UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE
ÁREA DE CIÊNCIA DA COMPUTAÇÃO

RAYANE MENESES DA SILVA

UMA ONTOLOGIA DE APLICAÇÃO PARA APOIO À TOMADA DE DECISÕES EM SITUAÇÕES DE AMEAÇA À SEGURANÇA DA INFORMAÇÃO

São Luís
2015

RAYANE MENESES DA SILVA

UMA ONTOLOGIA DE APLICAÇÃO PARA APOIO À TOMADA DE DECISÕES EM SITUAÇÕES DE AMEAÇA À SEGURANÇA DA INFORMAÇÃO

Dissertação de Mestrado apresentada ao curso de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão, como parte dos requisitos para a obtenção do título de Mestre em Engenharia de Eletricidade, na área de Ciência da Computação.

Orientadora*:* Profa. Drª Rosario Girardi

São Luís
2015

Silva, Rayane Meneses da

Uma ontologia de aplicação para apoio à tomada de decisões em situações de ameaça à segurança da informação / Rayane Meneses da Silva. – São Luís, 2015.

82f.

Impresso por computador (fotocópia).

Orientadora: María del Rosario Girardi Gutiérrez.

Dissertação (Mestrado) – Universidade Federal do Maranhão, Programa de Pós-graduação em Engenharia de Eletricidade. São Luís, 2015.

1. Ontologias 2. Informação-Segurança. 3. Sistemas de detecção de Intrusão. 4. Rede-Ataque. I Título.

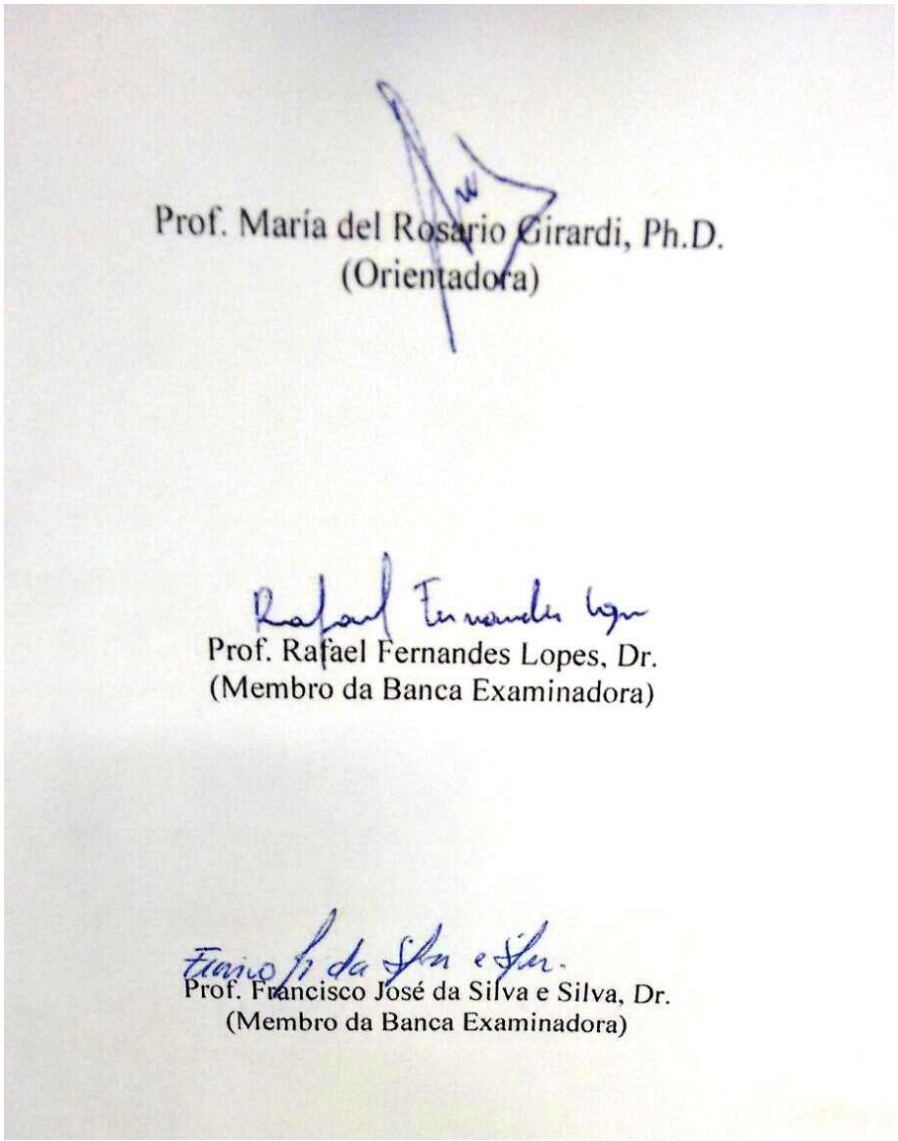
CDU 004.82

UMA ONTOLOGIA DE APLICAÇÃO PARA APOIO À TOMADA DE DECISÕES EM SITUAÇÕES DE AMEAÇA À SEGURANÇA DA INFORMAÇÃO

RAYANE MENESES DA SILVA

Dissertação aprovada em 24 de junho de 2015.

BANCA EXAMINADORA:



À minha querida mãe

AGRADECIMENTOS

Agradeço a Deus em primeira instância por realizar Sua boa vontade em minha vida, pela misericórdia e amor incondicionais e por me dar forças para lutar sempre, mesmo quando tudo parece dizer que Não.

À minha amada e muito querida mãe pela melhor amizade que posso ter, pelo incentivo, pelo imenso amor e por “tocar foguetes” em todas as minhas conquistas e vitórias, mesmo sendo elas tão pequenas.

Aos meus Avós (pais), pelo carinho, pela torcida e pela criação.

À minha irmã pela torcida, companhia e amizade.

Ao meu padrasto por ter me sustentado para que eu tivesse a oportunidade de estudar.

Aos meus tios e tias em especial à minha tia Glória que sempre me aconselhou, me acolheu em sua casa nos momentos em que precisei me isolar para estudar e por se alegrar com minhas vitórias.

Aos meus primos e primas em especial a minha pequena Rebeca, que alegra todos os meus dias com seu carinho e com sua alegria, e a Rafaela com seu esposo Enoc que também me acolheram em sua casa por alguns tempos para que eu pudesse me concentrar nos estudos.

Aos meus amigos do ensino médio com os quais descobri que podia chegar ao mais alto que eu quisesse independente da minha posição social.

Aos meus amigos mais chegados que irmãos Nayana Matos, Rayze Priscila, Juliane Evelyn, João Vieira, Anderson Henrique e Flávio Santos por torcerem sempre por mim, pela força em momentos difíceis e por não medirem esforços para cumprirem com seu papel de amigos.

Aos colegas do programa de pós graduação Suzane, Fhilipi, Ivo, Paulo, Messias, Thiago, Dhully, em especial a Raquel Machado pela companhia, amizade e ensinamentos, e a Adriana Leite, a pessoa que mais me ajudou, me ensinou e que

esteve sempre presente nos momentos de sucesso e de “fracasso” deste trabalho, me incentivando a continuar.

Aos professores do IFMA Monte Castelo Carla Marina, Marcelo Vidigal, Raimundo Osvaldo, Santiago, Eveline, Jeane, Raimundo Castro, Carla Faria e André Santos pelo incentivo, pelas orientações profissionais e estudantis e pela atenção quando precisei.

Aos professores e ex professores de matemática da UFMA Lívia Minami, Fabiano Borges, Maxuel Mariano e Nivaldo Muniz, pelos bons ensinamentos e incentivo.

Ao ex professor e hoje grande amigo Jackson, o cara que me motivou ainda mais a estudar e seguir em frente sem medo.

Aos Colegas dos cursos de Sistemas de Informação e técnico em informática pelos momentos de diversão, pela compreensão e pelo carinho, em especial a Ana Elisa e Adriano Escorcio.

Aos mais novos amigos e companheiros de trabalho Flávio, Vangleis, Robinson, Salvino e Lewi pela amizade e compreensão.

À professora Dra. Rosário Girardi, pela orientação, apoio intelectual, dedicação e paciência, as quais foram fundamentais para realização desta dissertação.

Aos professores Francisco José e Rafael Fernandes por suas sugestões e correções que contribuíram para melhorar a qualidade deste trabalho.

Ao programa de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão, pela oportunidade de realização do curso.

Por fim, a todos que contribuíram direta e indiretamente para a conclusão deste trabalho.

Sábio é o ser humano que tem coragem de ir diante do espelho da sua alma para reconhecer seus erros e fracassos e utilizá-los para plantar as mais belas sementes no terreno de sua inteligência.

Augusto Cury

RESUMO

Muitos mecanismos de segurança, como os Sistemas de Detecção de Intrusão têm sido desenvolvidos para abordar o problema de ataques à Segurança da Informação. Porém, a maioria deles são sistemas de informação tradicionais nos quais seus repositórios de ameaças não são representados semanticamente. As ontologias são estruturas de representação do conhecimento que permitem o processamento semântico das informações bem como a construção dos sistemas baseados em conhecimento, os quais fornecem uma maior efetividade em relação aos sistemas tradicionais. Neste trabalho propõe-se uma ontologia de aplicação denominada "*Application Ontology for the Development of Case-based Intrusion Detection Systems"* que representa formalmente os conceitos relacionados ao domínio de Segurança da Informação, dos sistemas de detecção de intrusão e do "*Case-Based Reasoning"*. O "*Case-Based Reasoning"* é uma abordagem para resolução de problemas nos quais é possível reutilizar conhecimentos de experiências passadas para resolver novos problemas. A avaliação da ontologia foi realizada por meio do desenvolvimento de um Sistema de Detecção de Intrusão que permite detectar ataques a redes de computadores e recomendar soluções a esses ataques. A ontologia foi especificada na linguagem "*Ontology Web Language"* utilizando o editor de ontologias Protegé e, logo após, mapeada a uma base de casos em Prolog utilizando o ferramenta "*Thea"*. Os resultados mostraram que o Sistema de Detecção de Intrusão desenvolvido apresentou boa efetividade na detecção de ataques e portanto, conclui-se que a ontologia proposta conceitualiza de forma adequada os conceitos de domínio e tarefa abordados.

Palavras-chave: Ontologias; Segurança da Informação; Sistemas de Detecção de Intrusão; Ataques de rede.

ABSTRACT

Many security mechanisms, such as Intrusion Detection Systems (*IDSs*) have been developed to approach the problem of information security attacks but most of them are traditional information systems in which their threats repositories are not represented semantically. Ontologies are knowledge representation structures that enable semantic processing of information and the construction of knowledge-based systems, which provide greater effectiveness compared to traditional systems. This paper proposes an application ontology called "*Application Ontology for the Development of Case-based Intrusion Detection Systems"* that formally represents the concepts related to information security domain of intrusion detection systems and "*Case Based Reasoning"*. The "*Case Based Reasoning"* is an approach for problem solving in which you can reuse the knowledge of past experiences to solve new problems. The evaluation of the ontology was performed by the development of an Intrusion Detection System that can detect attacks on computer networks and recommend solutions to these attacks. The ontology was specified using the "*Ontology Web Language"* and the Protégé ontology editor and. It was also mapped to a cases base in Prolog using the "*Thea*" tool. The results have shown that the developed Intrusion Detection System presented a good effectiveness in detecting attacks that the proposed ontology conceptualizes adequately the domain concepts and tasks.

Keywords – Ontologies; Information Security; Intrusion Detection Systems; Network Attacks.

SUMÁRIO

LISTA DE SIGLAS

IDS	Intrusion Detection System
ONTOID	Application Ontology for the Development of Case-based Intrusion Detection Systems
CBR	Case-Based Reasoning
OWL	Ontology Web Language
NSL-KDD	Network Security Lab - Knowledge Discovery and Data Mining
DoS	Denial of Service
DDoS	Distributed Denial of Service
HIDS	Host Intrusion Detection System
NIDS	Network Intrusion Detection System
R2L	Root to Local
U2R	User to Root Attacks
DAML	DARPA Agent Markup Language
OIL	Ontology Inference Layer
DAMLJessKB	DARPA Agent Markup Language Java Expert System Shell Knowledge Base
TCP	User Control Protocol
UDP	User Datagram Protocol
CAPEC	Common Attack Pattern Enumeration and Classification
IP	Internet Protocol

SWRL	Semantic Web Rule Language
IDMEF	Intrusion Detection Message Exchange
OrBAC	Organization Based Access Control
MADAE-Pro	Multiagent Domain and Application Engineering Process
CBR-IDS	Case Based Reasoning Intrusion Detection System
LPA Win-Prolog	Logic Program Associates Win-Prolog
NCP	New Case Problem
KBCP	Knowledge Base Case Problem
BC	Base de Conhecimento
ICMP	Internet Control Message Protocol

LISTA DE TABELAS

LISTA DE FIGURAS

1 INTRODUÇÃO

A Segurança da Informação está relacionada diretamente com a proteção de um conjunto de informações, com o intuito de manter preservado o seu valor tanto para pessoas quanto para organizações. Os princípios básicos que devem ser considerados para manter essa proteção são a confidencialidade (garante a proteção das informações contra acesso e/ou divulgação não autorizada), a integridade (garante a proteção das informações contra a modificação não autorizada) e a disponibilidade (garante que o acesso a recursos da rede esteja disponível sempre que solicitada por entidades autorizadas).

Manipular, gerenciar, armazenar e manter as informações protegidas está se tornado uma tarefa cada vez mais complexa por causa do grande volume de informações e de seu alto valor, despertando o interesse de pessoas mal intencionadas e comprometendo a segurança destas informações.

Algumas dessas ameaças podem ser causadas de forma acidental, como uma falha de *software* ou de hardware, enquanto outras são provocadas de forma intencional por pessoas com diversos objetivos, que vão desde uma simples curiosidade até o interesse em obter ganhos financeiros, espionagem industrial, venda de informações confidenciais, entre outras razões. Essas ameaças intencionais são chamadas de ataques e são classificadas, de acordo com a ação que realizam, como ataques de interceptação, modificação ou interrupção. Muitos mecanismos de segurança têm sido desenvolvidos para mitigar esses problemas de ataques à Segurança da Informação. Dentre eles, merecem destaque os Sistemas de Detecção de Intrusão (*IDSs*) que são definidos, segundo Debar (2004), como “um mecanismo de segurança que monitora o tráfego de uma rede de computadores em busca de intrusões, atividades de usuários não autorizados e ocorrências de práticas mal intencionadas”. As principais abordagens utilizadas pelos IDSs para detecção de intrusões são a detecção baseada em assinaturas e a detecção baseada em anomalias.

A abordagem por assinatura é o processo de se comparar assinaturas com eventos observados para identificar possíveis ocorrências de incidentes (SCARFONE e MELL, 2007). Tais assinaturas são padrões de ataques pré-armazenados na base de dados do IDS. A abordagem por anomalia é o processo de

se comparar definições de atividades consideradas normais com eventos observados, para identificar desvios significantes (SCARFONE e MELL, 2007).

A maioria dos Sistemas de Detecção de Intrusão são sistemas de informação tradicionais nos quais seus repositórios de ameaças não são representados semanticamente e, a ontologia é uma abordagem que permite esse tipo de representação.

1.1 Justificativa

As ontologias são estruturas de representação do conhecimento que permitem uma representação das relações semânticas entre conceitos de um determinado domínio bem como a construção dos sistemas baseados em conhecimento, os quais fornecem uma maior efetividade em relação aos sistemas tradicionais. Elas possibilitam também a comunicação e a interoperabilidade entre agentes de *software* e agentes humanos sobre uma mesma representação de dados (GUALBERTO, 2012). As ontologias descrevem conceitos e relacionamentos relevantes para um determinado domínio, estabelecendo um vocabulário comum, além de fornecer uma especificação formal do significado dos termos utilizados nesse vocabulário (NOY e MCGUINNESS, 2001). Aplicando ontologias para a detecção de intrusão é possível expressar, além de uma taxonomia de ataques, os relacionamentos entre os dados coletados, e utilizá-los para deduzir que determinados dados caracterizam um ataque (UNDERCORFFER, JOSHI e PINKSTON, 2003). As principais vantagens de se utilizar ontologias para representar um domínio de conhecimento é o fato de elas proverem uma semântica explícita e formal, serem reutilizáveis, e pela sua capacidade de compartilhamento (ROSA, 2011).

O Raciocínio Baseado em Casos (*Case Base Reasonig* - *CBR*) é uma das tecnologias mais populares e disseminadas para o desenvolvimento de sistemas baseados em conhecimento (WANGENHEIM e WANGENHEIM, 2003); é “um paradigma de resolução de problemas no qual é possível utilizar conhecimentos de experiências passadas para resolver novas situações” (MENDES, GIRARDI e LEITE, 2013). Em uma abordagem *CBR*, “as ontologias permitem maior efetividade no processo de seleção de casos por causa da sua expressividade semântica” (MENDES, GIRARDI e LEITE, 2013).

Este trabalho propõe uma ontologia de aplicação chamada ONTOID ("*Application Ontology for the Development of Case-based Intrusion Detection Systems*") com domínio em Segurança da Informação para o desenvolvimento de um sistema de detecção de intrusão baseado em casos que detecte ataques e tome decisões sobre eles, fornecendo soluções ou possíveis soluções para os ataques detectados.

A ONTOID foi construída no editor de ontologias Protegé (STANFORD, 2014), representada utilizando a linguagem *OWL* ("*Ontology Web Language*") (MCGUINNESS e HARMELEN, 2004) e, para povoá-la com casos de ataques, foi utilizado o conjunto de dados NSL-KDD (*Network Security Lab - Knowledge Discovery and Data Mining*) (REVATHI e MALATHI, 2013). O protótipo de um IDS foi construído para realizar uma avaliação preliminar e experimental da ONTOID.

1.2 Objetivos

Objetivo Geral:

Contribuir com soluções, baseadas em ontologias e no paradigma *CBR*, para o problema de ameaças que comprometem a segurança dos sistemas de informação.

Objetivos Específicos:

- Analisar o estado da arte dos problemas de ataques à Segurança da Informação;
- Analisar o estado da arte dos sistemas de detecção de intrusão (IDS) e ontologias para o desenvolvimento de IDS;
- Construir uma ontologia de aplicação para representar os conceitos de domínio e tarefas na área de Segurança da Informação;
- Avaliar a efetividade da ontologia por meio de um estudo de caso no desenvolvimento do protótipo de um IDS, baseado em casos, capaz de apoiar o processo de tomada decisões em ataques à Segurança da Informação.

1.3 Estrutura

O presente trabalho está organizado em seis capítulos, incluindo esta introdução. O segundo trata da fundamentação teórica utilizada neste trabalho incluindo os conceitos de Segurança da Informação, ontologias e o conjunto de dados NSL-KDD utilizado no desenvolvimento do estudo de caso para avaliação da ontologia proposta. No terceiro capítulo, apresenta-se alguns trabalhos que abordam ontologias no domínio de Segurança da Informação. No quarto capítulo é apresentado o desenvolvimento da ONTOID, bem como a descrição de cada conceito representado nela e suas tarefas. O quinto capítulo apresenta um estudo de caso aplicado à IDSs para avaliação da ontologia proposta e, por fim, as conclusões sobre o trabalho realizado são apresentadas no sexto capítulo, destacando os resultados, as publicações e possíveis trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta o embasamento teórico necessário para que a proposta deste trabalho seja bem compreendida, ou seja, conceitos relevantes do domínio de Segurança da Informação e o conceito e classificação de ontologias são discutidos.

2.1 Segurança da Informação

Antes de definir Segurança da Informação é necessário conhecer a definição de informação. A informação é o resultado do processamento, manipulação e organização de dados que representa um acréscimo ao conhecimento da pessoa que o recebe (STRASSBUG et al., 2007), ou seja, é um conjunto de dados trabalhados, úteis com valor significativo atribuído a eles e com um sentido natural e lógico para quem a utiliza.

2.1.1 Definição

Segundo Associação Brasileira de Normas Técnicas - ABNT (2005) a Segurança da Informação é a proteção da informação de vários tipos de ameaças para garantir a continuidade do negócio, minimizar o risco ao negócio, maximizar o retorno sobre os investimentos e as oportunidades de negócio. ABNT (2005) apresenta também uma definição mais concisa para a Segurança da Informação: é a preservação da confidencialidade, integridade e disponibilidade da informação, conceitos que serão discutidos na Seção 2.1.3. Para Gualberto (2012), a Segurança da Informação diz respeito à proteção da informação contra ameaças que possam valer-se das vulnerabilidades de um ativo de informação, preservando-a assim suas propriedades.

2.1.2 Conceitos Básicos

Para melhor compreensão deste trabalho faz-se necessária a apresentação de alguns conceitos fundamentais inerentes à Segurança da Informação. A seguir, estão descritos os conceitos de ativo, vulnerabilidade, incidente, ameaças e ataques.

Ativo de informação, segundo ABNT (2005) é tudo que tenha valor para a organização. Como exemplos podemos citar os funcionários, computadores, aplicativos, banco de dados de uma organização, etc.

Vulnerabilidade é uma fragilidade de um ativo de informação, como um defeito ou problema presente na especificação, implementação, configuração ou operação de um *software* ou sistema, que possa ser explorado para violar as propriedades de segurança do mesmo (MAZIERO, 2013). Por exemplo, um erro de programação no serviço de compartilhamento de arquivos que permita a usuários não autorizados o acesso a outros arquivos do computador local, além daqueles compartilhados (MAZIERO, 2013).

Incidente de segurança é indicado por um simples ou por uma série de eventos de Segurança da Informação indesejados ou inesperados, que tenham uma grande probabilidade de comprometer as operações do negócio e ameaçar a Segurança da Informação (ABNT, 2005). Segundo Maziero (2013), é qualquer fato intencional ou acidental que comprometa um dos princípios de segurança do sistema. Ele cita como exemplo a intrusão de um sistema de negação de serviços e um vazamento acidental de informações.

2.1.2.1 Ameaças e Ataques

Uma ameaça consiste em uma possível violação de um sistema computacional e pode ser acidental ou intencional (PINHEIRO, 2007). As ameaças acidentais são aquelas ocorridas por causas naturais como uma falha de *software* ou de *hardware*, e as ameaças intencionais são aquelas provocadas por uma pessoa mal intencionada por diversos motivos, como uma simples curiosidade ou até mesmo o interesse em obter ganhos financeiros, espionagem industrial, venda de informações confidenciais, entre outras razões. Um ataque é uma ameaça intencional e eles são classificados, de acordo com a ação que realizam, como ataques de interceptação, modificação ou interrupção. As ameaças, tanto intencionais como não intencionais podem causar um impacto negativo sobre os princípios de Segurança da Informação, ou seja, uma ameaça pode deixar de assegurar a confidencialidade, integridade e disponibilidade das informações de um usuário (pessoa física ou organização). Stallings (2013) adota a perspectiva da

existência de um fluxo de informação entre a fonte e o destino, existindo vários tipos de ameaças e ataques: por interrupção do fluxo de informação, por intercepção, por modificação e por produção.

A Figura 2.1 é uma representação de como acontece o fluxo desses tipos de ameaças e ataques entre duas máquinas, a partir de uma máquina "atacante". A interrupção consiste em impedir o fluxo normal das informações ou acessos; a interceptação consiste em obter acesso indevido a um fluxo de informações, sem necessariamente modificá-las; modificação consiste em modificar de forma indevida informações ou partes do sistema; a produção consiste em produzir informações falsas ou introduzir módulos ou componentes maliciosos no sistema.

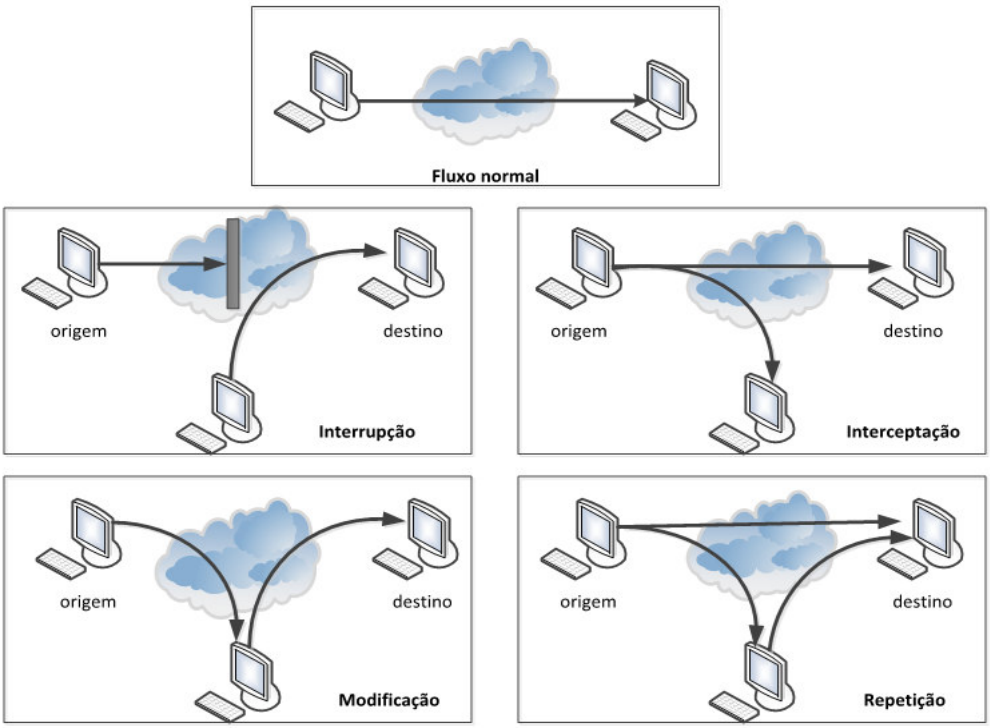


Figura 2.1 Ameaças à Segurança da Informação. Fonte: adaptado de (STALLINGS, 2010)

Esses tipos de ataques costumam acontecer na Internet, por meio de várias técnicas e por diversos motivos como, por exemplo, mostrar poder, vangloriar-se, aplicar golpes financeiros, tornar inacessíveis sites de concorrentes etc. Para alcançar seus objetivos, os atacantes se utilizam de técnicas como "*Exploits*", *"Scan"*, "*E-mail Spoofing"*, "*Sniffing*", "*Brute Force*", "*Defacement*" e "*DoS"* (*Denial of Service*) e "*DDoS*" (*Distributed Denial of Service*) que são descritas pela CERT.br (2012) a seguir:

"*Exploits*" – utilizando-se de uma vulnerabilidade, tenta executar ações maliciosas, como invadir um sistema, acessar informações confidenciais, disparar ataques contra outros computadores ou tornar um serviço inacessível;

"Scan" – é uma técnica que consiste em efetuar buscas minuciosas em redes com o objetivo de identificar computadores ativos e coletar informações sobre eles como, por exemplo, serviços disponibilizados e programas instalados;

"*E-mail Spoofing"* – técnica que consiste em alterar campos do cabeçalho de um e-mail, de forma a aparentar que ele foi enviado de uma determinada origem quando, na verdade, foi enviado de outra;

"*Sniffing*" – técnica que consiste em inspecionar os dados trafegados em redes de computadores, por meio do uso de programas específicos chamados de *sniffers*;

"*Brute Force*" – técnica que consiste em adivinhar, por tentativa e erro, um nome de usuário e senha e, assim executar processos e acessar sites, computadores e serviços em nome e com os mesmos privilégios deste usuário;

"*Defacement*" – técnica que consiste em alterar o conteúdo de uma página Web;

"*DoS"* - técnica conhecida como negação de serviços, a qual um atacante utiliza um computador para tirar de operação um serviço, um computador ou uma rede conectada à Internet. Quando utilizada de forma coordenada e distribuída, ou seja, quando um conjunto de computadores é utilizado no ataque, recebe o nome de negação de serviço distribuído, ou *DDoS* (*Distributed Denial of Service*).

2.1.3 Princípios de Segurança da Informação

A Segurança da Informação protege a informação de diversos tipos de ameaças, garantindo a continuidade do negócio, minimizando os danos, e também, o tempo do retorno do negócio da organização (LIMA, 2006). Ela busca reduzir os riscos de vazamento, fraudes, erros, uso indevido, roubo de informações, sabotagens, entre outras ameaças.

Um usuário espera que suas informações estejam sempre disponíveis no momento em que desejar, que estas sejam confiáveis, verdadeiras e que pessoas não autorizadas não tenham acesso ao conteúdo de suas informações. Para diminuir os riscos citados anteriormente e satisfazer as expectativas dos usuários, o padrão da ANBT (2005) se apoia nos seguintes princípios básicos:

Confidencialidade – é a proteção dos dados contra a divulgação não autorizada (TELES, 2012). Ela consiste em garantir que a informação não seja acessada, copiada e/ou divulgada por indivíduos que não tenham sido autorizados pelo proprietário dela. Há perda de confidencialidade quando há quebra de sigilo de uma determinada informação (ex: quando uma pessoa não autorizada obtém a senha de um usuário ou administrador de sistema). Ataques de "*phishing"* são um exemplo comum de problema que atinge a confidencialidade das informações pois, o "*phishing"* é um tipo de fraude por meio do qual um golpista tenta obter dados pessoais e financeiros de um usuário pela utilização combinada de meios técnicos e engenharia social (CERT.br, 2012).

Integridade – é a proteção dos dados contra a modificação não autorizada (TELES 2012). Consiste em garantir a veracidade do conteúdo da mensagem, ou seja, que não tenha sido alterada por pessoas não autorizadas ou violada de forma indevida. Há perda de integridade quando a informação fica exposta a manuseio por uma pessoa não autorizada, que a modifica de forma não aprovada pelo proprietário dela. O "*Defacement*" é um exemplo de ataque à integridade, pois é uma técnica que consiste em alterar o conteúdo de uma página na Internet.

Disponibilidade – é a proteção de acesso aos recursos da rede para que estejam disponíveis quando requisitados pelas entidades autorizadas (TELES, 2012). Ela consiste em garantir que os serviços prestados pelo sistema não sejam degradados e/ou interrompidos de maneira que permitam ao usuário ter acesso aos dados sempre que precisar. Há perda de disponibilidade quando a informação deixa de estar acessível por usuários que precisam dela. Um exemplo de ataque contra a disponibilidade das informações é o de "*DoS"* (negação de serviços) que são tentativas de fazer com que computadores como servidores web, por exemplo, tenham dificuldade ou sejam impedidos de executar sua tarefa.

2.1.4 Softwares Maliciosos

Como o próprio nome já diz, "*malwares"* são programas maliciosos criados para diversos fins: invadir um computador ou sistema, causar danos e apagar dados, roubar informações, divulgar serviços, etc. A palavra "*malware*" engloba programas perigosos, invasivos e mal intencionados que podem atingir um computador, bem como as informações nele contidas. Em quase todos os casos eles agem de forma que os usuários nem percebam. Existem vários tipos de "*malwares"*, a seguir são descritos alguns deles.

"Vírus": é um trecho de código que se infiltra em programas executáveis existentes no sistema operacional, usando-os como suporte para sua execução e replicação (MAZIERO, 2013). Eles podem apagar dados, capturar informações e alterar o funcionamento normal do computador e se diferenciam dos outros "*malwares"* por sua capacidade de infectar um sistema, fazer cópias de si mesmo em vários computadores em uma rede, modificar objetos de sistema no disco ou memória, corrompendo o funcionamento normal do sistema operacional. O principal foco dos "*vírus*" são os sistemas operacionais *Windows*, por serem os mais utilizados no mundo inteiro.

"Worms": são programas capazes de se propagar automaticamente pelas redes, enviando cópias, de si mesmo, de computador para computador (CERT.br, 2012) são programas maliciosos que utilizam uma rede de Internet para se espalhar por vários computadores sem precisar da interferência de um usuário. Eles podem infectar o computador de maneira totalmente discreta, explorando falhas em aplicativos ou no próprio sistema operacional. São perigosos, pois podem ser disparados, aplicados e espalhados em um processo totalmente automático sem precisar se anexar a nenhum arquivo. Enquanto o vírus busca modificar e corromper arquivos, os "*worms"* costumam consumir banda em uma rede.

"Bot" ou "Botnet": assim como o "*worm"* é um programa capaz se propagar automaticamente, explorando vulnerabilidades existentes ou falhas na configuração de *softwares* instalados em um computador (CERT.br, 2012), porém com finalidade diferente. O "*bot"* é um código que dispõe de mecanismos de comunicação com o atacante, permitindo que uma máquina infectada seja acessada remotamente. Por

este motivo, um computador infectado por "*bot"* costuma ser chamado de zumbi ("*zombie computer"*). "*Botnet"* é uma rede de computadores formada por "*zombie computers"* e que permite potencializar as ações danosas executadas pelos "*bots"* (CERT.br, 2012).

"Trojan" (cavalo de Tróia): assim como os "*vírus*", os "*trojans*" causam perda de arquivos, falhas na memória e erros em periféricos, porém o *"trojan*" pode ser considerado um "*vírus*" inteligente pois é controlado à distância pela pessoa que o instalou (ASSUNÇÃO, 2002). Os *"trojans*" atuais são disfarçados de programas legítimos (por exemplo, um jogo) embora, diferentemente dos *"vírus*" e *"worms"*, não criam réplicas de si (COZZOLINO, 2012). Cada *"trojan*" entra no computador e libera uma porta para uma possível invasão, permitindo um acesso remoto à máquina. Ele pode roubar dados do usuário, executar instruções que ordenam uma exclusão de arquivos, destruição de aplicativos, etc.

2.1.5 Mecanismos de Segurança

Para garantir os princípios de segurança e evitar que *softwares* maliciosos invadam um computador, foram desenvolvidos mecanismos de segurança que quando configurados e utilizados corretamente podem auxiliar uma pessoa a se proteger dos riscos envolvendo o uso da Internet. Abaixo tem-se alguns mecanismos citados pelo CERT.br (2012):

Contas e senhas: mecanismo de autenticação usado para controle e acesso do usuário a sites oferecidos pela Internet. Por meio deste mecanismo os sistemas conseguem identificar quem é o usuário e definir as ações que pode realizar.

Criptografia: mecanismo no qual uma informação é transformada de sua forma original para uma outra ilegível por meio de uma chave secreta e, só quem a possui é que pode ler a informação correta. É usado para proteger os dados contra acesso por pessoas não autorizadas.

"Backups": mecanismo que permite fazer cópias dos arquivos armazenados no computador para evitar perda de informação nos casos em que o disco rígido for infectado ou danificado, ou mesmo em situações que ele precise ser formatado.

Ferramentas "*antimalware"*: são aquelas que procuram detectar e então, anular ou remover os códigos maliciosos de um computador (CERT.br, 2012). Como exemplo tem-se o "*antivírus", "antispyware"* e o *"antispam"*.

"Firewall" – uma combinação de hardware e *software* que isola a rede de uma organização da Internet, bloqueando alguns pacotes e permitindo a passagem de outros (KUROSE, 2008). Ele analisa o tráfego em uma rede, dando permissão ou não para a passagem de dados a partir de um conjunto de regras. O "*firewall"* garante o controle da conexão entre duas ou mais redes, filtra os acessos feitos da empresa para a Internet e da Internet para a empresa e protege a rede interna de ataques externos.

IDS ("*Intrusion Detection System"*) – dispositivo que gera alertas quando observa tráfegos potencialmente mal intencionados (KUROSE,2008). Ele pode ser usado para detectar diversos tipos de ataques. Na Seção 2.1.7 será explorado a respeito desse tipo de mecanismo.

2.1.6 Fases da segurança em redes de computadores

Segundo Zseby, Pfeffer e Steglich (2009) é possível destacar quatro fases para a proteção contra ataques em rede de computadores: Prevenção, Detecção, Defesa e Forense conforme ilustra a Figura 2.2.

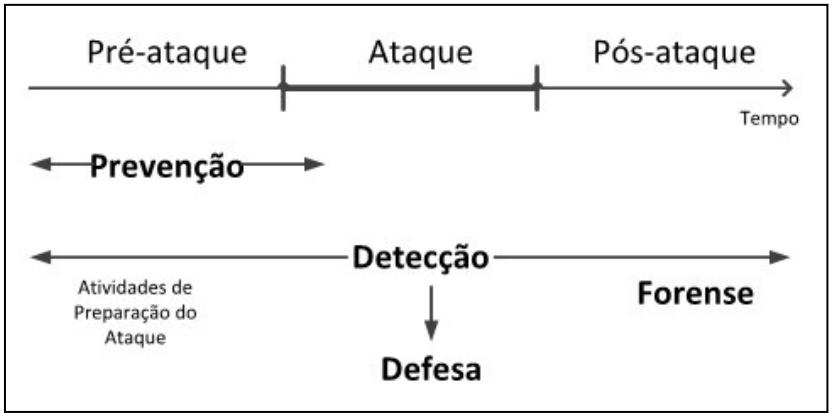


Figura 2.2 Quatro fases da proteção contra ataques. Fonte: (ZSEBY, PFEFFER e STEGLICH, 2009)

Os mecanismos de Prevenção são responsáveis pela primeira fase (Pré-ataque) na segurança de uma rede de computadores. São tipicamente implantados para controlar acesso a recursos e à informação que transita na rede (TELES,

2012). Tais mecanismos são conhecidos como *IPS* ("*Intrusion Prevention System*"). A próxima etapa de proteção contra incidentes, utilizada nos casos em que a prevenção falha, é a Detecção. Ela é o processo de descoberta de um ataque ou da preparação para uma realização, ou qualquer outra atividade maliciosa, desde um "*port scan*" até indícios de quebra de senhas por técnica de "*brute force*" (TELES, 2012). A fase de Defesa inicia-se após a detecção o de um tráfego malicioso. Para que essa fase seja iniciada é necessário que um sistema implemente as fases de Detecção e Defesa. Se um ataque for bem sucedido, significa que as três fases anteriores falharam. Para abordar esse problema tem-se a próxima fase chamada Forense que, tem como objetivo realizar uma investigação para descobrir detalhes específicos de ataques, apresentando resultados que contribuam de alguma maneira para a melhoria da proteção da rede (TELES, 2012).

2.1.7 IDS - Sistemas de Detecção de Intrusão

Intrusões são violações ou tópicos iminentes de violações das políticas de segurança de computadores (SAWAT e ITKAR, 2014). Detecção de intrusão é o processo de se monitorar os eventos ocorrendo em um sistema computacional ou em uma rede para identificar indivíduos que estejam utilizando estes recursos sem autorização ou que estejam excedendo seus privilégios (SNAPP et al., 1991). Um sistema de detecção de Intrusão ("*Intrusion Detection System* – IDS") é uma ferramenta de segurança que monitora o tráfego de uma rede em busca de sinais de intrusão, atividades de usuários não autorizados e a ocorrência de práticas mal intencionadas (DEBAR, 2004).

O IDS é um sistema de segurança voltado para lidar com as invasões no ambiente do computador e alertar os administradores da rede para que eles possam tomar as ações preventivas e corretivas para evitar a intrusão. Ele não só alerta os administradores da rede como também pode reagir às invasões.

Sawant e Itkar (2014) destacam as principais funções de um sistema de detecção de intrusão:

- Examinar o tráfego de uma rede;
- Identificar possíveis eventos, monitorando o usuário e o sistema;
- Registrar informações sobre o usuário do sistema;

- Analisar configurações e vulnerabilidades do sistema;
- Avaliar a integridade dos arquivos do sistema;
- Reconhecer atividades e padrões típicos de ataques;
- Enviar um alerta para o administrador de segurança.

Um IDS pode ser de dois tipos: IDS de *Host* ("*Host Intrusion Detection System – HIDS"*) ou IDS de Rede ("*Network Intrusion Detection System – NIDS"*), ambos podem ser desenvolvidos segundo duas abordagens: IDS baseado em assinaturas (conhecimento prévio dos ataques e suas características) ou IDS baseado em anomalia (comportamento da rede).

Um HIDS analisa o tráfego sobre um servidor ou uma estação qualquer e o NIDS analisa o tráfego de uma rede. Um HIDS monitora o tráfego (somente para um host onde está instalado), logs de sistemas, modificação de acesso de arquivos, modificações de configurações do sistema, processos em execução e atividades de aplicações. É instalado em dispositivos conectados a uma rede como servidores e computadores pessoais. Um NIDS é geralmente instalado em uma fronteira entre redes e é muito útil para detectar acessos não autorizados externos à rede, ataques "DoS" (ataques que visam destruir mensagens ou tornar sistemas inativos ou inutilizáveis) e aumentos significativos no consumo de banda.

Um IDS baseado em assinaturas mantém uma base de dados de assinaturas de ataques conhecidos. Cada assinatura é um conjunto de regras relacionadas a um evento de ataque observado na rede (ex: número de portas de origem e destino, tipo de protocolo, e uma sequência de bits em uma carga útil de um pacote. Analisa os pacotes que passam pela rede comparando-os com as assinaturas da base de dados e gerando um alerta ao administrador da rede caso as assinaturas sejam compatíveis. Estes IDSs são muito eficientes para detectar ameaças conhecidas, porém não detectam novas ameaças nem variações de ameaças conhecidas. Possuem baixa taxa de falsos positivos, ou seja, ao detectar um ataque, a possibilidade de não ser um ataque é muito baixa, porém ainda é muito suscetível à detecção de falsos negativos, ou seja, pode deixar de detectar alguns ataques caso a base não esteja atualizada.

Um *IDS* baseado em anomalias analisa o comportamento da rede e cria um perfil do tráfego normal dela. Procura por cadeias de pacotes que são estatisticamente incomuns, por exemplo, uma porcentagem irregular de pacotes *ICMP* ou um crescimento exponencial de "*scanners"* de portas e varreduras de "*ping"* (KUROSE, 2008). Estes IDSs possuem grande potencial para detectar novos ataques porém, a sua taxa de falsos positivos é muito alta, ou seja, pode detectar muitos comportamentos que são normais, como um ataque.

2.2 NSL-KDD Dataset

O NSL-KDD (REVATHI e MALATHI, 2013) é a versão atualizada do KDD CUP 99 (TAVALLAEE et al., 2009) o qual é um conjunto de dados inclui os dados de uma sessão de conexão a uma determinada rede e a correspondente classificação desses dados em um determinado tipo de ataque. Ele é composto por pacotes de dados (em conexões de rede) capturados a partir da simulação de uma rede de computadores que representava uma rede típica encontrada em agências governamentais norte-americanas, durante nove semanas no ano de 1998 em que, cada conexão é caracterizada por 41 atributos com informações (valores) que a identificam como uma conexão "ruim" ou "boa", ou seja, um ataque ou tráfego normal da rede, respectivamente. Os ataques, nesse conjunto de dados, são divididos em quatro categorias segundo Tavallaee et al. (2009), descritas a seguir:

- *"DoS Attacks"* – ataques em que o atacante faz com que algum recurso de computação ou de memória fique muito ocupado ou muito cheio de requisições/solicitações para lidar com as solicitações legítimas, ou nega, a usuários legítimos, o acesso a uma máquina;

- *"Probe"* – ataques em que o atacante começa com o acesso a uma conta de usuário normal no sistema (talvez ganho por meio de senhas "*sniffing"* ou engenharia social) e é capaz de explorar alguma vulnerabilidade para obter acesso root ao sistema;

- *R2L* ("*Root to Local"*) – ocorre quando um atacante que tem a capacidade de enviar pacotes para uma máquina em uma rede, mas que não tem uma conta

nessa máquina explora alguma vulnerabilidade para obter acesso local como um usuário da máquina;

- *U2R* ("*User to Root Attacks*") – tentativa de reunir informações sobre uma rede de computadores com o objetivo aparente de burlar os controles de segurança.

Na Tabela 2.1 tem-se as características de uma conexão de rede, bem como a descrição de cada uma. As características representadas como discretas significa que o atributo pode ser representado por um único valor para cada caso, e as características tidas como contínuas significa que são representadas por faixa de valores.

Tabela 2.1 Características de uma conexão de rede. Fonte: (SILVA e MAIA, 2004)

Recurso	**Descrição**	**Tipo**
duration	Duração em segundos da conexão	contínuo
protocol_type	Tipo de protocolo usado na conexão, (ex: tcp, udp, etc.)	discreto
service	Serviço de rede sendo utilizado, (ex: http, telnet, ftp etc.)	discreto
src_bytes	Número de bytes enviados da fonte para o destino	contínuo
dst_bytes	Número de bytes enviados do destino para a fonte	contínuo
flag	Status da conexão (normal ou erro)	discreto
land	1 se conexão é de/para o mesmo host; 0 caso contrário.	discreto
wrong_frament	Número de fragmentos com erro.	contínuo
urgent	Número de pacotes com flag urgente habilitado.	contínuo
hot	Número de indicadores chaves ("hot").	contínuo
nun_failed_logins	Tentativas de login sem sucesso	contínuo
logged_in	1 se login efetuado com sucesso; 0 caso contrário.	discreto
num_compromised	Número de condições de "comprometimento"	contínuo
root_shell	1 se shell root foi obtido; 0 caso contrário	discreto
su_attempted	1 se comando "su root" foi tentado; 0 caso contrário	discreto
num_root	Número de acessos como root.	contínuo
num_file_creations	Número de operações de criação de arquivos.	contínuo
num_shells	Número de "shells prompts" obtidos.	contínuo
num_acess_files	Número de operações em arquivos de controle de acesso.	contínuo
num_outbount_cmds	Número de comandos externos em uma sessão ftp.	contínuo
is_hot_login	1 se o login pertence a lista "hot"; 0 caso contrário.	discreto
is_guest_login	1 se o login usou a conta guest; 0 caso contrário.	discreto
count	Número de conexões iguais a esta para este mesmo "host" nos últimos 2 segundos.	contínuo
srv_count	Número de conexões para o mesmo serviço que é usado nesta conexão nos últimos 2 segundos.	contínuo
serror_rate	% de conexões que possuem erro SYN.	contínuo
srv_serror_rate	% de conexões que possuem erro SYN para este serviço.	contínuo
rerror_rate	% de conexões que possuem erro REJ.	contínuo

srv_rerror_rate	% de conexões que possuem erro REJ para este serviço.	contínuo
same_srv_rate	% de conexões para um mesmo serviço.	contínuo
diff_srv_rate	% de conexões para serviços diferentes.	contínuo
srv_diff_host_rate	% de conexões deste mesmo serviço para hosts diferentes.	contínuo
dst_host_count	Número de conexões com mesmo host de destino que esta.	contínuo
dst_host_srv_count	Número de conexões com mesmo host de destino e mesmo serviço que esta.	contínuo
dst_host_same_srv_count	% de conexões com o mesmo host de destino e mesmo serviço que esta.	contínuo
dst_host_diff_srv_rate	% de conexões com o mesmo host de destino e services diferentes que esta.	contínuo
dst_host_same_src_port_rate	% de conexões com o mesmo host de destino e a mesma porta de origem que a conexão atual.	contínuo
dst_host_srv_diff_host_rate	% de conexões para o mesmo serviço vindo de diferentes hosts.	contínuo
dst_host_serror_rate	% de conexões para o mesmo host que o da conexão atual e que possui um erro S0.	contínuo
dst_host_srv_serror_rate	% de conexões para o mesmo host e serviço que o da conexão atual e que possui um erro S0.	contínuo
dst_host_rerror_rate	% de conexões para o mesmo host que apresentem flag RST.	contínuo
dst_host_srv_rerror_rate	% de conexões para o mesmo host e serviço da conexão atual que apresentem flag RST.	contínua

A Figura 2.3 mostra a representação de parte do conjunto de dados NSL-KDD, em que cada linha representa uma sessão de conexão de rede e as características dessas conexões, ou seja, os valores associados a cada característica/atributo separados por "," (vírgula), de acordo com as características descritas na tabela acima.

```
0,tcp,http,SF,223,486,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,7,7,0.00,0.00,0.00,0.00,1.00,0.00,0.00,50,50,1
0,tcp,http,SF,222,1282,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,51,51,
0,tcp,http,SF,168,124,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,52,52,1
0,tcp,http,SF,169,9020,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,53,53,
0,tcp,http,SF,219,1337,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,54,54,
0,tcp,http,SF,218,1364,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,55,55,
0,tcp,http,SF,165,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,3,3,0.00,0.00,0.00,0.00,1.00,0.00,0.00,56,56,
0,tcp,http,SF,223,1295,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,4,4,0.00,0.00,0.00,0.00,1.00,0.00,0.00,57,57,
0,tcp,http,SF,220,1228,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,5,5,0.00,0.00,0.00,0.00,1.00,0.00,0.00,58,58,
0,tcp,http,SF,217,2032,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,6,6,0.00,0.00,0.00,0.00,1.00,0.00,0.00,59,59,
0,tcp,http,SF,223,486,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,7,7,0.00,0.00,0.00,0.00,1.00,0.00,0.00,60,60,1
0,tcp,http,SF,222,1282,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,61,61,
```

Figura 2.3 Parte do NLS-KDD dataset. Fonte: (ISCX, 2009)

2.3 Ontologias

As ontologias são estruturas de representação do conhecimento assim como as Redes Semânticas, a Lógica de Primeira Ordem e os *Frames*. Uma ontologia é definida por Gruber (1995) como "A especificação formal e explícita de uma conceituação compartilhada". Formal pelo fato de a especificação ser declarativamente definida (por meio de lógica descritiva por exemplo) e assim ser

compreensível a agentes e sistemas computacionais; explícita pois, as características dos objetos representados pelo conjunto de conceitos, os relacionamentos, e as propriedades a eles associados, são definidos de forma explícita; conceituação refere-se a um modelo abstrato que representa um determinado contexto do mundo; e compartilhada porque o conhecimento é consensual, ou seja, aceito por um grupo.

Diversos são os benefícios apresentados na literatura para a utilização de ontologias (NOY e MCGUINNESS, 2001). Alguns deles são relacionados ao compartilhamento, reuso, estruturação da informação, interoperabilidade e confiabilidade.

Formalmente, uma ontologia pode ser definida como uma tupla O = {C, H, I, R, P, A} em que,

$C = C_C \cup C_I$ é o conjunto de entidades da ontologia. São designados por um ou mais termos em linguagem natural. O conjunto C_C é composto pelos identificadores das classes, isto é, conceitos que representam entidades que descrevem um conjunto de objetos (por exemplo, “Professor” $\in C_C$) enquanto o conjunto C_I é constituído por identificadores de instância (por exemplo, “instância Anne Smith” C_I), descrevendo objetos, que são entidades únicas.

H: $C_C \times C_C$; $H = \{(c_1,c_2)|c_1 \in C_C \wedge c_2 \in C_C\}$ é o conjunto de relacionamentos taxonômicos entre conceitos, que definem uma hierarquia de conceitos, as quais significam que c_1 é uma subclasse de c_2, por exemplo, “(Professor, Pessoa)”.

I: $C_C \times C_I$; $I = \{(c_1,c_2)|c_1 \in C_C \wedge c_2 \in C_I\}$ é o conjunto das relações entre os identificadores de classes e os identificadores de instâncias correspondentes, exemplo “(Pessoa, “instância Anne Smith ”)”.

rel_k: $R \rightarrow O(C)$; $R = \{rel_k (c_1,c_2,\ldots, c_n) \mid \forall i, c_i \in C_C\} \cup \{rel_k (c_1,c_2,\ldots, c_n) \mid \forall i, c_i \in C_I\}$ é o conjunto de relacionamentos não taxonômicos de uma classe da ontologia, por exemplo, “mãe_de(“instância Anne Smith”, “instância Kate Smith”)”.

$prop_k$: $P \rightarrow C_C \times Tipo\_data$; $P = \{prop_k(c_i,Tipo\_data)|c_i \in C_C\} \cup \{prop_k(c_i,valor)| c_i \in C_I$ é o conjunto de propriedades de uma classe da ontologia e seus tipos de dados básicos, por exemplo, “data_de_aniversário (Pessoa, data)” e instâncias correspondentes, por exemplo “data_de_aniversário(“instancia Anne Smith”, 06/07/2000)”.

$rule_k : R \rightarrow O(C)$ é um conjunto de axiomas, regras que permitem verificar a consistência de uma ontologia e obter novos conhecimentos através da inferência; $rule_k$ é uma regra do tipo condição$_x \Rightarrow$ conclusão$_y$ onde, condição$_x$ = $(cond_1, cond_2, \ldots, cond_n) | \forall z, cond_z \in H \cup I \cup R \cup F$. Por exemplo, "$\forall$ Pessoa$_1$, Pessoa$_2$, Pessoa$_3$, mãe_de (Pessoa$_1$, Pessoa$_2$) $\wedge$ mãe_de (Pessoa$_1$, Pessoa$_3$) $\Rightarrow$ irmão_de (Pessoa$_2$, Pessoa$_3$)" é um axioma que indica que se duas pessoas têm a mesma mãe então, elas são irmãs.

Segundo Guarino (1998) uma ontologia é classificada de acordo com sua generalidade, como ontologia de alto-nível, domínio, tarefa ou aplicação. A ontologia de alto-nível descreve conceitos genéricos que são independentes de um domínio particular como espaço e tempo. Ontologia de domínio representa conceitos explícitos, relacionados a um determinado domínio de conhecimento, e os relacionamentos existentes entre eles como conceitos de Segurança da Informação. As ontologias de tarefa descrevem atividades que podem ser usadas em um ou vários domínios, como "detectar ataques" e "recomendar solução". Por último, a ontologia de aplicação descreve uma aplicação que depende tanto dos conceitos de uma ontologia de domínio quanto de tarefas específicas para esse domínio. Ela é uma especialização dos termos da ontologia de domínio e de tarefa correspondentes, por exemplo, o IDS é uma aplicação que depende dos conceitos de Segurança da Informação (domínio) e executa as atividades de "detectar" e "alertar" (tarefas) sobre ataques para administradores de rede.

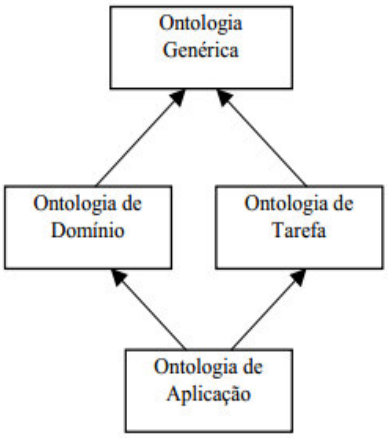


Figura 2.1 Taxonomia de ontologias de acordo com seu nível de generalidade. Fonte: (Guarino, 1998)

2.4 Considerações finais

Foram abordados, neste capítulo, os conceitos do domínio de Segurança da Informação necessários para uma boa compreensão da ontologia desenvolvida neste trabalho. Os principais subconceitos relacionados a esse domínio; conceitos de ameaças, ataques, os principais tipos de ataques e os princípios básicos da Segurança da Informação; os mecanismos existentes para abordar o problema de ataques à Segurança da Informação destacando os IDSs por serem o foco da ontologia de aplicação desenvolvida e o sistema mais confiável para lidar com as invasões no ambiente do computador e alertar os administradores da rede.

Este capítulo mostrou ainda o conjunto de dados NSL-KDD (REVATHI e MALATHI, 2013), que é utilizado como instâncias de ataques na ontologia para avaliação desta e os conceitos de ontologia, bem como a taxonomia de ontologias proposta por Guarino (1998).

3 TRABALHOS RELACIONADOS

Neste capítulo são discutidos três trabalhos relacionados ao que está sendo proposto. Uma análise da ontologia de cada um deles é realizada destacando características relevantes como o tipo de ontologia, as tarefas suportadas, a ferramenta usada para a construção das ontologias, os elementos que cada ontologia comporta, a linguagem na qual foram desenvolvidas e a aplicação que cada trabalho desenvolveu.

3.1 Proposta de Undercoffer, Joshi e Pinkston (2003): Modeling Computer Attacks: An Ontology for Intrusion Detection

Os autores relatam que muitas taxonomias de ataques para IDS têm sido propostas pela comunidade acadêmica e que, algumas utilizam uma linguagem descritiva, mas a maioria não. Eles mencionam alguns problemas das taxonomias como o fato de que os modelos de dados das taxonomias devem ser codificados em um sistema de *software* para se fazer operações entre instâncias e, qualquer alteração ou mudança feita nesse modelo, é necessário alterar o sistema de *software* também. Outro problema é que as taxonomias só proporcionam esquemas para classificação, ou seja, elas não possuem condições necessárias e suficientes para que um sistema de *software* possa raciocinar sobre uma instância na taxonomia. Apresentam ainda que a maioria das linguagens utilizadas nessas taxonomias não são extensíveis nem transmissíveis entre sistemas não-homogêneos e sua semântica muitas vezes são vagas.

Para mitigar esses problemas os autores sugerem que seja feito a uma mudança da utilização de taxonomias de ataques para ontologias. Com essa ideia em mente, os eles construíram um modelo de dados que caracteriza o domínio de ataques e invasões de computadores como uma ontologia. Produziram uma ontologia que especifica um modelo de ataques a rede de computadores para mostrar os benefícios dessa mudança de taxonomias para ontologias. Eles sugerem essa mudança para diminuir os problemas encontrados na representação em taxonomias. A ontologia desenvolvida foi especificada nas linguagens *DAML* (“*DARPA Agent Markup Language*”) e *OIL* (“*Ontology Inference Layer*”). Os axiomas da ontologia são implementados usando *DAMLJessKB* (KOPENA e REGLI, 2003), que é uma máquina de inferência para *DAML*.

Para a construção da ontologia, os autores se basearam em uma análise empírica das características e atributos, e suas inter-relações, de mais de 4.000 classes de ataques e intrusões. A Figura 3.1 representa uma visão de alto nível dessa ontologia.

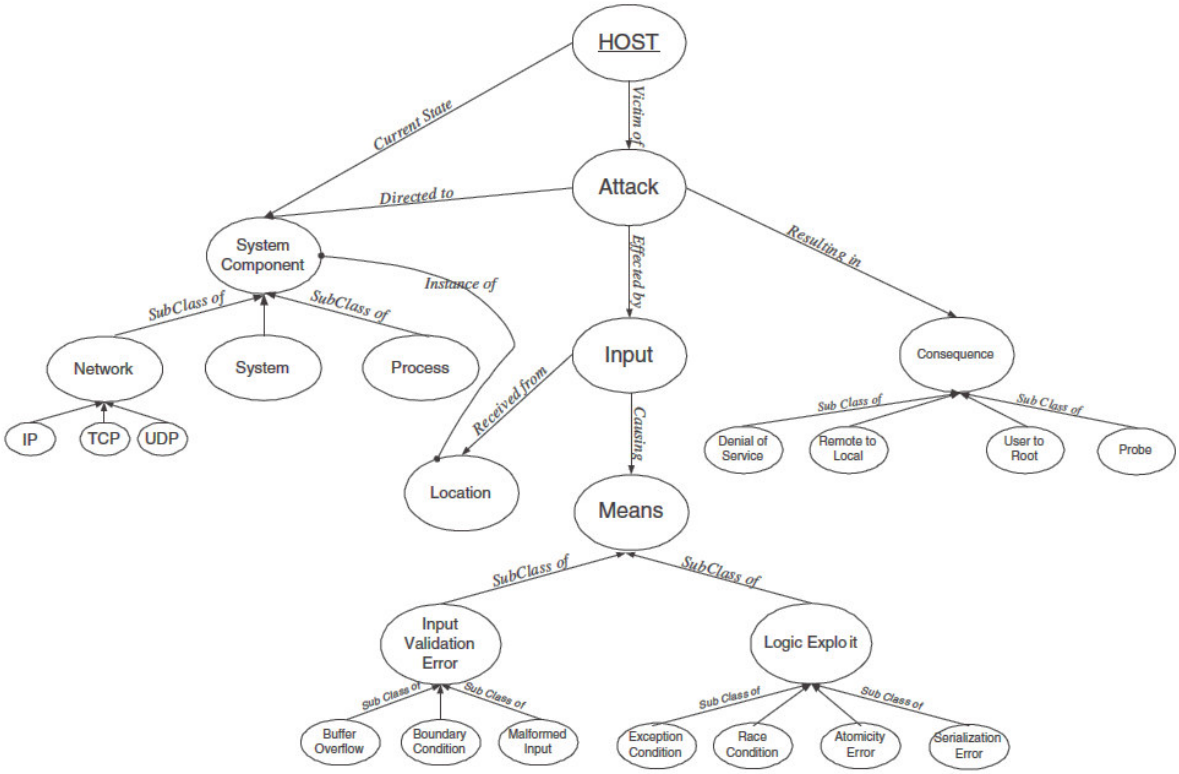


Figura 3.1 Ontologia em alto nível para IDS. Fonte: (Undercoffer,Joshi and Pinkson 2003)

No nível mais alto da ontologia está definida a classe *Host*, que possui os relacionamentos não taxonômicos *Current State* e *Victim of* com as classe *System Component* e *Attack,* respectivamente.

A classe "*System Component*" possui as subclasses "*Network*", "*System*" e *Process*". "*Network*" representa a camada de rede. Como os autores focam apenas no protocolo "*TCP/IP*", somente as subclasses "*IP*" ("*Internet Protocol*"), "*TCP*" e "*UDP*" ("*User Datagram Protocol*") são consideradas abaixo de "*Network*". A subclasse "*System*" representa o sistema operacional do "*host*", inclui atributos que representam o uso geral da memória e o uso do processador. Essa subclasse também possui atributos relacionados ao número de usuários atuais, uso de disco, tabela de chamadas de sistema, etc. A subclasse "*Process*" possui atributos que representam os processos que devem ser monitorados. Esses atributos incluem o

valor atual do apontador de instruções, o topo da pilha e o número de processos filhos.

A classe "*Attack*" possui relacionamentos "*Directed to*", "*Effected by*" e "*Resulting in*". Essa construção foi escolhida pelos autores pela noção de que um ataque consiste de algum conteúdo que é direcionado a algum componente do sistema e resulta em alguma consequência. Desta forma, as classes "*System Component*", "*Input*", e "*Consequence*" são os objetos respectivamente correspondentes. Esta última possui diversas subclasses, entre elas "*Denial of Service*" (negação de serviço), "*User Access*" (acessos não privilegiados), "*Root Access*" (acessos privilegiados) e "*Probe*" (escaneamento, podendo obter informações de perfis do sistema).

Finalmente, a classe "*Input*" possui os atributos "*Received from*" e "*Causing*", em que a última define a relação entre as classes "*Means*" e "*Input*". Os autores definem as subclasses "*Input Validation Error*" e "*Logic Exploits*" para a classe "*Means*". "*Input Validation Error*" representa erros causados quando conteúdo malformado é recebido pelo sistema e não é tratado apropriadamente, e possui as subclasses a seguir: "*Buffer Overflow*" (utilização extensiva da memória através do envio de estruturas de dados complexas), "*Boundary Condition Error*" (um processo tenta ler ou escrever além do último endereço válido) e "*Malformed Input*" (quando um processo aceita conteúdo sintaticamente incorreto). A subclasse "*Logic Exploits*" representa vulnerabilidades de hardware ou *software* que podem ser exploradas, e possui as subclasses a seguir: "*Exception Condition*" (erro originado pela falha em manipular uma exceção gerada), "*Race Condition*" (erro que ocorre durante uma janela de tempo entre duas operações), "*Serialization Error*" (serialização inapropriada de operações) e "*Atomicity Error*" (erro que ocorre quando uma estrutura de dados parcialmente modificada é utilizada por outro processo). Os autores não especificam ou não desenvolveram nem uma aplicação para a ontologia desenvolvida.

3.2 Proposta de Rosa (2011): Uma Ontologia para Sistema de Detecção de Intrusão em Web Services

O autor propôs o uso de ontologias para construir uma BC (Base de conhecimento) de ataques para IDS, baseada nas ações desses ataques. Ele considera que os IDSs baseados em anomalias tentam deduzir um ataque baseados em perfis de ataques conhecidos, mas quando erram geram falsos positivos e que, os baseados em assinaturas erram menos, porém só detectam ataques que já estejam em sua base de ataques, propiciando ataques *zero-day* (ataques que surgem todos os dias). Assim, com sua proposta, o autor pretende melhorar o mecanismo de detecção de ataques a serviços *web* com a construção de uma ontologia de aplicação cujo domínio de conhecimento são ataques e suas ações são tarefas para detectar ataques. Além disso, propõe um mecanismo que captura pacotes da rede e usa a ontologia como uma base de ataques para IDSs, de forma a detectar ataques conhecidos e, deduzir novos ataques por meio de mecanismos de inferência.

Para construir a ontologia, o autor utilizou a taxonomia de ataques apresentada pela *CAPEC* ("*Common Attack Pattern Enumeration and Classification"*) (CAPEC, 2011), modelou-a utilizando a ferramenta Protegé (STANFORD, 2014), e a linguagem OWL. A ontologia é composta de classes e propriedades (Figura 3.2), instâncias (Figura 3.3) e axiomas (um exemplo é ilustrado na Figura 3.4). As classes principais da ontologia são "*AttackAction"* e "*WebServicesAttack"*. "*AttackAction"* possbi subclasses contendo instâncias que representam ações suspeitas ("*payloads"*) que podem ser capturadas na rede.

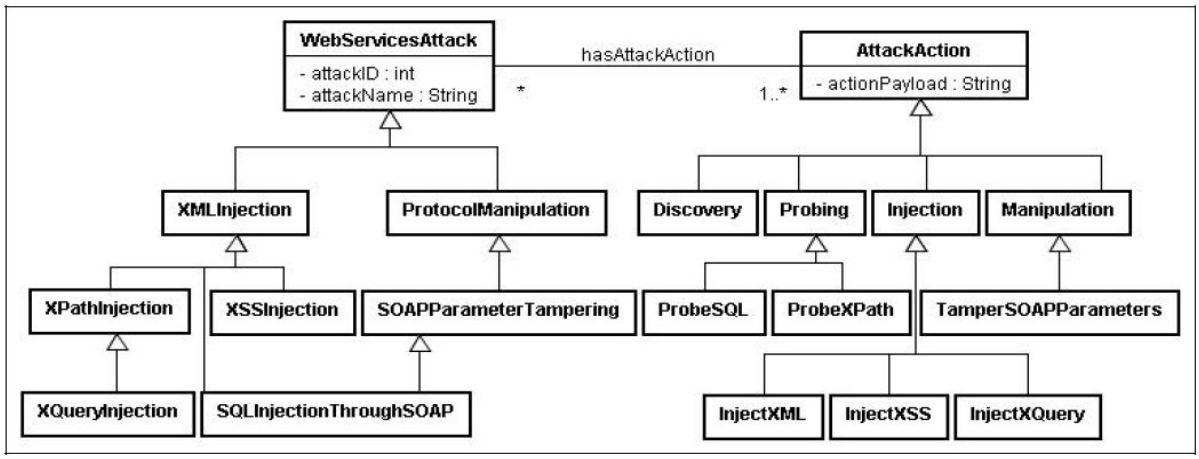


Figura 3.2 Diagrama de classes da ontologia. Fonte: (Rosa, 2011)

A classe “*WebServicesAttack”* possui subclasses representando categorias de ataques a *web services*. Cada subclasse possui instâncias representando assinaturas (estratégias específicas) de ataques conhecidos. Uma instância de ataque pode ter uma ou mais ações e uma ação pode ser parte de várias instâncias de ataque.

Na Figura 3.3 tem-se dois exemplos de instâncias da ontologia: “xpathInjection1” (instância da subclasse de ataques “*XPathInjection”*) e “*xqueryInjection1”* (instância da subclasse de ataques “*XQueryInjection”*). A instância “*xpathInjection1”* tem relacionamento “*hasAttackAction”* com as ações “*getWSDL”* (instância da subclasse de ações “*Discovery”*) e “*probeXPath1”* (instância da subclasse de ações “*ProbeXPath”*). Enquanto que “*xqueryInjection1”* tem relacionamento com as mesmas ações de “*xpathInjection1”* e também com a ação “*injectXQuery1”* (instância da subclasse de ações “*InjectXQuery”*).

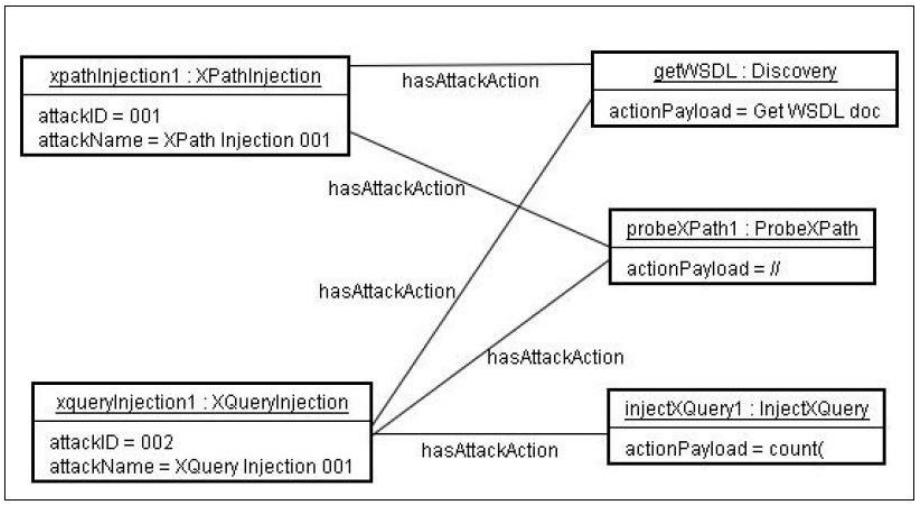


Figura 3.3 Exemplos de Instâncias da ontologia. Fonte: (Rosa, 2011)

O autor exemplifica um axioma (Figura 3.4) criado na ontologia para a classe “*XQueryInjection”*, representado com lógica de primeira ordem que instrui a máquina de inferência a deduzir que qualquer ataque que possui uma ação do tipo “*Discovery”*, uma ação do tipo “*ProbeXPath”*, e outra ação do tipo “*InjectXQuery”* terá que obrigatoriamente ser uma instância da classe “*XQueryInjection”*. Isto, na lógica de detecção do IDS, significa que um ataque do tipo “*XQueryInjection”* ocorreu.

"XQueryInjection ≡ ∃hasAttackAction.Discovery ⊓ ∃hasAttackAction.ProbeXPath ⊓ ∃hasAttackAction.InjectXQuery" (1)

Figura 3.4 Exemplo de um axioma. Fonte: (Rosa, 2011)

A Figura 3.5 ilustra a ontologia proposta pelos autores no Protegé, no qual o lado esquerdo mostra a estrutura das classes da ontologia no Protegé, criada baseando-se no diagrama de classes da Figura 3.2. Do lado direto dessa figura tem-se um exemplo de instância de ataque ("*xqueryInjection1"*) e suas propriedades.

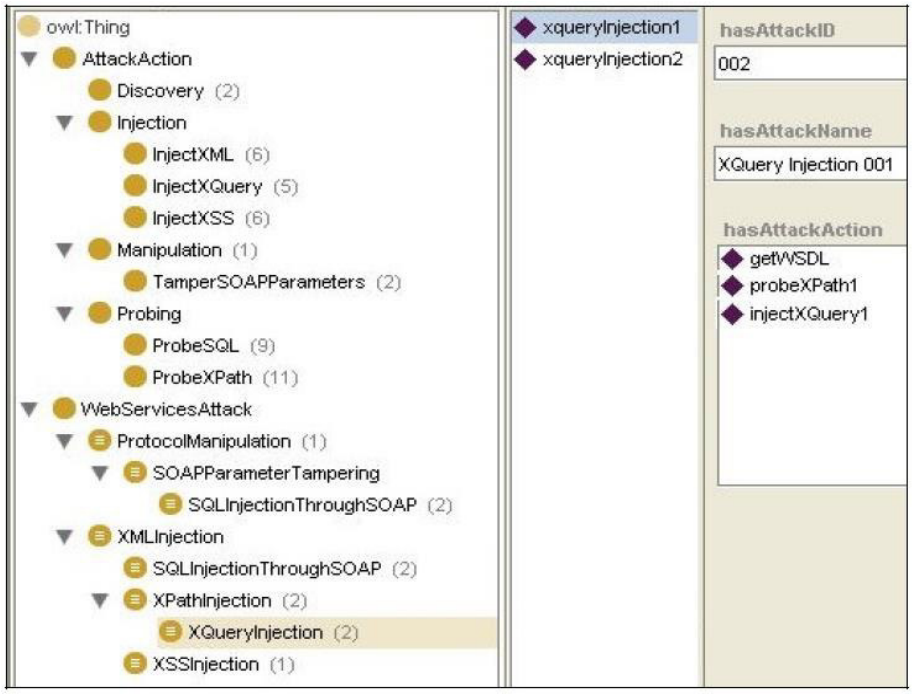


Figura 3.5 Ontologia proposta no Protegé. Fonte: (Rosa, 2011)

3.3 Proposta de Boulahia et al. (2009): An Ontology-based Approach to React to Network Attacks

Os autores relatam que, um dos motivos pelo qual o número de ataques baseados em Protocolo de Internet (*IP*) tem aumentado é pelo fato de que operadores e prestadores de rede estão oferecendo novos serviços, tais como voz ou televisão baseados em *IP* e que, para abordar a evolução dos incidentes de segurança nas redes de comunicação atuais, é importante reagir rapidamente e de forma eficiente a um ataque. Assim, propuseram uma abordagem baseada em ontologias para especificar políticas de segurança. Para eles, essa abordagem oferece uma maneira de mapear alertas em contextos de ataque, que são usados

para identificar as políticas a serem aplicadas na rede para resolver a ameaça. Eles definem ontologias para descrever alertas e políticas, utilizando regras de inferência para realizar tais mapeamentos. Essas ontologias são aplicadas à segurança em rede de computadores e foram desenvolvidas com o editor de ontologias Protegé usando as linguagens *OWL* e *SWRL*, sendo que a *OWL* é a linguagem usada para descrever conceitos *IDMEF* ("*Intrusion Detection Message Exchange"*) (DEBAR, CURRY e FEINSTEIN, 2007) – usado para propagar alertas por meio do envio de mensagens – e conceitos *OrBAC* ("*Organization Based Access Control"*) (ABOU-EL-KALAM et al., 2003) – é um conjunto de regras de segurança contextual correspondente a permissões, proibições e obrigações – e, o *SWRL* é usado para mapear informações *IDMEF* para *OrBAC.* Essas ontologias são compostas por todos os elementos da tupla O = {C, H, I, R, P, A}. A regras OrBAC aplicam-se quando seus contextos associados são ativados. A Figura 3.6 exibe a ontologia de políticas OrBAC e a Figura 3.1, a ontologia de alertas IDMEF, ambas definidas por eles.

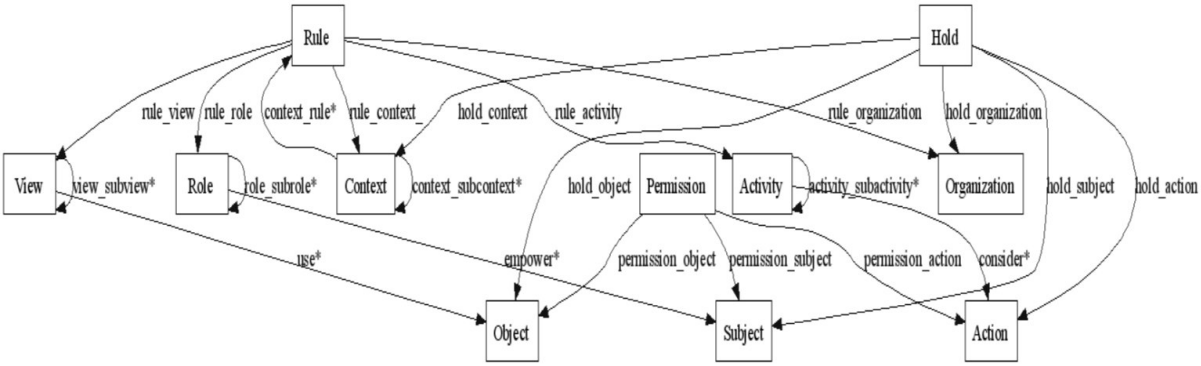


Figura 3.6 Ontologia de políticas OrBAC. Fonte: (VERGARA et al., 2009) (BOULAHIA et al., 2009)

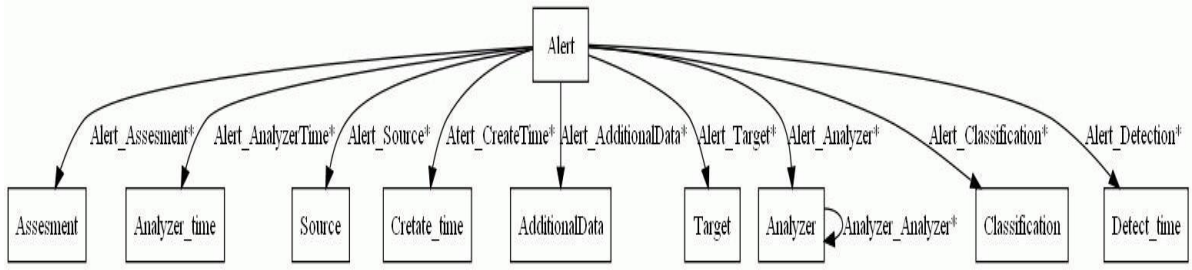


Figura 3.1 Ontologia IDMEF (Relacionamentos de Alertas). Fonte: (BOULAHIA et al., 2009)

3.4 Comparativo

Todas essas propostas são ontologias de aplicação e estão bem definidas, porém suas tarefas são bastante limitadas e nem todas fornecem uma ontologia de domínio para que possa ser reusada na construção de outras ontologias de aplicação. As ontologias de Undercorffer, Joshi e Pinkston (2003) e Rosa (2011), possuem uma tarefa importantíssima que é "Detectar ataques", entretanto, o primeiro não define axiomas e o segundo tem uma representação semântica pobre, na qual a maioria das classes não possui relacionamentos entre si, sendo o restante da ontologia uma taxonomia. A ontologia de Boulahia et. al (2009) permite identificar contexto de alertas de ataques à rede para gerar novas políticas mas, não permite detectar ataques na rede. As relações entre os principais aspectos de cada trabalho para efeito comparativo entre eles estão listadas na Tabela 3.1.

Tabela 3.1 Comparativo entre os trabalhos estudados

Proposta	**Tipo de ontologia**	**Tarefas suportadas**	**Ferra-menta usada**	**Elementos que compõem**	**Lingua-gem usada**	**Aplica-ções desen-volvidas**
Uma Ontologia para Sistemas de detecção de Intrusão em Web Services. (ROSA, 2011)	Aplicação	Detectar ataques	Protegé	Classes, propriedades, hierarquias, instâncias e axiomas	OWL	Protótipo de IDS
An Ontology-based Approach to react to Network Attacks (BOULAHIA et. al 2009)	Aplicação	Especificar políticas de segurança	Protegé	Classes, relacionamentos não taxonômico, propriedades, instâncias e axiomas	OWL e SWRL	Não definido
Modeling Computer Attack: An Ontology for Intrusion Detection. (UNDERCOFFER, JOSHI e PINKSTON, 2003).	Aplicação	Detectar ataques	Não definido	Classes, hierarquias, propriedades e relacionamentos não taxonômicos	DAML e OIL	Não definido

3.6 Considerações Finais

Um estudo do estado da arte de algumas abordagens ontológicas no domínio de Segurança da Informação foi realizado neste capítulo. Os três trabalhos estudados fornecem uma ontologia de aplicação e os trabalhos de Undercoffer, Joshi e Pinkston (2003) e Rosa (2011) possuem a tarefa de detectar ataques enquanto que a ontologia de Boulahia e seus colegas possui a tarefa de identificar

políticas de segurança no contexto de alertas de ataques. O trabalho mais completo e que mais se aproxima do trabalho proposto é o de Rosa (2011), o único que desenvolveu um pequeno sistema (Protótipo de um IDS) para avaliar sua ontologia. Todos esses trabalhos são bem estruturados e estão relacionados com o trabalho proposto, principalmente por abordarem problemas relacionados a ataques à Segurança da Informação.

4 UMA ONTOLOGIA DE APLICAÇÃO PARA SISTEMAS DE DETECÇÃO DE INTRUSÃO BASEADO EM CASOS

Este capítulo apresenta a ontologia de aplicação proposta, denominada ONTOID, que fornece conhecimento no domínio de Segurança da Informação e, uma aplicação dessa ontologia para o desenvolvimento de IDSs baseados em casos.

4.1 Metodologias e técnicas usadas no desenvolvimento da ONTOID

A ontologia proposta, denominada ONTOID foi construída no editor de ontologias Protegé na linguagem OWL, reusando parte da ontologia desenvolvida por Mendes (2013). Para guiar a construção da ONTOID utilizou-se o processo MADAE-Pro "*Multi-agent Domain and Application Engineering Process*" (GIRARDI e LEITE, 2011). O principal passo realizado durante a concepção da ONTOID foi a definição do escopo da ontologia. Para isso, foram construídos o Modelo de Conceitos (Figura 4.1) e o Modelo de Objetivos (Figura 4.2), conforme as diretrizes do MADAE-Pro.

No Modelo de Conceitos definiu-se os principais conceitos do domínio da Segurança da Informação, por exemplo, o conceito "*System to detect threats to information security*" possui os relacionamentos "*identify*", "*monitors*", "*uses*", "*provides*" e "*analyzes*" com os conceitos "ONTOID", "*Data package*", "*Computer network*", "*Solutions*" e "*Data Package*" respectivamente. O conceito "ONTOID" tem o relacionamento "*represents*" com o conceito "*Domain concepts information security*", pois a ontologia representa conceitos do domínio como "CIA *properties*" (proprideades de Confidentiality, Integrity e Availability), "*Threat*", "*Information Asset*", "*Attacker*", "*Attack Type*", "*Vulnerability*" e "*Risk*". O conceito "*Data package*", por sua vez, possui dois tipos de subconceitos, o *"Normal package*" e o "*Anormal package*". Uma rede de computadores possui ameaças que afetam um ou mais ativos de informação e pode ser natural ou intencional, esta última mais conhecida como ataque. Os sistemas de informação possuem vulnerabilidades que, por sua vez, causam ameaças naturais e possuem um risco (uma consequência negativa, de uma vulnerabilidade, que pode ocorrer caso seja explorada). Um ataque à Segurança da Informação afeta propriedades da Segurança da Informação e determinados sistemas operacionais. Ele possui um tipo de ataque, pertence a uma

das categorias de ataque do NSL-KDD e é realizado por um atacante que usa uma vulnerabilidade para a realização de um ataque.

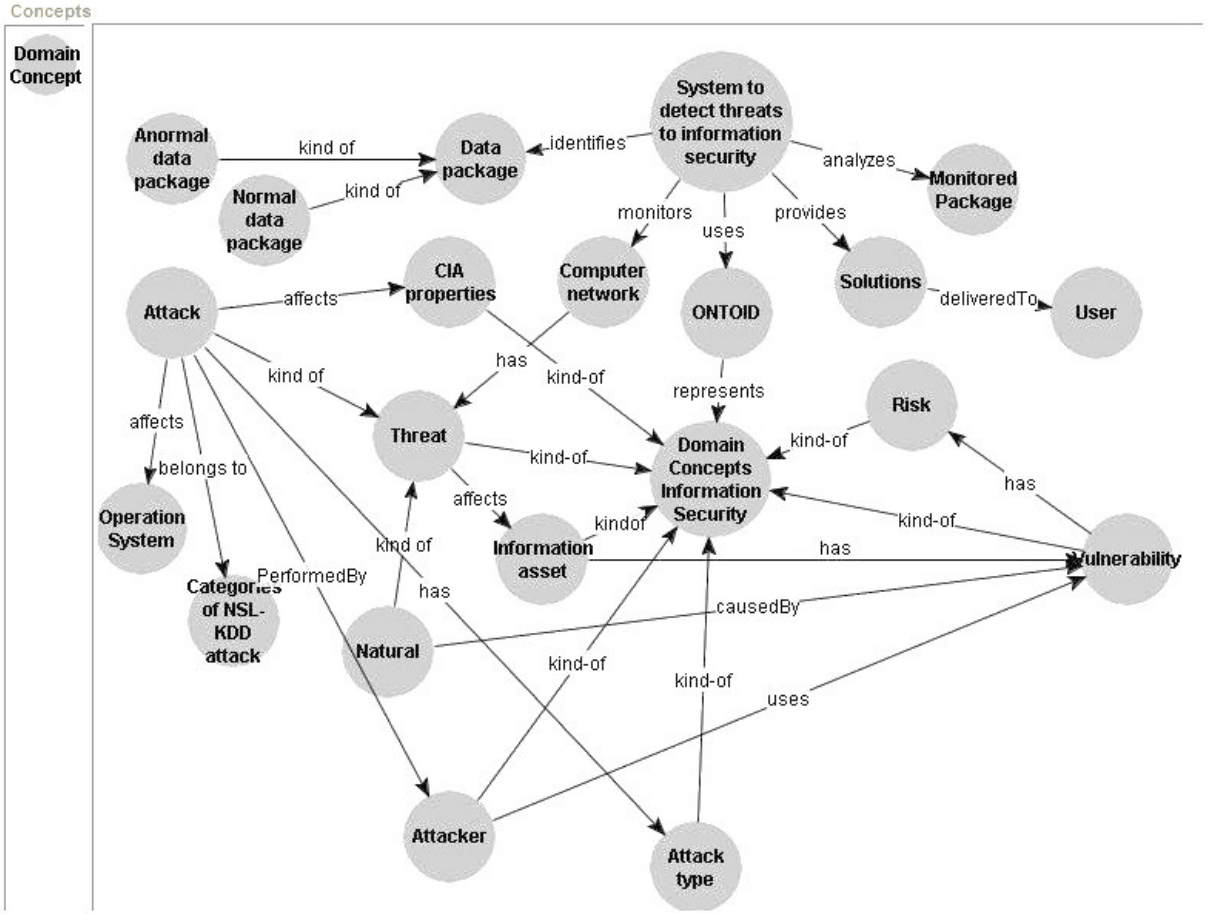


Figura 4.1 Modelo de conceitos da ONTOID

No Modelo de objetivos definiu-se os objetivos geral e específicos do IDS, as responsabilidades que são necessárias para que cada objetivo específico seja alcançado e a entidade externa ao sistema. Para alcançar o objetivo geral que é “fornecer uma solução baseada em casos para ataques à Segurança da Informação”, é necessário “manter uma base de conhecimento contendo casos de ataques” que possui as responsabilidades de “armazenar casos” e “atualizar a ontologia”; “monitorar a rede de computadores” com a responsabilidade de “obter informações sobre os pacotes” de dados monitorados; “detectar ataques” à rede, sendo necessário “criar uma instância de um caso problema na ontologia” e fazer a “análise de similaridade” entre esses pacotes; e “entregar uma solução para o ataque identificado”, para o administrador, com a responsabilidade de “escolher a solução do caso mais similar”.

A partir dos Modelos de Conceitos e de Objetivos, definiu-se as classes principais da ontologia, sua hierarquia, os relacionamentos não taxonômicos.

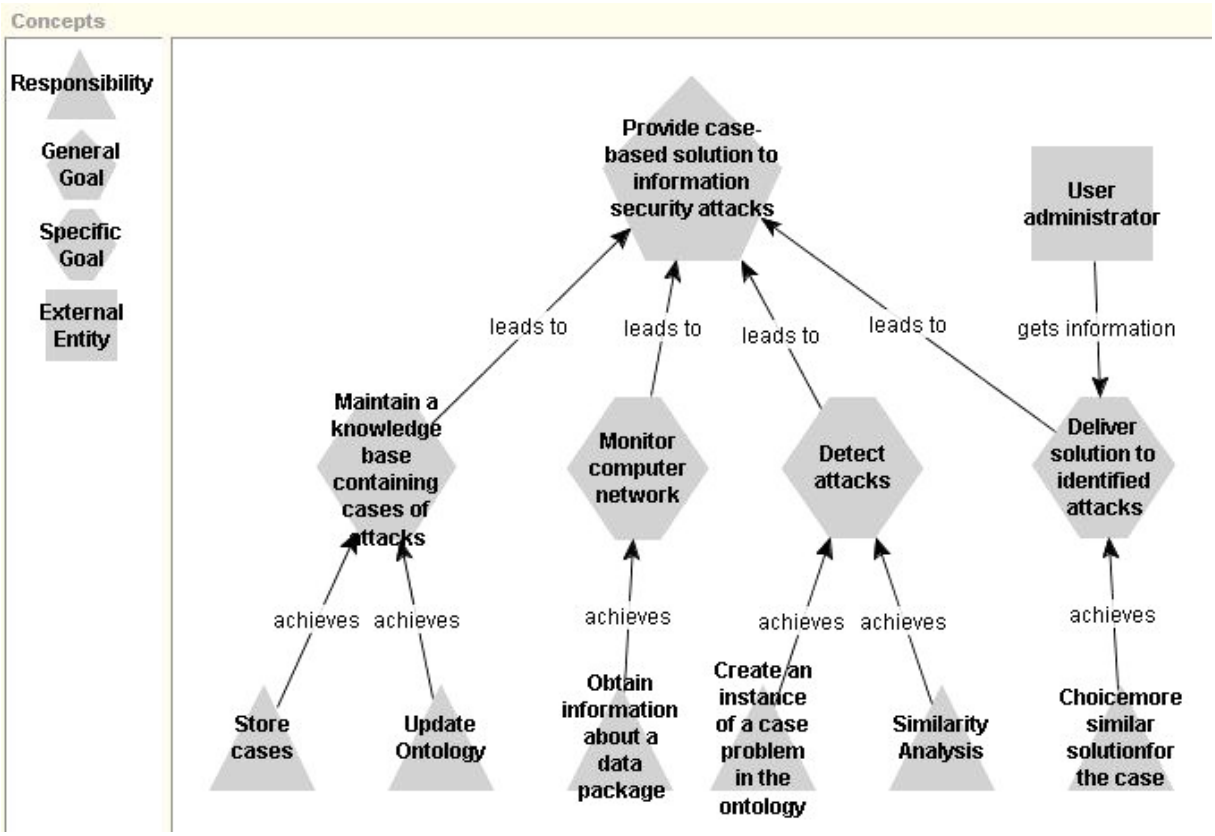


Figura 4.2 Modelo de objetivos da ONTOID

A representação dos casos na ONTOID é baseada no paradigma CBR que, como já foi dito, é um paradigma para resolução de problemas que se baseia no conhecimento de experiências anteriores para resolver novas situações de problemas. Um caso no CBR é composto de duas partes básicas, a que corresponde à descrição do problema e é utilizada para encontrar problemas similares e a que especifica a solução do problema (MENDES, GIRARDI e LEITE, 2013).

4.2 Representação dos casos

A ONTOID é composta de cinco classes principais denominadas "*AttackCase*", "*AttackCaseProblem*", "*AttackCaseSolution*", "*DomainSecurityConcepts*" e "*TasksConcepts*", em que as três primeiras

representam os casos na ONTOID e são descritas nesta seção. A classe "*DomainSecurityConcepts*" representa os principais conceitos do domínio em questão e é descrita na Seção 4.3. A classe "*TasksConcepts*" representa as tarefas da ONTOID, discutidas na Seção 4.4.

A classe "*AttackCase*" representa os casos de ataques consistentes de um problema e uma solução, sendo que "*AttackCaseProblem*" expressa os problemas de ataques à rede de computadores e "*AttackCaseSolution*" a solução para cada um desses problemas. A Figura 4.3 mostra uma representação dos relacionamentos não-taxonômicos entre essas classes.

As propriedades da classe "*AttackCaseProblem*" são características mapeadas do conjunto de dados NSL-KDD (REVATHI e MALATHI, 2013). O NSL-KDD é a versão atualizada do KDD CUP 99 (TAVALLAEE et al., 2009) o qual é um conjunto de dados gerados a partir da simulação de ataques de intrusão ao ambiente de uma rede de computadores. Esse conjunto é composto por dados de conexões capturados da rede onde, cada sessão de conexão possui 41 características com informações (valores) que o identificam como uma concexão pacote "ruim" ou "boa", ou seja, um ataque ou tráfego normal da rede, respectivamente. Essas características estão descritas na Tabela 2.1, Seção 2.2.

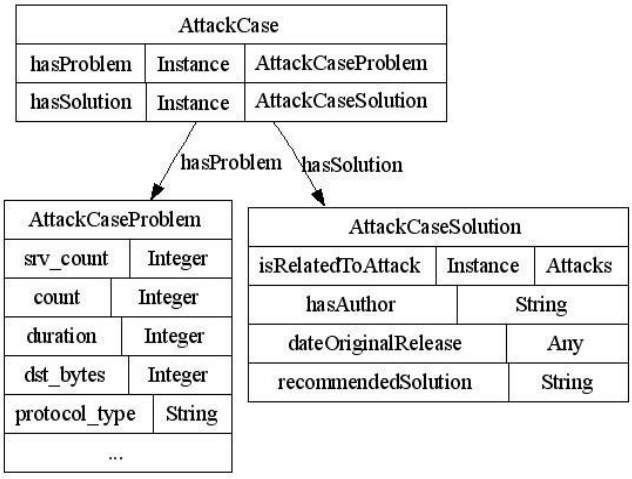


Figura 4.3 Classes da ONTOID para a representação de um caso de ataque

4.3 Representação dos conceitos do domínio

A Figura 4.4 apresenta a superclasse "*SecurityDomainConcepts*" que descreve os principais conceitos do domínio de Segurança da Informação como

“ameaça”, “mecanismo de segurança”, “ativo de informação” entre outros. Uma ameaça pode ser natural, por exemplo, uma falha de *software*, ou intencional, mais conhecida como ataque. Ela pode prejudicar um ou mais ativos de informação – qualquer coisa que tem valor para uma organização como hardware, *software*, dados e pessoas – e afeta um ou vários princípios de Segurança da Informação como a disponibilidade e a integridade, por exemplo, um ataque de *smurf* afeta a disponibilidade das informações, pois ele provoca uma lentidão na rede através do envio de uma grande quantidade de pacotes *icmp* para o endereço *IP* de broadcast da rede. Um ataque possui um tipo, ou seja, um ataque pode ser de interrupção, interceptação ou modificação; é criado e realizado por um atacante (pessoa mal intencionada) para afetar determinados tipos de sistemas operacionais. Um ativo de informação pode ter vulnerabilidades que, por sua vez, são usadas por atacantes para realizar ataques, o que torna o ativo frágil e mais sujeito às ameaças e, elas possuem riscos – perigo ou possibilidade de perigo.

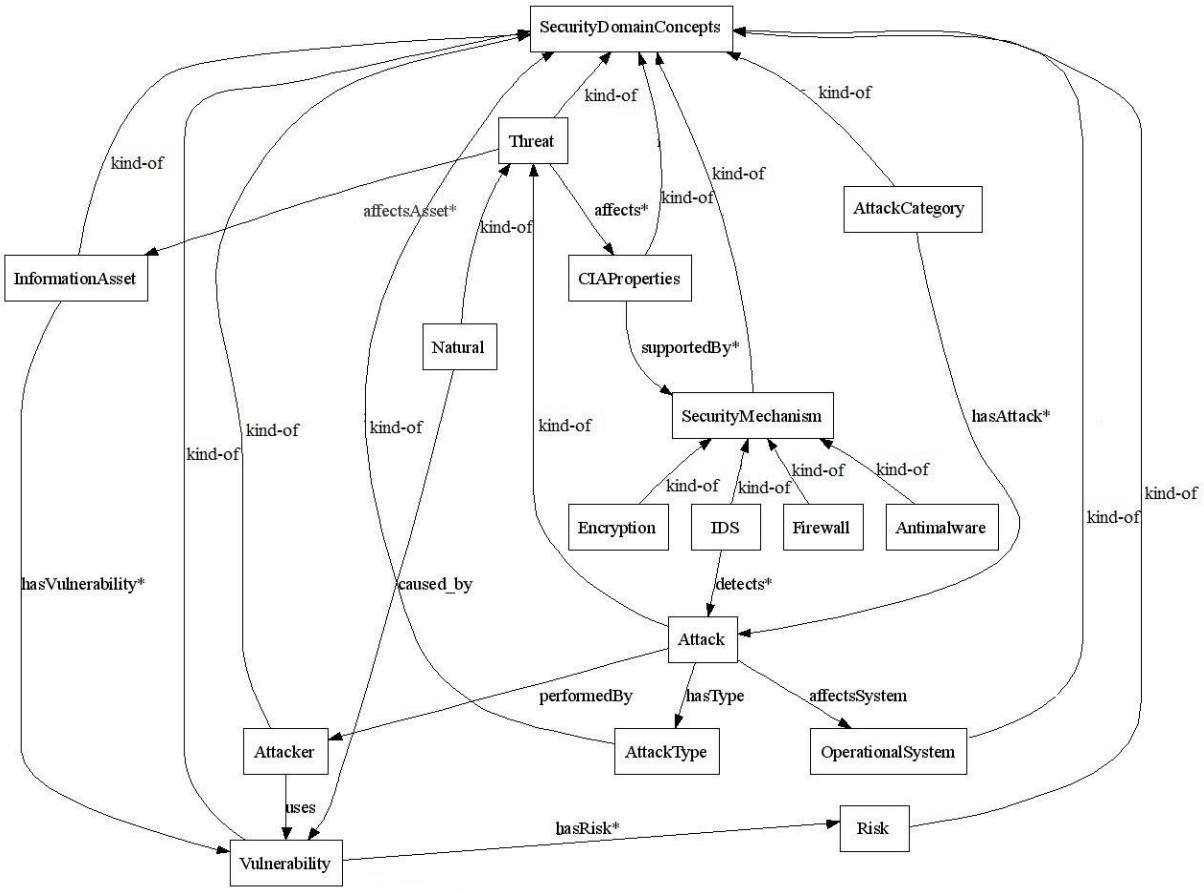


Figura 4.4 Principais conceitos do domínio da Segurança da Informação e seus relacionamentos

Os princípios de Segurança da Informação são suportados por um ou vários mecanismos de segurança – sistemas, *softwares* que servem como meio de proteger um usuário dos ataques existentes, como exemplo temos os “IDSs”, “*Firewalls*”, “*Antimalware*” e “*Encryptation*” – ou seja, os mecanismos dão suporte para garantir que os objetivos dos princípios sejam alcançados. A classe “*AttackCategory*” representa as categorias de ataques utilizadas no NSL-KDD que são: “*DoS attacks”*, “*Probe”*, R2L(“*Root to Local*”) e U2R (“*User to Root Attacks*”). Essas categorias estão definidas na Seção 2.2 e, cada uma delas possui um determinado conjunto de ataques.

4.4 Representação das tarefas da ONTOID

A superclasse “*TasksConcepts*” representa os conceitos utilizados na realização das tarefas da ONTOID. A Figura 4.5 ilustra esses conceitos e as tarefas da ONTOID que, como dito na seção anterior, foram baseadas nas funções de IDS descritas por Sawat e Itkar (2014) e citadas na Seção 2.1.7. A classe “*ComputersNetwork*” especifica a rede na qual o IDS está monitorando e “*MonitoredPackage*” representa os pacotes que estão sendo monitorados (os novos pacotes). As tarefas específicas são monitorar o tráfego de uma rede de computadores, analisar os pacotes que são monitorados, identificar problemas de ataques, detectar ataques, fornecer solução para um ataque encontrado, além de calcular a similaridade de um novo problema com os problemas da base. Como a ontologia é baseada em casos, inclui-se também as classes “*AttackCase*”, “*AttackCaseProblem*” e “*AttackCaseSolution*”, já mencionadas anteriormente. A classe “*AttackCaseSolution*” tem um relacionamento “*has_Attack*” com a classe “*Attack*”, onde estão instanciados esses ataques. Um exemplo de um caso instanciado na ONTOID é ilustrado na Figura 4.1. Nela temos a instância de um caso, chamada “*AttackCase_1*” que possui o problema “*AttackProblem_1*” (instância da classe “*AttackCaseProblem*”) e a solução “*AttackSolution_1*” (instância da classe “*AttackCaseSolution*”), cada uma com suas respectivas propriedades e valores.

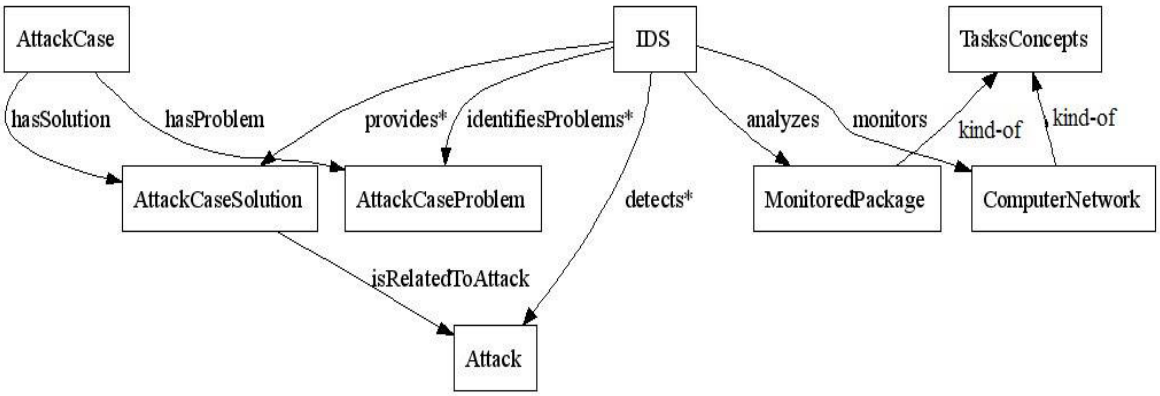


Figura 4.5 Principais tarefas de um IDS representadas na ONTOID

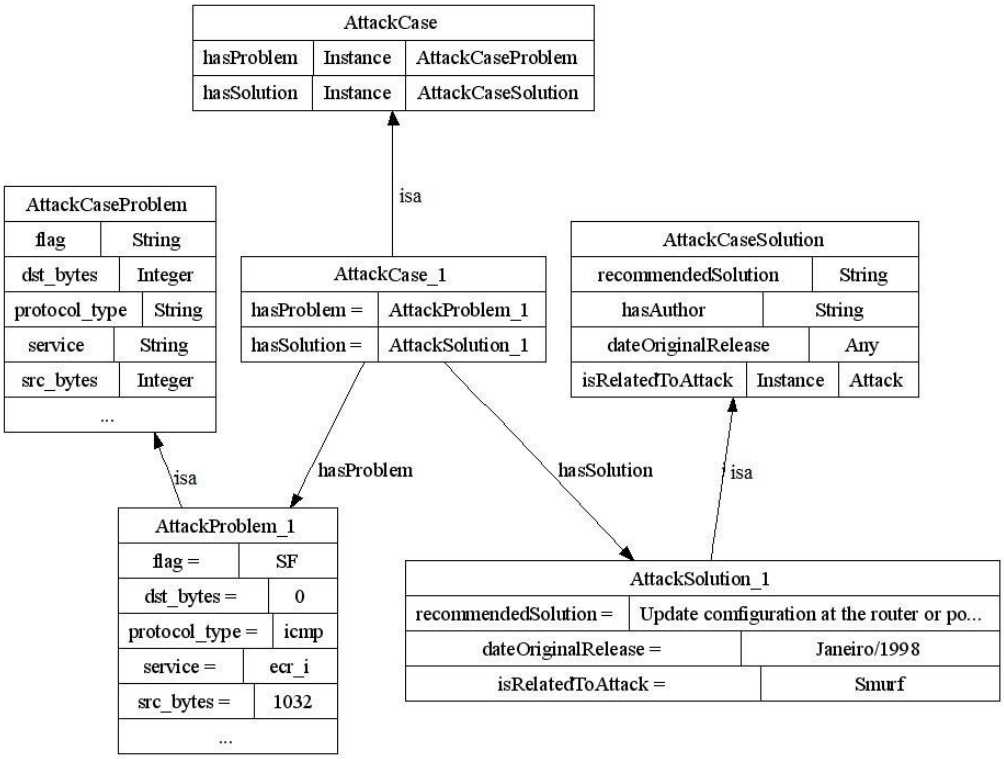


Figura 4.1 Exemplo de um caso de ataque na ONTOID

Como exemplo de um caso de ataque tem-se a instância "*AttackCase_1*" que possui o problema ataque "*AttackProblem_1*" e a solução "*AttackSolution_1*". A instância "*AttackProblem_1*", como todos os outros problemas, possui 41 propriedades refletidas do conjunto de dados NSL-KDD, que caracterizam sess~es de conexões de rede, como exemplo, "*protocol_type*" (tipo de protocolo usado pelo pacote). As propriedades de "*AttackProblem_1*" caracterizam um ataque "*smurf*". A solução para esse problema é identificada pela instância "*AttackSolution_1*" que possui como propriedades o nome do ataque possuidor de tais características, uma

recomendação para solucionar o problema, a data de publicação dessa solução e o autor da mesma.

A Tabela 4.1 apresenta algumas das propriedades das classes da ONTOID, bem como uma pequena descrição sobre elas. Outras propriedades estão explícitas na Figura 4.4 e na Figura 4.5, que são os relacionamentos não taxonômicos como "*affectsAssets*" entre as classes "*Threat*" e "*CIAProperties*", "*hasProblem*" entre as classes "*AttackCase*" e "*AttackCaseProblem*", *uses* entre as classes "*Attacker*" e "*Vulnerabillity*", entre outras.

Tabela 4.1 Parte das propriedades da ONTOID.

Classe	Propriedades	Descrição
AttackCaseSolution	hasAutor	Entidade que fornece a solução do ataque.
	dateOriginalRelation	Data em que a solução foi divulgada.
DomainSecurityConcepts	description	Descrição de cada conceito.
	informationSource	Fonte de informação de cada conceito.
	synonumous	O sinônimo de cada conceito.
SecurityMechanism	supplier	Entidade fornecedora do mecanismo
ComputerNetwork	computerQuantity	Quantidade de Computadores conectados na rede
	networkType	Tipo da rede (corporativa, pessoal)
	IPRange	É a faixa de números do endereço IP que identifica a rede.

4.5 Considerações finais

Apresentou-se neste capítulo, a ontologia proposta (ONTOID). Ela é uma ontologia de aplicação para IDSs e foi construída no editor de ontologias Protegé utilizando a linguagem OWL. A ONTOID tem como principais tarefas "detectar ataques" e "recomendar solução" para esses ataques detectados. Ela possui uma representação de casos que permitiu o desenvolvimento de um sistema de detecção de intrusão baseado em casos denominado CBR-IDS, e utilizado para sua avaliação, o qual será definido no capítulo seguinte. A Tabela 4.2 mostra um comparativo das ontologias dos trabalhos relacionados que foram estudados no capítulo 3 com a ontologia desenvolvida neste capítulo.

Tabela 4.2 Comparativo dos trabalhos relacionados com a ONTOID

Proposta	**Tipo de ontologia**	**Tarefas suportadas**	**Ferra-menta usada**	**Elementos que compõem**	**Lingua-gem usada**	**Aplica-ções desenvol-vidas**
Modeling Computer Attack: An Ontology for Intrusion Detection. (UNDERCOFFER, JOSHI e PINKSTON, 2003).	Aplicação	Detectar Ataques	Não definido	Classes, hierarquias, propriedades e relacionamentos não taxonômicos	DAMLe OIL	Não definido
An Ontology-based Approach to react to Network Attacks (BOULAHIA et. al 2009)	Aplicação	Especificar políticas de segurança	Protegé	Classes, relacionamentos não taxonômico, propriedades, instâncias e axiomas	OWL e SWRL	Não definido
Uma Ontologia para Sistemas de detecção de Intrusão em Web Services. (ROSA, 2011)	Aplicação	Detectar ataques	Protegé	Classes, propriedades, hierarquias, instâncias e axiomas	OWL	Protótipo de IDS
Ontologia de Aplicação para o Desenvolvimento de Sistemas de Detecção de Intrusão Baseado em Casos – ONTOID	Aplicação	Detectar ataques e fornecer solução	Protegé	Classes, relacionamentos não taxonômicos, propriedades, hierarquia e instâncias	OWL	CBR-IDS

5 ESTUDO DE CASO

O estudo de caso apresentado neste capítulo visa realizar uma avaliação preliminar da ONTOID através do protótipo de um IDS baseado em casos denominado CBR-IDS ("*Case Based Reasoning Intrusion Detection System*"), o qual utiliza a ontologia desenvolvida como BC (Base de Conhecimento). Pretende-se neste capítulo avaliar a efetividade da ONTOID e, para este fim, será avaliada a efetividade do CBR-IDS, pois uma forma de avaliar a efetividade de uma ontologia é avaliando a efetividade do sistema que a utiliza.

5.1 Implementação do CBR-IDS

O CBR-IDS utiliza o Raciocínio Baseado em Casos para recomendar soluções e detectar ataques a partir da identificação de ataques similares já representados na ONTOID. Ele foi desenvolvido em Prolog utilizando o ambiente *LPA Win-Prolog* (STEEL, 2005) e a ontologia especificada na linguagem *OWL* foi mapeada automaticamente para uma BC em Prolog utilizando a ferramenta *Thea* (VASSALIADIS, WIELLEMARKER e MUNGALL, 2009). Para povoar a ontologia com casos de ataques utilizou-se 11799 instâncias distribuídas, por ataques, conforme Tabela 5.2, retiradas do repositório NSL-KDD (REVATHI e MALATHI, 2013) apresentado na Seção 2.2.

Tabela 5.2 Conjunto de percepções instanciadas na ONTOID, classificadas por ataque

Nome do Ataque	**Quantidade de instâncias na ONTOID**
neptune	1579
guess_passwd	1226
warezmaster	939
satan	727
smurf	627
back	359
portsweep	156
ipsweep	141
nmap	73
pod	41
Demais ataques (land, worm, imap,etc)	5931

5.2 Funcionamento do CBR-IDS

A Figura 5.1, exibe em pseudocódigo a funcionalidade básica do CBR-IDS. Ele tem como entrada pacotes de dados de ataques, que representam uma sessão

de conexão, simulados a partir dos pacotes de dados do NSL-KDD, como o exemplo a seguir, o qual foi explicado na seção 2.2.

0, icmp, ecr_i, SF, 1032, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 80, 80, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 255, 255, 1.00, 0.00, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00.

Após a entrada desses pacotes de dados, o sistema cria uma instância na ONTOID representando uma conexão de rede, conforme ilustra a Figura 5.2, onde "*MonitoredPackage_1*" é o nome da instância gerada e os caracteres "SF", "0", "icmp", "ecr_i", "1032", "1", "1" e "0" são valores das propriedades "*flag*", "*duration*", "*protocol_type*", "*service*", "*src_bytes*", "*srv_count*", "*count*" e "*land*" respectivamente.

```
1.  Algoritmo  CBR-IDS
2.  Entrada Um conjunto de pacotes de dados do NSL-KDD
3.  Inicia
4.      Gera novo caso problema instanciando os pacotes de dados na ONTOID
5.          Para cada caso na ONTOID faça
6.              Calcula a similaridade entre o novo caso problema e os casos já existentes na
7.              ONTOID
8.              Se existe na ONTOID um caso similar ao novo caso problema faça
9.                  Seleciona o caso mais similar e sua solução
10.                 Recomenda a solução
11.             Fim se
12.         Fim para
13. Fim
```

Figura 5.1 Pseudocódigo da funcionalidade básica do CBR-IDS

PackageMonitored_1	
flag =	SF
duration =	0
protocol_type =	icmp
service =	ecr_i
src_bytes =	1032
srv_count =	1
count =	1
land =	0
...	

Figura 5.2 Exemplo de parte de um novo pacote monitorado e instanciado na ONTOID

Cada conjunto de pacotes de entrada representa um Novo Caso Problema (NCP). O sistema efetua o cálculo de similaridade do NCP com cada Caso Problema da Base de Conhecimento (KBCP) através das fórmulas 2 e 3 (MENDES, GIRARDI e LEITE, 2013), onde a similaridade total é igual ao somatório da similaridade entre cada atributo do NCP com o atributo correspondente do KBCP, sendo que a cada atributo é atribuído um peso que reflete sua importância na descrição do ataque.

$$sim\_case(NCP,KBCP) = \sum_{j=1}^{n} w_j * sim\_attribute(Cncp_j, Ckbcp_j) \quad (2)$$

$$onde, \sum_{j=1}^{n} w_j = 1$$

NCP – *New Case Problem*;

KBCP – *Knowledge Base Case Problem*;

j é o índice associado a cada atributo do problema utilizado na recuperação;

w_j é o peso atribuído ao j-ésimo atributo;

sim_attribute é a função de calcular a similaridade entre os valores do atributo do NCP e do KBCP. É dada pela Fórmula 3 a seguir:

$$sim\_attribute(Cncp_j, Ckbcp_j) = \frac{2 * \left| Cncp_j \cap Ckbcp_j \right|}{\left| Cncp_j \right| + \left| Ckbcp_j \right|} \quad (3)$$

$Cncp_j$ é o conjunto formado pelo(s) valor(es) do atributo do NCP, e por sua hierarquia de superconceitos, quando houver, por exemplo: $Cncp_{duration}$ = 0, onde 0 é o valor do atributo "*duration*" do NCP;

$Ckbcp_j$ é o conjunto formado pelo(s) valor(es) do atributo do KBCP e por sua hierarquia de superconceitos, quando houver, por exemplo $Ckbcp_{service}$ = 'http' onde, 'http' é o valor do atributo "*service*" do KBCP.

$|Cncp_j \cap Ckbcp_j|$ representa a quantidade de elementos que contém na interseção dos conjuntos $Cncp_j$ e $Ckbcp_j$, enquanto que $|Cncp_j| + |Ckbcp_j|$ representa a soma da quantidade de elementos do conjunto $Cncp_j$ com a quantidade de elementos do conjunto $Ckbcp_j$.

Na Tabela 5.2 tem-se a representação de um NCP e de um KBCP, com parte de seus atributos os respectivos valores de cada atributo, utilizados para o cálculo de similaridade.

Tabela 5.2 Representação de um NCP e de um KBCP, utilizando parte dos atributos de um problema.

Novo Caso Problema (NCP)		**Caso Problema da Base de Conhecimento (KBCP)**	
Propriedade	**Valor**	**Propriedade**	**Valor**
count	487	count	385
diff_srv_rate	0.00	diff_srv_rate	0.00
dst_bytes	0	dst_bytes	0
dst_host_count	255	dst_host_count	255
dst_host_diff_srv_rate	0.00	dst_host_diff_srv_rate	0.00
dst_host_rerror_rate	0.00	dst_host_rerror_rate	0.00
dst_host_same_src_port_rate	1.00	dst_host_same_src_port_rate	1.00
dst_host_same_srv_rate	1.00	dst_host_same_srv_rate	1.00
dst_host_serror_rate	0.00	dst_host_serror_rate	0.00
dst_host_srv_count	255	dst_host_srv_count	255
dst_host_srv_diff_host_rate	0.00	dst_host_srv_diff_host_rate	0.00
dst_host_srv_rerror_rate	0.00	dst_host_srv_rerror_rate	0.00
dst_host_srv_serror_rate	0.00	dst_host_srv_serror_rate	0.00
duration	0	duration	0
flag	SF	flag	SF
hot	0	hot	0
srv_count	487	srv_count	385
is_host_login	0	is_host_login	0
land	0	land	0
src_bytes	1032	src_bytes	520

Aplicando esses valores nas fórmulas 3 e 2, e considerando que todos os atributos tenham a mesma relevância, ou seja, o mesmo peso ($w_j = 1/20 = 0{,}05$) obtém-se: $sim\_case(NCP, KBCP) = 0{,}850$.

O CBR-IDS foi implementado de forma a considerar também que pesos diferentes sejam dados para atributos mais relevantes, os quais variam de ataque para ataque e são listados na Tabela 5.3. Ainda aplicando os dados da Tabela 5.2, nas fórmulas 3 e 2 e, levando em consideração que pesos maiores sejam dados para os atributos mais relevantes obtém-se $sim\_case(NCP,KBCP) \cong 0{,}851$. Às características mais relevantes deu-se peso aproximado de 0,0714 e às demais 0,0384. Consideremos que se trata de um ataque *smurf*. Suas características mais relevantes e as dos demais ataques, que foram utilizados como NCP no CBR-IDS, estão listadas na Tabela 5.3. Essas características foram definidas baseando-se em Olusola, Oladele e Abosede (2010).

Tabela 5.3 Atributos mais relevantes para alguns ataques

Nome do Ataque	**Atributos mais relevante**	**Quan-tidade**
neptune	count,diff_srv_rate,dst_host_count,dst_host_same_src_port_rate, dst_host_same_srv_rate,dst_host_serror_rate,dst_host_srv_serror_rate, flag,same_srv_rate,src_bytes srv_diff_host_rate,srv_serror_rate	12
guess_passwd	service,flag,dst_bytes,num_failed_logins	4
warezmaster	dst_bytes, duration	2
satan	diff_srv_rate, rerror_rate, service	3
smurf	diff_srv_rate,dst_bytes,dst_host_count,dst_host_same_src_port_rate, dst_host_srv_serror_rate ,logged_in,protocol_type, same_srv_rate ,serror_rate,service,src_bytes	11
back	dst_bytes, src_bytes	2
portsweep	srv_rerror_rate	1
ipsweep	dst_host_same_src_port_rate	1
nmap	src_bytes	1
pod	wrong_fragment	1

Após realizar o cálculo de similaridade, o CBR-IDS recupera casos que tenham similaridade maior que 0,9 (ponto de corte obtido experimentalmente) com o NCP. Se não houver casos similares na ONTOID, o sistema encerra, pois não há nada a fazer. Caso contrário o CBR-IDS seleciona a solução do caso mais similar e a recomenda ao administrador da rede. Essa solução é recomendada fornecendo o nome do ataque que corresponde ao caso problema recuperado, a similaridade entre eles e a descrição da medida a ser tomada para solucionar o problema, conforme ilustra a Figura 5.3.

```
| ?- recomenda_solução.

# Abolishing c:\cbr-ids\ontoid.pl

# 0.982 seconds to consult c:\cbr-ids\ontoid.pl

Novo Caso Problema:  MonitoredPackage_44

Problema mais similar encontrado:  AttackProblem_1004

Ataque Relacionado: smurf

Similaridade: 0.9512178

Solução: Configurar atualizacao no roteador ou, eventualmente,
         no trafego de firewall.
```

Figura 5.3 Exemplo de uma solução recomendada pelo CBR-IDS

Para fins de avaliação da ONTOID, através do CBR-IDS, utilizou-se um conjunto de 50 NCPs, ou seja, 50 novas conexões do conjunto NSL-KDD escolhidas aleatoriamente (diferentes das utilizadas para povoar a BC). A ONTOID foi instanciada com todos os outros casos de ataque desse mesmo conjunto, sendo eles um total de 11799 e a efetividade das soluções recomendadas foi avaliada através da medida de *precision* e *recall* tradicional da área de recuperação de informação (SALTON e BUCKLEY, 1987). A *precision* é calculada como a razão entre o número de soluções relevantes recuperadas e o número total de casos recuperados, e o *recall* é calculado como a razão entre o número de soluções relevantes recuperadas e o número total de casos relevantes da BC. Assim, para cada NCP do CBR-IDS foi verificado, do conjunto de casos recuperados, quais eram relevantes, e calculado o valor de *precision* e *recall*. Esses resultados são apresentados na próxima seção e discutidos na Seção 5.4.

Neste estudo de caso realizou-se duas experiências cujos dados para cada uma estão listados na Tabela 5.4: na primeira foi considerado que os atributos de cada caso possuem a mesma relevância, ou seja, designou-se o mesmo peso (0,02439) para todos os atributos; na segunda experiência considerou-se os atributos mais relevantes, designando pesos maiores para eles e pesos menores para atributos menos relevantes (demais atributos). A forma de escolha dos pesos é apresentada na Seção 5.4. Em ambas as experiências utilizou-se os pontos de corte 0,7, 0,8 e 0,9 com a finalidade de comparar os resultados e encontrar o ponto de corte mais efetivo para ser utilizado no CBR-IDS.

Tabela 5.4 Dados utilizados nas experiências

Experiências	Pontos de Corte	Ataque	Peso	
			Atributos mais relevantes	Demais atributos
Exp. 1	0.7, 0.8, 0.9	Todos os ataques da base, descritos na Tabela 5.2.	0,02439, para cada um dos 41 atributos.	
Exp. 2	0.7, 0.8, 0.9	neptune	0,03214	0,02037
		guess_passwd	0,037	0,02378
		warezmaster	0,035	0,02384
		satan	0,0333	0,02368
		smurf	0,0333	0,02068
		back	0,035	0,02384
		portsweep	0,03000	0,02425
		ipsweep		
		nmap		
		pod		

5.3 Resultados

No cálculo da *precision* e do *recall*, considera-se como “casos relevantes”, os casos que especificam o mesmo ataque que o caso do NCP e, como casos recuperados, todos os casos que obtiveram o valor de similaridade, com o NCP, maior ou igual 0,9. Esse ponto de corte foi obtido pela realização de experimentos (ver seção 5.4) realizados antes das experiências 1 e 2.

Na primeira experiência considerou-se o mesmo valor de peso para todos atributos e foram utilizados 3 pontos de corte (0,7; 0,8; 0,9) e na segunda experiência atribuiu-se pesos maiores para os atributos mais relevantes com os mesmos pontos de corte da primeira experiência. As tabelasTabela 5.5 eTabela 5.6 tem-se as médias de *precision* e *recall* para ambas as experiências com os três pontos de corte já citados. Onde, a Tabela 5.5 apresenta os resultados da experiência 1 a Tabela 5.6 apresenta os resultados da experiência 2.

Tabela 5.5 Médias de *Precision* e *Recall* considerando o mesmo peso para todos os atributos

Ponto de Corte	Média *Precision*	Média *Recall*
0,7	0,214427208	0,782832622
0,8	0,609315146	0,631830136
0,9	0,927620708	0,290499868

Tabela 5.6 Médias de *precision* e *recall* considerando pesos maiores para os atributos mais relevantes

Ponto de Corte	Média *Precision*	Média *Recall*
0,7	0,239052998	0,762598718
0,8	0,63327438	0,59731751
0,9	0,943294276	0,29552032

Na Tabela 5.7 tem-se os resultados do valor de precision e *recall* com o ponto de corte 0,9 na segunda experiência para parte dos NCPs utilizados na avaliação. Nela está representado o nome da instância gerada pelo CBR-IDS na ONTOID para cada NCP, o nome do ataque (NA) correspondente a esse NCP e os valores de *precision* e *recall* que ele obteve. O NCP "*MonitoredPackage_17*" é o exemplo de um resultado obtido que especifica o ataque *Satan*. Ele obteve 97 casos recuperados com valor de similaridade maior ou igual a 0,9. Desses casos recuperados 96 correspondem a um ataque *Satan*. Nesse caso tem-se uma *precision* de aproximadamente 99% e *recall* de aproximadamente 13%. Na base de casos o *Neptune* possui 727 conforme descrito na Tabela 5.2.

Tabela 5.7 Medida de *precision* da recuperação de casos para alguns NCPs utilizados e recuperados na avaliação

NCP	NA	*Precision*	*Recall*
MonitoredPackage_2	Neptune	1	0,0145662
MonitoredPackage_8	Guess_paswd	1	0,0422421
MonitoredPackage_13	Warezsmaster	1	0,0900424
MonitoredPackage_17	Satan	0.9896907	0.1320495
MonitoredPackage_21	Smurf	1	0,7527911
MonitoredPackage_26	Back	1	0,0779944
MonitoredPackage_32	Portsweep	0,9661017	0,3653846
MonitoredPackage_36	Ipsweep	0,6470588	0,3120567
MonitoredPackage_41	Nmap	1	0,4109589
MonitoredPackage_46	Pod	1	0,5853659

Dos 50 casos de ataque utilizados como NCP, 49 obtiveram soluções recomendadas. O CBR-IDS obteve uma *precision* geral aproximada de 93% e *recall* de 30% levando em consideração os resultados da experiência 2 como ilustrado na linha 4 da Tabela 5.6. Esses resultados levam a considerar como efetiva as soluções recomendadas pelo CBR-IDS baseado na ONTOID. Assim, este estudo de caso mostra que a atual conceitualização da área da Segurança da Informação na

ONTOID é efetiva, embora maiores experiências sejam necessárias para incrementar a *precision* e o *recall* das recomendações.

5.4 Discussão

O estudo de caso realizado teve como objetivo avaliar a efetividade do sistema desenvolvido para que se pudesse concluir a efetividade da ontologia proposta, ou seja, calcular a *precision* e *recall* na recuperação dos ataques e soluções mais relevantes da ONTOID para os novos casos problema (casos de ataque).

Antes de serem realizadas as experiências 1 e 2 foi realizada a experiência para determinar o ponto de corte ideal para a obtenção de uma efetividade equilibrada em termos de *recall* e *precision*. O ponto de corte define o valor mínimo de similaridade necessário para que ao menos uma solução seja recuperada pelo sistema. Há a necessidade de se identificar o ponto de corte ideal, pois ao utilizar pontos de corte com valores muito altos, o sistema pode deixar de recuperar solução para alguns ataques, enquanto que pontos de corte muito baixos podem fazer com que o sistema recomende soluções com valores de similaridade muito baixos e, portanto, incorretas. Assim, essa primeira experiência visa encontrar esse ponto de corte ideal, isto é, o maior ponto de corte possível que retorne soluções com altos valores de *precision* para a maioria dos ataques identificados. Os pontos de corte 0.7, 0.8 e 0.9, utilizados nas experiências 1 e 2, foram escolhidos em testes realizados previamente, nos quais obtém-se muitas soluções irrelevantes com pontos de cortes inferiores a 0.7 e muitas soluções relevantes não foram recuperadas com pontos de corte superiores a 0.9.

Na fórmula do cálculo de similaridade para a experiência 1, considerou-se que todos os atributos dos novos casos-problema e dos casos problema da ONTOID tivessem pesos iguais, e como a soma dos pesos *w* dos atributos de cada caso tem que ser igual a 1 (um), dividiu-se 1 (um) pela quantidade total dos atributos de cada caso, sendo o resultado dessa divisão, o valor do peso de cada atributo. Já na experiência 2, foi considerado que atributos mais relevantes tenham pesos maiores que os demais atributos de maneira que a soma dos pesos de todos os atributos seja igual a 1 (um). Essa distribuição foi realizada dando um peso aproximado de

0,03 para os mais relevantes e 0,02 para os menos relevantes, variando as demais casas decimais (a partir da 3ª) para cada ataque.

Analisando a Tabela 5.5 e a Tabela 5.6 na seção dos resultados, pode-se observar que, em ambas as experiências realizadas, com o ponto de corte 0,7 a *precision* é muito baixa, isso significa que o sistema recuperou muitas soluções irrelevantes, enquanto que os valores de *recall* foram bons. Com o ponto de corte 0,9 a *precision* foi muito boa, significando que quase todas as soluções recuperadas são relevantes, porém o *recall* foi muito baixo, isso implica que o sistema deixou de recuperar muitos casos da BC considerados relevantes. Já com o ponto de corte 0,8 pode-se observar que tanto a *precision* quanto o *recall* se mantêm equilibrados, porém nem bom, nem ruim. Na segunda experiência obteve-se maiores valores de *precision* em todos os pontos de corte utilizados em relação aos valores obtidos com a primeira experiência.

Para o problema de recomendações a ataques de redes, o valor de *precision* é mais importante do que de o de *recall*, uma vez que é mais interessante que o sistema ofereça poucas soluções ao usuário (administrador de rede), de preferência apenas uma, e que as mesmas sejam efetivas, evitando soluções de tentativa-erro que são mais custosas tanto no esforço quanto no tempo gasto, fatores críticos para a área de Segurança da Informação. Por isso, considera-se o ponto de corte 0,9 o mais adequado para o domínio abordado.

Ainda observando os resultados das experiências 1 e 2 nas tabelas Tabela 5.5 e Tabela 5.6 da seção anterior, pode-se perceber que os valores das duas experiências são muito próximos. Porém, é possível notar claramente que a *precision* obtida com a segunda experiência é superior à primeira. Assim, conclui-se que o sistema é mais efetivo quando se utiliza um ponto de corte de 0,9 e atribui-se pesos maiores aos atributos mais relevantes.

5.5 Considerações finais

Este capítulo apresentou um estudo de caso do CBR-IDS para avaliação da efetividade da ONTOID. É importante ressaltar que a detecção utilizando CBR é similar à abordagem baseada em assinaturas, levando em conta que casos de ataques estão pré-armazenados na BC. Descreveu-se inicialmente a maneira como

o CBR-IDS foi implementado, destacando a ferramenta usada e a linguagem na qual foi escrita. Em seguida apresentou-se o funcionamento do CBR-IDS através de um pseudocódigo e foi realizada uma descrição ilustrativa do seu funcionamento com um exemplo do domínio em questão. A efetividade do sistema foi medida por meio do cálculo da *precision* e do *recall* definidos por Salton e Buckley (1987) que são dados em função dos casos recuperados, dos casos recuperados relevantes e dos casos relevantes da BC. Para a recuperação das soluções utilizou-se o modelo de similaridade adotado por Mendes, Girardi e Leite (2013). Nesse estudo de caso foi possível mostrar que a ONTOID possui uma boa efetividade na recuperação de casos de ataques e de soluções para esses casos, pois o sistema que a utiliza (CBR-IDS) obteve uma *precision* de aproximadamente 94% na recuperação de casos relevantes, sendo considerado muito bom.

6 CONCLUSÕES

Neste trabalho apresentou-se uma análise do estado da arte dos problemas de ataques à Segurança da Informação, seus tipos, as propriedades nas quais esses ataques atingem e alguns mecanismos de segurança que servem para proteger os sistemas computacionais, desses ataques. Dentre esses mecanismos, estudou-se e apresentou-se os Sistemas de Detecção de Intrusão (*IDS's*), bem como seus tipos, funções que realizam, abordagens e ontologias criadas para o desenvolvimento dos mesmos. A necessidade de realizar um estudo dos IDS's se deu pela proposta deste trabalho que é descrita no parágrafo seguinte.

A proposta deste trabalho foi uma ontologia de aplicação cujo domínio é na área de Segurança da Informação e as tarefas são baseadas nas funções dos Sistemas de Detecção de Intrusão. A ontologia desenvolvida possibilitou a criação de uma aplicação para o desenvolvimento de IDS's baseados em casos. Essa ontologia é denominada ONTOID e tem como principais vantagens: fornecer o reuso, tanto da própria ontologia, como do sistema que a utiliza para a construção de novas aplicações no domínio de Segurança da Informação, que tem como consequência a rapidez e baixo custo no desenvolvimento de novos sistemas; o compartilhamento de conhecimento e informações que são comuns a pessoas e a sistemas de *software*.

A ontologia proposta foi avaliada através de um estudo de caso no desenvolvimento de Sistemas de Detecção e Intrusão, com a implementação do protótipo de um IDS baseado em casos, denominado CBR-IDS, o qual é capaz de detectar ataques a partir de um conjunto de ataques similares pré-armazenados na ONTOID e recomendar soluções para um administrador de rede de computadores. O CBR-IDS detecta esses ataques realizando uma análise de similaridade entre os novos casos que chegam no sistema e os casos que já existem na BC.

6.1 Resultados e contribuições

O trabalho de pesquisa realizado obteve as seguintes contribuições:

- Uma ontologia de domínio para o domínio Segurança da Informação que permite o reuso e compartilhamento de conhecimento para o desenvolvimento de ontologias de aplicação voltadas para esse domínio;

- Uma ontologia de aplicação para Sistemas de Detecção de Intrusão, a qual incorporou conceitos da ontologia de domínio;
- O protótipo de um sistema de detecção de intrusão também foi uma contribuição relevante deste trabalho. Ele é capaz de acessar a ontologia interpretando seus conceitos, relacionamentos, instâncias e propriedades para fornecer informações ao usuário. O CBR-IDS, além de coletar informações, também é capaz de inserir informações na ONTOID, criando instâncias de novos casos de ataques;
- Uma análise do estado da arte da Segurança da Informação, dos sistemas de detecção de intrusão e de ontologias para abordar problemas de ataques à Segurança da Informação.

6.2 Limitações

O sistema desenvolvido é efetivo de acordo com as condições apresentadas até então, ou seja, quando o ponto de corte utilizado é 0,9 e os pesos atribuídos aos atributos são maiores para os mais relevantes. Embora este trabalho tenha alcançado o objetivo principal, que foi mostrar, através de uma avaliação, que a ontologia desenvolvida é efetiva, há algumas limitações que, se exploradas, podem melhorar os resultados obtidos pelo sistema. A principal limitação deste trabalho está no tipo de atributos adotado na realização da avaliação, pois o conjunto de dados NSL-KDD possui atributos de valores contínuos e atributos de valores discretos. Porém, a medida de similaridade utilizada não se adequa aos casos de valores contínuos. Para superar tal limitação é necessário que seja realizado uma discretização nos valores de atributos contínuos ou que se utilize uma medida de similaridade capaz de suportar tanto atributos com valores contínuos quanto discretos.

6.3 Publicação

MENESES, R., LEITE, A., GIRARDI, R. **Ontologia de Aplicação para o Desenvolvimento de Sistemas de Detecção de Intrusão Baseado em Casos**. In: 10ª Conferencia Ibérica de Sistemas y Tecnologías de Información, Aveiro, 2015, apresentou uma versão resumida de todo o trabalho aqui desenvolvido. **Qualis Capes: B4**. (Artigo Aceito).

Nesta publicação apresentou-se um resumo de todo o trabalho com avaliação prévia realizada apenas com o primeiro experimento, ou seja, o experimento em que se considerou que os atributos de cada caso possuem a mesma relevância, ou seja, o mesmo peso para todos os atributos.

6.4 Trabalhos futuros

Diante das contribuições e dos resultados obtidos foi possível identificar os seguintes trabalhos futuros:

- Construir um agente de *software* com aprendizado, complementando o sistema desenvolvido neste trabalho, para melhorar o apoio à tomada de decisões e o desempenho do sistema, pois os agentes de *software* possuem autonomia, ou seja, são capazes de responder questões que não foram explicitamente determinadas e, a partir do aprendizado, pode-se generalizar regras, diminuindo a quantidade de atributos a ser comparado com os pacotes que entram no sistema, o que torna o sistema mais rápido. A aprendizagem também contribui para que o sistema seja capaz de se adaptar às mudanças do ambiente, o que é essencial para os ambientes dinâmicos como as redes de computadores;

- Incrementar a ontologia com os conceitos secundários do domínio de Segurança da Informação, conceitos esses que foram discutidos no Capítulo 2 porém, que não foram modelados na ONTOID, para que outras tarefas possam ser realizadas como pesquisas a respeito do domínio.

REFERÊNCIAS

ABOU-EL-KALAM, A. et al. “Organization based access control”. **In IEEE 4th International Workshop on Policies for Distributed Systems and Networks (POLICY 2003)**, Lake Como, Italy, 2003.

ALVES, C. B. **Segurança da Informação Vs. Engenharia Social:** como se proteger para não ser mais uma vítima. 2010. Monografia (Curso de Sistemas de Informação) – Centro Universitário do Distrito Federal, Brasília, 2010.

ARORA, S.; BAWA, R. A. Review on Intrusion Detection System to Protect Cloud Data. **International Journal of Innovations e Advancement in Computer Science**, v. 3, n. 5, p.30-34, 2014.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 27001**: sistemas de gestão de Segurança da Informação - requisitos, Rio de Janeiro, 2006.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 27002**: código de prática para a gestão de Segurança da Informação, Rio de Janeiro, 2005.

BENMOUSSA, H.; EL KALAM, A. A. E.; OUAHMAN, A. A. Towards a new intelligent generation of intrusion detection system. **Proceedings of the 4th Edition of National Security Days** (JNS4), p.1-5, 2014.

BOULAHIA, C. N. et al. An ontology-based approach to react to network attacks. **International Journal of Information and Computer Security**, v.3, n. 3, p. 280-305, 2009.

BRANDÃO, Antonio; MARTIMIANO, Luciana; MOREIRA, Edson. O Uso de Ontologia em Alertas de Vulnerabilidades. **22º Simpósio Brasileiro de Redes de Computadores**, p. 75-86, 2004.

BRASIL. Tribunal de Contas da União. **Boas práticas em Segurança da Informação**. Tribunal de Contas da União. 2. ed. Brasília: TCU, Secretaria de Fiscalização de Tecnologia da Informação, 2007.

CERT.br. **Cartilha de Segurança para Internet**. 2nd ed., São Paulo: CGI.br, 2012.

COZZOLINO, M. F. **Detecção de Variantes Metamórficas de Malware por Comparação de Códigos Normalizados**. 2012. 60f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade de Brasília, Brasília, 2012.

CRUCIOL, L. L. B. **Modelagem de Apoio à Decisão para o Problema de Espera no Ar Utilizando Sistemas Multiagente e Aprendizagem por Reforço**. Dissertação (Mestrado em Informática) – Universidade de Brasília, Brasília, 2012.

CRUZ, Anderson. **Um Sistema Multiagente para Geração Automática de uma Rede Bayesiana para Análise de Riscos de Tecnologia da Informação**. Dissertação (Metrado em Computação Aplicada). Universidade do Vale do Rio dos Sinos. São Leopoldo, 2011.

DEBAR H, CURRY D.; FEINSTEIN B. The Intrusion Automatic Text retrieval. **Information Processing & Management**, v. 24, n. 5Detection Message Exchange Format (IDMEF). **IETF Request for Comments 4765**, 2007.

DEBAR, H. Intrusion Detection Systems: Introduction to Intrusion Detection and Analysis, Security and Privacy in Technologies. **Nato Science Series III Vol. 193, IOS Press**, edited by Borka Jerman Blazic, Wolfgang Schneider and Tomaz Klobular, 2004.

DOULIGERIS, C.; MITROKOTSA, A. DDoS attacks and defense mechanisms: classification and state-of-the-art. **Computer Networks**, v. 44, n. 5, p. 643-666, 2004.

ELSHOUSH, H. T.; OSMAN, Izzeldin Mohamed. Alert correlation in collaborative intelligent intrusion detection systems – A survey. **Applied Soft Computing**, v. 11, n. 7, p. 4349-4365, 2011.

FARIA, C. G. **Um Processo Independente de Domínio para o Povoamento Automático de Ontologias a Partir de Fontes Textuais**. Tese (Doutorado em Engenharia de Eletricidade). Universidade Federal do Maranhão. Maranhão, 2013.

GIRARDI, M. R. An Analysis of the Contributions of the Agent Paradigm for the Development of Complex Systems. In: **ISAS-SCI**, Vol 1, p. 388-393, 2001.

GIRARDI, M. R. Engenharia de software baseada em agentes. **In: Congresso Brasileiro de Ciência da Computação**, Edição Especial 39, 2004.

GIRARDI, R.; LEITE, A. Knowledge Engineering Support for Agent-Oriented Software Reuse. In: **M. Ramachandran, ed. Knowledge Engineering for Software Development Life Cycles**. Hershey, PA: IGI Global, pp. 177–195, 2011.

GRUBER, T. Toward Principles for the Design of Ontologies used for Knowledge sharing. **International Journal of Human-Computer Studies**, p. 907-928, 1995.

Gualberto, E. S. et al. Proposição de uma Ontologia de Apoio à Gestão de Riscos de Segurança da Informação. **Revista Brasileira de Sistemas de Informação**, v. 6, n.1, pp.30-43, 2012.

Guarino, N. **Formal Ontology in Information Systems**: Proceedings of the First International Conference (FOIS'98) 1st ed., Trento, Italy: IOS press, 1998.

HENDLER, J. **DARPA Agent Markup Language+Ontology Interface Layer**, 2001. Disponível em: <http://www.daml.org/2001/03/daml+oil-index. Acesso em: 1 de Março de 2014.

ISCX. **Nsl-kdd data set for network-based intrusion detection systems**, 2009. Disponível em: < http://nsl.cs.unb.ca/NSL-KDD/ >. Acesso em: junho de 2014.

JÚNIOR, B. L. M. et al. Proteja o maior bem da sua empresa, a informação, com: política de Segurança da Informação. **Publicar Revista Digital**, p. 57, 2007.

KABIRI P.; GHORBANI A. A. Research on intrusion detection and response: A survey. **International Journal of Network Security**, v. 1, n. 84, p.102, 2005.

KOPENA, J.; REGLI, W. DAMLJessKB: A Tool for Reasoning with the Semantic Web. In: D. Fensel, K. Sycara, e J. Mylopoulos, eds. The Semantic Web - ISWC. **Lecture Notes in Computer Science**. Sanibel Island, USA: Springer Berlin Heidelberg, p. 628–643, 2003

KUROSE, J. F. and ROSS, K. W. **Redes de Computadores e a Internet**: Uma abordagem top-down. Addison Wesley, 4 edition, 2008.

L. T. Yang, and M. K. Denko, editors, **Autonomic Computing and Networking**, p. 355–380. Springer US, 2009.
LAUDON, K.; LAUDON, J. P. **Management Information Systems**: organization and technology. 4. ed. New Jersey: Prentice-Hall, 1996.

LEITE, A. C. **MADAE-Pro: Um processo baseado no conhecimento para Engenharia de Domínio e de Aplicações Multiagente**. Dissertação (Mestrado em Engenharia de Eletricidade). Universidade Federal do Maranhão, 2009.

LEITE, A.; GIRARDI, R. A Case-Based Reasoning Architecture of a Hybrid Software Agent. In: Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014. **IEEE/WIC/ACM International Joint Conferences on IEEE**, pp. 79–86, 2014.

LI, Yabo. Design of the intrusion detection system based on multi-agents in the e-commerce system. In: Computer Science and Information Technology, 2008. ICCSIT'08. **International Conference on. IEEE**, 2008. p. 760-764.

LIAQAT, I. Vulnerabilities in Security Products for Computers 2006. Tese de Doutorado (Linkoping University). Tekniska Hogskolan, 2009.

LIMA, C. F. L. **Agentes inteligentes para detecção de intrusos em redes de computadores**. Dissertação (Mestrado em Engenharia de Eletricidade). Universidade federal do Maranhão. Maranhão, 2002.

LIMA, L. F. F. M. **Percepção de Segurança de Sistemas de Informação e sua relação com a qualidade percebida de serviços, perfil de liderança e perfil dos seguidores, entre as Diretorias do Inmetro**. Dissertação (Mestrado Profissional em Sistemas de Gestão). Universidade Federal Fluminense, Niterói, 2006.

MARCELO, A.; PEEIRA, M. A arte de Hackear Pessoas. Brasport, Rio de Janeiro, 2005.

MARCIANO, J. L. P. **Segurança da Informação: uma abordagem social**. 212 f. Tese (Doutorado em Ciência da Informação e Comunicação). Universidade de Brasília, Brasília, 2006.

MAZIERO, Carlos A. **Sistemas Operacionais**: Conceitos e Mecanismos. Universidade Tecnológica Federal do Paraná. Paraná, 2013.

MCGUINNESS, D. L.; HARMELEN, F. V. **OWL Web Ontology Language**,2004. Disponível em: <http://www.w3.org/TR/owl-features>. Acesso em: 15 de Maio de 2014.

MENDES, W. **Arquitetura Baseada em Ontologias de um Agente RBC**. Dissertação (Mestrado em Engenharia de Eletricidade). Universidade Federal do Maranhão, 2013.

MENDES, W.; GIRARDI, R.; LEITE, A. Arquitetura Baseada em Ontologias de um Agente RBC. In Information Systems and Technologies (CISTI), 2013. **8th Iberian Conference on IEEE**. IEEE, p. 1–6, 2013.

MORAES, F. A. L. **Segurança e confiabilidade em IDS baseados em agentes**. Dissertação (Mestrado em Engenharia de Eletricidade). Universidade Federal do Maranhão, 2009.

MORITZ, G. O.; PEREIRA, M. F. **Processo Decisório. SEAD/UFSC**, Santa Catarina 2006.

NAKAMURA, E. **Segurança em de redes em ambientes cooperativos**. São Paulo: Novatec, 2007.

NOY, N. F.; MCGUINNESS, D. L. Ontology Development 101: A Guide to Create Your First Ontology. **Knowledge Systems Laboratory Technical Report KSL-01-05**, Stanford University. 25p, 2001.

OLUSOLA, A. A.; OLADELE, A. S.; ABOSEDE, D. O. Analysis of KDD'99 Intrusion detection dataset for selection of relevance features. In: **Proceedings of the World Congress on Engineering and Computer Science**, p. 20-22, 2010.

PEREIRA, L. R. **Segurança e Proteção de Computadores Pessoais na Internet**. Monografia (Tecnólogo em processamento de Dados). Faculdade de Tecnologia de São Paulo, São Paulo, 2012.

PEREIRA, P. J. F. Segurança da informação digital. **Cadernos BAD**, v. 2005, n. 1, p. 66-80, 2005.

PINHEIRO, J. M. S. Ameaças e Ataques aos Sistemas de Informação: Prevenir e Antecipar. **Cadernos UniFOA**, Rio de Janeiro, ed. 5, p. 11-21, 2007.

PINHEIRO, J. M. S. Os benefícios da política de Segurança Baseada na Avaliação de Riscos e na Integração de Ferramentas: Prevenir e Antecipar. **Cadernos UniFOA**, Rio de Janeiro, ed. 4, p. 28-34, 2007.

REVATHI, S.; MALATHI, A. A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection. **International Journal of Engineering Research e Technology (IJERT)**, v. 2, n.12, p.1848–1853, 2013.

ROSA, T. M. **Uma Ontologia para Sistema de Detecção de Intrusão em Web Services**. Dissertação (Mestrado em Informática). Pontifícia Universidade Católica do Paraná. Curitiba, 2011.

RUSSEL, S. NORVIG, P. **Artificial Inteligence**: A Modern Approach, Prentice-Hall,2009.

SALTON, G. e BUCKLEY, C. Term Weighting Approaches in, p. 513-523, 1988.

SANTOS, V. **Sistemas de Detecção de Intrusões (IDS – Intrusion Detection Systems) usando unicamente softwares Open Source**, 2010. Disponível em <http://www.seginfo.com.br/sistemas-de-deteccao-de-intrusoes-ids-intrusion-detection-systems-usando-unicamente-softwares-open-source/>. Acesso em: 13 de abril 2014.

SAWANT, T. S.; ITKAR, S. A. A Survey and Comparative Study of Different Data Mining Techniques for Implementation of Intrusion Detection System. **International Journal of Current Engineering and Technology**, India, v. 4, n. 3, p. 1288-1291, 2014.

SCARFONE, K.; MELL, P. **Guide to Intrusion Detection and Prevention Systems (IDPS)**. Recommendations of the National Institute of Standards and Technology, National Institute of Standards and Technology, Gaithersburg, p. 1-127, 2007.

SILVA, R.; MAIA, M. Redes Neurais Artificiais Aplicadas a Detecção de Intrusos em Redes TCP/IP. In: **Simpósio de Segurança da Informação, Instituto Tecnológico de Aeronáutica (ITA**). São José dos Campos, SP, 2004.

SILVA, R.; MAIA, M. Redes Neurais Artificiais Aplicadas a Detecção de Intrusos em Redes TCP/IP. **In Simpósio de Segurança da Informação, Instituto tecnológico de Aeronáutica (ITA)**. São José dos Campos, SP, 2004.
SNAPP, S. R. et al. DIDS (Distributed Intrusion Detection System) - Motivation, Architecture, and An Early Prototype. **In Proceedings of the 14th National Computer Securit Conference**, p. 167-176, 1991.

SOUZA, V. E. S.; MYLOPOULOS, J. Monitoring and diagnosing malicious attacks with autonomic software. In: Conceptual Modeling-ER 2009. **Springer Berlin Heidelberg**, 2009. p. 84-98.

STALLINGS, W. **Cryptography and Network Security**: Principles and Pratice. 5th ed. Prentice Hall, 2010.

STANFORD. **The Protégé Ontology Editor and Knowledge Acquisition System**. Disponível em: <http://protege.stanford.edu>. Acesso em: maio de 2014.

STEEL, B. **Chimera Agents for WIN-PROLOG**, 2005. Disponível em: <http://www.lpa.co.uk/chi.htm>. Acesso em: maio de 2014.

STRASSBURG, U. et al. A Importância do sistema de informação contábil como fonte de informações para tomada de decisões. **VI Seminário do Centro de Ciências Sociais Aplicadas de Cascavel**. Cascavel, PR, 2007.

TAVALLAEE, M. et al. A detailed analysis of the KDD CUP 99 data set. In: **2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications. IEEE**, pp. 1–6, 2009.

TELES, A. S. **AutonomicSec: Um mecanismo Autonômico para Segurança de Redes Baseado em Decepção**. Dissertação (Mestrado em Engenharia de Eletricidade). Universidade Federal do Maranhão, 2012.

TELES, A. S.; MENDES, J. P. M.; ABDELOUAHAB, Z. Autonomic computing applied to network security: A survey. **Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), November Edition**, p. 7-14, 2011.

TEODORO, P. G. et al. Anomaly-based network intrusion detection: Techniques, systems and challenges. **Computers e security**, v. 28, n. 1, p. 18-28, 2009.

UNDERCOFFER, J.; JOSHI, A.; PINKSTON, J. Modeling computer attacks: An ontology for intrusion detection. In: Recent Advances in Intrusion Detection. **Springer Berlin Heidelberg**, p. 113-135, 2003.

VASALIADIS, V.; WIELEMAKER, J.; MUNGALL, C. Processing OWL2 Ontologies using Thea: An Application of Logic Programming. **OWLED**, vol. 529, 2009.

VERGARA, Jorge E. et al. Use of ontologies for the definition of alerts and policies in a network security platform. **Journal of Networks**, v. 4, n. 8, p. 720-733, 2009.

VIEIRA, F. L. **Um Sistema Multiagente para Apoio as Decisões ao Processo de Licitação Pública**. Dissertação (Mestrado em Eng. de Eletricidade). Universidade Federal do Maranhão, 2013.

WANGENHEIM, C.; WANGENHEIM, A. **Raciocínio Baseado em Casos**, 1. ed. Barueri-SP: Manole, 2003.

WHITMAN, M. E. In defense of the realm: understanding the threats to information security. **International Journal of Information Management**, v. 24, n. 1, p. 43-57, 2004.

WOOLDRIDGE, M. **An Introduction to Multiagent Systems.** 2nd ed., John Wiley & Sons Ltd, 2009.

ZHANG, R. A Model of Collaborative Intrusion Detection System Based on Multi-agents. In: Computer Science e Service System (CSSS), 2012. **International Conference on IEEE**, 2012. p. 789-792.

ZHANG, R. et al. Multi-agent based intrusion detection architecture, in Proceedings of 2001. **IEEE International Conference on Computer Networks and Mobile Computing**, pp. 494–501, Oct. 2001.

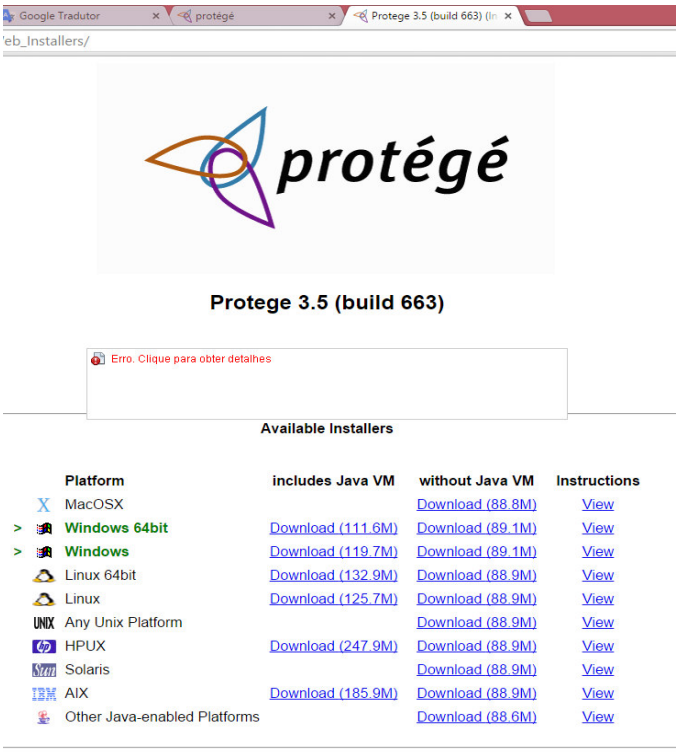
ZSEBY, T.; PFEFFER, H.; STEGLICH S. Concepts for self-protection. In Y. Zhang,

ANEXO - TUTORIAL PARA INSTALAÇÃO DA FERRAMENTA THEA

Thea é uma biblioteca Prolog para gerar e manipular conteúdo OWL ("*Web Ontology Language*"), onde a mesma usa o analisador de Web Semântica do SWI-Prolog a fim de converter as ontologias do padrão OWL em fatos Prolog do tipo *.pl. Para isso, o primeiro passo seria a criação de ontologias no padrão OWL através do *software* Protégé.

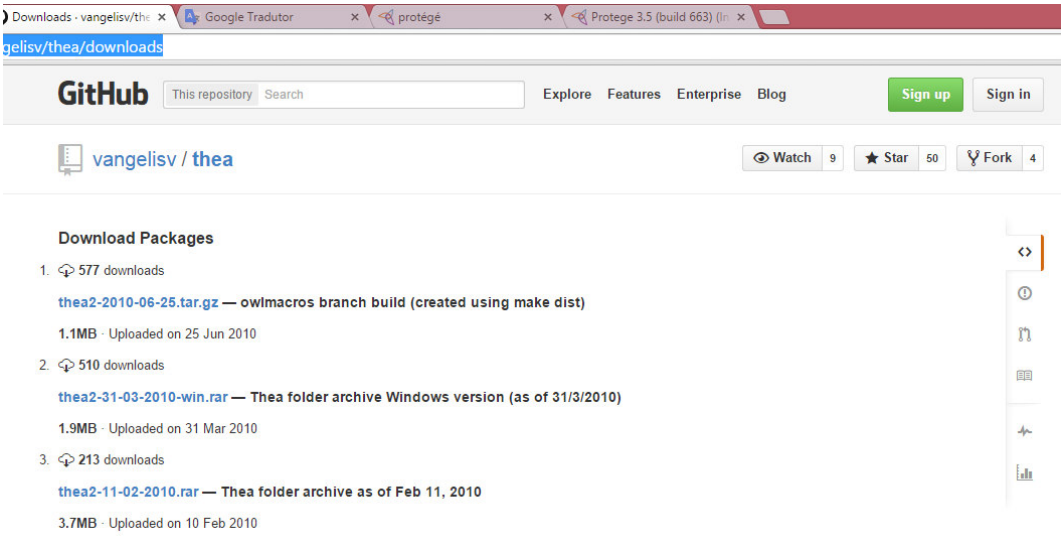
A versão do Protégé utilizada neste tutorial é a 3.5 (escolher o padrão do Sistema Operacional, se 32bits ou 64bits), encontrado no link: http://protege.stanford.edu/download/protege/3.5/installanywhere/Web_Installers/

Como o Protégé foi criado como uma aplicação Java, a máquina JVM (Java Virtual Machine) deverá estar instalada.



A biblioteca THEA não possui um instalador, e sim um arquivo compactado com toda a estrutura de arquivos pronta para a implantação da mesma, onde deverá ser baixada no link: https://github.com/vangelisv/thea/downloads

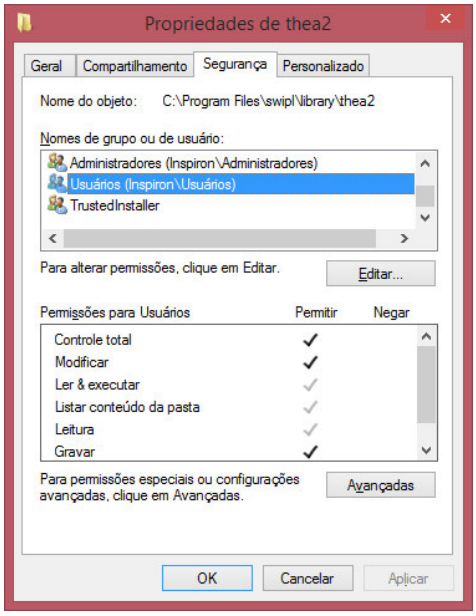
E aqui será escolhida a versão de acordo com o Sistema Operacional.



Após baixar o arquivo compactado, o mesmo deverá ser descompactado com a estrutura intacta em uma pasta de fácil acesso, ou mesmo dentro da pasta de instalação do SWI-Prolog. Por exemplo,

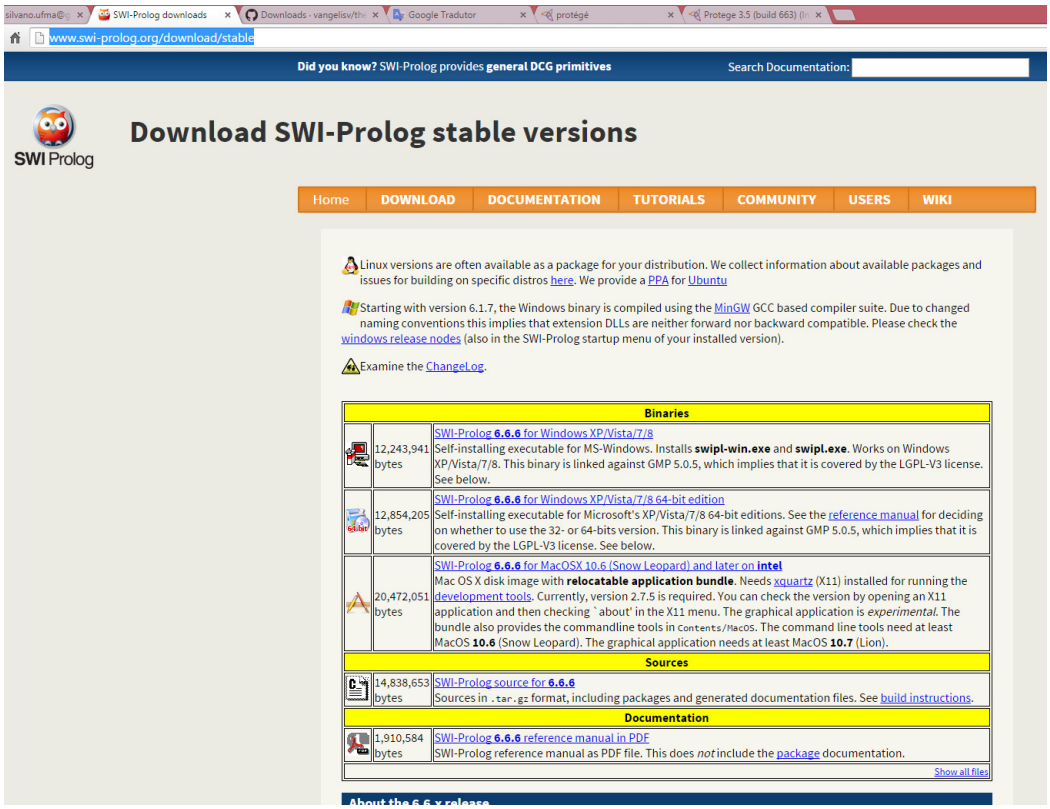
C:\Program Files\swipl\library

onde, no Windows, a pasta thea2 deverá ter suas propriedades de segurança mudadas para Controle total do Acesso em usuários do sistema. Conforme figura abaixo:



E mesmo sem precisar de um instalador para a biblioteca THEA, o *software* SWI-Prolog é necessário para executar o THEA. Portanto, segue o link de download do SWI-Prolog:

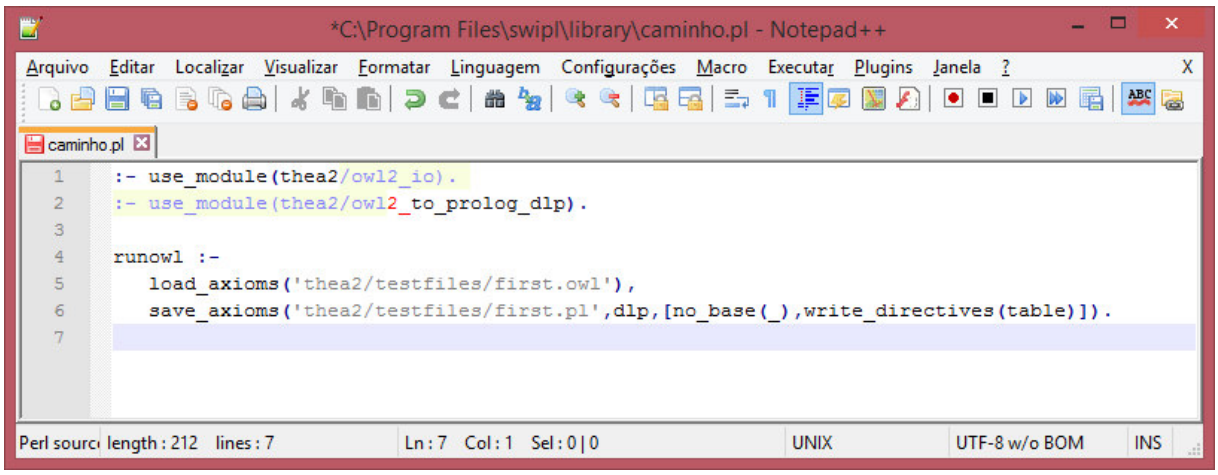
http://www.swi-prolog.org/download/stable



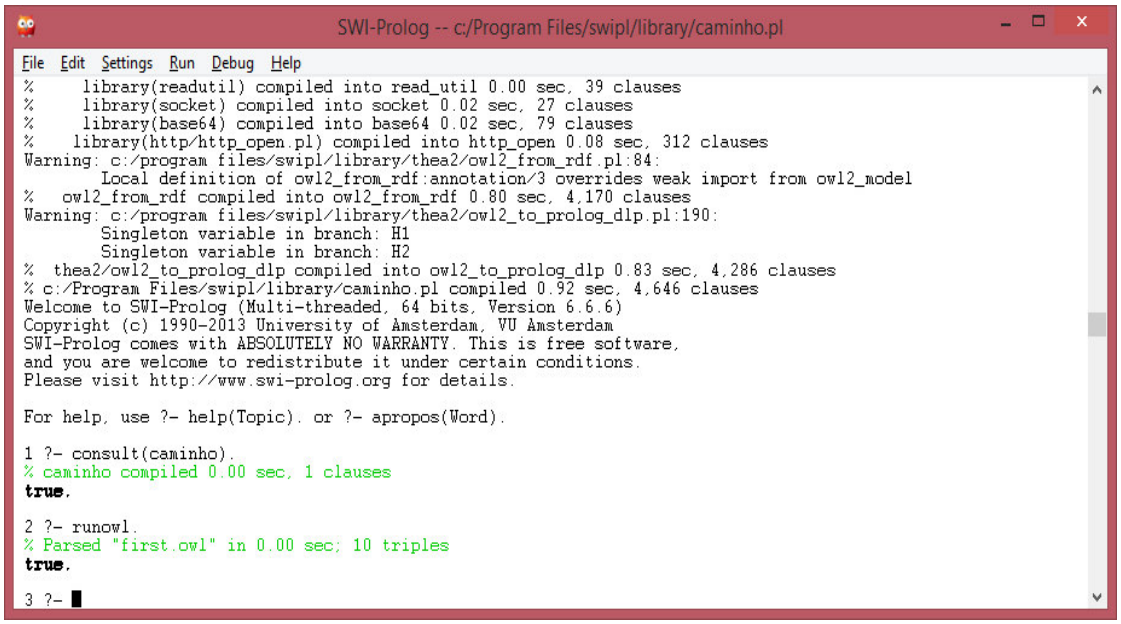
Na tela acima deverá ser escolhida a versão a ser baixada de acordo com o Sistema Operacional usado.

Para executar a aplicação e converter uma ontologia OWL para PL, os seguintes passos deverão ser feitos:

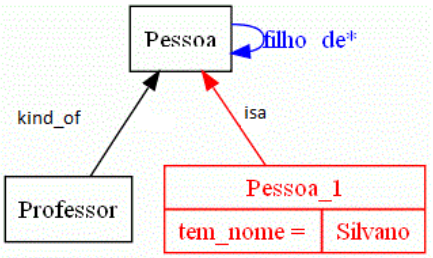
1. Criar um arquivo de nome caminho.pl, na pasta C:\Program Files\swipl\library, com o seguinte conteúdo:



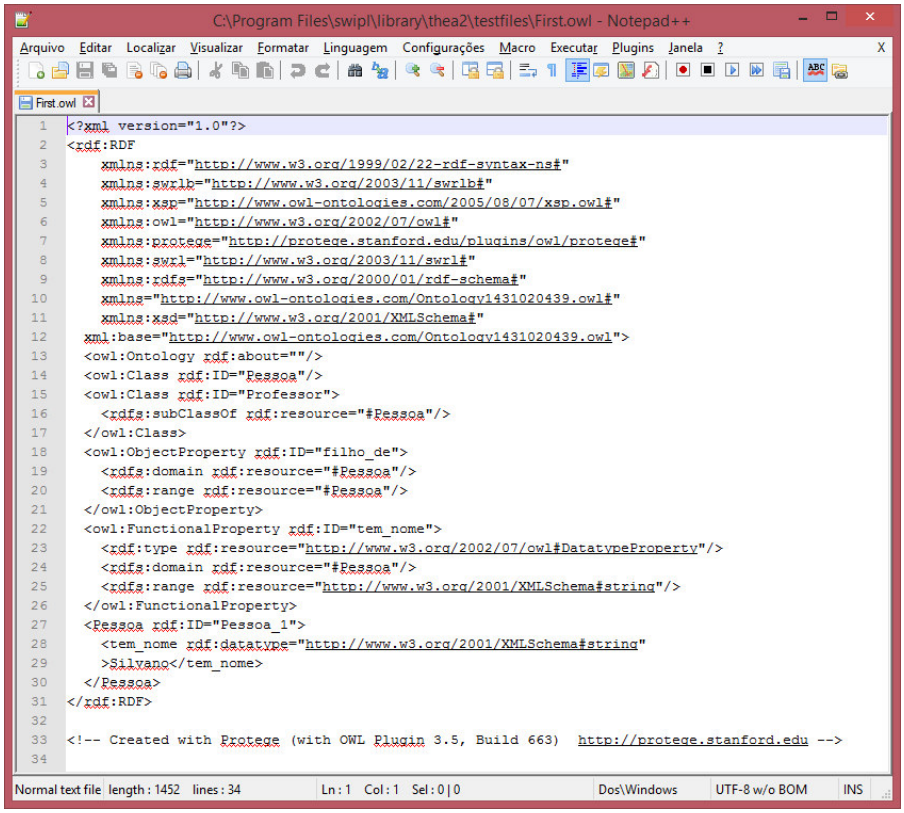
2. Trocar o nome dos arquivos.owl e .pl, das linhas 5 e 6, para os que estiver usando.
3. Colocar o arquivo *.owl na pasta thea2\testfiles.
4. Executar o arquivo caminho.pl a partir do iniciador de arquivos como SWI-Prolog.
5. Em seguida, digitar os seguintes comandos de dentro do SWI-Prolog:
    a. consult(caminho).
    b. runowl.
6. Verificar se o resultado será True, e se o arquivo .pl foi criado na pasta testfiles.



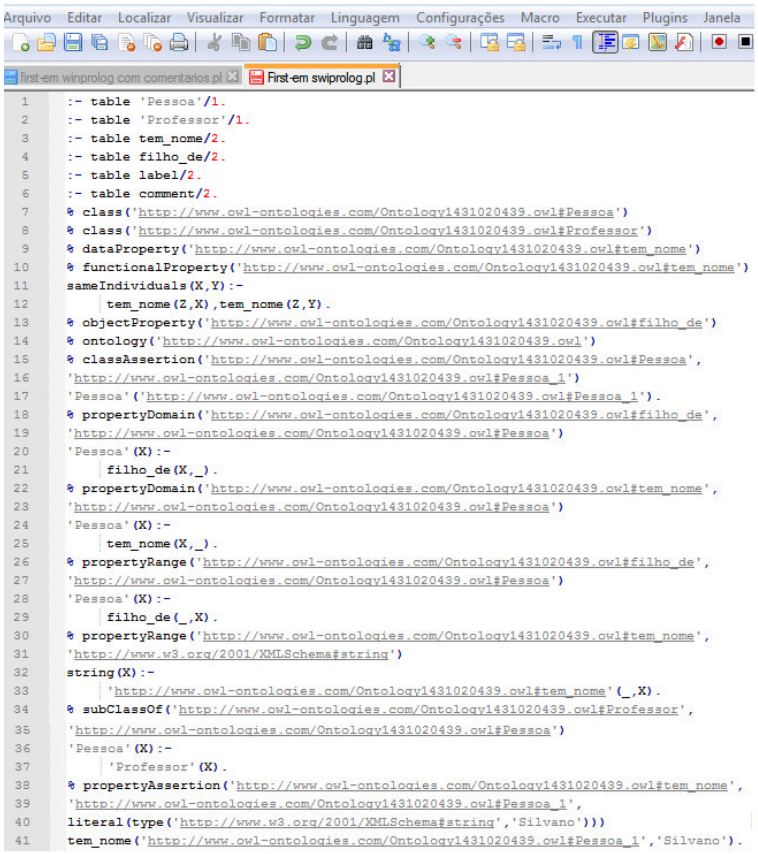
Como exemplo, tem-se a ontologia abaixo:



A ontologia OWL gerada pelo Protégé é a seguinte:

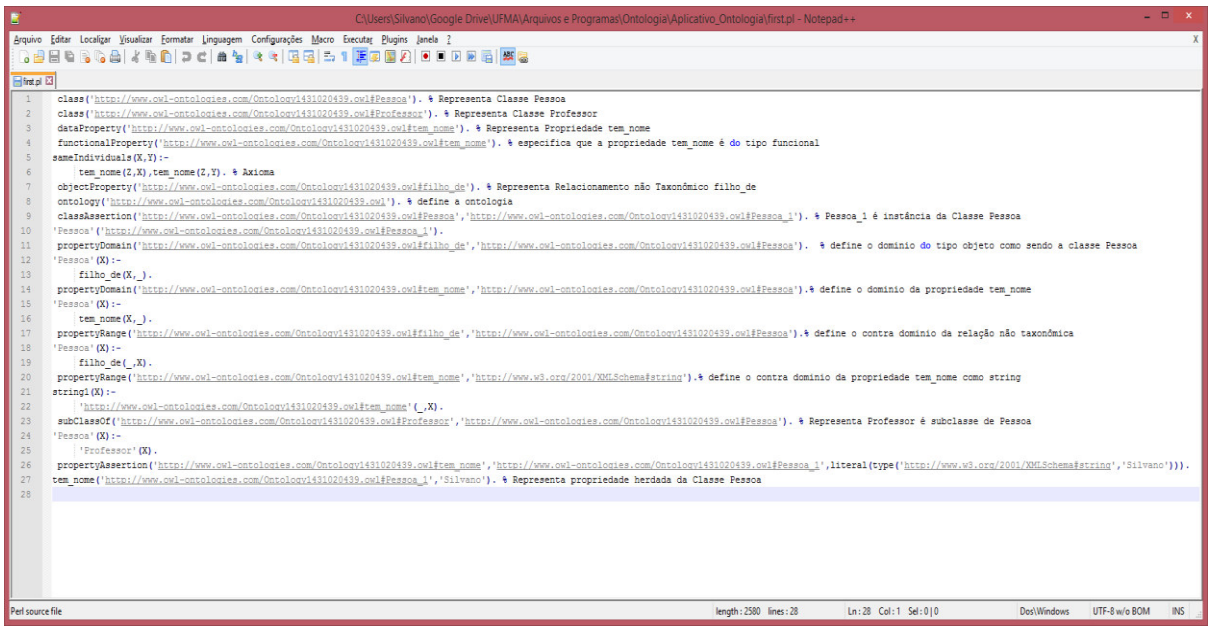


A figura abaixo mostra a ontologia após aplicar transformação de OWL para .pl usando a ferramenta THEA.



Porém, para que o Win-Prolog possa entender o resultado, é necessário fazer a limpeza de alguns caracteres a fim de que o mesmo possa ser executado.

Segue abaixo o resultado no Win-Prolog:



Segue abaixo a transcrição da figura acima:

```
class('http://www.owl-ontologies.com/Ontology1431020439.owl#Pessoa'). % Representa Classe
Pessoa
class('http://www.owl-ontologies.com/Ontology1431020439.owl#Professor'). % Representa Classe
Professor
dataProperty('http://www.owl-ontologies.com/Ontology1431020439.owl#tem_nome'). % Representa
Propriedade tem_nome
functionalProperty('http://www.owl-ontologies.com/Ontology1431020439.owl#tem_nome'). %
especifica que a propriedade tem_nome é do tipo funcional
sameIndividuals(X,Y):-
     tem_nome(Z,X),tem_nome(Z,Y). % Axioma
objectProperty('http://www.owl-ontologies.com/Ontology1431020439.owl#filho_de'). % Representa
Relacionamento não Taxonômico filho_de
ontology('http://www.owl-ontologies.com/Ontology1431020439.owl'). % define a ontologia
classAssertion('http://www.owl-ontologies.com/Ontology1431020439.owl#Pessoa','http://www.owl-
ontologies.com/Ontology1431020439.owl#Pessoa_1'). % Pessoa_1 é instância da Classe Pessoa
'Pessoa'('http://www.owl-ontologies.com/Ontology1431020439.owl#Pessoa_1').
propertyDomain('http://www.owl-
ontologies.com/Ontology1431020439.owl#filho_de','http://www.owl-
ontologies.com/Ontology1431020439.owl#Pessoa').  % define o dominio do tipo objeto como sendo
a classe Pessoa
'Pessoa'(X):-
     filho_de(X,_).
propertyDomain('http://www.owl-
ontologies.com/Ontology1431020439.owl#tem_nome','http://www.owl-
ontologies.com/Ontology1431020439.owl#Pessoa'). % define o dominio da propriedade tem_nome
'Pessoa'(X):-
     tem_nome(X,_).
propertyRange('http://www.owl-ontologies.com/Ontology1431020439.owl#filho_de','http://www.owl-
ontologies.com/Ontology1431020439.owl#Pessoa'). % define o contra dominio da relação não
taxonômica
'Pessoa'(X):-
     filho_de(_,X).
propertyRange('http://www.owl-
ontologies.com/Ontology1431020439.owl#tem_nome','http://www.w3.org/2001/XMLSchema#string'). %
define o contra dominio da propriedade tem_nome como string
```

```
string1(X):-
     'http://www.owl-ontologies.com/Ontology1431020439.owl#tem_nome'(_,X).
subClassOf('http://www.owl-ontologies.com/Ontology1431020439.owl#Professor','http://www.owl-
ontologies.com/Ontology1431020439.owl#Pessoa'). % Representa Professor é subclasse de Pessoa
'Pessoa'(X):-
     'Professor'(X).
propertyAssertion('http://www.owl-
ontologies.com/Ontology1431020439.owl#tem_nome','http://www.owl-
ontologies.com/Ontology1431020439.owl#Pessoa_1',literal(type('http://www.w3.org/2001/XMLSchema
#string','Silvano'))).
tem_nome('http://www.owl-ontologies.com/Ontology1431020439.owl#Pessoa_1','Silvano'). %
Representa propriedade herdada da Classe Pessoa
```

A figura abaixo mostra uma consulta realizada no Win-Prolog para verificação da consistência do arquivo .pl gerado.