FERNANDO AUGUSTO TEIXEIRA

DETECÇÃO DE INTRUSOS POR OBSERVAÇÃO EM REDES DE SENSORES SEM FIO

Belo Horizonte

Outubro de 2005

FERNANDO AUGUSTO TEIXEIRA

DETECÇÃO DE INTRUSOS POR OBSERVAÇÃO EM REDES DE SENSORES SEM FIO

Dissertação apresentada ao Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Belo Horizonte

Outubro de 2005

Resumo

Este trabalho propõe um sistema detecção de intrusos para Redes de Sensores Sem Fio (RSSF) baseado nas informações disponíveis na estação base. RSSF são redes ad hoc compostas por diversos nós sensores com recursos limitados, podendo ser usadas para monitorar áreas de interesse. Sua aplicação varia desde reconhecimento de campos de batalha até proteção ambiental. Em algumas dessas aplicações a rede certamente será atacada, e um sistema de detecção de intrusos precisará ser implantado para detectar a presença de intrusos. Dadas suas características específicas, as RSSF precisam de um sistema de detecção de intrusos próprio. Entretanto, seu projeto e implementação são desafiadores devido às restrições de recursos. O sistema proposto não requer alterações na rede que será analisada. Propomos um modelo de informação para detecção de intrusão condizente com as RSSF e uma arquitetura extensível, que pode ser adaptada para diferentes modelos de comportamento e diferentes estratégias de detecção. Nos experimentos realizados, a taxa de detecção ficou acima de 80% e a quantidade de falsos positivos ficou em 405 para 4000 eventos analisados.

Abstract

This work proposes an Intrusion Detection System for Wireless sensor networks (WSNs) using the information available in the base station. WSNs are ad hoc networks comprised mainly of small sensor nodes with limited resources, and can be used for monitoring areas of interest. Applications range from battlefield reconnaissance to environmental protection. In some of these applications the network will certainly be attacked, and an intrusion detection system needs to be in place to detect the presence of intruders. Given their distinguishing characteristics, WSNs require tailored intrusion detection systems. However, their design and implementation are challenging because of the resource constraints. The proposed system does not require changed in the analyzed network. We propose an information model for intrusion detection in the RSSF and an extensible arquitecture, which can be adapted for different comportment models and different detection strategies. The experiments show that the detection rate was above 80% and the number of false positives was 405 in 4000 analyzed events.

Às minhas queridas filhas Fernanda e Eduarda.

À minha esposa Jacqueline.

À minha mãe Helena e pai Nivaldo.

E às minhas irmãs Jaíne e Daiane.

Agradecimentos

Agradeço a Deus pela oportunidade de fazer o mestrado nesta Universidade e por todas as outras bênçãos que recebi.

Ao meu orientador José Marcos, pelos ensinamentos, dicas e companheirismo durante a realização do mestrado e desde quando começamos a trabalhar juntos em projetos do SIS. Agradeço, também, sua paciência e confiança em meu trabalho.

À minha co-orientadora, Professora Wong, que me mostrou a beleza da área de segurança em sistemas distribuídos e procurou me passar o que há de melhor no trabalho profissional de pesquisa.

Ao Professor Loureiro, meu primeiro orientador, que, ainda nos tempos de graduação, despertou em mim a paixão pela pesquisa e me ajudou a dar os primeiros passos nessa arte que a cada dia me interesso mais.

À professora Linnyer Beatryz que me estimulou a trabalhar com Redes de Sensores Sem Fio e sempre prestou um apoio decisivo em meus trabalhos, mesmo antes de me enveredar pelo mestrado.

Ao professor Dorgival, pelo incentivo e confiança durante o processo de seleção no mestrado e pelos ensinamentos profissionais e acadêmicos durante o tempo que trabalhamos juntos.

À amiga e parceira de pesquisa Ana Paula, pelos ensinamentos e compartilhamento de objetivos de pesquisa de forma ética e profissional.

A todos os amigos do grupo sensornet, em especial ao grupo de segurança, Sérgio, Leo e Ana Paula e ao pessoal do laboratório SIS.

Em especial agradeço à minha família. Minha mãe, Helena, que sempre batalhou muito para garantir que eu pudesse estudar e é a principal responsável por eu ter chegado até aqui. Agradeço também pela sua atuação como "babá" da Eduarda e da Fernanda para que eu pudesse continuar me dedicando ao mestrado na reta final. Ao meu pai, Nivaldo, pela ética e garra profissional na qual me inspiro. Às minhas irmãs Jaíne e Daiane, pela ajuda direta ou indireta. À minha esposa Jacqueline e às minhas filhas, Fernanda e Eduarda, por terem tolerado minha ausência e sacrificado horas de seu próprio lazer em prol deste trabalho.

Sumário

Lista de Figuras

Capítulo 1

Ataques e Detecção de Intrusos em RSSF

Redes de Sensores Sem Fio (RSSF) são redes ad-hoc formadas por nós sensores dotados de recursos escassos (diminuta reserva de energia, pequena largura de banda, baixo poder computacional e pequena capacidade de armazenamento), um ou mais nós soverdouros que conectam a rede a uma estação base através da qual um observador pode extrair e analisar dados da rede [90,92], a Figura 1 ilustra uma configuração típica desse tipo de rede. Dotadas tipicamente de centenas de nós, essas redes possuem diversas aplicações em potencial, incluindo reconhecimento de campos de batalha, operações de salvamento e proteção ambiental [73, 74, 75, 76, 77]. Algumas dessas aplicações poderão ser críticas como, por exemplo, aplicações militares em caso de guerra e mapeamento das reservas de petróleo, o que certamente atrairia ataques de adversários.

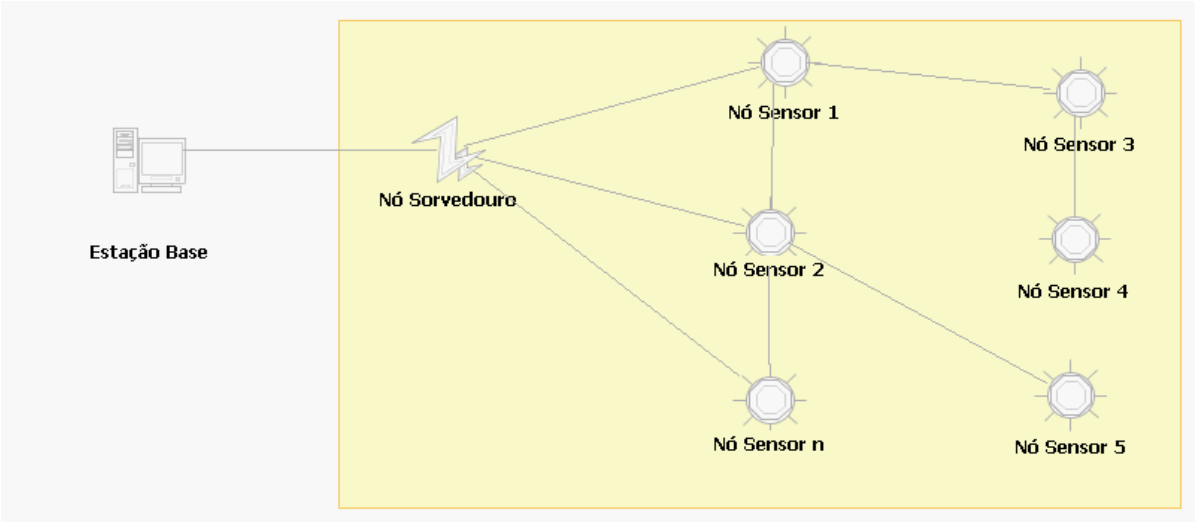


Figura 1: Exemplo da organização típica de uma RSSF.

As RSSF possuem características que as tornam vulneráveis a ataques [5, 6, 13]. Além das conhecidas vulnerabilidades associadas à comunicação sem fio e organização ad-hoc, os nós sensores são dispositivos pequenos e baratos que dificilmente contarão com proteção contra violação física. Além disso, essas redes são usualmente depositadas em áreas abertas e desprotegidas, muitas vezes hostis.

Pode-se proteger as RSSF contra alguns tipos de ataques utilizando-se mecanismos preventivos. Para prevenir que nós adversários se façam passar por membros legítimos de uma rede, por exemplo, podem ser utilizados mecanismos de identificação baseados em métodos criptográficos [6,7,8,9,10,11,12,14]. Existem, entretanto, ataques para os quais não são conhecidos mecanismos de prevenção. O ataque de "canalização" (*wormhole*) [5, 13] é um exemplo. Para esses casos, a rede precisaria de um sistema de detecção de intrusos (IDS). Mesmo para ataques para os quais já existam mecanismos de prevenção, a necessidade de um IDS se justifica porque mesmo os mecanismos mais eficazes de prevenção podem falhar.

Os IDSs existentes, como por exemplo os propostos em [29, 30], não são diretamente aplicáveis às RSSF. A detecção de intruso pressupõe que o comportamento do intruso se diferencia do comportamento normal da rede. Porém, o comportamento normal de uma rede estruturada ou de uma rede ad-hoc é muito diferente do comportamento normal de uma RSSF. Como conseqüência, o comportamento do intruso também é diferente. Mesmo os sistemas projetados para redes que possuem recursos escassos [31], como as redes ad-hoc, não são diretamente aplicáveis. Uma pesquisa de detecção de intrusos dirigida a RSSF torna-se, assim, imprescindível.

1.1. Objetivos e Contribuições

Propomos a criação de um sistema de detecção de intruso capaz de monitorar RSSF já projetadas ou novas, sem a necessidade de alterar o software ou hardware dos nós sensores e, por conseqüência, sem disputar recursos dos mesmos. As contribuições deste trabalho são:

- Proposição de um IDS, baseado no modelo de informação das RSSF e adequado às suas limitações e especificidades, sem a necessidade de alterar o software ou hardware dos nós sensores;
- Definição de um modelo de informação para detecção de intrusos em RSSF baseado nas características da rede e organizado através de mapas modelados através de técnicas de orientação a objetos [99];

- Definição de um modelo de comportamento normal e uma estratégia de detecção de intrusos;
- Desenho de uma arquitetura do IDS, adaptável a diferentes tipos de redes;
- Projeto e implementação de um protótipo do IDS;
- Avaliação da eficácia do IDS proposto, para a detecção de cinco diferentes tipos de ataques;
- Resenha e categorização dos principais artigos estudados.

Como apresentamos em [35], a detecção de intrusos em RSSF apresenta grandes desafios. Os resultados de pesquisa na área de detecção de intrusos para outros tipos de redes auxiliam na definição de métodos e arquitetura de sistema de detecção de intrusos para RSSF, mas as especificidades das RSSF impedem que as soluções existentes sejam utilizadas diretamente.

Apresentaremos a seguir as limitações relacionadas à segurança em RSSF, exemplos de ataques contra a RSSF já documentados, fundamentos da detecção de intrusos e os principais trabalhos relacionados.

1.2. Formas de Intrusão em Redes de Sensores Sem Fio

Diversas formas de intrusão vêm sendo documentadas na literatura. A seguir, apresentaremos as principais classes de ataques aos quais as RSSF são vulneráveis [5, 13].

Um dos possíveis ataques à camada física é a "interferência" (*jamming*), onde o intruso inunda as freqüências de rádio utilizadas pela rede com ruídos, por um período de tempo, de forma a impossibilitar transmissões de mensagens. Um ataque semelhante, porém mais refinado, é o ataque de "colisão" (*collision*). O intruso, neste tipo de ataque, injeta um ruído no meio exatamente após uma transmissão ter sido iniciada, de forma a alterar os sinais que estão sendo transmitidos. A recepção da mensagem fica, assim, comprometida.

Como RSSF são, muitas vezes, depositadas em áreas desprotegidas, o intruso pode ter acesso físico aos nós, o que lhe permite violá-los fisicamente. A "violação física" (*tampering*) pode visar modificação, substituição ou destruição de hardware ou software. Com essa violação, o intruso pode ter tanto o intuito de obter informações secretas, por exemplo chaves criptográficas, como ter o intuito de levar os protocolos a um comportamento anômalo, prejudicando a aplicação. Uma outra classe de ataques consiste na exaustão dos recursos da rede. Num ataque de "exaustão" (*exhaustion*), o intruso poderia, por exemplo, levar um nó a fazer um número elevado de transmissões, desperdiçando sua energia. Outros tipos de ataques são possíveis se o intruso

conseguir inserir-se na rede, fazendo se passar por um nó legítimo. Neste caso, ele poderia, por exemplo, comprometer o roteamento das mensagens através de ataques de "negligência" (*neglect and greed*) ou "retransmissão seletiva" (*selective forwarding*). Neste tipo de ataque, o intruso ignora seu papel de roteador, deixando de retransmitir algumas mensagens. O intruso pode também simular várias identidades através de ataques de *sybil,* atrapalhando os protocolos de roteamento e como conseqüência prejudicando o envio de mensagens para a estação base.

Muitos ataques são facilitados se o intruso conseguir influenciar o estabelecimento de rotas na rede, manipulando a comunicação entre nós legítimos. Essa manipulação inclui injeção de mensagens maliciosas na rede, replicação de mensagens antigas e modificação do conteúdo de mensagens válidas. As rotas, influenciadas pelo intruso, podem se passar por um nó invasor ou um nó legítimo, já violado, configurando o ataque de "buraco negro" (*blackhole* ou *sinkhole*). Um objetivo do intruso, nesse caso, é obter informações destinadas à estação base ou aplicar o ataque de negligência.

Outros ataques de manipulação de comunicação podem ser usados: num ataque de "canalização" (*wormhole*), por exemplo, o intruso canaliza uma mensagem disponível num ponto da rede para um outro ponto distante, utilizando transmissores potentes. A canalização pode levar uma mensagem a um ponto da rede aonde ela não chegaria, ou chegaria com uma latência maior. Com isso, o intruso conseguiria influenciar rotas, e, em conjunto com outros ataques, omitir informações, prejudicando a aplicação.

Alguns desses ataques podem ser prevenidos, através de diversas técnicas que vem sendo propostas tais como o SPINS [8], TinySec [7], INSENS [38], TinyPK [8], SERP [2], SEF [3], entre outros . Pode-se prevenir injeção de mensagens maliciosas, replicação de mensagens antigas e modificação do conteúdo de mensagens válidas, através do uso de protocolos criptográficos bem projetados [2, 8, 10, 11, 12]. Entretanto, existem ataques que são difíceis de serem prevenidos [5]; para estes casos e para os casos onde os mecanismos de prevenção forem comprometidos, a utilização de um IDS torna-se primordial.

1.3. Detecção de Intrusos

A detecção de intrusos é uma área de pesquisa bastante ativa mesmo para as redes convencionais. Como aponta Stallings [15], a principal motivação para o desenvolvimento de sistemas de detecção de intrusos se baseia no fato de não ser possível ou viável criar um mecanismo de defesa totalmente seguro. A detecção de intrusão, dessa forma, torna-se uma importante aliada na busca pela segurança de um sistema. Através da detecção de intrusão, é possível verificar que um

mecanismo de defesa foi violado, possibilitando a reação automática ou através da intervenção do administrador da rede. Além disso, as informações disponibilizadas através do sistema de detecção de intrusão podem ser usadas para melhorar os mecanismos de defesa.

Segundo [17], de maneira geral, há duas estratégias de detecção de intrusos:

1. **Detecção baseada em anomalias (*Anomaly Detection*)** [42, 43, 44]: Cada usuário tem um perfil em relação ao sistema que não é muito alterado ao longo do tempo. Então, qualquer desvio significante de comportamento pode ser considerado uma anomalia. Exemplos de IDS baseados em anomalias são o NIDES [17] e ESMERALD [39].
2. **Detecção baseada em mau uso (*Misuse detection*)** [39, 51, 52]: A idéia é que qualquer intrusão pode ser descrita por sua assinatura caracterizada por valores de suas funcionalidades. Sistemas que utilizam essa estratégia usam diferentes modelos como análise de transição de estados, por exemplo o STAT [50], ou em classificação mais formal de padrões, como o IDIOT [48] e o SNORT [49].

Nas redes tradicionais normalmente existe a figura do usuário, alguém que irá usar a rede e será responsável pelo padrão de tráfego da mesma. Em uma rede de sensores o comportamento é diferente: eventos são observados e coletados pelos nós sensores que devem encaminhá-los até um ponto de observação onde, então, o usuário ou observador irá analisar as informações coletadas. O comportamento do usuário, neste contexto, não é interessante do ponto de vista de detecção de intrusos porque ele não influencia o comportamento da rede, salvo em situações onde o usuário interage com a rede para configurá-la ou provoca algum tipo de estímulo. Mas isso não significa que a utilização de métodos de detecção de intrusos já existentes não seja viável. A viabilidade depende da adaptação dos métodos ao modelo de informação das RSSF. Por exemplo, a idéia de definir um comportamento como normal ou anormal baseado em um conjunto de informações disponíveis pode ser aplicado na estação base, utilizando o conjunto de informações que essa estação possui sobre a RSSF. O método utilizado pode ser baseado em regras, por exemplo. Mas as regras serão bem diferentes daquelas encontradas em uma rede TCP/IP de uma empresa, por exemplo. Não faz sentido falarmos em tentativas de *login*, ou investigarmos a seqüência de comandos do UNIX executada pelo usuário, como é feito em alguns IDS já existentes. Mas podemos pensar em quantidade de eventos gerados em um período de tempo ou consumo médio de energia dos nós sensores. Ou seja, mapas da rede poderão ser utilizados como fonte da informação sobre a qual o comportamento será traçado.

1.4. Uma Taxonomia de IDS

Há diversas propostas de taxonomias para sistemas de detecção de intrusos. Silva et al. [34] partiram das propostas apresentadas em [29, 30] e apresentaram uma nova taxonomia adaptada para o contexto de RSSF. Nas figuras 2, 3 e 4 apresentamos uma nova proposta de taxonomia, onde, a partir da proposta de Silva et al., acrescentamos a classificação quanto à forma de implantação, o impacto sobre a RSSF, a abrangência, a forma de interação com a rede, e quanto ao local onde a análise e conclusão sobre intrusão serão feitas. Além disso dividimos a classificação em três grandes grupos, de acordo com o processo de detecção de intrusão: classificação geral do IDS no contexto das RSSF, classificação quanto à maneira de se obter os dados e classificação quanto à forma de análise e conclusão sobre a presença do intruso.

Em relação à classificação geral do IDS no contexto de RSSF, mostrada na Figura 2, julgamos necessário acrescentar a categorização quanto à implantação e quanto ao impacto sobre a rede, além de manter a classificação quanto ao comportamento na detecção conforme a proposta original. Quanto à implantação, o IDS pode ser classificado de acordo com o local onde deve ser implantado: na estação base, nos nós comuns ou em nós especiais que devam ser projetados para abrigar o IDS. Outro ponto que julgamos importante acrescentar na taxonomia foi a classificação do IDS quanto ao impacto que o mesmo precisa provocar na rede para garantir seu funcionamento. Vemos duas possibilidades, o IDS pode ser invasivo no sentido que as características da rede, como software, hardware e protocolos precisem ser modificados para garantir o funcionamento do IDS ou o IDS pode ser projetado de forma que ele se adapte às características da rede sem exigir que a mesma seja alterada. Quanto ao comportamento na detecção, o IDS pode ser ativo ou reativo. Um IDS ativo informa sobre a presença de intruso enquanto um IDS reativo reage à presença do intruso.

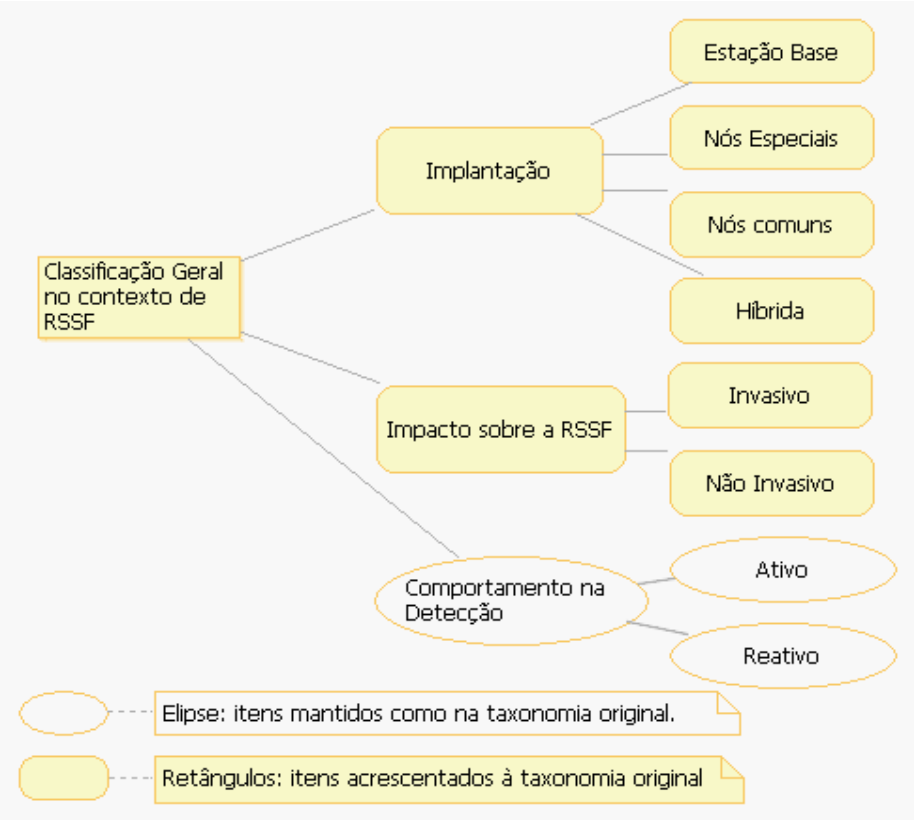


Figura 2: Classificação do IDS de forma geral no contexto de RSSF.

Quanto à obtenção de dados, acrescentamos à taxonomia proposta por Silva et al. [34], a forma de interação e a abrangência. Quanto à coleta, o IDS pode ser classificado em ativo ou passivo. Na forma ativa, informações sobre os nós ou sobre a rede são obtidas através de testes e interrogação a outros nós. Na forma passiva, as informações para a análise de intrusão são extraídas das mensagens transmitidas na própria rede. Quanto à abrangência, as informações obtidas podem ser referentes a um nó individual ou a um conjunto de nós, podendo abranger a rede inteira. Mantivemos as categorizações quanto à fonte de dados e o local de coleta de dados conforme proposto por Silva et al. [34]. A fonte de dados é definida de acordo o local onde a informação usada como entrada para o IDS está disposta, podendo ser arquivos de log, chamadas de sistemas, pacotes de rede, entre outros. O local da coleta de dados é definido como o local onde o IDS fará a coleta dos dados, podendo ser centralizado ou distribuído.

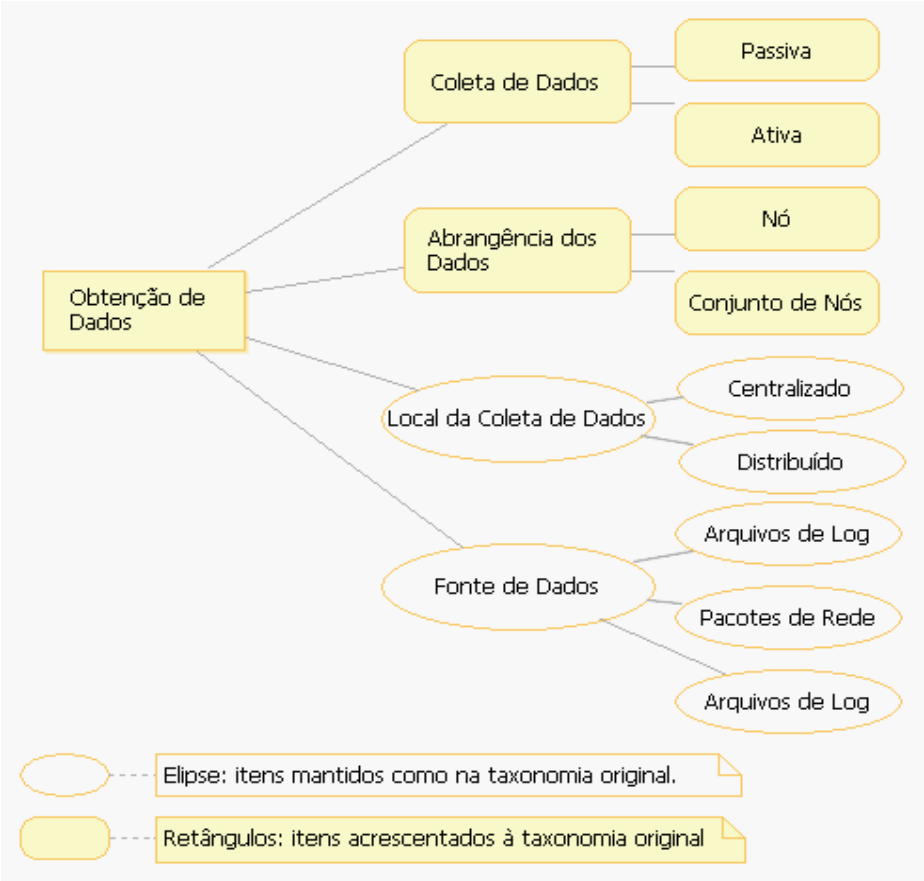


Figura 3: Taxonomia do IDS quanto à forma como a informação pode ser obtida[1].

Quanto à análise e conclusão sobre a intrusão, estendemos o conceito de local de processamento proposto na taxonomia de Silva et al. [34], através da criação de uma classificação quanto ao local onde a análise e a conclusão sobre a intrusão podem ser realizadas. Consideramos quatro locais onde a análise e a conclusão podem ser realizadas: na estação base, nos nós sensores, em nós auditores, destinados a abrigar instâncias do IDS, ou em mais de um tipo de nó. Na abordagem híbrida, a análise é feita, por exemplo, utilizando os sensores em uma primeira etapa e a estação base para realizar a conclusão. Mantivemos as categorizações realizadas na proposta original quanto ao método, o tempo e o processamento para detecção. Quanto ao método de detecção, o IDS pode ser baseado no conhecimento, no comportamento ou na especificação. Quanto aos requisitos de tempo para detecção o IDS pode ser de tempo real ou não. E, quanto à periodicidade de processamento, o IDS pode ser contínuo ou periódico.

[1] Na taxonomia original proposta por Silva et al. [34], o que chamamos de fonte de dados foi chamado de fonte de auditoria.

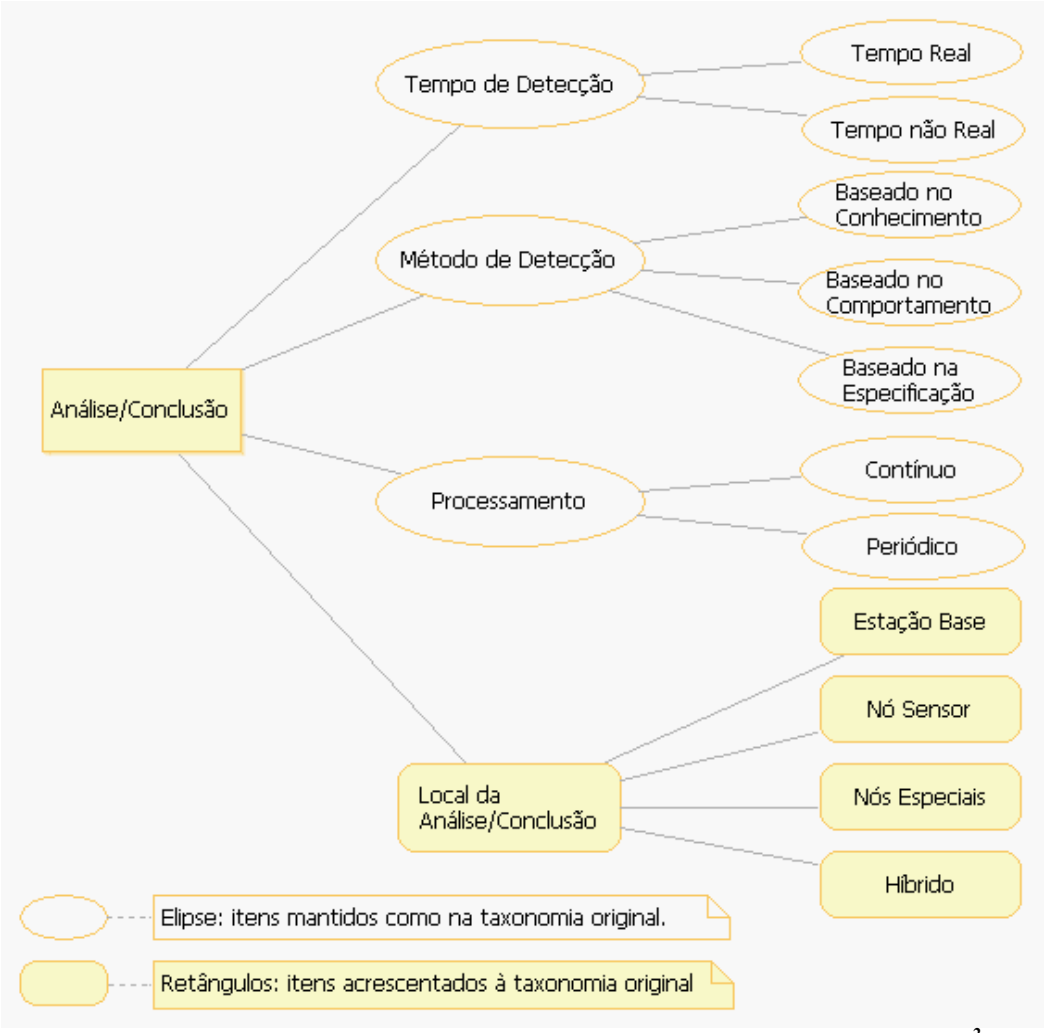


Figura 4: Taxonomia do IDS quanto à análise e conclusão sobre a presença do intruso[2].

1.5. Trabalhos Relacionados

Diversos trabalhos foram publicados propondo IDS para redes estruturadas convencionais, os artigos [29, 30] apresentam muitos exemplos destes trabalhos. Há algumas propostas também para redes móveis *ad-hoc* (MANET) como, por exemplo, o IDS proposto por Zhang e Lee [31]. Mas muito pouco tem sido feito em relação às RSSF. Os IDSs tradicionais [29, 30] foram projetados com foco nos sistemas operacionais tradicionais e em redes estruturadas, onde há recursos suficientes para o tratamento de grandes volumes de informação, muitas vezes necessários para detecção do intruso. A escassez de recursos das RSSF impede que esses sistemas sejam aplicados diretamente, motivando a realização deste trabalho. As RSSF possuem

[2] Na taxonomia original proposta por Silva et al. [34], o que chamamos de Local de Análise/Conclusão foi chamado de Local de Processamento e as subclassificações eram "Centralizado" e "Distribuído".

similaridades com as MANETs e existem algumas propostas de IDSs para este tipo de rede. Zhang and Lee [31] propõe um sistema de detecção para MANETs (*Mobile Ad-Hoc Networks)* onde cada nó da rede se comporta como um IDS e os nós cooperam entre si para detectar intrusos. As diferenças entre as RSSF e as MANETs impedem que essa solução seja aplicada diretamente em RSSF.

Diversas estratégias para detectar ataques específicos foram propostas para redes *ad-hoc* e RSSF. Hu et al. [32] apresentam uma estratégia baseada em avaliação do tempo gasto para transmitir mensagens e na autenticação dos nós para detectar o ataque de canalização em redes *ad-hoc*. Para o mesmo ataque, agora em RSSF, Pires et al. [33] propõe uma estratégia baseada na potência de recebimento do sinal. O artigo [36] propõe técnicas que podem ser utilizadas para detecção do ataque *sybil*. Marti et al. [69] apresentam uma estratégia para detectar a supressão de mensagens que deveriam ser retransmitidas por nós vizinhos. Todas essas estratégias podem ser utilizadas para detecção de tipos específicos de ataques em RSSF. No entanto, nos propomos a estudar uma solução mais abrangente, que possa detectar diversos tipos de ataques.

Além dos trabalhos relacionados à detecção de intrusos, também foram propostas estratégias para tolerar intrusos em RSSF. Deng et al [38], por exemplo, propõe um protocolo de roteamento que mantém a rede funcional sob a presença de um número moderado de intrusos, utilizando redundância de rotas. Entretanto, a maioria dos ataques listados não poderá ser tolerada sob a pena de desperdiçar os recursos da rede ou de comprometer sua segurança, o que reforça a necessidade de se desenvolver um IDS para as RSSF.

Apresentamos em [35] os principais aspectos e desafios relacionados à detecção de intruso em RSSF. O estudo serviu como ponto de partida para este trabalho e para o trabalho realizado por Silva et al. [34]. Estes últimos propõem a utilização de uma abordagem descentralizada para detecção de intrusos. O IDS proposto por eles, utiliza um método baseado na especificação do comportamento da rede e utiliza nós monitores para detectar desvios em relação a essa especificação. Os nós monitores são nós sensores que abrigam instâncias do IDS. O trabalho apresenta bons resultados para diversos tipos de ataques analisados e procurou mostrar ser possível implantar o IDS em nós físicos da classe Mica Motes [91].

Outra iniciativa, voltada para RSSF, foi apresentada por Paula et. al.[37], onde foi proposto um mecanismo para a proteção de RSSF contra alguns ataques de negação de serviço no roteamento e utilizando rotas redundantes para reagir à intrusão. A solução proposta baseia-se no estabelecimento de rotas múltiplas entre os nós sensores e na distribuição de informações através dessas rotas. A solução pressupõe o conhecimento da topologia e da conectividade da rede e, para

realizar a detecção de intrusão, utiliza um algoritmo recursivo baseado em grafo, onde a raiz é a estação base e cada nó da rede é um nó desse grafo. Após a identificação do intruso a estação base isola os nós maliciosos através do envio de mensagens para a rede e atualização do protocolo de roteamento.

Nossa abordagem difere da abordagem de Silva et. al. [34] e Paula et. al. [37], uma vez que propomos uma abordagem centralizada e baseada na observação das informações disponíveis na estação base. Apesar de ser viável alterar os nós para permitir a detecção de intrusão ou, até mesmo, construir o IDS nos nós sensores, acreditamos que muitas vezes não é conveniente utilizar os recursos, já escassos, dos nós sensores para abrigar um IDS, ou mesmo sacrificar parte dos nós sensores para agir como monitores. Propomos a utilização da estação base para implantação de um IDS capaz detectar intrusos por observação. Desta forma, evitamos a alteração das aplicações e dos protocolos da rede de sensores, desacoplando o projeto do IDS do projeto da aplicação e dos protocolos das RSSF. Além disso, mesmo as RSSF já existentes poderão usufruir de um IDS sem que suas aplicações e protocolos tenham que ser alterados e sem ter que reservar recursos dos nós sensores ou parte dos nós para a realização da análise de intrusão.

É interessante observar que a abordagem que propomos neste trabalho pode ser complementada com as soluções propostas por Silva et. al. [34] e Paula et. al. [37]. Nossa solução utiliza mapas de informação, como foi proposto por Ruiz et al. [90]. Desta forma, a informação derivada do trabalho de Silva et. al. [34] pode ser facilmente organizada na forma de um mapa e ser utilizada no contexto deste trabalho para refinamento da nossa proposta. Além disso, a arquitetura proposta permite que algoritmos como o de Paula et. al. [37] sejam utilizados na estratégia de análise.

1.6. Organização do Texto

O texto é composto por 6 capítulos e um Apêndice. O capítulo 2 apresenta o escopo para o qual o trabalho se propõe a resolver o problema de detecção de intrusos; o capítulo 3 descreve a estratégia para detectar intrusos de forma não invasiva, o modelo de informação e o método de análise de intrusão; o capítulo 4, apresenta uma proposta de arquitetura para o IDS, baseada na estratégia descrita no capítulo 3; o capítulo 5, mostra como a solução foi validada e apresenta os resultados da avaliação realizada; o capítulo 6 apresenta as conclusões do trabalho e indica trabalhos futuros; o apêndice apresenta a resenha de alguns trabalhos relacionados que foram utilizados para determinar o modelo de comportamento ou o método de detecção.

Capítulo 2

Requisitos de um Sistema Detecção de Intrusos por Observação

2.1. Introdução

Muitas vezes é necessário ou desejável utilizar um sistema de detecção de intrusos sem, entretanto, alterar a RSSF alvo. Ou seja, será necessário utilizar uma solução não invasiva e centralizada na estação base, que não imponha, portanto, alterações nas características da RSSF a ser protegida. Um exemplo disso, são possíveis redes que forem projetadas sem a preocupação com a segurança inicialmente. Para este caso a introdução de detecção de intrusos na RSSF que envolvam alterações no software ou hardware dos nós pode ser cara ou até proibitiva, devido a restrições de hardware ou particularidades das aplicações ou dos protocolos da rede que não foram projetados levando-se em consideração requisitos de segurança. Outro exemplo, são as RSSF onde as aplicações e protocolos de rede necessitem dos recursos do hardware dos nós em sua plenitude. Para estes casos, mudar o algoritmo dos nós para comportar parte ou todo o IDS pode ser inviável pois implicaria em otimizar ainda mais os algoritmos das aplicações ou aumentar a capacidade física dos mesmos.

Neste capítulo apresentaremos os principais requisitos relacionados à construção de um IDS para RSSF utilizando uma abordagem centralizada. Em especial, será definido o escopo de desenvolvimento do trabalho. O principal requisito é que a solução seja eficiente e possa ser aplicada em RSSF já existentes, sem alteração de suas características. Consideramos que um IDS não invasivo deve obedecer as seguintes restrições:

- Os protocolos da rede não poderão ser modificados para atender ao IDS, que deve utilizar apenas informações já disponíveis na RSSF alvo.
- O software dos nós sensores não poderá ser alterado: não serão feitas modificações no software dos nós sensores única e exclusivamente para atender ao IDS.
- O hardware dos nós sensores não poderá ser expandido: não há recursos ou não se deseja utilizar sensores com maior capacidade de hardware exclusivamente para viabilizar o IDS.

Em RSSF as decisões devem ser tomadas levando-se em conta os recursos disponíveis e a aplicação considerada. A aplicação poderá exigir mais ou menos investimento em segurança, dependendo da sua necessidade de proteção. Os recursos disponíveis limitarão a capacidade do sistema de segurança tanto do ponto de vista de prevenção como do ponto de vista de detecção e reação.

2.2. Características das RSSF consideradas no trabalho

Consideramos que a rede alvo será homogênea, plana, simétrica, estática e contínua, conforme a classificação proposta por Ruiz et al [90]. Em uma RSSF homogênea, os nós sensores possuem mesmo hardware e software. Em uma RSSF simétrica[3], um nó A pode se comunicar com o nó B se e somente se B puder se comunicar com o nó A. Consideramos redes estáticas, ou seja, aquelas onde os nós não se movem após sua implantação. Finalmente, consideramos redes onde os nós produzem dados em intervalos contínuos, ou seja, a cada *x* segundos é esperado que um dado sensoriado seja enviado para a estação base. Exclui-se, portanto, as redes orientadas a eventos.

A exclusão das redes orientadas a eventos[4] restringe a solução proposta. A inclusão deste tipo de rede no escopo tornaria a solução restrita a apenas algumas aplicações porque a determinação do comportamento esperado passa a depender do tipo de evento e da freqüência em que os eventos ocorrem. O mesmo ocorre em relação às redes dinâmicas[5]. Para analisar o comportamento normal de redes dinâmicas é necessário avaliar o padrão de movimentação de sensores na RSSF alvo. Ou seja, para analisar o comportamento de redes orientadas a eventos e redes dinâmicas, é necessário determinar uma aplicação específica e estudar qual é o

[3] A comunicação direta e unidirecional da estação base para os nós é viável em alguns tipos de RSSF, mas, para que a solução fique mais abrangente, não contaremos com essa característica.
[4] Redes Orientadas a eventos são redes onde o dado sensoriado só é transmitido quando um evento de interesse ocorre [90]
[5] Redes Dinâmicas são redes onde os nós sensores podem se mover [90].

comportamento esperado. Preferimos restringir a solução às redes contínuas e estáticas ao invés de restringi-la a algum tipo de aplicação específico.

As redes consideradas no escopo do trabalho são planas. Ou seja, não há líderes de grupos ou hierarquia na rede. Em RSSF hierárquicas é comum parte dos nós realizarem agregação, fusão ou filtragem de dados o que dificulta a detecção não invasiva pois a estação base acaba não tendo acesso às informações dos nós das pontas. Neste tipo de rede, uma solução de detecção com o IDS implantado nos nós sensores é mais indicada.

Os nós da rede são identificados individualmente, desta forma a estação base tem condições de determinar qual nó produziu determinada informação. Alguns autores [81, 94, 95] consideram que os nós sensores não possuem identificação individual, mas há diversos trabalhos onde o nó possui identificação [72, 73, 74, 75]. Um exemplo prático de nós que possuem identificador são os Mica Motes que vem sendo usados em diversos trabalhos [70, 71,77, 78, 79, 80].

Há na rede, pelo menos, uma estação base e dezenas ou centenas de nós sensores. Estação base é o nome dado à entidade que coleta os dados dos nós sensores, serve de interface com outras entidades tais como redes ou usuários e não possui restrições de hardware tais como os nós sensores. Os nós sensores são equipamentos de baixa potência e baixo custo. Os mica motes são exemplos deste tipo de nó [5, 91], agrupados em três famílias MICA, MICA2 e MICA2DOT. Constitui uma pequena unidade de processamento e sensoriamento, cerca de 1 polegada de tamanho, com CPU, fonte de energia, rádio e elementos de sensoriamento com sistema operacional TinyOS. O nó MICA2, por exemplo, possui CPU Atmel ATmega 128L com freqüência de 4MHZ e 8 bits de endereçamento, 128KB de memória de instrução, 4KB de memória RAM e 512KB de memória flash. A CPU consome 5,5 mA sob tensão de 3 volts quando ativa e 2 ordens de grandeza a menos quando no modo sleeping. Possui rádio CC1000 Chipcon que atua na freqüência de 916MHz e taxa de transmissão de 76,8 Kbps tensão de 2,7 a 3 V e alcance de 150 metros[6]. O rádio consome 4,8mA (3v) no modo de recebimento, 12mA no modo de transmissão e 5µA no modo *sleep*. Possui um *sensor board* que permite a instalação de diferentes tipos de sensores tais como temperatura, acelerômetro, magnetômetro, som entre outros. Os nós são alimentados por duas baterias AA que fornecem 2050*mA* hora sob tensão de 3 volts.

[6] Em experimento em frente a reitoria da UFMG conseguimos um alcance de no máximo 12 metros, mas segundo o fabricante é possível um alcance máximo de 150 metros.

2.3. Caracterização do Intruso e Restrições de Segurança

A seguir caracterizaremos o intruso através da apresentação das principais restrições de segurança consideradas em relação ao meio de transmissão, aos nós sensores, à estação base e ao ambiente onde os nós serão implantados.

Como os nós são implantados em ambiente inseguro, o intruso é capaz de utilizar diversos meios para invadir a RSSF, como ilustra a Figura 5. O intruso é capaz de capturar nós da rede ou implantar nós com capacidade de hardware e software similar aos dos nós legítimos. Além disso, os nós não são resistentes à violação física, o que implica que o intruso é capaz de extrair dados e código armazenado e ou sobrescrever sua memória. Tal suposição é importante uma vez que uma das principais restrições de projeto dos nós sensores é que eles sejam de baixo custo o que inviabiliza a construção de sensores resistentes à violação. Outro problema, decorrente da insegurança do ambiente, é que o intruso será capaz de utilizar laptops ou outros equipamentos para realizar os ataques.

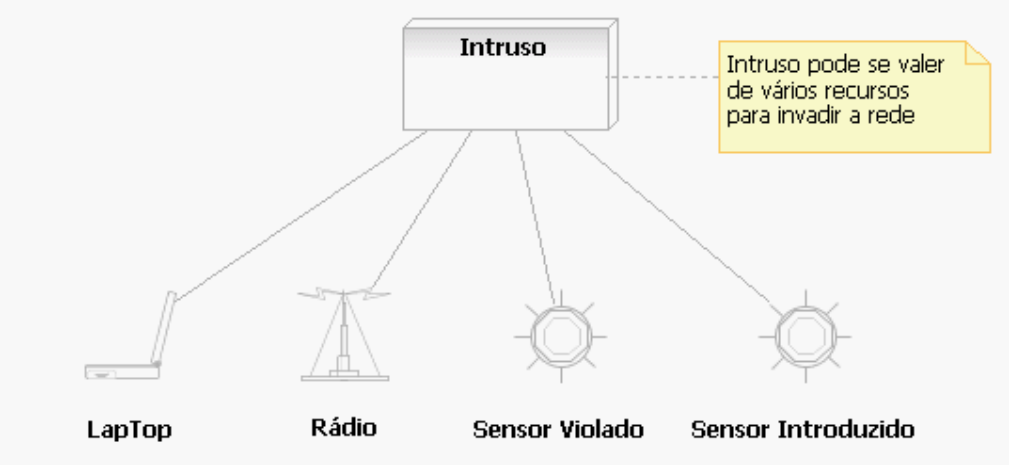


Figura 5: Exemplos de recursos utilizados pelo Intruso.

Os enlaces de rádio são inseguros, ou seja, o intruso é capaz de interceptar pacotes de bits enviados, replicá-los ou injetar bits no meio de transmissão. Não é considerado como premissa que a rede possua algum mecanismo de prevenção a ataques como, por exemplo, criptografia dos dados transmitidos. A estação base, por outro lado, é considerada segura. Os problemas de segurança da estação base fogem do escopo do trabalho uma vez que recaem no problema de redes convencionais. E, finalmente, assumiremos que não poderemos confiar nos nós sensores porque eles podem ter sido violados ou acrescentados pelo intruso. Desta forma as informações geradas pelos nós podem ser suprimidas, alteradas ou replicadas pelo intruso.

2.4. Informações Necessárias para o Funcionamento do IDS

Assumimos um subconjunto mínimo de informações utilizado pelo IDS formado por: produção, estado operacional e roteamento. Tais informações foram escolhidas porque é possível para estação base obtê-las diretamente ou inferi-las e são relativamente suficientes para o IDS analisar se há intrusão ou não. Obviamente, se for possível trabalharmos com um conjunto maior de informações a eficácia do IDS tende a aumentar, mas preferimos trabalhar com este subconjunto mínimo com o objetivo de garantir que o IDS aqui proposto abranja um maior número de redes.

As informações sobre produção indicam quais nós estão produzindo e quais são os valores dos dados produzidos e são facilmente observadas pela estação base, pois é esperado que os dados sensoriados cheguem até a estação base em intervalos periódicos. A informação sobre a estrutura de roteamento é necessária para verificar, por exemplo, se a ausência de informação de um nó ou de um conjunto de nós está sendo provocada por uma falha no caminho. Outra informação importante é se o nó está operacional, ou seja, se está apto a produzir ou não. O nó pode deixar de produzir porque não está mais funcionando, por falta de energia por exemplo, ou porque está configurado para não produzir. Há duas classes de situações onde o nó deixa de produzir:

- Falha acidental interna: houve uma falha no nó ou na rota que o nó utiliza para transmitir seus dados. Essas falhas estarão embutidas no modelo de falhas da rede.
- Falha provocada por intruso: a ação do intruso impede que o nó produza ou que sua produção chegue até a estação base.

Dentre essas informações a mais complexa de se obter é se o nó está operacional ou não, pois, como não confiamos nos sensores e não queremos alterar seus protocolos e *softwares*, não podemos interrogá-los. A mesma dificuldade será encontrada para os casos onde a estação base não toma conhecimento da árvore de roteamento criada. Para esses casos, veremos no próximo capítulo, que poderemos trabalhar com modelos probabilísticos que nos fornecerão, de forma aproximada, as informações que necessitamos.

2.5. Conclusão

Neste capítulo apresentamos as principais características das RSSF para as quais propomos um mecanismo de detecção de intrusão. Estamos interessados nas RSSF planas, homogêneas, contínuas, estáticas, com nós semelhantes aos nós reais mica motes. Listamos as principais restrições de projeto que farão da solução uma alternativa não invasiva, tais como a não alteração de hardware, software e protocolos da rede. Definimos o modelo de segurança onde o intruso é

capaz de escutar e alterar dados do ambiente, violar e inserir nós e utilizar outros tipos de hardware como um laptop. Finalmente, definimos o subconjunto mínimo de informações que estamos considerando no escopo deste trabalho, que consiste nas informações de produção, estado operacional e informações sobre o roteamento. No próximo capítulo, apresentaremos uma proposta de solução considerando as restrições, premissas e requisitos aqui apresentados.

Capítulo 3

Estratégia de Detecção

3.1. Introdução

Neste capítulo apresentaremos uma estratégia para detectar intrusos observando as informações disponíveis na estação base. Em particular, mostraremos uma forma de organizar as informações com o objetivo de detectar indícios de intrusão. Além disso, apresentaremos uma estratégia de análise dessa informação de maneira diferenciar o comportamento normal do comportamento na presença de intruso e lidar com a incerteza inerente da falta ou inviabilidade de se obter algumas informações.

Um dos objetivos do trabalho é propor uma estratégia de detecção de intrusos de maneira não invasiva, ou seja, deseja-se um IDS que seja capaz de detectar os intrusos sem alterar as aplicações, protocolos ou o hardware dos nós sensores. Para tanto, nos propomos a utilizar as informações já disponíveis na estação base, ou seja, propomos uma estratégia centralizada na estação base de maneira que não seja necessário alterar as aplicações, protocolos ou o hardware dos nós sensores, como ilustra a Figura 6. Dessa maneira a estação base passa a ser o elemento principal na estratégia de detecção, mas os nós sensores participarão direta ou indiretamente fornecendo informações.

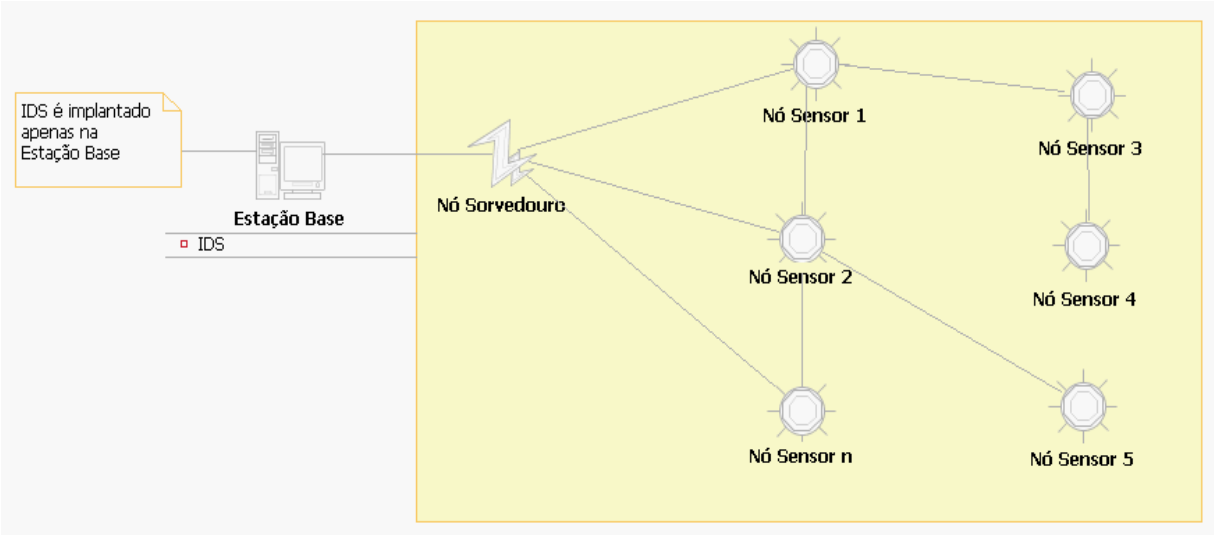


Figura 6: Visão de implantação da solução proposta.

Informações sobre os nós ou sobre a rede podem ser obtidas através de testes e interrogação a outros nós (forma ativa), ou podem ser conseguidas através de informações extraídas de protocolos já utilizados na rede (forma passiva). As informações obtidas podem ser referentes a um nó individual ou a um conjunto de nós, podendo abranger a rede inteira. Utilizaremos a abordagem passiva uma vez que temos como premissa provocar o mínimo de interferência na rede e não confiar nas respostas dos nós sensores, pois os mesmos podem ter sido invadidos.

O IDS utilizará como entrada os eventos que chegam até a estação base e será responsável por analisá-los com objetivo de detectar a presença de intruso. Dessa forma, o observador – a estação base – possui uma visão global da rede, o que possibilita a correlação de eventos. Além disso, a estação base não possui restrições de recursos tão severas quanto aquelas encontradas nos nós, o que facilita a análise dos eventos.

Trata-se, portanto, de uma abordagem centralizada e não invasiva. Tal abordagem se mostra atraente principalmente nos casos onde os nós sensores não sejam capazes de participar diretamente do IDS ou não se deseje alterar sua configuração. Nessa abordagem apenas a estação base precisa receber o IDS, o que facilita sua implantação e manutenção. Além disso, a estação base possui a visão global da rede e suas restrições de recursos são menos severas o que abre a possibilidade de se utilizar métodos encontrados nos IDS já existentes.

3.2. Modelo de informação

Segundo Stallings [58], o problema de detecção de intruso, em última instância, se resume em diferenciar o comportamento normal de um comportamento considerado anômalo devido à presença de um intruso. Para se definir o que é normal, é necessário conhecer as características das redes que serão analisadas, ou seja, é necessário conhecer o modelo de informação utilizado pela rede alvo.

Nas redes tradicionais existe a figura do usuário, alguém que irá usar e será responsável pelo padrão de tráfego da rede. Em uma rede de sensores o comportamento é diferente. Eventos são observados e coletados pelos nós sensores que devem encaminhá-los até um ponto de observação onde, então, o usuário irá analisar as informações coletadas. Neste contexto, o comportamento do usuário não é interessante do ponto de vista de detecção de intrusos porque ele não influencia o comportamento da rede, salvo em situações onde o usuário interage com a rede para configurá-la ou provoca algum tipo de estímulo, o que foge do escopo deste trabalho.

Mas isso não significa que a utilização de métodos de detecção de intrusos já existentes não seja viável. A viabilidade depende da adaptação dos métodos ao modelo de informação das RSSF. Por exemplo, a idéia de definir um comportamento como normal ou anormal baseado em um conjunto de informações disponíveis pode ser aplicado à estação base utilizando o conjunto de informações que a estação possui sobre a rede de sensores. O método utilizado pode ser baseado em regras como acontece em vários IDS tradicionais [48, 49, 50, 51, 52], por exemplo. Mas as regras serão bem diferentes daquelas encontradas em uma rede TCP/IP de uma empresa. Não faz sentido falar em tentativas de login, ou investigar a seqüência de comandos do UNIX executada pelo usuário, como é feito em alguns IDS já existentes. Mas podemos pensar em quantidade de eventos gerados em um período de tempo ou no consumo médio de energia dos nós sensores.

Propomos que o modelo de informação numa abordagem centralizada seja composto por mapas montados a partir de informações reportadas pelos nós. Diversos tipos de mapas podem ser considerados. Em [90] foi proposta uma arquitetura de gerenciamento para RSSF, chamada *Manna*, cuja arquitetura de informação baseava-se em informações estáticas e dinâmicas. As informações dinâmicas foram organizadas em forma de mapas de informações dos nós ou da rede, tais como:

- Mapa de energia: energia remanescente de cada nó;
- Mapa de área de Sensoriamento;

- Mapa de área de cobertura de comunicação: deriva de informações de conectividade entre os nós e potência de transmissão;
- Mapa de produção: informação de quais nós estão coletando e retransmitindo dados;
- Modelo de dependência: informação de dependência funcional entre nós;
- Modelo estrutural: representa a agregação e conectividade entre elementos de rede;
- Modelo de cooperação: representa as relações de interação entre elementos de rede;
- Mapas de auditoria: representa registros que permitam verificar se houve uma violação de segurança ;
- Mapa de cobertura: determinado através da combinação de informações de cobertura de rede e área de sensoriamento;

Os mapas desse tipo, podem ser organizados de forma que definam o comportamento normal da rede. Além disso podem ser construídos ao longo do tempo de maneira a representar o comportamento real da rede alvo. O mapa representa uma foto da rede ou de informações dos nós, ou seja, representa um conjunto de informações em um momento específico. O conjunto das fotos indicará qual é o comportamento da rede. O comportamento normal da rede pode ser representado pelo conjunto de mapas da rede esperados ao longo do tempo. A comparação entre o conjunto de mapas que representa o comportamento real e o conjunto de mapas que representa o comportamento esperado nos leva à detecção do intruso.

A seguir apresentaremos uma visão estruturada do modelo de informações de uma RSSF que servirá como base para nossa estratégia de detecção de intrusos. O modelo é inspirado na visão de mapas proposto por Ruiz et al [90], mas procuramos organizar as informações de maneira que possam ser utilizadas por sistemas de detecção de intrusos definindo quais informações estão, normalmente disponíveis, como essas informações se relacionam e como os mapas podem ser organizados para representar o comportamento normal e o comportamento real da rede.

3.3. Estruturação do Modelo de Informações

Utilizaremos uma abordagem orientada a objetos para organizar as informações presentes em uma RSSF, de maneira que o sistema de detecção de intrusos possa utilizá-las em sua análise.

O principal objeto no contexto das RSSF é o nó sensor. Conforme ilustra a Figura 7, o nó sensor pode ser visto como um objeto que possui uma ou mais informações. A informação do nó sensor pode ser de diversos tipos ou natureza. Como exemplo, o nó sensor pode possuir informações sobre temperatura ambiente, se possuir sensor de temperatura, informação sobre seu

nível de energia, quantidade de falhas ocorridas em um intervalo de tempo, potência do sinal relativo ao último evento recebido, definição do seu estado operacional (se está produzindo dados ou não), entre uma série de outras informações dependendo do tipo de nó e do tipo de aplicação da rede.

Figura 7: Modelo de Domínio dos nós sensores

Em RSSF os nós podem ser representados como um conjunto de objetos, organizados na forma de um mapa, conforme ilustra a Figura 8. Os mapas são *fotos* da rede e possuem como atributo o momento no qual foram obtidos.

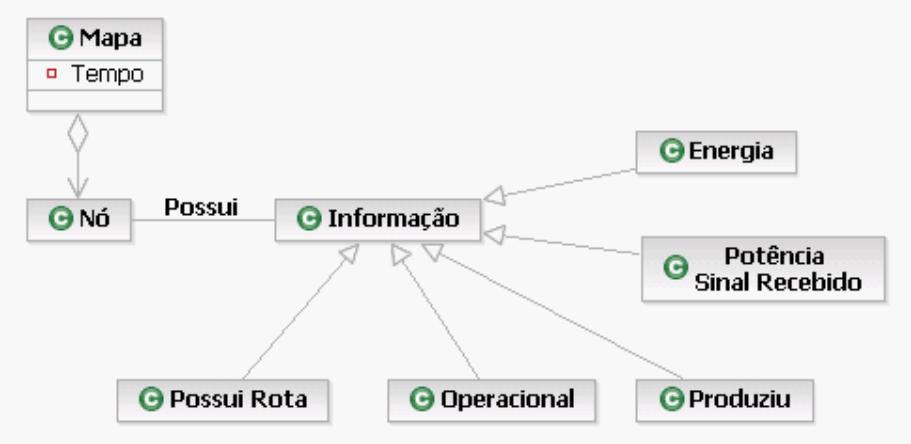


Figura 8: Representação do Mapa como uma agregação de nós.

No contexto da detecção de intruso, diversos mapas podem ser considerados de acordo com a informação que se deseje analisar. Os mapas mostram diferentes visões da mesma rede. Se estivermos interessados na forma como os nós se organizam para rotear informações até a estação base, podemos considerar Mapas de Roteamento. Por outro lado se, para a mesma rede, estivermos interessados em saber se o nó está operacional ou não, podemos considerar o Mapa Operacional da rede. O mesmo raciocínio pode ser aplicado se estivermos interessados em saber quais nós produziram ou não dados que chegaram até a estação base. Ou seja, diversos tipos de visões da

rede podem ser montadas a partir de informações disponíveis nos nós ou de informações que chegam até a estação base.

Se organizarmos esses mapas ao longo de um período de tempo, teremos o comportamento da rede, como ilustra a

Figura 9. O comportamento da rede pode ser modelado como um conjunto de mapas organizados no tempo, sendo esses mapas do mesmo tipo de informação ou não. Por exemplo, podemos ter um comportamento da rede definido através da correlação de mapas de roteamento, estado operacional e dados produzidos.

O *Comportamento da Rede* pode ser de diversos tipos. Podemos considerar o comportamento observado da rede como sendo o comportamento aferido de um a rede ao longo de um período. Ou seja, o "Comportamento Observado" é aquele que a estação base verifica a partir das informações recebidas durante um período *p*.

Um outro tipo de comportamento, seria o comportamento esperado da rede ou comportamento normal. O "Comportamento Esperado" é aquele definido a partir de estudos realizados sobre a rede alvo considerando um subconjunto de informações de interesse. Por exemplo, podemos definir modelos de falhas, de produção, de consumo de energia e de baterias e combiná-los de forma a definir o comportamento normal da rede.

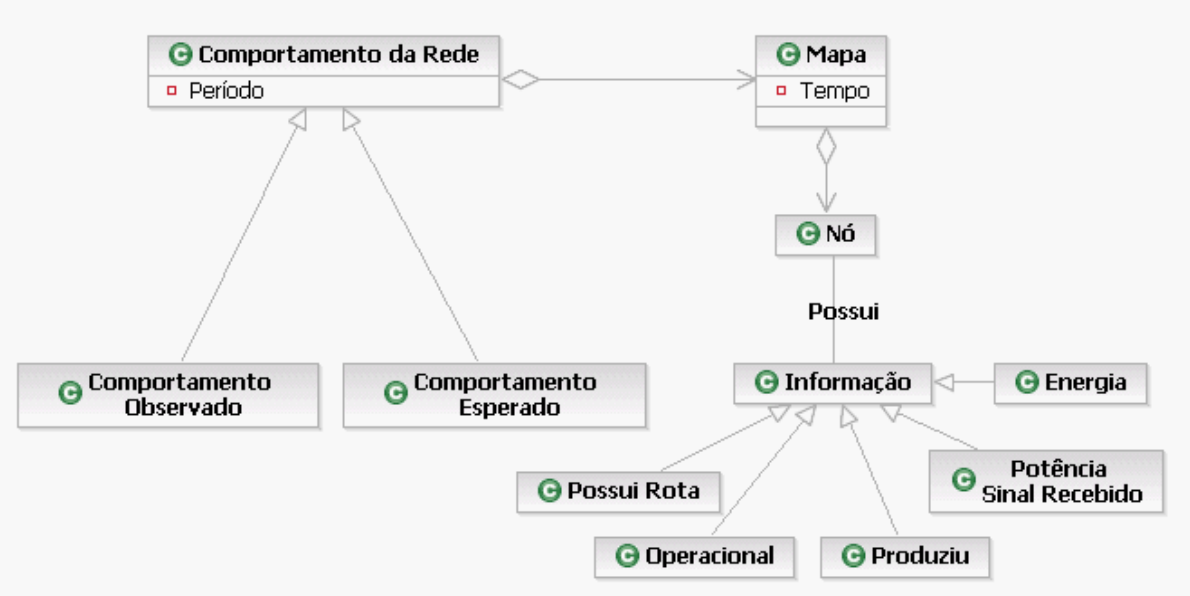


Figura 9: Representação do Comportamento da Rede.

Uma vez definido o modelo de informação, o problema de detecção de intruso passa a ser a construção dos mapas, sua organização para representar o comportamento normal e real e a comparação entre esses comportamentos para definir se há distorções provocadas por intruso.

3.4. Construção dos Mapas

Precisaremos de um mapa correspondente para cada informação necessária definida no subconjunto de informações mínimas, ou seja, precisamos de pelo menos três mapas para a análise de intrusão: Mapa de Produção, Mapa de Estado Operacional e Mapa de Roteamento. O Mapa de Produção indica quais nós produziram e quais deixaram de produzir. O Mapa de Estado Operacional indica se o nó está apto a produzir ou não. E o Mapa de Roteamento indica qual é a rota que um determinado nó utiliza para se comunicar com a estação base. A seguir será apresentada nossa estratégia para construção de cada um desses mapas.

3.4.1. Mapa de Produção

O mapa de produção será montado de forma passiva, ou seja, baseado na observação das informações enviadas pelos nós. A estação base irá observar basicamente três informações: quem enviou o dado, valor do dado sensoriado e a freqüência com que o dado é enviado. Dessa forma, serão mantidos intactos o algoritmo e os protocolos dos nós sensores apesar da implantação do IDS. Não será necessário gerar novas mensagens e não precisaremos confiar nos nós, mas apenas nos fatos que a estação base já é capaz de observar.

3.4.2. Mapa de Roteamento

Assumimos que nas mensagens enviadas por um nó, a identificação da origem e do pai do mesmo está disponível, como acontece no TinyOS Beaconing [5]. Sabemos em um dado momento *t* quais foram as rotas usadas no tempo *t*-1, o que permite a construção do mapa de roteamento.

Sabemos que, dependendo do protocolo de roteamento, essa premissa pode não ser verdadeira, como acontece com o *Directed Diffusion* [84], por exemplo. Nesses casos, uma estratégia semelhante à proposta por Staddon et al.[68] pode ser utilizada para informar quem é o pai do nó. Um *byte* pode ser acrescentado a cada informação enviada para a estação base informando quem é o pai do nó que originou aquela informação. Desta forma é possível acrescentar a informação sem aumentar o número de mensagens, mas teríamos que fazer uma pequena alteração no algoritmo dos nós sensores em prol da viabilização da detecção de intrusos.

É importante observar que o que precisamos é saber se existe rota entre o nó sensor e a estação base. Caso não seja possível incluir informações sobre o pai do nó de origem, outra forma de construção do mapa pode ser estudada de acordo com as especificidades da RSSF alvo.

3.4.3. Mapa de Estado Operacional

O Mapa de Estado Operacional é importante para sabermos se o nó está operacional, produzindo. Permite determinar, por exemplo, se uma determinada informação deixou de chegar à estação base ou porque não foi produzida ou por outro motivo como, por exemplo, devido à ação de um intruso. Sua construção não é trivial, considerando as premissas assumidas. Há diversas possibilidades de se montar esse mapa mas a maioria delas fere a premissa de disponibilizar uma solução não invasiva. A seguir discutiremos algumas possibilidades de se construir esse mapa.

Uma possível solução seria considerarmos uma rede onde o nó informa seu nível de energia a cada mensagem transmitida e utilizarmos modelos de predição, como aqueles propostos em [62]. Baseando-se nas informações disponíveis no tempo *t-1* e em um modelo de gasto de energia, a estação base pode inferir se o nó deverá estar operacional no tempo *t*. O problema com essa solução é que ela não se aplica às RSSF onde os nós não reportam a energia. A não ser que os algoritmos dos nós sensores fossem alterados para enviar informações sobre sua energia. O que tornaria a solução cara e invasiva. Além disso, encontramos dificuldades na obtenção de informações de gasto de energia em experimentos realizados com os mica motes no projeto SensorNet [93] e outros fatores além da energia podem fazer com que o nó pare de produzir, como por exemplo uma falha de *hardware*.

Uma outra possibilidade seria a criação de um protocolo onde a estação base interroga, periodicamente, os nós sensores para saber o estado operacional de cada um. A principal desvantagem dessa solução, em nosso contexto, é que precisaríamos confiar na resposta dos nós, o que vai contra a premissa de que os nós não são confiáveis. Além disso, a solução torna-se cara e intrusiva porque precisaremos alterar o algoritmo dos nós sensores para interrogá-los sobre seu estado operacional, introduzindo novas mensagens na rede. Dependendo da rede alvo, essa estratégia poderá ser usada como um refinamento da solução aqui proposta mas, a princípio, ela não será utilizada.

As duas soluções discutidas anteriormente mostram ser difícil determinar o *Mapa de Estado Operacional* de forma exata, sem alterar os algoritmos dos nós ou sem utilizar mais informações do que consideramos nas premissas do trabalho. Uma alternativa nesse caso é consideramos mapas baseados na probabilidade dos nós sensores estarem operacionais. Ou seja,

construiremos um Mapa de Estado Operacional com um grau de "incerteza" e passarmos a considerar esse grau de incerteza na análise.

O grau de incerteza pode ser representado por probabilidades, ou seja, podemos trabalhar com a probabilidade do nó estar operacional ou não baseado no comportamento esperado dos nós sensores. A probabilidade pode ser definida através de modelos semelhantes aos propostos em artigos de simulação [55, 56] ou emulação [57], modelando-se a bateria, o rádio, o processador e a taxa de erros do canal de comunicação. Há trabalhos relacionados à simulação de rede de sensores que definem modelos de bateria e de gastos de energia, como aqueles propostos para o SensorSIM [56] ou aqueles propostos por Varshney e Bagrodia [55]. Propomos o uso desses modelos para prever, com certo grau de incerteza, se o nó está com energia ou não. Aliando a isso a probabilidade do nó falhar por mau funcionamento usando uma estratégia semelhante a do TOSSIM [57], onde é atribuída uma probabilidade de falhas aos canais de comunicação, poderemos definir qual é a probabilidade do nó estar operacional.

Um desafio desse tipo de solução é como lidar com a incerteza introduzida com os modelos de probabilidade condicional. Entre as técnicas para lidar com a incerteza, a teoria de Bayes [20, 23], em especial a teoria das Redes Bayesinas, merece ser destacada, tanto pela forma simples como organiza as probabilidades condicionais, quanto por sua eficiência na prática. Além disso, as Redes Bayesianas poderão ser utilizadas também para refinar a definição do Mapa Operacional através do uso de Redes Bayesianas Dinâmicas para o desenvolvimento de raciocínios probabilísticos ao longo do tempo. Por exemplo, o cálculo da probabilidade de um nó estar operacional pode ser refinado baseando-se no fato da estação base ter recebido ou não dados do nó no instante *t-1*. Na próxima seção mostraremos como a Teoria de Bayes pode ser utilizada para lidar com a incerteza e como uma Rede Bayesiana pode ser modelada utilizando o nosso modelo de informação com o objetivo de detectar intrusos.

3.5. Estratégia de Análise das Informações

Uma vez definida a estratégia para construção dos mapas, o próximo desafio passa ser definir como esses mapas podem ser combinados de maneira a nos indicar se o comportamento observado difere ou não do comportamento esperado. Uma maneira de se fazer isso é modelar Redes Bayesianas e alimentar essas redes com informações esperadas e evidências.

Por exemplo, se tivermos uma evidência na estação base de que um nó não produziu, mas que ele possui rota para a estação base e a probabilidade dele estar operacional é alta isto indicará, com certo grau de incerteza, que há um intruso. Ou seja, se conseguirmos organizar as evidências

recebidas pela estação base e as probabilidades dos nós estarem operacionais em uma Rede Bayesiana, teremos bons indicadores da existência ou não de intrusos. Como apontaram Russell e Norvig [23]:

> *"O formalismo denominado Redes Bayesianas foi criado para permitir a representação eficiente do conhecimento incerto e o raciocínio rigoroso com a utilização deste tipo de conhecimento. Essa abordagem supera amplamente muitos problemas dos sistemas de raciocínio probabilísticos das décadas de 1960 e 1970; agora ele domina a pesquisa de Inteligência Artificial sobre raciocínio incerto e sistemas especialistas. A abordagem admite o aprendizado a partir da experiência e combina o melhor da Inteligência Artificial clássica e redes neurais."*

A construção de Redes Bayesianas é um método já utilizado em algumas propostas de detecção de intrusos [16, 17, 21, 41]. Vem sendo utilizado com sucesso, também, em filtros *anti-spam* de leitores de *e-mail* de ampla utilização atualmente, como o *Mozilla* [58] e sistemas de diagnóstico e suporte, como nos sistemas operacionais da linha *Windows*. Entre as principais vantagens do uso de Redes Bayesianas podemos destacar que a complexidade computacional é linear, que a técnica lida bem com o princípio da incerteza e evidências, possui uma estrutura e construção simples e permite a construção da rede de maneira incremental, de forma que pode ser facilmente atualizada. A principal desvantagem é que as Redes Bayesianas assumem que as evidências são estatisticamente independentes, ou seja, uma evidência não influencia na geração de outra, o que na maioria dos modelos não é verdade. Teoricamente essa premissa inviabiliza a aplicação das Redes Bayesianas na maioria dos problemas o que, segundo Russel e Norvig [23], atrasou a ampla utilização da teoria de Bayes. Na prática essa premissa vem sendo ignorada e, apesar disso, bons resultados vem sendo obtidos na área de detecção de intrusos [17] dentre outras áreas.

Apesar de optarmos pelo uso de redes bayesianas como método de análise, acreditamos que há outros métodos que também são promissores. Wu et al. [27], por exemplo, propuseram o uso da Teoria da Evidência de Dempster-Shafer [26] como fundamentação matemática para o desenvolvimento de um novo sistema de detecção de DoS. A Teoria da Evidência d e Dempster-Shafer pode ser vista como uma extensão da Teoria de Bayes e trabalha com o princípio da ignorância ao invés do princípio da incerteza. É indicado para casos onde não haja um bom modelo do estado normal da rede. Os autores compararam a estratégia com o método de redes Bayesianas e apontaram as seguintes desvantagens do método de Bayes:

- Necessidade do conhecimento *"a priori"* da distribuição de probabilidade dos estados
- Não provê nenhuma informação sobre a qualidade dos resultados em termo da confiança nas evidências ou em conflitos entre as evidências.

Apesar dessas desvantagens apontadas por Wu et al. [27] optamos por uma solução baseada no método de Bayes por ser um método mais difundido e com bons resultados práticos como foi apontado por Amor et al. [17]. No Apêndice A, apresentamos uma resumo do artigo de Wu et al. com as vantagens e desvantagens do método de Dempster-Shafer (DS) e também de trabalhos relacionados a redes bayesianas.

3.6. Construção de uma Rede Bayesiana para Detectar Intrusos em RSSF

Em uma rede Bayesiana, cada variável do mundo real é representada como uma variável Bayesiana e representa um conjunto de estados. Uma relação causal é definida entre as variáveis e são verificadas quais variáveis influenciam diretamente a variável analisada e quais outras variáveis são influenciadas diretamente por esta variável. Em uma Rede Bayesiana padrão, cada variável é representada graficamente por uma elipse colorida chamada de nó e cada influência causal é descrita por uma linha (arco) conectando a variável de influência. O arco é direcionado para a variável influenciada, formando dessa forma um grafo direcionado, como mostra a Figura 10. A terminologia padrão é que um nó é uma variável Bayesiana e um arco conecta um nó pai, variável que influencia, a um nó filho, variável influenciada.

Para se criar uma Rede Bayesiana é necessário definir um conjunto de variáveis que representem os elementos distintos da situação a ser modelada. Para nosso caso teríamos as seguintes variáveis:

- Produção
- Rota para estação base
- Operacional
- Intruso

Define-se, para cada variável, o conjunto de estados possíveis. O conjunto deve ser mutuamente exclusivo e exaustivo, de forma que cubra todas as possibilidades para a variável e que haja distinções importantes entre cada estado. No nosso caso teremos os seguintes estados:

- Produção: Produziu ou não Produziu.
- Rota para estação base: Possui rota ou não possui rota.
- Operacional: Nó está apto a produzir ou não.
- Intruso: Há intruso ou não há intruso.

A parti daí, se estabelece uma relação causal entre as variáveis, o que envolve a criação de arcos direcionados conectando os nós pais aos nós filhos. Finalmente, define-se as prioridades *a priori* para cada variável, o que significa definir probabilidades numéricas para cada variável. Em nossa modelagem, consideramos a visão de cada nó e utilizamos um modelo onde as falhas estão embutidas nos valores das probabilidades para as quais, a princípio, definimos valores arbitrários. A idéia é começar com esses valores, depois expandir o modelo para *n* nós, utilizando composição de probabilidades e, finalmente, calibrar os valores das probabilidades com dados reais, de acordo com as particularidades da RSSF alvo[7].

A rede bayesiana aqui modelada é composta por quatro variáveis: "*Produção"*, "*Rota"*, "*Operacional"* e "*Intruso"*, como mostra a Figura 10. Ou seja, há uma variável para cada mapa definido no escopo do trabalho. Há um arco ligando "*Operacional"* e "*Produção",* já que a produção depende do nó estar operacional. Há um arco ligando "*Produção"* e "*Rota"* , pois a existência da rota influencia a produção, de maneira que se não houver rota entre o nó e a estação base, a produção não chegará ao observador. A existência do intruso, por sua vez, influencia a existência da rota e o estado do nó que, dependendo do ataque, pode deixar de ficar operacional. Dessa forma, indiretamente, o intruso influencia a produção do nó, seja pela influência na rota, seja pela influência no estado operacional do nó.

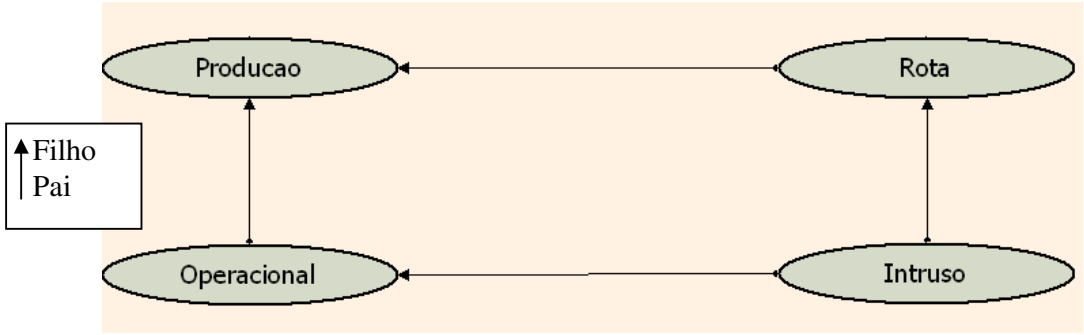


Figura 10: Rede Bayesiana para Detecção de Intrusos em RSSF

Como dissemos, para cada nó da Rede Bayesiana precisamos definir qual é probabilidade *a priori*, ou seja, a probabilidade do evento se manifestar considerando que não temos evidências. Além disso é necessário definir qual é a probabilidade condicional, ou seja, qual a probabilidade de um evento em razão de uma outra variável ou uma evidência.

[7] Um alternativa interessante, mas que foge do escopo deste trabalho, é estabelecer uma fase de treinamento onde o Administrador da Rede, através de um sistema de apoio, verifica os resultados gerados pelo IDS e calibra as probabilidades, usando uma técnica semelhante à proposta em [39].

A título de ilustração, utilizamos um software de modelagem de Redes Bayesianas chamado MSBNx [24] para definir as prioridades *a priori* e definir as evidências, utilizando a Rede Bayesiana utilizada na Figura 10. As probabilidades foram definidas de maneira empírica[8]. A primeira probabilidade definida foi a da existência do intruso que foi definida como 50% ou 0,5 em uma escala 0 a 1, como mostra a
Figura 11. Uma vez definida a probabilidade *a priori* da existência de intruso, o próximo passo é definição da probabilidade condicional da existência do nó estar operacional, ou seja, qual a probabilidade de "Operacional" considerando a existência ou não de um intruso? No exemplo apresentado na Figura 12, considerou-se que a probabilidade do nó estar operacional dado a existência de um intruso é 0,2 e a probabilidade do nó estar operacional na ausência do intruso é 0,8 uma vez que diversos tipos de falhas podem levar o nó a não estar operacional.

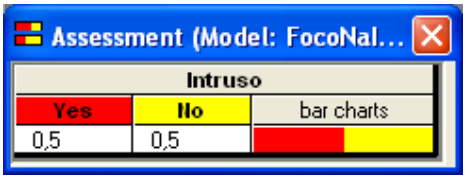


Figura 11: Probabilidade da existência de Intruso definida no MSBNx

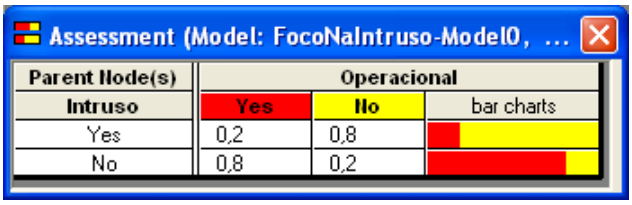


Figura 12: Probabilidade condicional de o nó estar Operacional

Em relação à rota, a pergunta é: Qual é a probabilidade de haver rota entre o nó e a estação base se existir um intruso? Ou seja, qual a probabilidade do intruso prejudicar a rota que liga o nó com a estação base? Nesse caso nos interessa saber qual a probabilidade do intruso ter prejudicado um dos nós que compõem a rota entre o nó avaliado e a estação base. Para esse exemplo consideramos que na ausência de um intruso a probabilidade de haver uma rota entre o observado e a estação base é 0,8. Com a presença do intruso a probabilidade cai para 0,5, conforme ilustra a Figura 13.

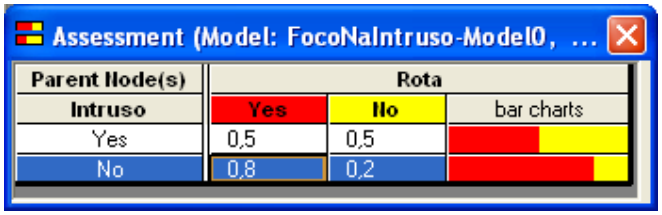


Figura 13: Probabilidade de existir rota considerando a presença/ausência de intruso

[8] Para a aplicação real deve ser feita uma estimativa inicial usando o conhecimento de um especialista, como foi feito aqui, e , em seguida, os valores devem ser calibrados com medições sobre a rede real.

Para essa rede bayesiana, a probabilidade da existência de um intruso pode ser definida considerando evidências coletadas na rede. Por exemplo, se conseguirmos verificar que existe rota disponível e que o nó não produziu, mesmo não conhecendo o estado operacional do nó, teremos uma probabilidade de 0,7143 de haver um intruso, como mostra a Figura 14. Ou seja, se calibramos as probabilidades de cada variável e coletarmos Mapas de Produção e de Rota poderemos inferir se há ou não intrusos usando o conceito de redes bayesianas.

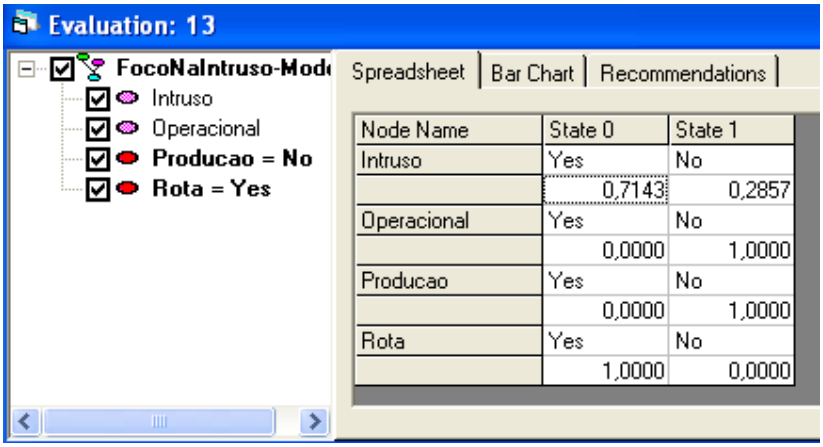


Figura 14: Probabilidade de haver intruso dado evidências que existe rota e o nó não produziu.

3.7. Classificação do IDS proposto

De acordo com a taxonomia proposta no Capítulo 1, quanto à classificação geral, o IDS proposto neste trabalho não é implantado na estação base, não é invasivo (não altera o comportamento da RSSF) e não é reativo, ou seja, quando detecta uma intrusão emite alertas mas não reage ao intruso.

Quanto à obtenção de dados o IDS proposto é capaz de lidar com diferentes fontes de dados como arquivos de *log*, pacotes de rede entre outras. A coleta de dados é feita na estação base de forma centralizada e passiva, ou seja, não é feita interrogação sobre os nós mas sim uma análise sobre os dados que chegam até a estação base.

Quanto ao método de análise e conclusão sobre intrusão, o IDS é baseado na especificação das características da rede por um especialista. A detecção pode ser feita em tempo real e o processamento das informações é feito periodicamente.

3.8. Conclusão

Apresentamos conceitualmente, a estratégia de solução para detecção de intrusos em RSSF de forma não invasiva. A solução apresentada possui três pontos principais: definição de um modelo de informação que represente as especificidades das RSSF; estratégias para construção dos mapas que são as estruturas nas quais as informações são organizadas; utilização de um método de análise de intrusão que consiga tratar com o princípio da incerteza de forma lógica e consistente. Para tanto propomos a utilização de Redes Bayesianas e apresentamos a modelagem de uma Rede Bayesiana no contexto do trabalho. Além disso, classificamos nossa estratégia de solução de acordo com a taxonomia apresentada no Capítulo 1.

No próximo capítulo faremos o desenho da solução, mostrando como essas idéias podem ser implementadas segundo uma arquitetura orientada a objetos e preparando a solução para ser implementada em linguagens disponíveis atualmente.

Capítulo 4

Arquitetura do Sistema de Detecção de Intrusos

4.1. Introdução

No capítulo anterior apresentamos a estratégia para detecção de intrusão obedecendo ao escopo definido no Capítulo 2. Neste capítulo, apresentaremos a arquitetura de um sistema de detecção que viabiliza a execução da estratégia proposta no capítulo anterior e mostraremos como os mapas podem ser construídos e como a análise de detecção de intrusão pode ser realizada.

Do ponto de vista de arquitetura, a solução pode ser dividida em quatro partes principais: fonte de dados, construção dos mapas, base de conhecimento e análise de intrusão. Propomos a criação de uma base arquitetural com essas quatro partes principais formando um esqueleto do IDS, conforme ilustra a Figura 15. A base arquitetural é formada por:

- Fonte de Dados
- Base de Conhecimento
- Estratégia de Análise
- Mapas de Informação

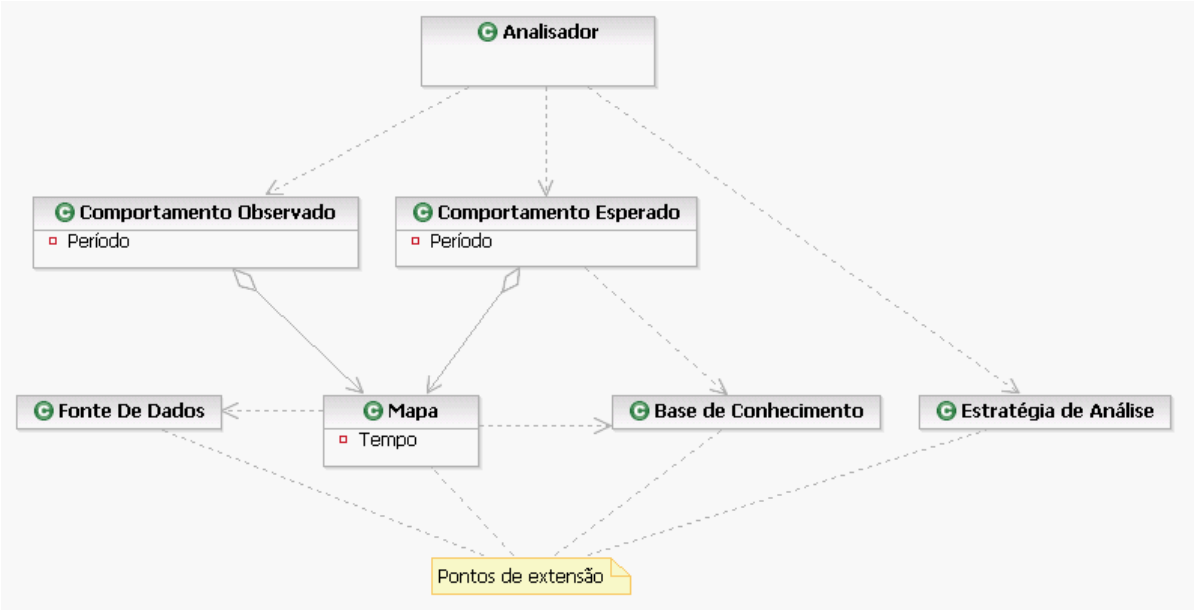


Figura 15: Visão Lógica da Arquitetura do IDS[9].

A construção do IDS é realizada através da especialização dos pontos básicos da arquitetura.

Os mapas são construídos baseados no padrão de projeto *Prototype* [96], a arquitetura prevê a disponibilização de um *protótipo* e os mapas específicos, como de produção, de estado operacional e de rotas são especializações desse protótipo.

A fonte de dados é definida conforme a informação disponibilizada na estação base. Por exemplo, a fonte de dados pode ser baseada em arquivos de *log* ou pacotes recebidos através de uma interface serial onde o nó sorvedouro é conectado. Criamos o conceito de fonte de dados, seguindo o padrão de projeto *DAO* (*Data Access Object)* [98], para tornar o IDS independente da forma como os dados são gerados e isolar o sistema das diferentes fontes de dados que possam ser utilizadas em diferentes estações base.

Desejamos que o sistema seja um observador das informações geradas pela rede. Para tanto, seguimos o padrão *Observer* [96], ou seja, o sistema se registra como um observador e, a cada evento gerado, a classe responsável pela fonte de dados analisa as informações necessárias para construção dos mapas.

O sistema utiliza uma abstração que representa a base de conhecimento. A base de conhecimento abrigará o conjunto de conhecimentos que define o comportamento normal da rede, considerando os mapas escolhidos. O conjunto de conhecimento poderá ser formado por axiomas,

asserções e modelos de predição tais como: modelo de consumo de energia, modelo de bateria, modelo de rádio, modelo de sensoriamento, modelo de roteamento. A abstração prevista na arquitetura é concretizada através da base de conhecimento real, ou seja, a abstração representa um protótipo da base de conhecimento que é implementado pela versão real, seguindo o padrão de projeto *Prototype* [96].

A estratégia de análise é outro pilar previsto na arquitetura. A arquitetura proposta prevê que essa estratégia poderá ser revista de acordo com a rede alvo e com as informações disponíveis. Para facilitar a troca ou atualização da estratégia, foi utilizado o padrão de projeto *Strategy* [96], que prevê o encapsulamento de algoritmos possibilitando que os mesmos sejam trocados sem afetar os clientes que desejem usá-los.

Com isso, as alterações e evoluções do IDS serão facilitadas porque novos mapas, estratégias ou bases de conhecimentos podem ser adicionados ao IDS, pela substituição ou inclusão de instâncias de cada uma dessas partes. A Figura 16 ilustra um exemplo de uma implementação usando a arquitetura proposta[10]. Nesse exemplo, a fonte de dados baseia-se na classe *SensorAnalyser* conforme utilizado no *Surge*, aplicação de exemplo do *TinyOS*, ou em um arquivo de log gerado na estação Base. Os mapas instanciados foram os de produção, operacional e rotas, além do mapa de intrusão. A base de conhecimento utiliza asserções, axiomas e redes bayesianas. Finalmente, é ilustrado que diferentes estratégias de detecção podem ser utilizadas como a montagem de um mapa de intrusão, estratégias baseadas em composição de probabilidades ou votação [23].

Por exemplo, no escopo deste trabalho optamos por não utilizar mapa de energia. Se desejarmos gerar uma nova versão com o mapa de energia, bastará adicionar uma nova instância de um mapa, uma nova instância da base de conhecimento baseados nos protótipos já existentes e atualizar o algoritmo de estratégia de detecção para que passe a considerar as novas informações relacionadas à energia e o novo conhecimento acrescentado na Base de Conhecimento.

[9] A notação utilizada segue o padrão UML, onde os retângulos representam classes, a seta tracejada representa uma relação de dependência e as linhas com um losango representam agregações.

[10] Uma parte do que é exposto na Figura 16 foi implementado como um protótipo do IDS e será apresentado no próximo capítulo.

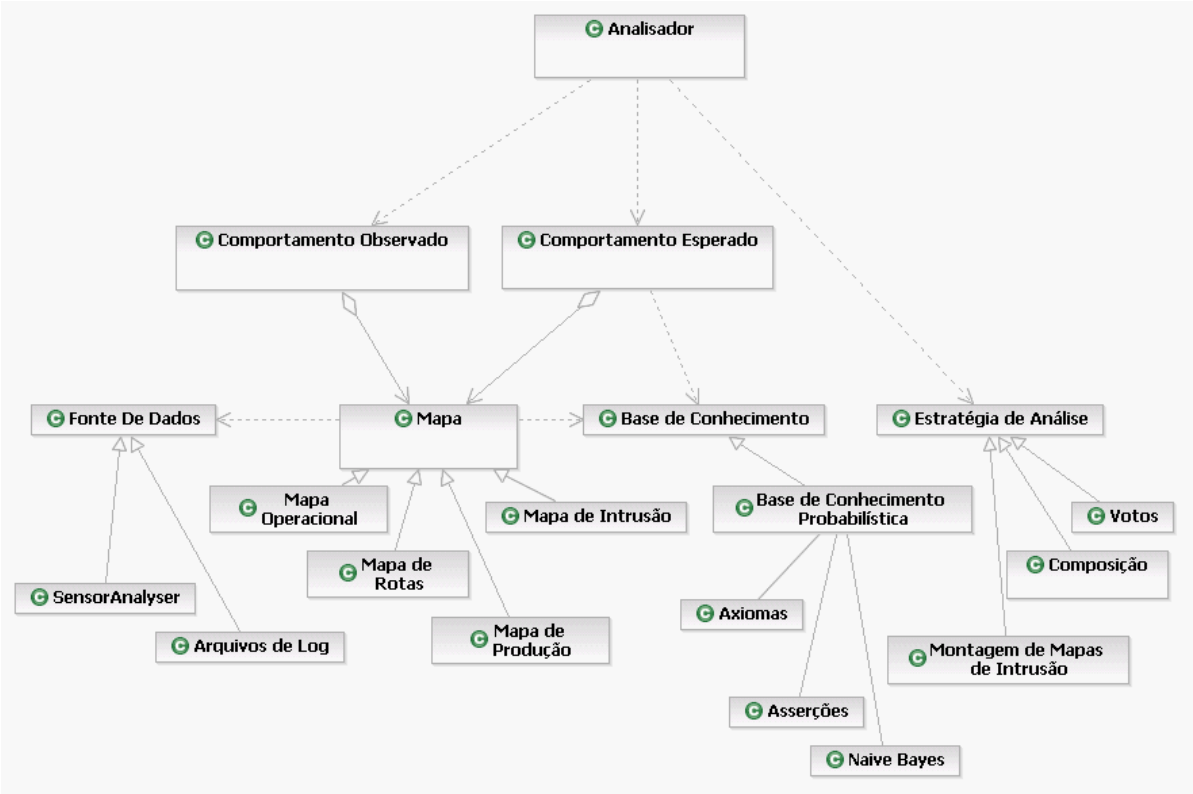


Figura 16: Exemplo de extensões seguindo a arquitetura de referência[11].

4.2. Fonte de Dados

Toda informação sobre a rede que chega até a estação base é candidata a ser fonte de dados. Entretanto, é desejável que o sistema se adapte a diferentes fontes de dados. Para isso é importante isolar o sistema de detecção das especificidades das fontes de dados. Para atingir esse objetivo, a arquitetura prevê uma camada de acesso a dados que é implementada de acordo com a fonte de dados alvo.

Por exemplo, nas aplicações disponibilizadas junto ao TinyOS, como o Surge, uma série de informações alcançam a estação base através da interface serial. A interface serial passa a ser uma candidata à fonte de dados. Se o projetista do IDS preferir, a fonte de dados poderá ser considerada em nível de abstração maior. Seguindo o exemplo do TinyOS e, mais especificamente, no caso do Surge, a fonte de dados pode ser uma instância da classe SensorAnalyser, que já é preparada para estruturar as informações que chegam via interface serial. Independente da forma como a aplicação trata ou recebe as informações, com a arquitetura proposta, essas informações podem ser repassadas para o sistema de detecção sem a necessidade de tornar o IDS restrito a um tipo

[11] As extensões são representadas por linhas cheias com um triângulo na ponta conforme padrão UML.

específico de fonte de dados. Para atingir esse objetivo, basta que uma instância da fonte de dados seja implementada isolando o resto do sistema da forma como a informação chega até a estação base.

4.3. Construção de Mapas

O processo de construção de mapas consistirá em analisar os dados brutos da rede que chegam à estação base e são repassados para o sistema através da fonte de dados. Os dados brutos podem ser informações de medidas enviadas pelos sensores, cabeçalhos da mensagem e até mesmo o não recebimento de mensagens esperadas. A partir dessas informações, o processo de construção de mapas é realizado analisando os dados de acordo com perspectiva de cada mapa.

O IDS monitora o recebimento de dados de maneira promíscua e espera o momento de atualizar os mapas. A cada período de tempo o IDS desencadeia a atualização de cada tipo de mapa baseando-se nas últimas informações recebidas. O resultado deste processo é a construção de um conjunto de mapas correspondente àquele período de tempo, que pode ser somado aos outros para compor o comportamento observado ao longo do tempo.

A construção dos mapas foi projetada segundo o padrão de projeto *Visitor* [96], onde cada mapa é uma classe visitante que analisa o conjunto de dados contidos nos nós segundo sua perspectiva de interesse. Com isso, a inclusão de novos mapas é facilitada, uma vez que basta criar um novo visitante. Além disso, conseguimos economizar espaço em memória, uma vez que evitamos , em uma mesma época, replicar nós sem necessidade[12].

A classe que constrói o Mapa de Produção baseia-se exclusivamente no fato do nó ter produzido ou não no instante da análise, ou seja, no tempo *t* onde a classe visitante está construindo o mapa. O Mapa de Roteamento é atualizado utilizando-se as informações contidas nos cabeçalhos das mensagens, onde há o identificador do nó origem e de seu "pai", ou seja, quem é o nó que serve como roteador para quem está originando a mensagem. A partir dessas informações é possível construir a árvore de roteamento. O Mapa Operacional é atualizado cruzando informações do Mapa de Produção, da Base de Conhecimento e do Mapa de Roteamento. Na Base de Conhecimento, como veremos adiante, há axiomas que, a partir de evidências se o nó produziu ou não e se ele possui rota para a estação base, permitem determinar a probabilidade de o nó estar operacional. Por exemplo, se o nó não produziu e existe uma rota entre ele e estação base, a probabilidade desse nó não estar operacional é alta. Como veremos a seguir,

[12] Uma alternativa seria representar o nó várias vezes. Cada representação do nó possuiria apenas as propriedades de interesse, mas o nós seriam replicados para cada tipo de mapa.

muitas informações como essas podem ser organizadas na forma de uma base de conhecimento, utilizada principalmente para construir o mapa operacional e analisar a existência ou não do intruso.

4.4. Base de Conhecimento

A base de conhecimento possui o conjunto de conhecimento que define o comportamento normal da rede. O conjunto de conhecimento é formado por axiomas, asserções e modelos de predição, tais como modelo de consumo de energia, modelo de bateria, modelo de rádio, modelo de sensoriamento e modelo de roteamento.

A Base de Conhecimento é composta por regras definidas por axiomas probabilísticos. O objetivo da Base de Conhecimento é organizar de forma sistemática o conhecimento prévio do especialista na rede e armazenar de forma organizada o conhecimento adquirido com informações fornecidas pela própria rede. A Tabela 1 apresenta uma lista com exemplos de regras utilizadas na validação do trabalho[13], considerando o estado normal da rede, ou seja, o que deve acontecer se não houver intruso ou se o intruso não interferir na propriedade analisada. É importante observar que o estado é definido seguindo a perspectiva da estação base. A lista não é exaustiva e pode ser configurada de acordo com as especificidades de cada RSSF.

Tabela 1: Exemplos de regras da Base de Conhecimento

Número	Regra	Explicação
1	P(Operacional) = 1 - P(Falha no Sensor) =~ 0,8	Nessa regra a probabilidade do nó estar operacional é definida como 0,8. Uma probabilidade *a priori* é definida, mas poderá ser revista de acordo com a RSSF alvo.
2	P(Produzir | Não Operacional) = 0	O nó pode ficar não operacional por diversos motivos: não tem energia, não pode transmitir informação, o sensor está com defeito, etc. Essa regra define que independentemente do motivo exato, o fato de não estar operacional implica que o nó não será capaz de produzir informações.
3	P(Produzir | Operacional) = 1 - P(Falha na Rota)	Se o nó estiver operacional, irá produzir[14]. A não ser que haja problema na rota entre o nó produtor e a estação base.
4	P(Operacional | Recebeu Produção) = 1	Em situação normal, se a estação base recebeu a produção do nó significa que o nó está operacional.
5	P(Falha na Rota| Recebeu Produção)	Em situação normal, se a estação base recebeu a

[13] Conforme será detalhado no Capítulo 5

[14] É importante observar que em nosso contexto, produzir significa que o nó produziu e a produção chegou até a estação base.

	= 0	produção do nó não há falha na rota que liga o nó à estação Base.
6	P(Falha no Nó Origem | Recebeu Produção) = 0	Se a produção do nó pode ser observada pela estação base então não há falha no nó de origem.
7	P(Falha na Rota) = P(Falha no Nó j) ∨ P(Falha no Nó j-1) ∨ ... ∨ P(Falha no Nó 1), onde J representa a posição do nó na rota	Se houver falha em um dos nós da rota que interliga o nó produtor à estação base, toda a rota estará comprometida.
8	P(Receber Produção) = P(Rota) ∧ P(Operacional)	Se um nó não produziu, o nó não está operacional ou não há rota entre o nó e a estação base, ou há falha na rota e no nó.
9	P(Falha na Rota | Nós da rota produziram) =~ 0	Por definição, se o dado chega à estação base, todos os nós que compõem a rota estão operacionais. Uma exceção a essa regra acontece quando há nós com sensores defeituosos e rádio funcionando. Mas, a princípio, não iremos considerar essa exceção
10	P(Operacional | Nó está roteando) =~ 1	Se o nó é capaz de rotear ele é capaz de produzir. Como dito na regra anterior, há exceções mas não a consideraremos a princípio.
11	P(Rota | Pai tem rota ∧ Operacional) = 1	Se o nó está operacional e há rota entre o pai dele e a estação Base, então há rota entre o nó e a estação Base.

4.5. Estratégia de Análise

A análise de intrusão é feita utilizando os diversos mapas construídos ao longo do tempo. Suas informações são relacionadas e confrontadas com a base de conhecimento. O Analisador recebe os Mapas construídos e usa a Base de Conhecimento para calcular a probabilidade de intrusão utilizando a informação de cada nó. O resultado desse processo é um mapa de probabilidade de intrusão que representa a probabilidade de haver um intruso na rede considerando a ótica de cada nó sensor. A estratégia de análise é encapsulada de forma que possa ser trocada por outra caso haja necessidade, seguindo o padrão de projeto *Strategy* [96].

Um problema relacionado à análise da presença de intruso, considerando o princípio da incerteza e a ação do IDS ao longo do tempo, é que o número de mapas considerados na análise não pode ser infinito. Restrições de recursos computacionais e limite de tempo para recebimento de respostas tornam necessário limitar o número de mapas utilizados na análise.

Neste trabalho, o número de mapas foi limitado a um, correspondente à versão gerada mais recentemente. Em outras palavras, os dados gerados no período *p* são armazenados no mapa de produção *t*, onde *t* é o instante no qual o mapa é analisado e *p* é período entre a análise atual e a última análise realizada. O nó sempre possui o último dado sensoriado. Os dados sensoriados anteriormente ou pelo menos a informação de se o nó produziu ou não precisam ser armazenadas em um histórico de mapas. Por exemplo, o *mapa (t-1)* possui as temperaturas coletadas no período *p-1*. Desta forma precisamos de uma estrutura, que chamaremos de comportamento, que é um vetor de mapas indexados pelo tempo.

Consideramos o tempo discreto, de forma que mapas ocorridos no intervalo entre *t* e *t-1* serão ignorados na análise. Ao final de cada período *p* é tirada uma foto da rede e esse instante é representado por *t*. Ou seja, uma visão dos nós, com as informações pertinentes, será montada e armazenada formando, assim, o mapa(*t*). O comportamento da rede será o conjunto de mapas em um intervalo de análise *i*, ou seja, todos os mapas gerados nos últimos *x* minutos por exemplo. A limitação a um intervalo é necessária para garantir um conjunto finito de mapas.

O raciocínio apresentado se baseia na Hipótese de Markov [23,28] que afirma que o estado atual depende apenas de um histórico finito de estados anteriores. Os processos que satisfazem essa hipótese foram estudados em profundidade pelo estatístico russo Andrei Markov, e são chamados processos de Markov ou cadeias de Markov. Tais processos existem em muitas variedades, sendo o mais simples chamado de processo de Markov de primeira ordem, em que o estado atual depende apenas do estado anterior e não de quaisquer estados mais antigos. Em nosso caso, iremos supor que os mapas estudados constituem um processo de Markov de primeira ordem, ou seja, não necessitaremos de todos os mapas para concluirmos sobre a existência ou não de intrusos, utilizaremos apenas o último mapa coletado. Futuramente o modelo poderá ser expandido para utilizar Redes Bayesianas Dinâmicas, ou seja, variáveis coletadas no instante *t-1* passariam ser cruzadas com variáveis coletadas no tempo *t* com o objetivo de traçar conclusões sobre a presença de intruso.

4.6. Conclusão

Apresentamos, neste capítulo, a arquitetura de implementação da estratégia de detecção do Capítulo 3. Mostramos que a construção dos mapas será feita a partir das informações que chegam à estação base, que a base de conhecimento será composta por axiomas probabilísticos e que a análise de intrusão será feita utilizando os diversos mapas construídos ao longo do tempo. Consideramos que a construção dos mapas se encaixa em um processo de Markov de primeira

ordem, ou seja, utilizaremos em cada análise apenas o último mapa gerado e não um histórico de mapas.

No próximo capítulo mostraremos como a solução foi validada, através da implementação de um protótipo da solução, de testes feitos com simulação da rede e da análise de falsos negativos e positivos na detecção de ataques.

Capítulo 5

Avaliação da Solução

Para validar a solução proposta e mostrar a viabilidade de implementação, construímos um protótipo do IDS seguindo a arquitetura apresentada no capítulo anterior. Além do protótipo, simulamos uma RSSF e diversos cenários de intrusão e construímos um módulo para analisar falsos positivos e falsos negativos gerados pelo protótipo do IDS.

5.1. Protótipo do Sistema de Detecção de Intrusão

No protótipo do IDS, construído em Java, montamos os mapas, construímos a base de conhecimento e calculamos a probabilidade de intrusão, seguindo a arquitetura proposta no capítulo anterior. O mapa de produção foi construído observando quais nós geraram dados e quais ficaram silenciosos em cada período. O mapa de roteamento foi construído baseado nos identificadores de origem dos pacotes e dos pais dos nós correspondentes. Para o mapa operacional, definimos uma probabilidade inicial de o nó estar operacional. A princípio, a probabilidade foi fixada em 80% uma vez que o número de falhas em RSSF tende a ser alto visto que pode haver falhas de sensoriamento, transmissão, parada de funcionamento do nó sensor por falta de energia ou defeito no hardware entre outras. Mas, baseado em regras adicionadas à base de conhecimento e em evidências recebidas da rede, essa probabilidade é atualizada a cada período de tempo. Conforme apresentado no capítulo anterior, a solução prevê a possibilidade de serem acoplados modelos de simulação do comportamento normal da rede, o que permitirá calcular a probabilidade do nó estar operacional com maior confiabilidade.

Implementamos no protótipo uma base de conhecimento com os axiomas apresentados na seção 4.4 e um analisador de anomalias. O analisador recebe os mapas como entrada, cruza a

informação de cada nó com as regras especificadas na base de conhecimento e calcula a probabilidade de haver intruso, gerando como saída um mapa de intrusão.

5.2. Simulação de Intrusões em uma RSSF

Para avaliar quantitativamente a solução, simulamos uma rede e ataques sobre essa rede utilizando o simulador construído em [34]. É um simulador simples, mas bastante eficiente, que possibilita a realização de experimentos sem consumir muitos recursos computacionais. Os nós simulados geram dados continuamente e os ataques propostos em [5]. Alteramos o simulador incluindo nas mensagens o identificador nó de origem e do seu pai. Alteramos, também, as mensagens de saída do simulador para facilitar o processo de análise de falsos positivos e falsos negativos. Outros simuladores ou técnicas de simulação [55, 56, 57, 58, 59, 60] foram analisados[15], mas o simulador proposto em [34] nos foi mais adequado devido a sua simplicidade, por já simular a rede e os ataques e por já ter sido projetado com o objetivo de ser utilizado na pesquisa de técnicas para detecção de intrusos em RSSF.

5.3. Taxa de Detecção e Alarmes Falsos

A eficácia na detecção foi analisada através da taxa de detecção e a quantidade de alarmes falsos gerados. Se houver um ataque em um intervalo de tempo, verifica-se se o ataque foi detectado corretamente, em caso positivo contabiliza-se um sucesso, em caso negativo, um fracasso (falso negativo). A taxa de detecção é determinada pela razão entre a quantidade de falsos negativos e o total de ataques realizados na simulação. Caso não haja intrusão mas sim detecção contabiliza-se um alarme falso.

Construímos, em Java, um programa para calcular a taxa de detecção e contabilizar os alarmes falsos gerados pelo protótipo do IDS. Ele sumariza os resultados de cada experimento calculando a média e o desvio padrão. Esse programa foi chamado de Analisador.

O Analisador recebe os dados de saída do IDS e realiza um cruzamento de informações para verificar quantas detecções incorretas (falsos negativos) e quantos alarmes falsos (falsos positivos) foram gerados. Se a saída do simulador mostra que houve um ataque e o IDS não gerou alarme, o Analisador contabiliza um falso negativo. Da mesma forma, se a saída do simulador mostra que não houve ataque e o IDS gera um alarme, o Analisador contabiliza um falso positivo.

5.4. Experimentos

Verificamos a eficácia do IDS em uma rede de sensores sem fio. Utilizamos a mesma rede proposta em [34] e mostrada na Figura 17. Trata-se de uma rede plana e fixa com 100 nós sensores distribuídos aleatoriamente em um *grid* de 20x20 metros, onde mensagens de dados são enviadas a intervalos regulares de 40 em 40 iterações. Os nós são identificados unicamente e possuem alcance de rádio fixo. Para simplificar os experimentos, utilizamos sempre a mesma árvore de roteamento. O roteamento é feito de maneira *multihop* seguindo a árvore de roteamento gerada a partir do algoritmo distribuído de Propagação de Informação [97].

Foram utilizados três tipos de nós: o nó comum, a estação base e o intruso. O nó comum possui a função de sensor e roteador, ou seja, faz o sensoriamento do ambiente, envia os dados sensoriados para a estação base e repassa as mensagens recebidas de um vizinho em direção à estação base. A estação base é responsável por receber as mensagens dos nós e por gerar a saída que será analisada pelo IDS. O intruso[16] é nó responsável por introduzir os ataques na rede de acordo com os parâmetros de configuração repassados ao simulador.

[15] Chegamos a simular no TOSSIM [57] também usando a aplicação Surge e, mais especificamente, o SensorAnalyser como fonte de dados. Mas a simulação com o simulador [34] foi mais eficaz para os nossos objetivos visto que com o TOSSIM o experimento travava com a rede de exemplo utilizada.

[16] Optamos pela inclusão de apenas um intruso pois este cenário corresponde ao mais complexo de ser detectado.

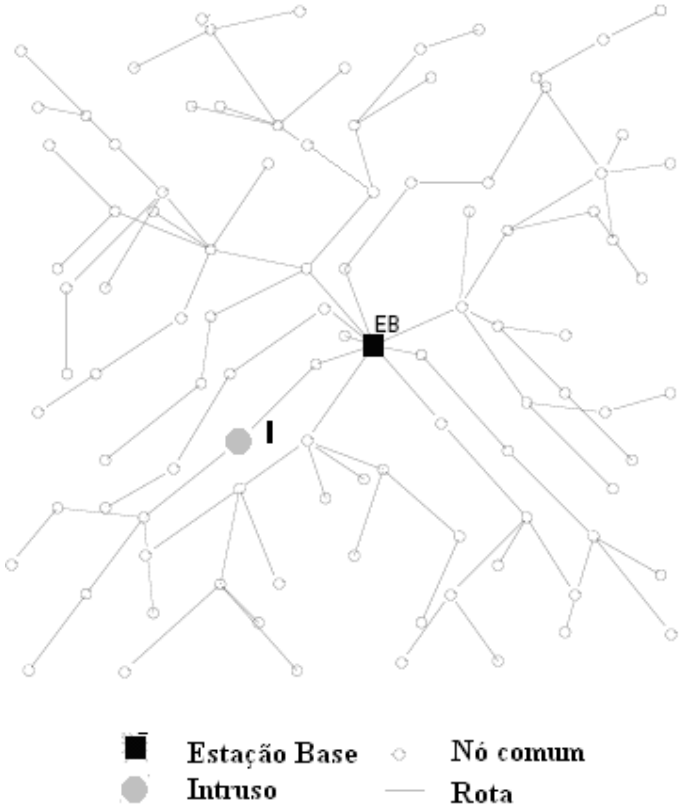


Figura 17: Árvore de roteamento utilizada na simulação.

A eficácia na detecção foi analisada através da taxa de detecção e a quantidade de alarmes falsos gerados pelo IDS. Se houver um ataque em um intervalo de tempo, verifica-se se o ataque foi detectado corretamente, em caso positivo contabiliza-se um sucesso, em caso negativo, um fracasso (falso negativo). A taxa de detecção é determinada pela razão entre a quantidade de falsos negativos e o total de ataques realizados na simulação. Caso não haja intrusão, mas sim detecção, contabiliza-se um alarme falso (falso positivo).

Os experimentos foram repetidos pelo menos 35 vezes cada um e foram consideradas as médias dos valores e o desvio padrão. As simulações foram realizadas considerando um tempo virtual de 4000 iterações, variando-se a taxa de ocorrência de ataque entre zero e cem por cento, com intervalos de cinco por cento. A taxa de ocorrência de ataque indica em qual freqüência o intruso realiza o ataque. Uma taxa de 40%, por exemplo, indica que o ataque do intruso é simulado em 40% das iterações.

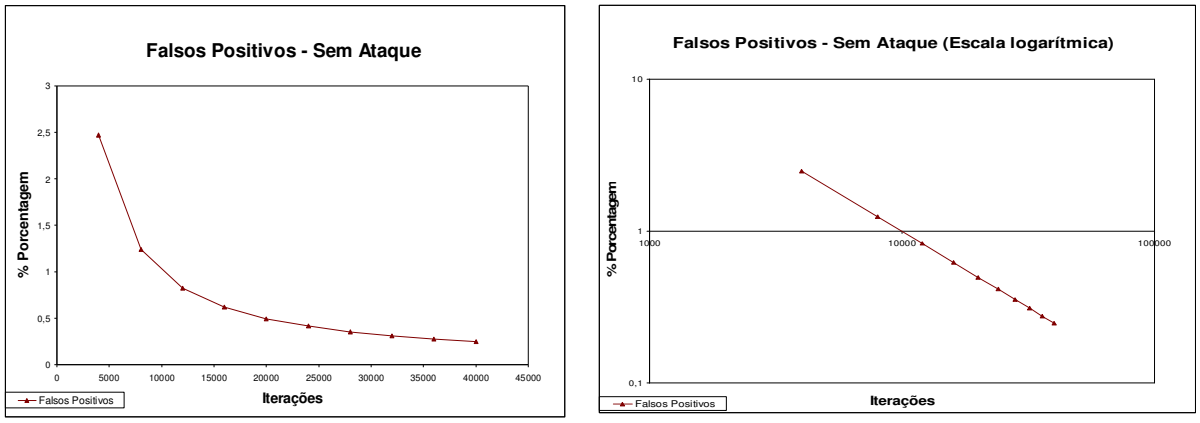
Os experimentos foram realizados simulando os seguintes ataques: Blackhole, Selective Forward[17], Negligência, Wormhole e Jamming. Fizemos, também, experimento simulando a rede sem intrusos, onde analisamos a quantidade de alarmes falsos na ausência de ataque. Variamos a quantidade de iterações entre 5000 e 45000 iterações, com intervalos de 5000. Neste caso esperamos não ter falsos positivos[18].

A seguir apresentaremos os resultados obtidos na tentativa de detectar cada um dos ataques e também no caso de não haver intruso.

5.4.1. Rede sem intrusos

Neste experimento avaliamos o comportamento do IDS na ausência de intrusos. O objetivo foi avaliar a quantidade de alarmes falsos gerados pelo IDS. Os testes foram executados variando-se a quantidade de iterações e medindo-se a quantidade de alarmes falsos apontados pelo IDS.

O gráfico da Figura 18 (a) mostra que a porcentagem de falsos positivos na ausência de intruso ficou abaixo de 2,5%, sendo reduzida à medida que o número de iterações é aumentado. Na verdade podemos observar através do gráfico da Figura 18(b), em escala logarítmica, que a porcentagem de falsos positivos cai de forma constante de acordo com o aumento de iterações. Isso acontece porque o número de falsos positivos foi praticamente constante e como a porcentagem é calculada considerando o total de iterações, à medida que aumentamos a amostra, ou seja, aumentamos as iterações, a porcentagem cai. O ideal neste modelo seria que nenhuma indicação de intruso fosse levantada, mas consideramos que uma porcentagem de 2,5% é um resultado bastante satisfatório se compararmos com resultados típicos [34, 53, 54].



(a) Escala Linear **(b) Escala Logarítmica**

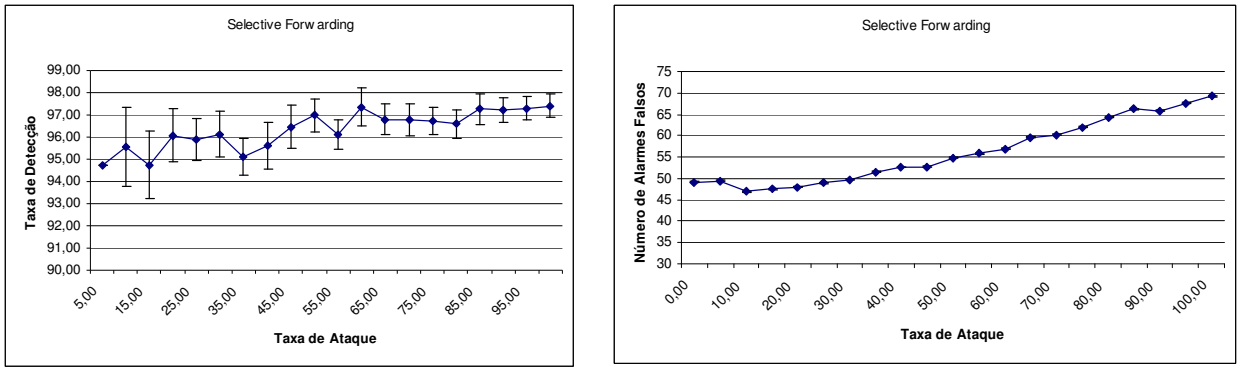
Figura 18: Falsos Positivos na ausência de intrusos.

[17] No ataque de Selective Forwarding, realizamos experimentos mantendo a taxa de ocorrência de ataque em 70% e a probabilidade de haver mensagens suprimidas pelo intruso, em cada ataque, variou entre 0 e 100%.
[18] Na ausência de intrusos não faz sentido analisar falsos negativos, uma vez que não são realizados ataques.

5.4.2. Experimento com ataque Selective Forwarding

Neste experimento, o código de um dos nós foi modificado para simular a ação do intruso realizando o ataque de Selective Forward, ou seja, um nó impede que parte dos pacotes seja retransmitida. A taxa de ocorrência de ataque define a porcentagem de iterações onde o nó intruso realiza o ataque. Como se trata do ataque de Selective Fowarding existe a probabilidade das mensagens não serem suprimidas. Neste experimento mantemos a taxa de ocorrência de ataque fixa em 70% e variamos a probabilidade de suprimir as mensagens.

Na Figura 19 (a) é apresentado o gráfico de taxa de detecção para este tipo de ataque. Em geral, à medida que a taxa de probabilidade é aumentada a taxa de detecção apresentou um pequeno aumento. Tal comportamento é esperado uma vez que à medida que os ataques aumentam fica mais fácil o IDS detectá-los. É interessante observar que mesmo sob a taxa de ataque de 5%, a taxa de detecção foi superior a 94%.



(a) Taxa de Detecção **(b) Alarmes Falsos**

Figura 19: Taxa de detecção e alarmes falsos na detecção do ataque de Selective Forwarding[19].

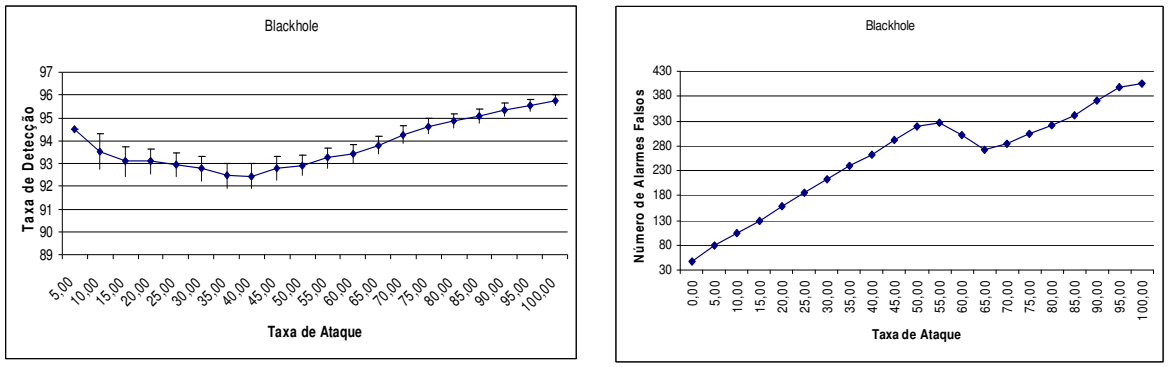
Na Figura 19 (b) é apresentada a quantidade de alarmes falsos versus a probabilidade de suprimir mensagens para simulação de 4000 iterações e taxa de ocorrência de ataque fixa em 70%. À medida que a probabilidade de suprimir mensagens aumenta, também aumenta o número de alarmes falsos, o que pode ser explicado pelo aumento de perda de mensagens ocorridas na rede devido ao aumento do número de ataques. O IDS interpreta as perdas como novos ataques sendo que às vezes é apenas a conseqüência de um ataque em interações anteriores.

[19] Os valores representados no gráfico correspondem à média obtida nas 35 vezes que o experimento foram realizados e as barras verticais representam o desvio padrão.

5.4.3. Experimento com ataque de Blackhole

Neste experimento, o código de um dos nós foi modificado para simular a ação do intruso realizando o ataque de Blackhole, onde um nó impende que todos os pacotes que passem por ele sejam retransmitidos.

Na Figura 20 (a) é apresentado o gráfico com as taxas de detecção para este tipo de ataque. A taxa de detecção variou muito pouco em relação à taxa de ataque apresentando uma ligeira queda até 45% de taxa de ataque e uma ligeira alta a partir deste valor até o final. Tal comportamento pode ser explicado pelo fato de em taxas de ataques menores alguns ataques passarem despercebidos pelo IDS e em taxas maiores os efeitos dos ataques ficam mais claros e detecção é facilitada.



(a) Taxa de Detecção **(b) Alarmes Falsos**

Figura 20: Taxa de detecção e alarmes falsos na detecção do ataque de Blackhole.

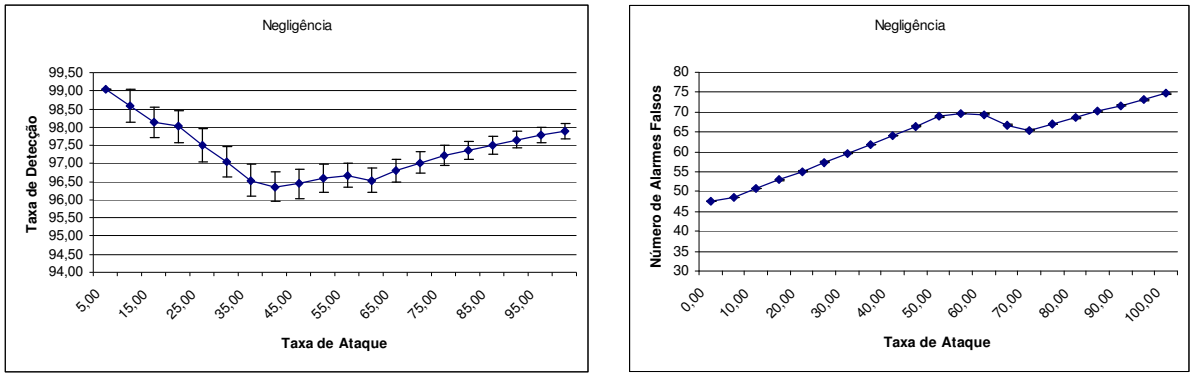
Na Figura 20(b) é apresentada a quantidade de alarmes falsos versus a taxa de ocorrência de ataques para simulação de 4000 iterações. À medida que a taxa de ataque foi aumentada, a porcentagem de alarmes falsos também aumentou porque os efeitos dos ataques se propagaram em interações subseqüentes confundindo o IDS. Mas é importante observar que o número de alarmes falsos não chegou a passar de 430 alarmes falsos, ou seja, 11% do total de eventos analisados.

5.4.4. Experimento com ataque de Negligência

Para este experimento, o código de um dos nós foi modificado para simular a ação do intruso realizando o ataque de Negligência, onde um nó deixa de sensoriar e enviar dados para a estação base.

Na Figura 21 (a) são apresentadas as taxas de detecção para este tipo de ataque. A taxa de detecção foi superior a 96% obtendo valores melhores quando submetido a taxa de ataque

inferiores a 40% e superiores a 65%. A ligeira baixa de desempenho obtida no taxas de ataque intermediárias ocorre porque o IDS confunde a perda devido aos ataques com falhas naturais da rede.



(a) Taxa de Detecção **(b) Alarmes Falsos**

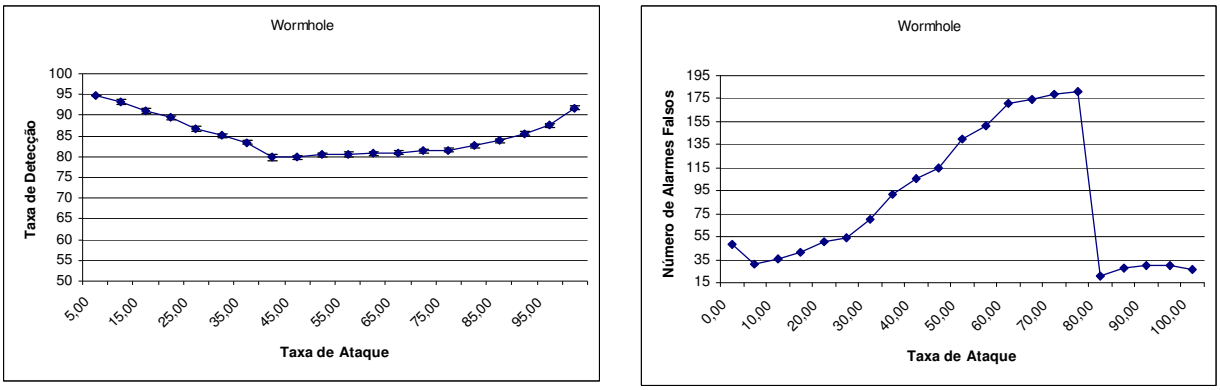
Figura 21: Taxa de detecção e alarmes falsos na detecção do ataque de Negligência.

Na Figura 21(b) é apresentado o número de alarmes falsos versus a taxa de ocorrência de ataques. O número de alarmes falsos foi sempre inferior a 75 ocorrências apresentando um crescimento praticamente constante à medida que o número de ataques aumentou o que era esperado uma vez que a principal causa de alarmes falsos são falhas provocadas em conseqüência de ataques ocorridos em interações anteriores que fazem com que o IDS contabilize um novo ataque.

5.4.5. Experimento com ataque de Wormhole

Para este experimento o código de um dos nós foi modificado para simular a ação do intruso realizando o ataque de Wormhole, onde um nó se apresenta falsamente como vizinho de vários outros, apesar dos vizinhos não possuírem alcance direto com o nó intruso.

Na Figura 22 (a) é apresentado o gráfico com as taxas de detecção para este tipo de ataque. Até atingir a taxa de ataque de 35%, à medida que a taxa de ataque é aumentada, a taxa de detecção diminuiu até atingir 80%. A partir daí a taxa de detecção passa a aumentar até atingir 90%.



(a) Taxa de Detecção **(b) Alarmes Falsos**

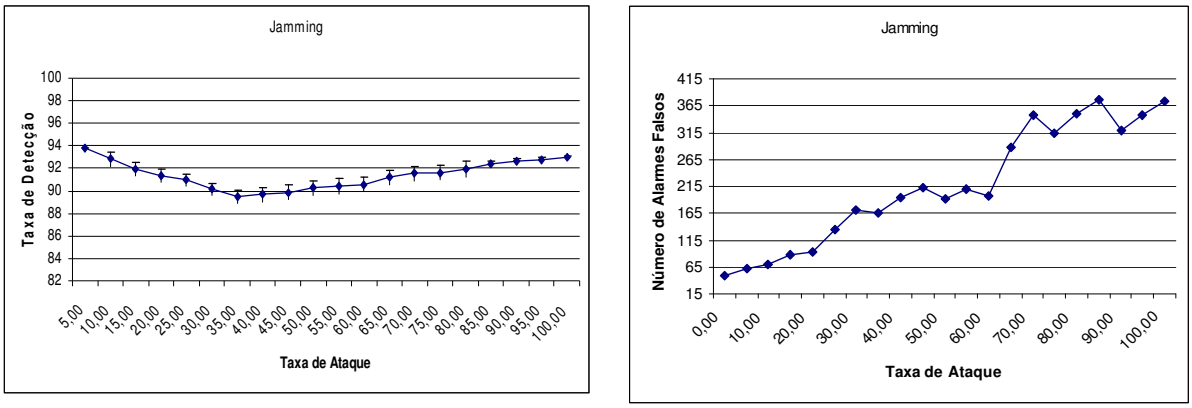
Figura 22: Taxa de detecção e alarmes falsos na detecção do ataque de Wormhole.

Na Figura 22 (b) o número de alarmes falsos é apresentado. Até a taxa de ataque de 80%, o número de alarmes falsos aumenta até atingir o valor de 175 alarmes falsos em 4000 eventos analisados. A partir daí, o IDS reage e a porcentagem de falsos positivos volta a ficar inferior a 35. A queda brusca de alarmes falsos a partir da taxa de ataque de 80% é explicada pelo fato de até então o IDS confundir falhas naturais da rede com falhas provocadas pelos ataques, a partir daí a o número de falhas fica muito acima do normal uma vez que praticamente em todos as iterações o intruso provoca perdas consideráveis na rede, facilitando assim a detecção.

5.4.6. Experimento com ataque de Jamming

Para este experimento o código de um dos nós foi modificado para simular a ação do intruso realizando o ataque de Jamming, onde o intruso provoca interferência no meio, causando a perda de mensagens entre vizinhos. Na Figura 23 (a) é apresentado o gráfico com as taxas de detecção para este tipo de ataque.

Até atingir a taxa de ataque de 35%, à medida que a taxa de ataque é aumentada a taxa de detecção diminui ligeiramente, atingindo o valor de 89%. A partir daí a taxa de detecção passa a aumentar até atingir 93%.



(a) Taxa de Detecção **(b) Alarmes Falsos**

Figura 23: Taxa de detecção e alarmes falsos na detecção do ataque de Jamming.

Na Figura 23(b) é apresentado o número de alarmes falsos que aumenta à medida que a intensidade do ataque aumenta chegando a atingir 377 alarmes falsos. O grande número de alarmes falsos se deve ao fato do efeito do ataque se propagar em iterações subseqüentes à iteração onde o ataque foi realizado, mas assim como nos demais ataques, não chega a ultrapassar 11% do total de eventos analisados.

5.5. Análise dos Resultados

Como vimos, para quatro dos cinco ataques analisados a taxa de detecção foi superior a 88%. A única exceção foi para o ataque de Wormhole onde foi superior a 80%. O número máximo de alarmes falsos, no pior caso, foi inferior a 405 em um universo de 4000 eventos analisados como mostra a Tabela 5.1. O ataque que apresentou o maior número de alarmes falsos foi o Blackhole porque os efeitos deste ataque se propagaram nas iterações posteriores em uma intensidade maior do que nos demais ataques confundindo assim o IDS.

Tabele 5.1 – Quantidade máxima de alarmes falsos (falsos positivos) em relação a 4000 eventos analisados

Experimento	Alarmes Falsos
Negligência	75
Selective Forwarding	69
Blackhole	405
Wormhole	181
Jamming	374

De maneira geral a taxa de detecção apresentou picos em torno da taxa de ocorrência ataque de 25% a 50%, o que pode ser explicado por termos uma maior combinação de

comportamento normal e comportamento comprometido pelo intruso do que acontece nos extremos onde o intruso não se manifesta muito ou onde o intruso ataca com mais freqüência. Os falsos positivos por sua vez, em geral, aumentaram com o aumento da taxa de ocorrência do ataque. A exceção foi o ataque de Wormhole, onde a porcentagem de falsos positivos aumentou até a taxa de 80% e a partir da taxa de 80% caiu abruptamente e se manteve baixa até o final, o que pode ser explicado pelo fato que com o aumento da taxa de ocorrência desse tipo de ataque a rede tende a se comprometer mais seriamente dado o grande números de subárvores que é comprometido.

Os resultados são bons se comparamos com aqueles apresentados em [53] e [54]. Na avaliação feitas nesses trabalhos, para IDS voltados para sistemas convencionais, foram apontados resultados onde a taxa de detecção variou entre 63% a 93% dependendo da quantidade de alarmes falsos por dia. Os resultados também são satisfatórios se compararmos com os obtidos por Silva et al. [34], onde a taxa de detecção ficou em torno de 75%.

Mas é importante ressaltar que é possível melhorar esses resultados através de um ajuste fino nas probabilidades *a priori* do IDS de acordo com aplicação alvo e com o modelo de falhas correspondente.

5.6. Conclusão

Implementamos a estratégia de detecção do Capítulo 3 utilizando a arquitetura descrita no Capítulo 4 em um protótipo funcional do sistema e simulamos intrusões em uma RSSF. Criamos um analisador de taxa de detecção e alarmes falsos e fizemos experimentos utilizando cinco tipos diferentes de ataques, além de analisar o comportamento do IDS na rede sem intrusos. Apresentamos resultados satisfatórios se comparados com aqueles encontrados em [34, 53, 54].

Capítulo 6

Conclusão

O objetivo deste trabalho, que foi projetar uma solução para o problema de detecção de intrusos em RSSF sem que os nós sensores precisassem ser alterados, foi atingido pela proposição de uma solução centralizada na estação base e focada em utilizar as informações já disponíveis.

Vimos à necessidade de se trabalhar esse tema de pesquisa dado a falta de trabalhos nessa área no contexto de RSSF, além da grande utilidade da detecção de intruso dessa no objetivo de garantir a segurança da rede, uma vez que todo sistema de prevenção pode falhar.

Delimitamos o escopo do trabalho às RSSF planas, homogêneas, contínuas, estáticas, com nós semelhantes aos mica motes e uma estação base. Listamos as principais restrições de projeto necessárias para garantir uma solução não invasiva, tais como a não alteração de hardware, software e protocolos da rede. Definimos o modelo de segurança onde o intruso é capaz de escutar e alterar dados do ambiente, violar e inserir nós e utilizar outros tipos de hardware como um laptop. Além disso, delimitamos um subconjunto mínimo de informações que poderia ser utilizado no escopo da solução, que consistiu em informações de produção, estado operacional e árvore de roteamento.

Procuramos delimitar as premissas considerando as informações das RSSF atuais e com embasamento prático obtido através de experimentos e desenvolvimento de protótipos realizados no contexto do projeto SensorNet [93]. Por exemplo, quando assumimos que teríamos a produção do nó fizemos inspirados no fato das redes reais que manuseamos já possuírem essa informação. Da mesma maneira quando optamos por não contar com o mapa de energia o fizemos inspirados pela dificuldade de se conseguir essas informações nos nós que manuseamos. Mas a pesquisa na

área de RSSF está muita ativa e acreditamos que hoje já seja viável assumir a informação de energia como disponível, como apontado por muitos [63, 80, 84, 90], desta forma o mapa de energia poderá ser anexado à solução para aumentar sua eficiência.

Um dos grandes desafios na solução do problema, foi estruturar o modelo de informação de modo que técnicas de detecção de intrusos pudessem ser aplicadas em RSSF, uma vez que os IDSs atuais utilizam um modelo de informação muito diferente mesmo para redes *ad hoc* [31, 32]. Para tanto, trabalhamos com o conceito de mapas, conceito inspirado pelo modelo de informação proposto em [90]. Mas adaptamos a idéia de se utilizar mapas para o contexto de detecção de intrusos e o estendemos criando o conceito de comportamento, como sendo um conjunto de mapas gerados em um período de tempo.

Outra contribuição do trabalho foi estruturar e propor maneiras de organizar o modelo de informação dos nós sensores, além da proposição de técnicas para analisar a presença de intruso baseado no modelo de informação proposto. Para o Mapa Operacional, verificamos não ser viável sua construção de forma totalmente confiável, dado as premissas nas quais nos propomos a trabalhar. Para resolver esse problema, propomos a oficialização dessa incerteza através da definição de mapas operacionais baseados em probabilidades e da utilização de redes bayesianas como método de análise, uma vez que é um método capaz de tratar com o princípio da incerteza de forma lógica e consistente. Nesse contexto, apresentamos a modelagem de uma rede bayesiana aplicada ao modelo de informação proposto. Mas no protótipo do IDS trabalhamos com um valor de probabilidade fixo que procurou encapsular a probabilidade do nó falhar. O ideal é que esse valor seja calculado dinamicamente através da realimentação da rede bayesiana com informações extraídas da rede em tempo real.

Apresentamos, também, uma arquitetura extensível que viabiliza a construção de um sistema de detecção baseado na estratégia proposta. A arquitetura se apóia em quatro pilares principais: mapas, base de conhecimento, estratégia de análise de intrusão e fonte de dados. Mostramos que a construção dos mapas pode ser feita a partir das informações que chegarem à estação base. Utilizamos uma base de conhecimento composta por axiomas probabilísticos. A análise de intrusão foi feita utilizando os diversos mapas construídos ao longo do tempo, sendo que consideramos que comportamento da rede pode ser classificado como um processo de Markov de primeira ordem, ou seja, utilizaremos em cada análise apenas o último mapa gerado e não todo o histórico de mapas ao longo do tempo. É importante que as dependências entre as informações seja estudado mais profundamente para comprovar que se encaixa com um processo de Markov de primeira ordem. Além disso, acreditamos que o uso de Redes Bayesianas Dinâmicas tende a

aumentar a eficiência da solução dentro de um custo computacional aceitável, uma vez que esse método permite introduzir o conceito de "memória" na estratégia de análise de forma que os mapas do período em análise possam ser confrontados com mapas de períodos passados aumentando o volume e a qualidade das informações para análise.

Para validação da solução construímos um protótipo em Java, simulamos a rede e cenário de intrusão baseados em cinco ataques para: Negligência, Selective Forwarding, Blackhole, Wormhole e Jamming. Utilizamos duas métricas para validar a solução: falsos positivos e falsos negativos. Para facilitar a análise construímos um programa em Java que calcula o número de falsos negativos e falsos positivos, além de sumarizar esses dados calculando a média e o desvio padrão. Outros ataques não foram analisados e sabemos para que nosso IDS seja eficiente para ataques de repetição e alteração de mensagens o método de análise precisa ser estendido.

Para quatro dos cinco ataques analisados a taxa de falsos negativos ficou abaixo de 12%. A única exceção foi para o ataque de Wormhole onde ficou abaixo de 20%. A porcentagem de falsos positivos ficou abaixo de 12% para todos os ataques analisados e no caso de ausência de intruso essa porcentagem se limitou a 2,5%. Os resultados são bons se comparados com aqueles apresentados em [53, 54] para redes convencionais, onde a porcentagem de falsos negativos variou de 7% a 37%. Os resultados também são satisfatórios se compararmos com os obtidos por Silva et al. [34] , onde a porcentagem de falsos negativos ficou em torno de 25% em média.

Vemos aumentar a eficácia do IDS testando-o com outros tipos de ataques e utilizando o conhecimento de um especialista para ajustar o sistema de acordo com ataques ou aplicações específicas. O cálculo da probabilidade do nó estar operacional pode ser melhorado através de modelos que simulem o comportamento da rede ou da utilização de informações mais específicas da RSSF alvo, por exemplo, um modelo de falhas especificado pelo projetista da rede ou construído a partir de dados empíricos será muito útil na aplicação real do IDS.

Outra possibilidade de melhoramento, é explorar a característica de a arquitetura proposta permitir pontos de extensão. Por exemplo, é possível criar novos mapas e atualizar ou trocar as estratégias de análise, além de ser possível atualizar a base de conhecimento. A própria integração deste trabalho com a solução proposta por Silva et. al [34] nos parece ser muito promissora, uma vez que os dados gerados nesta solução poderão ser organizados na forma de um "Mapa de Intrusão" e ser utilizado como mais um mapa na arquitetura da solução proposta neste trabalho. Além disso, a utilização de outros métodos de análise, baseado no método de Dempster-Shafer [26] por exemplo, pode ser explorada bastando para isso alterar a estratégia de análise de acordo com a arquitetura proposta.

A extensão da solução para atender a redes consideradas como fora do escopo deste trabalho também é bem interessante. Para redes orientadas a eventos, por exemplo, a base de conhecimento pode ser estendida para representar o comportamento normal da aplicação. Para tanto é necessário estudar a rede alvo e organizar seu comportamento seguindo o modelo de informação proposto neste trabalho. Para redes hierárquicas, uma alternativa é considerar uma rede formada pelos pontos de agregação, como se correspondessem aos nós individuais na solução plana. Desta forma quando a rede for hierárquica os nós que constituirão os mapas serão apenas aqueles que produzem informação que será diretamente direcionada à estação base.

O estabelecimento de uma fase de treinamento onde o administrador da rede verifica os resultados gerados pelo IDS e calibra as probabilidades, através de técnicas semelhantes àquelas propostas por Axelson [16] pode ser utilizada para calibrar o IDS e melhorar sua eficiência. Ou, a fase de treinamento pode ser feita como proposto por Kurugel e Vigna [41], onde Markov [28] são construídos através do uso de redes bayesianas durante o treinamento. O modelo construído é utilizado durante a detecção.

Dos desafios proposto em [35] boa parte foram solucionados através deste trabalho e do trabalho de Silva et. al [34], mas ainda há muitos desafios a serem explorados na área de detecção de intrusão em RSSF.

Referências Bibliográficas

SEGURANÇA EM RSSF

1 Saurabh Ganeriwal and Mani B. Srivastava. Reputation-based Framework for High Integrity Sensor Network. *ACM SANS'04*, October 25, 2004.

2 S. Ganeriwal, R. Kumar, C. C. Han. S. Lee, M. B.Srivastava. Location & Identity based Secure Event Report Generation for Sensor Networks. *NESL Technical Report*,May 2004.

3 F. Ye, H. Luo, S. Lu, L. Zhang. Statistical En-route Detection and Filtering of Injected False Data in Sensor Networks. *In Proceedings of IEEE Infocom*, 2004.

4 B. Przydatesk, D. Song, A. Perrig. SIA: Secure Information Aggregation in Sensor Networks. *In Proceedings of ACM SenSys*, 2003.

5 C. Karlof, D. Wagner. Secure routing in sensor networks:Attacks and countermeasures. *Elsevier AdHoc Networks journal*, May 2003.

6 A. Perrig, J. Stankovic, D. Wagner. Security in Wireless Sensor Networks. *Communications of the ACM*, 2004.

7 C. Karlof, N. Sastry, D. Wagner. TinySec: Link Layer Encryption for Tiny Devices. *To appear in ACM SenSys*,2004.

8 R. Watro, D. Kong, S. F. Cuti, C. Gardiner, C. Lynn, P. Kruus. TinyPK: Securing Sensor Networks with Public Key Technology. *To appear in second workshop on Security in Sensor and Ad-hoc Networks*, 2004.

9 A. Perrig, R. Szewczyk, V. Wen, D. Culler, D. Tygar. SPINS: Security Protocols for Sensor Networks. *Wireless Networks Journal*, September 2002.

10 L. Eschenauer, V. D. Gligor. A key Management Scheme for Distributed Sensor networks. *In Proceedings of ACM CCS*, November 2002.

11 H. Chan, A. Perrig, D. Song. Random Key Predistribution Schemes for Sensor Networks. *In Proceedings of IEEE Symposium on Security and Privacy*, 2003.

12 D. Liu, P. Ning. Establishing pairwise keys in distributed sensor networks. *In Proceedings of ACM CCS*, October 2003.

13 Anthony D. Wood and John A. Stankovic. Denial of Service in Sensor Networks. *IEEE Computer*, October 2002.

14 S. de Oliveira, H. C. Wong , J. M. S. Nogueira. NEKAP: Estabelecimento de Chaves Resiliente a Intrusos em RSSF. *SBRC 2005*.

SEGURANÇA DE REDES

15 Willian Stallings. Cryptography and Networks Security: principles and practice. *Prentice Hall*, 2nd edition, 1998.

IDS QUE USAM REDES BAYESIANAS OU TRABALHOS RELACIONADOS ÀS REDES BAYESIANAS

16 Stefan Axelson. Combining a Bayesian Classifier with Visualisation: Understanding the IDS. *In Proceedings of ACM VizSEC/DMSEC*, October 2004.

17 Nahla Ben Amor, Slem Benferhat, Zied Elouedi. Naive Bayes vs Decision Trees in Intrusion Detection Systems. *In Proceeding of ACM SAC*, March 2004.

18 W. Lee and D. Xiang. Information-theoretic measures for anomaly detection. *In IEEE Symposium on Security and Privacy*, Oakland, California, USA, 14–16 May 2001. IEEE.

19 M. Ramadas, S. Ostermann, and B. Tjaden. Detecting anomalous network traffic with self-organizing maps. *In Proceedings of the Sixth International Symposium on Recent Advances in Intrusion Detection*, LNCS, Pittsburgh, PA, USA, 8–10 September 2003. Springer Verlag.

20 Cooper, G. F.: Computational complexity of probabilistic inference using Bayes belief networks. *Artificial Intelligence*, Vol. 42, 393-405, 1990.

21 Valdes, A., Skinner K.: Adaptive Model-based Monitoring for Cyber Attack Detection. *In proceedings of Recent Advances in Intrusion Detection (RAID 2000)*, Toulouse, France, 80-92, 2000.

22 John, G.: Enhancements to the Data Mining Process. *PhD thesis*, Stanford University, 1997.

23 S. Russell, P. Norvig: Inteligência Artificial. Tradução da Segunda Edição. *Editora Campus*, 2004.

24 Bayesian Network Editor and Toolkit. http://research.microsoft.com/adapt/MSBNx/. Version 1.4.2 Outubro de 2001. Site acessado em setembro de 2005.

DEMPSTER-SHAFER'S THEORY OF EVIDENCE

25 Christos Siaterlis, Basill Maglaris. Towards Multisensor Data Fusion for DOS detection. *In Proceedings of ACM SAC*, March 2004

26 G. Shafer. A Mathematical Theory of Evidence. *Princeton University Press*, Princeton, 1976.

27 H. Wu, M. Siegel, R. Stiefelhagen, and J. Yang. Sensor fusion using Dempster-Shafer theory. *In Proceedings of IEEE Instrumentation and Measurement Technology Conference*, Anchorage, AK, USA, 2002.

MODELOS DE MARKOV

28 Andreas Stolcke and Stephen Omohundro. Hidden Markov Model Induction by Bayesian Model Merging. *Advances in Neural Information Processing Systems*, 1993.

IDS CONVENCIONAIS

29 Herve Debar, Marc Dacier and Andreas Wespi. A Revised Taxonomy for Intrusion-Detection Systems. *Research Report*, Zurich Research Laboratory, October 1999.

30 Rafael Saldanha Campello and Raul Fernando Weber. Sistemas de detecção de Intrusão. *In 19o Simpósio Brasileiro de Redes de Computadores*. SBRC Janeiro de 2001

IDS OU DETECÇÃO DE ATAQUES EM REDES AD-HOC

31 Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," *in Mobile Computing and Networking*, 2000, pp. 275–283.

32 Y.-C. Hu, A.Perrig, and D. B. Johnson, "Wormhole detection in wireless ad hoc networks," Department of Computer Science, Rice University, *Technical Report TR01-384*, June 2002.

DETECÇÃO DE INTRUSOS OU ATAQUES EM RSSF

33 W. R. Pires Jr, T. H. P. Figueiredo, H. C. Wong, and A. A.F. Loureiro, “Malicious node detection in wireless sensor networks,” *in 18th International Parallel and Distributed Processing Symposium*, Santa Fe, NM, April 2004.

34 Ana Paula Ribeiro da Silva, Antonio Alfredo Loureiro, Hao Chi Wong. Detecção de Intrusos Descentralizada em Redes de Sensores Sem Fio. *Dissertação de Mestrado UFMG*, Abril de 2005.

35 Ana Paula Ribeiro da Silva, F. A. Teixeira, H. C. Wong, and J. M. S.Nogueira, “Aspectos de detecção de intrusos em redes de sensores sem fio,” *in 22o Simpósio Brasileiro de Redes de Computadores*, maio 2004, pp. 575 – 578.

36 J. Newsome, R. Shi, D. Song, and A. Perrig, “The sybil attack in sensor networks: Analysis and defenses,” *in Proceedings of IEEE International Conference on Information Processing in Sensor Networks (IPSN 2004)*, Apr. 2004.

37 W. P. de Paula, S. de Oliveira, J. M. S. Nogueira, H. C. Wong. Detecção de Intrusos por Rotas Redundantes em RSSF. *SBSeg 2005*.

TOLERÂNCIA À INTRUSÃO

38 J. Deng, R. Han and S. Mishra. The Performance Evaluation of Intrusion-Tolerant Routing in Wireless Sensor Networks. *In the Proceedings of IPSN*, April, 2003.

ANOMALY DETECTION

39 Ilgun, K., Kemmerer., R. A., Porras, P. A.: State transition: A rule-based intrusion detection approach. IEEE Transactions on Software Engineering, 21(3), 181-199, 1995.

40 Porras, P. A., Neumann., P. G., EMERALD: Event monitoring enabling responses to anomalous live disturbances. *In proceedings of the 20th National Information Systems Security Conference*, Baltimore, Maryland, USA, NIST, 353-365, 1997.

41 C. Kuregel and G. Vigna. Anomaly Detection of Web-based Attacks. *ACM CCS’03* October 2003.

42 A.K. Ghosh, J. Wanken, and F. Charron. Detecting Anomalous and Unknown Intrusions Against Programs. *In Proceedings of the Annual Computer Security Applications Conference (ACSAC'98)*, pages 259{267, Scottsdale, AZ, December 1998.

43 C. Ko, M. Ruschitzka, and K. Levitt. Execution Monitoring of Security-Critical Programs in Distributed Systems: A Specification-based Approach. *In Proceedings of the 1997 IEEE Symposium on Security and Privacy*, pages 175{187, May 1997.

44 T. Lane and C.E. Brodley. Temporal sequence learning and data reduction for anomaly detection. *In Proceedings of the 5th ACM conference on Computer and communications security*, pages 150{158. ACM Press, 1998.

45 W. Lee and S. Stolfo. A Framework for Constructing Features and Models for Intrusion Detection Systems. *ACM Transactions on Information and System Security*, 3(4), November 2000.

46 C. Kruegel, T. Toth, and E. Kirda. Service Specific Anomaly Detection for Network Intrusion Detection. *In Symposium on Applied Computing (SAC)*. ACM Scientific Press, March 2002.

47 K. Tan and R. Maxion. "Why 6?" Defining the Operational Limits of Stide, an Anomaly-Based Intrusion Detector. *In Proceedings of the IEEE Symposium on Security and Privacy, pages 188{202,Oakland*, CA, May 2002.

MISUSE DETECTION

48 Kumar, S., Spafford., E. H.: A software architecture to support misuse intrusion detection. *In proceedings of the 18th National Information Security Conference*, 194-204, 1995.

49 R. Marty: Snort the open source network IDS, http://www.snort.org/, 2001.

50 K. Ilgun, R.A. Kemmerer, and P.A. Porras. State Transition Analysis: A Rule-Based Intrusion Detection System. *IEEE Transactions on Software Engineering*, 21(3):181{199, March 1995.

51 U. Lindqvist and P.A. Porras. Detecting Computer and Network Misuse with the Production-Based Expert System Toolset (P-BEST). *In IEEE Symposium on Security and Privacy*, pages 146{161, Oakland, California, May 1999.

52 V. Paxson. Bro: A System for Detecting Network Intruders in Real-Time. *In Proceedings of the 7th USENIX Security Symposium*, San Antonio, TX, January 1998.

AVALIAÇÃO DE IDS

53 Lippmann,R.P.,Fried, D., Graf, I., Haines, J., Kendall, K., Mcclung, D., Webber, D., Webster, S., Wyschograd, D., Cunninghan, R., and Zissman, M. 2000. Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. *In Proceedings of the on DARPA Information Survivability Conference and Exposition (DISCEX '00, Hilton Head, South Carolina, Jan. 25-27). IEEE Computer Society Press*, Los Alamitos, CA, 12-26.

54 Stefan Axelsson. The base-rate fallacy and its implications for the difficulty of intrusion detection. *In 6th ACM Conference on computer and communications security*, pages 1--7, Kent Ridge Digital Labs, Singapore, November 1999.

SIMULAÇÃO EM RSSF

55 Maneesh Varshney, Rajive Bagrodia. Detailed Models for Sensor Network Simulations and their Impact on Network Performance. *In Proceedings of the ACM MSWiM'04*, October 2004, Venezia, Italy.

56 S. Park, A. Savvides, and M. B. Srivastava. SensorSim: A Simulation Framework for Sensor Networks. *In Proceedings of the 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2000.

57 P. Levis, N. Lee, M. Welsh, and D. Culler. TOSSIM: Accurate and scalable simulation of entire tinyos applications. *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2003.

58 L. F. Perrone and D. Nicol. A Simulator for TinyOS Applications. *In Proceedings of the 2002 Winter Simulation Conference*, 2002.

59 X. Zeng, R. Bagrodia, and M. Gerla. Glomosim: a library for parallel simulation of large-scale wireless networks. *Parallel and Distributed Simulations (PADS)*, May 1998.

60 Qualnet. http://www.scalable-networks.com, Agosto 2005.

BATERIAS E ENERGIA

61 D. Linden and T. B. Reddy. Handbook of Batteries. Third edition.

62 D. Rakhmatov and S. B. K. Vrudhula. Energy management for battery-powered embedded systems. *ACM Transactions on Embedded Computing Systems*, 2002.

63 R. A. F. Mini, B. Nath, and A. A. F. Loureiro, "Prediction based approaches to construct the energy map for wireless sensor networks," *in 21o. Simpósio Brasileiro de Redes de Computadores*, Maio 2003.

NS E EXEMPLOS DE TRABALHOS QUE USARAM O NS PARA SIMULAÇÃO EM RSSF

64 Ns-2. http://www.isi.edu/nsnam/ns, maio 2005.

65 Y. Yao and J. Gehrke. The Cougar Approach to In-Network Query Processing in Sensor Networks. *ACM SIGMOD Record*, 31(3):9–18, 2002.

66 S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. GHT: A Geographic Hash Table for Data-Centric Storage. *In Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

67 D. Braginsky and D. Estrin. Rumor Routing Algorithm for Sensor Networks. *In Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

DETECÇÃO DE FALHAS OU NÓS FALTOSOS

68 Jessica Staddon, Dirk Balfanz, Glenn Durffee. Efficient Tracing of Failed Nodes in Sensor Networks. *In Proceeding of the ACM WSNA*. September 2002, Atlanta, USA.

69 Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. *Mobile Computing and Networking*, pages 255--65, 2000.

SISTEMAS E APLICAÇÕES EM RSSF

70 J. Elson, L. Girod, and D. Estrin. Fine-Grained Network Time Synchronization using Reference Broadcasts. *In Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002)*, Boston, MA, USA., dec 2002.

71 P. Levis and D. Culler. Mat´e: A Tiny Virtual Machine for Sensor Networks. *In International Conference on Architectural Support for Programming Languages and Operating Systems*, San Jose, CA, USA, Oct. 2002.

72 J. Liu, P. Cheung, L. Guibas, and F. Zhao. A Dual-Space Approach to Tracking and Sensor Management in Wireless Sensor Networks. *In Proceedings of First ACM International Workshop on Wireless Sensor Networks and Applications*, September 2002.

73 S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks. *In OSDI*, 2002.

74 A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless Sensor Networks for Habitat Monitoring. *In ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, Atlanta, GA, USA, Sept. 2002.

75 Y. Yao and J. Gehrke. The Cougar Approach to In-Network Query Processing in Sensor Networks. *ACM SIGMOD Record*, 31(3):9–18, 2002.

76 Mani Srivastava, Richard Muntz, and Miodrag Potkonjak, "Smart Kindergarten: Sensor-based Wireless Networks for Smart Developmental Problem-solving Environments," in

The Seventh Annual International Conference on Mobile Computing and Networking, July 2001, pp. 132-138.

PROTOCOLOS EM RSSF

77 J. S. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan. Building Efficient Wireless Sensor Networks with Low-Level Naming. *In Proceedings of the 18th ACM Symposium on Operating Systems Principles*, Banff, Canada, October 2001.

78 C.-Y. Wan, A. T. Campbell, and L. Krishnamurthy. PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks. *In Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications*, pages 1–11. ACM Press, 2002.

79 A. Woo and D. Culler. A Transmission Control Scheme for Media Access in Sensor Networks. *In International Conference on Mobile Computing and Networking (MobiCom 2001)*, Rome, Italy, July 2001.

80 W. Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. *In Proceedings of IEEE Infocom 2002*, New York, NY, USA., June 2002.

81 C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable And Robust Communication Paradigm For Sensor Networks. *In Proceedings of the International Conference on Mobile Computing and Networking*, Aug. 2000.

82 Ana Paula R. Silva, Fernando A. Teixeira, Rafael K. Lage, Antônio A.F. Loureiro, José Marcos S. Nogueira, Linnyer B. Ruiz. Using a Distributed Snapshot Algorithm in Wireless Sensor Networks. *The 9th International Workshop on Future Trends of Distributed Computing Systems*, 2003.

83 A. Manjeshwar and D. P. Agrawal. Teen: A routing protocol for enhanced efficiency in wireless sensor networks. *In 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing (IPDPS)*, April 2001.

84 W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, Energy efficient communication protocol for wireless microsensor networks, *in 33rd Annual Hawaii International Conference on System Sciences*, 2000, pp. 3005–3014.

85 P. Buonadonna, J. Hill, and D. Culler. Active message communication for tiny networked sensors. *In Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'01)*, April 2001.

86 Y. Yu, R. Govindan, and D. Estrin, Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks, University of California at Los Angeles Computer Science Department, *Tech. Rep. UCLA/CSD-TR-01-0023*, May 2001.

87 B. Karp and H. T. Kung, GPSR: greedy perimeter stateless routing for wireless networks, *in Mobile Computing and Networking*, 2000, pp. 243–254.

88 F. Ye, A. Chen, S. Lu, and L. Zhang. A scalable solution to minimum cost forwarding in large sensor networks. *In Tenth International Conference on Computer Communications and Networks*, 2001, pp. 304–309.

89 D. Braginsky and D. Estrin. Rumour routing algorithm for sensor networks. *In First ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

GERÊNCIA EM RSSF

90 Linnyer Beatrys Ruiz, José Marcos Nogueira and Antonio A. F. Loureiro. MANNA: A Management Architecture for Wireless Sensor Networks. *IEEE Communications Magazine*, 41(2):116-125, February 2003.

TINYOS E MICA MOTES

91 D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler. The nesC Language: A Holistic Approach to Networked Embedded Systems. *In Proceedings of Programming Language Design and Implementation (PLDI)*, June 2003.

RSSF EM GERAL

92 Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for network sensors. *In Proceedings of ASPLOS*, 2000.

93 Sensornet: Arquitetura, protocolos, gerenciamento e aplicações em redes de sensores sem fio. *Departamento de Ciência da Computação - (DCC – UFMG)*, http://wwww.sensornet.dcc.ufmg.br. Site acessado em setembro de 2005.

ENDEREÇAMENTO EM RSSF

94 John S. Heidemann, Fabio Silva, Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, and Deepak Ganesan. Building efficient wireless sensor networks with low-level naming. *In Symposium on Operating Systems Principles*, pages 146{159, 2001.

95 L. Nirupama, B. Deborah, and E. Deborah. Scalable coordination for wireless sensor networks: Self-configuring localization systems. *International Symposium on Communication Theory and Applications (ISCTA)*, July 2001.

DESENVOLVIMENTO DE SOFTWARE/ALGORITMOS

96 E. Gamma et al: “Design Patterns: Elements of Reusable Object-Oriented Software”, *Addison-Wesley*, 1995.

97 A. Segall. Distributed network protocols. *IEEE Transactions on Information Theory*, 29:23-35, 1983.

98 D. Alur, J. Crupi, D. Malks: “Core J2EE Patterns: Best Practices and Design Strategies”, *Prentice Hall*, 2003. http://java.sun.com/blueprints/corej2eepatterns/

99 Booch, G. Object Solutions. *Addison-Wesley*, 2nd edition,1995.

Apêndice A

Resenha de alguns Trabalhos Relacionados

Neste apêndice apresentamos algumas resenhas de trabalhos estudados com o objetivo de determinar o método de detecção de intruso e o comportamento normal da rede.

A.1 Trabalhos estudados com o objetivo de definir o Método de Detecção

REPUTATION-BASED FRAMEWORK FOR HIGH INTEGRITY SENSOR NETWORK [1]

Objetivo/Contribuição

O objetivo do artigo é usar o conceito de reputação, como existe na sociedade, para definir o nível de confiança que um nó deve ter em relação ao seu vizinho.

Pontos Fortes/Interessantes

O artigo apresenta boas justificativas para a necessidade de segurança em RSSF tais como:

- Criptografia não pode impedir, totalmente, a inserção de dados maliciosos, que sejam oriundos de nós adversários ou defeituosos.
- A habilidade das RSSF executarem suas tarefas depende não somente da habilidade de comunicar com outros nós como também de sua habilidade de sensoriar o ambiente e coletivamente processar os dados. A estratégia de tomada de decisão de forma coletiva,

baseada em uma confiança implícita entre os nós, pode ser utilizada por adversários para se aproveitar de brechas de segurança através do comprometimento de nós legítimos.

- O nós são projetados para serem de baixo custo, o que torna inviável a fabricação de nós resistentes à violação. Um adversário pode, de maneira imperceptível, tomar o controle de um nó comprometendo-o fisicamente. Um adversário pode potencialmente inserir dados ou decisões falsas para comprometer toda a rede.
- Cita trabalhos que tentam restringir o impacto dos ataques através de redundância tais como o SERP [2], SEF [3] e SAI [4]
- Lembra que dados autenticados não são necessariamente dados íntegros
- Resume um conjunto de trabalhos relacionados à segurança em RSSF e os agrupa como trabalhos que estudam os impactos dos ataques [5, 6] ou trabalhos que procuram garantir a segurança da comunicação considerando as restrições dos nós sensores tais como o SPINS [8], TinySec [7], INSENS [38], TinyPK [8], SERP [2], SEF [3], etc. O estabelecimento e gerenciamento de chaves criptográficas [8, 2, 10, 11, 12] formam a espinha dorsal desses esquemas. O autor justifica que o foco do artigo é outro (estabelecer grau de confiança), uma vez que apenas a criptografia não é suficiente.
- Discute a idéia de "Gerenciamento de Confiança" para segurança de sistemas distribuídos, referenciando trabalhos relacionados à Internet.
- Aponta que existem soluções parecidas em redes *ad-hoc* mas que nunca foram desenvolvidas e coloca como um dos objetivos do trabalho mostrar o sistema proposto funcionando em nós sensores como o Berkeley motes

Resumo da solução proposta

Foi proposto um framework chamado "Reputation Based Framework for Sensor Network (RFSN)", onde os nós mantêm a reputação dos outros nós da rede. Um nó monitora o comportamento dos outros nós, baseado nisto ele constrói sua reputação ao longo do tempo. A reputação é usada para avaliar seu grau de confiança e prever seu comportamento. Na hora de colaborar, um nó só coopera com aqueles que confia. O objetivo final da RFSN é gerar uma comunidade de nós sensores confiáveis.

Para validar a solução foram feitas simulações usando uma instância do framework BRSN e uma versão hipotética. A versão BRSN utiliza uma formulação Bayesiana para representar a reputação e atualizá-la continuamente baseado em novas observações diretas e indiretas. Os autores analisaram a qualidade do sistema contra ataques comuns realizados contra sistemas de

reputação de comércio eletrônico (como ebay e yahoo). Não foi realizada uma análise do custo e viabilidade da solução em RSSF, apesar dos autores argumentarem que a solução é viável e escalável porque as informações de reputação são armazenadas e trocadas em um contexto local, ou seja, apenas entre vizinhos.

COMBINING A BAYESIAN CLASSIFIER WITH VISUALISATION: UNDERSTANDING THE IDS [16]

Objetivo/Contribuição
Utilização de um mecanismo de visualização que permita que o administrador do sistema visualize o estado do classificador bayesiano de modo que possa entender como o IDS está "aprendendo" e como isso afetará os resultados.

Pontos Fortes/Interessantes
O autor advoga a favor do uso de redes bayesianas e cita o exemplo de sucesso do leitor de e-mail do grupo Mozilla, onde um classificador bayesiano é utilizado em um filtro de SPAM e o usuário treina o classificador. Defende a necessidade de uma interface onde um operador possa interferir no processo decisório do classificador bayesiano de um IDS. Na ferramenta proposta, o administrador pode marcar as ações como maliciosas, neutras ou boas. O autor cita Paul Graham (http://www.paulgraham.com/spam.html) como o responsável pela popularização das redes bayesianas.

NAIVE BAYES VS DECISION TREES INTRUSION DETECTION SYSTEMS [17]

Objetivo/Contribuição
O trabalho apresenta um estudo experimental do uso de Naive Bayes em detecção de intrusos. Afirmam que redes bayesianas são ferramentas poderosas para decisão e raciocínio baseados em incertezas e que Naive Bayes é um tipo muito simples de rede bayesiana particularmente eficiente para tarefas de inferência. Entretanto Naive Bayes baseiam-se em premissas muito forte de independência entre evidências. Os autores mostram que apesar disso, conseguem resultados muito competitivos quando comparados com outros métodos de detecção, inclusive quando comparado com Árvores de Decisão.

Pontos Fortes/Interessantes

A introdução apresenta um resumo bem estruturado sobre área de detecção de intrusos, descreve as redes bayesianas, citando o trabalho de Valdes [20] que propôs uma estratégia híbrida de detecção baseado em redes de Bayes. Define redes bayesianas como ferramentas para lidar com a incerteza através de um framework baseado na teoria da probabilidade. São compostas por grafos acíclicos e direcionados que representam relações causais, e probabilidade condicionais (para cada nó dado seus pais) para expressar incertezas sobre as relações causais. Valdes [20] usa uma forma simples de redes bayesianas, chamada Naive Bayes, composta por dois níveis: um nó raiz que representa uma classe de estados (normal e diferentes tipos de ataques), e alguns nós folha, cada um deles contém uma informação de uma conexão.

Naive Bayes possui uma série de vantagens dado sua estrutura simples. Em particular, a construção das Naive Bayes é muito simples. A inferência (classificação) é realizada em tempo linear (enquanto inferências em Redes Bayesianas que trabalham com uma estrutura mais genérica é um problema NP - completo [20]). Além disso, a construção de Naive Bayes é incremental, no sentido que pode ser facilmente atualizada (ou seja, é sempre fácil considerar e utilizar novos casos coletados). Entretanto, Naive Bayes faz uma forte suposição de relação de independência: evidências são independentes no contexto do conjunto de estados. Tal premissa nem sempre é verdadeira e pode ter uma influência negativa nos resultados inferidos. O trabalho procura mostrar que apesar dessa ressalva as Naive Bayes são bastante eficazes na prática.

Os autores mostram através de resultados experimentais que o desempenho das Naive Bayes são muito competitivos uma vez que, comparados com árvores de decisão, que é considerado um dos melhores métodos de aprendizado, as diferenças dos resultados não foi significativa. Além disso, do ponto de vista computacional, Naive Bayes é muito mais eficiente nas tarefas de aprendizados e classificação. A construção de Naive Bayes é linear enquanto a de árvores de decisão ótimas é um problema NP - Completo[20].

TOWARDS MULTISENSOR DATA FUSION FOR DOS DETECTION [25]

Objetivo/Contribuição

[20] Hyafil, L., Rivest, R. L: Constructing optimal binary decision trees is NP-complete. Information Processing Letters, 5(1):15-17, 1976.

Os autores defendem o uso da Teoria da Evidência de Dempster-Shafer [26,27] como fundamentação matemática para o desenvolvimento de um novo sistema de detecção de DoS.

A principal razão da escolha é que eles não possuem um bom modelo do estado normal da rede, o que fez com eles excluíssem métodos "físicos" tais como filtros de Kalman que requerem o conhecimento da matriz de transição de estados. Descartaram, também, métodos que necessitam de dados de treinamento, como redes neurais, porque dados representativos do estado normal são difíceis de serem obtidos e consomem muito tempo para serem construídos. Além disso, a solução proposta pretende utilizar diferentes e heterogêneas fontes de informação.

Os autores comparam a estratégia com o método de redes Bayesianas e apontam as seguintes desvantagens do método de Bayes:

- Necessidade do conhecimento *"a priori"* da distribuição de probabilidade dos estados
- Não provê nenhuma informação sobre a qualidade dos resultados em termo da confiança nas evidências ou em conflitos entre as evidências.

Os autores definem o método de Dempster-Shafer (DS) como uma extensão das redes Bayesianas. Sumarizam o método e apontam suas vantagens e desvantagens:

Vantagens:

- As probabilidades são atribuídas apenas a elementos simples do conjunto de estados e não a elementos do superconjunto dos possíveis estados.
- A Teoria da Evidência faz distinção entre incerteza e ignorância, permitindo lidar com a incerteza baseando-se em informações incompletas ou até mesmo, contraditórias.
- Não necessita de um conhecimento *a priori* ou de distribuição de possíveis estados do sistema, como acontece nas redes bayesianas e, portanto, é mais bem aplicado a problemas onde não se possui um modelo do sistema. Segundo os autores, a "Teoria da Evidência" possui vantagens "definitivas" em ambientes vagos ou desconhecidos.

Desvantagens:

- Assume que as evidências são estatisticamente independentes umas das outras, o que dificulta a detecção de ataques simultâneos.
- A complexidade computacional é exponencial em relação ao conjunto de estados modelados. Os autores argumentam que para muitas aplicações práticas com poucos elementos focais a estratégia da força bruta é factível. Por exemplo, para o trabalho proposto o conjunto de estados é formado por {Normal, SYN-Flood, UDP-Flood, ICMP-Flood}

ANOMALY DETECTION OF WEB-BASED ATTACKS [41]

Os autores apresentam um IDS que usa um conjunto de diferentes técnicas de detecção de anomalias para detectar ataques contra servidores e aplicações baseadas na WEB. Discutem a diferença e citam vários artigos relacionados à *anomaly detection* [43,42,44] e *misuse detection* [39,52,51].

Citam que várias técnicas foram desenvolvidas para analisar evidências em um conjunto de dados tais como *data mining*, análise estatística e análise de seqüências de chamadas do sistema operacional. Destaca o trabalho de Lee et al. [45], e afirmam que a estratégia adotada é semelhante mas que o mecanismo de aprendizado do IDS proposto é diferente, uma vez que é puramente baseada em dados passados, como proposto em [46].

Os autores definiram um modelo de dados e um modelo de detecção. O modelo de dados é composto por logs do servidor Web Apache, onde são avaliadas as consultas realizadas e, mais especificamente, os atributos da consulta. O modelo de detecção é responsável por atribuir um valor probabilístico para cada consulta. O valor da probabilidade reflete a probabilidade da ocorrência daquela situação dado um perfil. A idéia é que eventos com probabilidades suficientemente baixas indicam potenciais ataques. A decisão é tomada calculando o peso da anomalia individualmente. Quando um ou mais pontuações de anomalia excede o limiar estabelecido durante uma fase de treinamento, a *query* é considerada anômala.

O modelo de detecção pode operar em um dos dois modos: Treinamento ou Detecção. A fase de treinamento é necessária para determinar as características de eventos normais e para estabelecer os limiares de distinção entre entradas normais e anômalas. Esta fase é dividida em dois passos. Primeiro, o sistema cria perfis para cada programa e seus atributos. Durante o segundo passo, os limiares são estabelecidos. No modo de detecção os graus de anomalia são calculados e alarmes são gerados.

O autor cita o uso de redes *bayesianas* para construir modelos de Markov [28]. Durante o treinamento o modelo de Markov é construído e durante a detecção ele é usado. Essa técnica é usada para inferir a estrutura dos parâmetros da query. Outras técnicas utilizadas, foram:

- Tamanho do Atributo: Durante o treinamento calcula-se a média e a variância do tamanho dos atributos. A Detecção é realizada usando a inequação de *Chebyshev* que define um

limite superior para a probabilidade da diferença entre o valor de uma variável randômica e a média dos valores excederem certo limiar.

- Distribuição dos caracteres do atributo: avalia a freqüência em que os caracteres aparecem nos atributos. Ordena-se as freqüências e usa como distribuição dos caracteres. Na detecção os atributos são subdividos em grupos de 6 [47] caracteres e usa-se o teste X^2 para avaliar.
- Localização de *tokens*: classifica-se o atributo como uma enumeração ou randômico. Usa-se variância e co-variância pra calcular a correlação estatística entre valores de atributos
- Presença ou ausência de atributo: cria-se o conjunto de valores que podem ou devem aparecer.
- Ordem dos atributos: define-se o conjunto de pares ordenados. Constrói-se um grafo direcionado onde os vértices são os atributos distintos.

A validação do IDS foi divida em duas partes: Validação do Modelo, onde foi verificado se o modelo é capaz de descrever as propriedades de interesse; Eficiência da Detecção, onde foi verificado o número de sucessos e falsos positivos.

A.2 Trabalhos estudados com o objetivo definir o Comportamento Normal da Rede e o Modelo de Falhas

Os estudos se concentraram em duas áreas de pesquisa: Simulação e Detecção de Falhas. Na área de Simulação vários modelos são usados para simular o comportamento normal da rede, desta forma os artigos apresentam modelos de rádio, consumo de energia, bateria, etc. Os artigos de Detecção de Falhas foram analisados porque normalmente assumem um modelo de falhas para avaliar a estratégia propostas e, tais modelos, podem auxiliar na definição do nosso modelo de falhas.

A.2.1 Artigos relacionados à Simulação

SENSORSIM: A SIMULATION FRAMEWORK FOR SENSOR NETWORKS [56]

Objetivo

Apresentar o SensorSim, um framework de simulação que introduz novos modelos e técnicas para o projeto e análise de RSSF.

Pontos Fortes/Interessantes

O SensorSim[21] é baseado no ns-2[44]. O trabalho estende o ns-2 para o contexto de RSSF provendo novos modelos de consumo de energia e protocolos de comunicação, suporte simulação híbrida e uma nova interface de usuário (GUI). Os autores criaram dois tipos de modelos. O primeiro, chamado de modelo funcional, representa a abstração do software do sensor o inclui todas as funcionalidades de software, tais como pilha de protocolos de rede, aplicações e pilha de protocolos de sensoriamento. As pilhas de protocolos de rede e de sensoriamento são coordenadas por camadas superiores denominadas middleware e camadas de aplicação. O segundo tipo de modelo, é o modelo de consumo de energia que simula a abstração de hardware (CPU, módulo de rádio, geofone e microfone).

Entre as novas funcionalidades introduzidas com o SensorSim estão a noção de "canal de sensoriamento" e a simulação híbrida. O canal de sensoriamento é visto como um meio através do qual os dispositivos de sensoriamento podem detectar eventos. O SensorSim suporta a funcionalidade chamada de simulação híbrida, onde o simulador suporta comunicação de entradas e saídas entre o ambiente simulado e aplicações externas.

O modelo de consumo de energia consiste em um simples provedor de energia e múltiplos consumidores. Atualmente, a bateria é o único provedor de energia com uma quantidade finita de energia armazenada. Os consumidores de energia são constituídos pelo módulo de rádio, CPU e outros vários dispositivos de sensoriamento incluindo o geofone, o detector de infravermelho, o microfone, etc. O modelo de consumo de energia foi dividido nos seguinte módulo: Modelo de Bateria, Modelo de Rádio, Modelo de CPU e Modelo de Dispositivo de Sensoriamento. Cada modelo é detalhado no trabalho, sendo que o modelo de bateria e o modelo de radio foram implementados e estão mais bem detalhados.

Obs.: Varshney e Bagrodia [55] criticam o modelo de bateria usado pelo SensorSim e propõe um modelo que parece ser mais real.

TOSSIM: ACCURATE AND SCALABLE SIMULATION OF ENTIRE TINYOS APPLICATIONS [57]

[21] Não está mais disponível para uso.

TOSSIM se propõe a capturar o comportamento da rede com grande fidelidade e ainda poder escalar para milhares de nós, através da emulação dos mica motes. O modelo de falha é composto por um modelo de erros probabilísticos para a rede[22], com o objetivo de manter o TOSSIM simples e eficiente mas expressivo o suficiente para capturar uma grande gama de interações de rede.

Segundo os autores, muitos pesquisadores concluíram ser intratável criar simuladores muito detalhados e ao mesmo tempo manter sua eficiência e têm empregado simulações muito mais abstratas para estudar o comportamento com muitos nós e simulações mais detalhadas são feitas com nós individuais. O ns-2 vem sendo adaptado para prover analogia com o comportamento das RSSF com o objetivo de avaliar o comportamento de protocolos contra cargas de trabalho sintéticas [65, 66, 67]. Os autores defendem que, normalmente, as representações de algoritmos avaliados através de simulações são muito diferentes das implementações reais. Embora as simulações sejam muito importantes, é essencial ter um veículo que permita estudar a implementação atual de algoritmos na escala de execução real das aplicações. O autor argumenta, que isto é especialmente importante em novas áreas onde não há anos de experiência, que permitam realizar a abstração de forma segura sem sacrificar a precisão. Além disso, se a simulação e o ambiente de implantação são muito diferentes, a implantação pode ser inibida: alguém tem que implementar os algoritmos duas vezes.

O artigo apresenta várias referências de sistemas [70, 71], protocolos [77, 78, 79, 80] e aplicações [72, 73, 74, 75] que vêm sendo construídas usando o TinyOS. O TinyOS é um sistema operacional projetado especificamente para RSSF. Possui um modelo de programação baseado em componentes, provido pela linguagem de programação nesse [91], um dialeto de C. O artigo resume as principais características desse sistema operacional, do hardware dos mica motes e da aplicação Surge.

Os autores citam o TinyDB [73] como um exemplo de aplicação onde os pesquisadores usaram o TOSSIM além de alguns bugs do TinyOS e do Surge que foram corrigidos com a ajuda do simulador.

Os autores citam o TOSSF [58] como uma alternativa ao TOSSIM e apontam suas vantagens e desvantagens.

[22] O modelo de erro é definido como um objeto do emulador, podendo ser facilmente trocado por outro objeto com outra distribuição de probabilidade. O TOSSIM já fornece dois modelos de erro, um chamado "simples" que considera que não há erros e outro que considera que os erros acontecem segunda uma distribuição de probabilidade gaussiana.

DETAILED MODELS FOR SENSOR NETWORK SIMULATIONS AND THEIR IMPACT ON NETWORK PERFORMANCE [55]

Objetivo/Contribuição

Descrever o projeto e arquitetura de um simulador de RSSF que provê uma rica suíte de modelos: modelo de pilha de sensoriamento com canais de sensoriamento baseado em ondas e difusão, modelo de bateria apurado, modelo de consumo de energia do processador, modelo de consumo de energia e modelo de tráfego baseado em RSSF. Foi objetivo do trabalho, também, mostrar o impacto de modelos detalhados na predição do desempenho da rede quando estes modelos são usados.

Pontos Fortes/Interessantes

Os autores classificam as tendências em simulação de RSSF entre estratégias menos flexíveis e mais precisas baseadas em emulação e estratégias mais genéricas e menos detalhadas baseada no modelo de simulação de rede. Além disso, definem que os trabalhos relacionados à simulação podem ser categorizados em quatro seções: uso de simuladores de redes puros, extensões de simuladores de rede, construção de simuladores a partir do zero e emulação do hardware do sensor.

Os simuladores normalmente utilizados são o NS-2[64], o GloMoSim [59] e seu sucessor Qualnet [60], etc. Estes simuladores foram enriquecidos com protocolos específicos de RSSF como o MAC [80] e de rede [81], mas eles não modelam nenhum outro aspecto dos nós sensores.

O SensorSim [56] estende o ns-2 com modelos de canais de sensoriamento, modelo de bateria e consumo de energia. Cada nó possui uma pilha de sensoriamento que atua como um sorvedouro de sinais nos canais dos nós sensores ou podem gerar estes sinais. Os autores afirmam que estenderam as funcionalidades do SensorSim, uma vez que incluíram o modelo de difusão de canais de sensoriamento, análise de código de execução e um modelo de bateria mais sofisticado.

Os autores afirmam que a emulação restringe a escalabilidade da simulação e restringe o simulador a uma arquitetura específica. Descrevem os vários modelos que foram adicionados no simulador de rede Qualnet, divididos em cinco categorias:

- Canais e Pilha de Sensoriamento: implementaram modelos de sensoriamento baseados em ondas e difusão. Para ondas usaram estratégia semelhante ao modelo de propagação de

rádio do Qualnet. Para difusão usaram a lei de Fick para gradiente de temperatura e a lei de Fourrier para o gradiente de concentração[23].

- Modelo de bateria: lembraram que a vida real de uma bateria mostra um comportamento de descarga não-linear e efeitos de recuperação [61] e propõe um modelo baseado em [62] que leva esses fatos em consideração.
- Modelo de Processador
- Modelo de Consumo de Energia: para cada tipo de hardware definiram um conjunto de estados e uma tabela que mapeia o estado do dispositivo com o que ele gasta naquele estado
- Pilha de protocolos sem fio e aplicações, os autores consideram uma rede dirigida a eventos e fazem duas definições: 1) a fonte é um grupo de nós que estão próximos entre si. O tamanho do grupo depende da densidade dos nós e da área na qual o efeito sensoriado é observado; 2) todas as fontes em um grupo conectam-se ao mesmo destino, diferentes grupos podem se conectar a diferentes estações.

Os autores finalizam o artigo apresentando vários experimentos que procuram mostrar a importância de se utilizar modelos mais precisos.

A.2.2 Artigos relacionados à Detecção ou Tolerância a Falhas

EFFICIENT TRACING OF FAIELED NODES IN SENSOR NETWORKS [68]

Objetivo

Demonstrar que a topologia da rede pode ser descoberta de forma eficiente pela estação base permitindo uma rápida determinação de nós defeituosos com um moderado custo na comunicação. Segundo os autores, os algoritmos propostos trabalham em conjunto com as funções normais da rede, de forma que não são necessárias mensagens adicionais.

Pontos Fortes/Interessantes

Os autores lembram que nós sensores podem falhar por diferentes razões: a bateria pode acabar, podem ser destruídos acidentalmente, um adversário pode inutilizá-lo, etc. Fazendo-se necessária a traceabilidade dos nós defeituosos.

[23] Não explicaram exatamente porque e não entraram em detalhes de como foi feito.

Os autores alegam que movendo o trabalho para a estação base, os nós defeituosos podem ser identificados rapidamente e de maneira eficiente usando a visão compreensiva da estação base sobre a rede e sem nenhuma mensagem adicional sendo transmitida pelos nós sensores.

A idéia é que os nós sensores repassem para a estação base a informação sobre quem são seus vizinhos e a estação base monta a topologia da rede. Os autores afirmam que em muitos protocolos de descobrimento de rotas [83,9] os nós já conhecem a identidade de seus vizinhos e para repassar essa informação para a estação base basta anexar um byte de informação sobre seus vizinhos para cada uma das medidas realizadas. Desta forma, depois de certo tempo, a estação base terá informação de adjacência de toda a rede e com algum algoritmo de descoberta de rotas [85], poderá construir sua topologia. Uma vez que a estação base conhece a topologia os nós defeituosos poderão ser traçados.

Os algoritmos propostos realizam mudanças na topologia de roteamento para determinar o estado de nós silenciosos. As rotas são reconstruídas e nós que permanecem silenciosos são considerados mortos. Modelo de rede considerado[24]:

- Uma estação base e n nós restritos [92]
- A estação base é capaz de se comunicar diretamente com os nós mas não o contrário, como em [83, 9]
- Os nós são capazes de enviar mensagens a longas distâncias mas ao custo de gastar mais energia
- Redes contínuas
- Nós podem ser mortos ou injetados na rede
- Nós são suficientemente resistentes a violação
- Será implementado um esquema de criptografia simétrica onde os nós e mensagens são autenticados através de uma chave privada compartilhada com a estação base.

Os autores definem um nó como “morto” se ele tiver expirado. Um nó é dito “silencioso” se não envia medidas de sensoriamento e seu estado não pode ser traçado. Um nó é dito “traçado” quando seu estado é determinado.

[24] Assume premissas muito fortes: nós são resistentes a violação, estação base pode se comunicar com todos os nós, nós podem enviar mensagens em longas distâncias