Neural Networks and Artificial Immune Systems: Similarities and Differences.** Proc. of the IEEE SMC, 1, pp. 873-878, 1997.

[50] _____., **Artificial Immune Systems and their applications.** Springer-Verlag New York, Inc. Secaucus, NJ, USA No p. 300, 1998.

[51] DASGUPTA, D.; FORREST, S., **Artificial immune systems and their applications.** In: [S.l.]: Springer-Verlag Berlin and Heidelberg GmbH, cap. An Anomaly DetectionAlgorithm Inspired by the Immune System, p. 262–277. 1999.

[52] DASGUPTA, D. and GONZALEZ, F., **An Immunity-based technique to characterize intrusions in computer networks.** In IEEE Transactions on Evolutionary Computation, Vol. 6, No. 3, pp. 281-291, June 2002.

[53] DEBAR, H.; DACIER, M. and WESPI, A., **Computer Networks: the international journal of computer and telecommunications networking.** In Special issue on computer network security, Vol. 31, issue 9, pp. 805-822, April, 1999.

[54] DECKERD, G., **Wireless Attacks from an Intrusion Detection Perspective.** In Sans Institute InfoSec Reading Room. November 23, 2006. At: http://www.sans.org/reading_room/whitepapers/honors/wireless_attacks_from_an_intrusion_detection_perspective_1681?show=1681.php&cat=honors Acessado em 23 de dezembro de 2009.

[55] DE CASTRO, L. N., **Engenharia Imunológica: Desenvolvimento e aplicação de ferramentas computacionais inspiradas em sistemas imunológicos artificiais.** Tese de Doutorado, DCA – FEEC/Unicamp, Campinas/SP, Brasil, Maio de 2001.

[56] DE CASTRO, L. N. e ZUBEN, F. V., **An Evolutionary Immune Network for Data Clustering.** In Proceedings IEEE Symposium Artificial Neural Network, pp. 84-89, 2000.

[57] _____., **The Clonal Selection Algorithm with Engineering Applications.** In Proceedings of The Genetic and Evolutionary Computation Conference 2000 (GECCO'OO) - Workshop Proceedings. July 8-12, Las Vegas, USA (2000) 36-37, 2000.

[58] _____., **AiNet: An Artificial Immune Network for Data Analysis. In Data Mining: A Heuristic Approuch.** pp. 231-259, June 2002.

[59] _____., **Learning and Optimization Using the Clonal Selection Principle.** IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems (IEEE) 6 (3): 239–251, 2002.

[60] DEMAZEAU, Y. and MULLER, J. P., **Decentralized Artificial Intelligence.** Amsterdam, Elsevier Science Publisher B. V., pp. 3-13, 1990.

[61] DENNING, D., **An intrusion detection model.** Em Proceedings of the Seventh IEEE Symposium on Security and Privacy, p´aginas 119–131, 1986.

[62] DHAESELEER, P.; FORREST, S. and HELMAN, P., **An Immunological approach to change detection: algorithms, analysis and applications.** In Proceedings of the 1996 IEEE Symposium on Computer Security and Prvacy, USA: IEEE Press, pp. 110-119, 1996.

[63] D'OTREPPE, T., **Aircrack-ng.** At http://www.aircrack-ng.org/ Acessado dia 11 de julho de 2009.

[64] DREHER, H., **The Immune Power Personality.** Penguim Books, 1995.

[65] DURFEE, E. H.; LESSER, V. R. and CORKILL, D. D., **Trends in Cooperative Distributed Problem Solving.** In: IEEE Transactions on Knowledge and Data Engineering, KDE-1(1), pages 63-83, March 1989.

[66] DURFEE, E. and ROSENSCHEIN, J. S. A., **Distributed Problem Solving and Multi-Agent Systems: comparisons and examples.** In 13th International Workshop on Distributed Artificial Intelligence, Washington, July 17-19, 1994.

[67] EARLE, A. E., **Wireless Security Handbook.** Auerbach Publications, Boca Raton, New York. 403p. 2006.

[68] FARMER, J. D., PACKARD, N. H. and PERELSON, A. S., **The Immune System, Adaptation, and Machine Learning.** Physica 22D, pp. 187-204, 1986.

[69] FERBER, J., **Reactive Distributed Artificial Intelligence: principles and applications.** In Foundations of Distributed Artificial Intelligence, eds, G. O-Hare and N. Jennings, 287-314, New York Wiley, 1996.

[70] FERBER, J. and GASSER, L., **Intelligence Artificielle Distribuée.** Tutorial Notes of the 11th Conference on Expert Systems and theirs Applications, Avignon, France, 1991.

[71] FERNANDES, J. H. C., **Ciberespaço: Modelos, Tecnologias, Aplicações e Perspectivas da Vida Artificial à Busca por uma Humanidade Auto-Sustentável.** I Jornada de Atualização em Informática, 1998. Disponível On-line em: http://www.cic.unb.br/~jhcf/MyBooks/ciber/doc-ppt-html/Welcome.html , acessado dia 19 de novembro de 2009.

[72] FEWER, S., **ARP Poisoning An investigation into spoofing the Address Resolution Protocol.** In Harmony Security, 2007. At: http://www.harmonysecurity.com/files/HS-P004_ARPPoisoning.pdf , Acessado dia 22 de dezembro de 2009.

[73] FININ, T.; FRITZSON, R.; MCKAY, D. and MCENTIRE, R., **KQML as an agent communication language.** Proceedings of the International Conference on Information and Knowledge Management. ACM Press, NY, 1994.

[74] FLORES-MENDEZ, R. A., **Towards a Standardization of multi-agent system frameworks.** At. http://www.acm.org/crossroads/xrds5-4/multiagent.html, 1999. Acessado em 01 de agosto de 2009.

[75] FLUHRER, S.; MANTIN, I. and SHAMIR, A., **Weaknesses in the Key Scheduling Algorithm of RC4.** In the Eighth Annual Workshop on Selected Areas in Cryptography, August 2001. 23 p. At

http://www.crypto.com/papers/others/rc4_ksaproc.ps Acessado dia 25 de outubro de 2009.

[76] FORREST, S., A. PERELSON, ALLEN, L. & CHERUKURI, R., **Self-Nonself Discrimination in a Computer.** Proc. do IEEE Symposium on Research in Security and Privacy, pp. 202-212. 1994.

[77] FORREST, S. & HOFMEYR, S. A., **John Holland's Invisible Hand: An Artificial Immune System.** Presented at the Festschrift held in honor of John Holland, University of Michigan (1999). At: http://all.net/books/iw/iwarstuff/ftp.cs.unm.edu/pub/forrest/festschrift.pdf Acessado em 15 de setembro de 2009.

[78] _____., **Immunology as Information Processing.** In Design Principles for the Immune System and Other Distributed Autonomous Systems. SEGEL, L. A. and COHEN, I. Eds, New York: Oxford University Press, 2000.

[79] FORSDYKE, D.R., **The Origins of the Clonal Selection Theory of Immunity.** In FASEB. Journal 9:164-66, 1995.

[80] FRANKLIN S. and GRAESSER, A., **Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents.** Proceedings of the Third International Workshop on Agent Theories, Springer-Verlag, 1996.

[81] FU, H.; YUAN, X.; ZHANG, K.; ZHANG, X.; XIE, Q., **Investigating Novel Immune-Inspired Multi-Agent Systems for Anomaly Detection.** In: IEEE Asian-Pacific Services Computing Conference (APSCC 2007), 2007.

[82] FU, H.; YUAN, X.; WANG, N., **Multi-agents Artificial Immune System (MAAIS) Inspired by Danger Theory for Anomaly Detection.** In IEEE International Conference on Computational Intelligence and Security Workshops, 2007.

[83] GASSER, L., **Social conceptions of knowledge and action: DAI foundations and open system semantics.** Artificial Intelligence 47, 1-3, pp. 107-138, 1991.

[84] GHAFFAR, A. and NAGARKATTI, P., **Tolerance and Autoimmunity**. In University of South Carolina School of Medicine, Microbiology and Immunology On-line, 2010. Acessado em 01 de fevereiro de 2010.

[85] GOMES, J. DASGUPTA, D., **Evolving Fuzzy Classifiers for Intrusion Detectio.** In Proceedings of third annual information assurance workshop, June 17-19, 2002.

[86] GRAMP, F. T. and MORRIS, R. H., **UNIX Operating System Security.** AT&T Bell Laboratories Technical Journal 63, pp. 1649-1672, October, 1984.

[87] GREENSMITH, J.; AICKELIN, U. and TYCROSS, J., **Detecting Danger: Applying a Novel Immunological Concept to Intrusion Detection Systems.** Poster Proceedings of ACDM 2004 Engineers' House, Bristol, UK, 2004.

[88] GREENSMITH, J., **The Dendritic Cell Algorithm.** PhD Thesis, University Of Nottingham, 2007.

[89] GREENSMITH, J. and AICKELIN, U., **Dentric Cells for SYN Scan Detection.** In Proceedings of the IEEE Genetic and Evolutionary Computation Conference (GECCO 2007), pp. 49–56, London, England, United Kingdom, 2007.

[90] GREENSMITH, J.; TWYCROSS, J. and AICKELIN, U., **Dendritic Cells for Anomaly Detection.** In: IEEE Congress on Evolutionary Computation (CEC 2006), Vancouver, BC, Canada, p. 664-671, 2006.

[91] GREENSMITH, J.; AICKELIN, U.; CAYZER, S., **Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomaly Detection.** In Proceedings of the Fourth International Conference on Artificial Immune Systems (ICARIS-2005), 14-17 August, Banff AB, Canada, p. 153-167, 2005.

[92] GREENSMITH, J.; AICKELIN, U. and TWYCROSS, J., **Articulation and clarification of the dendritic cell algorithm.** In International Conference on Artificial Immune Systems (ICARIS06), LNCS 4163, pp. 404-417, 2006.

[93] GREENSMITH, J.; AICKELIN, U. and TEDESCO, G., **Information Fusion for Anomaly Detection with the Dendritic Cell Algorithm.** In Elsevier Science Publisher B. V. Vol. 11. Issue 1, pp. 21-34, January 2010.

[94] GREENSMITH, J.; FEYEREISL, J. and AICKELIN, U., **The DCA:SOMe Comparison. A comparative study between two biologically inspired algorithms**. Journal Article - Evolutionary Intelligence 1 (2), Springer Berlin, pp. 85-112, June, 2008.

[95] GREGIO, A., **Falhas em Políticas de Configuração: uma análise do risco para as redes sem fio da cidade de São Paulo.** Anais do 6° Simpósio de Segurança em Informática, São José do Rio Preto, 2004.

[96] GU, J., LEE, D., PARK, S. & SIM, K., **An Immunity-based Security Layer Model.** Proc. do GECCO'00, Workshop on Artificial Immune Systems and Their Applications, pp. 47-48. 2000.

[97] GU, F.; AICKELIN, U. and GREENSMITH, J., **An Agent-based Classification Model.** In 9th European Agent Systems Summer School (EASSS2007), Durham, UK, 2007.

[98] GU, F.; GREENSMITH, J. and AICKELIN, U., **Exploration of the Dendritic Cell Algorithm with the Duration Calculus.** Proceedings of the 8th International Conference on Artificial Immune Systems (ICARIS 2009), Lecture Notes in Computer Science 5666, York, UK, pp. 54-66, 2009.

[99] HOFFMANN, G. W., **A Theory of Regulation and Self-Nonself Discrimination in an Immune Network.** Eur. J. Imm., 5, pp. 638-647. 1975.

[100] HOFMEYR, S., **An Immunological Model of Distributed Detection and its Application to Computer Security.** PhD Tesis, Department of Computer Sciences, University of New Mexico, 1999.

[101] HOFMEYR, S. e FORREST, S., **Immunity by design: An artificial immune system.** Em Proceedings of the Genetic and Evolutionary Computation Conference, volume 2, páginas 1289–1296. Morgan Kaufmann, 1999.

[102] _____., **Architecture for an artificial immune system.** In Evolutionary Computation, Vol. 8, No 4, pp. 443-473, 2000.

[103] HURLEY, C., **Wardriving & Wireless Penetration Testing.** Syngress Publishing, Inc. Rockland. 435 p. 2007.

[104] IEEE. **Institute of Electrical and Electronics Engineers.** At http://www.ieee.org/portal/site (Acessado em fevereiro de 2010).

[105] IEEE, **802.11 – 2007: revision of 1999.** In IEEE Standards Association. At http://standards.ieee.org/getieee802/download/802.11-2007.pdf. Acessado em Outubro de 2008.

[106] IEEE, **802.1x - Port-Based Network Access Control.** At http://standards.ieee.org/getieee802/download/802.1X-2004.pdf 2004, Acessado em 30 de outubro de 2009.

[107] IEEE, **802 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.** July, 2004.

[108] JANEWAY, C. A.; TRAVERS, P.; WALPORT, M. & CAPRA, J. D., **Immunobiology: The Immune System in Health and Disease.** 4th Ed., Garland Publishing. 1999.

[109] JANEWAY Jr., C. A., **The Immune System Evolved to Discriminate Infectious Nonself from Noninfectious Self.** Imm. Today, 13(1), pp. 11-16, 1993.

[110] JENNER, E., **An Inquiry Into the Causes and Effects of the Variolæ Vaccinæ, or Cow-Pox.** London, 1798. In King's College London Archives At http://www.aim25.ac.uk/cgi-bin/search2?coll_id=7135&inst_id=8

[111] _____., **Further Observations on the Variolæ Vaccinæ, or Cow-Pox.** London, 1799.

[112] _____., **A Continuation of Facts and Observations Relative to the Variolæ Vaccinæ, or Cow-Pox.** London, 1800.

[113] _____., **On the Origin of Vaccine Innoculation. London.** 1801.

[114] JENNINGS, N. R. and CAMPOS, J.R., **Towards a social level characterization of socially responsible agents.** IEE Proc. Software Engineering 144 (1), pp. 11–25, 1997.

[115] JENNINGS, N. R.; WOOLDRIDGE M. A., **Intelligent agents: Theory and practice.** The Knowledge Engineering Review, vol 10, no2, p.115-152, 1995.

[116] JERNE, N.K., **Towards a network theory of the immune system.** Ann. Immunol.(Paris) 125C: 373-389. 1974.

[117] JUNAID, M.; MUFTI, M. and UMAR ILYAS, M., **Vulnerabilities of IEEE 802.11i Wireless LAN CCMP Protocol.** In Transaction on Engineering, Computing and Technology V11, February, 2006. At: http://download.aircrack-ng.org/wiki-files/doc/Vulnerabilities%20of%20IEEE%20802.11i%20Wireless%20LAN%20CCMP%20Protocol.pdf, Acessado dia 11 de dezembro de 2009.

[118] KAHN, C.; PORRAS, P. A.; STANIFORD-CHEN, S. and TUNG, B., **A Common Intrusion Detection Framework.** Journal of Computer Security, July 1998.

[119] KANTHA, S. S. **The legacy of von Behring and Kitasato.** Immunology Today, Sept.1992, 13(9): 374.

[120] KEPHART, J., **A biologically inspired immune system for computers.** In Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems, páginas 130–139, 1994.

[121] KIM, J. e BENTLEY, P., **An artificial immune model for network intrusion detection.** Em Proceedings of the Seventh European Congress on Intelligent Techniques and Soft Computing, 1999.

[122] KIM, J.; WILSON, W.; AICKELIN, U. and MCLEOD, J., **Cooperative automated worm response and detection immune algorithm (CARDINAL) inspired by t-cell immunity and tolerance.** In ICARIS-04, LNCS 3239, 2005.

[123] KIM, J.; GREENSMITH, J.; TWYCROSS, J. and AICKELIN, U., **Malicious Code Execution Detection and Response Immune System inpired by the Danger Theory.** In Adaptive and Resilient Computing Security Workshop (ARCS-05), Santa Fe, USA, 2005.

[124] KLEIN, J., **Immunology.** Blackwell Scientific Publications, 1990.

[125] KOREK, **Chopchop Theory.** At. http://www.aircrack-ng.org/doku.php?id=chopchoptheory, (Acessado em Janeiro de 2010).

[126] KOREK, **Chopchop Attack.** At http://www.netstumbler.org/f49/need-security-pointers-11869/ 2004. Acessado em 30 de outubro de 2009.

[127] KRUEGEL, C.; VALEUR, F. and VIGNA, G., **Intrusion Detection and Correlation: Challenges and Solutions.** University Of California, Santa Barbara, USA, Springer-Verlag. Vol. 14, 2004.

[128] KLOMP, J. M., **Security Problems for small companies.** In SANS Institute InfoSec Reading Room, 2001.

[129] KUROSE, J. F.; ROSS, K. W., **Redes de Computadores e a Internet: uma abordagem top-down.** Pearson Addison Wesley, São Paulo, 2006.

[130] LEHTINEN, R., **Computer Security Basics.** 2nd Edition, O´Reilly. 310 p. June, 2006.

[131] LI, X.; FU, H.; HUANG, S., **Design of a Dendritic Cells Inspired Model Based on Danger Theory for Intrusion Detection System.** In IEEE Asian-Pacific Services Computing Conference (APSCC 2007), 2007.

[132] LINHARES, A. G. e GONÇALVES, P. A. da S., **Uma Análise dos Mecanismos de Segurança de Redes IEEE 802.11: WEP, WPA, WPA2 e IEEE 802.11w.** Em I Jornada Científica da UNIBRATEC, 2006. Acesso em 28 de outubro de 2009.

[133] LUTZ, M. and SCHULER, G., **Immature, semi-mature, and fully mature dendritics cells: which signals induce tolerance or immunity?** In Trends in immunology, 23(9):9911045, 2002

[134] MAES, P., **Modeling Adaptive Autonomous Agents. Artificial Life, an Overview.** C. Langton (ed.). MIT Press. Cambridge MA. (1995).

[135] MATETI, P., **Hacking Techniques in Wireless Networks.** In Department Of Computer Science and Engineering, Dayton, Ohio. 2005. At: http://www.cs.wright.edu/~pmateti/InternetSecurity/Lectures/WirelessHacks/Mateti-WirelessHacks.htm#_Toc77524668 Acessado dia 12 de outubro de 2009.

[136] MATZINGER, P., **Tolerance, danger, and the extended family.** Annu Rev Immunol 12: 991–1045, 1994.

[137] _____., **The real function of the immune system or tolerance and the four d's (danger, death, destruction and distress).** At:

http://cmmg.biosci.wayne.edu/asg/polly.html, 1996. Acessado em 15 de setembro de 2009.

[138] _____., **The Danger Model: A Renewed Sense of Self.** Science, 296, 301-305, 2002.

[139] MICHELAN, R., **Evolução de redes imunológicas para coordenação automática de comportamentos elementares em navegação autônoma de robôs.** FEEC - Universidade Estadual de Campinas – UNICAMP, Dissertação de Mestrado, Campinas, SP: [s.n.], 2003.

[140] MILLER, S. S., **Wi-Fi Security, Enhance security and maintain privacy of mission-critical data, even when going wireless.** McGraw-Hill Companies, Inc. USA, 340 p. 2003.

[141] MOEN, V.; RADDUM, H. and HOLE, K. J., **Weaknesses in the Temporal Key Hash of WPA.** ACM SIGMOBILE Mobile Computing and Communications Review, vol.8, pp.76–83, 2004.

[142] MOSKOWITZ, R., **Weakness in Passphrase Choice in WPA Interface.** November, 4, 2003. At: http://wifinetnews.com/archives/2003/11/weaknessin_passphrase_choice_in_wpa_interface.html Acessado dia 01 de novembro de 2009.

[143] MOSMMAN, T. R. and LIVINGSTONE, A. M., **Dendritic Cells: the immune information management experts.** Nature Immunology, pp. 564-566, 2004.

[144] MUNOZ, F. F.; NINO, L. F.; CLARIBED, C. Q., **An Artificial Immune System Based on the Innate Immune Response for a Multi-agent Object Transportation.** In Proceedings of the IEEE 3 rd International Conference on Intelligent System and Knowledge Engineering (ICISKE 2008), 2008.

[145] NAKAMURA, E. T. and LIMA, M. B., **Rogue Access Point, um Grande Risco para WLAN. 2003.** Em http://www.las.ic.unicamp.br/srac/imagens/SSI2003-RogueAP.pdf. Acessado dia 25 de novembro de 2009.

[146] NEISSER, U. et .al., **Intelligence: Knowns and Unknowns.** In American Psychologist Association, Inc. Vol. 5, No. 2, 77-101, February, 2006.

[147] NETO, R. M., **A Evolução dos Mecanismos de Segurança para Redes sem fio 802.11.** Monografia, Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, novembro, 2004. Em http://www-di.inf.puc-rio.br/~endler/courses/Mobile/Monografias/04/Miyano-Mono.pdf Acessado em 04 de novembro de 2009.

[148] NIKITA, B.; GOLDBERG, I. and WAGNER, D., **Intercepting Mobile Communications: the insecurity of 802.11.** In Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, pp. 180-189. 2001.

[149] NIST, **Specification for the Advanced Encryption Standard (AES).** FIPS 197, U.S. National Institute of Standards and Technology. November, 26, 2001. At. http://www.nist.gov/aes. Acessado dia 05 de novembro de 2009.

[150] NORTHCUTT, S. and NOVAK, J., **Network Intrusion Detection.** Third Edition, New Riders Publishing, USA, 456 p. 2003.

[151] NOSSAL, G. J. V., **Life, Death and the Immune System.** Scientific American, 269 (3), pp. 21-30, 1993.

[152] _____, **The Molecular and Cellular Basis of Affinity Maturation in the Antibody Response.** Cell, 68, pp 1-2, 1993.

[153] _____, **Negative Selection for Lymphocytes.** Cell, 76, pp 229-239, 1994.

[154] NWANA, H. S., **Software Agents: An Overview.** In: Knowledge Engineering Review. [s.n.], v. 11, p. 205–244, 1994. In: http://agents.umbc.edu/introduction/ao/, Acessado em 11 de setembro de 2009.

[155] OECHSLIN, P., **Making a Faster Cryptanalytic Time-Memory Trade-O.** In Ecole Polytechnique F´ed´erale de Lausanne, 2003. At: http://lasecwww.epfl.ch/~oechslin/publications/crypto03.pdf . Acessado dia 14 de dezembro de 2009.

[156] OHIGASHI, T. and MORRI, M., **A Practical Message Falsification Attackon WPA.** 2009. At: http://jwis2009.nsysu.edu.tw/location/paper/A%20Practical%20Message%20Falsification%20Attack%20on%20WPA.pdf. Acessado em 07 de dezembro de 2009.

[157] O'HARE, G. J. M. and JENNINGS, N. R., **Foundations of Distributed Artificial Intelligence.** New York: John Wiley & Sons, p. xii-ix, 1996.

[158] OMG, **Agent Technology.** Green Paper. In Agent Platform Special Interest Group, OMG Document agent/00-09-01, Version 1.0. September, 2000.

[159] PAPADIMITRIOU, G. A.; POMPORTSIS, A. S.; NICOPOLITIDIS, P. and OBAIDAT, M. S., **Wireless Networks.** John Wiley & Sons, LTD, Chichester, England, 2003.

[160] PERELSON, A. S., MIRMIRANI, M. & OSTER, G. F., **Optimal Strategies in Immunology II. B Memory Cell Production.** J. Math. Biol., 5, pp. 213-256, 1978.

[161] RACHEDI, A. and BENSLIMANE, A., **Impacts and solutions of control packets vulnerabilities with IEEE 802.11 MAC.** In Wireless Communications and Mobile Computing, John Wiley & Sons, Ltd. pp.469-488, 2008.

[162] RAMACHANDRAN, V. and AHMAD, Md S., **Cafe Latte with a Free Topping of Cracked WEP: Retrieving WEP Keys From Road-Warriors.** In 9th Toorcon Hacker´s Conference. October 19th-21st, 2007.

[163] RC4. **Conference Papers.** In Weizmann Institute of Science. At http://www.wisdom.weizmann.ac.il/~itsik/RC4/rc4.html Acesso em 02 de novembro de 2009.

[164] REIS, M. A., **Forense computacional e sua aplicação em segurança imunológica.** Dissertação de Mestrado. IMECC - Universidade Estadual de Campinas, 2003.

[165] RENSBERGER, B., **In Self-Defense.** Life Itself, Oxford University Press, pp. 212-228, 1996.

[166] RFC 2865. **Remote Authentication Dial In User Service (RADIUS).** At http://www.ietf.org/rfc/rfc2865.txt Acesso em 30 de outubro de 2009.

[167] RFC 2866, **Radius Accounting.** At http://www.ietf.org/rfc/rfc2866.txt Acesso em 30 de outubro de 2009.

[168] RFC 3748, **Extensible Authentication Protocol (EAP).** At http://www.ietf.org/rfc/rfc3748.txt. 2004. Acesso em 30 de outubro de 2009.

[169] RICHTER, P. H., **A Network Theory of the Immune System.** Eur. J. Imm., 5, pp. 350-354. 1975.

[170] _____., **The Network Idea and the Immune Response.** Theoretical Immunology, G. I. Bell, A. S. Perelson & G. H. Pimbley Jr. (Eds.), Marcel Dekker Inc., pp. 539-569. 1978.

[171] RIVEST, R. L., **The RC4 Encryption Algorithm.** RSA Data Security, Inc., Mar. 12, 1992. (Proprietary).

[172] RUSSEL, S. and NORVIG, P., **Artificial Intelligence: a modern approach.** Prentice Hall, 1995.

[173] SCHMALSTIEG, F. C. and GOLDMAN A. S., **Ilya Ilich Metchnikoff (1845-1915) and Paul Ehrlich (1854-1915): the centennial of the 1908 Nobel Prize in Physiology or Medicine.** In Journal of Medical Biography, pp. 96–103, England, 16 May, 2008.

[174] SCROFERNEKER, M. L. & POHLMANN, P. R., **Imunologia Básica e Aplicada.** Sagra Luzzatto, 1998.

[175] SEGEL, L. A. and PERELSON, A. S., **Computations in shape space: A new approach to immune network theory.** In Theoretical Immunology, Part Two, SFI Studies in the Sciences of Complexity, A. S. Perelson, Addison-Wesley, Reading, MA, pp. 321-343, 1988.

[176] SHAH, M., **Instructive Immunology: interplay between the innate and the adaptive immune system.** In Indian J Allergy Asthma Immunol, pp. 87-92, 2004.

[177] SCHWARTZ, R. S.; BANCHEREAU, J.; DODET, B. and TRANNOY, E., **Immune Tolerance.** Elsevier Science, Paperback. pp. 211-218. December, 1996.

[178] SICHMAN, J.; DEMAZEAU, Y. and BOISSIER. O., **When can Knowledge-based Systems be Called Agents.** Em: IX Simpósio Brasileiro de Inteligência Artificial SBC, Rio de Janeiro, Brasil, 1992.

[179] SILVEIRA, K. H., **Desafios para os Sistemas de Detecção de Intrusos (IDS).** Rede Nacional de Ensino e Pesquisa, 2000. Disponível em:<http://www.rnp.br/newsgen/0011/ids.html>. Acessado em: 11 de novembro de 2009.

[180] SILVERSTEIN, A. M., **History of Immunology.** Cellular Immun., 91, pp. 263-283, 1985.

[181] SOMAYAJI, A.; HOFMEYR, S. A.; e FORREST, S., **Principles of a computer immune system.** Em Proceedings of the 1997. New Security Paradigms Workshop, pp. 75–82, Langdale, Cumbria UK, ACM Press. 1997.

[182] SPRENT, J., **T and B Memory Cells.** Cell, 76, pp. 315-322, 1994.

[183] STEWART, J. & VARELA, F. J., **Morphogenesis in Shape-space. Elementary Metadynamics in a model of the Immune Network.** J. theor. Biol., 153, pp. 477-498. 1991.

[184] STUBBLEFIELD, A.; IOANNIDIS, J. and RUBIN, A. D., **Using the Fluhrer, Mantin, and Shamir Attack to break WEP.** In AT&T Labs Technical Report TD-4ZCPZZ. August 6, 2001. At http://www.securitytechnet.com/resource/hot-topic/wlan/wep_attack.pdf Acessado dia 30 de outubro de 2009.

[185] SUTTON, R. S. and BARTO, A. G., **Reinforcement Learning an Introduction.** In A Bradford Book. 342 pp. March, 1998.

[186] SUZUKI, J. & YAMAMOTO, Y., **iNet: A Configurable Framework for Simulating Immune Network.** In Proc. do IEEE SMC'00, pp. 119-124. 2000.

[187] SYCARA, K. P., **Multiagent Systems.** In AI Magazine Volume 19, No 2, pp. 79-92, Summer, 1998.

[188] TANENBAUM, A. S., **Computer Networks.** Prentice Hall, Inc. NJ, USA 2003.

[189] TARAKANOV, A, O., **Immunocomputing for Intelligent Intrusion Detection.** In IEEE Computational Intelligence Magazine, pp. 22-30. May 2008.

[190] TAUBER, A. I., **Historical and Philosophical Perspectives on Immune Cognition.** Journal of the History of Biology 30, pp. 419-440, 1997.

[191] TEDESCO, G.; TWYCROSS, J. and AICKELIN, U., **Integrating Innate and Adaptive Immunity**. In Intrusion Detection. Proceedings of the 5th International Conference on Artificial Immune Systems (ICARIS2006), Lecture Notes in Computer Science 4163, 2006.

[192] TEWS, E. WEINMANN, R. P. and PYSHKIN, A., **Breaking 104 bit WEP in less than 60 seconds.** In Cryptology ePrint Archive: Report 2007/120 At http://www.cdc.informatik.tu-darmstadt.de/aircrack-ptw/ 2007. Accesso em 01 de novembro de 2009.

[193] TIMMIS, J., **Artificial Immune Systems: A Novel Data Analysis Technique Inspired by the Immune Network Theory**. Tese de Doutorado, Department of Computer Science, University of Whales, September, 2000.

[194] TIZARD, I. R., **Immunology an Introduction.** Saunders College Publishing, 4th Ed. 1995.

[195] TURING, A. M., **Computing machinery and intelligence.** Mind, 59, pp. 433-460. 1950.

[196] TWYCROSS, J. and AICKELIN, U., **Towards a conceptual framework for innate immunity.** In International Conference of Artificial Immune Systems (ICARIS05), LNCS 3627, 2005.

[197] _____., **Libtissue – implementing innate immunity.** In Congress of Evolutionary Computation (CEC-2006), pp. 499-506, 2006.

[198] TYRREL, A., **Computer Now thy Self: a biological way to look at fault tolerance.** In Proceedings of the 2nd Euromicro/IEEE workshop on Dependable Computing Systems, pp. 129-135, Milan, 1999.

[199] VARELA, F. J., COUTINHO, A. DUPIRE, E. & VAZ, N. N., **Cognitive Networks: Immune, Neural and Otherwise.** Theoretical Immunology, Second Part, A. S. Perelson (Ed.), p. 359-375, 1988.

[200] VARELA, F. J. & COUTINHO, A., **Second Generation Immune Networks.** Immunology, Today, 12(5), pp. 159-166. 1991.

[201] VERSIGNASSI, A. e AXT, B., **Os Donos do Mundo.** Em Revista Superinteressante, Editora Abril, pp. 52-59, Edição 268, Agosto, 2009.

[202] ZHANG, R.; QIAN, D.; BAO, C.; WU, W.; GUO, X., **Multi-agent based intrusion detection architecture.** In Proceedings of the 2001 IEEE International Conference on Computer Networks and Mobile Computing (ICCNMC'01), pp. 494-501, 2001.

[203] ZHU, X.; HUANG, Z. and ZHOU, H., **Design of a Multi-agent Basied Intelligent Intrusion Detection System.** In IEEE International Symposium on Pervasive Computing and Applications. pp. 290-295, 2006.

[204] ZINKERNAGEL, R. M. & KELLY, J., **How Antigen Influences Immunity.** The Immunologist, 4/5, pp. 114-120, 1998.

[205] WALKER, J. R., **Unsafe at any key size: an analysis of the WEP encapsulation.** IEEE Document 802.11-00/362, October 2000. At http://md.hudora.de/archiv/wireless/unsafew.pdf. Acessado em 25 de outubro de 2009.

[206] WANG, J., **Computer Network Security: theory and practice.** Springer Berlin Heidelberg, pp. 19, p. 400, 2009.

[207] WI-FI ALLIANCE., **Wi-Fi Protected Access: Strong, standards-based, interoperable security for today's Wi-Fi networks.** 2003.

[208] WIKEPEDIA. **Sistema Multiagente.** Disponível em: http://pt.wikipedia.org/wiki/Sistema_multiagente Acessado em 15 de setembro de 2009.

[209] WIKEPEDIA. **Cognição.** Disponível em: http://pt.wikipedia.org/wiki/Cogni%C3%A7%C3%A3o Acessado em 15 de setembro de 2009.

[210] WILLIAMS, C. A.; HARRY, R. A. and MCLEOD, J. D., **Apoptotic cells induce dendritic cell-mediated suppression via interferon-γ-induced IDO.** Journal of Immunology, V. 124(1): 89–101, 2008.

[211] WIRESHARK, **Network Analyzer Protocol.** At: http://www.wireshark.org, acessado dia 15 de janeiro de 2010.

[212] WNDW. **Wireless Networking in the Developing World: Second Edition.** Hacker Friendly LLC, 2007.

[213] WOOLDRIDGE, M., **Agent-based Software Engineering.** In: IEEE Proceedings on Software Engineering, Vol. 144(1), pp. 26-37, 1997.

Apêndice A

Produção Científica

Dois trabalhos foram produzidos durante a concepção e desenvolvimento desta pesquisa. O primeiro foi publicado em anais de congresso e disponibilizado no Portal ACM/IEEE Computer Society. O segundo foi submetido e aguarda aceitação. A seguir são apresentadas informações sobre os trabalhos.

1) **Evento:** IEEE 9th International Conference on Intelligent Systems Design and Applications, ISDA 2009;
   Título: Danger Theory and Multi-agents Applied for Addressing the Deny of Service Detection Problem in IEEE 802.11 Networks;
   Local: Pisa, Itália;
   Data: 30 de novembro – 02 de dezembro;
   Páginas: 695-702
   DOI: 10.1109/ISDA.2009.136

Abstract

Deny of service (DoS) detection problem is a common and annoying network difficulty, but for IEEE 802.11 standards it becomes even more troublesome. Addressing this issue, we introduce a new approach to promptly warn the user. The detection algorithm put forward, combines second generation of Artificial Immune Systems, Danger Theory and Multi-Agent System. For the detection system, we used the dendritic cells algorithm, modified to IEEE 802.11 environments. Experimental results carried out in controlled setups have shown that the model can easily and effectively be applied for detecting DoS in IEEE 802.11 networks.

2) **Evento:** 2010 IEEE World Congress on Computational Intelligence (WCCI);
   Título: Detecting Problems in IEEE 802.11 Networks with Hybrid IDS Based on Danger Theory, Naive Bayes and Multi-agent System;
   Local: Barcelona, Espanha;
   Data: 18 e 19 de Julho de 2010;
   Páginas: 8 páginas;

Abstract

Many problems with wireless networks are directly related to the very means used to transport data, in this case radio waves. In addition to that lack of adaptable algorithms and mis-configured equipment make wireless networks a major target for attacks. New tools to refrain that are greatly in need, but due to the fact that it is easy to attack and tough to defend wireless networks, good candidate tools would be the ones that could profit from intelligent techniques. In this paper, we use the Danger Theory (DT) and a Bayesian classifier (using naive Bayes) embedded in a military style multi-agent system (MAS) to create a light, adaptable and dynamic detection system for wireless networks (WIDS). Experimental results show that the artificial immune aspect of the system is capable of detecting unknown issues and to identify them automatically with considerable few false alarms and low cost for the network traffic.

<Página deixada em branco deliberadamente>

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo