UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE ELETRICIDADE
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE

ALINE LOPES DA SILVA

MODELO DE IDS PARA USUÁRIOS DE DISPOSITIVOS MÓVEIS

São Luís
2008

ALINE LOPES DA SILVA

MODELO DE IDS PARA USUÁRIOS DE DISPOSITIVOS MÓVEIS

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão para obtenção do título de Mestre em Engenharia de Eletricidade, área de Concentração: Ciência da Computação.

Orientador: Ph.D. Zair Abdelouahab

São Luís
2008

Silva, Aline Lopes de.
Modelo de IDS para Usuários de Dispositivos Móveis / Aline Lopes da Silva. – São Luis, 2008.
115 f.

Orientador: «F5».
Impresso por computador (fotocópia).
Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia de Eletricidade, Universidade Federal do Maranhão, 2008.

1. Segurança 2. Detecção de Intrusão 3. Dispositivos Móveis 4. Redes sem Fio I. Título.

CDU «F11»

MODELO IDS PARA USUÁRIOS DE DISPOSITIVOS MOVÉIS

Aline Lopes da Silva

Dissertação aprovada em 26 de junho de 2008.

Prof. Zair Abdelouahab, Ph.D.
(Orientador)

Profa Daniela Barreiro Claro, Dra.
(Membro da Banca Examinadora)

Prof. Denivaldo Cicero Pavão Lopes, Dr.
(Membro da Banca Examinadora)

Prof Francisco José da Silva e Silva, Dr.
(Membro da Banca Examinadora)

A Deus e a minha família.

AGRADECIMENTOS

A Deus por tudo.

Aos meus pais, Francisca Vilma e Alberto Compasso, e as minhas irmãs, Aliane Lopes e Ariane Lopes.

Ao meu orientador, Prof. Ph.D. Zair Abdelouahab, pelo apoio e confiança em todos os momentos.

Ao Prof. Dr. Denivaldo Lopes, pelos direcionamentos fundamentais nos momentos de indecisão.

Às minhas amigas, Helaine Sousa e Renata Cantanhêde, pelas palavras de incentivos e ajuda em todos os momentos.

Ao meu amigo Irlandino Almeida pelo apoio e incentivo, mesmo estando distante.

Aos meus amigos Falkner Moraes, Flávio Ramos, Geysa Chaves, Johnneth Fonsêca, Osvaldo Júnior e Ricardo Ataíde que contribuíram para realização deste trabalho.

A minha sempre amiga e professora Msc. Maria Auxiliadora Freire.

A todos os que contribuíram ou incentivaram para a realização deste trabalho.

“Qualquer coisa que a mente do homem pode conceber, pode também, alcançar.”

William Clement Stone

RESUMO

Os dispositivos móveis são uma realidade cada vez mais comum em redes *wireless* e se integraram ao ambiente *wireless,* contribuindo para facilidade e disponibilidade da informação. Entretanto, o ambiente *wireless* está sujeito a vulnerabilidades, devido à forma de propagação da informação que se dá através do ar, estando sujeito a intercepção ou até mesmo roubo das informações. Dispositivos móveis além de estarem sujeitos a essas vulnerabilidades comuns em ambientes *wireless*, são dispositivos com algumas limitações físicas, como pouca capacidade de processamento e memória, além da vida útil de bateria limitada. Estas limitações tornam-se críticas neste tipo ambiente, quando ameaças não identificadas são direcionadas a dispositivos móveis. Torna-se necessário a implementação de sistema de detecção de intrusão voltado para estes dispositivos a fim de identificar comportamentos intrusivos, levando em consideração suas limitações físicas. Este trabalho propõe um sistema de detecção de intrusão (IDS, *Intrusion Detection System*) em redes *wireless* destinados a dispositivos móveis como adaptação e extensão do IDS-NIDIA (*Intrusion Detection System-Network Intrusion Detection System based on Intelligent Agents*). O mecanismo utiliza dois processos: o primeiro faz o monitoramento de informações sobre o comportamento do dispositivo e o segundo através do monitoramento de tráfego da rede *wireless,* analisando o tráfego destinado e originado aos dispositivos monitorados. A implementação da arquitetura e os testes realizados demonstram a viabilidade da solução.

Palavras-chave: Segurança. Detecção de Intrusão. Dispositivos Móveis. Redes sem Fio.

ABSTRACT

Mobile devices are increasing common reality in wireless networks and have integrated the wireless environment, helping to ease and to make available information. Meanwhile, the wireless environment is subject to vulnerabilities because of the way of spreading information that is given through the air, and is subject to interception or even information theft. Mobile Devices in addition of its vulnerability to these vulnerabilities common in wireless environments, are devices with some physical limitations such as lack of processing capacity and memory, beyond the limited battery life. These limitations become critical in this kind of environment, when unidentified threats attack are directed mobile devices. It is necessary to develop an intrusion detection system dedicated to these devices to identify intrusive behaviour, taking into account their physical limitations. This work proposes an intrusion detection system (IDS, Intrusion Detection System) for wireless networks and mobile devices. This is an adaptation and extension of NIDIA-IDS (Intrusion Detection System-Network Intrusion Detection System based on Intelligent Agents). The system acts with two processes: the first one is an information tracking on the device performance and the second one is a wireless network traffic monitoring, analyzing both the traffic of monitored devices. As proof of concepts a prototype was developed and some experiments were carried to validate this solution.

Keyworks: Security. Intrusion Detection. Mobile Devices. Wireless Network.

LISTA DE FIGURAS

p.

LISTA DE TABELAS

p.

LISTA DE CÓDIGO

p.

LISTA DE SIGLAS

ACK - Acknowledgement

AP - Access Point

BBID - Battery Based Intrusion Detection

BSIPS - Battery Sensing Intrusion Protection System

BSD - Berkeley Software Distribution

BSS - Basic Service Sets

CIDE - Correlation Intrusion Detection Engine

CIDF - Common Intrusion Detection Framework

DDoS - Distributed Denial of Service

DFDB - Standard of Intruders and Intrusion Data Base

DHCP - Dynamic Host Configuration Protocol

DoS - Denial of Service

DSSS - Direct Sequence Spread Spectrum

DTC - Dynamic Threshold Calculation

EUA - Estados Unidos da América

ESS - Extended Service Set

FFT - Fast Fourier Transform

FHSS - Frequency Hopping Spread Spectrum

FIN - No More Data from Sender

FTP – File Transfer Protocol

GPL - General Public License

GPS - Global Positioning System

HASTE - Host Analyze Signature Trace Engine

HIDE - Host Intrusion Detection Engine

IBSS - Independent Basic Service Set

ICMP - Internet Control Message Protocol

IDE - Integrated Development Environment

IDPS - Intrusion Detection Prevention System

IDS - Intrusion Detection System

IEEE - Institute of Electrical and Electronics Engineers

IIDB - Incidents of Intrusion and Forensic Information DataBase

IP - Internet Protocol

IPS - Intrusion Prevention System

JDOM - Java Document Object Model

LANs - Local Area Networks

MAC - Media Access Control

MB – MegaByte

MCA - Main Controller Agent

MiTM - Man-in-The Middle

MIMO – Multiple IN Multiple Out

MMDS - Multi-level Monitoring and Detection System

NIC - Network Interface Cards

NIDIA - Network Intrusion Detection System based on Intelligent Agents

OFDM - Orthogonal frequency-division multiplexing

OSI - Open Systems Interconnection

PC - Personal Computer

PDA - Personal Digital Assistants

PSH - Push Function

RADB - Reaction DataBase

RAM - Random Access Memory

RF - Radio Frequency

RFC - Request for Comments

RPM - Red Hat Package Manager

RST - Reset the Connection

SEA - Security Evaluation Agent

SMA - System Monitoring Agent

SNAP - Subnetwork Acess Protocol

SPIE - Scan Port Intruse Engine

SSIDs - Service Set IDentifier

STA - STAtion

STDB - Strategy DataBase

SUA - System Updating Agent

SYN - Synchronize

TCP - Transmission Control Protocol

UML - Unified Modeling Language

URG - Urgent Pointer

VPNs - Virtual Private Networks

WPA - Wireless Access Point

WEP - Wireless Equivalent Protocol

WIDE - Wireless Intrusion Detection Extensions

WIDS - Wireless Intrusion Detection System

Wi-fi - WIreless Fidelity

WLAN - Wireless Local Area Network

XML - eXtensible Markup Language

SUMÁRIO

1 INTRODUÇÃO

O crescente advento de redes sem fio IEEE[1] 802.11(ANSI/IEEE 802.11, 1999), também conhecidas como redes *Wi-fi* ou *wireless*, tem deixado usuários e organizações expostos a novos tipos de ameaças.

A mobilidade trouxe ao usuário um novo mundo, onde acessibilidade e disponibilidade da informação são permitidas a qualquer momento. Não há necessidade de conexão através de cabos para obter acesso à rede, somente se requer a disponibilidade do serviço e dispositivos dotados com placas de rede *wireless*.

A facilidade de acesso à informação em redes *wireless* é bastante vulnerável, devido à forma de propagação dos dados que se dá através do ar, difusamente. Esta característica das redes *wireless* permite a interceptação de dados para análise, alteração ou até mesmo roubo.

Ao contrário do ambiente de redes cabeadas, onde o atacante necessita obter acesso físico para realizar qualquer atividade maliciosa, os ataques em redes *wireless* podem partir de todas as direções e atingir qualquer um dos nós da rede.

Uma ampla gama de dispositivos usa a tecnologia *wireless*, sendo em dispositivos móveis a tecnologia predominante para comunicação através da rede. Os dispositivos móveis são considerados verdadeiros computadores de mão, são dotados de quase todas as funcionalidades existentes em estações normais. São utilizados pelos mais diversos setores entre os quais vale ressaltar: o estudantil, acadêmico, empresarial, governamental e até mesmo ambientes hospitalares.

Pesquisas revelam um aumento expressivo no uso destes equipamentos. Em 2011, serão mais de 82 milhões de dispositivos portáteis pelo mundo. A liderança do setor no último ano ficou com os EUA (Estados Unidos da América), com 42% (quarenta e dois por cento) do total de equipamentos fabricados (PC WORLD, 2007) .

Os dispositivos móveis são caracterizados principalmente por possuir capacidade de armazenamento, poder de processamento e carga de bateria como recursos extremamente limitados que podem esgotar-se com o uso intensivo do equipamento. Desta forma, quaisquer medidas de segurança que venham a ser desenvolvidas para estes tipos de equipamentos devem levar em consideração estas limitações.

---

[1] *Institute of Electrical and Electronics Engineers*

Políticas de segurança, planos de continuidade de negócios, controle de acessos (físicos e lógicos), a utilização de *firewalls*, antivírus, criptografia, sistemas detectores de intrusão são alguns elementos que uma vez implantados nas organizações podem prover maior grau de segurança. Cada uma dessas medidas traz consigo suas peculiaridades, o que deve ser levado em consideração quando da sua utilização (SILVA, 2006).

1.1 DESCRIÇÃO DO PROBLEMA

As redes *wireless* cresceram muito em popularidade. Elas são facilmente instaladas e configuradas. Permitem aos usuários mobilidade e acesso a informação de qualquer localização onde haja disponibilidade do serviço.

Um ambiente com facilidade e disponibilidade da informação foi proporcionado aos usuários, mas com razoável confiabilidade nos requisitos básicos de segurança: confidencialidade, disponibilidade e integridade da informação.

Dispositivos móveis dotados com recursos que permitam a comunicação dentro do ambiente de redes *wireless*, passaram a fazer parte deste ambiente compartilhando e acessando informações. Estes se tornaram rapidamente alvos de ataques dentro deste tipo de ambiente, da mesma forma em que ganharam popularidade pelas funcionalidades e facilidades proporcionadas na disponibilidade da informação. A possibilidade de ter informações pessoais sempre disponíveis em um dispositivo que acessa *Internet,* armazena contatos, agenda informações, permite leitura de arquivos e funciona como telefone foram os fatores que contribuíram para que estes equipamentos integrassem-se o cotidiano das pessoas.

Tanta facilidade e disponibilidade acarretaram em certas limitações. Os dispositivos em geral possuem dimensões reduzidas o que implica limitações computacionais. Limitações como baixo poder de processamento, armazenamento e surgimento de uma nova variável que também deveria ser considerada no ambiente para redes *wireless*, o consumo de bateria, alertava sobre as possíveis ameaças a que estes dispositivos poderiam ser submetidos.

Várias ameaças já foram identificadas em dispositivos móveis, na sua grande maioria ataques de negação de serviço, onde o usuário é impedido de ter acesso às informações contidas no dispositivo, uma vez que estes tipos de ataques têm como alvo principal exaurir a bateria destes dispositivos. (BUENNEMEYER, 2007)

Os IDS (*Intrusion Detection System*) são sensores de detecção de intrusão que, dispostos de forma inteligente na rede monitoram as violações de segurança ou utilizações indevidas originadas dentro ou fora da rede.

Diferenças estruturais e comportamentais existentes em redes *wireless* tornam as ferramentas existentes de sistemas detectores de intrusão voltados para redes cabeadas inaplicáveis para redes *wireless*.

As redes wireless têm como forma de propagação da informação o ar. Esta característica pode ser facilitador para que uma pessoa não autorizada possa ter acesso às informações que trafegam através da rede como aos dados armazenados nos *hosts* uma vez que não há necessidade conexão física entre os *hosts* da rede para acesso aos recursos do ambiente.

Outros fatores que impulsionaram uso de ferramentas de segurança específicas para redes wireless foram: limitações no consumo de largura de banda e o aumento de processamento computacional são recursos limitados em redes *wireless* que contribuíram para a necessidade de ferramentas específicas para este tipo de rede.

Assim, um sistema de IDS aplicado a dispositivos móveis se encaixaria junto às demais ferramentas de segurança para usuários finais.

1.2 METODOLOGIA

A metodologia empregada para o desenvolvimento deste trabalho utilizou inicialmente a pesquisa documental, com o intuito de coletar informações de livros, teses e dissertações, periódicos, anais de congressos e documentos hipermídia para uma total ambientação da literatura especializada.

A pesquisa literária baseou-se em um estudo mais aprofundado sobre segurança, redes w*ireless*, dispositivos móveis e desenvolvimento de aplicativos para este dispositivos.

Posteriormente, um estudo e utilização das seguintes ferramentas:

- Eclipse IDE (*Integrated Development Environment*) que se constitui em uma plataforma para a integração de ferramentas de desenvolvimento;
- bibliotecas para captura de tráfego para redes *wireless;*
- ferramentas para detecção de intrusão para redes *wireless*;
- bibliotecas para desenvolvimento de aplicativos para dispositivos móveis;
- ferramentas para geração de ataques.

Com base nestes estudos, uma ferramenta de detecção de intrusão para usuários de dispositivos móveis foi desenvolvida, adaptando e estendendo o NIDIA (*Network Intrusion Detection System based on Intelligent Agents*) (LIMA, 2001).

1.3 OBJETIVOS GERAIS E ESPECÍFICOS DA DISSERTAÇÃO

Neste trabalho, objetiva-se dispor de uma ferramenta de detecção de intrusão para o usuário final, no caso usuários de dispositivos móveis, que seria de grande ajuda e justificada perante o crescente número de ataques e invasões realizadas com sucesso e os impactos que elas podem causar.

Esta dissertação visa propor um modelo e uma implementação de um IDS para usuários de dispositivos móveis. Este modelo adapta e estende o NIDIA (LIMA, 2001) utilizando técnicas dos sistemas detectores de intrusão para identificação de possíveis ameaças. Desta forma, os usuários que usam estes dispositivos para acesso a informações da rede contam com um mecanismo de segurança adaptado a sua realidade, onde são respeitados os limites de hardware e software destes dispositivos.

Neste documento, apresenta-se a especificação dos componentes do modelo, como estes se relacionam, seu modelo de comunicação e sua troca de informação. A aplicabilidade do modelo é demonstrada através da implementação de seus componentes.

Dentro deste modelo, técnicas de detecção específica para estes tipos de dispositivos serão demonstradas, especificamente ataques de negação de serviço, onde o enfoque principal vem a ser o aumento da carga de processamento do dispositivo que implica em uma maior demanda por energia e, conseqüentemente, consumo de bateria. Desta forma, a exposição de dois ataques ao usuário final será apresentada, assim como a captura e as contramedidas tomadas pelo sistema implementado.

Esta dissertação tem como objetivos específicos:

a) apresentar os requisitos necessários para o desenvolvimento deste modelo;
b) apresentar um modelo de um IDS voltado para usuários de dispositivos móveis;
c) apresentar a especificação dos componentes que são necessários ao modelo;
d) apresentar a implementação dos componentes do modelo, demonstrando as operações realizadas pelos mesmos;
e) implementar um protótipo do modelo de IDS remoto estendendo o NIDIA;
f) demonstrar o funcionamento do sistema através da simulação de dois ataques ao usuário final, fazendo com que o sistema capture informações sobre os referidos ataques e tome as devidas contramedidas.

1.4 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação está divida em cinco capítulos. No primeiro Capítulo, uma visão geral sobre o trabalho proposto e os objetivos a serem atingidos.

O Capítulo 2 apresenta o referencial teórico da dissertação, expondo conceitos sobre as bases de nossa proposta. Conceitos sobre as redes sem fio e seus mecanismos de segurança, ferramentas de detecção de intrusos, dispositivos móveis, mecanismos de segurança voltados para os mesmos.

O Capítulo 3 contém uma apresentação da modelagem proposta através de diagramas de caso de uso, diagramas de classe, diagramas de seqüência, diagramas de atividades e diagramas de implantação. As características do sistema proposto em termos de seus componentes são apresentadas. A cooperação entre os componentes é discutida, assim como a comunicação e a troca de informações.

O Capítulo 4 apresenta a implementação do protótipo para o Modelo de IDS para usuário de dispositivos móveis que é uma extensão e adaptação do NIDIA. Apresenta-se: o desenvolvimento dos componentes que pertencem ao modelo, o funcionamento do sistema, utilização do sistema, a demonstração de dois ataques ao usuário final e as contramedidas tomadas pelo sistema.

O Capítulo 5 contém as considerações finais da dissertação, ressaltando as contribuições da pesquisa realizada, as limitações e também sugestões para trabalhos futuros.

2 FUNDAMENTOS E ESTADO DA ARTE

Neste capítulo, os conceitos fundamentais das tecnologias que servem de base para desenvolvimento desta proposta de dissertação são apresentados. Estes conceitos tiveram papel relevante no entendimento individual de cada tecnologia e na obtenção do conhecimento para melhor integrá-las.

2.1 REDES WIRELESS 802.11

As redes wireless possibilitam a dispositivos com interfaces *wireless* usarem recursos computacionais sem estarem fisicamente conectados a uma rede. Os dispositivos simplesmente precisam estar dentro do raio de cobertura (conhecido como extensão) da infra-estrutura da rede *wireless*. Uma WLAN (*Wireless Local Area Network*) consiste de um grupo de nós *wireless* dentro de uma área geográfica limitada, que são capazes de estabelecer comunicação via rádio (NIST, 2002).

WLANs são tipicamente usadas por dispositivos dentro de uma extensão claramente delimitada, como no interior de um edifício, e são geralmente implementadas como extensões às LANs (*Local Area Networks*) existentes para prover uma mobilidade aprimorada aos usuários.

Desde o princípio das redes wireless, vários padrões e tecnologias têm sido desenvolvidos para WLANs. A principal organização de padronização que aborda as redes *wireless* é o IEEE (IEEE, 2007), sendo que esses padrões e a arquitetura das redes 802.11 são descritos nas próximas seções.

2.2.1 Os padrões de redes Wireless 802.11

As tecnologias de WLANs tornaram-se disponíveis inicialmente em 1990, quando os fabricantes começaram a lançar no mercado produtos que operavam na banda de freqüência de 900 MHz. Estas soluções, que usavam projetos proprietários e não padronizados, permitiam taxas de transferência de aproximadamente 1 Mbps. Isto era significantemente mais lento que os 10 Mbps de velocidade permitidos pela maioria das LANs daquela época. Em 1992, os fabricantes começaram a lançar produtos que usavam a banda de 2.4 GHz. Embora tais produtos permitissem taxas de transferências de dados muito mais altas que os produtos da banda de 900 MHz, eles também usavam projetos proprietários.

A necessidade de interoperabilidade entre diferentes marcas de produtos para WLANs levou várias organizações ao desenvolvimento de padrões para redes wireless.

Em 1997, o IEEE aprovou o padrão 802.11 para WLANs. O padrão IEEE 802.11 usava a taxa de freqüência de 2.4 GHz e suportava velocidades de transmissão de 1 Mbps, utilizando a tecnologia de modulação FHSS (*Frequency Hopping Spread Spectrum*), e de 2 Mbps para DSSS (*Direct Sequence Spread Spectrum*).

Diferentes grupos do padrão IEEE 802.11 foram criados para revisão e criação do padrão 802.11. Assim, para cada nova criação ou revisão do padrão um novo sufixo era acrescido ao padrão IEEE 802.11.

Em 1999, o IEEE aprovou duas alterações para o padrão IEEE 802.11, o 802.11a e o 802.11b, que definem os métodos de transmissão via rádio a serem usados. O padrão 802.11a utiliza a freqüência de 5GHz e taxa para transmissão de dados 54 Mbps.

O padrão IEEE 802.11b transmite na freqüência 2.4 GHz, oferecendo taxas de dados de até 11 Mbps. O IEEE 802.11b foi criado para prover performance, *throughput*[2] e segurança comparáveis com as redes locais cabeadas. Com isso, os equipamentos para WLANs baseados no padrão 802.11b tornaram-se rapidamente a tecnologia *wireless* predominante.

Em 2003, o IEEE publicou a alteração IEEE 802.11g, que específica uma freqüência 2.4 GHz e pode suportar taxas de dados de até 54 Mbps. Convém destacar que os produtos IEEE 802.11g são perfeitamente compatíveis com os produtos 802.11b.

As aprovações de alterações mais recentes constam de novembro de 2005, quando o IEEE aprovou o padrão IEEE 802.11e, que provê aprimoramentos de qualidade de serviço ao IEEE 802.11, melhorando a transferência de conteúdo multimídia. O surgimento de um novo grupo previsto para junho 2009, o 802.11n, que específica melhorias para o 802.11 que possibilitarão a taxa de transmissão de dados de 300 Mbps na freqüência 5GHz ou 2.4GHz.

Um resumo das tecnologias para o padrão 802.11 é listado na Tabela 2.1.

2.2.2 Componentes e Modelos de Arquiteturas das Redes 802.11

A arquitetura do IEEE 802.11 tem dois componentes fundamentais:

---

[2] A quantidade de dados transferidos dentro da rede.

- STA (*Station*): é um dispositivo *wireless* que fica no extremo da arquitetura. Típicos exemplos de STAs são *laptops*, *PDAs (Personal Digital Assistants*), *smart fones* e outros dispositivos com interfaces 802.11;
- AP (*Access Point*): Conecta logicamente as STAs com um sistema de distribuição, o qual é normalmente a infra-estrutura cabeada de uma organização. Os APs podem também conectar STAs *wireless* entre si sem acessar um sistema de distribuição.

Tabela 2.1 - Tecnologias do padrão IEEE 802.11

Protocolo	Data Aprovação	Freqüência	*Throughput*	Taxa de Dados	Técnica de Modulação
802.11	1997	2.4 GHz	0.9 Mbit/s	2 Mbit/s	OFDM/DSSS
802.11a	1999	5 GHz	23 Mbit/s	54 Mbit/s	OFDM
802.11b	1999	2.4 GHz	4.3 Mbit/s	11 Mbit/s	DSSS
802.11g	2003	2.4 GHz	19 Mbit/s	54 Mbit/s	OFDM
802.11n	2009, Junho	2.4 GHz 5 GHz	74 Mbit/s	248 Mbit/s	MIMO

O padrão IEEE 802.11 também define duas estruturas de projeto ou configurações para WLANs:

- Modo *ad-hoc*: não usa APs. Apenas a comunicação ponto-a-ponto é realizada entre as STAs;
- Modo infra-estruturado: um AP conecta STAs entre si ou para um sistema de distribuição. É o modo mais comumente usado para *WLANs*.

A topologia do modo *ad-hoc* está representada conceitualmente na Figura 2.1.

Este modo de operação também conhecido como *peer-to-peer* (ponto-a-ponto), é possível quando duas ou mais STAs são capazes de se comunicar diretamente.

A Figura 2.1 mostra três dispositivos comunicando-se diretamente, sem qualquer infra-estrutura. Um conjunto de STAs configuradas no modo *ad-hoc* é conhecido como um IBSS (*Independent Basic Service Set*). O círculo na Figura 2.1 representa uma IBSS. É importante considerá-lo como a área de cobertura de rádio dentro da qual as estações podem permanecer em comunicação. Uma propriedade fundamental da IBSS é que ela não define roteamento ou encaminhamento de mensagens, requerendo que cada dispositivo esteja dentro da cobertura de rádio de todos os demais.

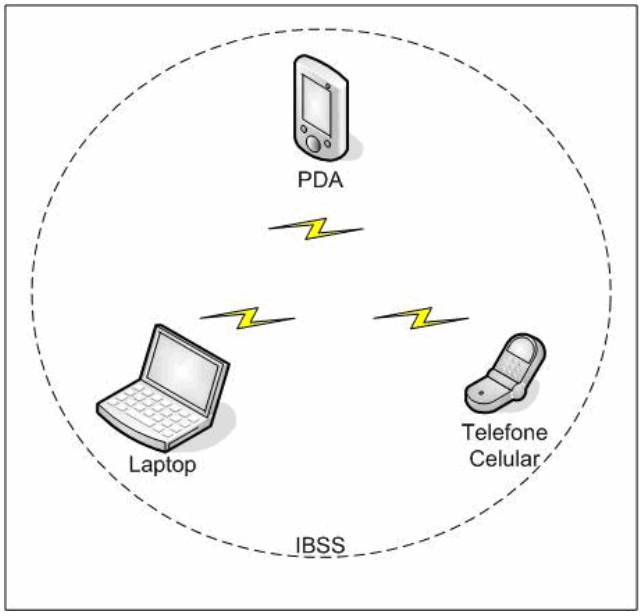


Figura 2.1 - Modo Ad-hoc

A principal vantagem de uma WLAN *ad-hoc* é que teoricamente ela pode ser criada a qualquer momento e em qualquer lugar, permitindo que os usuários criem conexões *wireless* de forma barata, rápida e fácil, com uma necessidade mínima de *hardware* de manutenção. Uma rede *ad-hoc* pode ser criada por vários motivos, tais como permitir o compartilhamento de arquivos ou a troca rápida de *e-mails*. Entretanto, uma WLAN *ad-hoc* não pode comunicar-se com redes externas. Outra complicação é que ela pode interferir na operação de um AP de uma rede infra-estruturada, que porventura exista dentro do mesmo espaço *wireless*.

No modo infra-estruturado, uma WLAN IEEE 802.11 compreende um ou mais BSSs (*Basic Service Sets*), que são os blocos básicos de construção de uma WLAN. Um BSS inclui um AP e uma ou mais STAs. O AP em um BSS conecta as STAs ao sistema de distribuição, que é a maneira pela qual as STAs podem comunicar-se com a rede cabeada da organização e com redes externas, como a *Internet*. O modo infra-estruturado IEEE 802.11 é representado na Figura 2.2.

O sistema de distribuição e o uso de múltiplos BSSs e seus APs associados permitem a criação de redes *wireless* de tamanho e complexidade arbitrários. Na especificação IEEE 802.11, este tipo de rede multi-BSS é referido como um ESS (*Extended Service Set*). (ATAÍDE, 2007).

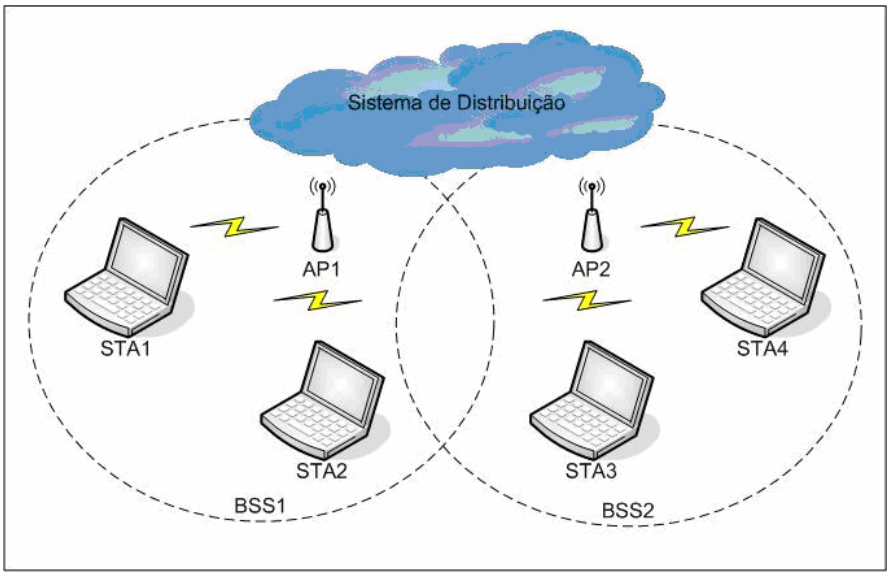


Figura 2.2 - Modo Infra-estruturado.

2.2 DISPOSITIVOS MÓVEIS

Atualmente dispositivos computacionais móveis estão disponíveis nas mais variadas formas como, por exemplo, *tablet PCs (Personal Computer)*, *PDAs*, *smartphones* e telefones celulares, entre outros.

O uso de dispositivos móveis abre novos campos para automatização e informatização. O sistema de informação deixa, por um lado, de estar confinado aos limites físicos dos edifícios, podendo o usuário continuar realizando suas atividades onde quer que ele esteja.

Tecnologias de computação móvel e de rede *wireless* têm evoluído muito rapidamente, de forma que muitos destes dispositivos possuem hoje considerável capacidade de processamento, armazenamento e comunicação.

As diversas tecnologias de rede *wireless* que existem atualmente, tais como sistemas celulares, WLANs e *Bluetooth*, permitem que seus usuários possam ter acesso a dados corporativos, pessoais e conteúdos da *Internet* de modo conveniente em qualquer lugar e a qualquer hora. Todas estas funcionalidades tem tornado estes dispositivos cada vez mais populares e utilizados por diversos grupos de pessoas para os mais variados fins (GOMES et al., 2007).

Estes dispositivos são utilizados nas mais diversas áreas que vão desde ambiente educacional, com grande ênfase no ambiente empresarial e cada vez ganha mais espaço nos ambientes hospitalares.

Fisicamente os dispositivos portáteis apresentam mais de 100 MB (*Megabyte*) de memória com suporte para cartões de expansão, processadores de com velocidades acima de 300 Mhz, tela colorida com alta resolução e sensível ao toque, pesam em média 150 g (cento e cinqüenta gramas) e possuem placa de rede sem fio para comunicação *Wi-fi (WIreless FIdelity*), *Bluetooth* e *InfraRed*.

Funcionalmente, eles permitem acesso à *Internet*, leitura e armazenamento de arquivos, armazenamento de dados pessoais, lista de contatos, armazenamento e visualização de fotos e vídeos. A Figura 2.3 apresenta alguns modelos PDA's.

Figura 2.3 - Modelos de dispositivos móveis

2.2.1 Considerações sobre as limitações de Dispositivos Móveis

Devido à reduzida capacidade computacional dos dispositivos móveis, se comparado aos computadores tradicionais, estes ficam sujeitos as limitações de processamento, armazenamento e tempo de vida útil de bateria.

Os dispositivos móveis são equipados com pouca memória RAM (*Random Access Memory*), processadores mais lentos, memória não volátil pequena, programas que têm uma demanda maior por processamento podem ter um impacto considerável sobre estes recursos. Um exemplo típico é criptografia que em geral é mais difícil de ser utilizada nestes

dispositivos, mas que já conta com trabalhos em fase evolução nesta área (TSB et al., 2007), (MALTORA et al., 2007). A capacidade limitada de processamento, característica dos pequenos dispositivos, também significa que a necessidade de recorrer à computação extra para realizar a criptografia pode diminuir consideravelmente a velocidade de processamento do dispositivo.

Estes também dependem da energia fornecida por bateria e esta tem vida útil limitada. A expectativa de vida da bateria desses dispositivos pode ser mensurada em horas ou semanas. Entretanto, o tempo de vida das baterias é relacionado com sua atividade, assim quanto maior a demanda por processamento, menor será a expectativa de vida das baterias. Isso deixa os dispositivos móveis expostos à escassez de recursos e ataques de exaustão. Esta forma de ataque consiste em reduzir o tempo de vida das baterias consumindo todo o poder da mesma deixando o dispositivo inapto para uso.

2.3 SEGURANÇA EM REDES WIRELESS

As WLANs precisam suportar vários objetivos de segurança. A intenção é que isso seja alcançado através de uma combinação de ferramentas de segurança embutidas no próprio padrão de redes *wireless*. Os objetivos mais comuns são:

- Confidencialidade: garante que os dados da comunicação não possam ser lidos por indivíduos não autorizados;
- Integridade: detecta qualquer mudança, intencional ou não, que possa ocorrer nos dados durante a transmissão;
- Disponibilidade: garante que dispositivos ou indivíduos possam acessar uma rede e seus recursos sempre que houver necessidade;
- Controle de Acesso: restringe o direito de dispositivos ou indivíduos em acessar uma rede ou recursos dentro de uma rede.

Os objetivos de segurança para redes *wireless* e redes cabeadas são os mesmos, assim como as maiores categorias de ameaças e ataques que elas enfrentam. Figura 2.4 provê uma taxonomia geral dos ataques em redes *wireless*, dividindo-os basicamente entre ataques ativos e ataques passivos (NIST, 2002).

Um ataque passivo é aquele no qual um indivíduo não autorizado obtém acesso à informação, mas não modifica seu conteúdo. Ataques passivos podem ser: escuta ou análise de tráfego, algumas vezes chamado de análise de fluxo de tráfego.

No ataque de escuta, o atacante fica monitorando passivamente as transmissões na rede, em busca do conteúdo das mensagens, incluindo credenciais de autenticação.

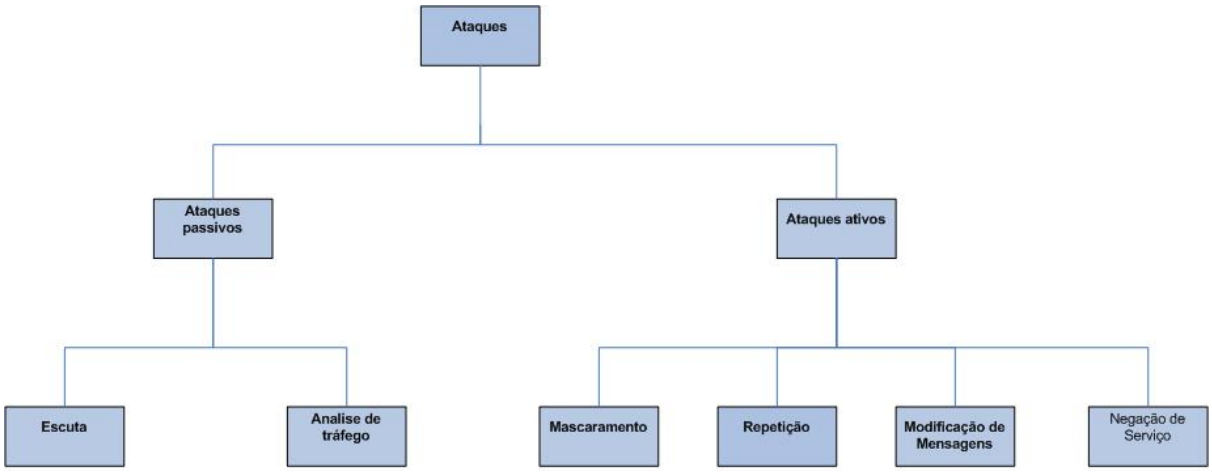


Figura 2.4 - Taxonomia de Ataques (NIST, 2002)

Na análise de tráfego, o atacante obtém informações confidenciais sobre a rede, monitorando passivamente as transmissões em busca de padrões de comunicação. Isso é possível porque uma quantidade considerável de informações sobre o sistema é contida nas próprias mensagens que trafegam na rede.

Um ataque ativo é aquele através do qual um indivíduo não autorizado realiza modificações em mensagens, fluxos de informações ou arquivos. É possível detectar esse tipo de ataque, mas pode não ser possível evitá-lo. Os ataques ativos podem tomar a forma de um dos quatro ataques (ou uma combinação deles): mascaramento, repetição, modificação de mensagens, negação de serviço.

No ataque de mascaramento, o atacante personifica um usuário autorizado para obter privilégios para os quais não possui autorização.

No ataque de repetição, o atacante fica monitorando passivamente as transmissões e retransmitindo mensagens, como se fosse o usuário legítimo.

No ataque de modificação de mensagens, o atacante altera mensagens legítimas excluindo-as, acrescentando conteúdo, modificando-as ou reordenando-as.

Nos ataques de negação de serviço, DoS (*Denial of Service*), o atacante normalmente não rouba informações. Ele simplesmente impede que os usuários acessem os serviços de rede, fazendo com que esses serviços sejam interrompidos ou atrasados. Suas conseqüências podem se estender desde uma redução moderada na performance, até a falha total do sistema. Existe uma variedade de tipos de ataques de DoS possíveis. Alguns dos mais importantes são: rogue APs, MiTM (*Man-in-The Middle*) e roubo de sessão (ATAÍDE, 2007).

Um tipo particular de ataque de DoS são os APs não autorizados (rogue APs), que são APs instalados sem o conhecimento ou autorização por parte do administrador da rede. Esse ataque pode ser de dois tipos: interno e externo. No tipo interno, algum usuário da rede instala um AP sem habilitar qualquer mecanismo de segurança para controle de acesso à rede, permitindo que qualquer indivíduo com um dispositivo com interface de rede 802.11 possa conectar-se na rede corporativa. Isso permite com que a rede tenha o acesso disponível para eventuais indivíduos mal intencionados, aproveitando-se desta falha se segurança. No tipo externo, um atacante instala na área externa a organização um AP conectado a uma rede falsa, porém com as mesmas configurações de um AP autêntico e da rede interna.

Desse modo, os atacantes fazem com que os usuários se conectem ao falso AP, na ilusão de estar usando os serviços da rede verdadeira.

Outro tipo de ataque de DoS é o ataque MiTM, no qual o atacante é capaz de ler, inserir e modificar mensagens no caminho da comunicação entre duas partes legítimas da rede, sem que nenhuma delas saiba que a comunicação entre elas está comprometida. No contexto de uma WLAN, o ataque MiTM pode ser realizado através de um falso AP ou uma estação falsa que se parecem com *hosts* legítimos da rede para obter informações sobre a rede dos usuários conectados a ela.

Outro tipo de ataque de DoS é o roubo de sessão (*session hijacking*), no qual o atacante espera até que um cliente tenha se autenticado na rede, para então enviar-lhe uma mensagem de desautenticação usando o endereço MAC (*Media Access Control*) do AP verdadeiro, como se o AP verdadeiro estivesse desautenticando o cliente. Feito isto, o atacante pode começar a transmitir quadros na rede, usando o endereço MAC do cliente que foi desconectado, roubando efetivamente a sua sessão. Na próxima reautenticação, o atacante não poderá ser reautenticado e será banido, necessitando roubar outra sessão válida.

A maioria dos ataques contra WLANs envolve um atacante com acesso ao enlace de rádio entre uma STA e um AP ou entre duas STAs. Vários dos ataques da Figura 2.4 contam com a habilidade do atacante em interceptar e injetar tráfego na rede. Isto evidencia a diferença mais significante entre proteger redes *wireless* e proteger redes cabeadas: a relativa facilidade de interceptar o tráfego da rede o que nas redes cabeadas o atacante precisa estar conectado ao sistema remotamente. Em uma rede *wireless*, o atacante simplesmente precisa estar dentro da extensão da infra-estrutura *wireless*. Além disso, ele pode usar antenas direcionais altamente sensíveis, as quais aumentam significativamente a extensão efetiva da rede local *wireless*, para uma cobertura muito além do padrão.

2.3.1 Ataques em Dispositivos Móveis

Como citado na seção anterior,há uma variedade de ataques que comprometem a comunicação em ambiente *wireless*. Quando se trata de dispositivos móveis, os ataques voltados para esta categoria possuem a mesma finalidade. Estes tendem comprometer a disponibilidade do serviço ao usuário ou até mesmo roubar as informações contidas no mesmo, que na maioria das situações tendem a ser pessoais e confidenciais.

Redes *wireless* e dispositivos móveis aumentam o número de pontos de exposição para ataques. Assim, os atacantes que se especializam em interromper e roubar a comunicação destes dispositivos está conquistando um número cada vez de maior de adeptos para esta atividade.

A fraca performance dos dispositivos móveis é atualmente a indicação de que os ataques são dirigidos à bateria dos mesmos ou de alguma forma acelerar a exaustão deste recurso.

Assim, os dispositivos estão sujeitos a uma forma de negação de serviço também conhecida como ataque de exaustão de bateria no qual o atacante tenta exaurir a bateria do dispositivo rapidamente (JACOBY et al., 2006). Este ataque foi primeiramente identificado em redes *ad-hoc* sobre redes sensor *wireless* (MARTIN et al., 2001). Nestas redes, a energia consumida pela comunicação é o fator dominante do tempo de vida da bateria dos sensores. Estes normalmente entram em modo *power sleep* para conservar energia e estender o tempo de vida da bateria. O ataque *sleep deprivation* é uma forma de exaurir a bateria do dispositivo com altas taxas de demanda por processamento que normalmente são vistas quando se está sob uso normal. Estes tipos de ataques exploram o gerenciamento de energia do dispositivo para inibi-lo da habilidade de mudar para o estado de redução de consumo de bateria.

Em dispositivos móveis, este tipo de ataque pode tomar maiores dimensões, dificultando sua detecção e reação ao mesmo. Três categorias de ataques foram desenvolvidas (MARTIN et al., 2004): ataques nocivos, ataques benignos e ataques de requisição de serviço.

Ataques nocivos são programas criados ou modificados com o propósito de exigir mais poder de processamento do dispositivo e assim exaurir a bateria do dispositivo mais rapidamente. Exemplo típico desse tipo de ataque são vírus e *trojans* que são cada vez mais comuns para dispositivos móveis.

Os ataques benignos são ataques onde não há modificação de programas com aplicações nocivas, o foco principal é o aumento no consumo de recursos por processos

válidos para dispositivo. Um exemplo seria a execução de um *applet*[3], que seria invisível para o usuário mas que exigiria uma demanda maior de recursos para seu processamento.

Os ataques de requisição de serviço são aqueles onde o atacante realiza várias e repetidas tentativas de conexão ao dispositivo. São mais comuns em serviços de requisição através da rede. Ao contrário dos ataques típicos de rede onde o principal objetivo da requisição é tentar obter acesso ao servidor e usar algum serviço, neste tipo de ataque o objetivo é exigir performance forçando o dispositivo desperdiçar energia na consulta por em resposta a número excessivo de requisições decidindo se atende ou não a requisição.

Em (NASH et al., 2005), uma nova taxionomia é proposta para ataques de exaustão de bateria onde são apresentadas duas principais categorias: os ataques nocivos e benignos. Os ataques baseados em serviços de requisição foram agrupados como uma forma de ataque benigno. Entretanto, a maioria dos ataques nocivos pode ser conceitualmente transformada em um ataque distribuído.

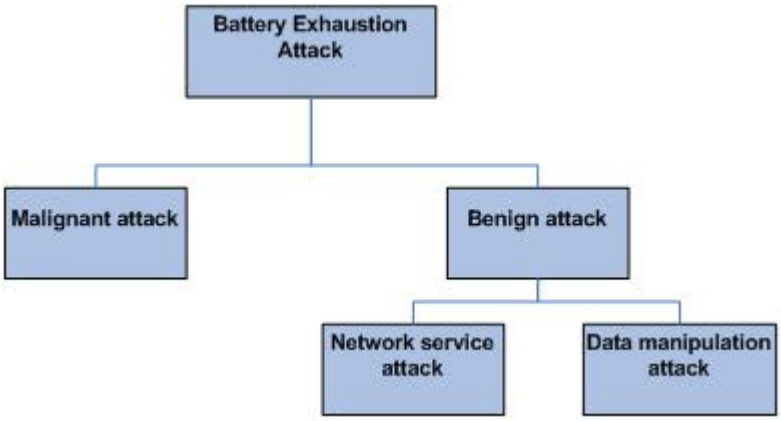


Figura 2.5 - Nova Taxonomia de Ataques de Exaustão de Bateria (NASH et al., 2005)

2.4 SISTEMAS DE DETECÇÃO E PREVENÇÃO DE INTRUSOS PARA REDES WIRELESS

Uma intrusão pode ser definida como qualquer conjunto de ações que tentem comprometer a integridade, confidencialidade ou disponibilidade dos dados e/ou sistema (SERVILLA et al., 1990). Uma forma de defender- se de intrusões é fazer uso de Sistemas de Detecção de Intrusões (IDS, *Intrusion Detection System*).

[3] Applet é um software aplicativo que é executado no contexto de outro programa. Os Applets geralmente têm algum tipo de interface de usuário, ou fazem parte de uma parte de uma destas dentro de uma página da *web.*

A detecção de intrusos é o processo de monitoramento de eventos ocorrendo em um sistema computacional ou rede e a análise de tais eventos em busca de sinais de possíveis incidentes, que são violações ou ameaças iminentes de violações de políticas de segurança computacional, políticas de uso aceitável, ou práticas padronizadas de segurança (NIST, 2002).

Os incidentes podem ter várias causas, tais como códigos maliciosos, atacantes tentando obter o acesso não autorizado aos sistemas a partir da *Internet*, e usuários autorizados de sistemas que se aproveitam dos seus privilégios ou tentam obter privilégios adicionais para os quais não estão autorizados. A prevenção de intrusos, por sua vez, é o processo de realizar a detecção de intrusos e tentar parar possíveis incidentes detectados.

Um IDS é um software que automatiza o processo de detecção de intrusos identifica o ataque depois de ocorrido. Já um IPS (*Intrusion Prevention System*) é um software que tem todas as capacidades de um sistema de detecção de intrusos e pode também tentar parar possíveis incidentes antes do mesmo ocorrer impedindo a ação do atacante. Tecnologias IDS e IPS oferecem várias das mesmas capacidades, e o administrador pode eventualmente desabilitar ferramentas de prevenção em produtos IPS, fazendo com que os mesmos funcionem simplesmente como IDSs.

As tecnologias IDPS (*Intrusion Detection Prevention System*) (NIST, 2007) usam várias metodologias para detectar incidentes: baseadas em assinaturas, baseadas em anomalias e análise de estado dos protocolos.

A maioria das tecnologias IDPS usa múltiplas tecnologias de detecção, ou separadas ou integradas, para prover uma detecção mais ampla e apurada.

Uma assinatura é um padrão que corresponde a uma ameaça conhecida. A detecção baseada em assinaturas é o processo de comparar assinaturas contra eventos observados para identificar possíveis incidentes.

A detecção baseada em assinaturas é muito efetiva na detecção de ameaças conhecidas, mas é ineficaz na detecção de ameaças desconhecidas. Por exemplo, se um atacante modificar o nome de um determinado vírus de “freepics.exe” para “freepics2.exe” a busca da assinatura será feita buscando “freepics.exe” e assim o sistema poderá não detectá-lo.

A detecção baseada em anomalias é o processo de comparar definições de atividades consideradas normais contra eventos observados por um determinado período para identificar desvios significantes.

Um IDS usando detecção baseada em anomalias possui perfis que representam o comportamento normal de vários elementos, como usuários, *hosts*, conexões de rede ou aplicações. Os perfis são desenvolvidos pelo monitoramento das características da atividade típica em um dado período de tempo.

O maior benefício de métodos de detecção baseados em anomalias é que eles podem ser muito efetivos na detecção de ameaças previamente desconhecidas. Por exemplo, suponhamos que um computador se torne infectado por um vírus. Esse vírus pode consumir recursos de processamento do computador, enviar um grande número de e-mails, iniciar um grande número de conexões de rede e realizar outro comportamento que seja significantemente diferente dos perfis estabelecidos para o computador.

Produtos IDS baseados em anomalias freqüentemente produzem muitos falsos positivos por causa da atividade benigna que se desvia significante dos perfis, especialmente nos ambientes mais diversificados e dinâmicos. Outro problema notável com o uso de técnicas de detecção baseadas em anomalias é que freqüentemente é difícil para os analistas determinarem porque um alerta em particular foi gerado e se o alerta procede e não é um falso positivo, devido á complexidade e quantidade de eventos que podem ter causado a geração do alerta.

A análise de estado dos protocolos é o processo de comparar perfis pré-determinados da atividade benigna de cada estado dos protocolos, contra eventos observados no sistema, com o objetivo de identificar desvios. Diferentemente da detecção baseada em anomalias, que usa perfis de *host* ou de rede, a análise de estado dos protocolos leva em conta os perfis universais desenvolvidos pelo fabricante, que especificam como os protocolos em particular devem ou não ser usados. Desse modo, o IDS deve ser capaz de entender e rastrear o estado dos protocolos das camadas de rede, transporte e aplicação que possuem a noção de estado. Como exemplo tem-se quando um usuário inicia uma sessão de FTP (File Transfer Protocolo), a sessão está inicialmente no estado de que não foi autenticado. Usuários ainda não autenticados deveriam executar poucos comandos neste estado tais como visualizar as informações de ajuda ou fornecer *login* de usuário e senhas. Uma parte importante deste estado de protocolos é o casamento de requisições com as respostas, assim, quando uma tentativa de autenticação ocorre os IDS podem identificar se houve sucesso ou não no estabelecimento da sessão. Uma vez que o usuário foi autenticado com sucesso, a sessão passa para o estado de autenticada e os usuários podem executar vários comandos. A execução de muitos destes comandos enquanto no estado de não autenticado pode ser

considerado suspeito e no estado autenticado poderiam ser comandos considerados normais. (NIST, 2007).

A análise de estado dos protocolos pode identificar seqüências inesperadas de comandos, tais como a execução do mesmo comando repetidamente ou a execução de um comando sem primeiro executar um comando do qual ele é dependente.

A primeira desvantagem dos métodos de análise de estado dos protocolos é que eles fazem um consumo intensivo de recursos. Isso se deve à complexidade da análise e o *overhead* envolvido na realização do rastreamento de estados para várias sessões simultâneas. Um segundo problema é que os métodos de análise de estado dos protocolos não detectam ataques que não violem as características do comportamento geralmente aceito para os protocolos, como executar várias ações benignas em um curto intervalo de tempo, podendo causar uma negação de serviço. O terceiro problema é que o modelo do protocolo usado por um IDS pode entrar em conflito com a forma que o protocolo é implementado em versões particulares de aplicações específicas ou sistemas operacionais, ou mesmo como as diferentes implementações do protocolo interagem.

Baseado no tipo de eventos que podem monitorar e na forma como são desenvolvidas, as tecnologias IDS podem ser classificadas em (NIST, 2007):

- Baseada em Rede: monitora o tráfego da rede, para segmentos de rede particulares ou dispositivos, e analisa as atividades dos protocolos das camadas de rede e aplicação para identificar atividade suspeita. Pode identificar vários tipos diferentes de eventos de interesse. É mais comumente empregada na fronteira entre redes, como na proximidade de *firewalls* de borda ou roteadores, servidores de VPNs (*Virtual Private Networks*), servidores de acesso remoto e redes *wireless*;
- *Wireless:* monitora o tráfego de redes *wireless* e analisa seus protocolos para identificar atividades suspeitas. Não pode identificar atividade suspeita nos protocolos das camadas de rede, transporte ou aplicação em que o tráfego da rede wireless está transferindo. É mais comumente empregada dentro da extensão da rede wireless de uma organização para monitorá-la, mas pode também ser empregada em locais onde o acesso à rede wireless não autorizado possa estar ocorrendo;
- Análise do comportamento da rede: examina o tráfego da rede para identificar ameaças que gerem fluxos de tráfego não usuais, tais como um ataque DDoS (*Distributed Denial of Service*), certas formas de códigos

maliciosos e violações de política. Sistemas de análise do comportamento da rede são mais freqüentemente empregados para monitorar fluxos entre as redes internas de uma organização, mas podem também serem empregados para monitorar fluxos entre as redes de uma organização e redes externas, como a *Internet*;

- Baseada em *host*: monitora as características de um único host e os eventos ocorrendo naquele *host* em busca de atividades suspeitas. Exemplos de características que um IDS baseado em host pode monitorar são: o tráfego da rede (somente para esse host), *logs* do sistema, processos em execução, atividades das aplicações, acesso e modificação de arquivos e mudanças na configuração do sistema e das aplicações. Os IDSs baseados em *host* são mais comumente empregados em *hosts* críticos, como servidores de acesso público e servidores contendo informações sensíveis.

2.4.1 IDSs Wireless

Os componentes típicos em um IDPS *wireless* são os mesmos que nos IDSs baseados em rede: consoles, servidores de bancos de dados (opcionais), servidores de gerenciamento e sensores. Todos os componentes, exceto os sensores, têm essencialmente a mesma funcionalidade para ambos os tipos de IDS. Sensores *wireless* realizam o mesmo papel básico que os sensores de IDSs baseados em rede, mas funcionam de uma forma muito diferente devido às complexidades do monitoramento das redes *wireless*.

Diferentemente de um IDS baseado em rede, o qual pode ver todos os pacotes na rede que monitora, um IDS *wireless* trabalha por tráfego de amostra. (NIST, 2007)

Existem duas bandas de freqüência a monitorar (2.4 GHz e 5 GHz), e cada banda é separada em vários canais. Atualmente, não é possível para um sensor monitorar todo o tráfego em uma banda simultaneamente. Um sensor precisa monitorar apenas um canal por vez. Quando o sensor está pronto para monitorar um canal diferente, ele deve desligar sua interface de rádio, mudar de canal e então tornar a ligar sua interface de rádio.

Por quanto mais tempo um único canal for monitorado, torna-se mais provável que o sensor perca atividades maliciosas ocorrendo em outros canais. Para evitar isso, os sensores normalmente trocam de canal freqüentemente, o que é conhecido como varredura de canais, de modo que ele possa monitorar cada canal em uma fração de tempo a cada segundo. Para reduzir ou eliminar a varredura de canais, sensores especializados estão disponíveis no

mercado que usam várias antenas de rádio de alta potência, com cada par rádio/antena monitorando um canal diferente. Devido a sua altíssima sensitividade, as antenas de alta potência têm também uma extensão de monitoramento bem maior que as antenas regulares.

Algumas implementações coordenam padrões de varredura entre os sensores com extensões alternadas, de modo que cada sensor precise monitorar apenas alguns canais.

Sensores *wireless* estão disponíveis em várias formas:

- Dedicados: um sensor dedicado é um dispositivo que realiza as funções de IDS *wireless*, mas não passa o tráfego de rede da fonte para o destino. Sensores dedicados freqüentemente são completamente passivos, funcionando em um modo de monitoramento de RF (*Radio Frequency*) para capturar o tráfego da rede. Alguns sensores dedicados realizam a análise do tráfego que eles monitoram, enquanto que outros sensores redirecionam o tráfego de rede para um servidor de gerenciamento para análise. O sensor é tipicamente conectado à rede cabeada. Sensores dedicados são normalmente projetados para um de dois usos:
  - Fixo: o sensor é empregado em uma localização particular. Tais sensores são tipicamente dependentes da infra-estrutura da organização;
  - Móvel: o sensor é projetado para ser usado em movimento. Por exemplo, o administrador de segurança pode usar um sensor móvel enquanto caminha pelo prédio de uma empresa ou campus para encontrar APs falsos.
- Embutido em um AP: vários fabricantes têm adicionado capacidades de IDS aos seus APs. Um AP embutido tipicamente provê uma capacidade de detecção menos rigorosa que um sensor dedicado porque o AP precisa dividir seu tempo entre prover o acesso à rede e o monitoramento de vários canais ou bandas em busca de atividades maliciosas. Se o IDS precisa apenas monitorar uma única banda e canal por vez, uma solução embutida pode prover uma disponibilidade de rede e segurança razoáveis. Se o IDS precisa monitorar várias bandas ou canais, o sensor precisa realizar a varredura de canais, o que pode interromper as funções de AP do sensor tornando-o temporariamente indisponível em sua banda e canal primários;
- Embutido em um *switch wireless*: *switches wireless* são idealizados para auxiliar o administrador com o gerenciamento e monitoramento de dispositivos *wireless*. Os *switches wireless* tipicamente não oferecem

capacidades de detecção tão fortes quanto os APs embutidos ou sensores dedicados.

Visto que os sensores dedicados podem focar apenas na detecção e não precisam transmitir o tráfego *wireless*, eles tipicamente oferecem capacidades de detecção mais robustas do que os sensores *wireless* embutidos em APs ou sensores embutidos em *switches wireless*. Entretanto, sensores dedicados são freqüentemente mais caros que sensores embutidos, pois os sensores embutidos podem ser instalados sobre o *hardware* existente, enquanto que os sensores dedicados envolvem tanto hardware quanto softwares adicionais.

Alguns fabricantes também disponibilizam software para sensores IDSs wireless que podem ser instalados em STAs, como laptops. O software sensor detecta ataques dentro da extensão das STAs e como más configurações das STAs, e relata esta informação aos servidores de gerenciamento.

Os componentes de IDSs *wireless* são tipicamente inter-conectados através de uma rede cabeada. Tal como em um IDS baseado em rede, uma rede separada de gerenciamento ou mesmo a rede padrão da organização pode ser usada para as comunicações entre os componentes do IDS *wireless*.

A escolha da localização dos sensores para um IDS *wireless* é um problema fundamentalmente diferente da escolha da localização para qualquer outro tipo de sensor IDS. Se a organização usa WLANs, os sensores *wireless* devem ser empregados de modo que eles monitorem toda a extensão de RF da WLAN da organização (tanto APs quanto STAs), o que freqüentemente inclui componentes móveis como laptops e PDAs. Várias organizações também optam por empregar sensores para monitorar partes de suas instalações onde não pode haver atividade de WLAN, bem como canais ou bandas que a WLAN da organização não deve usar.

IDSs *wireless* provêem vários tipos de capacidades de segurança. Devido ao fato de que os IDSs *wireless* são uma forma relativamente nova de IDS, tais capacidades ainda variam muito entre os produtos. Com o passar do tempo, a tendência é que essas capacidades se tornem mais consistentes. As capacidades de segurança mais comuns são: coleta de informações, registro, detecção e prevenção.

A maioria dos IDSs *wireless* pode coletar informações em dispositivos *wireless*.

Exemplos destas capacidades de coleta de informações são:

- Identificação de dispositivos de WLANs: a maioria dos sensores pode criar e manter um inventário de dispositivos WLANs observados, incluindo APs, clientes de WLANs e clientes de redes ad-hoc. O inventário normalmente é

baseado no SSID e nos endereços MAC das placas de redes wireless dos dispositivos. A primeira porção de cada endereço MAC identifica o fabricante da placa. Alguns sensores podem também usar técnicas de impressões digitais (fingerprinting) no tráfego observado para verificar o fabricante, ao invés de confiar na informação do endereço MAC, a qual pode ser forjada. O inventário pode ser usado para identificar novos dispositivos WLAN e também para a remoção de dispositivos existentes;

- Identificação de WLANs: a maioria dos sensores dos IDSs mantém o rastreamento de WLANs observadas, identificando-as pelo seu SSID. Os administradores podem então marcar cada entrada como sendo uma WLAN autorizada, uma WLAN vizinha benigna, como outra organização no mesmo prédio, ou uma falsa WLAN. Esta informação pode ser usada para identificar novas WLANs, bem como para priorizar respostas aos eventos identificados.

Os IDSs *wireless* tipicamente realizam registros extensivos de dados relacionados aos incidentes detectados. Estes dados podem ser utilizados para confirmar a validade dos alertas, para investigar incidentes e para correlacionar eventos entre os IDSs e outras fontes de registro. Campos de dados comumente registrados pelos IDSs incluem (NIST, 2007

- Timestamp (frequentemente data e tempo);
- Tipo de evento ou alerta;
- Índice de prioridade ou severidade;
- Endereço MAC da fonte (o fabricante é freqüentemente identificado a partir deste endereço);
- Número do canal;
- ID do sensor que observou o evento;
- Ação de prevenção realizada (se houver).

Os IDSs *wireless* podem detectar ataques, configurações erradas e violações da política de segurança no nível de protocolo WLAN, primariamente examinando a comunicação nos protocolos IEEE 802.11a, b, g e i. Os IDSs *wireless* não examinam as comunicações em níveis mais altos, como endereços IP ou carga útil de aplicações.

Alguns produtos realizam somente uma simples detecção baseada em assinaturas, enquanto que outros usam uma combinação de detecção baseada em assinaturas, detecção baseada em anomalias e técnicas de análise de estado dos protocolos. Esta última configuração é a ideal para se alcançar uma detecção mais ampla e apurada.

Os tipos de eventos mais comumente detectados pelos sensores dos WIDSs incluem:

- WLANs ou dispositivos de WLANs não autorizados: através de suas capacidades de obtenção de informações, a maioria dos sensores de IDSs *wireless* pode detectar APs falsos, STAs não autorizadas e WLANs não autorizadas, tanto no modo infra-estruturado quanto no modo ad-hoc;
- Dispositivos de WLANs com pouca segurança: a maioria dos sensores de IDSs *wireless* pode identificar APs e STAs que não estão usando os controles de segurança apropriados. Isto inclui a detecção de configurações erradas e o uso de implementações fracas de protocolos WLANs. Por exemplo, um sensor pode detectar que uma STA está usando o WEP[4] (ANSI/IEEE 802.11, 1999) ao invés de WPA2[5] (ANSI/IEEE 802.11i, 2004) ou IEEE 802.11i. Os maiores tipos de eventos que podem ser detectados por IDSs *wireless* caem nesta categoria de detecção;
- Padrões de uso não usuais: alguns sensores podem usar métodos de detecção baseados em anomalias para detectar padrões de uso de WLANs não usuais. Por exemplo, os sensores podem alertar na ocorrência de várias tentativas de conexão sem sucesso em um curto período de tempo, o que pode indicar uma tentativa de obtenção de acesso não autorizado para a WLAN;
- O uso de scanners de redes wireless: tais scanners são usados para identificar WLANs inseguras ou com pouca segurança. Sensores de IDSs *wireless* podem detectar somente o uso de scanners ativos, que são os que geram tráfego na rede *wireless*. Eles não podem detectar o uso de scanners passivos que simplesmente monitoram e examinam o tráfego observado;
- Ataques e condições de negação de serviço: ataques de DoS incluem ataques lógicos, como o *flooding*, que envolve o envio de um grande número de mensagens para um AP a uma alta taxa; e ataques físicos, como o *jamming*, que envolve a emissão de energia eletromagnética nas freqüências da WLAN para tornar tais freqüências inutilizáveis pela WLAN. Ataques de DoS podem freqüentemente ser detectados através de análise de estado do protocolo ou métodos de detecção de anomalias;

[4] Wireless Equivalent Protocol
[5] Wi-Fi Protected Access

- Disfarce e ataques MiTM (*Man-in-The Middle*): alguns sensores de IDSs wireless podem detectar quando um dispositivo está tentando assumir a identidade de outro dispositivo. Isso pode ser feito identificando diferenças nas características da atividade de cada um dos dispositivos, tais como certos valores nos quadros.

A maioria dos IDSs *wireless* pode identificar a localização física de uma ameaça detectada através do uso da triangulação - estimando a distância aproximada da ameaça a partir de múltiplos sensores, através da intensidade do sinal da ameaça recebido em cada sensor, e então calculando a localização física da ameaça. Isto permite a uma organização enviar o pessoal de segurança para o local para tratar a ameaça mais efetivamente. Sensores de IDSs em *handhelds* podem também ser usados para identificar a localização da ameaça, particularmente se sensores fixos não oferecerem capacidades de triangulação ou se a ameaça estiver se movendo.

IDSs *wireless* oferecem algumas ferramentas de personalização. A maioria possui margens de tolerância que podem ser usados para a detecção baseada em anomalias. “Listas negras” e “listas brancas” podem ser usadas para carregar listas de dispositivos maliciosos conhecidos ou dispositivos benignos da WLAN, respectivamente. As listas podem também ser usadas para gravar NICs (*Network Interface Cards*) de fabricantes não autorizados, e alertas podem ser gerados quando qualquer NIC que não estiver na lista autorizada for usado para APs ou STAs.

Alertas individuais podem ser customizados, como pode ser feito para IDSs baseados em redes. A edição de código não é disponível para a maioria dos produtos, embora alguns fabricantes permitam aos administradores entrar expressões lógicas complexas para afinar certas capacidades de detecção.

Sensores de IDSs *wireless* oferecem dois tipos de capacidades de prevenção de intrusos:

- *Wireless*: alguns sensores podem encerrar conexões entre uma STA maliciosa ou mal configurada e um AP autorizado, ou ainda entre uma STA autorizada e um AP malicioso ou mal configurado. Isto é feito tipicamente pelo envio de mensagens para as extremidades da conexão, notificando os mesmos a se desassociarem da sessão corrente. A partir daí, o sensor passa a rejeitar qualquer tentativa de estabelecimento de uma nova conexão entre esses dispositivos;

- Cabeada: alguns sensores podem instruir um switch na rede cabeada a bloquear toda a atividade de rede envolvendo uma STA particular ou AP, baseando-se no endereço MAC do dispositivo ou porta do switch. Por exemplo, se uma STA está disparando um ataque contra um servidor na rede cabeada, um sensor pode direcionar o switch para bloquear toda a atividade relacionada a STA. Convém destacar que, apesar de ser efetiva para bloquear as comunicações das STAs ou APs maliciosos na rede cabeada, esta técnica não vai impedir a STA ou o AP de continuar realizando ações maliciosas através da rede wireless e seus protocolos.

Uma consideração importante é o efeito que a execução das ações de prevenção pode ter no monitoramento dos sensores. Por exemplo, se um sensor está transmitindo sinais para encerrar conexões, ele pode não ser capaz de realizar a varredura de canais para monitorar outros dispositivos até que ele complete a ação de prevenção em andamento. Para atenuar isso, alguns sensores possuem duas interfaces de rádio: uma para monitoramento e detecção e outra para a realização de ações de prevenção.

2.5 PRINCIPAIS IDSs WIRELESS E FERRAMENTAS EXISTENTES

Nesta seção, as principais arquiteturas propostas na área de pesquisa em IDSs *wireless* serão apresentadas, bem como os principais IDSs *wireless* já disponíveis para uso, além de uma variedade de ferramentas que podem ser empregadas tanto na realização de ataques contra as redes 802.11 quanto na tomada de contramedidas por parte da equipe de segurança da organização.

Em (PLESKONJIC, 2003), uma arquitetura para um WIDS (*Wireless Intrusion Detection System*) é proposta e consiste nos seguintes componentes: agente, sensor, console de gerenciamento e ferramentas de relatório. Essa arquitetura é baseada em agentes inteligentes e algumas de suas capacidades, como: auto-aprendizagem, cooperação, autonomia e poder de decisão. Esses agentes são integrados aos clientes da rede, onde realizam a coleta e filtragem local de dados e cooperam com os agentes vizinhos, constituindo assim um módulo de detecção cooperativa. Desse modo, as respostas aos ataques podem ser locais ou globais. Além disso, novos tipos de ataques podem ser detectados, graças ao poder de auto-aprendizagem da arquitetura, que utiliza técnicas de Inteligência Artificial como Redes Neurais e Lógica *Fuzzy*.

Em (YANG et al., 2004), uma arquitetura distribuída e colaborativa para um sistema de detecção de intrusos *wireless* é apresentada. Nessa arquitetura, cada nó móvel possui um agente IDS que monitora as atividades locais, incluindo atividades do usuário, do sistema e atividades de comunicação. Esse agente participa ativamente na detecção e resposta a intrusões, sendo responsável por detectar sinais de intrusões localmente e independentemente, colaborando também com seus nós vizinhos, para detectar intrusões em uma extensão maior. Cada módulo representa um agente móvel leve com certas funcionalidades, sendo que alguns desses módulos estão presentes em todos os *hosts* móveis, enquanto que outros estão distribuídos em apenas um grupo selecionado de *hosts* móveis.O modelo conceitual de cada agente IDS é constituído de um sensor e quatro módulos:

- Módulo de monitoramento de rede: monitoramento dos pacotes que trafegam na rede;
- Módulo de monitoramento de *host*: agentes para monitoramento de informações do sistema e aplicações do *host* monitorado;
- Módulo de decisão: analisa as informações de *host* e rede para identificação de ameaças;
- Módulo de ação: modulo responsável pela reação a ameaça identificada;
- Módulo de comunicação: responsável pela troca de comunicação sobre o comportamento maliciosa de algum segmento da rede ou de algum *host.*

Em (DASGUPTA et al., 2003), um sistema multi-agente denominado MMDS (*Multi-level Monitoring and Detection System*) é apresentado. Este realiza em tempo real o monitoramento, análise, detecção e geração de respostas as tentativas de intrusão. Esse sistema usa um módulo *Fuzzy* de suporte a decisões, que utiliza regras para diferentes ataques e tem como foco a detecção por anomalias tanto em redes *wireless ad-hoc* quanto infra-estruturadas. A modelagem do comportamento é elástica, ou seja, ela se adapta às flutuações normais de uso em função do tempo. O sistema provê um *framework* hierárquico de agentes de segurança, onde cada nó de segurança consiste de quatro agentes: agente de gerenciamento, agente monitor, agente de decisão e agente de ação. As atividades desses agentes são coordenadas pelo agente de gerenciamento durante os processos de percepção, comunicação e geração de respostas.

Em (SCHMOYER et al., 2004), uma arquitetura para um sistema de detecção e respostas a intrusões *wireless* é apresentada e utiliza estratégias de respostas adaptativas baseadas em confidência de alarmes, freqüência de ataques, avaliação de riscos e custos estimados de resposta. Nessa arquitetura, cada nó usa um agente IDS para monitorar a atividade local e responder a intrusões. Visto que a atividade local nem sempre provê dados suficientes para detectar ou determinar o tipo de um ataque, os agentes locais devem ser capazes de se comunicar de forma segura e agir coletivamente quando uma intrusão estiver sob suspeita. Um protótipo foi desenvolvido através da criação de uma ferramenta para detectar ataques e enviar *frames* de resposta 802.11. O conhecido ataque MiTM foi utilizado como estudo de caso.

Em (LACKEY et al., 2003), uma arquitetura para o monitoramento de redes *wireless* é descrita. Visto que grande parte desta arquitetura é bastante similar a topologias de avaliação de vulnerabilidades e sistemas de detecção de intrusos tradicionais, ela é chamada de WIDE (*Wireless Intrusion Detection Extensions*). A WIDE consiste de três componentes principais: o sensor, o analisador mestre e o adaptador de alertas. Cada sensor é configurado para enviar dados de forma segura para o mestre, para análise. Esse mestre pode residir no próprio sensor, para conservar largura de banda, ou pode residir em uma localização central de modo que a correlação entre múltiplos sensores seja possível. Em ambos os casos, o mestre é configurado com certo número de módulos de ataques, que são programas independentes que podem ser carregados individualmente no espaço de execução do mestre, para processar dados e gerar alertas. Por exemplo, o módulo de detecção do ataque de DoS usa métodos estatísticos sobre a intensidade do sinal e os níveis de ruído para determinar quando potenciais ataques de DoS estão ocorrendo.

O *AirDefense* (AIRDEFENSE, 2007) é um sistema de detecção e prevenção de intrusos *wireless*, que consiste de sensores dispostos por toda a rede, os quais são interfaceados com uma ferramenta de gerenciamento e administrados por um console de gerenciamento. Ele detecta APs não autorizados e ataques, além de diagnosticar vulnerabilidades potenciais, como más configurações. Além disso, o AirDefense oferece outras funções de gerenciamento tais como rastreamento de falhas e auditoria.

O *AirMagnet* (AIRMAGNET, 2007) é uma ferramenta comercial de monitoramento e diagnóstico de rede para Windows e Pocket PC (*personal computer)*, que roda em laptops e *handhelds*. Assim como *AirDefense*, ele incorpora a detecção de vulnerabilidades e intrusões. Para intrusões, o *AirMagnet* detecta pontos de acesso e clientes não autorizados e ataques de DoS por *flooding*. Esse software requer que um técnico se mova

ao redor da rede para detectar possíveis ameaças de segurança. Pode ser usado também por um intruso, mas esse uso é pouco provável devido ao seu alto custo.

O *AirSnare* (AIRSNARE, 2007) é um programa para Windows que funciona como sistema de detecção de intrusos, detectando requisições de DHCP (*Dynamic Host Configuration Protocol)* ou endereços MAC não-autorizados tentando se conectar com um ponto de acesso. A resposta à intrusão consiste de uma mensagem de alerta por e-mail para o administrador, a gravação da sessão inteira e, opcionalmente, uma mensagem para o intruso informando que o mesmo está sendo monitorado. Ele é compatível com o *Ethereal*, com o objetivo de oferecer recursos adicionais de análise e rastreamento.

O *Snort-Wireless* (SNORT-Wireless, 2007) é um WIDS *open-source* projetado para se integrar ao ambiente do *Snort* 2.x (SNORT, 2007), que é um IDS para redes cabeadas. Ele permite a criação de regras customizadas, baseadas na estrutura dos pacotes *wireless*, para a detecção de APs não autorizados, *wardrivers* e redes *ad-hoc*.

A Red-M (RED-M, 2007) desenvolveu um IDS *wireless* que monitora todos os serviços baseados em *Wi-Fi* e *Bluetooth*, identificando falhas de segurança ou fraquezas em sistemas que podem torná-los vulneráveis a ataques. Equipamentos de sonda são instalados em toda área de cobertura para monitorar e prevenir o acesso não autorizado de intrusos ou atividade de rogue APs. Esses equipamentos são controlados de forma centralizada, e enviam toda a informação e alertas para um servidor de detecção de intrusos. Além disso, o módulo de contramedidas pode interromper e isolar dispositivos intrusos tentando se infiltrar na rede.

O *Kismet* (KISMET, 2007) é um detector de redes *wireless* 802.11, *sniffer* e sistema de detecção de intrusos, baseado em Linux. Ele trabalha com qualquer placa de interface de rede *wireless* que suporte o monitoramento em modo promíscuo, podendo assim capturar o tráfego em redes 802.11a, 802.11b e 802.11g. O *Kismet* monitora passivamente o tráfego *wireless*, obtendo dados para identificar SSIDs (*Service Set IDentifier)*, endereços MAC, canais e velocidades de conexões, até mesmo de WLANs que não disseminam sinais de *beacon*. Também identifica dados com vetor de inicialização fraco, os quais podem ser usados por ataques contra a criptografia WEP. O *Kismet* pode ser usado com agentes remotos capturando dados e enviando-os para um servidor central para correlação e relatórios. Esta é uma arquitetura muito comum para IDSs em geral, de modo que muitas organizações têm empregado o *Kismet* com um IDS *wireless* bastante efetivo e extensível.

O NetStumbler (NETSTUMBLER, 2007), também conhecido como Network Stumbler, é um *sniffer* para Windows que possibilita a detecção de WLANs que usam os padrões 802.11a, 802.11b ou 802.11g. Ele envia um quadro de *probe request* para endereço

do broadcast da rede 802.11, que faz com que todos os APs na área respondam com um quadro de *probe response.* Estes quadros contêm informações de sua configuração de rede, como seu SSID, status WEP, endereço MAC, nome, canal em que está transmitindo, fabricante, tipo e outras informações. Geralmente, é utilizado para comprovar a integridade e o correto funcionamento da WLAN, localizar zonas onde há fraca cobertura, detectar outras redes que possam estar interferindo com WLAN e até mesmo pontos de acesso não autorizados. Uma versão do NetStumbler está disponível para Windows CE, sendo denominada MiniStumbler (NETSTUMBLER, 2007). Existe também uma ferramenta bastante similar ao NetStumbler, que roda em sistemas BSD (*Berkeley Software Distribution*), sendo denominada dStumbler (BSD-AIRTOOLS, 2007). Ela faz parte do pacote BSD *Air Tools*, que provê um conjunto completo de ferramentas para auditoria de redes *wireless* 802.11b. O MacOS também conta com uma ferramenta bastante similar, denominada MacStumbler (MACSTUMBLER, 2007). Tanto o NetStumbler quanto o *Kismet* têm a habilidade de trabalhar em conjunto com sistemas GPS (*Global Positioning System*), para mapear a exata localização de WLANs identificadas. Informações GPS para WLANs podem ser obtidas em (WIGLE, 2007). O *Kismet* possui uma ferramenta denominada GPSMap, que pode indicar através de mapas a localização física de dispositivos, para que os mesmos possam ser examinados e eventualmente desligados.

O Wireshark (WIRESHARK, 2007) é um analisador de protocolos de rede capaz de capturar dados em vários tipos de redes, inclusive as redes 802.11, com suporte para centenas de protocolos. Ele roda em várias plataformas, como Windows, Linux, OS X, Solaris, Free BSD e Net BSD. Pode realizar a análise através de uma captura em tempo real, ou através de dados importados a partir de vários formatos de arquivos de captura. Também pode exportar sua saída para os formatos XML (eXtensible Markup Language), PostScript, CSV (Comma Separated Values) ou texto simples.

O AirJack (AIRJACK, 2007) é um *driver* Linux customizado que dá ao usuário um acesso fácil e completo para a montagem de pacotes 802.11, permitindo-o forjar endereços MAC e injetar quadros de gerenciamento na rede. Essa ferramenta explora justamente os problemas do padrão 802.11b relativos a falta de autenticação dos quadros de gerenciamento. O *driver* suporta as placas baseadas nos *chipsets* PRISM2 e Hermes.

O AirSnort (AIRSNORT, 2007) é uma ferramenta que recupera chaves criptográficas em WLANs que usam apenas WEP como método de criptografia. Ele trabalha monitorando as transmissões passivamente e calculando a chave criptográfica assim que um número suficiente de pacotes tenha sido capturado. O AirSnort roda tanto no Windows quanto

no Linux, requerendo que a placa de interface de rede wireless suporte o monitoramento em modo promíscuo.

O WEPCrack (WEPCRACK, 2007) é uma ferramenta open source para quebrar chaves WEP, baseado na implementação do ataque descrito por Fluhrer, Mantin e Shamir (FLUHRER et al., 2001). Diferentemente do AirSnort, deve ser usado em conjunto com um *sniffer* de pacotes separado, visto que não possui a habilidade de capturar tráfego de rede.

O Host AP (HOSTAP, 2007) é um *driver* Linux para placas de interface de redes wireless baseadas nos chipsets Intersil Prism2/2.5/3. Esse *driver* assume as funções de gerenciamento 802.11 na estação em que está instalado, funcionando como um AP.

O Fake AP (FAKEAP, 2007) é um programa simples que simula uma lista de APs especificada pelo usuário, através da disseminação de quadros de *beacon* 802.11b. Isto confunde potencialmente qualquer intruso que estiver escutando a rede passivamente. Ele está disponível sob a GPL (*General Public Licence*) e roda em Linux e em algumas versões de BSD.

O AirSnarf (AIRSNARF, 2007) é uma ferramenta que configura um sistema Linux com uma placa de interface de rede *wireless* baseada em Prism2, para funcionar como um falso AP. Através do uso de servidores Web e DNS virtuais e de redirecionamento Web, um usuário que se associa com esse falso AP pode receber páginas falsas para portais comuns.

Em (NASH et al., 2001), propõe um modelo de IDS voltado para dispositivos móveis que toma com análise atividades em execução no dispositivo móvel tais como, processos ativos no dispositivo, acesso a disco, para estimar o poder consumido por processo ativo no dispositivo. E assim, identificar processos tem maior demanda para o dispositivo e detectar quais processos podem ser potencialmente ataques para exaustão de bateria.

Em (JACOBY et al., 2006), introduz-se um IDS baseado no comportamento de bateria dos dispositivos móveis. O BBID (*Battery Based Intrusion Detection*) é o composto de dois módulos, o HIDE (*Host Intrusion Detection Engine*) e o HASTE (*Host Analyze Signature Trace Engine*). Baseiam-se em conjunto de regras que tem como objetivo identificar qualquer comportamento anormal da bateria. HIDE analisa a energia dissipada pelo dispositivo por um período de tempo para identificar possíveis situações onde há um comportamento anormal da bateria. Este módulo é responsável pela captura e armazenamento de informações, tais como tráfico presente sobre a placa de rede do dispositivo, aplicações em execução, e permite que dados sejam visualizados pelo usuário. O SPIE (*Scan Port Intruse Engine*) é um módulo complementar que faz o monitoramento de portas ativas no dispositivo e captura o tráfico destinado a elas. O HASTE foi projetado para dispositivos mais robustos.

Ele foi usado para capturar informações sobre o comportamento da bateria e correlacionar com os possíveis padrões de assinatura de ataques. Isto é feito adquirindo a assinatura da energia do dispositivo para criar uma assinatura de freqüência para a mesma usando a FFT (*Fast Fourier Transform*) para detectar os altos índices de consumo de bateria pelo dispositivo. O usuário é informado através de alertas sobre qualquer atividade considerada anormal.

Em (BUENNEMEYER et al., 2008), utiliza-se o mesmo conceito para proposta de um IDS que toma com parâmetro atividades que têm maior incidência sobre a bateria do dispositivo móvel. Composto de dois módulos: BSIPS (*Battery Sensing Intrusion Protection System*) e CIDE (*Correlation Intrusion Detection Engine*). BSIPS provê o monitoramento de limites e notificações de alerta quando mudanças fora do padrão são detectadas sobre os dispositivos. Os *hosts* são empregados com sensores na rede *wireless* e formam a base do *Canary-Net* IDS (BUENNEMEYER et al., 2006). O monitoramento do poder consumido com comunicação *Wi-fi* e *Bluetooth* irregulares e atividades de ataque são detectadas e reportadas para o servidor para uma correlação com *Snort* alertas. O *Snort* atua como analisador do tráfego da rede para os possíveis ataques que podem partir dela. BSIPS usa um algoritmo DTC (*Dynamic Threshold Calculation*) para detectar quais atividades excedem seus limites e indicar um possível ataque. CIDE faz um rastreamento das informações informadas pelo cliente com as armazenadas para calcular um desvio padrão do comportamento do dispositivo e assim tentar detectar qualquer tentativa de ataque.

O *Kaspersky* Anti-Virus *Móbile* (KASPERSKY, 2008) é uma solução conveniente e confiável para a proteção de *smartphones* contra os programas mal-intencionados direcionados a plataformas móveis. Entre suas principais características pode-se destacar:

- Verificação por demanda: o usuário pode informar a verificação do dispositivo a qualquer momento e informar onde deseja fazer a verificação;
- Verificação programada;
- Verificação no acesso: inicializado quando usuário realiza qualquer operação sobre arquivos desconhecidos enviados através SMS, MMS e e-mail;
- *Anti-spam* para SMS/EMS/MMS;
- Atualizações automáticas.

O *E-Set Mobile Antivírus* (E-SET, 2008) é um anti-vírus também lançado recentemente para dispositivos móveis da plataforma *Symbian*.

O Symantec *Mobile AntiVirus* 4.0 (SYMANTEC, 2008) para *Windows Mobile*, desenvolvido para PocketPCs e *Smartphones*, baseados na plataforma *Windows Mobile* 5.0. O Symantec *Mobile AntiVirus* 4.0 para *Windows Mobile* protege automaticamente dispositivos móveis de ameaças transmitidas por e-mail e serviços de mensagem multimídia (MMS), transferidos de cartões de memória, rede celular e *Wi-Fi*, transmitidos por *Bluetooth* ou por meio de conexões em infravermelho.

O *McAfee* VirusScan (MCAFEE, 2008) PDA desenvolvido pela *McAfee* para dispositivos baseados no padrão *Pocket PC* e Microsoft Windows® Mobile. É uma ferramenta que complementa outros *softwares* de antivírus . Faz *scaneamento* de aplicações, dados, registros modificados, tipos conhecidos de arquivos conhecidos, executáveis e a memória *flash* .

A Tabela 2.1 apresenta um quadro comparativo com o resumo das funcionalidades dos principais IDS apresentados nesta seção, onde "X" representa que funcionalidade está presente na ferramenta especificada.

Tabela 2.1 – Tabela comparativa de funcionalidade das ferramentas IDS wireless

Ferramenta	Sensores	Agentes	Detecção		Monitoramento		Reação	Prevenção
			Assinaturas	Anomalias	Rede	Host		
(PLESKONJIC, 2003)	X	X	X		X	X	X	X
(YANG et. al , 2004)		X	X		X	X	X	
(DASGUPTA et al., 2003)		X		X	X		X	
(SCMOYER et al., 2004)		X	X	X		X	X	
(LACKEY et al. 2004)	X		X		X			
(AirDefense, 2007)	X		X		X		X	X
(NASH et al., 2001)	X			X		X		
(JACOBY et al., 2006)	X		X	X	X	X		
(BUENNEMEYER et al., 2006)	X		X		X	X		

2.6 CONCLUSÃO

Neste capítulo, conceitos sobre redes *wireless*, dispositivos móveis, segurança em redes *wireless*, ataques destinados a dispositivos móveis, segurança em redes *wireless* além de ferramentas e projetos para garantir a segurança para estas redes e dispositivos móveis foram apresentados.

Na seção de redes *wireless*, apresentou-se uma visão geral das mesmas, os principais padrões e com as mesmas podem ser estruturadas. Na seção de dispositivos móveis, apresentou-se uma descrição geral de suas características com ênfase em limitações e quais os impactos ocasionados por essas limitações. Na seção de segurança, apresentou-se uma série de vulnerabilidades a que as redes *wireless* estão expostas e as suas implicações neste tipo de ambiente. Na seção de IDS *wireless*, apresentou-se os conceitos básicos de intrusão e como estes ocorrem em redes *wireless* principais ataques, como eles acontecem e quais as principais ferramentas e projetos voltados para este tipo de rede e dispositivos móveis.

Nota-se que os projetos em segurança para redes *wireless* em sua grande maioria são voltados para proteção da rede local, sem uma proteção específica para os usuários que fazem parte, principalmente dispositivos móveis. Para este grupo, as pesquisas são iniciais e projetos vêm ganhando uma maior atenção devido a crescimento considerável no uso destes dispositivos em redes *wireless*.

No próximo capítulo, apresenta-se a proposta de uma arquitetura de um sistema detector de intrusão para usuários de dispositivos móveis como adaptação e extensão do IDS-NIDIA.

3 ARQUITETURA PROPOSTA: MODELAGEM

A arquitetura proposta tem como base o trabalho de (SILVA, 2006) que por sua vez é uma adaptação e extensão do IDS NIDIA.

A arquitetura proposta emprega a detecção por anomalias com estratégia de análise buscando identificar comportamentos anormais dos dispositivos monitorados. Este monitoramento baseia-se em informações coletadas nos dispositivos conjuntamente com as informações capturadas através do tráfego de rede destinado para aqueles dispositivos.

O comportamento normal para o dispositivo é diagnosticado através de dados históricos coletados durante um período de operação normal do dispositivo. Todo o processo de análise e diagnóstico de um possível ataque é feito pelo *IDS_Proxy* que dependendo do tipo de ataque informa ao dispositivo sobre a ameaça e como reagir ou dependendo do tipo de ataque o servidor poderá tomar a contramedida necessária.

A Figura 3.1 apresenta o cenário de utilização da solução proposta.

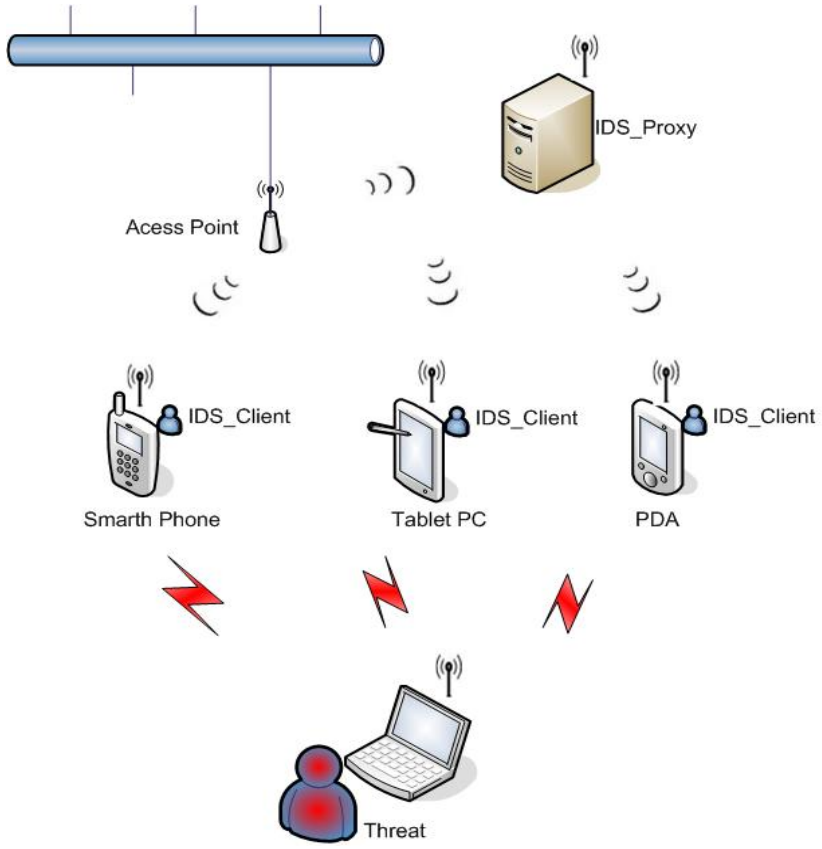


Figura 3.1 - Cenário de utilização da solução proposta

Conforme ilustrado na Figura 3.1, tem-se os seguintes componentes da arquitetura:

- *IDS_Proxy*: atua promiscuamente na rede capturando o tráfego destinado e originado dos dispositivos monitorados. Recebe os dados dos mesmos para uma identificação sobre o comportamento do dispositivo, além de armazenar os dados enviados pelos dispositivos e capturados na rede no formato XML (*eXtensible Markup Language*) e tomar também as possíveis contramedidas de acordo com os ataques identificados;
- *IDS_Client*: responsável pela coleta sobre das informações no dispositivo monitorado, essas informações são enviadas ao *IDS_Proxy* em intervalos regulares para que sejam analisadas;
- *Acess Point*: permite aos dispositivos ter acesso aos recursos da rede coordenando a comunicação entre os mesmos;
- Dispositivos móveis: representados na Figura 3.1 pelos *Smarth Phones*, PDAs e *Tablet PCs*;
- *Threat*: representam os ataques a que os dispositivos móveis estão submetidos.

3.1 CAPTURA DE REQUISITOS

A especificação de requisitos é uma etapa essencial do processo de desenvolvimento de software, que compreende uma definição completa do comportamento externo do sistema, tanto em termos de requisitos funcionais quanto não funcionais. Nesta fase são levantadas todas as informações essenciais para a elaboração do sistema.

Os requisitos do sistema ou de um software são condições ou capacitação que devem ser contempladas pelo software, geralmente necessitadas pelo cliente e/ou usuário para resolver um problema ou alcançar um objetivo. São fundamentais para elaborar um sistema que atenda e satisfaça plenamente os anseios do cliente e da equipe desenvolvedora do projeto (SOMMERVILLE, 2005).

Esses requisitos podem ser classificados em: funcionais que especificam declarações de funções que o sistema deve fornecer reagindo a entradas específicas e a determinadas situações particulares e não funcionais que relatam restrições sobre as funções

oferecidas pelo sistema, tais como restrições de tempo, restrições no processo de desenvolvimento, padrões, etc.

3.1.1 IDS_Client

O levantamento das especificações sobre as funcionalidades para o *IDS_Client* teve como fundamento o trabalho de (SILVA, 2006) que propôs uma ferramenta de detecção de intrusão para usuários finais, no entanto, este destinou-se a usuários em rede local. Assim, o *IDS_Client* propõe o mesmo objetivo, com funcionalidade especificas para dispositivos móveis em um ambiente *wireless*. Diferenças técnicas entre o ambiente da proposta de (SILVA, 2006) e usuário final implicaram em algumas modificação sobre as funcionalidades dos *IDS_Client*. Desta forma suas funcionalidades são as seguintes:

- Obter informações específicas de hardware e software do dispositivo móvel;
- Enviar e receber informações através da rede;

Os requisitos não funcionais identificados foram os seguintes:

- A execução do *IDS_Client* não poderá ter impacto sobre consumo de recursos do dispositivo ou interferir nos processos em execução do usuário;
- O dispositivo móvel necessita ter conexão de rede *wireless;*
- O sistema operacional para execução do *IDS_Client* é o Palm OS (PALM, 2008);

3.1.2 IDS_Proxy

Para validação da arquitetura proposta as funcionalidades definidas para *IDS_Proxy* são as seguintes:

- Captura e análise de informações na rede *wireless;*
- Recebimento e análise das informações enviadas pelo dispositivo móvel;
- Identificar ameaças dirigidas ao dispositivo móvel;
- Notificar ao dispositivo móvel qualquer ameaça dirigida ao mesmo;
- Executar reação cabível a uma ameaça identificada quando possível;
- Armazenar as informações capturadas;
- Gerenciamento das informações dos dispositivos móveis.

Os requisitos não funcionais para o IDS_Proxy são os seguintes:

- Sistema operacional para execução Linux;
- Máquina virtual Java (SUN, 2008) versão 1.6.0_05 ou superior;
- Placa de rede com conexão *wireless;*
- Memória RAM[6] com no mínimo 512 Mb.

A representação gráfica das funcionalidades para *IDS_Client* e *IDS_Proxy* foram definidas através de diagramas de caso no padrão da UML[7] (BOOCH, 2006) e são detalhados na seção seguinte.

3.2 DIAGRAMAS DE CASO DE USO

Os diagramas de caso de uso têm um papel central na modelagem do comportamento de um sistema, de um subsistema ou de uma classe. São importantes para visualizar, especificar e documentar o comportamento de um elemento, tais diagramas são modelados através de um conjunto de casos de uso e atores e seus relacionamentos.

Vários casos de uso poderiam ser gerados para este projeto. Foram escolhidos alguns, considerados de maior relevância para o entendimento do mesmo. São eles:

IDS_Client: Capturar dados, enviar dados, receber alertas do *IDS_Proxy* e executar contramedidas;

IDS_Proxy: Capturar *frames*, receber dados, identificar ataques, enviar notificação e executar contramedidas.

3.2.1 IDS_Client

O *IDS_Client* é executado periodicamente nos dispositivos previamente cadastrados pelo administrador do sistema. O *IDS_Client* captura informações sobre as aplicações instaladas no dispositivo, assim como, informações sobre a bateria e memória do dispositivo. Estes dados capturados são enviados para *IDS_Proxy* para análise e identificação de possíveis ataques.

---

[6] RAM Random Access Memory
[7] Unified Modeling Language

A Figura 3.2 apresenta o caso de uso *capturar dados*. A Tabela 3.1 especifica o caso Capturar dados, onde são descritos os atores que participam do caso de uso, a descrição do mesmo, suas pré-condições, fluxo principal, subfluxos e fluxos excepcionais.

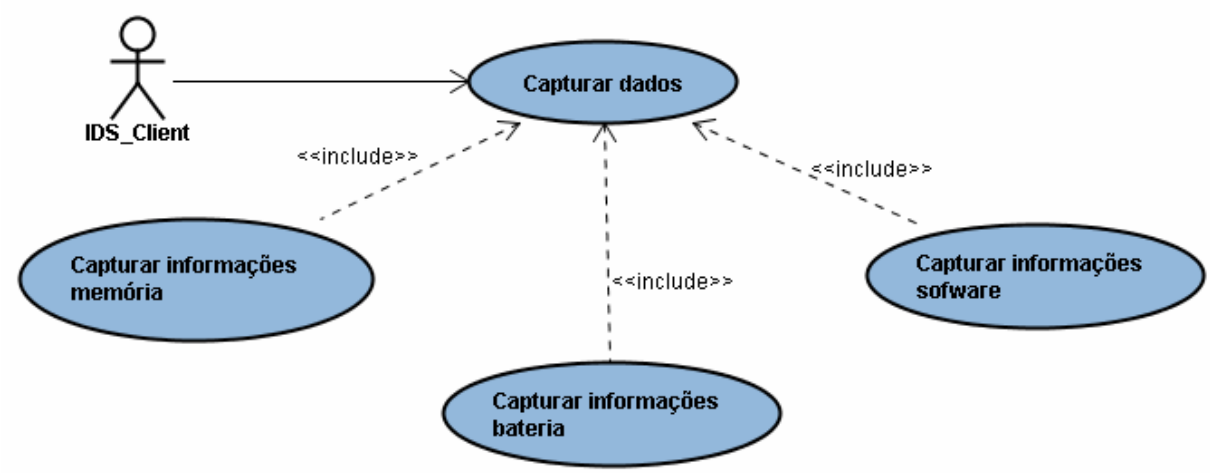


Figura 3.2 - Diagrama de caso de uso capturar dados

Tabela 3.1 - Descrição do caso de uso capturar dados

	Caso de Uso: Capturar dados
Atores	*IDS_Client*
Descrição	Obtém informações sobre software, bateria e memória do dispositivo.
Pré-Condições	*IDS_Client* instalado no dispositivo móvel.
Fluxo Principal	O *IDS_Client* é executado no dispositivo móvel para obter informações sobre softwares instalados, informações sobre memória e bateria do dispositivo móvel.
Subfluxos	
Fluxos Excepcionais	Erro na execução do *IDS_Client.*

A Figura 3.3 apresenta o caso de uso Enviar dados. A Tabela 3.2 faz a descrição do caso de uso enviar dados, detalhando os atores participantes do caso de uso e os fluxos gerados pelo mesmo.

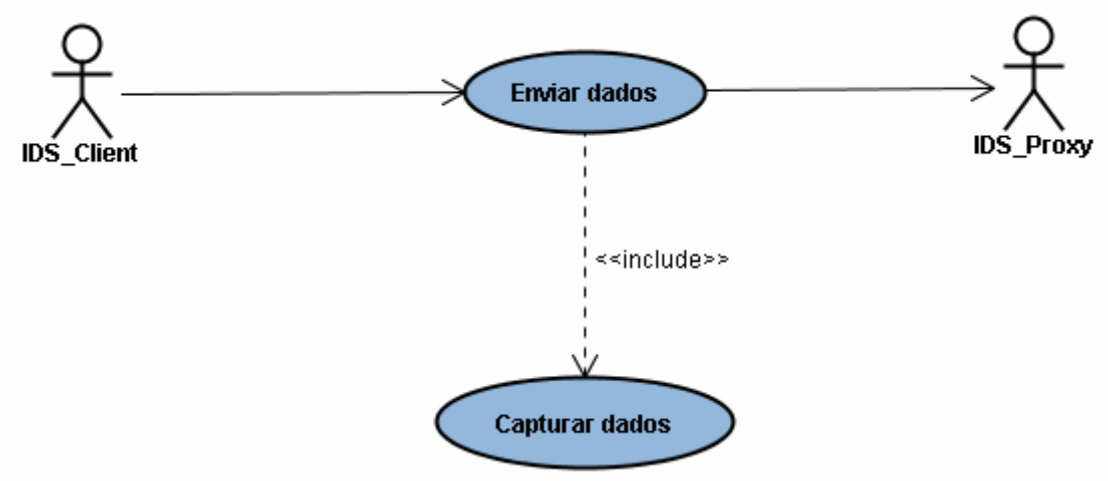


Figura 3.3 - Diagrama de caso de uso enviar dados

Tabela 3.2 - Descrição do caso de uso enviar dados

	Caso de Uso: Enviar dados
Atores	*IDS_Client* e *IDS_Proxy*
Descrição	Envia as informações capturadas pelo *IDS_Cliente* para *IDS_Proxy*
Pré-Condições	Conexões de rede disponível para comunicação com o servidor
Fluxo Principal	Após a leitura das informações pelo *IDS_Cliente*, estas são enviadas diretamente para o *IDS_Proxy* para criar um histórico sobre as informações do dispositivo e identificação de possíveis ataques.
Subfluxos	Leitura dos dados a serem enviados.
Fluxos Excepcionais	1- Erro na execução do *IDS_Client.* 2- Erro na execução do *IDS_Proxy*. 3- Não disponibilidade de conexão para envio das informações.

A Figura 3.4 apresenta o caso de uso receber alertas. A descrição dos atores e fluxos gerados pelo caso de uso receber alertas são detalhados na Tabela 3.3.

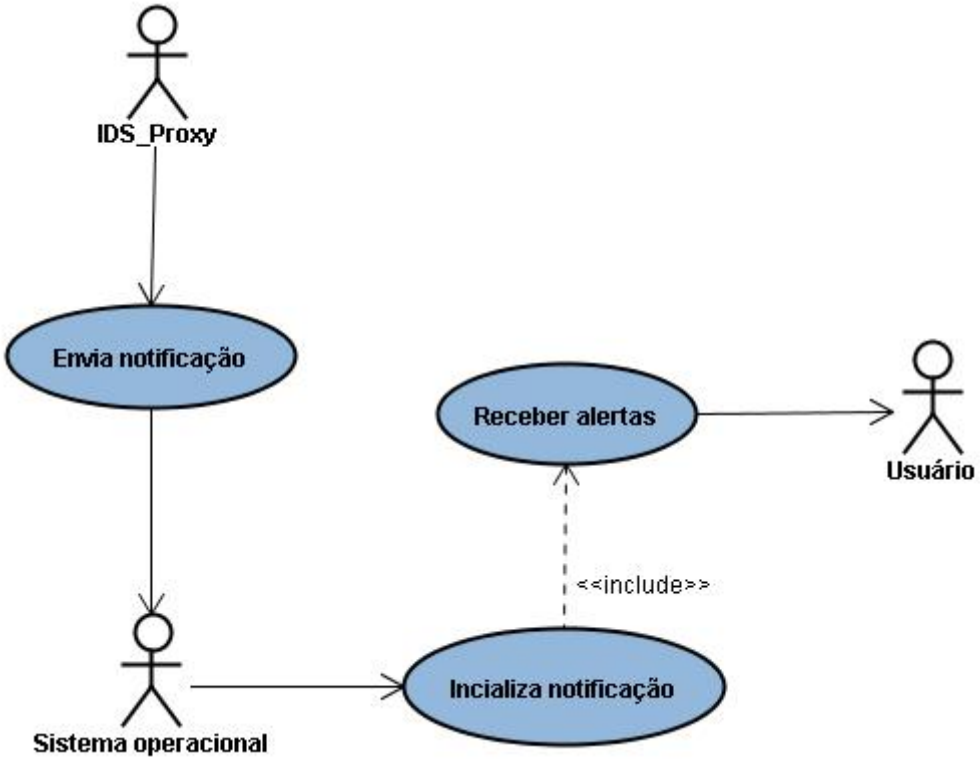


Figura 3.4 - Diagrama de caso receber alertas

Tabela 3.3 - Descrição do caso de uso receber alertas

	Caso de Uso: Receber alertas
Atores	*IDS_Proxy*, Sistema Operacional e Usuário
Descrição	Receber notificações do *IDS_Proxy*.
Pré-Condições	Conexões de rede disponível para comunicação com o servidor
Fluxo Principal	Informa ao cliente através de notificações registradas no sistema operacional sobre qualquer ameaça identificada para dispositivo, tentativas de ataque e ataques ocorridos.
Subfluxos	Inicializar o serviço de notificação através da notificação enviada pelo *IDS_Proxy*.
Fluxos Excepcionais	Não disponibilidade de conexão para recebimento das notificações

A Figura 3.5 apresenta o caso de executar contramedida. O detalhamento do mesmo é descrito na Tabela 3.4.

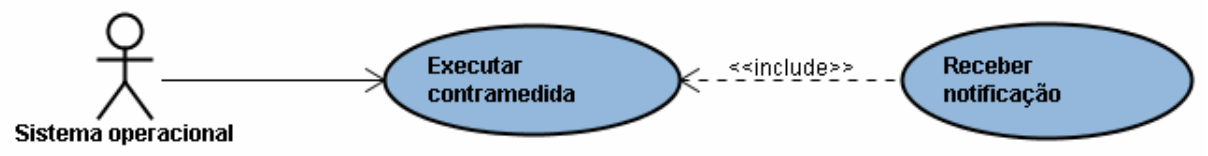


Figura 3.5 - Diagrama de caso de executar contramedida

Tabela 3.4 - Descrição do caso de executar contramedida

	Caso de Uso: Executar contramedida
Atores	Sistema Operacional e Usuário
Descrição	Executar contramedida em relação a um determinado ataque ou ameaça.
Pré-Condições	Conexões de rede disponível para comunicação com o *IDS_Proxy*
Fluxo Principal	Após notificação ao sistema operacional sobre uma determinada ameaça o sistema operacional executa a contramedida necessária, como por exemplo, executar fechamento da conexão de rede.
Subfluxos	Recebimento de notificação sobre ameaça identificada pelo *IDS_Proxy*.
Fluxos Excepcionais	Não disponibilidade de conexão para envio das informações.

3.2.2 IDS_Proxy

O *IDS_Proxy* é a estação centralizadora das informações capturadas na rede *wireless* e dos dados enviados pelos dispositivos móveis cadastrados no sistema. O *IDS_Proxy* atua promiscuamente na rede capturando informações que tenham como origem e destino os dispositivos cadastrados no sistema. A análise das informações é baseada nas características geradas a partir dos perfis dos dispositivos móveis. Estas informações são obtidas em situações consideradas normais para o dispositivo e através do tráfego de rede para aqueles dispositivos. As contramedidas dependendo da capacidade do dispositivo e da intensidade do ataque podem ser executadas pelo próprio *IDS_Proxy*.

A Figura 3.6 apresenta o caso de uso capturar *frames*. As especificações sobre o caso de captuarar *frames* são descritas na Tabela 3.5.

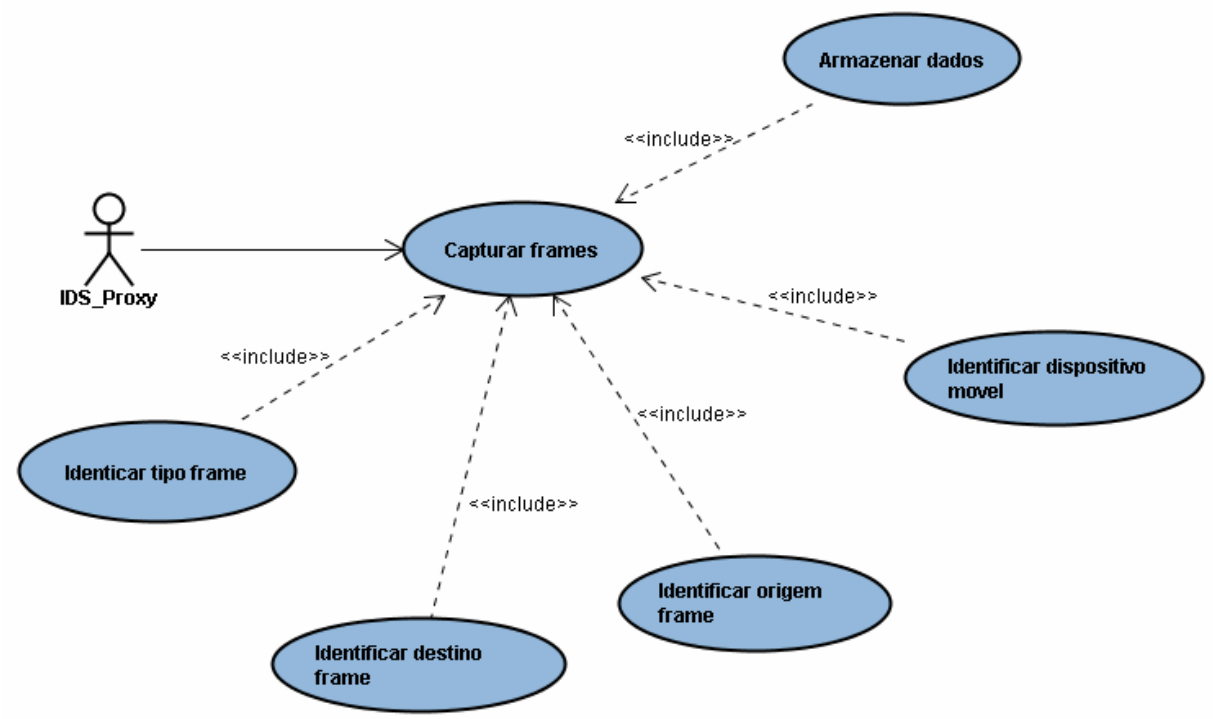


Figura 3.6 - Diagrama de caso capturar frames

Tabela 3.5 - Descrição do caso de capturar frames

	Caso de Uso: Capturar *frames*
Atores	*IDS_Proxy*
Descrição	Capturar tráfego na rede *wireless* promiscuamente.
Pré-Condições	Interface de rede sem fio que possa atuar em modo promíscuo.
Fluxo Principal	Após a inicialização do *IDS_Proxy* este fica capturando promiscuamente todo trafégo *wireless* destinado e originado dos dispositivos cadastrados e armazena os dados necessários para uma posterior auditoria.
Subfluxos	Analisar assinatura de ataques para comparação com tráfego destinado aos dispositivos móveis.
Fluxos Excepcionais	Erro na execução do *IDS_Proxy*.

A Figura 3.7 apresenta o caso de uso receber dados. O detalhamento das informações sobre atores e fluxos produzidos pelo caso de uso são descritos na Tabela 3.6.

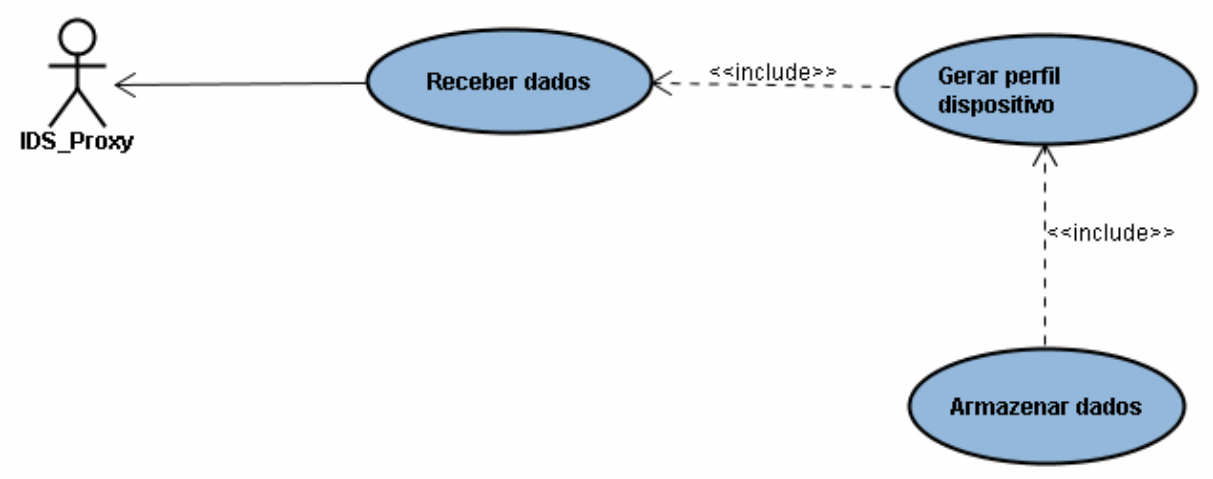


Figura 3.7 - Diagrama de caso receber dados

Tabela 3.6 - Descrição do caso de receber dados

	Caso de Uso: Receber dados
Atores	*IDS_Proxy*
Descrição	Receber informações relativas ao perfil dos dispositivos móveis .
Pré-Condições	Disponibilidade do serviço para recebimento das informações.
Fluxo Principal	Receber as informações capturadas pelo módulo *IDS_Client* e armazená-las para análise e formação do perfil do dispositivo móvel.
Subfluxos	Gerar perfil do dispositivo com as informações enviadas, como novos programas instalados, memória disponível, carga e voltagem da bateria que serão armazenadas para formação de histórico do comportamento do dispositivo.
Fluxos Excepcionais	Problemas na execução do serviço para recebimento de dados.

A Figura 3.8 apresenta o caso de uso identificar ataque. Os atores e descrição dos fluxos gerados pelo caso de uso são detalhados na Tabela 3.7.

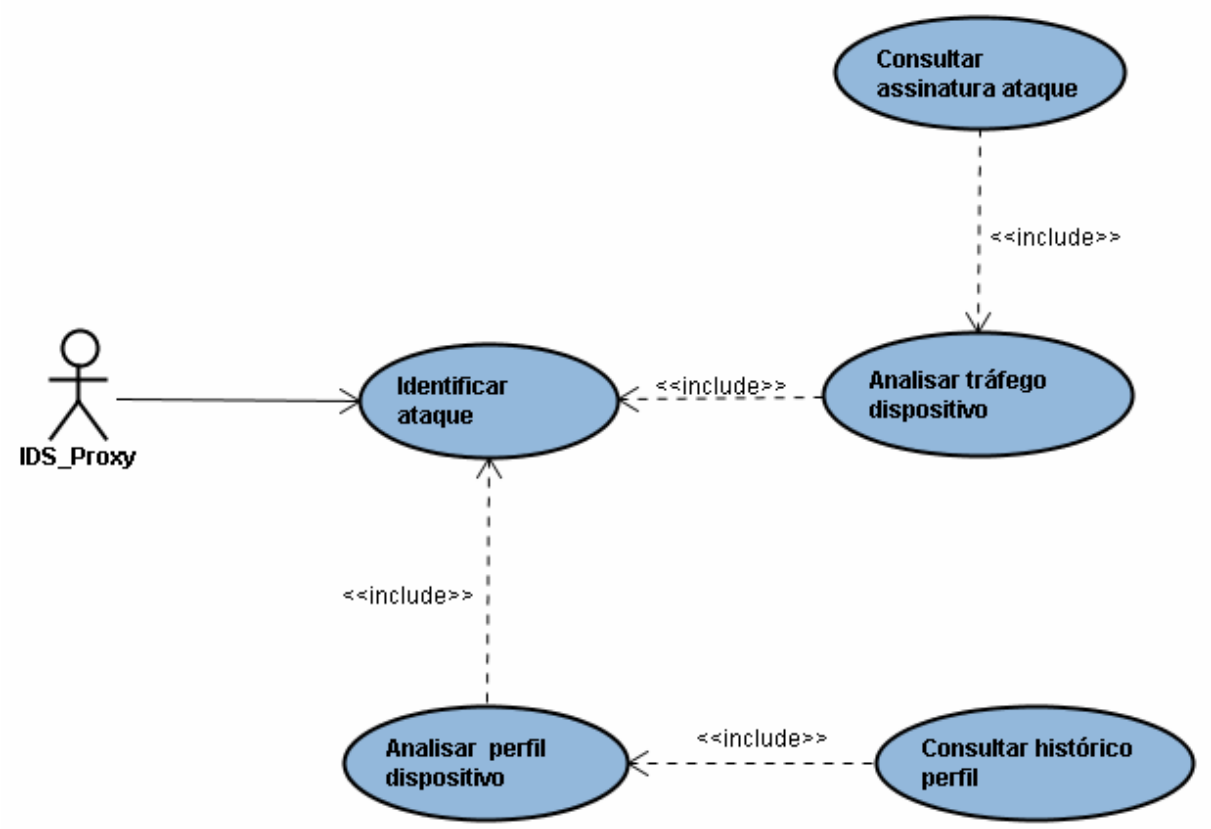


Figura 3.8 - Diagrama de caso identificar ataque

Tabela 3.7 - Descrição do caso de identificar ataque

	Caso de Uso: Identificar ataque
Atores	*IDS_Proxy*
Descrição	Identificar ataques ou qualquer tentativa dos mesmos.
Pré-Condições	
Fluxo Principal	Analisar o trafégo capturado na rede *wireless* juntamente com as informações enviadas pelo dispositivo móvel e identificar o ataque correlacionando estas informações com o banco de assinaturas de ataques e histórico do perfil do dispositivo móvel.
Subfluxos	Consulta ao banco de assinatura de ataques. Consulta ao histórico do perfil do dispositivo.
Fluxos Excepcionais	

A Figura 3.9 apresenta o caso de uso executar contramedida. As especificações sobre os atores envolvidos, assim como fluxos gerados pelo caso são detalhados na Tabela 3.8.

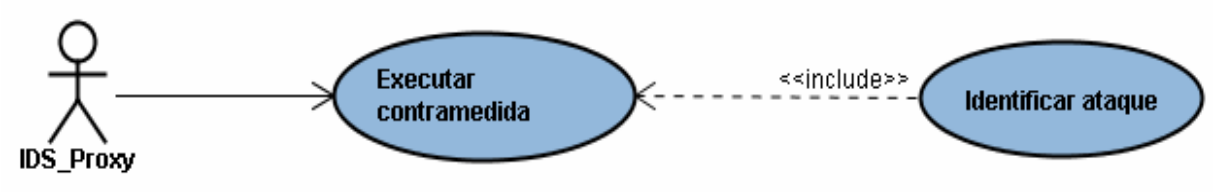


Figura 3.9 - Diagrama de caso executar contramedida

Tabela 3.8 - Descrição do caso de executar contramedida

	Caso de Uso: Executar contramedida
Atores	*IDS_Proxy*
Descrição	Executar contramedida através do próprio *IDS_Proxy*
Pré-Condições	
Fluxo Principal	Executar a contramedida contra um ataque quando dispositivo não possui condições de executar a mesma seja por falta de recursos do próprio dispositivo ou devido à intensidade de ataque
Subfluxos	Identificação do ataque baseado nas informações sobre o perfil do dispositivo e trafego de rede para o mesmo.
Fluxos Excepcionais	

3.3 DIAGRAMAS DE CLASSES

O diagrama de classes mostra um conjunto de classes, interfaces e colaborações e seus relacionamentos. Os diagramas de classes são usados para fazer a modelagem da visão estática do projeto de um sistema. Na maioria dos casos, envolve a modelagem do vocabulário do sistema ou a modelagem de esquemas.

3.3.1 IDS_Client

A modelagem para o *IDS_Client* foi baseada nas funções fornecidas pelo ambiente de desenvolvimento para aplicativos da Palm OS (PALM, 2008), o ACESS (ACESS, 2007) , uma vez que para o desenvolvimento do aplicativo utilizou-se uma linguagem procedural. Desta forma, a funcionalidade destas funções utilizadas e como estas se relacionam será descrito. A Figura 3.10 apresenta a classe *SystemDiagnostic* implementada no desenvolvimento do *IDS_Client.*

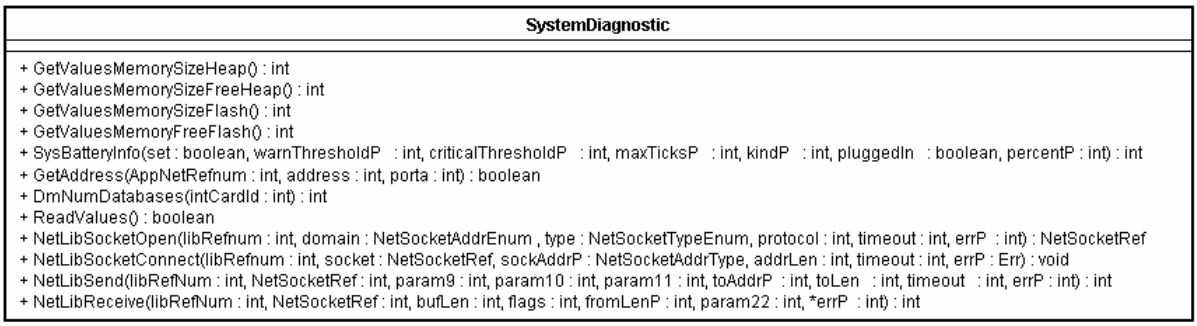


Figura 3.10 - Funções utilizadas no desenvolvimento IDS_Client

Os métodos da classe *SystemDiagnostic* são descritos a seguir:

- *GetValuesMemorySizeHeap* retorna a quantidade de memória usada por aplicações em execução no dispositivo;
- *GetValuesMemorySizeFreeHeap* retorna a quantidade de memória livre daquela destinada a aplicações em execução;
- *GetValuesMemorySizeFlash* retorna a quantidade de total de memória RAM do dispositivo;
- *GetValuesMemoryFreeFlash* retorna a quantidade de memória disponível;
- *SysBatteryInfo* é responsável pelo leitura sobre informações da bateria do dispositivo com quantidade de carga restante, voltagem de bateria atual, voltagem mínima permitida;
- *GetAddress* responsável pela passagem dos parâmetros para comunicação com servidor para envio das informações capturadas;
- *DmNumDatabases* retorna a quantidade de aplicações instaladas no dispositivo;
- *DmDatabaseInfo* responsável por consulta sobre informações mais específicas de cada aplicativo instalado;
- *NetLibSocketOpen* faz abertura do *socket* para uma posterior comunicação;
- *NetLibSocketConnect* é responsável pela realização da conexão com o servidor;
- *NetLibSend* é responsável pelo envio da informação ao servidor;
- *NetLibReceive* é responsável pela recepção de qualquer informação enviada ao dispositivo através de *sockets*.

3.3.2 Diagrama de Classes IDS_Proxy

A Figura 3.11 apresenta o diagrama de classes para o *IDS_Proxy*.

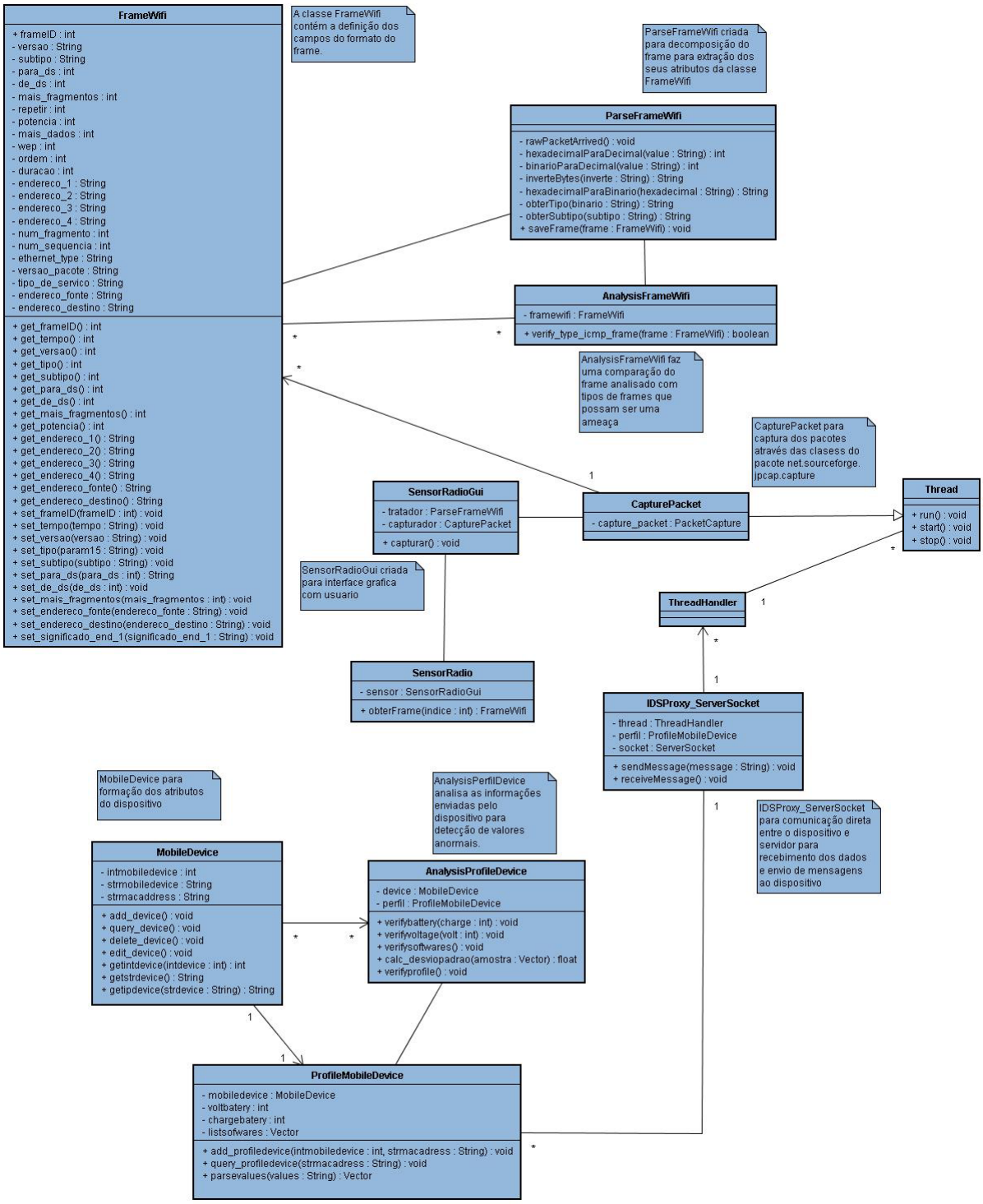


Figura 3.11 - Diagrama de Classes para o IDS_Proxy

A classe *FrameWifi* representa os *frames* capturados na rede *wireless*, os atributos da classe foram definidos de acordo com o formato do quadro da camada MAC para o padrão IEEE 802.11b. A classe *ParseFrameWifi* faz o tratamento do *frame* capturado para aqueles atributos representados na classe *FrameWifi*, ou seja, faz o tratamento do pacote para extração das informações referentes as camadas de enlace, rede e transporte de acordo modelo OSI[8] (DAY, 1983). A classe *AnalysisFrameWifi* faz a análise do *frame* para com os alguns tipos já identificados que possuem um comportamento padrão.

A classe *CapturePacket* é responsável pela captura dos *frames* utilizando a classe *PacketCapture* do pacote *net.sourceforge.jpcap.capture*, que é a classe núcleo da captura de pacotes da biblioteca *Jpcap (*JPCAP, 2007*)*. Ela provê uma interface de alto nível para a captura de pacotes de rede através do encapsulamento da biblioteca *Libpcap (*LIBPCAP, 2007*)* e herda o comportamento da classe Thread do pacote *java.lang* para que captura continue concorrente com as outras atividades do processador.

A classe *SensorRadio* possui como um de seus componentes a classe *SensorRadioGUI* que representa a interface gráfica do sistema para visualização dos *frames* capturados e das informações enviadas pelos dispositivos.

A classe *MobileDevice* define os atributos para identificação do dispositivo móvel que serão usados para identificação e na formação do perfil dispositivo. O perfil do dispositivo é formado pela classe *ProfileMobileDevice* através dos dados recebidos pela classe *IDSProxy_ServerSocket* a qual é responsável pela comunicação direta entre o dispositivo e o servidor. A classe *AnalysisPerfilDevice* faz a análise das informações atuais do dispositivo comparando estas informações com perfil do dispositivo considerado para padrão de normalidade.

3.4 DIAGRAMAS DE SEQÜÊNCIA

O diagrama de seqüência é um diagrama de interação que dá ênfase à ordenação temporal de mensagens. Estes diagramas têm como objetivo mostrar como as mensagens entre os objetos são trocadas no decorrer do tempo para a realização de uma operação.

---

[8] *Open Systems Interconnection*

A seqüência de atividades para o envio e processamento de dados monitorados no dispositivo móvel é mostrada na Figura 3.12[9]. Como ilustrado na Figura 3.12, a aplicação precisa ser inicializada no dispositivo (1) para leitura dos dados necessários (3) a serem enviados para o servidor (6) que fará o tratamento dos dados (7) e adicionará estes valores ao perfil do dispositivo (9) para verificação dos dados atuais com o perfil padrão do dispositivo (10). A verificação inclui a quantidade de bateria para o dispositivo (11), a voltagem da mesma (12) e programas instalados no dispositivo (13).

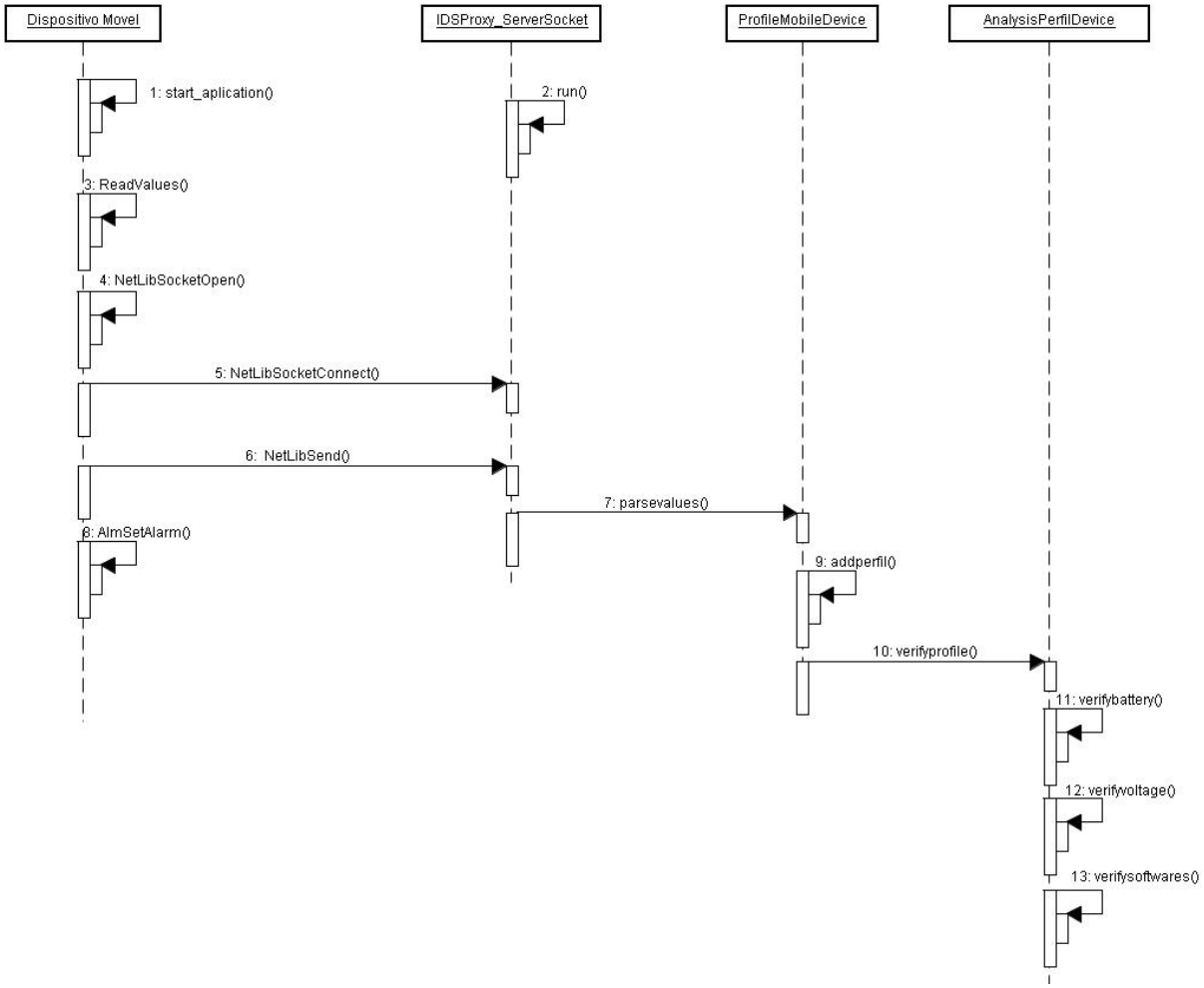


Figura 3.12 - Diagrama de Seqüência para o envio e processamento de dados monitorados no dispositivo móvel

A Figura 3.13 apresenta o diagrama de seqüências para captura de analise de pacotes na rede wireless pelo *IDS_Proxy*. O processo tem início com as informações cadastradas sobre os dispositivos a serem monitorados (1). No *IDS_Proxy*, dois serviços são inicializados e executam concorrentemente: serviço para o recebimento de dados enviado

[9] Os números entre parêntenses referem-se a ordem dos acontecimentos dos processos representados nos diagramas de seqüência.

pelos clientes (3) e o serviço para captura de pacotes que ocorre no tráfego da rede *wireless* (4). Para captura de pacotes, estes recebem o tratamento de análise para identificação dos tipos de pacotes e identificar se os mesmos representam alguma ameaça (6). No recebimento dos dados enviados pelo cliente, os dados são tratados e comparados com as informações para o perfil de comportamento normal para o dispositivo (7). O casamento destas informações analisadas sobre o dispositivo juntamente com o tráfego de rede permitem ao sistema inferir se o dispositivo sofre algum tipo de ameaça (8). Assim, o dispositivo é informado através de notificações de *socket* sobre o ataque (9) para que seja tomada a contramedida necessária (10,11).

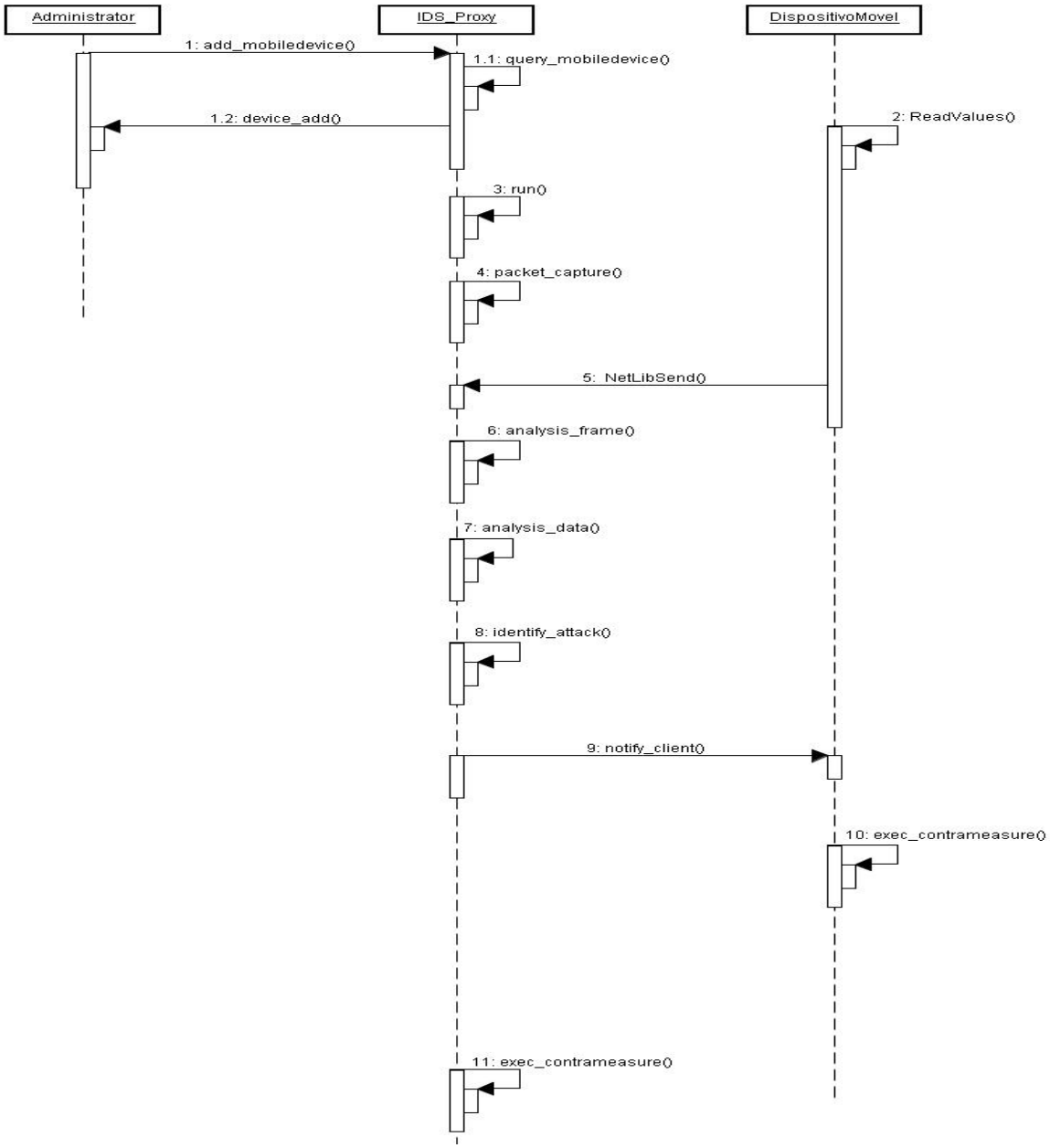


Figura 3.13 - Diagrama de Seqüência para captura e análise de pacotes da rede *wireless* pelo *IDS_Proxy*

3.5 DIAGRAMA DE ATIVIDADES

Os diagramas de atividade são empregados para fazer a modelagem de aspectos dinâmicos do sistema. Na maior parte, isso envolve a modelagem das etapas seqüenciais em um processo computacional.

Um diagrama de atividades é essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra. Uma atividade é uma execução em andamento não-atômica em uma maquina de estados. As atividades efetivamente resultam em alguma ação, formada pelas computações executáveis atômicas que resultam em uma mudança de estado do sistema ou retorno de um valor. As ações abrangem a chamada a outras operações, enviando um sinal, criando um objeto ou alguma computação pura, como o cálculo de uma expressão.

Os diagramas de atividade poderão permanecer isolados para visualizar, especificar, construir e documentar a dinâmica de uma sociedade de objetos, ou poderão ser utilizados para fazer a modelagem do fluxo de controle de uma operação.

Os diagramas de atividades para o *IDS_Client* e *IDS_Proxy* estão representados na Figura 3.14 e na Figura 3.15.

O fluxo de atividades para o *IDS_Client* inicia-se com a instalação da aplicação e execução da mesma que fará a leitura dos dados necessários, logo em seguida inicia a biblioteca para comunicação para abrir conexão com servidor para posterior envio dos dados capturados. Ao realizar todo esse processo com sucesso, o *IDS_Client* faz o agendamento do alarme para a próxima execução da aplicação. Havendo falha logo após a tentativa de conexão pelo *IDS_Client*, a aplicação é finalizada e alarmes não são cadastrados, pois uma vez não existe disponibilidade de conexão não há como fazer o envio de dados para análise dos dados. Assim, para evitar desperdício de consumo pelo dispositivo a aplicação é finalizada.

O fluxo de atividades para o *IDS_Proxy* é bem intenso. Com o início da aplicação, dois serviços são inicializados paralelamente, o primeiro para captura de *frames* na rede *wireless* e logo em seguida o servidor para recebimento dos dados enviados pelos clientes. Logo após a captura de frames, estes são analisados e identificados para comparação com determinados *frames* que representam ameaças. Paralelamente segue a analise dos dados enviados pelos dispositivos que também são analisados e comparados com perfis de normalidade para estes dispositivos. A análise desse conjunto de informações permite ao

sistema inferir e identificar uma possível ameaça que identificada positivamente será notificado ao dispositivo para realização da contramedida necessária.

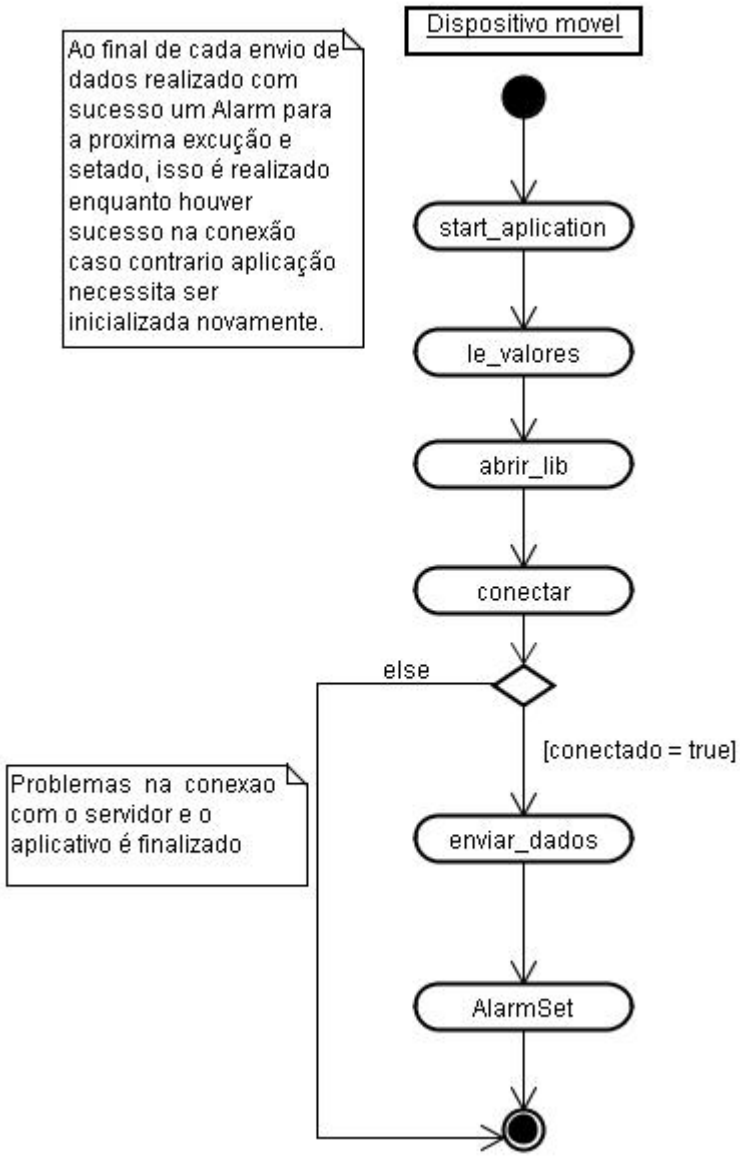


Figura 3.14 - Diagrama de Atividades para o *IDS_Client*

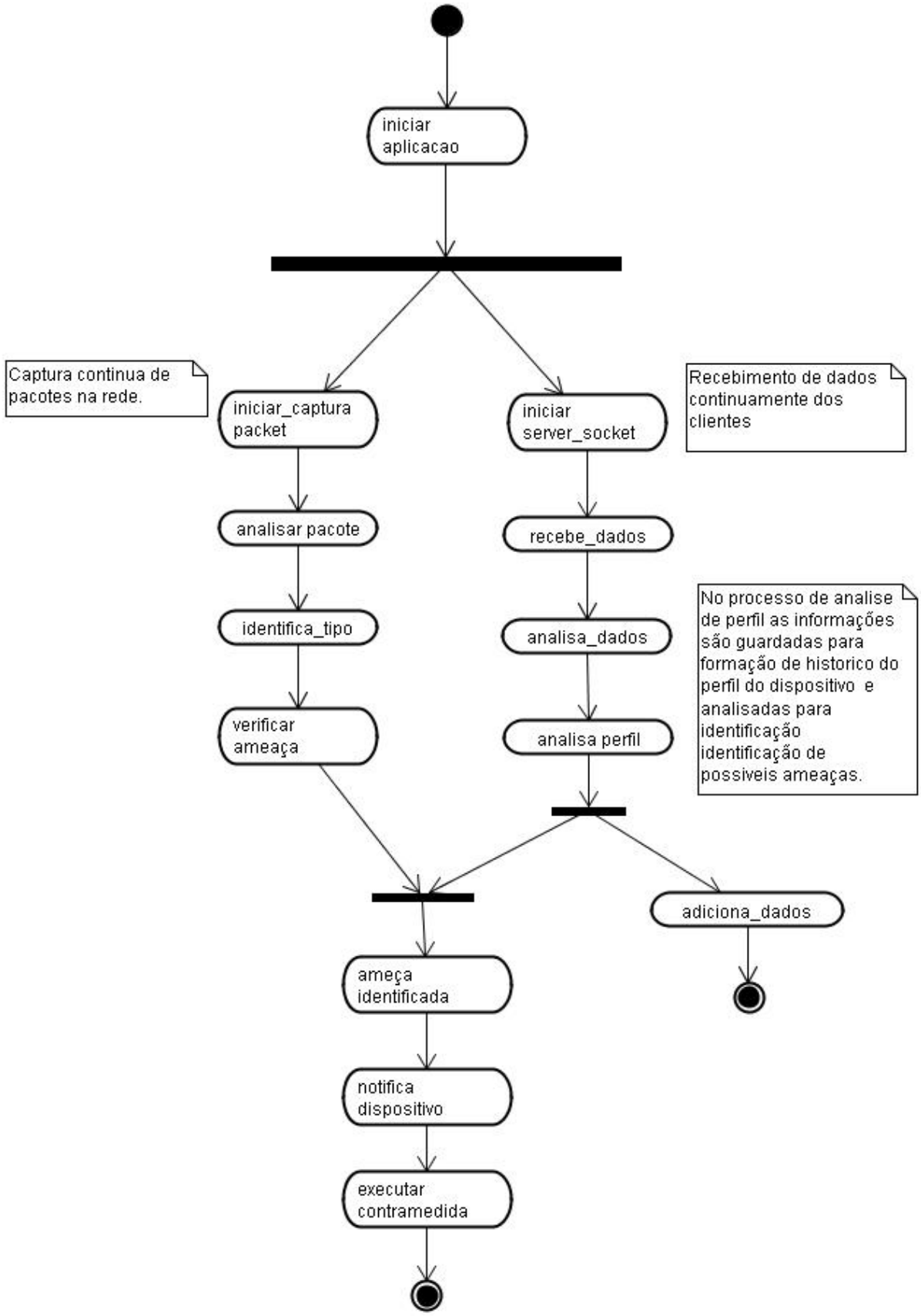


Figura 3.15 - Diagrama de Atividades para o *IDS_Proxy*

3.6 DIAGRAMA DE IMPLANTAÇÃO

O diagrama de implantação mostra a configuração de elementos de processamento em tempo de execução e os componentes de software, processos e objetos que neles são executados. É usado para modelar a arquitetura de distribuição em que o sistema será executado.

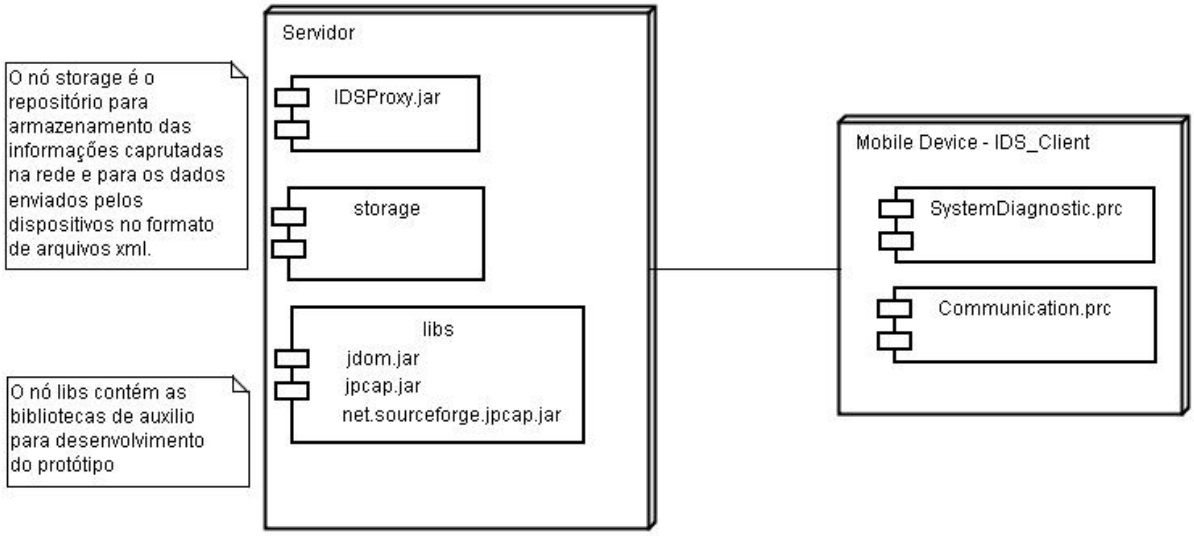


Figura 3.16 - Diagrama de Implantação

Para este protótipo, um arquivo de extensão *IDS_Proxy.jar* foi criado, reunindo os componentes do sistema. A aplicação é inicializada a partir da execução deste aplicativo apresentando uma interface gráfica ao usuário. O nó *storage* serve como repositório das informações capturadas através do tráfego da rede e as informações que são enviadas pelo dispositivo móvel através de outro nó *SystemDiagnostic.prc* que é a aplicação que executa no cliente responsável pela leitura das informações sobre o dispositivo móvel. A aplicação *Communication.prc* é responsável pelo fechamento das conexões de rede na identificação de ameaças destinada ao despositivo. O nó *libs* é que armazena as bibliotecas auxiliares no desenvolvimento do protótipo, onde *jdom.jar* é para manipulação dos arquivos no formato *xml* e *jpacap.jar* e *netsourceforge.jpacap.jar* são responsáveis pela captura de tráfego na rede *wireless.*

3.7 CONCLUSÃO

Neste capítulo, apresentou-se a modelagem da arquitetura proposta, descrevendo suas funcionalidades através dos diagramas UML (*Unified Modeling Language*). Utilizando esta abordagem, permitiu-se analisar a arquitetura sobre diferentes perspectivas, demonstrando os aspectos estáticos e dinâmicos e como estes se relacionam.

Para visualização das funcionalidades e processos envolvidos na execução do protótipo, os diagramas de caso de uso, seqüência e atividades foram modelados. As estruturas das informações capturadas e armazenadas foram detalhadas nos diagramas de classe da solução.

Utilizando as informações obtidas nesta fase de identificação de requisitos e funcionalidades, o desenvolvimento do protótipo, foi possível, tal como é descrito no próximo capítulo.

4 ESTENDENDO O IDS-NIDIA

Este capítulo tem como propósito apresentar a proposta de uma arquitetura para sistemas de detecção de intrusões para usuários de dispositivos móveis. Esta arquitetura é uma extensão do IDS NIDIA. Uma visão geral da arquitetura, seu funcionamento e sua aplicabilidade serão apresentadas.

4.1 PROJETO NIDIA

A proposta do sistema NIDIA (LIMA, 2001) é apresentar um sistema de detecção de intrusão, composto por um conjunto de agentes, fornecendo um modelo de detecção de intrusos, em tempo real, baseado na noção de sociedade de agentes capaz de detectar novos ataques através de uma rede neural.

O NIDIA é inspirado no modelo CIDF (*Common Intrusion Detection Framework*) (STANIFORD-CHEN, 1998), possuindo para esta finalidade agentes com a função de geradores de eventos (agentes sensores), mecanismos de análise dos dados (agentes de monitoramento e de avaliação de segurança), mecanismos de armazenamento histórico (base de dados) e um módulo para realização de contramedidas (agente controlador de ações). Além disso, existem agentes responsáveis pela integridade do sistema e pela coordenação das atividades do IDS como um todo.

Desta forma, os agentes do NIDIA possuem os seguintes objetivos gerais: gerar índices de suspeita de ataque a partir da análise de dados coletados de *logs* de *host* e de pacotes de tráfego na rede; executar contramedidas de acordo com os índices obtidos; aprender com os casos obtidos atualizando suas bases de conhecimento.

O modelo proposto provê a metodologia de detecção por abuso e anomalia para garantir uma maior robustez ao sistema.

A escolha da arquitetura multiagentes para o IDS-NIDIA tem os seguintes objetivos:

- Agentes podem ser adicionados ou removidos para/do sistema sem modificar outros componentes do sistema;
- Agentes podem ser reconfigurados ou atualizados sem causar problemas ao restante do sistema;

- Um agente ou um grupo de agentes podem realizar diferentes funções simples. Visto que os agentes podem trocar informações entre si, podem derivar resultados mais complexos.

O objetivo específico de cada agente que compõe o IDS-NIDIA é demonstrado através da sua arquitetura em camadas.

Arquiteturalmente, o NIDIA é composto por camadas. Cada camada possui atividades a desempenhar, sendo que estas atividades são executadas através do comportamento dos agentes que a compõe. É através destes agentes também que as camadas se comunicam trocando informações importantes para desempenhar suas atividades. A Figura 4.1 apresenta a arquitetura do NIDIA.

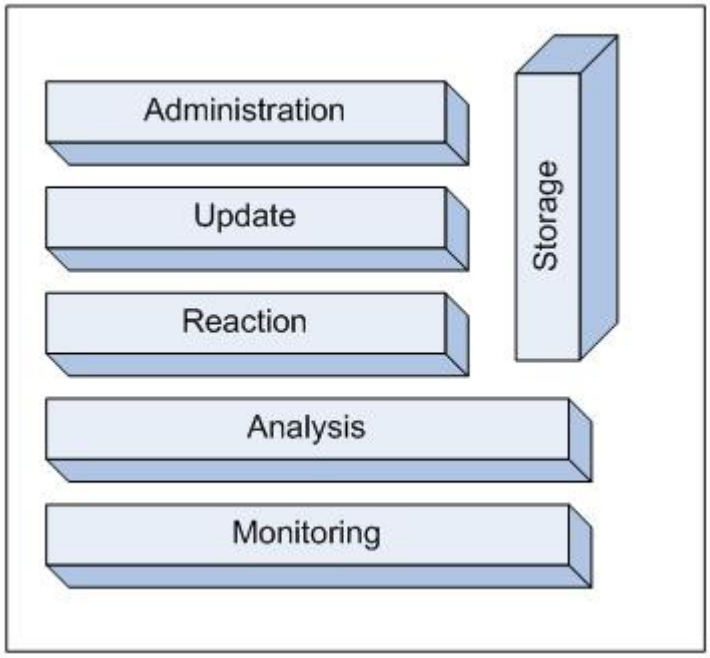


Figura 4.1 - Modelo em camadas do NIDIA

A descrição das funcionalidades das camadas do NIDIA e os seus respectivos agentes que a compõe seguem a seguir.

- **Camada de Monitoramento (*Monitoring*):** responsável por capturar a ocorrência de eventos no meio exterior e fornecer informações sobre o mesmo para o resto do sistema. Nesta camada, os agentes SMA (*System Monitoring Agent*) estão localizados. Estes agentes funcionam como “sentidos receptores” do sistema. Os agentes SMAs dividem-se em duas categorias:

  - Agentes sensores de rede: responsáveis por capturar os pacotes que estão trafegando na rede. Estes atuam em pontos estratégicos da rede e funcionam como monitores de rede passivo, trabalhando em modo promíscuo, desta forma não interferindo na desempenho e nem tráfego da rede;
  - Agentes sensores de *host*: trabalham coletando informações em tempo real de um *host* em particular (geralmente servidores) e disponibilizando-as para análise. Os dados obtidos recebem uma pré-formatação sendo em seguida repassado para o agente de avaliação de segurança.

- **Camada de Análise (*Analysis*):** responsável pela análise dos eventos recebidos da camada de monitoramento. Nesta camada, os eventos coletados são formatados de maneira que padrões de ataques possam ser identificados e posteriormente a confirmação de um ataque. Para isso, utilizam-se bases de conhecimento, como a base de dados de padrões de intrusões (IIDB, *Incidents of Intrusion and Forensic Information DataBase*), a base de dados de incidentes de intrusão (DFDB, *Standard of Intruders and Intrusion DataBase*) e a base de estratégias (STDB, *Strategy DataBase*). Nesta camada, localizam-se os agentes SEA (*Security Evaluation Agent*) que são responsáveis por realizar a análise dos eventos coletados e emitir um grau de suspeita sobre os eventos que foram previamente formatados;
- **Camada de Reação (*Reaction*):** responsável por tomar contramedidas caso um incidente de segurança seja detectado. Com base no parecer do SEA, esta camada deve tomar uma contramedida de acordo com as bases de dados de estratégia (STDB) e de ações (RADB, *Reaction DataBase*). Nesta camada localizam-se os agentes SCA (*System Controller Agent*);
- **Camada de Atualização (*Update*):** responsável pela atualização das bases de informações. As consultas poderão ser feitas diretamente de qualquer camada, porém inserções devem ser feitas somente através desta camada. Ela terá também a responsabilidade de manter a integridade e consistência das informações armazenadas. Nesta camada, localizam-se os agentes SUA (*System Updating Agent*). Estes são responsáveis pela atualização das bases DFDB, IIDB, RADB e STDB;

- **Camada de Administração (*Administration*):** responsável pela administração e integridade de todos os agentes do sistema. Nesta camada, localizam-se os agentes MCA (*Main Controller Agent*);
- **Camada de Armazenamento (*Storage*):** responsável por manter de forma persistente informações provenientes das demais camadas. Nesta camada, localizam-se as bases de dados utilizadas pelo NIDIA. Segue uma descrição das mesmas:
    - STDB (*Strategy DataBase*) é a base de dado responsável por registrar as estratégias adotadas por uma organização qualquer em relação à sua política de segurança. Ela é importante para garantir a adaptabilidade do IDS em diversos casos;
    - RADB (*Reaction DataBase*) estão contidas as informações referentes às ações que devem ser tomadas de acordo com a severidade do ataque detectado. Também varia de acordo com a política de cada instituição;
    - IIDB (*Incidents of Intrusion and Forensic Information DataBase*) registra os danos causados por ataques bem-sucedidos e tentativas de ataques. Este contém informações que podem ser úteis na identificação de tentativas de ataques provenientes de uma mesma origem de uma mesma origem ou domínio ou simplesmente serem usadas em investigações futuras.

4.1.1 Funcionamento

O funcionamento do sistema NIDIA pode ser demonstrado através da interação entre os agentes que compõe suas camadas. Cada camada dentro do NIDIA possui um objetivo bem definido. Este objetivo é alcançado através da interação e troca de informações entre as camadas. Para um melhor entendimento, o funcionamento do IDS-NIDIA está divido em três partes: funcionamento da captura, funcionamento da análise e funcionamento das contramedidas.

No funcionamento da captura, os agentes sensores de *host e* rede são os responsáveis por capturar a ocorrência de eventos e fornecer informações sobre os mesmos para o resto do sistema.

Os agentes sensores de *host* fazem leitura dos *logs* de segurança de servidores específicos e repassam para o agente SMA de host para que o mesmo formate os dados. Posteriormente tais informações são enviadas para o agente SEA de *host* para que seja feita a avaliação dos dados provenientes do SMA.

Os agentes sensores de rede capturam o tráfego de rede e a partir do tráfego capturado enviam o cabeçalho dos pacotes para os agentes SMA de rede específicos para formatar esse tipo de informação e enviam os pacotes contendo dados sobre as conexões para os agentes SMA de rede específico para formatar esse tipo de informação.

No funcionamento de análise, o processo é realizado pelos agentes SEA. Estes agentes têm como funcionalidade tratar o cabeçalho dos pacotes através de filtros, auxiliado pelas bases de informações pertencentes ao NIDIA.

O funcionamento da contramedida é baseado no produto gerado pelos agentes SEA que é o grau de suspeita dos dados verificados e enviados para o Agente Controlador de Ações, para caso necessário, tomar alguma contramedida baseada no STDB. Para os ataques não registrados nas bases do NIDIA, o SCA solicita que as atualizações sejam feitas a fim de registrar informações sobre o ataque no DFDB.

4.2 ADAPTAÇÃO DA ARQUITETURA PROPOSTA AO NIDIA

Originalmente, o NIDIA é um IDS projetado para ser executado em um ambiente com uma rede local, ou seja, as funcionalidades foram idealizadas para prover serviços de proteção e detecção de intrusão em uma rede local para servidores.

Durante o seu ciclo de evolução, várias propostas foram inseridas em seu modelo devido ao constante crescimento e variedade de ataques e as mudanças ocorridas na estrutura física das redes locais. A expansão das redes *wireless* e como elas se adaptaram às redes locais contribuíram para necessidade de novas adaptações.

A arquitetura proposta tem como objetivo proporcionar um ambiente seguro para usuários de dispositivos móveis que utilizando as informações sobre o monitoramento do tráfego de rede destinado a estes dispositivos. Assim, pretende-se identificar comportamentos intrusivos tendo como base o processo de análise dessas informações e a execução das contramedidas seja um processo realizado em tempo real.

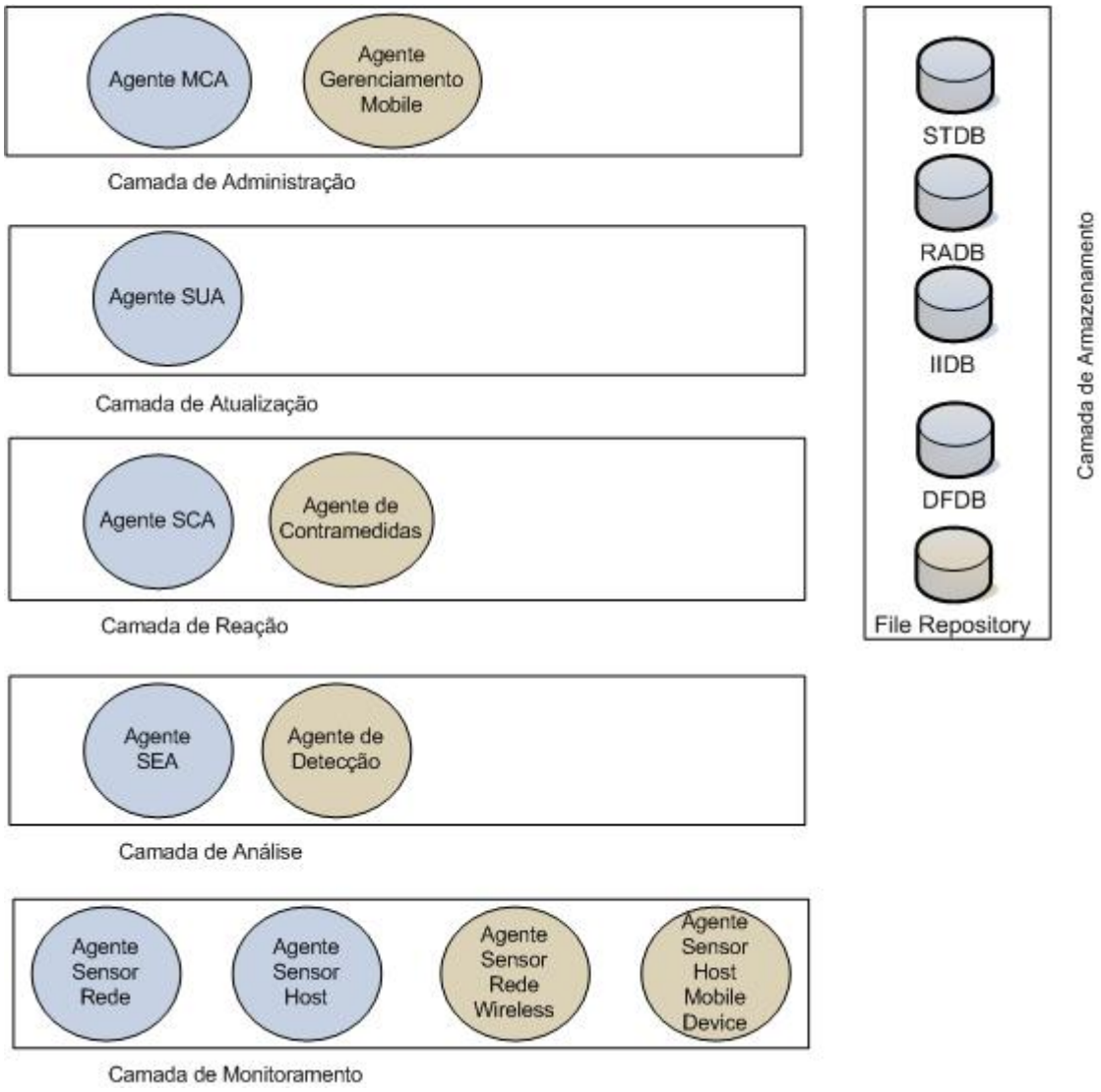


Figura 4.2 - Integração da Arquitetura Proposta ao NIDIA.

A integração da arquitetura torna-se viável devido a ambas as arquiteturas são estruturadas em camadas que possuem funcionalidades bastante similares nos dois modelos. Cada módulo da arquitetura proposta pode ser mapeado em um agente inteligente a ser inserido na sua respectiva camada na arquitetura do NIDIA. Na Figura 4.2 os elementos da cor azul representam os agentes e banco de dados pertencentes ao NIDIA, enquanto que os elementos na cor marrom representam os agentes e o repositório de arquivos da arquitetura proposta.

O Agente Sensor Rede Wireless e o Agente Sensor Host Mobile Device são inseridos na camada de monitoramento juntamente com o Agente Sensor Rede e o Agente Sensor Host. O Agente de Detecção é inserido na camada de análise com inclusão de detecção

no ambiente *wireless* e identificação dos comportamentos anormais dos dispositivos monitorados juntamente com a Agente SEA. O Agente de Contramedidas são inseridos no NIDIA na camada de reação juntamente com agente SCA, com a capacidade de inclusão da capacidade de decidir com as contramedidas apropriadas, além de executar ações efetivas contra intrusões no ambiente *wireless.* O Agente Gerenciamento Mobile é inserido no NIDIA na camada de administração juntamente com o Agente MCA. Isso permite ao NIDIA o gerenciamento dos dispositivos monitorados. E finalmente na camada de armazenamento *File Repository* será inserido para armazenamento das informações capturadas.

A arquitetura é baseada no modelo cliente-servidor, utilizando as técnicas de detecção de um IDS híbrido, que faz coleta de informações sobre o dispositivo móvel e ao mesmo tempo faz monitoramento do tráfego de rede. Desta forma, foram definidas as arquiteturas para o *IDS_Client* e *IDS_Proxy* apresentadas na Figura 4.3.

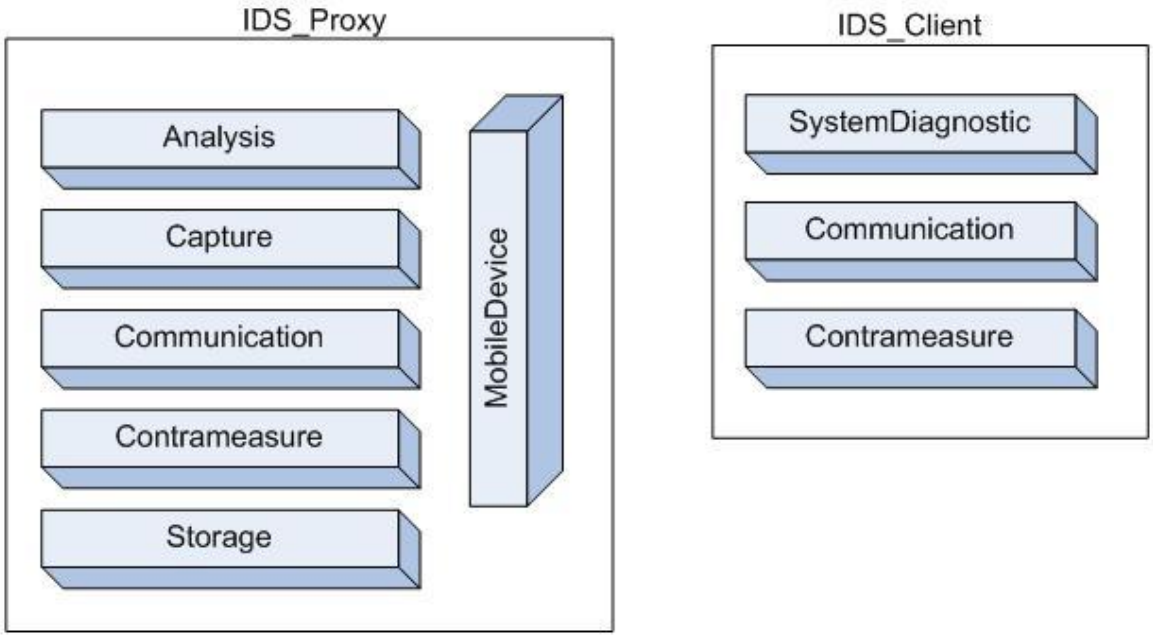


Figura 4.3 - Modelo em camadas da arquitetura proposta

O *IDS_Proxy* é um sensor que atua na rede coletando informações sobre o tráfego de rede dos dispositivos monitorados, recebe informações originadas destes dispositivos para análise e identificação de comportamentos anormais e faz o armazenamento destes dados para um possível processo de auditoria e armazenamento do registros de ataques para que os mesmos possam ser identificados novamente.

O *IDS_Client* é o aplicativo instalado no dispositivo móvel que faz a coleta das informações sobre o mesmo e as envia para o *IDS_Proxy* que faz a análise das informações

atuais com as armazenadas para identificação de variações bruscas de comportamento do dispositivo.

Os módulos *IDS_Proxy* e *IDS_Client* são compostos por camadas que desempenham funcionalidades específicas dentro do processo de detecção. A descrição das funcionalidades segue a seguir.

O IDS_Proxy é composto pelas seguintes camadas:

- **Camada de análise (*Analysis*):** é responsável pela análise das informações para identificação de possíveis ataques tendo como base as informações obtidas na camada de captura e os dados enviados pelos dispositivos móveis monitorados;
- **Camada de captura (*Capture*):** responsável pela captura do tráfego na rede *wireless*. O processo de captura é realizado por um sensor que atua promiscuamente na rede;
- **Camada de comunicação (*Communication*):** responsável pela comunicação direta entre o servidor e o cliente (dispositivo móvel). Um servidor de *sockets* fica em "escuta" aguardando os pedidos de conexões e envio das informações pelos dispositivos. Este camada também é responsável pelo envio das notificações ao dispositivo caso alguma ameaça seja identificada para o mesmo;
- **Camada de contramedidas (*Contrameasure*):** camada de reação na identificação de um possível ataque, contém as medidas a serem tomadas de acordo com o ataque identificado;
- **Camada de armazenamento (*Storage*):** responsável pelo armazenamento das informações do tráfego de rede capturado e das informações enviadas pelos dispositivos móveis. Estas informações são armazenadas em arquivos de extensão XML que poderão ser consultadas posteriormente em uma possível situação de auditoria e identificação dos ataques, caso estes ocorram novamente;
- **Camada *MobileDevice*:** responsável pela administração das informações sobre os dispositivos monitorados. As funcionalidades desta camada incluem cadastramento de dispositivos, consultas e geração de perfis de comportamento que serão utilizados no processo de análise para identificação de possíveis comportamentos fora do padrão normalidade.

O *IDS_Client* é composto pelas seguintes camadas:

- **Camada de diagnostico do sistema (*SystemDiagnostic*):** responsável pela leitura das informações sobre o dispositivo monitorado. As informações obtidas são: nível de bateria, voltagem da bateria, quantidade de memória disponível e ocupada e os aplicativos instalados. Estas informações são enviadas ao *IDS_Proxy* e analisadas para identificação de possíveis comportamentos anormais do dispositivo;
- **Camada de comunicação (*Communication*):** responsável pela comunicação direta entre o dispositivo e o servidor para envio das informações, utilizadas no processo de análise e recebimento de notificações, em situações de detecção de ameaças;
- **Camada de contramedidas (*Contrameasure*):** camada de reação na identificação de um ataque. Suas ações dependem da informação envida pelo IDS_Proxy que diz ao dispositivo qual ação deve ser executada em reação ao ataque identificado.

4.3 PROTOTIPAGEM

Para o protótipo da arquitetura proposta, *IDS_Client* e *IDS_Proxy* foram implementados.

O *IDS_Proxy* como citado anteriormente, é um sensor para captura das informações sobre o trafégo na rede *wireless.* As funcionalidades do mesmo foram estendidas e adaptadas do trabalho de (ATAÍDE, 2007). As novas funcionalidades inseridas consistiram na identificação dos protocolos da camada de rede, *Internet Protocol (*RFC 791*)* e *Internet Control Message Protocol (*RFC 792), e transporte: *Transmission Control Protocol (*RFC 793*),* do modelo OSI uma vez que a primeira adaptação do NIDIA para redes *wireless (*ATÍDE, 2007*)* destinou-se ao tratamento somente das informações da camada de enlace. O filtro de tráfego foi inserido tendo como baseado na origem e destino, armazenamento destas informações em arquivos de formato XML e módulo responsável pelo monitoramento e análise das informações provenientes dos dispositivos móveis cadastrados e o módulo de monitoramemnto de *host* destinado a dispositivos móveis permitindo ao NIDIA o monitoramento destes no ambiente *wireless*.

O protótipo para *IDS_Client* é responsável pela leitura das informações sobre dispositivo móvel, mantém uma comunicação direta com o servidor para envio das

informações e recebimento de notificações sobre qualquer anormalidade detectada no comportamento do dispositivo.

A implementação ficou divida nas seguintes etapas:

- Pesquisa de ferramentas para desenvolvimento do *IDS_Client* e adaptações no módulo servidor;
- Implementação do *IDS_Client;*
- Adaptação do sensor *IDS_Proxy*;
- Configuração do ambiente de captura;
- Geração de perfis normais para os dispositivos monitorados;
- Geração de ataques aos dispositivos;
- Análise e armazenamento dos registros de ataques;
- Resultados obtidos.

4.3.1 Ferramentas utilizadas

No desenvolvimento do aplicativo para o *IDS_Client,* o ambiente da *ACESS Company Profile (*ACESS, 2007*)* foi utilizado. Esta é uma plataforma de desenvolvimento para dispositivos móveis baseada na linguagem C/C++ para desenvolvimento de aplicativos para dispositivos com sistema operacional da Palm OS (PALM, 2007). O ambiente *ACESS Company Profile* é um ambiente integrado a plataforma de desenvolvimento IDE Eclipse para geração e compilação do código e um simulador para testes de aplicações.

Na adaptação do sensor *IDS_Proxy,* a plataforma de desenvolvimento IDE Eclipse (ECLIPSE, 2008) foi utilizada como ambiente de desenvolvimento para implementação das adaptações necessárias ao *IDS_Proxy*, utilizando a linguagem de programação Java (JAVA, 2007).

Para o armazenamento das informações capturadas, registros de ataques e as informações sobre os dispositivos móveis foi necessário o uso da API JDOM (Java *Document Object Model*), uma biblioteca *open source* para geração e manipulação de arquivos XML.

Para a captura dos pacotes na rede, a API Jpcap (JPCAP, 2007) foi utilizada . A Jpcap é executada sobre a API Libpcap (LIBPCAP, 2007). A *Libpcap* é uma interface independente de sistema para captura de pacotes em nível de usuário. Ela provê um *framework* portável para o monitoramento de rede de baixo nível, em sistemas *Linux*, BSD e derivados do Unix. O Jpcap é uma API para o desenvolvimento de aplicações de captura de

pacotes em Java (SUN, 2008). Além do conjunto de classes Java, ela inclui uma ferramenta para a visualização e análise do tráfego da rede wireless em tempo real.

No desenvolvimento do *IDS_Proxy*, o sistema operacional utilizado foi o Fedora (FEDORA, 2007), que é uma distribuição *Linux* baseada em pacotes RPM (*Red Hat Package Manager*), criada pela *Red Hat* (REDHAT, 2007). Para que o *Linux* pudesse usar a placa de rede sem fio, foi instalado o *MadWifi* (MADWIFI, 2007), que é um *driver open source* bastante avançado e estável disponível para placas de rede wireless baseadas em *chipsets Atheros* (ATHEROS, 2007), que é o caso do Adaptador DWL-G520. Este *driver* suporta os seguintes modos de operação: estação, ponto de acesso, *ad-hoc* e monitor. O modo monitor, também conhecido como modo promíscuo, é o modo utilizado pelo elemento sensor, pois permite que a interface de rede capture todos dos pacotes que trafegam pela rede, inclusive os pacotes não destinados a ela.

Para o acesso a rede cabeada foi utilizado o *access point* DWL-G520 PCI *Wireless* 2.4 GHz *AirPlus Xtreme* G, da D-Link (D-LINK, 2007). Essa placa permite a conexão a um Ponto de Acesso 802.11b ou 802.11g (para o modo infra-estruturado) ou outra placa *wireless* 802.11b ou 802.11g (para o modo *ad-hoc*, em redes ponto a ponto), podendo operar até 108Mbps de velocidade.

O dispositivo móvel usado foi um Palm T|X da *Palm One* com sistema operacional da *Palm OS® Garnet 5.4* com memória de 128 Mb e processador Intel de 312 MHz e como tecnologias sem fio *Wi-fi* 802.11b e *Bluetooth* 1.1.

4.3.2 Implementação do IDS_Client

O *IDS_Client* é um componente que possui três funções principais:

- Capturar informações sobre o dispositivo;
- Enviar as informações capturadas ao *IDS_Proxy*;
- Receber e executar as contramedidas indicadas pelos *IDS_Proxy* após a identificação de riscos.

O protótipo para o *IDS_Client* possui uma interface simples para o usuário com os dados sobre para qual servidor os dados estão sendo enviados com informação sobre o IP (*Internet Protocol*) e porta para conexão, *status* sobre a conexão e envio dos dados e o horário da próxima execução do aplicativo. O que pode ser observado na Figura 4.4.

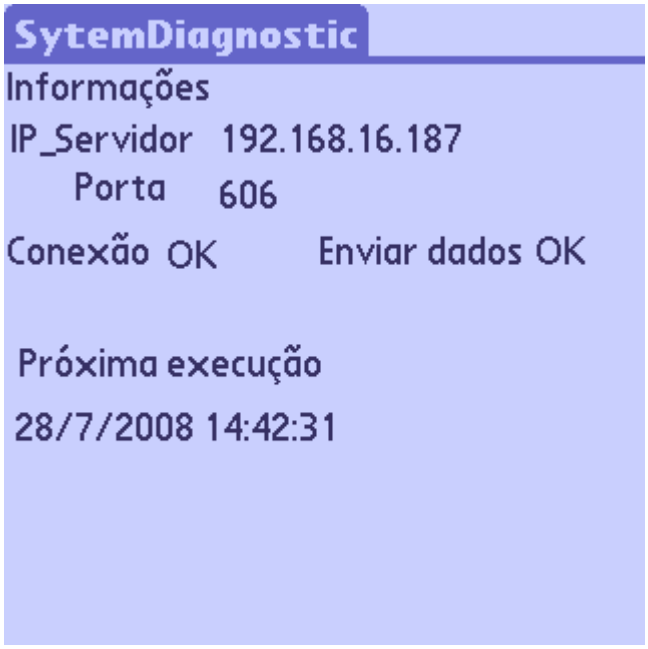


Figura 4.4 - Execução do IDS_Client

O *IDS_Client* é formado por dois aplicativos:

- *SystemDiagnostic.prc*: faz o leitura das informações sobre o dispositivo e as envia para o *IDS_Proxy*. Para obtenção de tais informações algumas funções existentes no ambiente de desenvolvimento *ACESS Company Profile Garnet OS C/C++* foram utilizadas, essas funções são comuns aos dispositivos que tenham *Palm OS* como Sistema Operacional;
- *Communication.prc*: é o aplicativo responsável pelo fechamento da conexão de comunicação da rede *wireless.* Este aplicativo é inicializado através da notificação recebida através do *IDS_Proxy* ao identificar uma ameaça que comprometa as atividades do dispositivo. A Figura 4.5 mostra o momento da execução do mesmo. Também se utilizou API's do ambiente de desenvolvimento ACESS Company Profile Garnet OS C/C++ no desenvolvimento do mesmo.

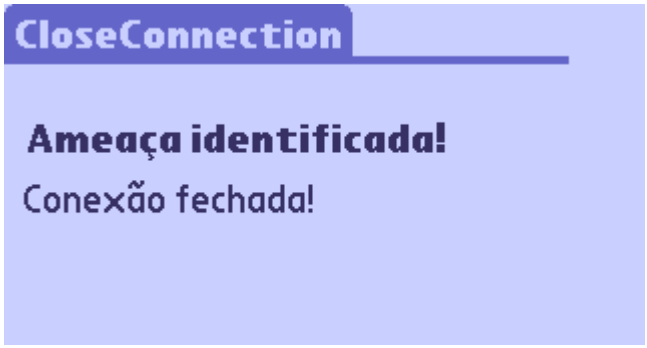


Figura 4.5 - Execução do aplicativo Communication.prc

4.3.3 Implementação do IDS_Proxy

O *IDS_Proxy* é o componente que tem as seguintes funcionalidades:

- Captura do tráfego de rede para os dispositivos monitorados;
- Análise do trafego capturado para identificação de ameaças que possam ocorrer através da mesma;
- Recebimento dos dados enviados pelo dispositivo móvel;
- Análise das informações enviadas pelo dispositivo para identificação de atividades que estejam fora do perfil de normalidades do dispositivo;
- Enviar notificações sobre ameaças ao dispositivo móvel;
- Armazenamento das informações capturadas na rede, registros de ataques e das informações enviadas pelos dispositivos.

O *IDS_Proxy* é uma adaptação e extensão da ferramenta utilizada no trabalho de (ATAÍDE,2007), a mesma foi desenvolvida e utilizada para captura do tráfego da rede *wireless* como inovação dos serviços de detecção de intrusão oferecidos pelo NIDIA em redes *wireless*. A adaptação da ferramenta consistiu das seguintes funcionalidades:

- Análise das informações capturadas na rede *wireless* estendida às camadas de rede e transporte para identificação dos protocolos IP, ICMP e TCP;
- Filtro de tráfego para os dispositivos móveis monitorados e, armazenamento das informações e registros de ataques em arquivos do formato XML;
- Comunicação direta com dispositivo para recebimento dos dados e envio de notificações;
- Análise das informações enviadas pelo dispositivo para identificação de comportamentos irregulares do dispositivo e identificação de registros de

ataques que destinados aos dispositivos móveis que propaguem através da rede;

- O módulo de monitoramento destinado a dispositivos móveis responsável pela captura das informações no dispositivo para uma análise posterior.

Como citado anteriormente, o *IDS_Proxy* foi desenvolvido usando a tecnologia Java com o uso de duas API externas: uma para captura de tráfego na rede, Jpcap, e a outra responsável pelo geração dos arquivos no formato XML para armazenamento das informações capturadas.

Os pacotes criados para implementação do protótipo para o *IDS_Proxy* segue conforme descrito a seguir.

- *br.idsproxy.analysis*: contém as classes para tratamento do tráfego capturado na rede *wireless* é composto por duas classes a *FrameWifi.java* que representa o *frame* capturado e a classe *ParseFrameWifi* que faz a análise e identificação das características dos *frames* capturados;
- *br.idsproxy.capture*: possui as classes que fazem a captura dos *frames* na rede *wireless* e a parte de interface gráfica com usuário e inicialização dos captura. As classes que compõe o pacote são: Associação.java, que identifica os pedidos de associação das estações ao *acess point*. A Classe *CapturePacket.java* é responsável pela captura dos pacotes na rede *wireless*, a classe *SensorRadio.jav* é a ,implementação do sensor para captura e a classe *SensorRadioGui.java* é responsável pela interface gráfica do sensor;
- *br.idsproxy.communication:* pacote que contém a classe *IDSProxy_ServerSocket.java* que faz o tratamento da comunicação direta entre o dispositivo e o *IDS_Proxy* para o recebimento das informações oriundas dos dispositivos e do envio de notificações ao dispositivo.
- *br.idsproxy.mobiledevice*: contém as classes para gerenciamento sobre as informações dos dispositivos cadastrados. Composto pelas classes *MobileDevice.java* que representa o dispositivo móvel com suas características físicas, a classe *ProfileMobileDevice.java* gera o perfil do dispositivo conforme as informações enviadas sobre o mesmo, a classe *AnalysisPerfilDevice.java* faz analise das informações recebidas para identificação dos comportamentos irregulares;

- *br.idsproxy.storage:* Pacote para armazenamento dos arquivos contendo as informações da captura do tráfego e as informações enviadas pelos dispositivo móveis.

A seguir, as principais telas do protótipo implementado serão descritas para um melhor entendimento do processo desenvolvimento deste trabalho. As capturas de tela de implementação do protótipo serão descritas a seguir.

A Figura 4.6 representa a tela inicial do *IDS_Proxy* onde os menus, divididos em módulos representam as funcionalidades do sistema. O menu Arquivo contém o submenu Sair que permitide ao administrador sair do programa a qualquer momento. No menu Captura existem os submenus Iniciar, Parar, Consultar que inicializa, pára e faz consulta de tráfego na rede *wireless*, respectivamente. Observando-se que no submenu Consultar o usuário pode informar a data de uma determinada captura e o sistema disponibilizará os dados solicitados. No menu Notificação permite ao administrador fazer o envio de notificações em situações de detecção de ameaças contra os dispositivos móveis monitorados. O menu Dispositivo Móvel contém as funcionalidades administrativas para dispositivos móveis. É composto pelos seguintes submenus:

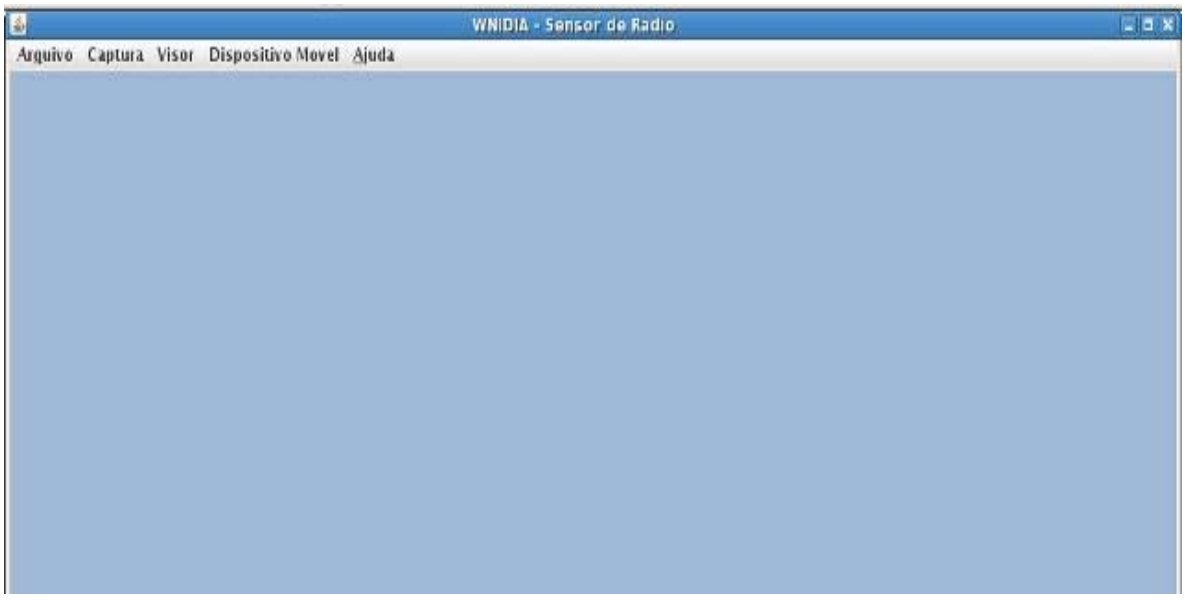


Figura 4.6 - Tela Principal do IDS_Proxy

- Cadastrar: faz o cadastramento das informações sobre o dispositivo móvel;
- Consultar: permite a consulta das informações sobre os dispositivos cadastrados;
- Editar: permite a alteração dos dados cadastrados;

- Excluir: permite a exclusão de um determinado registro;
- Consultar: perfil permite a consulta das informações enviadas pelos dispositivos.

A Figura 4.7 representa a tela de cadastro de dispositivos móveis onde o administrador informa o nome do dispositivo móvel e do seu endereço MAC e salva os dados informados que serão utilizados no processo de captura de pacotes no tráfego da rede *wireless*. A escolha do endereço MAC como chave de identificação por ser um identificador único que será associado ao endereço IP do dispositivo, este por sua vez pode variar a cada nova associação na rede *wireless*.

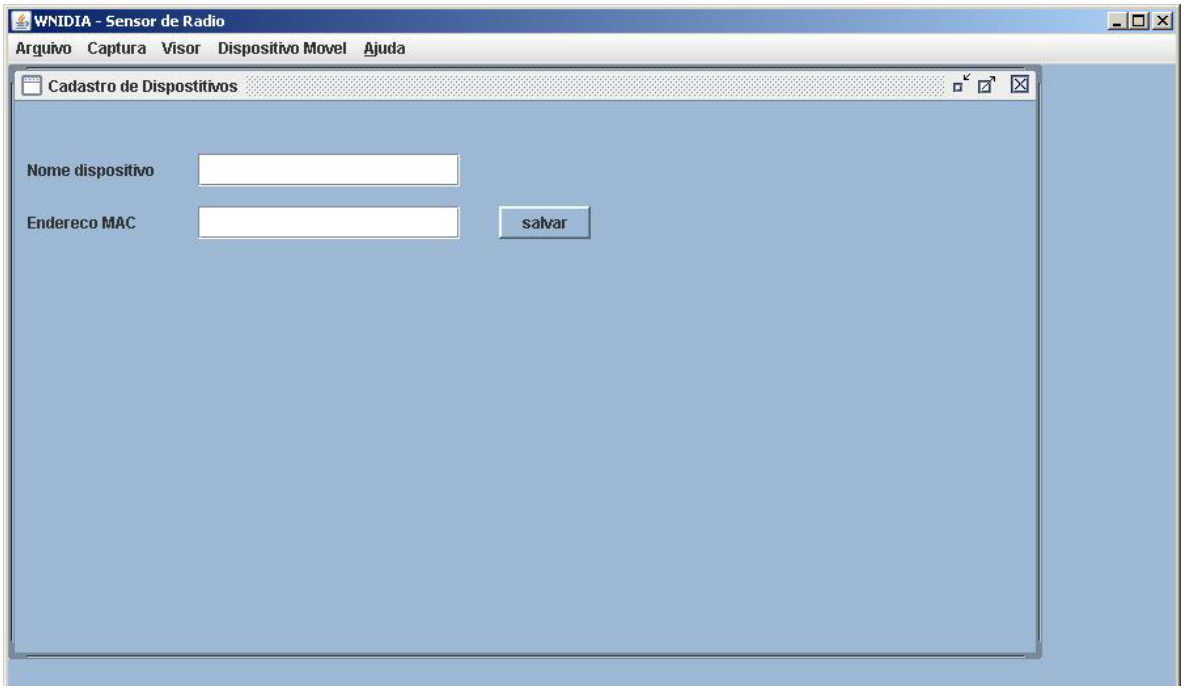
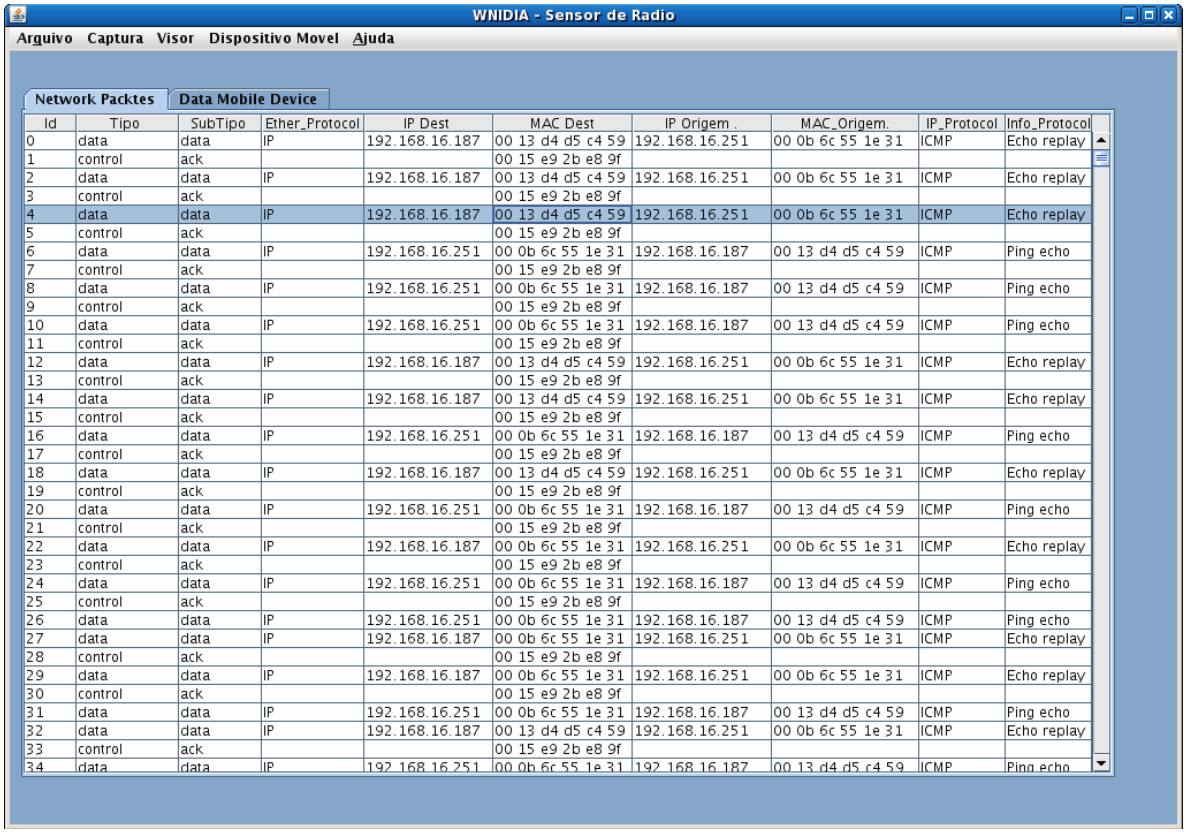


Figura 4.7 - Tela de Cadastro de dispositivos móveis

A Figura 4.8 apresenta tela de captura das informações de tráfego da rede *wireless*, durante o momento em que ocorria *ping flood* ataque.



Id	Tipo	SubTipo	Ether_Protocol	IP Dest	MAC Dest	IP Origem .	MAC_Origem.	IP_Protocol	Info_Protocol
0	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
1	control	ack			00 15 e9 2b e8 9f				
2	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
3	control	ack			00 15 e9 2b e8 9f				
4	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
5	control	ack			00 15 e9 2b e8 9f				
6	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo
7	control	ack			00 15 e9 2b e8 9f				
8	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo
9	control	ack			00 15 e9 2b e8 9f				
10	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo
11	control	ack			00 15 e9 2b e8 9f				
12	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
13	control	ack			00 15 e9 2b e8 9f				
14	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
15	control	ack			00 15 e9 2b e8 9f				
16	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo
17	control	ack			00 15 e9 2b e8 9f				
18	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
19	control	ack			00 15 e9 2b e8 9f				
20	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo
21	control	ack			00 15 e9 2b e8 9f				
22	data	data	IP	192.168.16.187	00 0b 6c 55 1e 31	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
23	control	ack			00 15 e9 2b e8 9f				
24	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo
25	control	ack			00 15 e9 2b e8 9f				
26	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo
27	data	data	IP	192.168.16.187	00 0b 6c 55 1e 31	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
28	control	ack			00 15 e9 2b e8 9f				
29	data	data	IP	192.168.16.187	00 0b 6c 55 1e 31	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
30	control	ack			00 15 e9 2b e8 9f				
31	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo
32	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
33	control	ack			00 15 e9 2b e8 9f				
34	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo

Figura 4.8 - Tela da captura de tráfego na rede wireless

Conforme apresentado na Figura 4.8, os seguintes campos depois de decodificados são mostrados ao administrador:

- Id: identificador na ordem em que o mesmo é capturado;
- Tipo: tipo de frame do capturado (*data*, *control*, *management*) , informação obtida a partir da camada MAC do frame;
- Subtipo: definido de acordo com tipo do frame;
- *EtherProtocol:* protocolo especificado pela subcamada SNAP (*Subnetwork Acess Protocol*) (SNAP, 2001);
- IP Dest: endereço IP de destino do frame;
- MAC Dest: endereço *MAC* de destino do frame, representados no formato hexadecimal;
- IP Origem: endereço IP origem do frame;
- MAC Origem: endereço MAC de origem, representados no formato hexadecimal;
- *IP Protocol*: cabeçalho encapsulado dentre do pacote IP;

- *Info Protocol:* identifica a funcionalidade do protocolo encapsulado dentro do pacote IP.

Conforme ilustrado na Figura 4.8, as trocas de mensagens durante o *ping flood* ataque, através dos excessivos *pings* destinados ao dispositivo móvel, identificado com o IP *address* 192.168.16.251. As mensagens foram identificadas através do protocolo ICMP (*Internet Control Message Protocol*) que é anexado ao datagrama IP. O *Ping echo* identifica a requisição feita pela máquina com IP *adreess* de origem 192.168.16.187 ao dispositivo móvel. O dispositivo responde com *Echo replay* informando o seu *status* na rede

A Figura 4.9 apresenta a tela de recepção das informações enviadas pelo dispositivo móvel durante o monitoramento do mesmo. As seguintes informações são apresentadas ao administrador:

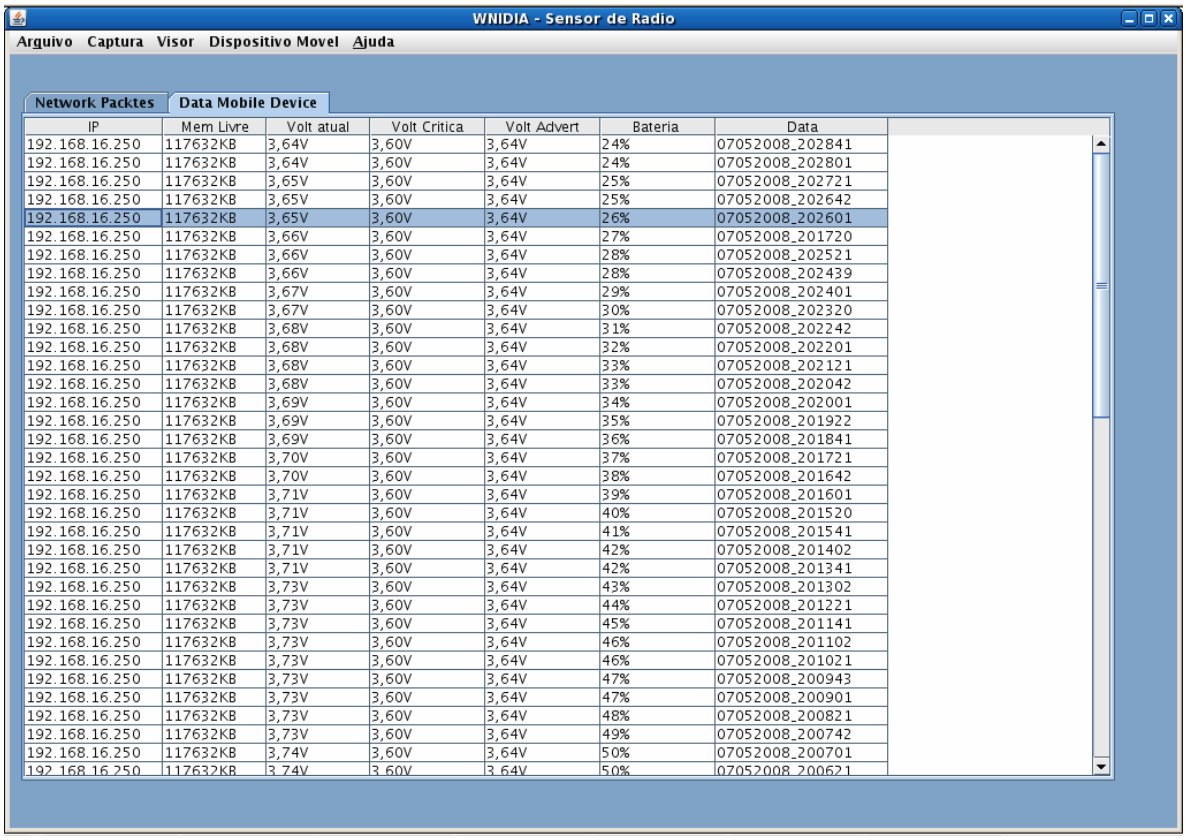


Figura 4.9 - Recebimento dos dados de dispositivos móveis

- IP: endereço IP do dispositivo que está enviando os dados;
- Mem Livre: indica a quantidade de memória disponível no dispositivo;
- Vol Atual: voltagem atual da bateria do dispositivo;

- Vol Critica: índice da voltagem considerada crítica para o dispositivo, valor indicado pelo dispositivo em que alguns serviços podem não estar disponíveis ao usuário;
- Vol Advert: índice voltagem de advertência para o dispositivo, com este índice o usuário será informado que sua bateria precisa ser carregada novamente;
- Bateria: a porcentagem de bateria disponível no dispositivo no momento do envio da informação;
- Data: indica a hora do envio da informação enviada pelo dispositivo, um intervalo de 40s (quarenta segundos) foi utilizado para envio dos dados do dispositivo. Os dados são mostrados no formato ddmmaaaa_hhmmss, onde dd é dia; mm o mês; aaaa o ano; hh a hora; mm os minutos e ss os segundos.

Na Figura 4.9, é constatado que a um decréscimo nos valores que indicam a porcentagem de bateria restante no dispositivo, representado pelo IP *address* 192.168.16.250, e para o valor que indica a voltagem atual da bateria, que com decorrer do tempo aproxima-se do valor de voltagem advertência do dispositivo.

O Código 4.1 apresenta o trecho do arquivo de captura da rede *wireless.* As informações foram armazenadas em arquivos com formato XML. A estrutura do documento é apresentada a seguir.

Os elementos que compõe a estrutura do arquivo são os seguintes:

- id_frame: identifica a ordem de captura na rede wireless;
- data_packet: fluxo capturado em formato hexadecimal na rede wireless;
- control_frame: campo de controle do frame, informação utilizada para identificar a função do frame na camada MAC;
- type_frame; tipo de frame (dados, controle ou gerenciamento);
- subtype_frame: subtipo do frame de acordo com seu tipo;
- to_Ds: indica se frame esta indo para o sistema de distribuição;
- from_Ds: indica se frame esta saindo do sistema de distribuição;
- more_fragment: indica se existem mais fragmentos do frame a serem recebidos para estação;
- frame_potency: indica se estação entrará em modo power save;
- more_repeat: se o frame esta sendo retransmitido;
- more_data: indica para a estação que mais frames a serem enviados;
- frame_ordem: indica se frames recebidos devem ser estritamente ordenados;

- duration: indica o tempo necessário para receber a próxima transmissão;

```
- <id_frame id_frame="2000">
<time_frame>1211474997.113358s</time_frame>
  +<data_packet></data_packet>
  <control_frame>0000100000001010</control_frame>
  <tipo_frame>data</tipo_frame>
  <subtype_frame>data</subtype_frame>
  <to_Ds>0</to_Ds>
  <from_Ds>1</from_Ds>
  <more_fragment>0</more_fragment>
  <frame_repeat>1</frame_repeat>
  <frame_potency>0</frame_potency>
  <more_data>0</more_data>
  <frame_ordem>0</frame_ordem>
  <duration>117</duration>
  <address_1>00 0b 6c 55 1e 31</address_1>
  <address_2>00 15 e9 2b e8 9f</address_2>
  <address_3>00 13 d4 d5 c4 59</address_3>
  <address_4 />
  <ethertype_protocol>08 00</ethertype_protocol>
  <source_address>192.168.16.187</source_address>
  <destination_address>192.168.16.250</destination_address>
  <ip_protocol>06</ip_protocol>
  <version_packet>4</version_packet>
  <IHL>20</IHL>
  <lenght>40</lenght>
  <identification>41048</identification>
  <offset>0</offset>
  <time_to_live>64</time_to_live>
  <header_checksum>37 72</header_checksum>
  <desc_ip_protocol>TCP</desc_ip_protocol>
  <info_ip_protocol>2</info_ip_protocol>
</id_frame>
```

Código 4.1 - Trecho do arquivo de captura do tráfego wireless

- address_1: MAC address da estação de origem;
- address_2: MAC address da estação destino;
- address_3: MAC addresd da BSSID;
- address_4: MAC address que indica o endereço MAC da estação que originou o frame nas situações onde para o frames tenham como origem e destino estações no sistema de distribuição;
- source_address: endereço IP da estação de origem;
- destination_address: endereço IP da estação de destino;

- ethertype_protocol: protocolo encapsulado dentro pacote especificado pela subcamada SNAP;
- version: versão do pacote datagrama;
- IHL: tamanho do cabeçalho na *Internet*;
- lenght: tamanho completo do datagrama incluindo cabeçalho e dados;
- identification: identificador de fragmentos do datagrama original;
- offset: indica a que datagrama o fragmento pertence;
- time_to_live: tempo de vida do pacote na rede;
- header_checksum: indica o valor de verificação;
- ip_protocol: define o protocolo encapsulado do datagrama IP;
- info_protocol: informação sobre a funcionalidade do protocolo encapsulado no datagrama.

O Código 4.1 apresenta o trecho do arquivo que representa captura de tráfego para o dispositivo móvel, identifica-se um datagrama IP, através de um pedido de requisição pela origem , representada pelo IP *address* 192.168.16.187, a um destino, representado pelo IP *address* 192.168.16.250.

```
- <id_device id_perfil="78">
    <ipdevice>192.168.16.250</ipdevice>
    +<sofwares></sofwares>
    <mem_free >95880KB</mem_free>
    <mem_total >117632KB</mem_total>
    <voltage_atual>3,64V</voltage_atual>
    <voltage_critica>3,60V</voltage_critica>
    <voltage_advertencia>3,64V</voltage_advertencia>
    <quantidade_bateria>24%</quantidade_bateria>
    <data_operacao>07052008_202841</data_operacao>
  </id_device>
```

Código 4.2 - Trecho do arquivo de armazenamento dos dados enviados pelo dispositivo

Os elementos que compõem a estrutura do documento de armazenamento das informações sobre o dispositivo são descritos a seguir:

- id_perfil: contador para quantidade de informações enviadas pelo dispositivo;
- softwares: lista de softwares instalados no dispositivo, no trecho, a lista ficou oculta devido(indicado pelo sinal “+” a lado do elemento) a extensão da lista;
- mem_total: quantidade de memória total no dispositivo;

- mem_free: quantidade de memória livre no dispositivo;
- voltage_atual: voltagem atual da bateria:
- voltage_critica: valor da voltagem considerada critica para o dispositivo;
- voltage_advertencia: voltagem de advertência para o dispositivo;
- quantidade_bateria: quantidade de bateria no dispositivo, apresentada em porcentagem;
- data_operacao: data em que foi registrado a recepção dos dados.

4.4 TESTES

Para validação da arquitetura proposta foram simulados dois ataques como forma de demonstrar a eficácia do modelo proposto. A simulação dos ataques e os impactos que os mesmos causaram serão descritos nas seguintes subseções.

4.4.1 Ataques

Em redes *wireless* a carga da bateria dos dispositivos móveis é um fator de extrema importância, principalmente quando se trata de pequenos dispositivos como *Smarthphones* e PDA's. Esses dispositivos possuem limitações físicas quanto a capacidade de processamento e armazenamento e baseado nessas limitações são gerados os ataques que demandam maior impacto sobre estes dispositivos. Ambos os ataques foram gerados através de requisições via rede com o objetivo de causar uma maior demanda de processamento ao dispositivo e conseqüente a exaustão da carga de bateria do mesmo. Os dois ataques simulados foram *Sleep Deprivation Attack* e *Syn Flood Attack* que terão seus cenários descritos nas seções seguintes.

4.4.1.1 Sleep Deprivation Attack

O *Sleep Deprivation Attack* é um tipo de ataque DoS que leva a exaustão da bateria do dispositivo impedindo que o mesmo entre em modo *power sleep, status* em que o dispositivo diminui a demanda por atividades com a finalidade de preservar recursos, principalmente a carga de bateria. O objetivo deste tipo de ataque é demandar o máximo de

atividades para dispositivo, conseqüentemente descarregá-lo e impedir que usuário tenha acesso às informações armazenadas no mesmo.

A simulação proposta para idealização deste tipo de ataque consistiu em demandar uma excessiva quantidade de requisições através da interface de rede do dispositivo com o objetivo de mantê-lo sempre com uma demanda operacional através de tráfego de rede.

Para impedir que o dispositivo entre em modo *power sleep* foram gerados ataques baseado no *ping flood attack*, onde centenas de requisições foram geradas para dispositivo com a finalidade manter certo nível de operação na interface de rede do dispositivo impedindo que mesmo chegue a uma situação de inatividade e entre em estado *power sleep.*

Para simulação deste ataque foram simulados dois tipos de ataque baseado nos seguintes comandos: o primeiro utilizado o comando *ping –i 0.000001 –s 136 192.168.16.250* e o segundo *ping –i 0.000001 –s 512 192.168.16.250.* Onde:

- -i: representa o intervalo para repetição do comando;
- -s: tamanho do pacote enviado;
- *"129.168.16.250"*: destino contra o qual será executado o comando.

Para simulação do primeiro cenário foram abertos em vinte terminais a partir de uma máquina com sistema operacional linux *Fedora Core* que executaram o comando *ping –i 0.000001 –s 136 192.168.16.250* durante um intervalo de tempo de 40min (quarenta minutos). O comando faz execução de pacotes com tamanho de 136 *bytes* com intervalos de 0.000001 segundos contra o dispositivo com o IP especificado. O tempo em que este comando ficou em execução foi suficiente para representar os efeitos que tomaram proporções consideráveis no nível de carga e voltagem de bateria do dispositivo. Esses dados ficam evidenciados nos gráficos abaixo. O gráfico da Figura 4.10 apresenta uma comparação entre a carga de bateria do dispositivo *versus* o tempo durante o qual o ataque foi executado.

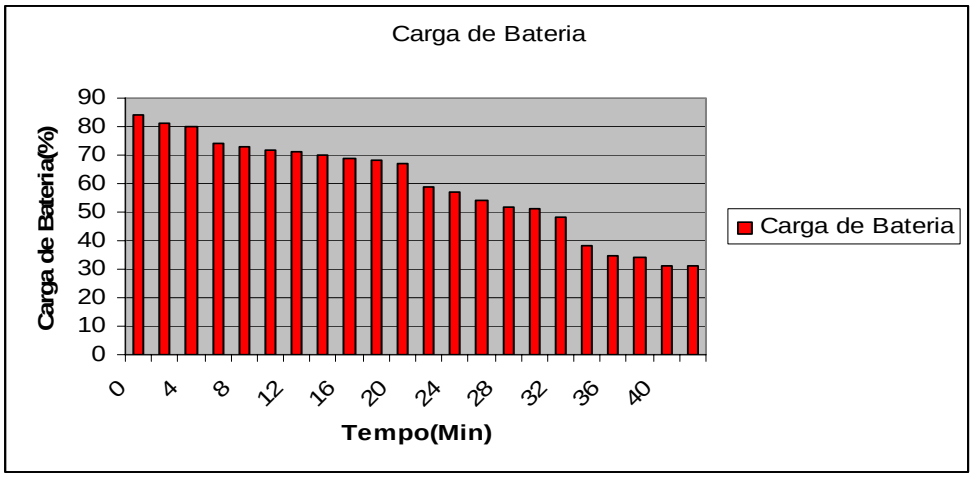


Figura 4.10 - Ping flood attack (bateria *versus* tempo)

O gráfico da Figura 4.11 traz um comparativo entre a voltagem da bateria versus o tempo durante o mesmo período de execução do ataque.

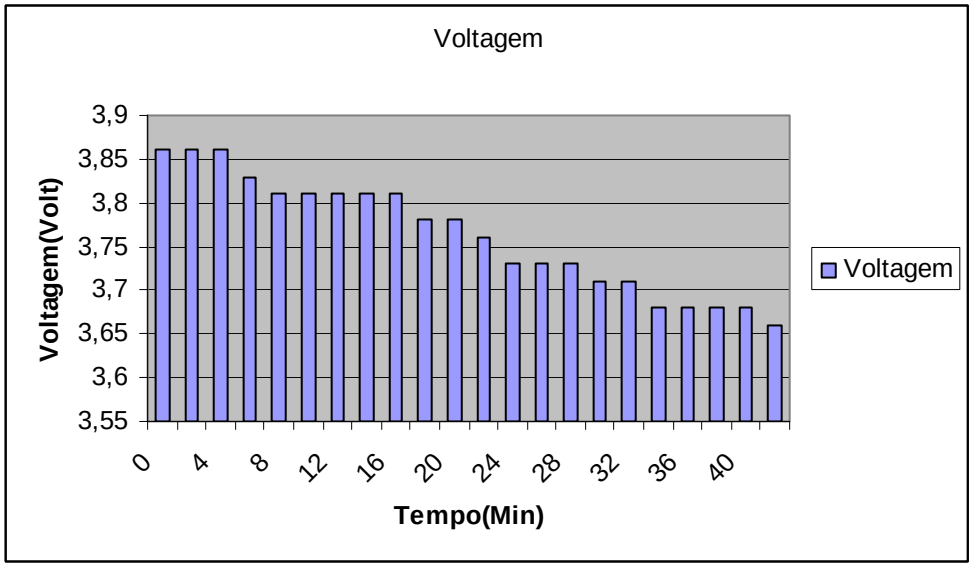


Figura 4.11 - Ping flood attack (voltagem *versus* tempo)

Para simulação do segundo cenário, as mesmas variáveis do primeiro teste foram mantidas com exceção do comando de execução que foi *ping –i 0.000001 –s 512 192.168.16.250*. O comando fez execução de pacotes com tamanho de 512 *bytes* com intervalos de 0.000001 segundos contra o dispositivo. Os resultados são mostrados nos gráficos da Figura 4.12 e da Figura 4.13.

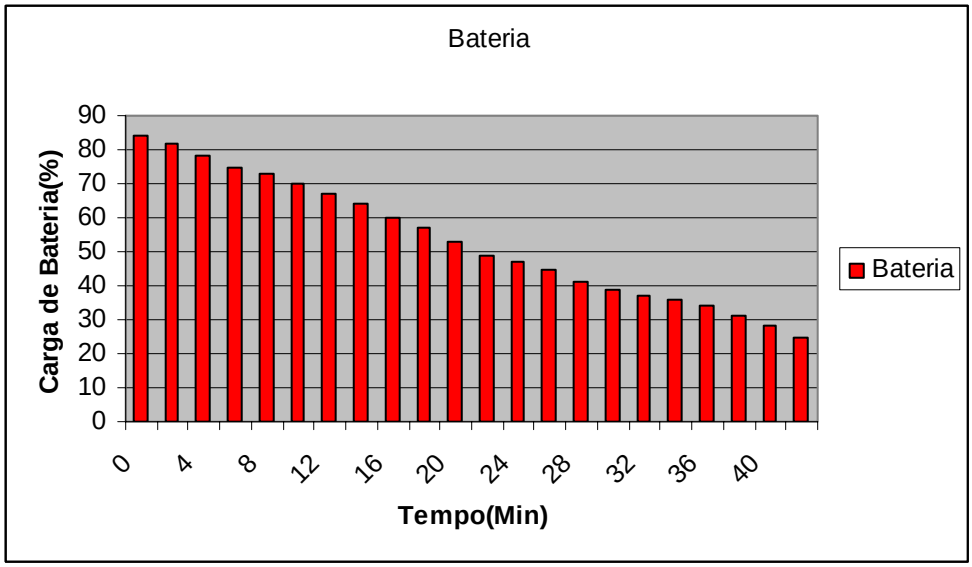


Figura 4.12 - Cenário 2: Simulação ping flood attack (bateria *versus* tempo)

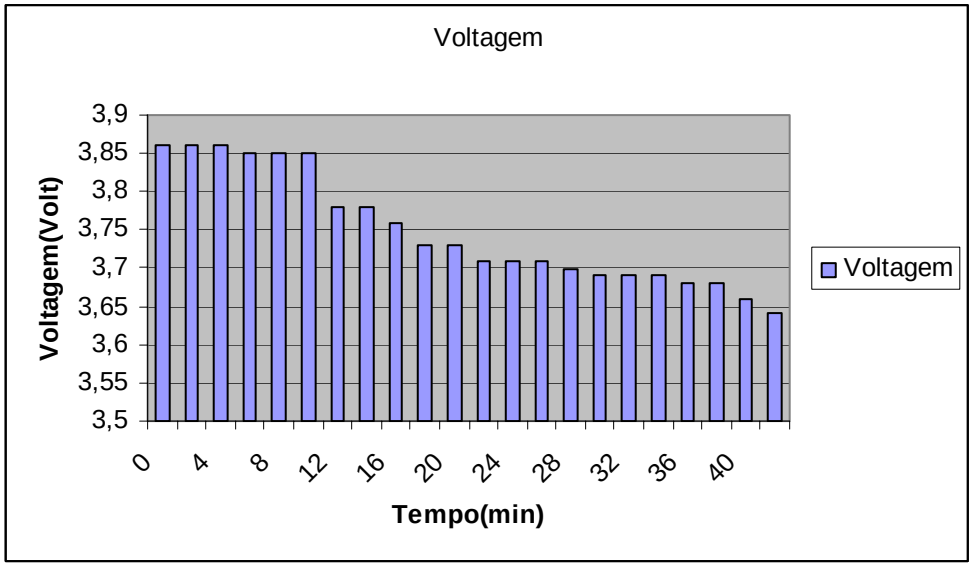


Figura 4.13 - Cenário 2: Simulação ping flood attack (voltagem *versus* tempo)

Analisando as informações obtidas através dos gráficos da Figura 4.10 e da Figura 4.12, conclui-se que *ping flood attack* durante um intervalo de tempo pode afetar consideravelmente a carga de bateria do dispositivo devido ao número de requisições excessivas feitas ao dispositivo. Com medidas em intervalos regulares de 40s (quarenta segundos), observou-se que em um intervalo de tempo de aproximadamente de 40min (quarenta minutos), a carga remanescente de bateria reduziu em menos de 50% (cinqüenta por cento) do valor obtido no início da realização do ataque para a primeira simulação. Situação semelhante pode ser observada nas medições da voltagem da bateria que também teve um decréscimo nos valores lidos. Em relação ao segundo cenário observou-se que aumentando a

carga do útil do pacote enviado ao dispositivo houve uma aceleração na descarga da bateria do dispositivo que atingiu 45% (quarenta e cinco por cento) de carga remanescente com aproximadamente 25min (vinte e cinco minutos) de simulação do ataque. A velocidade da descarga deve-se ao processamento exigido para processar pacotes com maior carga útil. Isso se reflete no consumo de energia que tem uma demanda maior, exigindo mais consumo de bateria do dispositivo.

4.4.1.2 SYN (Sincronize) flood attack

SYN flood é uma forma de ataque DoS no qual o atacante envia uma sucessão de requisições *SYN* para um determinado alvo, ou seja, é feita uma solicitação de abertura de conexão a um determinado *host*. Este tipo de ataque ocorre sobre o protocolo TCP (*Transmission Control Protocol*), onde o atacante envia uma *stream* de *SYN request* para a vitima, mas não responde com o segmento ACK que é enviado de volta contendo o endereço IP de origem com pacote contendo o segmento TCP SYN de solicitação de abertura de conexão.

O processo de comunicação para este ataque é baseado no *handshake* de três vias. Inicialmente, o cliente faz uma solicitação de requisição de conexão enviando uma mensagem SYN para um determinado alvo. O alvo reconhecendo o tipo de solicitação faz o envio de SYN-ACK como retorno para o cliente que fez a solicitação e este responde novamente com o ACK para que a conexão possa ser estabelecida. A Figura 4.14 resume a forma de comunicação baseado no *handshake* de três vias.

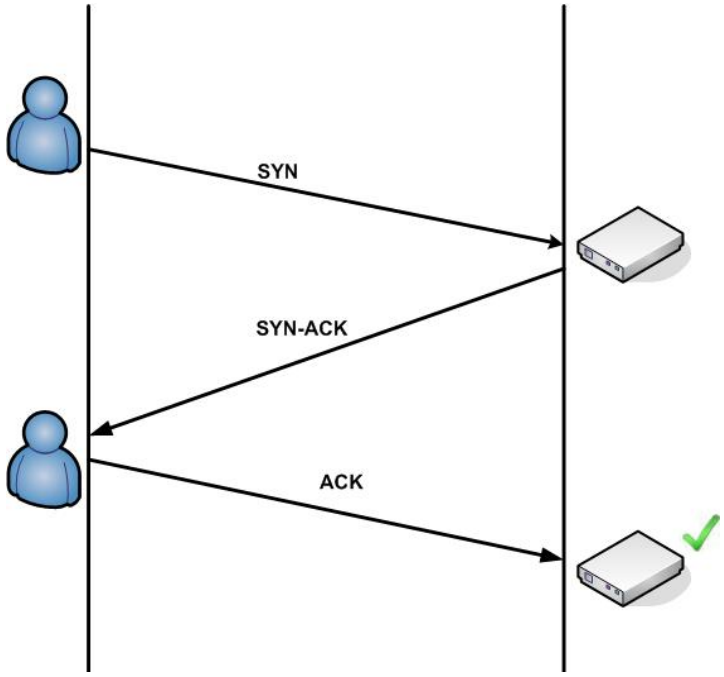


Figura 4.14 - Handshake de três vias

O objetivo deste ataque sobre o dispositivo é manter certo nível de atividade na sua interface de rede através da resposta que é retornada pelo dispositivo mesmo nas situações onde não for possível estabelecer conexão. A intenção é realmente fazer o dispositivo gastar energia e processamento no processo de resposta para essas requisições.

Para realização deste tipo de ataque, a utilização de uma ferramenta que gerasse pacotes com as características do ataque proposto foi necessária, ou seja, geração de pacotes com *flag* de controle SYN ativado. A ferramenta *hping2* (HPING, 1999) foi utilizada para este propósito. Esta é uma ferramenta de linha de comando para criar pacotes contendo *payloads* TCP, UDP ou ICMP que podem ser modificados e controlados usando sintaxes de linha de comando.

O ataque gerado para o protótipo foi baseado no comando *hping –I eth0 -S 192.168.1.250 –p ++80 –i u1000*. Este comando gera pacotes através da rede onde *–I* indica a interface de origem para geração do tráfego (*eth0*, utilizado no exemplo), *-S* indica que *flag* de controle SYN habilitado para um destino, indicado por *192.168.16.250* e o *–p* indica a porta para qual a requisição deve ser feita. O complemento para "++" antes do número da porta informa que o comando incrementará o número da porta em uma unidade a tentativa a cada solicitação de requisição e *–i* indica em que intervalo serão feitas as requisições na simulação foi utilizada a taxa para a cada mil microsegundos.

Este comando foi executado de duas formas diferentes onde a única alteração foi inicio da contagem para a porta que na primeira situação foi iniciada na porta de número 10

(dez) e em outro momento começando da porta 80 (oitenta). Os ataques tiveram com inicio solicitações nas portas 10 e 80, devido a conhecidos serviços ficarem executando neste intervalo ou acima dele. Estes dois comandos foram executados concorrentemente contra o dispositivo durante um intervalo de tempo de aproximadamente 40 min (quarenta minutos). As medidas de análise tiveram como base o nível de carga e voltagem da bateria. O impacto sobre estas primitivas fica demonstrado nos gráficos da Figura 4.15 e da Figura 4.16. Observando os gráficos, conclui-se que *SYN flood* ataque pode levar a exaustão de bateria do dispositivo, em um intervalo de tempo maior se comparado ao *ping flood* ataque. Os valores de descarga da bateria sofreram alterações mais discretas, mas de qualquer forma tem seu impacto estimável sobre a carga da bateria do dispositivo.

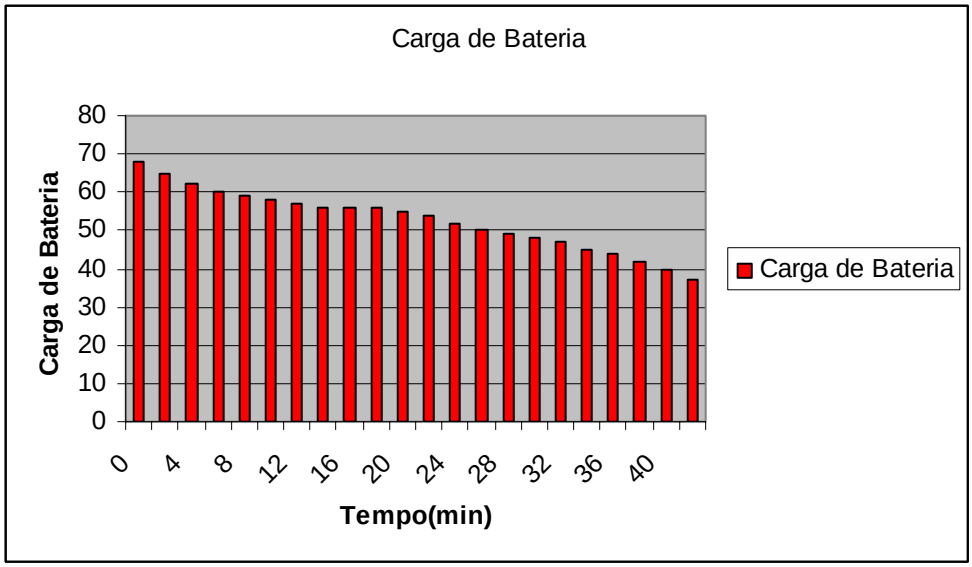


Figura 4.15 - SYN flood attack (bateria *versus* tempo)

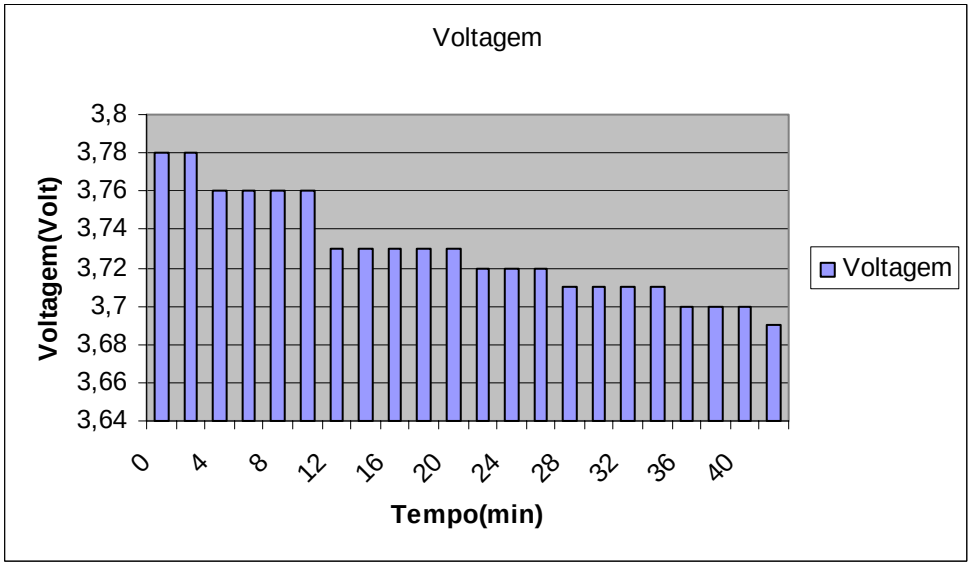


Figura 4.16 - SYN flood attack (voltagem *versus* tempo)

4.5 DETECÇÃO DOS ATAQUES

O processo de identificação consiste na análise de dois processos: o primeiro faz identificação do tráfego de rede para os dispositivos monitorados; e segundo monitora as informações enviadas pelos dispositivos. A junção destas informações foi utilizada no processo de detecção de ataques contra os dispositivos monitorados. A análise dessas informações será descrita nas seções seguintes.

4.5.1 Análise do tráfego wireless

O processo de análise do tráfego *wireless* consiste em capturar e identificar as informações enviadas e recebidas para os dispositivos cadastrados. O processo tem início a cada novo pedido de associação na rede onde é verificado através do endereço MAC *host* que solicita a associação que uma vez confirmada pelo *access point* é verificada se o dispositivo está presente no arquivo de cadastro dos dispositivos.

As informações capturadas são analisadas para extração dos dados das camadas MAC, rede e transporte do modelo OSI. A necessidade destas informações baseou-se que tipos de ataques ocorrem através da rede explorando as vulnerabilidades de determinados tipos de protocolos.

O formato dos quadros para o padrão IEEE 802.11b pode ser visualizado na Figura 4.17.

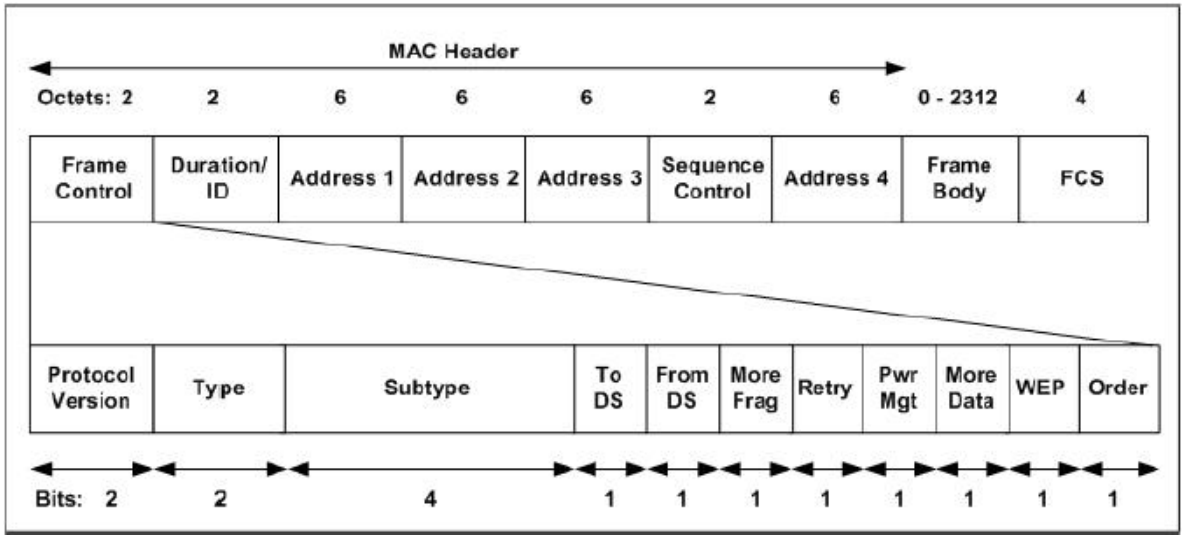


Figura 4.17 - Formato dos quadros IEEE 802.11b

De acordo com o padrão IEEE 802.11b, cada quadro possui um campo *Frame Control*, com 11 subcampos, um campo *Duration/ID*, informa por quanto tempo o quadro e sua confirmação ocuparão o canal, quatro campos *Address*, estes campos informam a origem e o destino de cada quadro, um campo *Sequence Control*, permite que os fragmentos sejam numerados, um campo *Frame Body*, que contém a carga útil do pacote, e um campo *FCS*, que possui um total de verificação. O campo *Frame Control* possui um subcampo *Protocol Version*, que permite a operação de duas versões do protocolo ao mesmo tempo na mesma célula, um subcampo *Type* (dados, controle e gerenciamento), um subcampo Subtype (porexemplo RTS ou CTS), os bits *To DS* e *From DS*, que indicam se o quadro está indo ou vindo do sistema de distribuição entre células, um bit *More Frag*, que indica a existência de mais fragmentos, um bit *Retry*, que indica a retransmissão de um quadro enviado anteriormente, um bit *Pwr Mgt*, usado pelo AP para deixar o receptor em estado de espera ou retirá-lo do estado de espera, um bit *More Data*, indica que o transmissor tem quadros adicionais para o receptor, um bit *WEP*, indica que o corpo do quadro foi criptografado com o WEP, e um bit *Order*, informa ao receptor que uma seqüência de quadros com esse bit tem que ser processada estritamente em ordem.

Uma análise minuciosa é realizada sobre estes quadros para que seja possível identificar certos tipos de quadros. Para os tipos de ataques gerados para o protótipo foi necessária a identificação dos protocolos TCP e ICMP.

O ICMP é um protocolo de controle de gerenciamento das *hosts* que fazem parte de rede. A análise consistiu em identificar quadros com formato correspondente às mensagens enviadas por este protocolo. Para esta finalidade foi necessária a identificação do cabeçalho deste protocolo. A Tabela 4.1 apresenta o formato cabeçalho das mensagens ICMP.

Tabela 4.1 - ICMP header

8	16	32 bits
Type	Code	Checksum
Identifier		Sequence number
Data		

De acordo como especificado na RFC 792 (RFC792, 2007) e mostrado na Tabela 4.1, o ICMP *header* é composto de seis campos. O campo *Type* é um campo de tamanho de oito bits que contém o tipo do pacote ICMP. O campo *Code* também um campo de oito bits que contém o código identificador da mensagem. O campo *Checksum* é um campo de

dezesseis bits que faz a validação da mensagem. O campo *Identifier* contém o identificador das mensagens que são enviadas no caso de mensagens de *ECHO_REPLAY*. O campo *Sequence* que possui o valor da seqüência que deve ser retornado em mensagens *ECHO_REPLAY*.

O *Ping flood attack* é um tipo ataque que ocorre sobre o protocolo ICMP. A análise consistiu em identificar mensagens com campo *Type* com o valor igual a oito (8) e *Code(0)* que representam mensagens de *ECHO_REQUEST* que identifica que *pings* estão lançados através da rede contra um determinado destino e mensagens com valor para *Type* e *Code* iguais a zero(0). Estes são os valores que identificam mensagens *ECHO_REPLAY* em resposta a solicitações de *ECHO_REQUEST*.

O TCP é um protocolo da camada de transporte responsável pelo estabelecimento de conexões. Como citado anteriormente, o processo de comunicação é baseado no *handshake* de três vias. Conforme definido na RFC 793 (RFC793, 2007) o formato do cabeçalho deste protocolo é especificado conforme a Tabela 4.2.

A Tabela 4.2 apresenta os campos s*ource port e destination port* de 16 *bits* que indicam respectivamente a porta de origem e destino para estabelecimento de comunicação. O campo *sequence number* tem o tamanho de 32 bits e indica o número de seqüência do segmento SYN em quadros onde flag de controle SYN é setado. O *acknowledgment number* tem o tamanho de 32 bits que indica o número de seqüência para o próximo ACK em quadros onde *flag* de controle do ACK é setado. O campo de *control bits* representado na figura pelas siglas URG (*Urgent Pointer*), ACK (*Acknowledgement*), PSH (*Push Function)*, RST (*Reset the Connection*), SYN (*Synchronize*), FIN (*No More Data from Sender*) é composto por seis *bits* e é o responsável pelo gerenciamento das atividades desempenhadas pelo protocolo através do acionamento dos *bits* para cada atividade. Uma descrição detalhada dos protocolos TCP e ICMP é descrito no Apêndice A desta dissertação.

Tabela 4.2 - TCP header

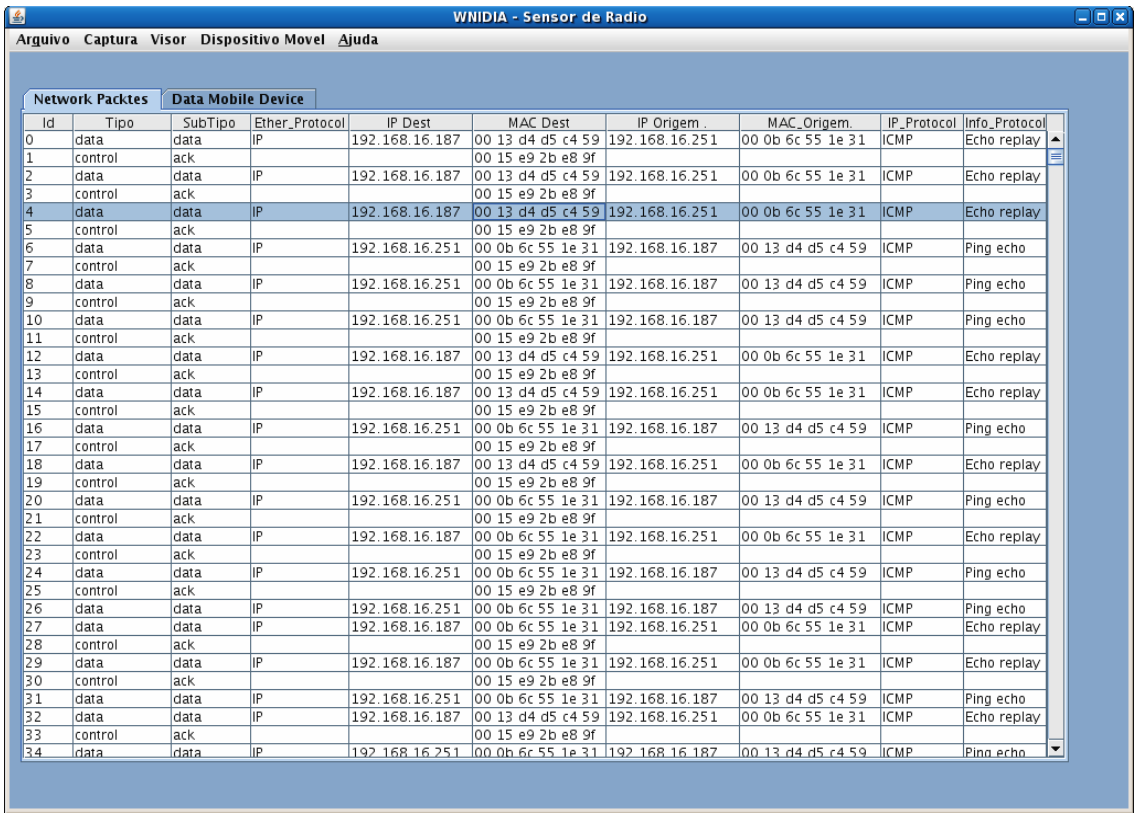
0					16			31bits
Source port								Destination port
Sequence number								
Acknowledgement Number								
HELEN	Reserved	URG	ACK	PSN	RST	SYN	FIN	Window
Checksum								Urgent pointer
Options								Padding
Data								

O SYN *flood attack* é um tipo de ataque que ocorre sobre o protocolo TCP. A análise consistiu na identificação de *flags* de controle existentes em pacotes responsáveis pelo estabelecimento de comunicação.

O momento da detecção para os dois ataques feito *IDS_Proxy* pode ser observado na Figura 4.18 para o *ping flood attack* e na Figura 4.19 para o *SYN flood attack*.

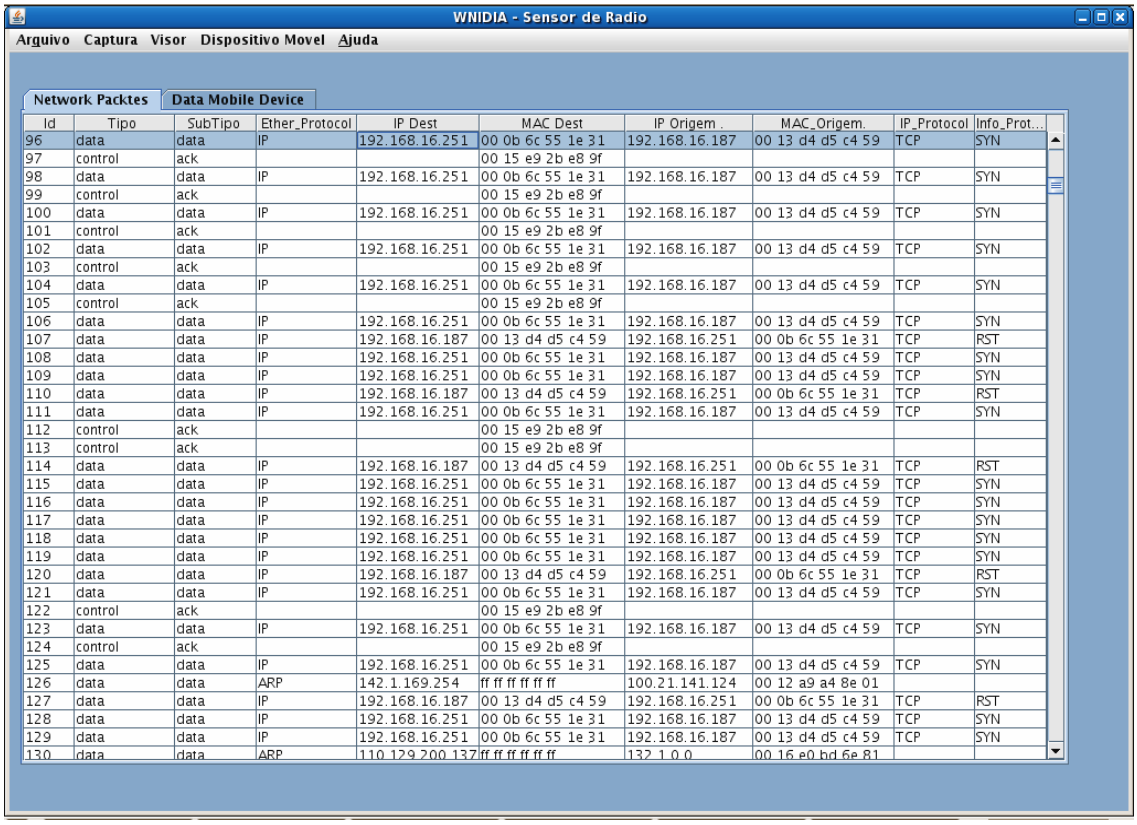
Observando-se a Figura 4.18, identifica-se a excessiva troca de mensagens entre o dispositivo móvel, identificado na rede com IP *address* 192.168.16.251, onde outra máquina na rede, identificada com IP *address* 192.168.16.187, lança vários *pings echo* contra o dispositivo e o mesmo responde com *echos replay*.

A Figura 4.19 apresenta os vários pedidos de requisições, feitas ao dispositivo móvel, identificado na rede com o IP *address* 192.168.16.251, este responde com pacotes *RST*, informando à origem que faça novamente o pedido, pois não foi possível atender ao pedido de conexão.



Id	Tipo	SubTipo	Ether_Protocol	IP Dest	MAC Dest	IP Origem .	MAC_Origem.	IP_Protocol	Info_Protocol
0	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
1	control	ack			00 15 e9 2b e8 9f				
2	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
3	control	ack			00 15 e9 2b e8 9f				
4	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
5	control	ack			00 15 e9 2b e8 9f				
6	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo
7	control	ack			00 15 e9 2b e8 9f				
8	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo
9	control	ack			00 15 e9 2b e8 9f				
10	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo
11	control	ack			00 15 e9 2b e8 9f				
12	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
13	control	ack			00 15 e9 2b e8 9f				
14	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
15	control	ack			00 15 e9 2b e8 9f				
16	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo
17	control	ack			00 15 e9 2b e8 9f				
18	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
19	control	ack			00 15 e9 2b e8 9f				
20	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo
21	control	ack			00 15 e9 2b e8 9f				
22	data	data	IP	192.168.16.187	00 0b 6c 55 1e 31	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
23	control	ack			00 15 e9 2b e8 9f				
24	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo
25	control	ack			00 15 e9 2b e8 9f				
26	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo
27	data	data	IP	192.168.16.187	00 0b 6c 55 1e 31	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
28	control	ack			00 15 e9 2b e8 9f				
29	data	data	IP	192.168.16.187	00 0b 6c 55 1e 31	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
30	control	ack			00 15 e9 2b e8 9f				
31	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo
32	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	ICMP	Echo replay
33	control	ack			00 15 e9 2b e8 9f				
34	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo

Figura 4.18 - Captura do tráfego wireless para o ping flood attack



Id	Tipo	SubTipo	Ether_Protocol	IP Dest	MAC Dest	IP Origem .	MAC_Origem.	IP_Protocol	Info_Prot...
96	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
97	control	ack			00 15 e9 2b e8 9f				
98	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
99	control	ack			00 15 e9 2b e8 9f				
100	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
101	control	ack			00 15 e9 2b e8 9f				
102	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
103	control	ack			00 15 e9 2b e8 9f				
104	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
105	control	ack			00 15 e9 2b e8 9f				
106	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
107	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	TCP	RST
108	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
109	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
110	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	TCP	RST
111	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
112	control	ack			00 15 e9 2b e8 9f				
113	control	ack			00 15 e9 2b e8 9f				
114	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	TCP	RST
115	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
116	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
117	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
118	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
119	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
120	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	TCP	RST
121	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
122	control	ack			00 15 e9 2b e8 9f				
123	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
124	control	ack			00 15 e9 2b e8 9f				
125	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
126	data	data	ARP	142.1.169.254	ff ff ff ff ff ff	100.21.141.124	00 12 a9 a4 8e 01		
127	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.251	00 0b 6c 55 1e 31	TCP	RST
128	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
129	data	data	IP	192.168.16.251	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	TCP	SYN
130	data	data	ARP	110.129.200.137	ff ff ff ff ff ff	132.1.0.0	00 16 e0 bd 6e 81		

Figura 4.19 - Captura do tráfego wireless para o SYN flood attack

4.5.2 Análise de perfil do dispositivo

Esta fase do processo consiste na análise das informações enviadas pelos dispositivos monitorados. Os dados são analisados tendo como base dados coletados em situações de operação consideradas normais para o comportamento do dispositivo. Situação típica são momentos em que o usuário fica conectado na rede e qual impacto que esta situação teria no consumo e voltagem de bateria do dispositivo.

Os dados são analisados pelo *IDS_Proxy*, logo após o recebimento dos mesmos. As informações analisadas são: carga de bateria do dispositivo, a voltagem de bateria e a lista de softwares instalados no dispositivo.

Para identificação de descargas de bateria usou-se a comparação entre a carga restante no momento atual e valor que o dispositivo possuía em 5 min (cinco minutos) anteriores. Se a diferença entre este dois valores é superior a 5% (cinco por cento), considera-se que há demanda por recursos fora do padrão normal. Uma vez que a média para o usuário realizando atividades comuns na rede acessando *e-mails* e visualizando páginas na *Internet* durante um intervalo de 30 min (trinta minutos) não excedeu uma diferença de 11% (onze por cento) para o intervalo entre o pedido de conexão e desconexão na rede. Os valores medidos

são mostrados na Tabela 4.3. A média para os valores obtidos nos sete cenários foi de 11,71, valor que para melhor tratamento das informações foi arredondado para 11, definindo a assim a média de 11% (onze por cento) de consumo de bateria para o dispositivo realizando atividades consideradas normais.

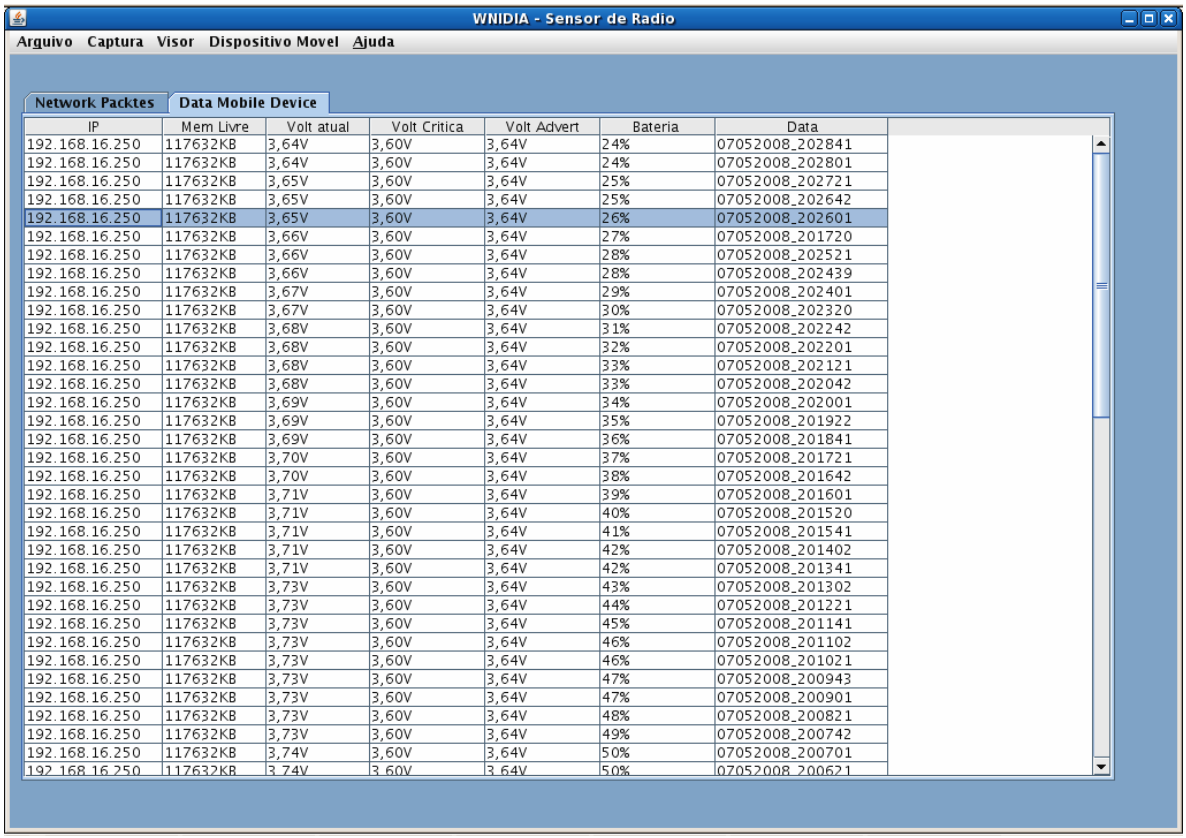
Tabela 4.4 – Valores lidos para a carga de bateria do dispositivo em modo de operação normal.

Cenário	Valor inicial	Valor final	Diferença
1º Cenário	84%	73%	11%
2º Cenário	95%	82%	13%
3º Cenário	69%	57%	12%
4º Cenário	75%	66%	9%
5º Cenário	57%	45%	12%
6º Cenário	53%	40%	13%
7º Cenário	48%	36%	12%

Embora a análise dos dados seja um fator bastante considerável no processo de detecção de ataques, os ataques são sempre confirmados pelo tráfego de rede do dispositivo monitorado. A atividade de comunicação realizada por dispositivos em redes *wireless* é uma atividade que demanda por quantidade considerável de recursos do dispositivo, que pode ter um maior impacto se o mesmo realiza atividades através da rede que também requerem uma quantidade considerável de recursos como assistir um vídeo, ouvir música ou mesmo visualizar páginas com grande quantidade de imagens. Durante os testes realizados a atividade assistir um vídeo pelo *Internet* durante 7 min chegou-se a consumir em torno de 5% (cinco por cento) a 6% (seis por cento) de carga de bateria do dispositivo.

A necessidade do cruzamento das informações enviadas pelo dispositivo com tráfego *wireless* capturado serve como critério para evitar situações de falsos positivos. Este tipo de situação poderia ocorrer em momentos onde o usuário poderia estar assistindo um vídeo na *Internet*, isso requer demanda de processamento e conseqüente carga de bateria do dispositivo que poderá exauri-se com tempo. Dependendo da duração do vídeo, essa situação poderia ser identificada como ataque ao dispositivo, pois se trata de comunicação continua com dispositivo que tem considerável impacto no consumo de bateria do dispositivo.

A Figura 4.20 mostra a rápida descarga da bateria do dispositivo em intervalos de tempo muito curtos, durante o *ping flood* ataque. Tal comportamento pode ser constatado na verificação do tráfego *wireless* ilustrado na Figura 4.20, onde é verificado que o valor carga de bateria às 202601 (20 h 26 min 01 seg.) na data de 07/05/2008, marcava o valor de 26% (vinte e seis por cento) (linha marcada nesta figura) e nos 5 min (cinco minutos) anteriores, na data de 07052008_ 262101, tinha como valor para carga de bateria atual 33% (trinta e três por cento). Com valores obtidos há uma diferença de 7% (sete por cento) nos últimos 5 min (cinco minutos). Essa diferença ultrapassa do limite estabelecido para o comportamento normal do dispositivo, que pode ser constatado com a captura do tráfego *wireless*.



IP	Mem Livre	Volt atual	Volt Critica	Volt Advert	Bateria	Data
192.168.16.250	117632KB	3,64V	3,60V	3,64V	24%	07052008_202841
192.168.16.250	117632KB	3,64V	3,60V	3,64V	24%	07052008_202801
192.168.16.250	117632KB	3,65V	3,60V	3,64V	25%	07052008_202721
192.168.16.250	117632KB	3,65V	3,60V	3,64V	25%	07052008_202642
192.168.16.250	117632KB	3,65V	3,60V	3,64V	26%	07052008_202601
192.168.16.250	117632KB	3,66V	3,60V	3,64V	27%	07052008_201720
192.168.16.250	117632KB	3,66V	3,60V	3,64V	28%	07052008_202521
192.168.16.250	117632KB	3,66V	3,60V	3,64V	28%	07052008_202439
192.168.16.250	117632KB	3,67V	3,60V	3,64V	29%	07052008_202401
192.168.16.250	117632KB	3,67V	3,60V	3,64V	30%	07052008_202320
192.168.16.250	117632KB	3,68V	3,60V	3,64V	31%	07052008_202242
192.168.16.250	117632KB	3,68V	3,60V	3,64V	32%	07052008_202201
192.168.16.250	117632KB	3,68V	3,60V	3,64V	33%	07052008_202121
192.168.16.250	117632KB	3,68V	3,60V	3,64V	33%	07052008_202042
192.168.16.250	117632KB	3,69V	3,60V	3,64V	34%	07052008_202001
192.168.16.250	117632KB	3,69V	3,60V	3,64V	35%	07052008_201922
192.168.16.250	117632KB	3,69V	3,60V	3,64V	36%	07052008_201841
192.168.16.250	117632KB	3,70V	3,60V	3,64V	37%	07052008_201721
192.168.16.250	117632KB	3,70V	3,60V	3,64V	38%	07052008_201642
192.168.16.250	117632KB	3,71V	3,60V	3,64V	39%	07052008_201601
192.168.16.250	117632KB	3,71V	3,60V	3,64V	40%	07052008_201520
192.168.16.250	117632KB	3,71V	3,60V	3,64V	41%	07052008_201541
192.168.16.250	117632KB	3,71V	3,60V	3,64V	42%	07052008_201402
192.168.16.250	117632KB	3,71V	3,60V	3,64V	42%	07052008_201341
192.168.16.250	117632KB	3,73V	3,60V	3,64V	43%	07052008_201302
192.168.16.250	117632KB	3,73V	3,60V	3,64V	44%	07052008_201221
192.168.16.250	117632KB	3,73V	3,60V	3,64V	45%	07052008_201141
192.168.16.250	117632KB	3,73V	3,60V	3,64V	46%	07052008_201102
192.168.16.250	117632KB	3,73V	3,60V	3,64V	46%	07052008_201021
192.168.16.250	117632KB	3,73V	3,60V	3,64V	47%	07052008_200943
192.168.16.250	117632KB	3,73V	3,60V	3,64V	47%	07052008_200901
192.168.16.250	117632KB	3,73V	3,60V	3,64V	48%	07052008_200821
192.168.16.250	117632KB	3,73V	3,60V	3,64V	49%	07052008_200742
192.168.16.250	117632KB	3,74V	3,60V	3,64V	50%	07052008_200701
192.168.16.250	117632KB	3,74V	3,60V	3,64V	50%	07052008_200621

Figura 4.20 - Dados enviados pelo dispositivo durante o ping flood ataque

4.6 CONCLUSÃO

Neste capítulo, apresentou-se uma visão das funcionalidades do protótipo implementado e comprovação de suas funcionalidades e sua eficácia através dos testes realizados na forma de ataques.

Foram descritos dois ataques:

- *Ping Flood* - apresenta um intervalo de detecção menor devido ao impacto do mesmo sobre o dispositivo, onde consumo sobre a carga de bateria é detectado rapidamente através de comparações a partir dos valores da carga de bateria e voltagem lidos anteriormente com os valores atuais. O motivo da descarga acelerada da bateria, poderá ser constatado através do tráfego anormal na rede wireless destinado e originado ao dispositivo;
- *SYN flood* - apresenta um maior intervalo de detecção, isso é ocasionado pelo menor impacto sobre o consumo dos recursos do dispositivo, sendo necessário um maior intervalo de tempo para gerar uma diferença considerável sobre os valores observados para o dispositivo.

Assim, uma vez detectado o tráfego wireless como malicioso, o histórico é analisado através das informações do perfil do dispositivo. A Tabela 4.4 apresenta o tempo de detecção para os ataques citados anteriormente. Portanto, uma vez que estes valores estiverem fora do padrão normal, serão consideradas como ameaças ao dispositivo.

Tabela 4.4 - Lista de Ataques

Ataque	Categoria	Tempo de detecção	Contramedida
Ping flood	Flooding	3 a 6 minutos	Encerrar conexão no cliente
SYN flood	Flooding	5 a 7 minutos	Encerrar conexão no cliente

Os resultados demonstram a potencialidade dos ataques e os impactos que eles podem ter, se realizados em grande escala. Apesar dos ataques utilizarem técnicas bem simples, as conseqüências têm efeitos consideráveis devido às características dos dispositivos móveis que se tornam ponto fraco dos mesmos e que podem ser amplamente exploradas por indivíduos mal intencionados nos ambiente inseguro.

5 CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

Este capítulo discute sobre as contribuições deste trabalho, considerações finais dos resultados alcançados, limitações do trabalho e sugestões para trabalho futuro.

5.1 CONCLUSÕES DO TRABALHO

As medidas de segurança voltadas para redes *wireless* constituem um trabalho em constante evolução. Entretanto, as técnicas de segurança destinam-se a proteção do ambiente *wireless* sem grandes soluções destinadas ao usuário final.

O ambiente *wireless* é um meio heterogêneo, composto por equipamentos de pequeno a grande porte. Assim, as medidas de proteção devem levar em consideração a composição dos componentes deste tipo de ambiente.

A mobilidade dentro do ambiente *wireless* permitiu o desenvolvimento de equipamentos onde a praticidade e a portabilidade são características essenciais. No entanto, essas facilidades acabaram tendo impactos no projeto destes dispositivos. Limitações de armazenamento, processamento e dependência de baterias como fonte de energia foram os fatores dados em troca da liberdade e mobilidade. Estes fatores são determinantes quando se pensa em medidas de segurança para dispositivos móveis. Assim, ferramentas de segurança voltadas para estes dispositivos devem abranger as limitações destes dispositivos. São necessárias soluções de segurança que tenham o menor impacto possível sobre os recursos computacionais destes dispositivos.

A principal contribuição deste trabalho é a proposta de uma arquitetura de sistema de detecção de intrusos híbrido em redes *wireless* que usa o modelo de detecção por anomalias para detectar desvios de comportamento para identificação de possíveis ameaças e o modelo de detecção por assinaturas para detectar possíveis ataques provenientes da rede *wireless*. A arquitetura proposta tem como base as informações sobre o comportamento dos dispositivos móveis monitorados. Este comportamento foi inferido através de observações sobre recursos disponíveis no dispositivo e monitoramento do tráfego de rede originado e destinado a estes dispositivos.

Os módulos para *IDS_Client* e *IDS_Proxy* foram implementados e avaliados através de testes que tiveram como objetivo verificar a eficácia da solução e o impacto que determinadas ameaças podem ter no comportamento dos dispositivos móveis. Isto foi demonstrado através das simulações de ataques realizados, em que as ameaças não

identificadas podem levar a situações como exaustão de bateria em pouco tempo, gerando assim, situações de negação[10] cada vez mais freqüente sem uma explicação aparente para o usuário.

Este trabalho fornece um ambiente seguro aos usuários de dispositivos móveis através de uma ferramenta de segurança IDS, levando em consideração as limitações do ambiente e dos dispositivos móveis. Assim, foi proposta uma ferramenta baseada no modelo de IDS híbrido estendo e adaptando as funcionalidades do IDS NIDIA (LIMA, 2001). Todo o trabalho de captura, análise e identificação de ameaças é realizado a partir do *IDS_Proxy*, evitando assim um menor impacto sobre o dispositivo móvel devido suas limitações físicas.

Os projetos de pesquisa com esta finalidade ainda estão em fase de aprimoramento, visto que as técnicas de segurança voltadas para redes *wireless* ainda buscam por soluções mais eficazes. Soluções neste ambiente voltadas para dispositivos móveis, ainda estão em seu estágio inicial, mas vem ganhando adeptos nas pesquisas destinadas à proteção de dispositivos móveis devido o rápido crescimento destes nos ambientes *wireless* e a falta de medidas de segurança voltadas especificamente para estes dispositivos.

As ameaças destinadas a esses dispositivos móveis existem e com efeitos consideráveis que foram comprovadas nos testes simulados. Situações como descarga de bateria antes do tempo previsto que dependendo da intensidade do ataque podem causar um atraso na respostas as requisições do usuário. Essa situação pode ser verificada no poder computacional exigido para processar as inúmeras requisições que são feitas ao dispositivo que se reflete em uma demanda maior no consumo de bateria.

As situações foram simuladas através de ataques relativamente simples que podem ser explorados para situações mais complexas com um grau de impacto maior sobre os dispositivos móveis.

5.2 LIMITAÇÕES

Apesar deste trabalho apresentar benefícios, algumas limitações não puderam ser ultrapassadas. As limitações encontradas foram as seguintes:

- No desenvolvimento do aplicativo para o dispositivo móvel, para a obtenção de informações consideradas específicas, como voltagem do dispositivo e

[10] Momentos em que dispositivo não disponha de carga de bateria suficiente para realização de tarefas essenciais, com ler sua caixa de *e-mails*

informações sobre a bateria do mesmo, não foi possível a utilização de uma API mais genérica que atendesse uma quantidade maior de dispositivos, isso fez com que fosse adotado o ambiente de desenvolvimento para aplicações da *Palm OS, ACESS Company Profile* fazendo o uso de API específicas do ambiente;

- Os testes foram aplicados somente aos dispositivos da Palm OS, mais especificamente em PDA's;
- O módulo de contramedidas foi desenvolvido apenas para *IDS_Client* levando em consideração as características dos ataques realizados, sendo necessário acréscimo de contramedidas também por parte do *IDS_Proxy*.

Estas limitações podem ser futuramente resolvidas, algumas delas são propostas como temas de trabalhos futuros, levando desta forma a um aprimoramento do IDS proposto.

5.3 TRABALHOS FUTUROS

Como sugestão para trabalhos futuros, bem como para melhoria deste, outras pesquisas podem ser realizadas, e novas técnicas e métodos podem ser incorporados. Podem-se citar como propostas:

- Aplicações de técnicas de comunicação segura e confiável entre o *IDS_Client* e *IDS_Proxy;*
- Adaptação do *IDS_Proxy* para outras formas de ataques voltados para dispositivos móveis, tais como ataques que ocorrem sobre o protocolo UDP;
- Implementação de contramedidas por parte do *IDS_Proxy* e ampliação de contramedidas no *IDS_Client;*
- Implementação dos agentes necessários para integração ao IDS NIDIA.

REFERÊNCIAS

ACESS, **Palm Developer Network**. Disponível em: <https://pdnet.palm.com/wps/portal/pdnet/developers>. Acesso em 10/2007.

AIRDEFENSE. **Enterprise Lan Security & WLAN Monitoring. AirDefense**. Disponível em <http://www.airdefense.net/>. Atlanta, USA. Acesso em: 09/2007.

AIRMAGNET. **Enterprise Wireless Network Security and Troubleshooting**. Disponível em <http://www.airmagnet.com>. Sunnyvale, USA Acesso em: 09/2007.

AIRSNARE. **Intrusion Detection Software for Windows AirSnare**. Disponível em: <http://home.comcast.net/~jay.deboer/airsnare/>. Acesso em 09/2007.

AIRJACK. **SourceForge.net: AirJack.** Disponível em <http://sourceforge.net/projects/airjack/>. Acesso em: 09/2007.

ANSI/IEEE 802.11 Edition. **Wireless LAN Medium Access Control and Physical Layer Specifications**, Disponível em: <http://standards.ieee.org>. New York, NY. Acesso em: 09/2007, 1999.

ANSI/IEEE 802.11i Edition. **Wireless LAN Medium Access Control and Physical Layer Specifications.** Amendment 6: Medium Access Control (MAC) Security Enhancements, 2004.

ATAIDE, Ricardo Luis da Rocha. **Uma arquitetura para a detecção de intrusos no ambiente wireless usando redes neurais artificiais**. 2007. 148 f. Dissertação (Mestrado. em Engenharia de Eletricidade) - Universidade Federal do Maranhão. São Luís, MA, 2007.

ATHEROS COMMUNICATIONS. **Atheros Communications**. Disponível em <http://www.atheros.com/> . Acesso em: 09/2007.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: Guia do Usuário**. Editora Campus. São Paulo. 2006.

BSD-AIRTOOLS. **Dachboden Labs Bsd-airtools. Bsd-airtools**. Disponível em <http://www.dachb0den.com/projects/bsdairtools.html>. Acesso em: 09/2007.

BUENNEMEYER, Timothy K.; JACOBY; GRANT A.; CHIANG Wayne G.; and MARCHANY Randolph C. **Battery-Sensing Intrusion Protection System.** In 7th Annual IEEE SMC Information Assurance Workshop, West Point, NY, 2006.

BUENNEMEYER, Timothy K.; GORA, Michael; MARCHANY, Randy C.; and TRONT, Joseph G. **Battery exhaustion attack detection with small handheld mobile computers**. In Proceddings of the 40st Hawaii International Conference on System Sciences, 2007.

BUENNEMEYER, Timothy K.; NELSON, Theresa M.; CLAGETT, Lee M.; and DUNNING, Jonh P. **Mobile Device profiling and intrusion detection using smart**

batteries. In Proceddings of the 41st Hawaii International Conference on System Sciences, 2008.

D-LINK. **D-Link**. Disponível em: <http://www.dlink.com/>. Acesso em: 06/2007.

DAY, J.D., and ZIMMERMANN, H.: **The OSI Reference Model.** In Proc. of the IEEE, vol. 71, pp. 1334-1340, New Yok, NY. Dec. 1983.

ECLIPSE. **Eclipse.org home.** Disponível em: <http://www.eclipse.org/>. Acesso em: 06/2007.

E-SET. **E-Set Mobile Anti-Virus**. Disponível em: <http://www.eset.eu/products/eset-mobile-antivirus>. Acesso em: 01/2008.

DASGUPTA, D.; GÓMEZ, J.; GONZÁLEZ, F.; KANIGANTI, M.; YALLAPU, K.; and YARRAMSETTI, R. **MMDS: Multilevel Monitoring and Detection System**. In Proceedings of the 15th Annual Computer Security Incident Handling Conference, Ottawa, Canada, Junho 22-27. 2003.

FAKEAP. **Black Alchemy Projects - FakeAP**. Disponível em <http://www.blackalchemy.to/project/fakeap/>. Acesso em: 11/2007.

FEDORA, **Projeto Fedora**. Disponível em: <http://fedoraproject.org/>. Acesso em: 06/2007

GOMES, Diego Sousa; SILVA, Francisco Jose da Silva; e ENDLER, Markus. **Integrando Dispositivos Móveis ao Middleware Integrade.** In I Workshop on Pervasive and Ubiquitous Computing. Gramado, Rio Grande do Sul, Brasil, 2007.

JACOBY, Grant A; DAVIS, Nathaniel J; and MARCHANY, Randy. **Using battery constrains with mobile hosts to improve network security**. Security e Privacy Magazine, IEEE, vol. 4. pp 40-49.2006.

JPCAP. **SourceForge.net: Jpcap - Network Packet Capture Facility for Java**. Disponível em: <http://sourceforge.net/projects/jpcap/>. Acesso: 09/2007.

KASPERSKY. **Kaspersky Mobile Security.** Disponível em**:** <http://www.kaspersky.com/productupdates>. Acesso em: 01/2008.

LACKEY, J.; ROTHS, A.; and GODDARD, J. **Wireless Intrusion Detection**, IBM Global Services, April 2003.

LIBPCAP. **SourceForge.net: The Libpcap Project**. Disponível em: <http://sourceforge.net/projects/libpcap/> . Acesso em: 09/2007

LIMA, C.F.L. **Agentes Inteligentes para Detecção de Intrusos em Redes de Computadores.** 2001. 110f Dissertação (Mestrado. em Engenharia de Eletricidade) - Universidade Federal do Maranhão. São Luís, MA, 2001.

MALTORA, K; GARDENER, S; Patz, R. **Implementation of Elliptic-Curve Cryptography on Mobile Healthcare Devices.** In Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control. London, Abril 2007.

MARTIN, Thomas; HSAIO, Michael; DONG, Ha; KRINSHNASWAMI, Jayan. **Denial of service attacks on battery-powered mobile computers**. In 2nd Annual IEEE Conference on Pervasive Computing and Communications. Orlando, Florida, 2004

MCAFEE. **McAfee Mobile Security.** Disponível em**:** http://www.mcafee.com/us/research/mobile_security_report_2008.html . Acesso em: 01/2008.

NASH, Daniel C.; THOMAS L. Martin; DONG S. Ha; and HSIAO, Michael S. **Towards an Intrusion Detection System for Battery Exhaustion Attacks on Mobile Computing Devices.** In 3rd IEEE Int'l Conference on Pervasive Computing and Communications Workshops. Kuai Island, HI, 2005.

NIST. **Special Publication 800-48 Wireless Network Security (2002). NIST 800-48**. Disponível em <http://csrc.nist.gov/publications/nistpubs/800-48/NIST_SP_800-48.pdf> Acesso em: 09/2007, 2002.

NIST. **Special Publication 800-94 (2007). Guide to Intrusion Detection and Prevention Systems (IDPS). NIST 800-94**. Disponível em <http://csrc.nist.gov/publications/nistpubs/ 800-94/SP800-94.pdf>. Acesso em: 09/2007.

IEEE. **Institute of Electrical and Electronics Engineers. IEEE**. Disponível em: <http://www.ieee.org/>. Acesso em: 09/2007

PALM. **PALM Products**. Disponível em: <http://www.palm.com/br/>. Acesso em: 11/2008.

PC WORLD. **Uso de smartphones e PDAs crescerá 54% em quatro anos.** Disponível em: <http://pcworld.uol.com.br/noticias/2007/06/01/idgnoticia.2007-06-01.8763506244/>. Acesso em 02/2008.

PIRRETI, Matthew; ZHU, Sencun; NARAYANAN, Vijaykrishnan; and BROOKS, Richard. **The sleep deprivation attack in sensor networks: analysis e methods of defense**. In International Journal of Distributed Sensor Networks, Volume 2, Issue 3, pages 267-287. September, 2006.

PLESKONJIC, Dragan. **Wireless Intrusion Detection Systems (WIDS).** In 19th Annual Computer Security Applications Conference Las Vegas, USA, December, 2003.

RFC 791. **Internet Protocol**. Disponível em: http://www.ietf.org/rfc/rfc791. Disponível em: <http://tools.ietf.org/html/ rfc791>. Acesso em: 12/2007.

RFC 792. **Internet Control Message Protocol**. Disponível em: <http://tools.ietf.org/html/ rfc792>. Acesso em: 12/2007.

RFC 793. **Transmission Control Protocol**. Disponível em: <http://www.ietf.org/ rfc/rfc793.txt>. Acesso em: 12/2007.

RED-M Home. **Red-M**. Disponível em <http://www.red-m.com/>. Acesso em: 11/2007.

SCHMOYER, T. R.; LIM, Y. X.; and OWEN, H. L **Wireless Intrusion Detection and Response: A case study using the classic man-in-the-middle attack**. In IEEE Wireless Communications and Networking Conference, Atlanta Ga., March 2004.

SERVILLA, Mark; HEADY, Richard; LUGER, George; and MACCABE, Arthur. **The architecture of a network level intrusion detection system.** Technical Report CS90-20, Department of Computer Science, University of New Mexico, August, EUA, 1990

SILVA, Mauro Lopes Carvalho, **Modelo de IDS Remoto Baseado na Tecnologia de Agentes, *Web Services* e MDA**. 2006. 142f. Dissertação (Mestrado. em Engenharia de Eletricidade) - Universidade Federal do Maranhão. São Luís, MA, 2006.

SNORT-Wireless. **Snort-Wireless Project. Snort- Wireless**. Disponível em: <http://www.snort-wireless.org/>. Acesso em: 09/2007.

SOMMERVILLE, IAN. **Engenharia de Software**. Tradução por Mauricio de Andrade. 6ª Edição. São Paulo. Addison Wesley. 2005.

STANIFORD-CHEN, S. **Common Intrusion Detection Frameowork.** Computer Emergency Response Team. Outubro, 1998.

SUN. **Sun Microsystems**. Disponível em: <http://www.sun.com/>. Acessado em: 07/2007.

TSB Sudarshan; RAKESH K; SATISH Kumar K. **A Prototype for Tiger Hash Primitive Hardware Architecture.** In Advanced Computing and Communications, 2007, ADCOM 2007.

SYMANTEC, **Mobile Anti-Virus**. Disponível em: <http://www.symantec.com/norton/ products/index.jsp> Acesso em: 01/2008.

SWAM, Yogesh Prem; and TSCHOFENIG, Hannes. **Protecting Mobile Devices from TCP flodding attacks.** In 1st ACM/IEEE international workshop on Mobility in the evolving internet architecture. San Francisco, Califórnia.2006.

YANG, H.; XIE L.; and Sun, J. **Intrusion Detection Solution to WLANs**. In IEEE 6th CAS Symp. On Emerging Technologies: Mobile and Wireless Comm., Shanghai, China, Mai 31 – Jun 2., 2004.

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